Build Your Own Kubernetes Operators

Mike Metral
Senior Architect

@mikemetral
Introduction - Mike Metral

Senior Architect

- Field Engineering @ CoreOS
  - Customer Enablement
  - Solution Design & Engineering

- Prev:
  - Product Architect @ Rackspace
    - OpenStack, Kubernetes
  - Software Eng. @ Sandia Nat’il Labs
    - Security & IaaS

@metral - GitHub, Slack
@mikemetral - Twitter
- Container Scheduling
- Container Management
- Orchestration Engine
- ...

kubernetes
kubernetes

- ...Ops Paradigm
- Extensive & versatile toolset
- Collection of best practices

Manage apps + compute in a highly-dynamic manner that is: modular, scalable, self-healing, and resilient.
Community

Top % 0.01 / 100

2500+

52k+

25k ✭
Kubernetes Controllers

- Deployments
- Endpoints
- DaemonSets
- Replication Controllers
- Node
- Ingress Controller
- Admission Controllers
How can we **extend** k8s?
Operators
New Resource
+ State
(optional)

Operational Knowledge

Controller
Operators

ThirdPartyResources (TPRs)

CustomResourceDefinitions (CRDs)

- Provides API object framework for a new resource
- Used to implement new resource types
  - e.g. Data model for Controllers

Controllers

- Active reconciliation process
  - Current State -> Desired State
- Used to automate app administration
  - \{add, extend, replace\} cluster functionality
TPRs -> CRDs

- Why The Move?
  - Remodel of API type + functionality
  - https://coreos.com/blog/custom-resource-kubernetes-v17
- TPRs & CRDs co-exist in k8s v1.7
  - TPR will be deprecated as of v1.8
- Migration Strategy for TPR data -> CRD:
  - https://github.com/kubernetes/kubernetes/kubernetes/pull/46677
TPRs -> CRDs

Major Changes for Operators:

- Client Packages moved from:
  - `k8s.io/client-go/pkg/api` -> `k8s.io/apimachinery/pkg/api`
  - e.g. API Types (in `/meta/v1`), errors, util etc.

- TPR code lived in:
  - `k8s.io/client-go/pkg/apis/extensions/v1beta1`

- CRD code lives in:
  - `k8s.io/apiextensions-apiserver/pkg/apis/apiextensions/v1beta1`
No “Right Way” to Create
{Operators, CRDs, Controllers}
Gophers Ahead
Kubernetes Client Libraries

- Working with CRD’s and Creating Controllers requires a k8s Client library
  - Golang
    - [github.com/kubernetes/client-go](https://github.com/kubernetes/client-go) (v3.0.0)
    - * Aim to use v4.0.0-beta.0
  - Community Client Libraries
type MyCustomResourceDefinition struct {
    // API obj kind & schema version
    metav1.TypeMeta

    // Standard object metadata (optional)
    Metadata api.ObjectMeta

    // Describe how the resource appears
    Spec v1beta1.CustomResourceDefinitionSpec

    // State of the CRD
    Status CustomResourceDefinitionStatus
}

for {
    current := getCurrentState()
    desired := getDesiredState()
    makeChanges(current, desired)
}
CRD Guidelines

- Creating a CRD is a privileged operation
- CRD Cluster Access
  - ClusterRole & ClusterRoleBindings required w/ RBAC
- CRD can be Cluster or Namespace scoped
  - * Cluster scoping is new
- Work in Progress
k8s.io/client-go Guidelines

Pods, RC, Nodes, PV
Secrets, ConfigMaps etc.

CustomResourceDefinitions

rest.RESTClient

rest.Interface

CoreV1Interface

rest.RESTClient

rest.Interface

kubernetes.Interface

implies a

has a

has many

implies a

CoreOS
k8s.io/client-go Guidelines

Pods, RCs, Nodes, PV Secrets, ConfigMaps etc.

