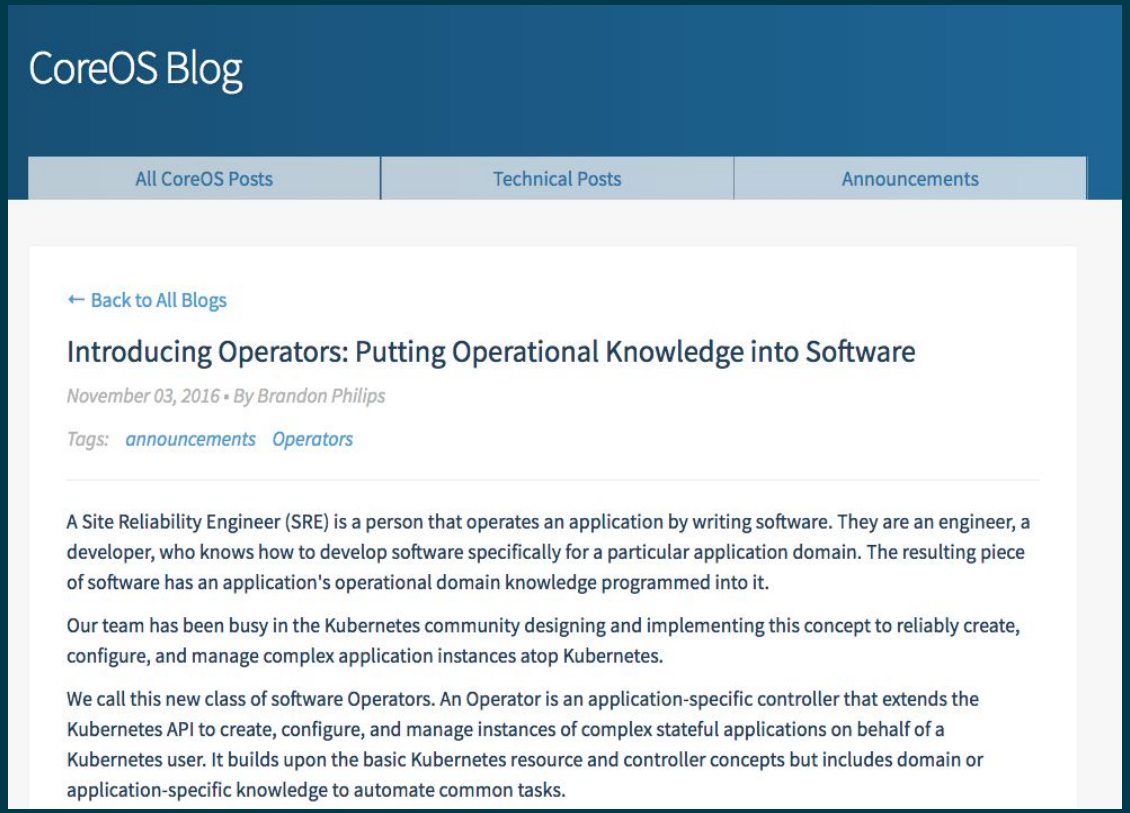
redhat

etcd Operator

redhat

Remember...



1 Resource

2 Controller

3 Knowledge

On that same day...November 3, 2016

CoreOS Blog

All CoreOS Posts | Technical Posts | Announcements

← Back to All Blogs

Introducing the etcd Operator: Simplify etcd cluster configuration and management

November 03, 2016 • By Hongchao Deng

Tags: announcements Operators

Today, CoreOS introduced a new class of software in the Kubernetes community called an *Operator*. An Operator builds upon the basic Kubernetes resource and controller concepts but includes application domain knowledge to take care of common tasks. They reduce the complexity of running distributed systems and help you focus on the desired configuration, not the details of manual deployment and lifecycle management.

etcd is a distributed key-value store. In fact, etcd is the primary datastore of Kubernetes; storing and replicating all Kubernetes cluster state. As a critical component of a Kubernetes cluster having a reliable automated approach to its configuration and management is imperative.

As a distributed consensus-based system, the cluster configuration of etcd can be complicated. Bootstrapping, maintaining quorum, reconfiguring cluster membership, creating backups, handling disaster recovery, and monitoring critical events are tedious work, and require etcd-specific expertise.

Today we are introducing the etcd Operator and the Prometheus Operator showing how to make applications like these easier to run on Kubernetes. In this post, we'll outline the importance of an Operator for etcd. Let's dive in.

