Application Lifecycle Management with OpenShift Operators

Ruud Zwakenberg
Sr. Solution Architect at Red Hat
Red Hat Forum - October 9th, 2019
Your wonderful OpenShift Cluster
Your Kubernetes Operator
What you actually want...
The Automated Kubernetes Operator
What is a Kubernetes Operator?

- An **operator** is a method of packaging, deploying and managing a Kubernetes application
- Extension of the Kubernetes Cluster Control pane API
Kubernetes Resources

- ETCD config store
- Kubernetes Cluster Control Pane API
- Custom Controller
- Pods
- Templates
- Users
- Routes
- CR

Operator =
- Custom Resource Description
- Custom Controller
Example application: Apache Kafka

https://kafka.apache.org/intro
Apache Kafka Cluster
The Kafka Operator

- Installs Custom Resources Definitions (CRD’s) for
  - Kafka Cluster
  - Kafka Topics
  - Kafka Connect
  - Kafka Mirror Maker
  - Kafka Bridge
  - Kafka User

![Provided APIs]

- Kafka Connect
  - Represents a Kafka Connect cluster
  - Create New

- Kafka Connect S2I
  - Represents a Kafka Connect cluster with Source 2 Image support
  - Create New

- Kafka MirrorMaker
  - Represents a Kafka MirrorMaker cluster
  - Create New

- Kafka Bridge
  - Represents a Kafka Bridge cluster
  - Create New
Added to the Developer Catalog

Project: kafka

Developer Catalog

Add shared apps, services, or source-to-image builders to your project from the Developer Catalog. Cluster admins can install additional apps which will show up here automatically.

All Items

Languages
Middleware
Other

Search: kafka

Kafka
provided by Red Hat, Inc.
Represents a Kafka cluster

Kafka Bridge
provided by Red Hat, Inc.
Represents a Kafka Bridge cluster

Kafka Connect
provided by Red Hat, Inc.
Represents a Kafka Connect cluster

Kafka Connect S2I
provided by Red Hat, Inc.
Represents a Kafka Connect cluster with Source-to-Image support

Kafka MirrorMaker
provided by Red Hat, Inc.
Represents a Kafka MirrorMaker cluster

Kafka Topic
provided by Red Hat, Inc.
Represents a topic inside a Kafka cluster

Kafka User
provided by Red Hat, Inc.
Represents a user inside a Kafka cluster
Create Kafka cluster

```
oc apply -f kafka/kafka-ephemeral.yaml
```
<table>
<thead>
<tr>
<th>NAME</th>
<th>NAMESPACE</th>
<th>POD LABELS</th>
<th>NODE</th>
<th>STATUS</th>
<th>READINESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>amq-streams-cluster-operator-9f6b9766-fkzs6t</td>
<td>kafka</td>
<td>n... =amq-streams-cluster-operator; pod-template...=9f6b9766-fkzs6t; strimzi...=cluster-operator</td>
<td>IP-10-0-143-219.eu-central-1.compute.internal</td>
<td>Running</td>
<td>Ready</td>
</tr>
<tr>
<td>my-cluster-entity-operator-96c5c05c85c836-j7b00</td>
<td>kafka</td>
<td>pod-template...=86c5c05c85c836-j7b00; strimzi/io/...=my-cluster-entity-operator</td>
<td>IP-10-0-146-202.eu-central-1.compute.internal</td>
<td>Running</td>
<td>Ready</td>
</tr>
<tr>
<td>my-cluster-kafka-0</td>
<td>kafka</td>
<td>control...=my-cluster-kafka-0; statefulset...=my-cluster-kafka-0; strimzi/io/...=my-cluster-kafka-0</td>
<td>IP-10-0-139-220.eu-central-1.compute.internal</td>
<td>Running</td>
<td>Ready</td>
</tr>
<tr>
<td>my-cluster-kafka-1</td>
<td>kafka</td>
<td>control...=my-cluster-kafka-1; statefulset...=my-cluster-kafka-1; strimzi/io/...=my-cluster-kafka-1</td>
<td>IP-10-0-146-202.eu-central-1.compute.internal</td>
<td>Running</td>
<td>Ready</td>
</tr>
</tbody>
</table>
Operator Hub
# Red Hat Certified Operators

