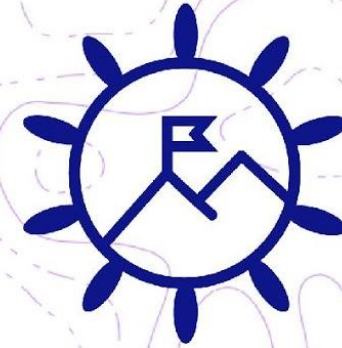




HELM
SUMMIT

At the Helm without a Steering Wheel

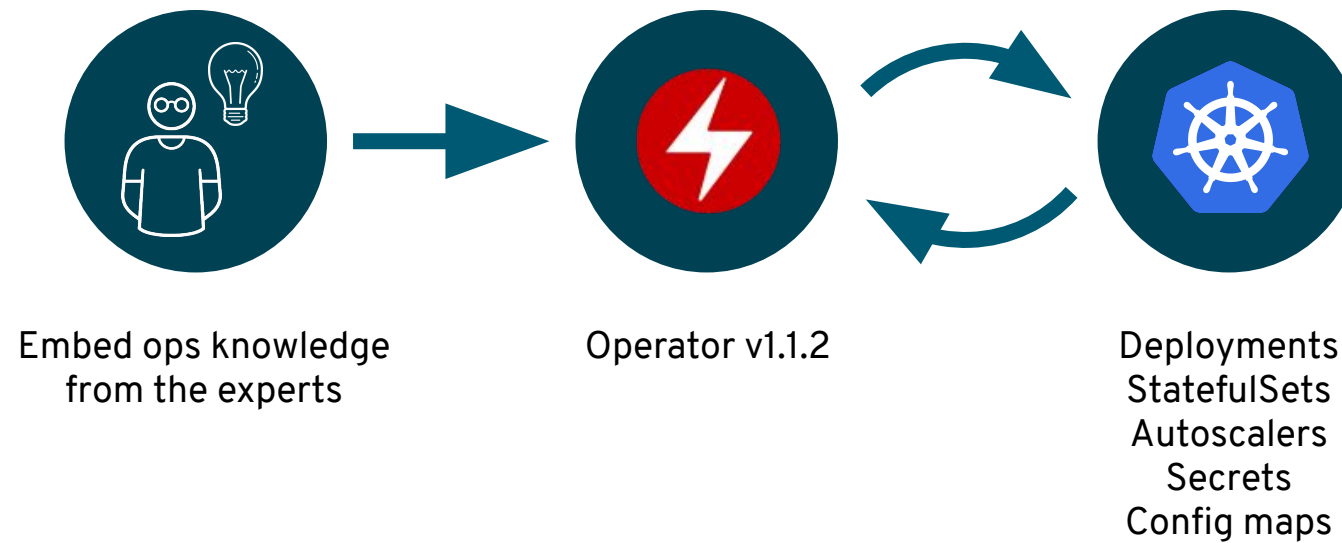
Your Chart as a Kubernetes Operator

Daniel Messer

Product Manager, Red Hat

dmesser@redhat.com

What is an Operator?



Why run an Operator?



Day-2 Operations

Operators allow implementation of typical lifecycle events like Backup / Restore, Failover / Failback and management of stateful applications that require complex orchestration



Managed Service-like UX

Users don't need to be an expert in how the managed application is deployed and run. Operator provides access to it with just the right amount of detail.



Resiliency

Operators constantly reconcile the desired state. Configuration drift is automatically remediated.

Why run your chart as an Operator?



User interface

Interact with your chart via Kubernetes Custom Resources and kubectl tooling. No client-side utilities required.



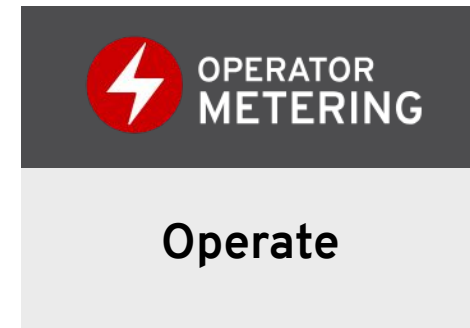
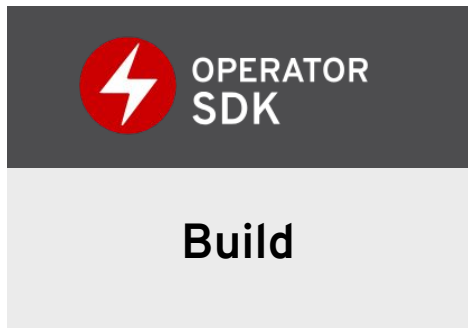
No Tiller

The chart only runs with the RBAC for the resources it's actually using. Access to the chart is subject to standard Kubernetes RBAC.



