Developing Operators for Kubernetes

by Pawel Kopiczko & Ross Fairbanks

Agenda

1. Operators Overview
2. Custom Resource Definition
3. Operator Task
4. Exercise 1. Implement using REST API
5. Exercise 2. Implement using client-go

Introductions

Tools

- Minikube 0.23

```
$ minikube start --kubernetes-version 'v1.8.0'
```

- Go 1.9
- Git
- Git clone giantswarm/operator-workshop

```
$ git clone https://github.com/giantswarm/operator-workshop
    $GOPATH/src/github.com/giantswarm/operator-workshop
```

Operators Overview

What is an Operator

- An application specific controller that extends the Kubernetes API
- Manages a CRD (Custom Resource Definition) which replace TPRs (Third Party Resources)
- CRD stores state and the operator performs operational tasks
- Pattern first proposed by CoreOS in November 2016

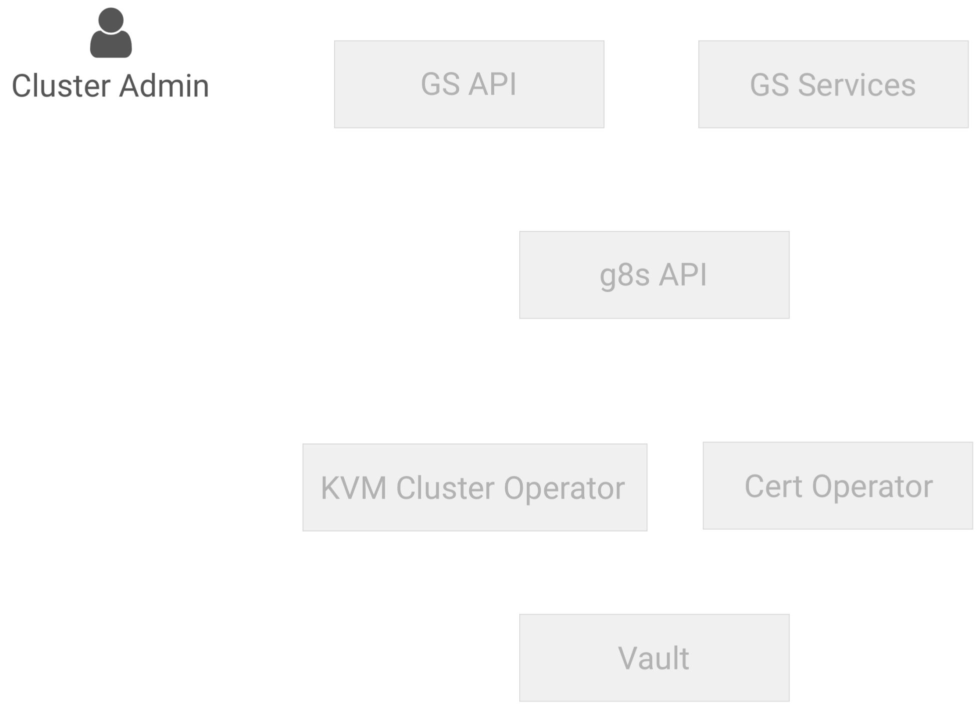
Example Operators

- Etcd
- Prometheus
- Rook

Giant Swarm Operators

- aws-operator
- kvm-operator
- cert-operator

Creating a guest cluster on g8s



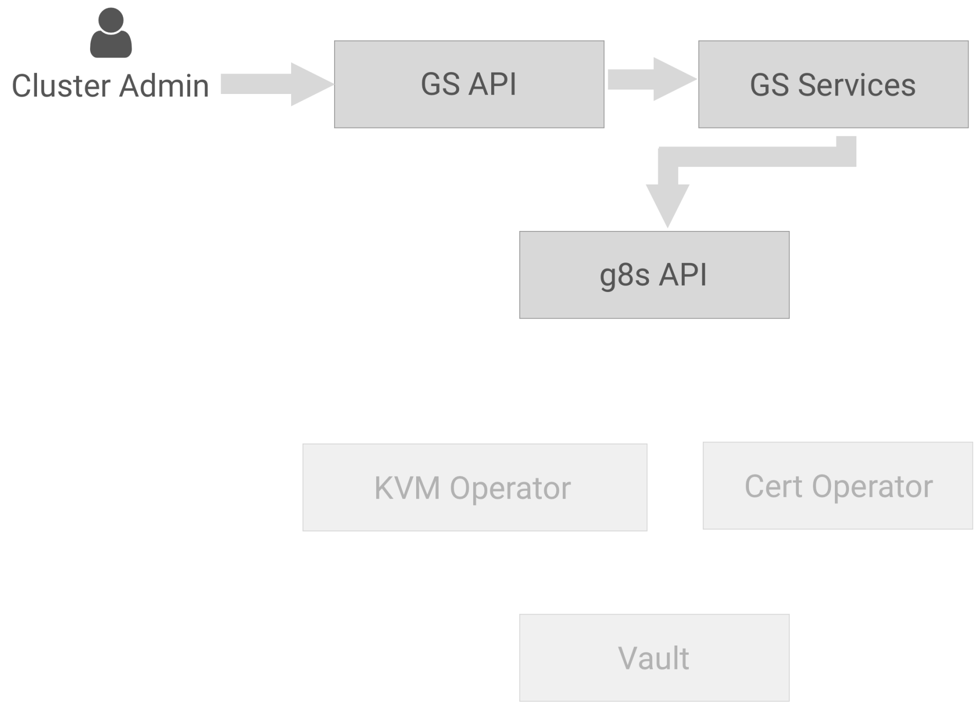
Kubernetes State

```
$ kubectl get pods
No resources found.

$ kubectl get namespaces
No resources found.

$ kubectl get secrets
No resources found.
```