- kubernetes.Interface
  - kubernetes.ClientSet
    - CoreV1Interface
      - rest.RESTClient
        - rest.Interface

CustomResourceDefinitions

- apiextensionsclientset.Interface
  - apiextensionsclientset.Clientset
    - rest.Interface
k8s.io/client-go Guidelines

- Docs == Code
- client.Get() / client.List() becomes expensive
- Local Cache and Event Handling
  - Informers
    - A local cache & controller for state event handling on a resource
  - Shared Informers
    - A single, optimized local cache store & controller for state event handling on multiple resources (e.g. kube-controller-manager)
Controller Guidelines

- Operate on one item at a time
- Never mutate original objects
- Level-driven design, not edge-driven
- Annotations are your friend
Custom App & Operator

● Memhog
  ○ App that wastefully allocates memory to buffers
  ○ github.com/metral/memhog

● Memhog Operator
  ○ Monitors an app’s memory utilization, vertically auto-scaling the app upon crossing a threshold
  ○ github.com/metral/memhog-operator
Memhog

Unreferenced data waits for the garbage collector
**Memhog**

**Pod**

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: memhog-01
labels:
  name: memhog
annotations:
  app-monitor.kubedemo.com/monitor: "true"
spec:
  containers:
    - name: memhog
      image: quay.io/metral/memhog:v0.0.1
      resources:
        limits:
          memory: 384Mi
        requests:
          memory: 256Mi
```

**AppMonitor (new CRD type)**

```yaml
apiVersion: kubedemo.com/v1
kind: AppMonitor
metadata:
  name: johnny-cache
  namespace: default
labels:
  app: johnny-cache
spec:
  memThresholdPercent: 75
  memMultiplier: 2
```
Memhug Operator

AppMonitor CRD

```yaml
apiVersion: kubedemo.com/v1
kind: AppMonitor
metadata:
  name: johnny-cache
  namespace: default
labels:
  app: johnny-cache
spec:
  memThresholdPercent: 75
  memMultiplier: 2
```

AppMonitorController

```python
for {
  ListWatch Pods
  SelectAnnotatedPods
  ListWatch AppMonitors
  currentBytes = queryPrometheus()
  threshold = AppMonitor * PodLimit
  if currentBytes > threshold
    redeployPod(pod, AppMonitor)
}
```
Demo
References

- TPR -> CRD
  - https://groups.google.com/forum/#!msg/kubernetes-dev/R749_-L_ssc/p-3tyl6mAQAJ

- CRD Example
  - https://github.com/kubernetes/apiextensions-apiserver/tree/master/examples/client-go

- The TPR is dead. Long live the CRD: Custom Resources in Kubernetes v1.7
  - https://coreos.com/blog/custom-resource-kubernetes-v17
References

- **Working with Controllers**
  - https://github.com/kubernetes/community/blob/master/contributors/devel/controllers.md

- **Writing a Custom Controller - Aaron Levy (CoreOS)**
  - https://github.com/aaronlevy/kube-controller-demo
  - youtu.be/_BuqPMLXfpE?list=PLj6h78yzYM2PAavlbv0iZkod4IVh_iGqV

- **CoreOS**
  - etcd - github.com/coreos/etcd-operator
  - Prometheus - github.com/coreos/prometheus-operator

- **Kubernetes**
  - DaemonSet Controller
    - github.com/kubernetes/kubernetes/blob/master/pkg/controller/daemon/daemoncontroller.go
References

- **TPRs:**
  - kubernetes.io/docs/user-guide/thirdpartyresources

- **TPR + CRD Discussions:**
  - github.com/kubernetes/features/issues/95
  - github.com/kubernetes/community/pull/524

- **TPR Deprecated**
  - https://github.com/kubernetes/client-go/tree/df46f7f13b3da19b90b8b4f0d18b8adc6fbf28dc/examples/third-party-resources-deprecated
Next Steps

- Let’s discuss your needs and workloads
  - sales@coreos.com
  - metral@coreos.com
CoreOS is running the world’s containers

We’re hiring:
careers@coreos.com

<table>
<thead>
<tr>
<th>OPEN SOURCE</th>
<th>ENTERPRISE</th>
</tr>
</thead>
<tbody>
<tr>
<td>90+ Projects on GitHub, 1,000+ Contributors</td>
<td>Support plans, training and more</td>
</tr>
<tr>
<td>container linux, rkt, etcd</td>
<td>TECTONIC</td>
</tr>
<tr>
<td>coreos.com</td>
<td><a href="mailto:sales@coreos.com">sales@coreos.com</a></td>
</tr>
</tbody>
</table>
Thank You!

bit.ly/ossna-k8s-operator

@mikemetral