RocketMQ Operator

24 Nov, 2018
谐云科技 魏欢
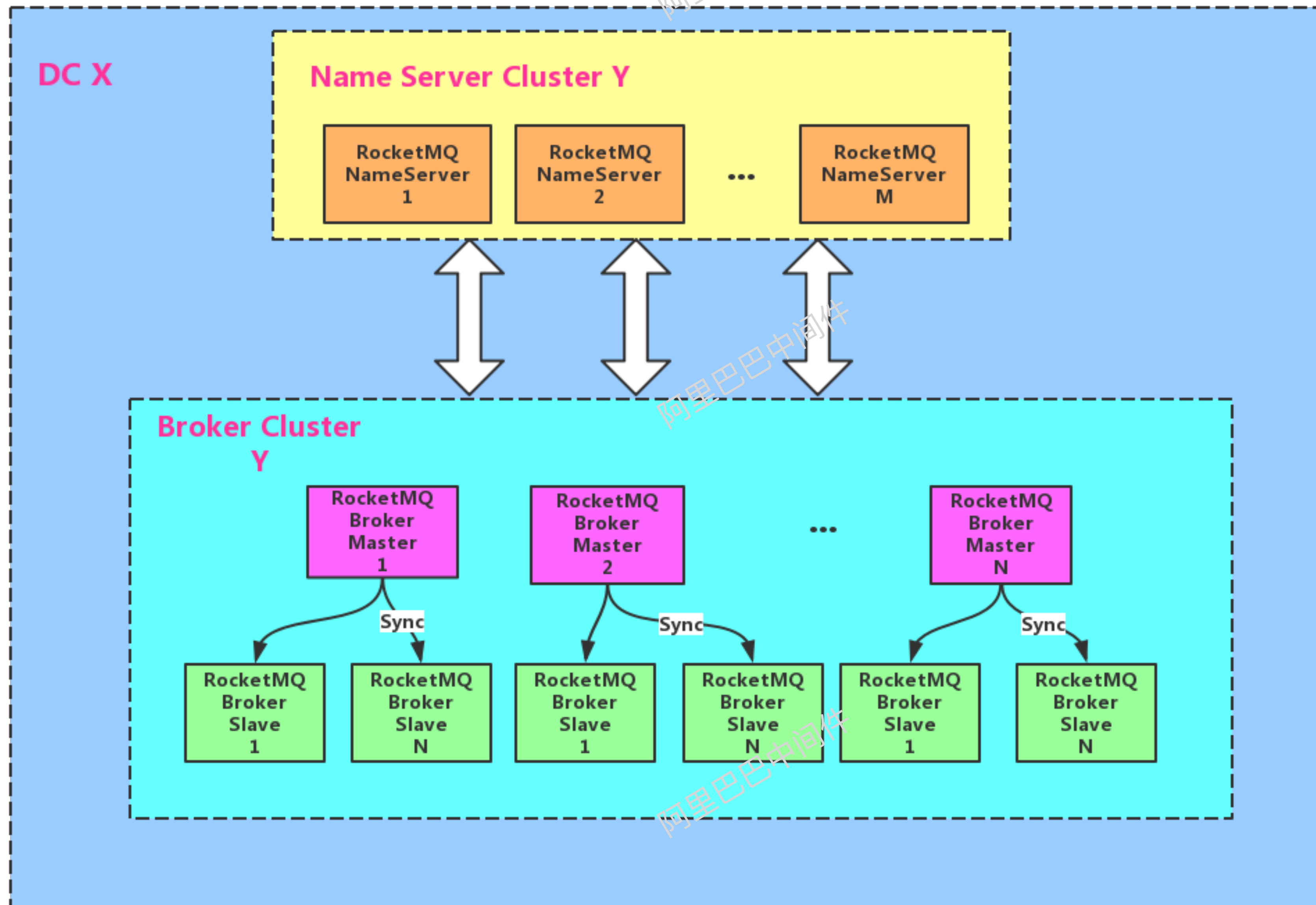
Agenda

- What is RocketMQ Operator

- How to design an Operator

- How to write RocketMQ Operator

- RocketMQ Operator Tutorial

- RocketMQ Operator Improvements Plan

Sample: RocketMQ Clusters In DC X

Question: How do we operate RocketMQ clusters?

