

# Operator Framework

---

Carlos Santana

STSM, Cloud Native

IBM Garage Solutions Engineering

# Operators

- Operators take human operational knowledge and encode it into software that is more easily shared.
- Operators are a method of packaging, deploying, scaling, backup/restore and managing Kubernetes cluster services.



# Operators

A method of packaging,  
deploying and managing a  
Kubernetes application

*An Operator represents human  
operational knowledge in  
software, to reliably manage  
an application*



# Operators is NOT

- Only and suggested way to build application
- Replacement for other technologies like Helm
- Standard or specification

# Operators = Kube-native Extensibility

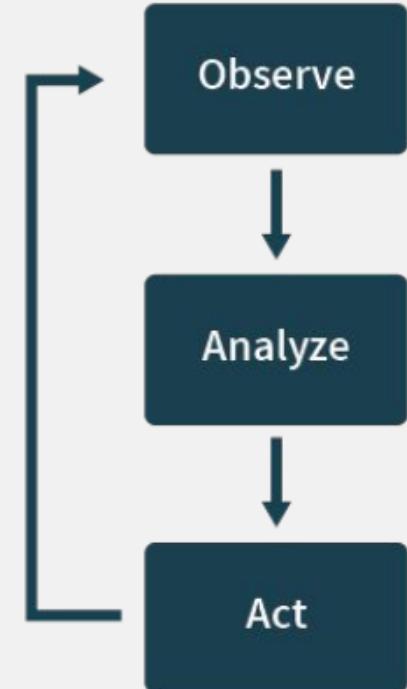
- Operators extend Kubernetes functionality
- Systematize human knowledge as code
- In a scalable, repeatable, standardized fashion
- Improve resiliency while reducing the burden on IT teams
- Ready for multi-cloud and hybrid cloud



# What does an Operator do?

# Operators

- Kubernetes API objects consist of resources and controllers
  - Resources: Pod, ConfigMap, secret, service, route, PersistentVolumeClaim, etc.
  - Controllers: Deployment, ReplicaSet, StatefulSet, DaemonSet, etc.
- Operator: Type of controller that manages resource
  - Represents human operational knowledge in software to manage application
  - Can use stock controllers or manage resource itself
  - Operators take advantage of custom resource definitions (CRDs)



# Install

- Installing cluster software most common use case for Operators.
- **How do I install x?**
  - Used to be:
    - “Ask Joe the Admin” or “Use Tiller and Helm CLI”
  - Today is simpler:
    - “Use an Operator from OperatorHub”

# Why use Operators?

- Operator include logic, controller long running process to perform.
- Operators can execute Day 2 operations, like upgrades, failover, or scaling.
- Operators is constantly watching your cluster's desired state for the software installed.

# Helm

- Helm is now a CLI that deploys k8s manifests(ie YAMLs), focuses on templatization, reusability for day 1 operations
- Operators on the other hand focus on managing workflows for day 2 operations including stateful/complex services.
- Operators can be developed using an existing Helm Chart



# Operator Framework

<https://coreos.com/blog/introducing-operator-framework>  
<https://github.com/operator-framework>

- **Operator SDK** - Allows developers to build, package and test an Operator based on your expertise without requiring all the knowledge of Kubernetes API complexities
- **Operator Lifecycle Manager (OLM)** - Helps you to deploy, and update, and generally manage the lifecycle of all of the Operators (and their associated services) running across your clusters
- **OperatorHub.io** - Publishing platform for Kubernetes Operators, allows for easy discovery and install of Operators using a graphical user interface

# Use Operators or Develop Operators?

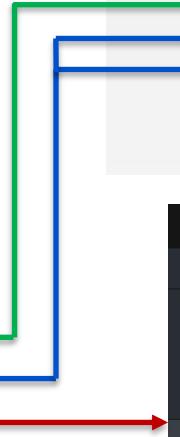
- Most users of OpenShift will use existing Operators.
- Applications will leverage the services (ie. Database, Cache, ServiceMesh, Pipelines, Kafka, etc..) provided by the Operators.
- Customers Applications don't get deployed via Operator; You can say they use the pre-installed "Operator" the Kubernetes Deployment
- Vendors, ISVs, OpenSource projects are the ones developing Operators today.