Resilient Releases

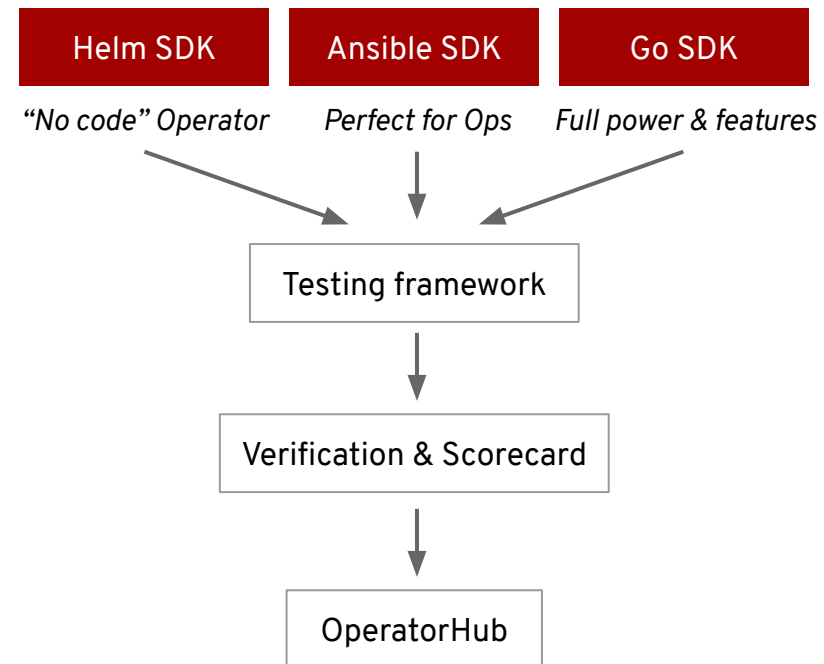
The release is constantly applied. Manual changes to the deployed resources are rolled back.



<https://github.com/operator-framework/getting-started>

Operator SDK

- “No code” approach to convert any helm chart with Helm SDK
- end2end testing framework for Golang-based Operators
- Molecule-testing for Ansible Operators
- SDK includes a “scorecard” to ensure your Operator is behaving correctly on cluster



Helm-based Operators with the SDK

- Easiest way to get started – “no code”
- Use templating from Helm
- Connect values.yaml to Kubernetes object

```
$ operator-sdk new tomcat-operator \  
  --type=helm \  
  --helm-chart=stable/tomcat
```

Helm-based Operators with the SDK

```
tomcat-operator/  
├── build  
│   └── Dockerfile  
├── deploy  
│   ├── crds  
│   │   ├── charts_v1alpha1_tomcat_cr.yaml  
│   │   └── charts_v1alpha1_tomcat_crd.yaml  
│   ├── operator.yaml  
│   ├── role.yaml  
│   ├── role_binding.yaml  
│   └── service_account.yaml  
├── helm-charts  
│   └── tomcat  
│       ├── Chart.yaml  
│       ├── README.md  
│       ├── charts  
│       ├── templates  
│       │   └── [...]  
│       └── values.yaml  
└── watches.yaml
```

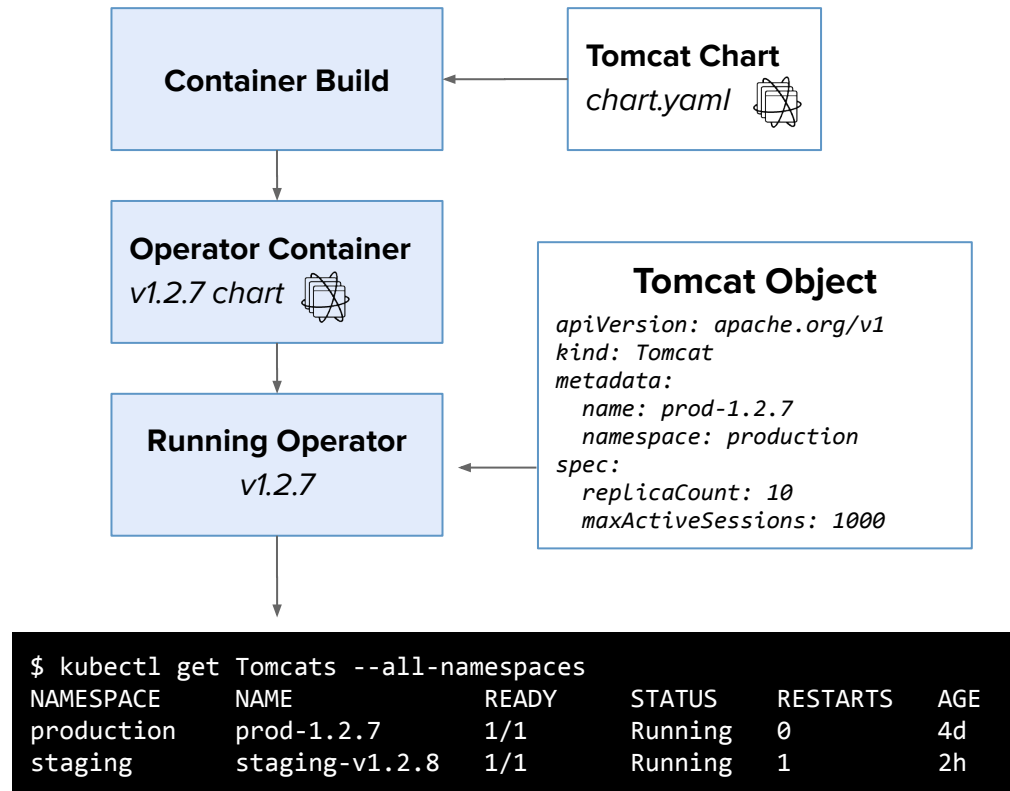
Helm-based Operators with the SDK

```
tomcat-operator/  
├── build  
│   └── Dockerfile  
├── deploy  
│   ├── crds  
│   │   ├── charts_v1alpha1_tomcat_cr.yaml  
│   │   └── charts_v1alpha1_tomcat_crd.yaml  
│   ├── operator.yaml  
│   ├── role.yaml  
│   ├── role_binding.yaml  
│   └── service_account.yaml  
├── helm-charts  
│   └── tomcat  
│       ├── Chart.yaml  
│       ├── README.md  
│       ├── charts  
│       ├── templates  
│       │   └── [...]  
│       └── values.yaml  
└── watches.yaml
```

