THE LINUX FOUNDATION
OPEN SOURCE SUMMIT

Automating stateful applications with Kubernetes Operators

Josh Wood and Ryan Jarvinen, Red Hat

@joshixisjosh9 :: @ryanj

THE LINUX FOUNDATION

Josh Wood

Red Hat
Developer Advocate, Kubernetes and OpenShift

Formerly: DocOps, CoreOS

- joshix@redhat.com
- @joshixisjosh9
- github.com/joshix
- speakerdeck.com/joshix

Ryan Jarvinen

Red Hat

Developer Advocate, Kubernetes and OpenShift

Formerly: CoreOS, LindenLab

- ryanj@redhat.com
- @ryanj
- github.com/ryanj
- bit.ly/k8s-workshops
  learn.openshift.com

Why you care about Operators

Every application on any platform must be installed, configured, managed, and upgraded over time. **Patching is critical to security.**

“Anything not automated is slowing you down.”

Operators are automated software managers for Kubernetes clusters: Install *and* Lifecycle.

THE LINUX FOUNDATION

Why you care about Operators

- You can use them right now to make deploying software and keeping it running easier: A killer new db, but you don’t want know all about its config file, the UI for backups, how to connect monitoring… how to shard/cluster/distribute it
- You can build them today with the Operator Framework SDK. It and our Operators are Open Source.

