Running Apache Kafka on Kubernetes and OpenShift

Jakub Scholz

DevConf.CZ, Brno, 26.1.2019
About me

Jakub Scholz

- Principal Software Engineer with the Red Hat AMQ Streams engineering team
- Long-term messaging specialist
- Former messaging team lead at Deutsche Börse

@scholzj
https://www.linkedin.com/in/scholzj/
The “What” ...

What is Apache Kafka

- Developed at LinkedIn back in 2010, open sourced in 2011
- Designed to be fast, scalable, durable and available
- Distributed by nature
- Data partitioning (sharding)
- High throughput / low latency
- Ability to handle huge number of consumers
The “What” …

What is Apache Kafka

- 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
- Replication
  - Each partition can exist in one or more Backup copies to achieve high availability in case of node failures
The “Why” ...

Why should you want to run Apache Kafka in Kubernetes or OpenShift?

- Designed as distributed and scalable
  - Kafka is distributed by nature and are good at scaling out
  - Workloads using Kafka are also distributed and scalable
- Kubernetes is a great abstraction
  - K8S provides primitives for running your software almost everywhere!
  - Make everything feel like a cloud!
  - Use the operator pattern to deploy cluster “on-demand”
- Why should you learn something new when you already know K8S or OpenShift?
The “How” ...

How to run Apache Kafka on Kubernetes or OpenShift?

- Using the Operator pattern
- Deploying software is easy
  - YAMLs, Helm Charts can do installation ...
  - ... but operators do it better!
- Keeping it running is hard
  - YAMLs or Helm Charts offer only little help here
  - Operators can take the applications through the whole lifecycle
    - Upgrades, cluster balancing, security
The “How” ...

How to run Apache Kafka on Kubernetes or OpenShift?

- Our answer is **Strimzi**
- Open source project licensed under Apache License 2.0
- Focuses on running Apache Kafka on Kubernetes and OpenShift:
  - Container images for Apache Kafka and Apache Zookeeper
  - Operators for managing and configuring Kafka clusters, topics or users
- Provides Kubernetes-native experience for running Kafka on Kubernetes and OpenShift

🌐 Web site: [http://strimzi.io/](http://strimzi.io/)
👩‍💻 GitHub: [https://github.com/strimzi](https://github.com/strimzi)
🐦 Twitter: [@strimziio](https://twitter.com/strimziio)
The “How” ...

How to deploy Apache Kafka using Strimzi
The “How” ...

How to update Apache Kafka using **Strimzi**

[Diagram showing the process of updating Apache Kafka using Strimzi]

- Kafka Custom Resource
- Cluster operator
- Topic & User operators
- Zookeeper cluster
- Kafka cluster
The “How” ...

How to manage Kafka users and topics

- **Operators** deploying **Operators** which deploy **Operators**? Isn’t that blasphemy?
- In Strimzi, we want more than just manage Kafka cluster!
  - Why should the users know the Kafka specific tools for managing topics or users?
  - Let’s give users ability to deploy Kafka clients “*the Kubernetes native way*”
  - Use Custom Resources and YAML to manage Kafka topics and users
  - Topic and User Operators are deployed per Kafka cluster
  - Topics and Users can be stored in YAML together with the applications using them
- **Topic Operator** is a bit special
  - Lot of applications creates topics directly in Kafka
  - Custom Resources cannot serve as a single source of truth
  - Bi-directional synchronization between Kafka topics and Custom Resources
The “How” ... How to use feature X

- Storage
- Annotations
- Mirroring
- Labels
- Affinity
- Tolerations
- Encryption
- Metrics
- CPU and RAM
- Healthchecks
- Logging
- Authorization
- Scale Up
- Off cluster access
- Scale Down
- Prometheus
- ImagePullSecrets
- Topic
- Zookeeper
- ImagePullSecrets
- Grafana
- Source2Image
- JVM Configuration
- Configuration
- Users
- Secrets
- Authentication
- Authorization
- Ingress
- Upgrades
- Secrets
- Labels
- Prometheus
- Kafka Connect
- HA
- Pod Disruption Budgets
- Network Policies
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prometheus
- Prom
The “Challenges” ...

Challenges of running stateful applications on Kubernetes and OpenShift

Stateful applications are hard ...

... why? We have StatefulSets!
The “Challenges” ...

Challenges of operating Apache Kafka on Kubernetes and OpenShift

- StatefulSets just give the pods a stable address
- Most stateful applications have their own way how to:
  - Handle the discovery of the cluster nodes
  - Elect leaders
  - Communicate within a cluster
- For Kafka, this is handled via Zookeeper dependency
  - Zookeeper configuration needs a list of all nodes in its config file
  - Scaling Zookeeper without losing quorum is sometimes impossible
  - Once Zookeeper cluster is running, Kafka cluster uses it for all the tasks above
The “Challenges” ...
Challenges of operating Apache Kafka on Kubernetes and OpenShift

- Strimzi operators need to be able to handle wide range of use cases
  - Run win development with Minishift and minimum of resources ...
  - ... but also in production with many dedicated nodes and 100s TBs of disk space
  - Be easy to use for a developer who just wants to run Kafka ...
  - ... while supporting all the important options to give the flexibility to expert users
  - Supporting many options and features gives users the flexibility ...
  - ... but can make it very hard to use and read the Custom Resources
The “Challenges” ...
Challenges of operating Apache Kafka on Kubernetes and OpenShift

- Kubernetes and OpenShift versions
  - Kubernetes and OpenShift are still work in progress
  - Every release has a lot of cool new features
  - It is tempting to play with the latest and hottest features in Kubernetes 1.13
  - But many users are running old versions
  - It is not always easy to support the new features while keeping the compatibility with older versions

- CRDs or ConfigMaps? RBAC?
  - CRDs are nice, but most users are not allowed to install them
The “Challenges” ...

Challenges of accessing Apache Kafka
The “Challenges” ...

Challenges of accessing Apache Kafka - inside Kubernetes

Per-pod Service

Bootstrap Service

Kafka brokers

Headless Service
The “Challenges” ...

Challenges of accessing Apache Kafka - off cluster access

Per-pod Service

Bootstrap Service

Kafka brokers

Headless Service
The “Challenges” ...

Challenges of accessing Apache Kafka - off cluster access

**OpenShift Routes**
- The easiest way how to expose Kafka.
- Hiddes the traffic as TLS Passthrough.
- Kafka traffic is routed through HAProxy.

**Load Balancers**
- Requires N + 1 load balancers.
- Easy in environments such as public clouds.
- Offloads performance from HAProxy.
- Supports encrypted and unencrypted connections.

**Node Ports**
- Delivers good performance.
- Has security implications.
- Supports encrypted and unencrypted connections.
Conclusion

- And one more thing …
  - … Strimzi is written in Java!

- The demos from this talk can be found here:

🌐 Web site: [http://strimzi.io/](http://strimzi.io/)
 чел GitHub: [https://github.com/strimzi](https://github.com/strimzi)
🐦 Twitter: [@strimziio](https://twitter.com/strimziio)
THANK YOU

plus.google.com/+RedHat
linkedin.com/company/red-hat
youtube.com/user/RedHatVideos
facebook.com/redhatinc
twitter.com/RedHatNews
Backup