# OperatorHub

- Community catalog on operatorhub.io
- OperatorHub on cluster accessible to admins
- Discovery/install of all optional components and apps
- Upstream and downstream content
- ISV partners will support their Operators

## TYPES OF OPERATORS

ISV Partners  
Community  
Red Hat Products



The screenshot displays two main views of the OperatorHub. On the left, the 'Community' catalog is shown, featuring a search bar, categories like API Management, Application Runtime, and Business Process Management, and a grid of operator cards. On the right, the 'Red Hat OperatorHub' is shown within the OpenShift Container Platform interface, with a sidebar menu for administrators and a list of installed operators such as 3scale Operator, AMQ Broker, and API Designer.

**Community Catalog (operatorhub.io)**

- Home > Software > OpenShift
- Search OpenShift Operators
- CATEGORIES: AI/Machine Learning, Application Runtime, Big Data, Cloud Provider, Database, Developer Tools, Integration & Delivery, Logging & Tracing, Monitoring, OpenShift Optional, Security, Storage, Streaming & Messaging
- PROVIDERS: Akka, Anchore, Apache CouchDB, Apache Spark Operator, Aqua Security Operator, Argo CD
- 103 ITEMS
- OperatorHub.io
- Welcome to OperatorHub.io
- OperatorHub is a new home for the Kubernetes community to share Operators. Find an existing Operator or fill your own today.

**Red Hat OperatorHub (OpenShift Container Platform)**

- Administrator
- Home
- Dashboards
- Projects
- Search
- Explore
- Events
- Operators
- OperatorHub
- Installed Operators
- Workloads
- Networking
- Storage
- Builds
- Monitoring
- Compute
- Administration

You are logged in as a temporary administrative user. Update the cluster OAuth configuration to allow others to log in.

Project: all projects

OperatorHub

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	All Items
Filter by keyword...	30 Items
3scale Operator	AMQ Broker
provided by Red Hat	provided by Red Hat, Inc.
3scale Operator to provision 3scale and publish/manage API	AMQ Broker Operator provides the ability to deploy and manage stateful AMQ Broker broker clusters
AMQ Certificate Manager	AMQ Interconnect
provided by Red Hat	Layer 7 Networking
AMQ Online	Business Automation
provided by Red Hat, Inc.	provided by Red Hat, Inc.
AMQ Streams	Business Automation Operator for deployment and management of RHPM/RHDM environments
Red Hat AMQ Streams is a massively scalable, distributed, and high performance data stream	API Designer
Provides messaging as a managed service on OpenShift	Manages the installation and upgrades of the Fuse API Designer

# Working with Operators

- Search

The screenshot shows the Red Hat OpenShift Container Platform interface. The top navigation bar includes the Red Hat logo, 'OpenShift Container Platform', and user information ('kube:admin'). The left sidebar has a dark theme with a 'Administrator' dropdown, followed by sections for Home, Dashboards, Projects, Search, Explore, Events, Operators (selected), OperatorHub (selected), Installed Operators, Workloads, Networking, Storage, Builds, Monitoring, and Compute. The main content area has a blue header bar with the message: 'You are logged in as a temporary administrative user. Update the cluster OAuth configuration to allow others to log in.' Below this is a dropdown menu for 'Project: all projects'. The central part of the screen is titled 'OperatorHub' and contains a message: '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.' On the left, there are two tabs: 'All Items' (selected) and '0 items'. The main list shows operators categorized under 'All Items': AI/Machine Learning, Application Monitoring, Application Runtime, Big Data, Cloud Provider, Database, Developer Tools, Integration & Delivery, Logging & Tracing, Monitoring, Networking, OpenShift Optional, Security, Storage, Streaming & Messaging, and Other. A specific operator, 'Elasticsearch Operator' (provided by Red Hat, Inc.), is highlighted with a larger preview box. This box contains the operator's icon (a lightning bolt), its name, provider information, and a brief description: 'The Elasticsearch Operator for OKD provides a means for configuring and managing an Elasticsearch cluster for'. To the right of this are other operator cards: Jaeger Operator (provided by Red Hat, Inc.), Kiali Operator (provided by Red Hat, Inc.), OpenShift Pipelines Operator (provided by Red Hat), Red Hat OpenShift Service Mesh Operator (provided by Red Hat, Inc.), OpenShift Serverless Operator (provided by Red Hat, Inc.), Prometheus Operator (provided by Red Hat), and Grafana Operator (provided by Red Hat). Each card includes the operator's icon, name, provider information, and a brief description.