## DEVOPS
- dynatrace
- Kong
- joget
- opsmx
- Citrix

## APM
- AppDynamics
- Instana
- New Relic
- Sysdig
- turbonomic
- sematext

## DATA SERVICES
- Gigaspaces
- Hazelcast
- PlanetScale
- Percona
- ProphetStor
- Starburst

## DATABASE
- MemSQL
- Couchbase
- mongoDB
- NuDB
- PingCAP
- Crunchy Data
- MariaDB

## SECURITY
- aqua
- anchore
- Black Duck Synopsys
- Tremolo Security
- tufin
- Twistlock

## STORAGE
- Robin
- StorageOS
- Portworx
Operator Hub

Discover Operators from the Kubernetes community and Red Hat partners, curated by Red Hat. Operators can be installed on your clusters to provide optional add-ons and shared services to your developers. Once installed, the capabilities provided by the Operator appear in the Developer Catalog, providing a self-service experience.

- All Items
- A/Machine Learning
- Application Runtime
- Big Data
- Cloud Provider
- Database
- Developer Tools
- Integration & Delivery
- Logging & Tracing
- Monitoring
- Networking
- OpenShift Optional
- Security
- Storage
- Streaming & Messaging

- AMQ Certificate Manager provided by Red Hat
- AMQ Interconnect provided by Red Hat
- AMQ Online provided by Red Hat, Inc.
- AMQ Streams provided by Red Hat, Inc.

- AMQ Certificate Manager
- AMQ Interconnect
- AMQ Online
- AMQ Streams

- Community
- Installed

Smart Gateway Operator provided by Red Hat
Operator Life Cycle Manager
Operator Life Cycle Manager

Install & update across clusters

Operator manifest → Cluster catalog

Namespace A

Namespace B

https://docs.openshift.com/container-platform/4.1/applications/operators/olm-understanding-olm.html
Operator Life Cycle Manager

YourOperator v1.1.2 Bundle

Deployment
Role
ClusterRole
RoleBinding
ClusterRoleBinding
ServiceAccount
CustomResourceDefinition

Operator Deployment
Custom Resource Definitions
RBAC
API Dependencies
Update Path
Metadata
Operator Life Cycle Manager

Operator Catalog

OPERATOR LIFECYCLE MANAGER

Subscription for YourOperator

Version

YourOperator v1.1.2

YourOperator v1.1.3

YourOperator v1.2.0

YourOperator v1.2.2

Time
Operator Catalog

Operator Life Cycle Manager

Subscription for YourOperator

Version

YourOperator v1.1.2
YourOperator v1.1.3
YourOperator v1.2.2
YourOperator v1.2.0
YourOperator v1.3.1
YourApp v3.0
YourApp v3.1

Time
Create your own Operator
Build your own Operator

Build operators from Helm chart, without any coding

Build operators from Ansible playbooks and APBs

Build advanced operators for full lifecycle management
Operator Maturity Model

Phase I: Basic Install
- Automated application provisioning and configuration management

Phase II: Seamless Upgrades
- Patch and minor version upgrades supported

Phase III: Full Lifecycle
- App lifecycle, storage lifecycle (backup, failure recovery)

Phase IV: Deep Insights
- Metrics, alerts, log processing and workload analysis

Phase V: Auto Pilot
- Horizontal/vertical scaling, auto config tuning, abnormal detection, scheduling tuning
Build your own Operator

- Testing is extremely important for Operators, now we have a testing framework built in
- SDK includes a “scorecard” to ensure your Operator is technically correct
- Support new RHEL universal base image

![Diagram]

- Helm SDK
- Ansible SDK
- Go SDK

“No code” Operator: Perfect for Ops, Full power & features

- Testing framework & Scorecard
- Red Hat verification
- OperatorHub
Contribute to the Operator Hub

- Create an Operator with the SDK
- Package your Operator
- Preview your Operator
- Submit your Operator
Automated Dependency Resolution

YourOperator v1.1.2 requires Jaeger Operator jaeger.jaegertracing.io/v1 resolves to Operator Framework Dependency Graphs

installed by OPERATOR LIFECYCLE MANAGER

CockroachDB Operator

cockroachdb.charts.helm.k8s.io/v1alpha1

installed by
Demo time
Installing an Operator

- Installing a Kafka Operator
Learn more

https://www.openshift.com/learn/topics/operators
Questions?

linkedin.com/company/red-hat

facebook.com/redhatinc

youtube.com/user/RedHatVideos

twitter.com/RedHat