



Building a Kubernetes Operator

With the Operator SDK

Michael Gasch

Platform Architect, Customer
Success

Agenda

Event-Driven Architecture

Terminology

4 Steps to write an Operator

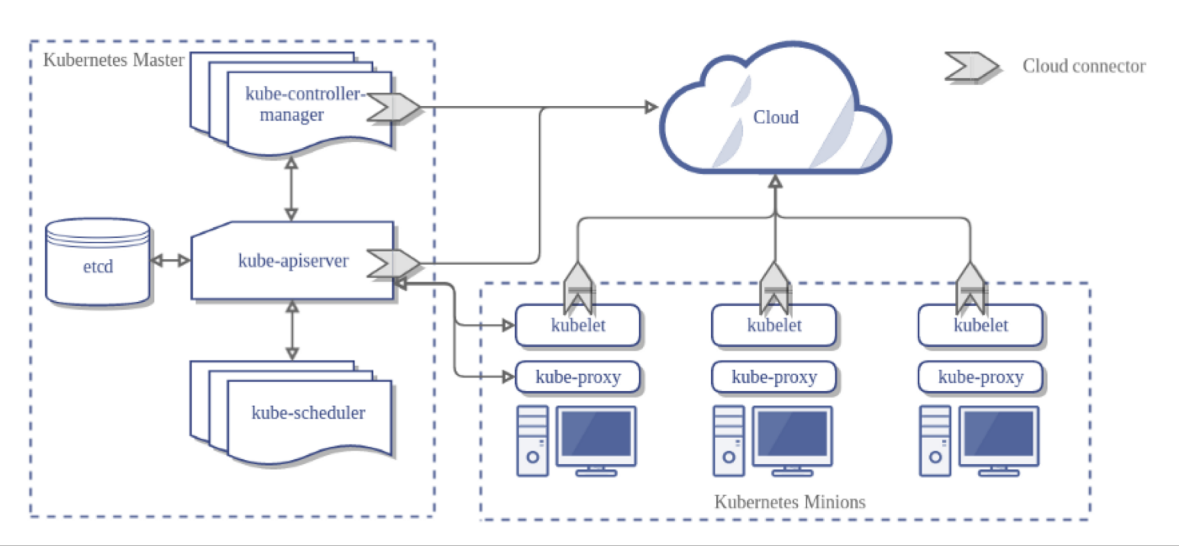
Tools and SDKs

// code() 😊

Going Deeper (Resources)

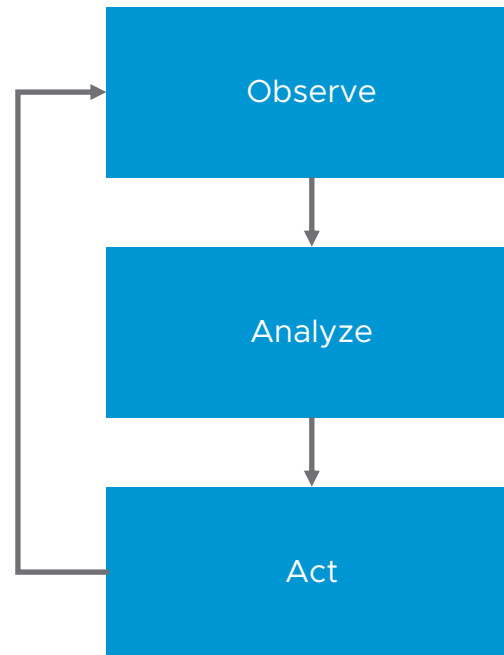
Event-Driven Architecture

Intrinsic to Kubernetes



Branch: master ▾ kubernetes / pkg / controller /	
msau42 Tolerate apiserver being older than controller-manager	
..	
apis/config	auto-generated file
bootstrap	Move from glog to klog
certificates	Merge pull request #71005 from
cloud	Add tests to ensure feature gate
clusterroleaggregation	Move from glog to klog
cronjob	update bazel
daemon	Add tests to ensure feature gate
deployment	Move from glog to klog
disruption	Move from glog to klog
endpoint	Move from glog to klog

Event-Driven Architecture



Terminology

Custom Controller

- Work with any Kubernetes Resource
- Especially effective when combined with Custom Resources

Custom Resource/ Custom Resource Definition

- Extension of the Kubernetes API that is not necessarily available on every Kubernetes Cluster
- E.g. “kind: Database”
- If CRD is an abstract Class, CR is a concrete Instance of it

Operator

- Uses Controller Pattern
- Adds human operational Knowledge in Software
- Single App Focus
- Discussion: <https://github.com/kubeflow/tf-operator/issues/300#issuecomment-357319596>

4 Steps to write an Operator

1. Define Custom Resource Definition
2. Generate Client Code
3. Listen for Events
4. Handle Events in Queue (your Business Logic 🧐 goes here)

Tools and SDKs

Operator Framework (what we'll use today)

- <https://github.com/operator-framework>

Kubebuilder

- <https://github.com/kubernetes-sigs/kubebuilder>

Helm App Operator Kit (leverages Operator SDK under the covers)

- <https://github.com/operator-framework/helm-app-operator-kit#instructions>

Metacontroller

- <https://github.com/GoogleCloudPlatform/metacontroller>

// code() 😊

Warning: do not use in Production ;)

```
apiVersion: sayer.example.com/v1alpha1
kind: Whalesay
metadata:
  name: moin-whalesay
spec:
  # Add fields here
  text: "Moin Moin!"
```

```
$ kubectl create -f deploy/crds/examples/say_moin.yaml
whalesay.sayer.example.com/moin-whalesay created
$ kubectl logs -f moin-whalesay-pod
```

< Moin Moin! >

The diagram illustrates a 2-loop calculation of the beta function. It includes a top-left dashed line, a central loop with a horizontal orange dashed line, and a bottom loop with a horizontal line labeled '0'. Various external lines (solid, dashed, wavy) and vertices (dots, crosses) are present, along with numerical coefficients like $\frac{1}{2}$, $\frac{1}{4}$, and $\frac{1}{8}$.

Resources

Resources

Awesome Operators

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

Writing Kube Controllers for Everyone (KubeCon 2018)

- <https://www.youtube.com/watch?v=AUNPLQVxvmw>

Extending Kubernetes 101

- <https://speakerdeck.com/mhausenblas/extending-kubernetes-101>

Extending Kubernetes with CRDs

- <https://www.youtube.com/watch?v=Ne4jQF-CPIM>

Building a Fault-Tolerant Resources Controller on Kubernetes

- <https://www.youtube.com/watch?v=v2gAcKKDf9Y>

Resources

Controllers: Lambda Functions for Extending your Infrastructure

- <https://www.youtube.com/watch?v=TM-2GgQ6Q2A>

Comparison of various Operator Tools and SDKs

- <https://admiralty.io/blog/kubernetes-custom-resource-controller-and-operator-development-tools/>

Deep Dive on the Operator SDK

- <https://itnext.io/analyzing-value-of-operator-framework-for-kubernetes-community-5a65abc259ec>



Thank You.