Building Kubernetes Operators In An Ansible-Native Way

Timothy Appnel
Senior Product Manager, Ansible

tappnel@redhat.com
@appnelgroup (Twitter)
@tima (Github)

CNCF Webinar
October 9, 2019

Red Hat

Building Kubernetes Operators with Ansible

What Are Kubernetes Operators?

Why Build Operators with Ansible?

Developing Your First Operator with Ansible

Next Steps

Red Hat

What is Kubernetes?

An open source orchestration system for implementing a microservices architecture as containerized applications run and coordinated across a cluster of nodes.

Red Hat® OpenShift® is a comprehensive enterprise-grade application platform built for containers with Kubernetes at its core.

Ansible fits naturally into a Kubernetes environment

Kubernetes and Ansible

- Both help make hard things easier through automation and orchestration
- Both are very active and widely used open source projects
- Both have vibrant communities working to solve common problems
- Both use YAML to describe the desired state of the world

YAML to describe the desired state of the world

KUBERNETES/KUBECTL

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: foo
  namespace: default
data:
  color: red
```

ANSIBLE

```
- name: create foo configmap
  k8s:
    definition:
      apiVersion: v1
      kind: ConfigMap
      metadata:
        name: foo
        namespace: default
      data:
        color: "{{ color }}"
```

Templating Kubernetes resource definitions with Ansible