# Working with Operators

- Search
- Click install

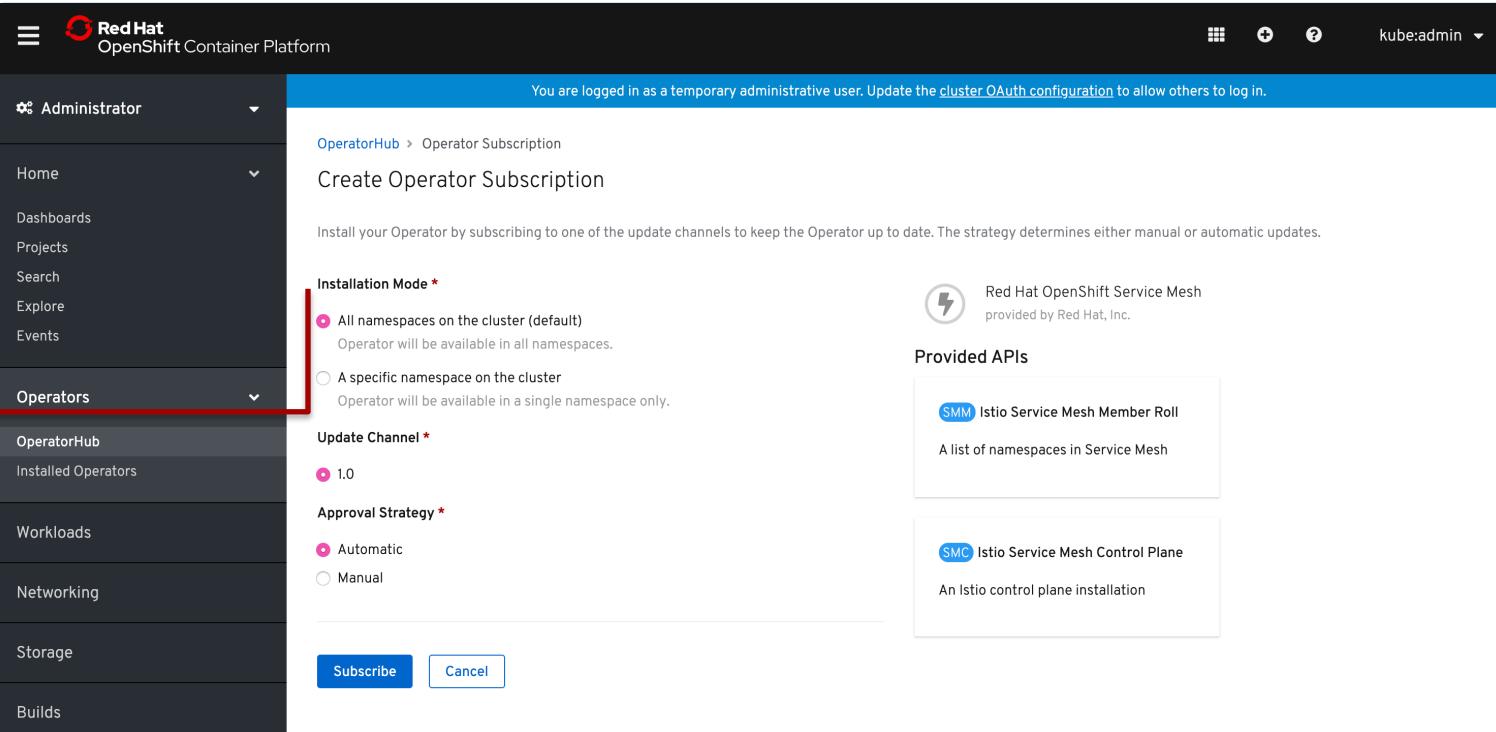
The screenshot shows the Red Hat OpenShift Container Platform web interface. The left sidebar is collapsed, and the top navigation bar includes the Red Hat logo, 'OpenShift Container Platform', and a user dropdown for 'kube:admin'. The main content area has a header 'Red Hat OpenShift Service Mesh' with a version of '1.0.4 provided by Red Hat, Inc.' Below this is a large 'Install' button. To the right of the button, detailed information about the operator is displayed:

OPERATOR VERSION	1.0.4
PROVIDER TYPE	Red Hat
PROVIDER	Red Hat, Inc.
REPOSITORY	<a href="https://github.com/maistra/a/istio-operator">https://github.com/maistra/a/istio-operator</a>
CONTAINER IMAGE	<a href="registry.redhat.io/openshift-service-mesh/istio-rhel8-operator:1.0.4">registry.redhat.io/openshift-service-mesh/istio-rhel8-operator:1.0.4</a>
CREATED AT	2020-01-07T17:54:30UTC
SUPPORT	Red Hat, Inc.

The central area is titled 'OperatorHub' and contains a search bar and a list of categories: All Items, AI/Machine Learning, Application Monitoring, Application Runtime, Big Data, Cloud Provider, Database, Developer Tools, Integration & Delivery, Logging & Tracing, Monitoring, Networking, OpenShift Optional, and Security. The 'service' category is selected. On the far right, there is a vertical sidebar with a 'Kiali Operator' section.

# Working with Operators

- Search
- Click install
- Create Subscription



The screenshot shows the Red Hat OpenShift Container Platform interface. The top navigation bar includes the Red Hat logo, 'OpenShift Container Platform', and a user dropdown for 'kube:admin'. The left sidebar has a 'Operators' section highlighted with a red box, containing links for 'OperatorHub', 'Installed Operators', 'Workloads', 'Networking', 'Storage', and 'Builds'. The main content area is titled 'Create Operator Subscription' and describes installing an Operator by subscribing to an update channel. It includes fields for 'Installation Mode' (radio buttons for 'All namespaces on the cluster (default)' and 'A specific namespace on the cluster'), 'Update Channel' (radio button for '1.0'), and 'Approval Strategy' (radio buttons for 'Automatic' and 'Manual'). To the right, there's a 'Provided APIs' section with two items: 'Red Hat OpenShift Service Mesh' (SMM) and 'Istio Service Mesh Member Roll', and another for 'Istio Service Mesh Control Plane' (SMC). At the bottom are 'Subscribe' and 'Cancel' buttons.

- *Installation Mode*
  - *Missed conception is that operator control plane is install in all namespaces*
  - *Watch all or specific namespace for Operator associated APIs via k8s Custom Resource*

# Working with Operators

- Search
- Click install
- Create Subscription
- Operator Installed
- Dependency Operators automatically installed

The screenshot shows the Red Hat OpenShift Container Platform web interface. The left sidebar has a dark theme with white text. The 'Operators' menu item is highlighted with a red box. The main content area is titled 'Installed Operators'. It displays a table of operators installed in the 'openshift-operators' namespace. The table columns are: Name, Namespace, Deployment, Status, and Provided APIs. The 'Status' column includes a green checkmark icon for 'InstallSucceeded' and a blue circle icon for 'Upgrade available'. The 'Provided APIs' column lists the services each operator manages. The 'Jaeger Operator' row is highlighted with a red box.