Mission Critical Distributed Key Value Store

/etc directory
distributed over lots of hosts
= etcd

Key Value Store

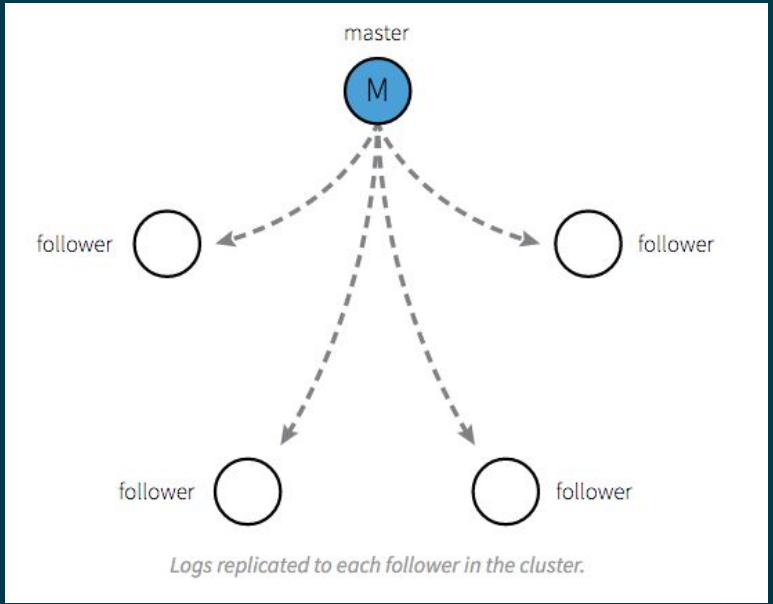
Key	Value
K1	AAAAAA
K2	BBBBBB
K3	ABABABABA
K4	CCCCDDDD
K5	CCCCCCCC

etcd

- Distributed key-value store that provides a reliable way to store data across a cluster of machines.
- Open-source and available on GitHub.
- Gracefully handles leader elections during network partitions and will tolerate machine failure, including the leader.

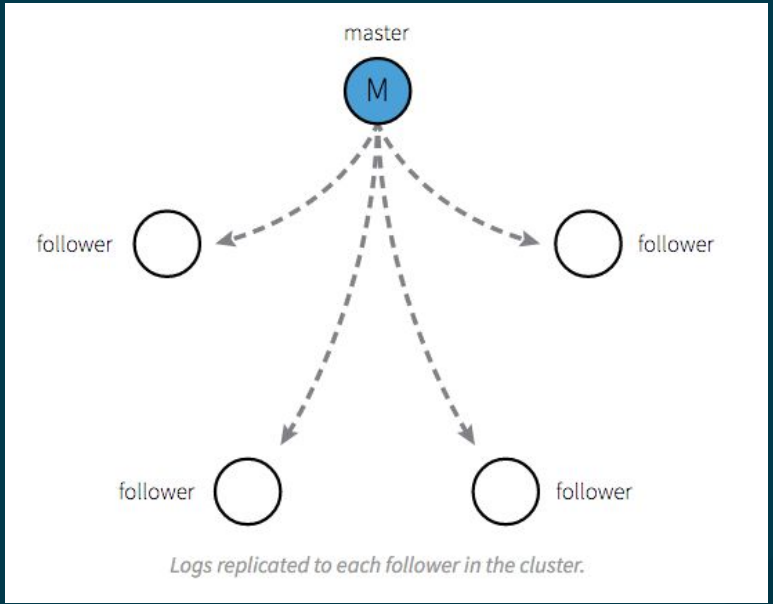
Concepts

- etcd communication between each machine is handled by the Raft consensus algorithm.
- One master node and follower node(s)
- When a master dies, then new leader will be elected. Data will be copied, and new leader will be serving.

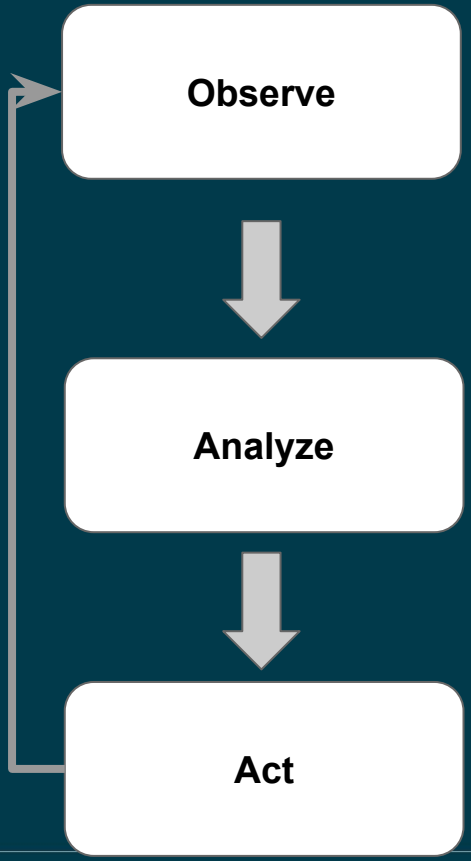


Concepts

- etcd communication between each machine is handled by the Raft consensus algorithm.
- One master node and follower node(s)
- When a master dies, then new leader will be elected. Data will be copied, and new leader will be serving.



Etcd Operator



Etcd Cluster “A” has two running pods:

- name: A-000, verison 3.0.9
- name: A-001, version 3.1.0

Differences from desired config:

- Should be version 3.1.0
- Should have 3 members

How to get desired config:

- Recover 1 member
- Backup cluster
- Upgrade to 3.1.0

- CRD - Custom Resource Definition
- Service Account
- Role - Allows access to API endpoints
- Role Binding
- Operator.yaml---> Deployment ---> Container image ---> Controller Binary ---> operator-sdk
- Example/Samples - Custom Resource
- README - provides details on how the Operator works and any other requirements

CR

```
Kind: EtcdCluster
Metdata:
  Name: mysample
Spec:
  Nodes: 3
  Version:
```