```
---
- name: create foo configmap
  k8s:
    definition: "{{ lookup('template', '/foo.yml') | from_yaml
}}"
```

Red Hat

What are Kubernetes Operators?

Stateless is easy,

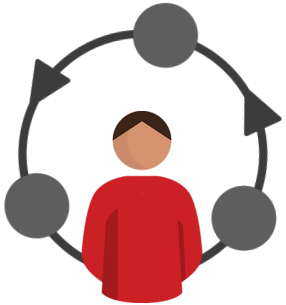
Stateful is hard

Kubernetes Operators

Operators simplify management of complex applications on Kubernetes

- Encode human operational knowledge
- **Automatically patch, upgrade, recover, and tune container-based apps and services**
- Kubernetes-native
- Purpose-built for a specific application or service
- Enable “day 2” management

Encoding and automating Ops knowledge



WITHOUT OPERATORS: REACTIVE

Continually checks for anomalies

Alert humans for response

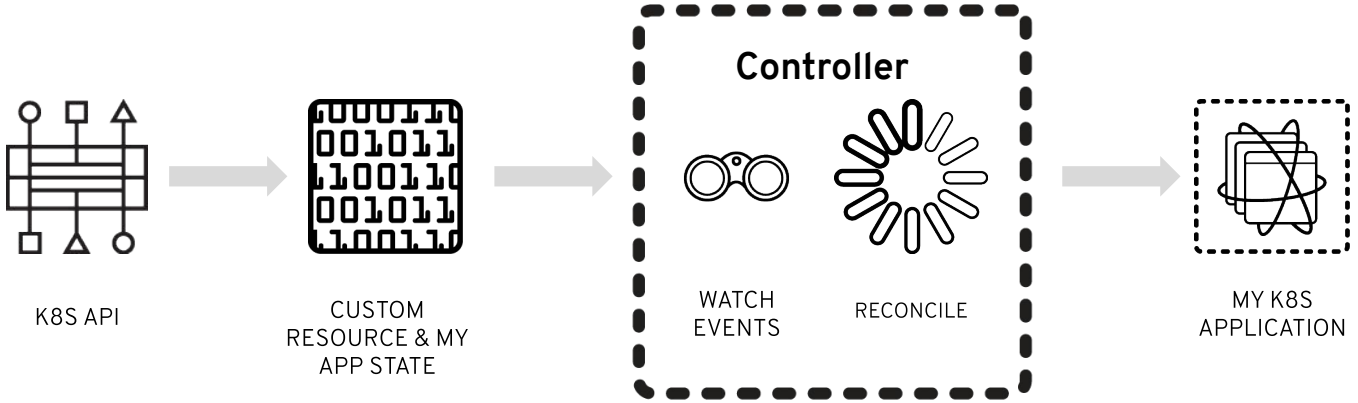
Requires manual change to fix

WITH OPERATORS: PROACTIVE

Continually adjusts to optimal state

Automatically acts in milliseconds

The Operator Pattern



Operator Framework