Name	Namespace	Deployment	Status	Provided APIs
Elasticsearch Operator 4.3.0-202001211731 provided by Red Hat, Inc.	openshift-operators	elasticsearch-operator	InstallSucceeded Upgrade available	Elasticsearch
Jaeger Operator 1.13.1 provided by Red Hat, Inc.	openshift-operators	jaeger-operator	InstallSucceeded Up to date	Jaeger
Kiali Operator 1.0.9 provided by Red Hat, Inc.	openshift-operators	kiali-operator	InstallSucceeded Up to date	Kiali Monitoring Dashboard
Red Hat OpenShift Service Mesh 1.0.4 provided by Red Hat, Inc.	openshift-operators	istio-operator	InstallSucceeded Up to date	Istio Service Mesh Member Roll Istio Service Mesh Control Plane

- Service Mesh (Istio) depends on Kiali and Jaeger
- Jaeger depends on ElasticSearch

# Working with Operators

- Search
- Click install
- Create Subscription
- Operator Installed
- Dependency Operators automatically installed
- Create NS to deploy service

The screenshot shows the Red Hat OpenShift Container Platform interface. On the left, there's a sidebar with various navigation options like Home, Dashboards, Projects, Search, Explore, Events, Operators (which is currently selected), OperatorHub, and Installed Operators. Below these are sections for Workloads, Networking, Storage, and Builds. A red box highlights the 'Operators' section. In the center, a modal window titled 'Create Namespace' is open. It has fields for 'Name \*' (set to 'istio-system'), 'Labels' (set to 'app=frontend'), and 'Default Network Policy' (set to 'No restrictions'). At the bottom of the modal are 'Cancel' and 'Create' buttons. To the right of the modal, a list of existing namespaces is shown, each with a status indicator (green circle with 'NS') and an 'Active' checkbox. A red box highlights the 'istio-system' entry in this list. A callout box at the bottom right contains the following text:

- Service Mesh (Istio) as a convention use the namespace "istio-system"
- Multiple Istio control planes can be installed on an OpenShift cluster

Callout text:

- Service Mesh (Istio) as a convention use the namespace "istio-system"
- Multiple Istio control planes can be installed on an OpenShift cluster

# Working with Operators

- Search
- Click install
- Create Subscription
- Operator Installed
- Dependency Operators automatically installed
- Create NS to deploy service
- Create Custom Resource (YAML) for control plane

The screenshot shows the Red Hat OpenShift Container Platform interface. The left sidebar has a red box around the 'Operators' section, which is currently selected. The main content area shows a 'Create Service Mesh Control Plane' form. The 'namespace' field is highlighted with a red box and contains the value 'istio-system'. Below the form, a list of components is shown:

```
1 apiVersion: maistra.io/v1
2 kind: ServiceMeshControlPlane
3 metadata:
4   name: basic-install
5   namespace: istio-system
6 spec:
7   istio:
8     gateways:
9       istio-egressgateway:
10      autoscaleEnabled: false
11      istio-ingressgateway:
12      autoscaleEnabled: false
13     mixer:
14     policy:
15     autoscaleEnabled: false
16     telemetry:
17     autoscaleEnabled: false
18     pilot:
19     autoscaleEnabled: false
20     traceSampling: 100
21     kiali:
22     enabled: true
23     grafana:
24     enabled: true
25     tracing:
26     enabled: true
27     jaeger:
28     template: all-in-one
```

**• Create a Custom Resource “basic-install” for the Operator to deploy ServiceMesh Control plane components**

**• Control plane components to be deploy in namespace “istio-system”**

# Working with Operators

- Search
- Click install
- Create Subscription
- Operator Installed
- Dependency Operators automatically installed
- Create NS to deploy service
- Create Custom Resource (YAML) for control plane
- Create Custom Resource (YAML) for member roll