- How to set up a new RocketMQ cluster in seconds?

- How to scale up/ scale down RocketMQ NameServer endpoints, or Broker endpoints?

- How about RocketMQ endpoints fail over?

- How to upgrade a RocketMQ cluster?

- How to operate all RocketMQ clusters in a data center?

- More concerns …

What is RocketMQ Operator ?

- It's slogan - Create, operate and scale self-healing RocketMQ clusters on Kubernetes.

- It's a Go program designed for running in Kubernetes cluster to automatically operate RocketMQ clusters.

- It's open sourced. Currently maintained by @huanwei from **HarmonyCloud.cn**

- Github - https://github.com/huanwei/rocketmq-operator

RocketMQ Operator Topology
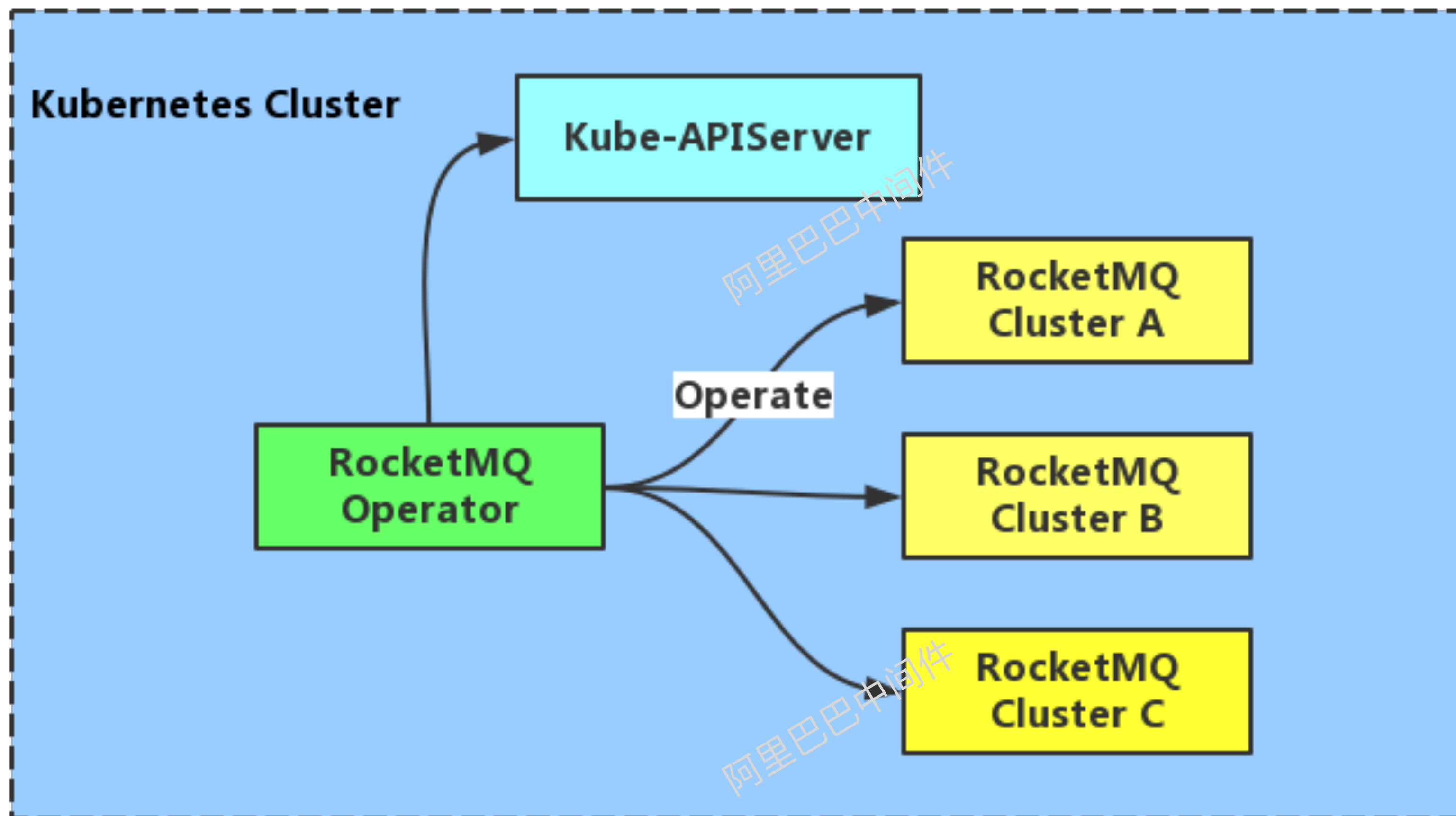
Declarative Programming Pattern

```
//just pseudo codes for a Controller
for {
  desired := getDesiredState()
  current := getCurrentState()
  makeChanges(desired, current)
}
```