An open source toolkit to manage application instances
on Kubernetes in an automated, scalable way

Build Operators without specialized knowledge of the Kubernetes API

Install, update, and manage Operators and their dependencies

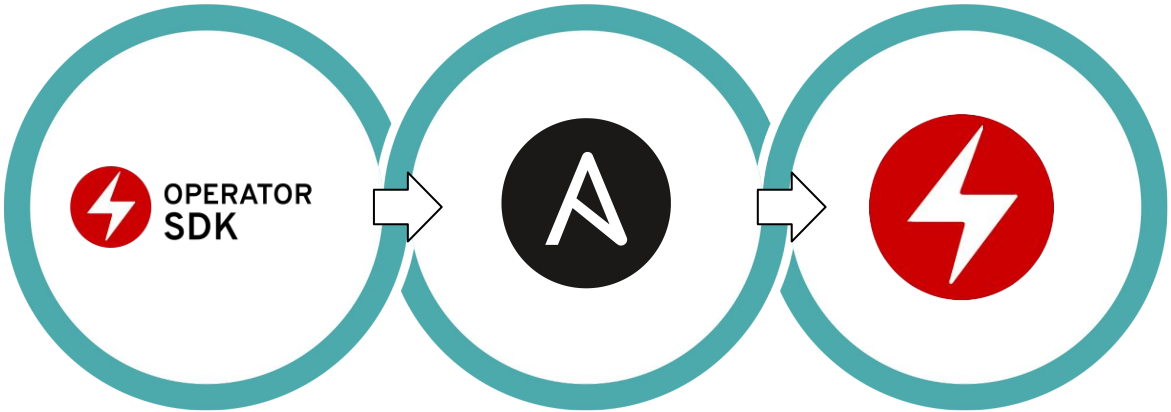
Enable usage reporting for Operators

github.com/operator-framework

Why build Operators with Ansible?

Ansible Operator SDK

Making it easier to deploy and manage Kubernetes apps in an Ansible-native way



operator-sdk new

Use the Operator SDK to create a new skeleton Operator.

Add Ansible Content

Use Ansible Roles and playbooks to manage lifecycle events for your containerized applications.

operator-sdk build

Use the Operator SDK to build and deploy your Operator to Kubernetes.

Why build Operators with Ansible?

EXISTING SKILLS & ECOSYSTEM

Same tried & trusted Ansible tooling

Utilize existing skills

Supports cloud-native & traditional IT automation with one simple language