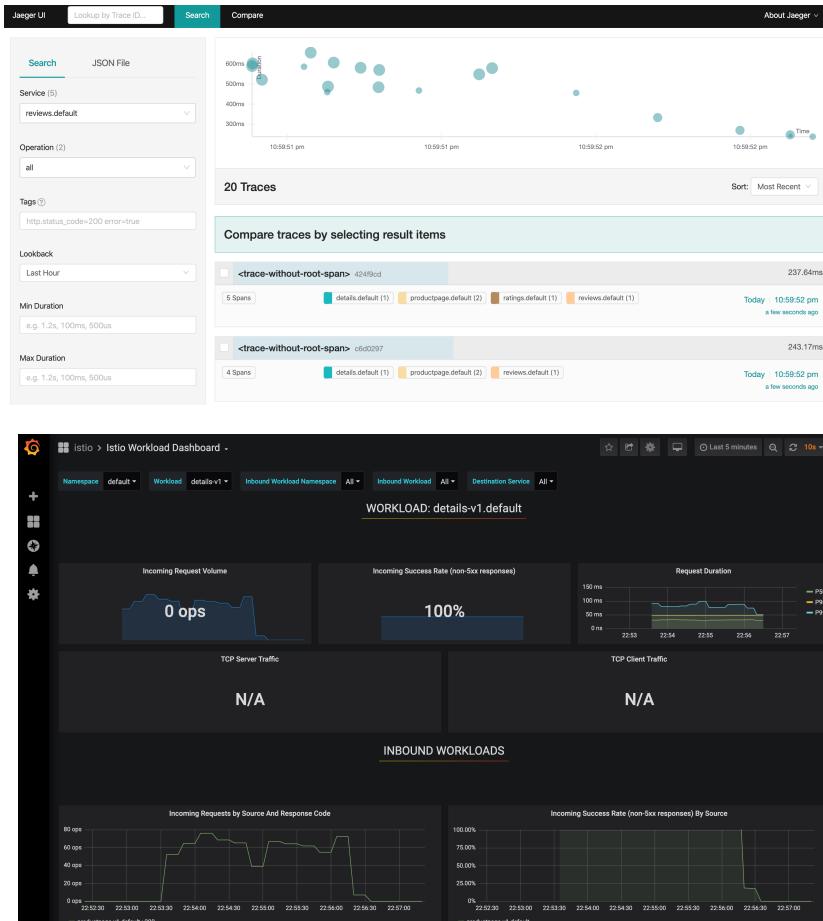
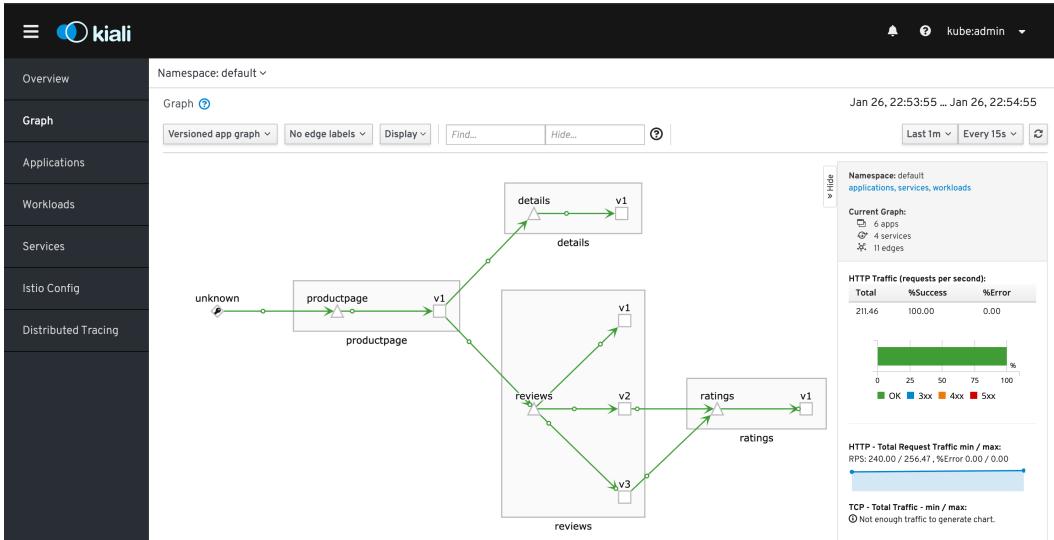
The screenshot shows the Red Hat OpenShift Container Platform interface. The top navigation bar includes the Red Hat logo, 'OpenShift Container Platform', and a user dropdown for 'kube:admin'. Below the header, a blue banner states, 'You are logged in as a temporary administrative user. Update the cluster OAuth configuration to allow others to log in.' The left sidebar has a dark theme with white text. It features a 'Project: istio-system' dropdown, followed by a list of categories: Home, Dashboards, Projects, Search, Explore, Events, Operators, OperatorHub, Installed Operators (which is highlighted with a red box), Workloads, Networking, Storage, and Builds. The main content area is titled 'Create Service Mesh Member Roll' under 'Red Hat OpenShift Service Mesh'. It contains a sub-header 'Create by manually entering YAML or JSON definitions, or by dragging and dropping a file into the editor.' Below this is a code editor window showing a YAML snippet for a 'ServiceMeshMemberRoll' object:

```
1 apiVersion: maistra.io/v1
2 kind: ServiceMeshMemberRoll
3 metadata:
4   name: default
5   namespace: istio-system
6 spec:
7   members:
8     - your-project
9     - another-of-your-projects
10    - default|
```