Custom Resource Definitions

No objects yet.

Cluster Admin performs POST to GS API



Kubernetes State

```
$ kubectl get pods
No resources found.
```

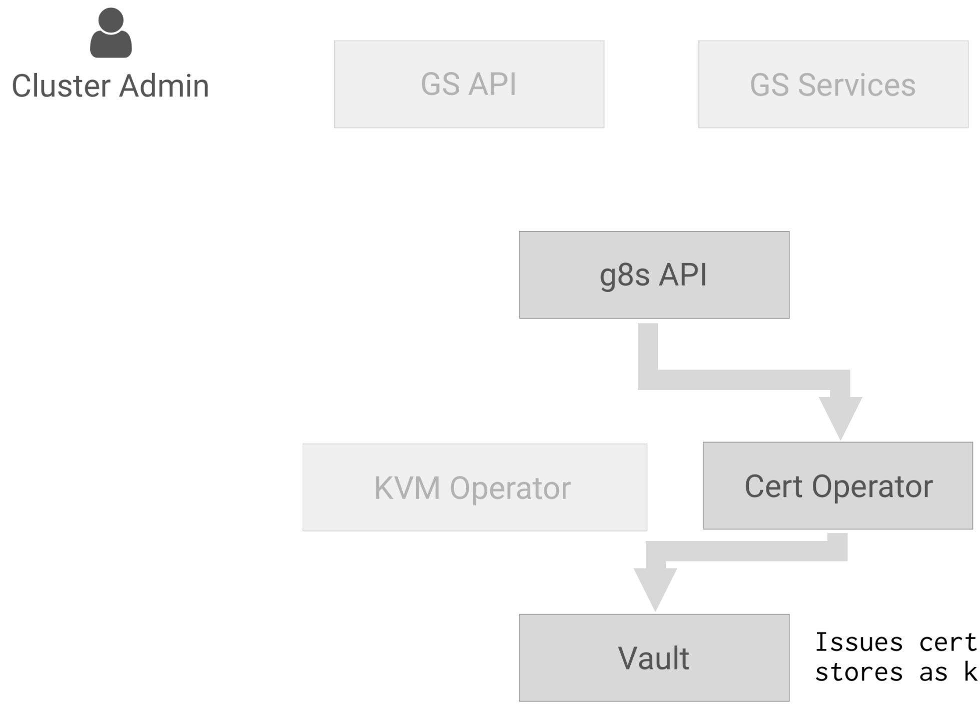
```
$ kubectl get namespaces

NAME     STATUS    AGE
uo91f    active    1m
```

```
$ kubectl get secrets
No resources found.
```