Leverages vibrant existing ecosystem

LOWER BARRIER OF ENTRY

No programming required

Faster iterations and easier maintenance

Declarative state definitions like K8s

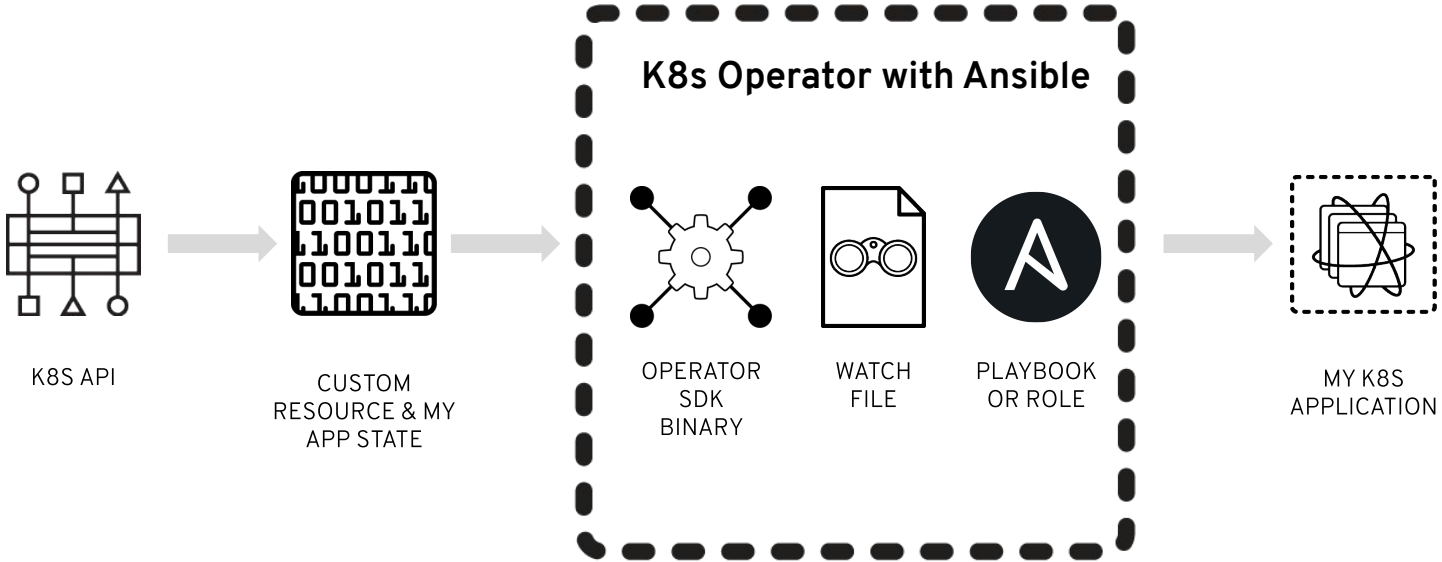
Templating of resources

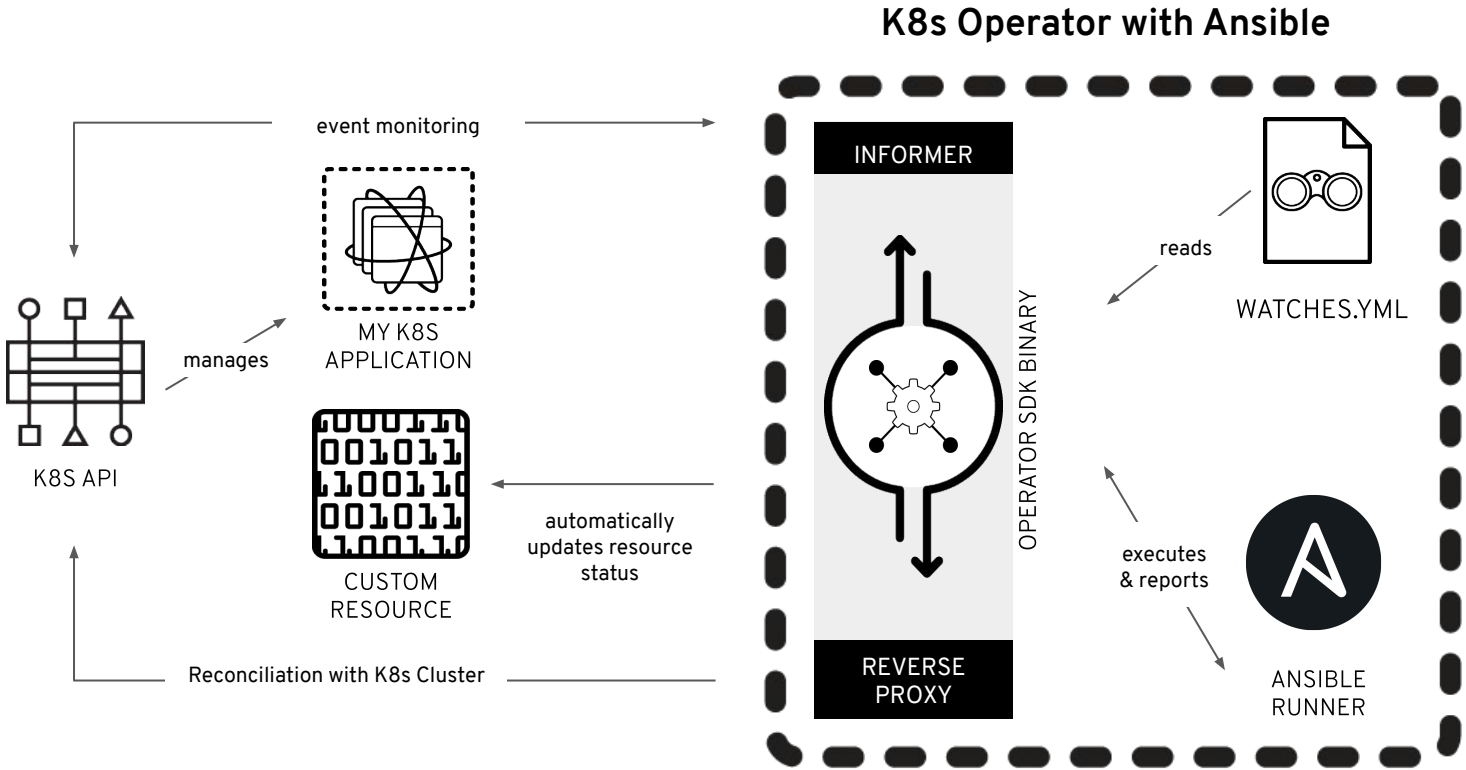
Abstraction layer & helpers that reduces necessary K8s API experience

Operator capability level

Phase I	Phase II	Phase III	Phase IV	Phase V
Basic Install	Seamless Upgrades	Full Lifecycle	Deep Insights	Auto Pilot
Automated application provisioning and configuration management	Patch and minor version upgrades supported	App lifecycle, storage lifecycle (backup, failure recovery)	Metrics, alerts, log processing and workload analysis	Horizontal/vertical scaling, auto config tuning, abnormal detection, scheduling tuning

Design overview




K8s Operator with Ansible
event monitoring
MY K8S APPLICATION