- `build/` contains Dockerfile to create the Operator image
- `deploy/` contains the Kubernetes manifest to deploy the Operator and the CRDs
 - one CRD per chart
- `helm-chart/` contains the original chart
- `watches.yaml` creates the connection between the CRD:

```
- version: v1alpha1  
  group: charts.helm.k8s.io  
  kind: Tomcat  
  chart: /opt/helm/helm-charts/tomcat
```

Helm-based Operators with the SDK



Demo

Further considerations

CHART UPDATES

Updated charts are simply copied into your SDK project
Updated charts should yield new versions of the Operator
New values should yield new versions of the Custom Resource Definition

SUPPORTED CHART FEATURES

Helm-based Operators use Tiller rendering engine internally
Composite Charts are supported
Chart hooks are supported

TAILORED RBAC

Operator-SDK generates default RBAC to allow accessing your CRD
Operator-SDK will attempt to derive required RBAC by analyzing your chart

OPERATOR UPDATES

New versions of your Operator should support older CRs
Operator Lifecycle Manager allows you to control the supported update paths
Operator Lifecycle Manager allows for over-the-air Operator updates

Publish & Share your Operators

OperatorHub.io

Search OperatorHub...

Contribute

Welcome to OperatorHub.io

OperatorHub.io is a new home for the Kubernetes community to share Operators. Find an existing Operator or list your own today.

CATEGORIES

AI/Machine Learning

Application Runtime

Big Data

Cloud Provider

Database

Developer Tools

Integration & Delivery

Logging & Tracing

Monitoring

Networking

OpenShift Optional

Security

Storage

Streaming & Messaging

PROVIDER

☐ Altinity (1)

☐ Amazon Web Services (1)

☐ Appsody (1)

☐ Aqua Security (1)

☐ AtlasMap (1)

Show 49 more

CAPABILITY LEVEL

☐ Basic Install (32)

☐ Seamless Upgrades (15)

☐ Full Lifecycle (15)

☐ Deep Insights (9)

☐ Auto Pilot (1)

72 ITEMS

VIEW

SORT A-Z

**Akka Cluster Operator**
provided by Lightbend, Inc.

Run Akka Cluster applications on OpenShift.

**Altinity ClickHouse Operator**
provided by Altinity

ClickHouse Operator manages full lifecycle of ClickHouse

**Apache Spark Operator**
provided by radanalytics.io

An operator for managing the Apache Spark clusters and intelligent applications that

**Appsody Operator**
provided by Appsody

Deploys Appsody based applications

**Aqua Security Operator**
provided by Aqua Security, Inc.

The Aqua Security Operator runs within Kubernetes cluster and provides a means to

**AtlasMap Operator**
provided by AtlasMap

AtlasMap is a data mapping solution with an interactive web based user interface, it

**AWS S3 Operator**
provided by Red Hat

Manage the full lifecycle of installing, configuring and managing AWS S3 Provisioning

**AWS Service Operator**
provided by Amazon Web Services, Inc.

The AWS Service Operator allows you to manage AWS

**Banzai Cloud Kafka Operator**
provided by Banzai Cloud

Installs and maintains Kafka

**Camel K Operator**
provided by The Apache Software Foundation

Apache Camel K is a lightweight integration

**CockroachDB**
provided by Helm Community

CockroachDB Operator based on the CockroachDB helm chart

**Community Jaeger Operator**
provided by CNCF

Provides tracing, monitoring and troubleshooting microservices-based

**Couchbase Operator**
provided by Couchbase

The Couchbase Autonomous Operator allows users to easily deploy, manage, and maintain

**Crunchy PostgreSQL Enterprise**
provided by CrunchyData.com

A Postgres Operator from Crunchydata.com

**Dynatrace OneAgent**
provided by Dynatrace LLC

Install full-stack monitoring of Kubernetes clusters with the Dynatrace OneAgent.

Thank you



<https://operatorhub.io>



http://youtu.be/1on_wRY2dzQ



<https://commons.openshift.org/sig/operators.html>



<https://github.com/operator-framework>