redhat.

Apache Kafka on OpenShift

Jakub Scholz

Basel, 22.11.2019

About me

Jakub Scholz

I work for Red Hat as Engineer in the Messaging and IoT team. I focus mainly on Apache Kafka and its integration with Kubernetes and OpenShift. Before joining Red Hat I worked as messaging and solution architect in the financial industry. I'm a member of the Project Management Committee of the Apache Qpid project and member of AMQP Technical Committees at OASIS.

Apache Kafka

- What is Apache Kafka
  - A publish/subscribe messaging system
  - A streaming data platform
  - A distributed, horizontally-scalable, fault-tolerant, commit log
- 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

Apache Kafka

- Why would you want to run Apache Kafka on OpenShift?
  - Designed as distributed and scalable
  - Workloads using Kafka are also distributed and scalable
  - Why should you learn something new when you already know OpenShift?
  - Make everything feel like a cloud!

Apache Kafka

- But running Kafka on OpenShift can be complicated
  - Stateful workloads are always hard
  - Kafka brokers require Apache Zookeeper cluster
  - Kafka clients need direct access to the individual brokers
    - Kafka brokers cannot be hidden behind load-balancers
  - Security needs to be configured across different components

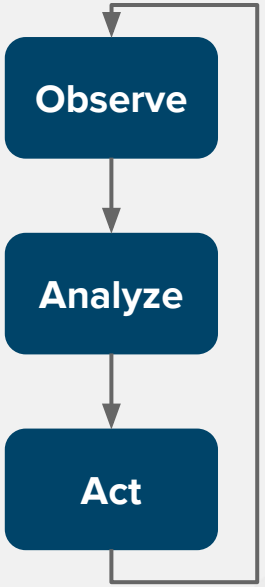
Strimzi

- 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
- Provide Kubernetes-native experience for running Kafka on Kubernetes and OpenShift

- Web site: http://strimzi.io/
- GitHub: https://github.com/strimzi
- Twitter: @strimziio

Operators

- Strimzi makes running Kafka on Openshift easy using the Operator pattern
- Operators?
  - Observe -> Analyze -> Act loop
  - User creates Custom Resource with description
  - Operator sees the resource, analyzes the changes which need to be done, and creates the Kafka cluster
  - The Custom Resource acts as a blueprint describing the cluster
  - Strimzi provides several operators



Cluster Operator

- Responsible for managing clusters
  - Kafka brokers (including Zookeeper)
  - Kafka Connect clusters
  - Kafka Mirror Maker
  - Responsible for
    - Deployment
    - Scale-up / Scale-down
    - Re-configuration

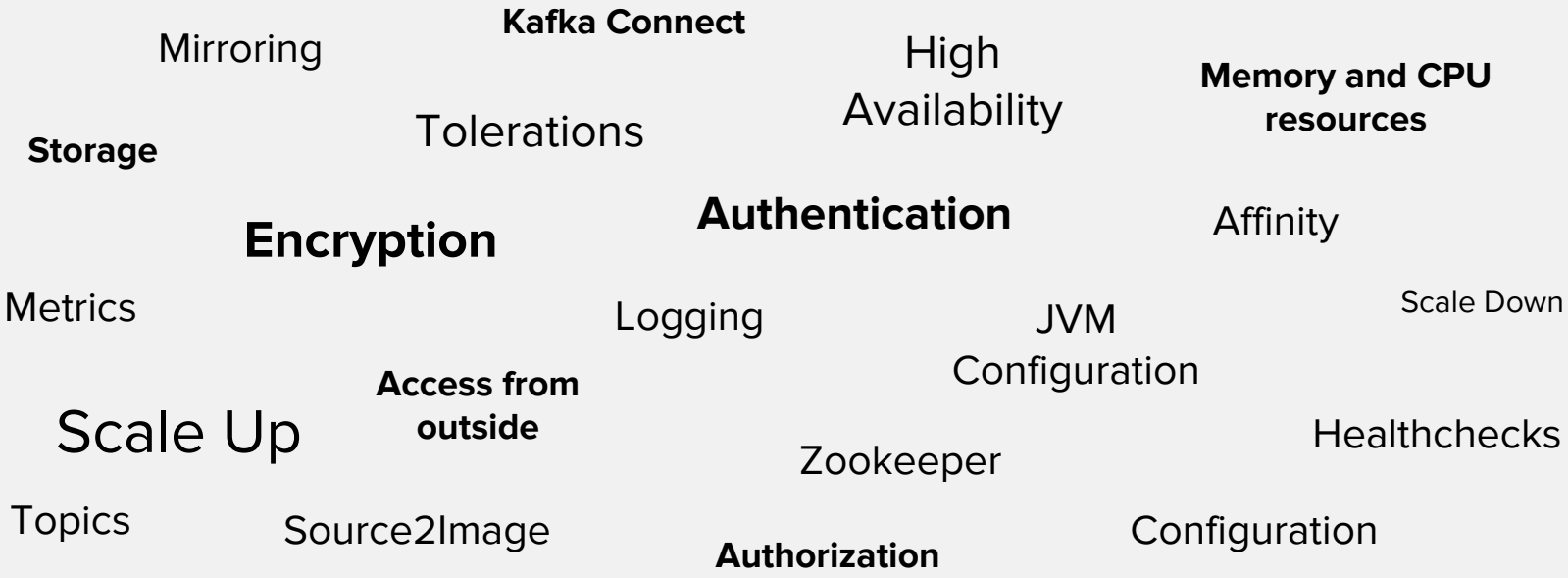
Topic Operator

- Responsible for managing Kafka topics
  - You can create, update and delete topics “the Kubernetes way”
  - No need to know Kafka commands
  - Applications can still create topics directly in Kafka
    - Topic operator synchronizes the topics bi-directionally
    - For topics created in Kafka, it will create Custom Resources
    - In case of conflicts, it will use 3-way-diff to resolve them

User Operator

- Responsible for managing users
  - Allows to create, update and delete users
  - Currently two supported authentication mechanisms
    - TLS client certificates
    - SASL SCRAM-SHA-512 (username and password based authentication)
  - Authorization manages using Kafka ACL plugin
    - Allowed / Denied operations can be defined together with the user

Main features



Demo time ...

http://jsch.cz/basel

AMQ Streams

STRIMZI

- Enterprise Distribution of Apache Kafka
- Available as
  - Standalone for bare-metal or virtual machines
  - On OpenShift based on the Strimzi project
- AMQ Streams 1.0.0 is based on
  - Apache Kafka 2.0.0
  - Strimzi 0.8.1
- Website
  - https://access.redhat.com/products/red-hat-amq

kafka

redhat.

THANK YOU

plus.google.com/+RedHat

facebook.com/redhatinc

linkedin.com/company/red-hat

twitter.com/RedHatNews

youtube.com/user/RedHatVideos