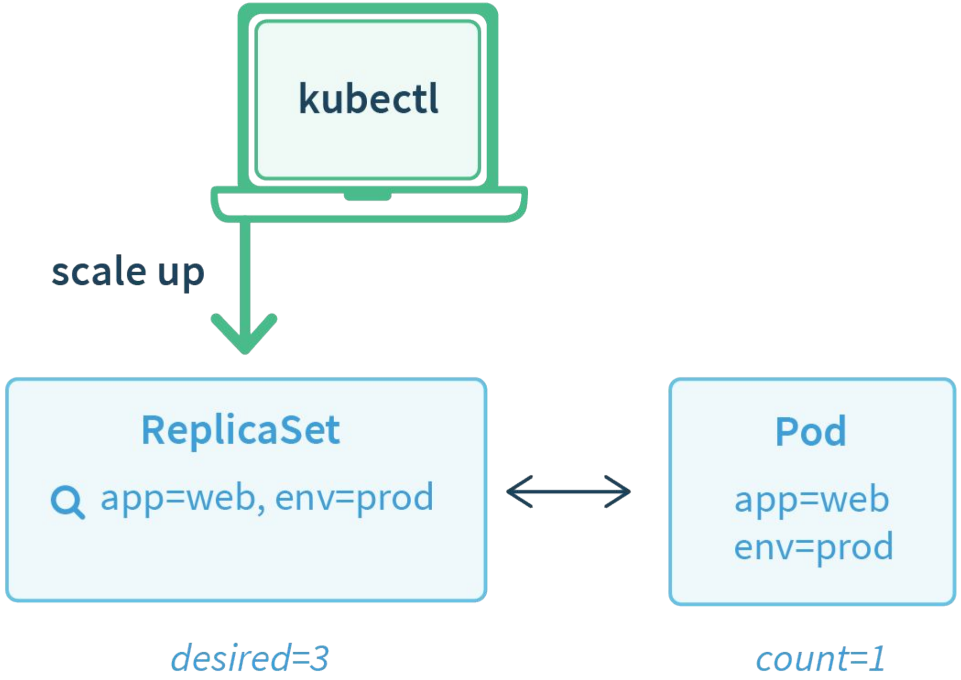
Overview

Scaling stateless apps: Easy

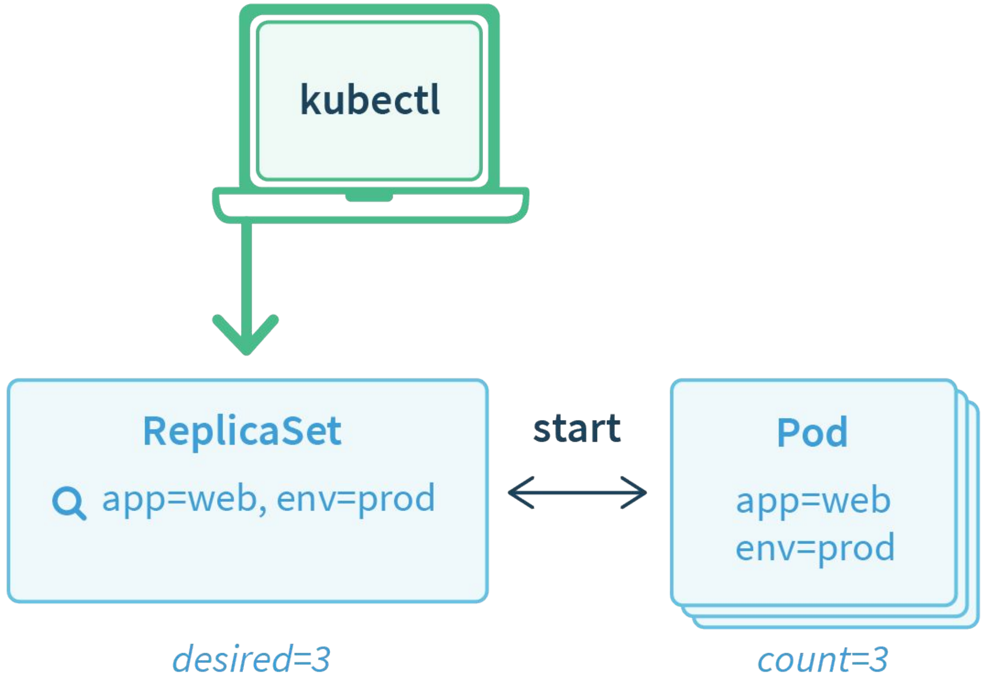
ReplicaSet

```
$ kubectl scale deploy/staticweb --replicas=3
```

ReplicaSet


kubectl
scale up
ReplicaSet
app=web, env=prod
Pod
app=web
env=prod
desired=3
count=1

ReplicaSet


kubectl
ReplicaSet
app=web, env=prod
start
Pod
app=web
env=prod
desired=3
count=3

Overview

What about apps that… store data?
Or have their own notion of a “cluster”?

Databases?

Creating a database is easy

```
$ kubectl run db --image=quay.io/my/db
```

Running it is harder

- **Resize/Upgrade** - coordination for availability
- **Reconfigure** - tedious generation / templating
- **Backup** - requires coordination among instances
- **Healing** - restore backups, rejoin db cluster

THE LINUX FOUNDATION

If only k8s knew…

Extend Kubernetes

The goal

```
$ cat database-cluster.yaml
spec:
 clusterSize: 3
 readReplicas: 2
 version: v4.0.1
[...]
```

What are Operators?

- Application-specific controllers that extend the Kubernetes API to create, configure, and manage instances of complex stateful applications on behalf of a Kubernetes user

- Extend the Kubernetes API through the Custom Resources (CRD) mechanism

Reconciling desired state for *your* application

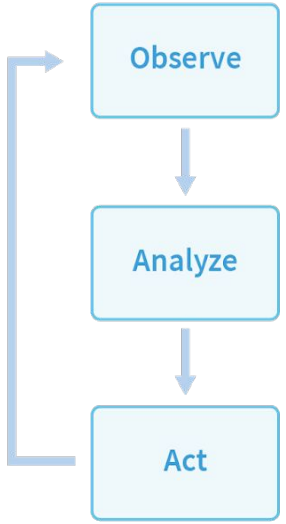
THE LINUX FOUNDATION

Simple example: etcd Operator

```
$ cat deployment.yaml
spec:
 clusterSize: 3
 version: v3.3.9
[...]
```