Sample: Controller Pattern In Kubernetes

https://github.com/kubernetes/kubernetes/blob/release-1.13/cmd/kube-controller-manager/app/
controllermanager.go#L357

```
354
355       // NewControllerInitializers is a public map of named controller groups (you can start more than one in an init func)
356       // paired to their InitFunc.  This allows for structured downstream composition and subdivision.
357       func NewControllerInitializers(loopMode ControllerLoopMode) map[string]InitFunc {
358           controllers := map[string]InitFunc{}
359           controllers["endpoint"] = startEndpointController
360           controllers["replicationcontroller"] = startReplicationController
361           controllers["podgc"] = startPodGCController
362           controllers["resourcequota"] = startResourceQuotaController
363           controllers["namespace"] = startNamespaceController
364           controllers["serviceaccount"] = startServiceAccountController
365           controllers["garbagecollector"] = startGarbageCollectorController
366           controllers["daemonset"] = startDaemonSetController
367           controllers["job"] = startJobController
368           controllers["deployment"] = startDeploymentController
369           controllers["replicaset"] = startReplicaSetController
370           controllers["horizontalpodautoscaling"] = startHPAController
371           controllers["disruption"] = startDisruptionController
372           controllers["statefulset"] = startStatefulSetController
373           controllers["cronjob"] = startCronJobController
374           controllers["csrsigning"] = startCSRSigningController
375           controllers["csrapproving"] = startCSRApprovingController
376           controllers["csrcleaner"] = startCSRCleanerController
377           controllers["ttl"] = startTTLController
378           controllers["bootstrapsigner"] = startBootstrapSignerController
379           controllers["tokencleaner"] = startTokenCleanerController
380           controllers["nodeipam"] = startNodeIpamController
381           if loopMode == IncludeCloudLoops {
382               controllers["service"] = startServiceController
383               controllers["route"] = startRouteController
384               // TODO: volume controller into the IncludeCloudLoops only set.
385               // TODO: Separate cluster in cloud check from node lifecycle controller.
386           }
```

RocketMQ Operator Controllers

```go
156        m := BrokerController{
157                opConfig: opConfig,
158
159                opClient:   opClient,
160                kubeClient: kubeClient,
161
162                clusterLister:       clusterInformer.Lister(),
163                clusterListerSynced: clusterInformer.Informer().HasSynced,
164                clusterUpdater:      newClusterUpdater(opClient, clusterInformer.Lister()),
165
166                serviceLister:       serviceInformer.Lister(),
167                serviceListerSynced: serviceInformer.Informer().HasSynced,
168                serviceControl:      NewRealServiceControl(kubeClient, serviceInformer.Lister()),
169
170                statefulSetLister:       statefulSetInformer.Lister(),
171                statefulSetListerSynced: statefulSetInformer.Informer().HasSynced,
172                statefulSetControl:      NewRealStatefulSetControl(kubeClient, statefulSetInformer.Lister()),
173
174                podLister:       podInformer.Lister(),
175                podListerSynced: podInformer.Informer().HasSynced,
176                podControl:      NewRealPodControl(kubeClient, podInformer.Lister()),
177
178                queue:    workqueue.NewNamedRateLimitingQueue(workqueue.DefaultControllerRateLimiter(), "brokercluster"),
179                recorder: recorder,
180        }
```