- *Create a Custom Resource “default” for the Istio Controller watch specific set of namespaces*
- *Namespaces included in ServiceMesh Member Roll (SMMR) will be watch for Istio Resources (ie VirtualService)*

# Service Mesh Capability

Installing Istio used to be very complex task, now with Operators is much more simplified with the ability to control version isolation



# Other Operators

## Installation Mode \*

All namespaces on the cluster (default)

This mode is not supported by this Operator

A specific namespace on the cluster

Operator will be available in a single namespace only.

**PR** default

## Update Channel \*

beta

## Approval Strategy \*

Automatic

Manual

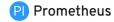
**Subscribe**

**Cancel**



Prometheus Operator  
provided by Red Hat

## Provided APIs



Prometheus  
A running Prometheus instance



Prometheus Rule  
A Prometheus Rule configures groups of sequentially evaluated recording and alerting rules.



Service Monitor  
Configures prometheus to monitor a particular k8s service



Pod Monitor  
Configures prometheus to monitor a particular pod



Alertmanager  
Configures an Alertmanager for the namespace



Grafana Operator  
provided by Red Hat

## Provided APIs



Grafana  
Represents a Grafana Instance



Grafana Dashboard  
Represents a Grafana Dashboard

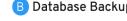


Grafana Data Source  
Represents a Grafana Data Source



PostgreSQL Operator by Dev4Ddevs.com  
provided by Dev4Ddevs.com

## Provided APIs



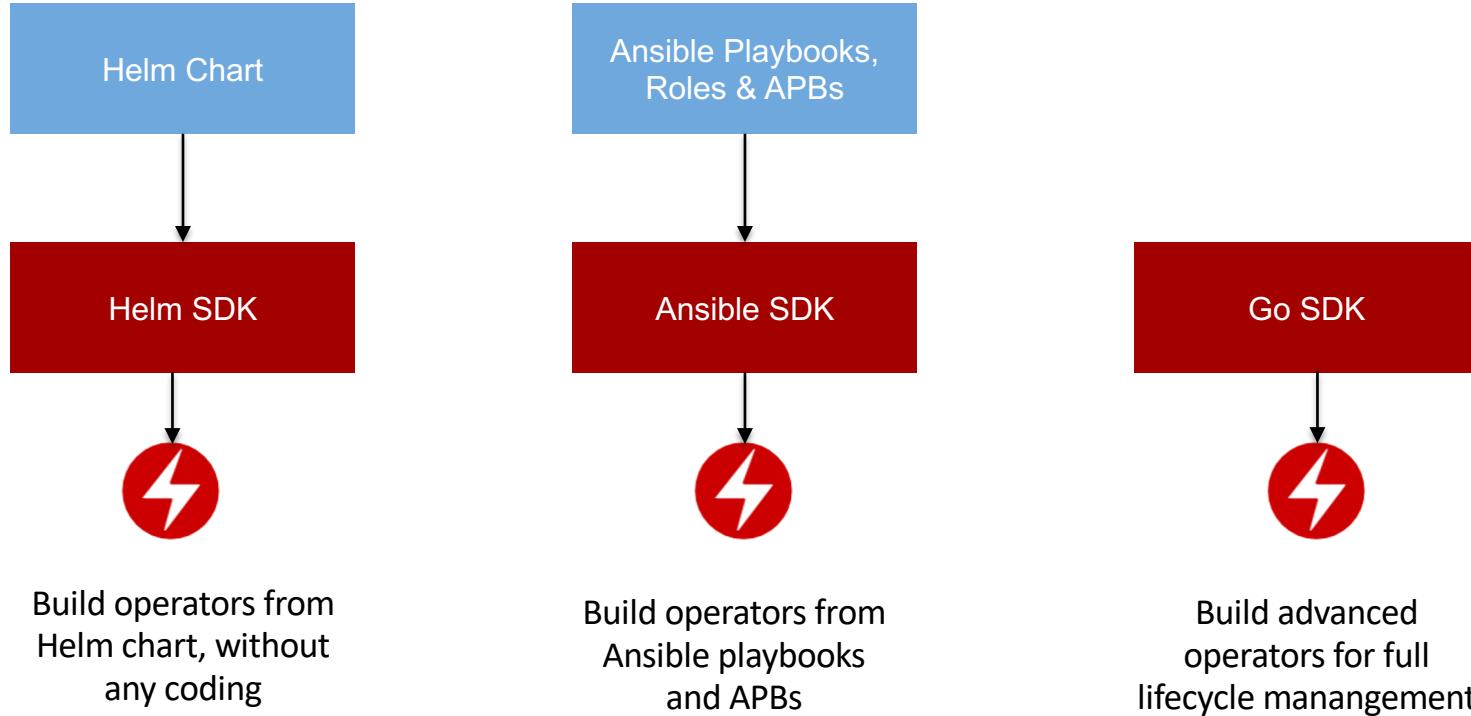
Database Backup  
Backup is the Schema for the backups API