Custom Resource Definitions

- Cluster object uo91f

Cert Operator acts on Cert CRDs



Cluster Admin

GS API

GS Services

g8s API

KVM Operator

Cert Operator

Vault

Issues certificates,
stores as k8s secrets

Kubernetes State

```
$ kubectl get pods
No resources found.
```

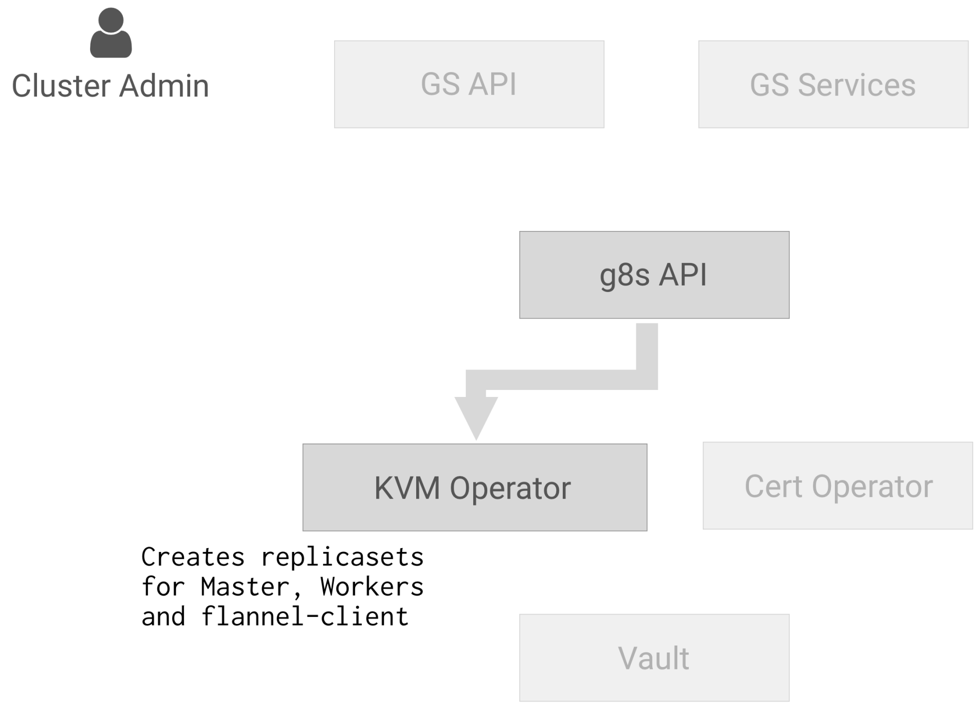
```
$ kubectl get namespaces
NAME    STATUS   AGE
uo91f   active   1m
```

```
$ kubectl get secrets
NAME               ...
uo91f-root-ca
uo91f-cert-1
...
```

Custom Resource
Definitions

- Cluster object uo91f
- Multiple cert objects for components

Cluster Operator acts on Cluster CRDs



Cluster Admin

GS API

GS Services

g8s API

KVM Operator

Cert Operator

Creates replicasets
for Master, Workers
and flannel-client

Vault

Kubernetes State

```
$ kubectl get pods
NAME
flannel-client-2256205...
flannel-client-2256205...
flannel-client-2256205...
master-3537008445-vnvxh
worker-2482331707-bx670
worker-2482331707-qq176

$ kubectl get namespaces
NAME     STATUS    AGE
uo91f    active    1m

$ kubectl get secrets
NAME                ...
uo91f-cert-1
...
```

Custom Resource Definition

Third Party Resources (Legacy)

- First mechanism for extending Kubernetes
- Added in Kubernetes 1.2
- Deprecated in Kubernetes 1.7
- Removed in Kubernetes 1.8

Custom Resource Definition

- Added in Kubernetes 1.7
- Reimplemented based on lessons learned from TPRs
- Namespace or cluster scoped
- Better pluralization support
- Shortname support
- Finalizers

Example CRD

```yaml
apiVersion: apiextensions.k8s.io/v1beta1
kind: CustomResourceDefinition
metadata:
  name: crontabs.stable.example.com
spec:
  group: stable.example.com
  version: v1
  scope: Namespaced
  names:
    plural: crontabs
    singular: crontab
    kind: CronTab
    shortNames:
    - ct
```

Example Custom Resource

```
apiVersion: "stable.example.com/v1"
kind: CronTab
metadata:
  name: my-cron-object
spec:
  cronSpec: "* * * * */5"
  image: my-cron-image
```

API endpoints

- Core group endpoint (also known as legacy group)

```
/api/v1/pods
```

- Named group endpoint

```
/apis/apps/v1beta1/deployments
```

- CRD endpoint

```
/apis/stable.example.com/v1/crontabs
```

Demo

Finalizers

- Available in Kubernetes 1.8
- Allow controllers to implement asynchronous pre-delete hooks

```yaml
apiVersion: "stable.example.com/v1"
kind: CronTab
metadata:
  name: my-cron-object
  finalizers:
  - finalizer.stable.example.com
```

Validation

- Alpha in Kubernetes 1.8
- Allows validation of custom objects using OpenAPI V3 schema

```
spec:
  validation:
   # openAPIV3Schema is the schema for validating custom objects.
    openAPIV3Schema:
      properties:
        spec:
          properties:
            cronSpec:
            replicas:
              type: integer
              minimum: 1
              maximum: 10
```