manages
K8S API
CUSTOM RESOURCE
automatically updates resource status
Reconciliation with K8s Cluster
INFORMER
REVERSE PROXY
OPERATOR SDK BINARY
reads
WATCHES.YML
executes & reports
ANSIBLE RUNNER

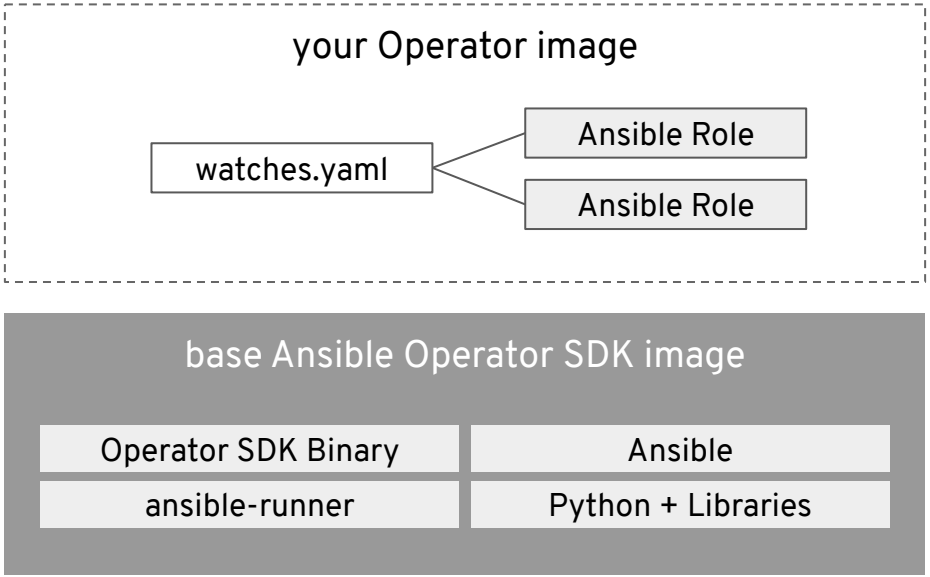
Developing your first Operator with Ansible

Red Hat

Developing your first Operator with Ansible

- Initialize Your Operator With Ansible
  - `$ operator-sdk new null-operator --api-version=cache.example.com/v1alpha1 --kind=Null --type=ansible`
- Automate With Ansible
  - Create new roles and playbooks or reuse an existing one
- Define a watches file
  - Map a Kubernetes object to your Ansible content
- Build Your Operator
  - `$ operator-sdk build null-operator:v0.0.1`
- Deploy Your Operator to a Kubernetes Cluster

Anatomy of Ansible-enabled Operator image



Examples & Demos

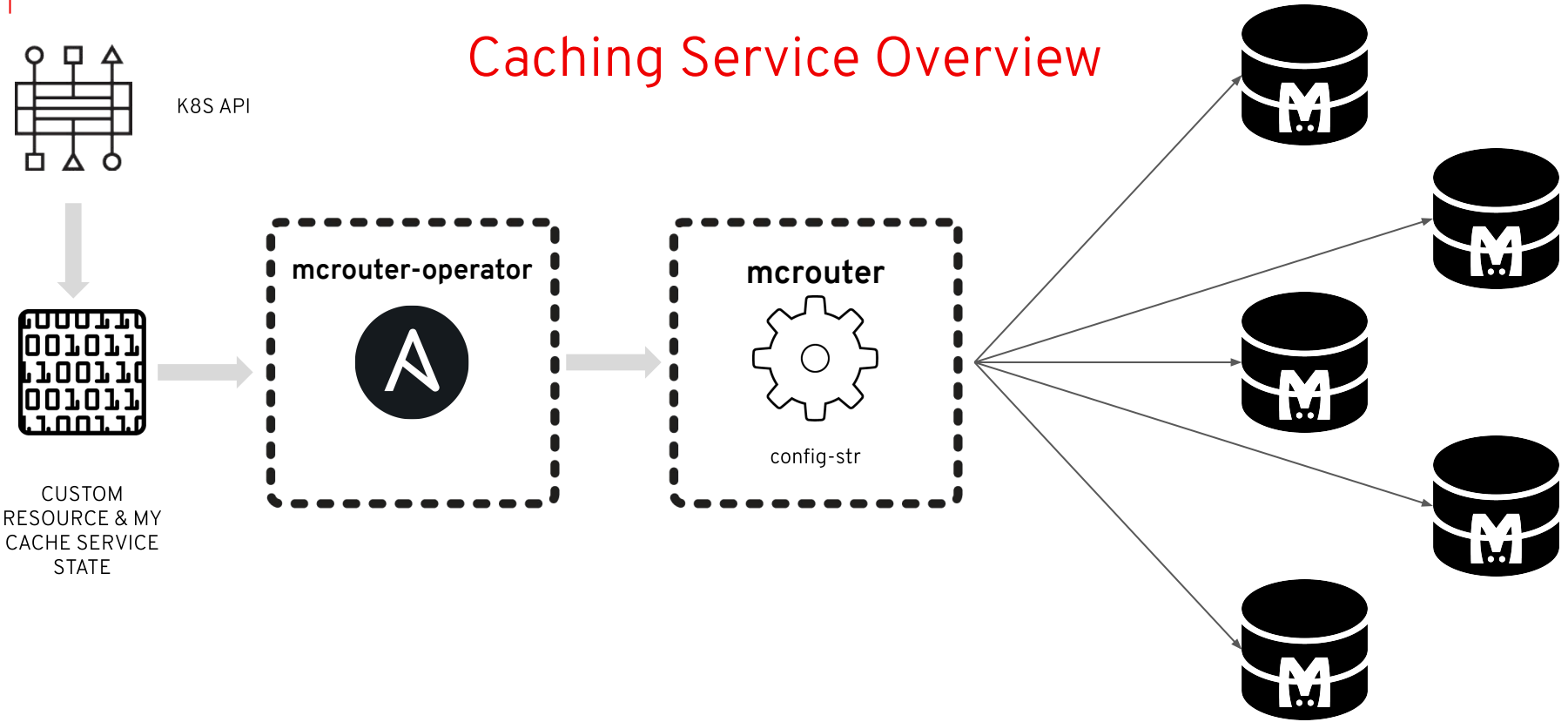
Demo: Initializing Your Operator and Walk Through