Database Database  
Database is the Schema for the Database Database API



# Operator SDK





# Operator SDK

<https://github.com/operator-framework/operator-sdk/blob/master/doc/user-guide.md>

<https://github.com/operator-framework/operator-sdk/blob/master/doc/helm/user-guide.md>

<https://github.com/operator-framework/operator-sdk/blob/master/doc/ansible/user-guide.md>

Operator Type	What the SDK generates	What you need to define
Go Operator	<ul style="list-style-type: none"><li>General go program structure</li><li>Boilerplate code to talk to the Kubernetes API</li><li>Boilerplate code to watch for Kubernetes objects of interest</li><li>An entry point to the reconciliation loop</li><li>An example YAML files based on CRDs</li></ul>	<ul style="list-style-type: none"><li>Custom objects via CRDs</li><li>Control loop logic in Go</li><li>Potentially artistic stunts only possible by talking directly to the API from Go</li></ul>
Ansible Operator	<ul style="list-style-type: none"><li>A Go program that runs an Ansible playbook or role every time a certain type of object is detected / modified</li></ul>	<ul style="list-style-type: none"><li>Ansible playbook or role</li><li>Custom objects via CRD</li></ul>
Helm Operator	<ul style="list-style-type: none"><li>A custom object via CRD containing the same properties as the chart's values.yaml</li><li>A Go program that reads a helm chart and deploys all its resources</li><li>Watch statements to detect changes in the custom object's specification, re-deploying all resources with updated values</li></ul>	<ul style="list-style-type: none"><li>The location / repository of the helm chart</li></ul>



## Operator Capability/Maturity Model





# Operator Framework Labs

<https://learn.openshift.com/operatorframework/>

Kubernetes API  
Fundamentals

START SCENARIO

Etcd Operator

START SCENARIO

Operator SDK with Go

START SCENARIO

Operator Lifecycle  
Manager

START SCENARIO

Ansible Refresher

START SCENARIO

Ansible Kubernetes  
Modules

START SCENARIO

Ansible Operator  
Overview

START SCENARIO

Mcrouter Operator  
powered by Ansible  
Operator

START SCENARIO

Operator SDK with Helm

START SCENARIO

# Lab for today

<https://github.com/csantanapr/faststart2020-operators-lab>

<https://learn.openshift.com/playgrounds/openshift42/>