etcd Operator

kubernetes
OPERATOR
etcd



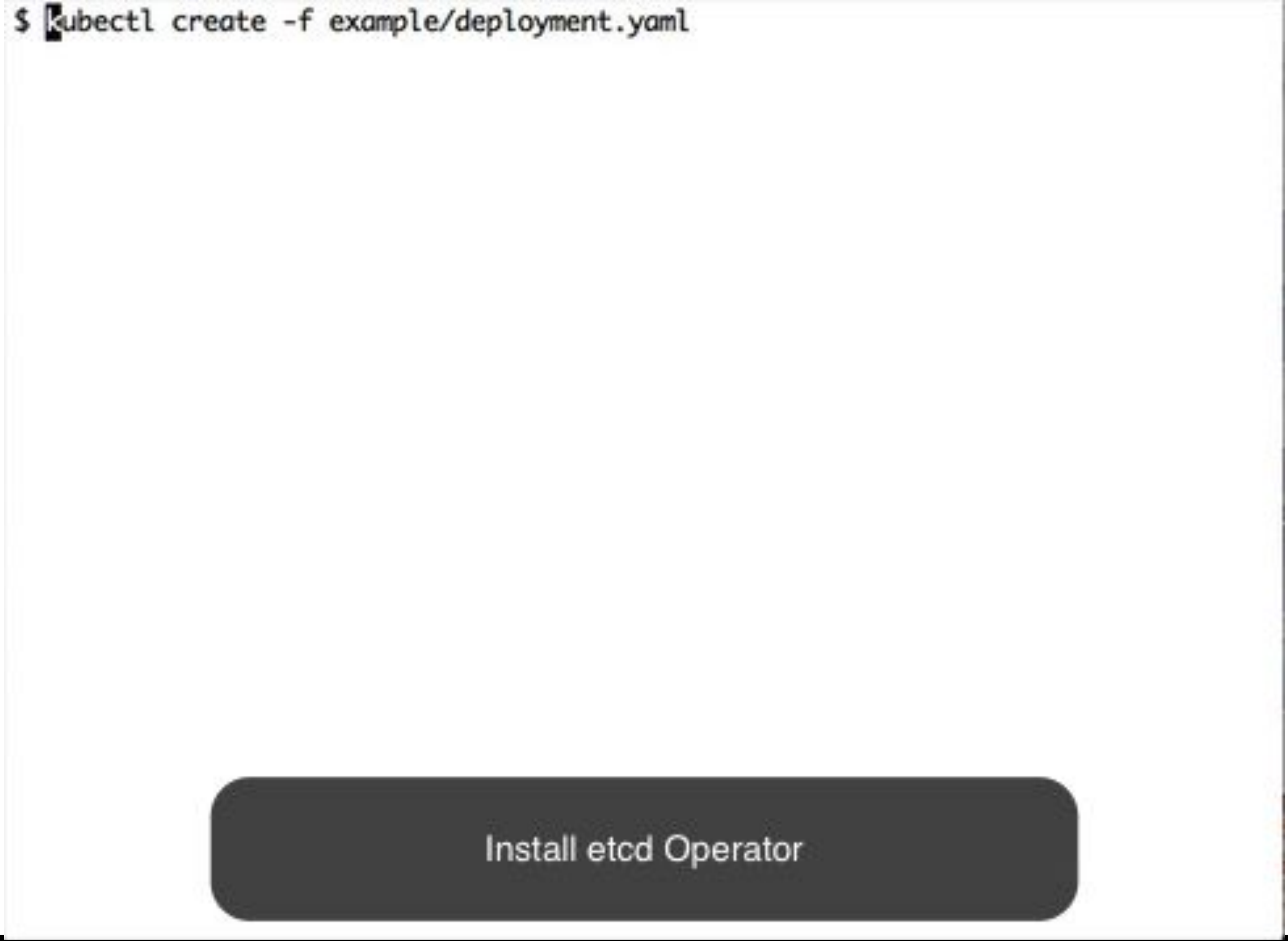
Cluster "A" has 2 running pods:
- name: A-000, version 3.3.8
- name: A-001, version 3.3.9