```
$ operator-sdk new null-operator
  --api-version=cache.example.com/v1alpha1
  --kind=Null
  --type=ansible
```

Demo: Deploying & Managing a Scalable Caching Service

- **Memcached** is a general-purpose distributed memory caching system, generic in nature, but intended for use in speeding up database calls, API calls, or page rendering.
- **Mcrouter** is a memcached protocol router for scaling memcached deployments such as connection pooling, flexible routing, replicated pools and a whole lot more.

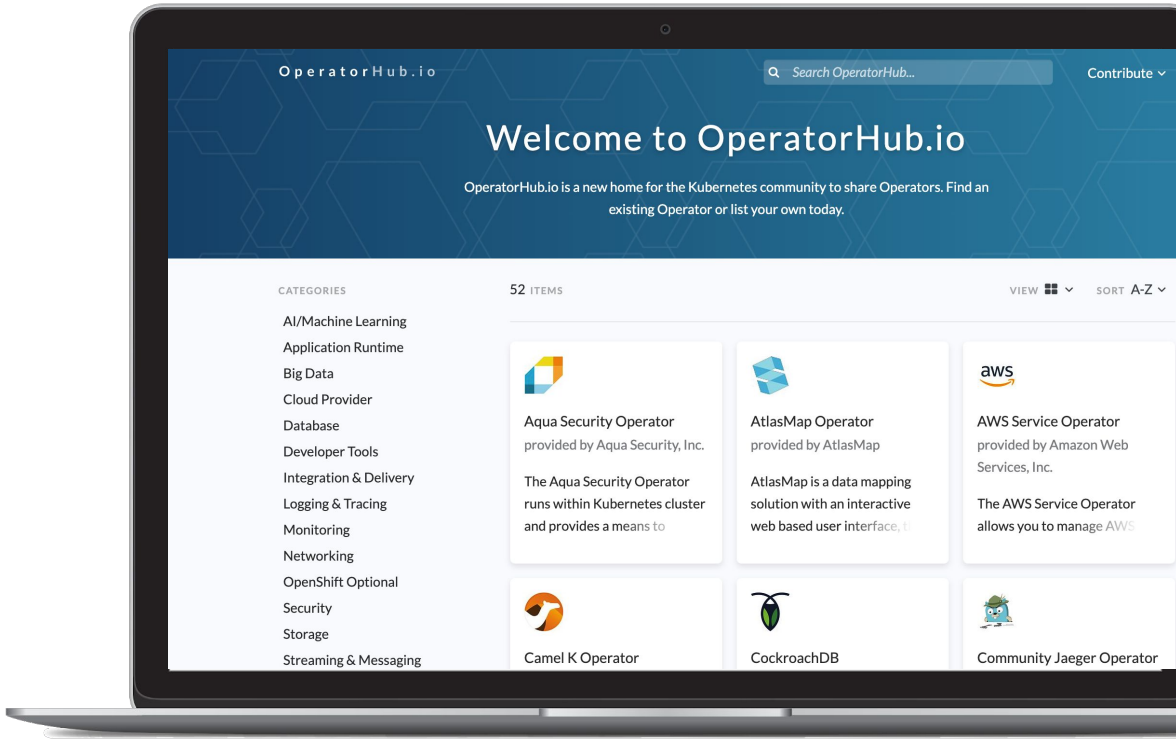
Caching Service Overview



Explore Operators

OperatorHub.io is a home for the Kubernetes community to share Operators.

Find an existing Operator or list your own today.



Next steps

Get started with Ansible:

ansible.com/get-started

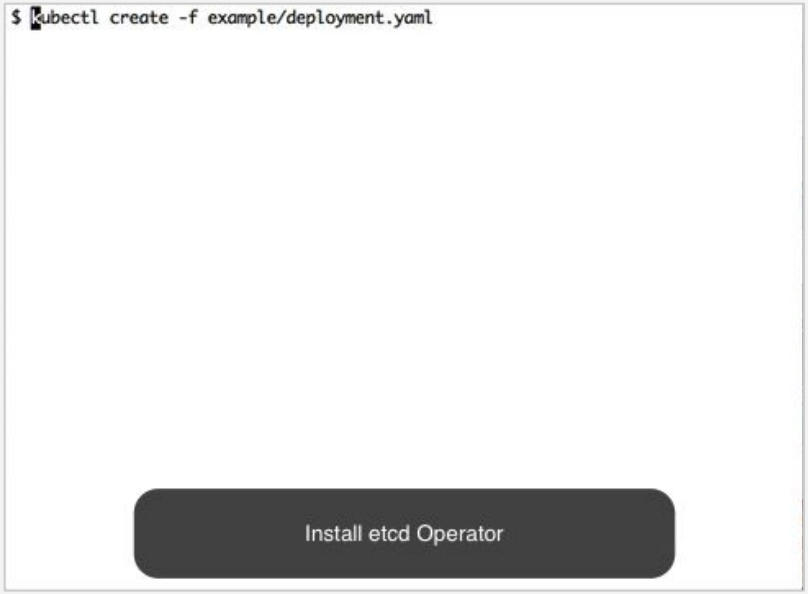
ansible.com/community

Get started with Operators:

github.com/operator-framework/getting-started

ansible.com/operators

More resources



etcd Operator
A great example of a sophisticated Kubernetes Operator using Ansible:
github.com/water-hole/etcd-ansible-operator

Memcached Operator
Simple walkthrough for building an Operators using the Ansible Operator SDK and Kubernetes CRDs
github.com/operator-framework/operator-sdk-samples/tree/master/memcached-operator

Red Hat

Thank You

linkedin.com/company/red-hat

youtube.com/user/RedHatVideos

facebook.com/redhatinc

twitter.com/RedHat

Red Hat