Differences from desired config:
- should be version
- should have 3 members

How to get to desired config:
- Recover 1 member
- Back up cluster
- Upgrade to 3.3.9

DEMO TIME

https://github.com/coreos/etcd-operator

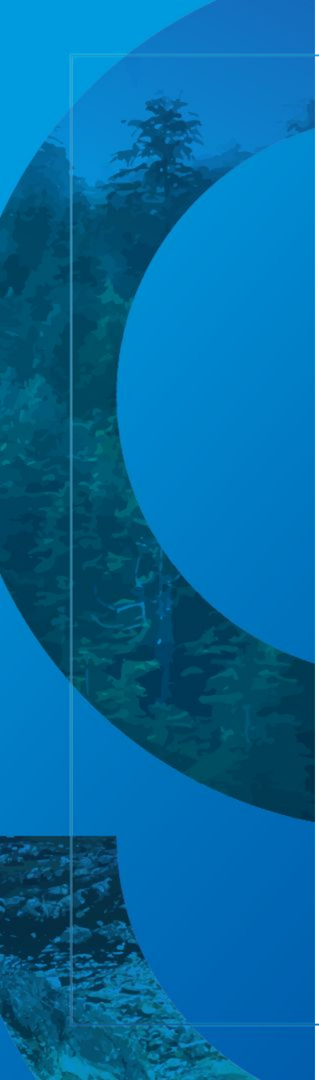


```
$ kubectl create -f example/deployment.yaml
```

Available today

https://github.com/operator-framework/awesome-operators

- Elastisearch
- etcd
- Prometheus
- MySQL
- Postgres (crunchy)
- “and many more!”



- Databases
- File, block, and object storage
- ...apps with their own notion of “cluster”
- Apps for *distribution* on Kubernetes

Build your Operator

https://github.com/operator-framework/operator-sdk

An Operator is a custom Kubernetes controller for your app. The SDK makes it easier to build Operators:

- High level APIs and abstractions to write operational logic
- Scaffolding and code generation to bootstrap new projects
- Extensions to cover common Operator use cases

Build an Operator to make your app *Kubernetes native*

Build your Operator

https://github.com/operator-framework/operator-sdk

1. Create a new operator project using the SDK Command Line Interface(CLI)
2. Define new resource APIs by adding Custom Resource Definitions(CRD)
3. Specify resources to watch using the SDK API
4. Define the operator reconciling logic in a designated **handler** and use the SDK API to interact with resources
5. Use the SDK CLI to build and generate the operator deployment manifests

Build your Operator

Operator SDK walkthrough:

Build your own memcached Operator:
https://github.com/operator-framework/getting-started/

Resources

- https://coreos.com/operators
- Operator Framework and SDK on Github
  https://github.com/operator-framework/
- Awesome Operators!
  https://github.com/operator-framework/awesome-operators
- Introducing the Operator Framework
  https://coreos.com/blog/introducing-operator-framework
- Make a Kubernetes Operator in 15 mins with Helm
  https://blog.openshift.com/make-a-kubernetes-operator-in-15-minutes-with-helm/
- Kubernetes Custom Resources Grow up in v1.10.0
  https://blog.openshift.com/kubernetes-custom-resources-grow-up-in-v1-10/

THE LINUX FOUNDATION

Coming Soon: Operators tutorials

OpenShift Learning Portal

http:// learn.openshift.com/

Thank You!

Thanks for attending

Automating with Operators

Josh Wood and Ryan Jarvinen

THE LINUX FOUNDATION

OPEN SOURCE SUMMIT

THE LINUX FOUNDATION