Extending Kubernetes API With Custom Resource Definitions: RocketMQ Operator Types

https://github.com/huanwei/rocketmq-operator/blob/release-0.1/pkg/apis/rocketmq/v1alpha1/types.go#L22

```go
22  type BrokerClusterSpec struct {
23      BrokerImage         string                          `json:"brokerImage"`
24      NameServers         string                          `json:"nameServers"`
25      AllMaster           bool                            `json:"allMaster`
26      ReplicationMode     string                          `json:"replicationMode`
27      GroupReplica        int32                           `json:"groupReplica, omitempty"`
28      MembersPerGroup     int32                           `json:"membersPerGroup, omitempty"`
29      Properties          map[string]string               `json:"properties, omitempty"`
30      NodeSelector        map[string]string               `json:"nodeSelector, omitempty"`
31      Affinity            *corev1.Affinity                `json:"affinity, omitempty"`
32      VolumeClaimTemplate *corev1.PersistentVolumeClaim   `json:"volumeClaimTemplate, omitempty"`
33      Config              *corev1.LocalObjectReference    `json:"config,omitempty"`
34  }
35
```

RocketMQ Operator Tutorial

https://github.com/huanwei/rocketmq-operator/blob/master/docs/tutorial.md

Prerequisites

- Kubernetes 1.9+, since StatefulSets are stable (GA) in 1.9.

Installation

Create a namespace

```
kubectl create ns rocketmq-operator
```

RocketMQ Operator Tutorial

Deploy the cluster

0) Create a name server cluster

Create a deployment for namesrv, and expose namesrv service with NodePort.

```
kubectl create -f deploy/00-namesrv.yaml
```

1) Create the CRD with kind BrokerCluster.

```
kubectl create -f deploy/01-resources.yaml
```

2) Create RBAC resources for Rocketmq Operator

```
kubectl create -f deploy/02-rbac.yaml
```

RocketMQ Operator Tutorial

3) Deploy Rocketmq Operator

```
kubectl create -f deploy/03-deploymentWithConfig.yaml
```
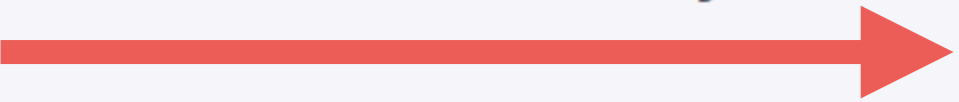
4) Create a broker cluster

Create a cluster with 2 masters:

```
kubectl create -f deploy/04-cluster-2m.yaml
```

OR:

Create a cluster with 2 masters and 2 slaves:

```
kubectl create -f deploy/04-cluster-2m-2s.yaml
```

```
huanteMacBook-Pro:deploy huan$ cat 04-cluster-2m-2s.yaml
-
apiVersion: "rocketmq.huanwei.io/v1alpha1"
kind: BrokerCluster
metadata:
  name: mybrokercluster
  namespace : rocketmq-operator
spec:
  brokerImage: huanwei/rocketmq-broker:4.3.0-operator
  nameServers: 10.10.103.181:9876;10.10.101.98:9876
  allMaster: false
  replicationMode: SYNC
  groupReplica: 2
  membersPerGroup: 2
```

RocketMQ Operator Tutorial

Verify the cluster

Below is the command line output for a cluster with 2 masters and 2 slaves.

```
[root@k8s-master deploy]# kubectl get pods -n rocketmq-operator -owide
NAME                                READY  STATUS    RESTARTS  AGE   IP                NODE
mybrokercluster-0-0                 1/1    Running   0         38s   192.168.122.145   k8s-node4
mybrokercluster-0-1                 1/1    Running   0         33s   192.168.196.17    k8s-node5
mybrokercluster-1-0                 1/1    Running   0         38s   192.168.196.16    k8s-node5
mybrokercluster-1-1                 1/1    Running   0         34s   192.168.122.146   k8s-node4
rocketmq-namesrv-7d7df8c7d7-4t2g2   1/1    Running   2         10d   10.10.103.181     k8s-node4
rocketmq-namesrv-7d7df8c7d7-6x47m   1/1    Running   2         7d    10.10.101.98      k8s-node5
rocketmq-operator-d9dc87954-nw262   1/1    Running   0         6m    192.168.196.14    k8s-node5
```

RocketMQ Operator Tutorial

Verify the cluster

Below is the command line output for a cluster with 2 masters and 2 slaves.

```
[root@k8s-master deploy]# kubectl exec -it mybrokercluster-0-0 -n rocketmq-oper
[root@mybrokercluster-0-0 bin]# cat ../conf/broker.conf
#This file was auto generated by rocketmq-operator. DO NOT EDIT.
brokerClusterName=DefaultCluster
brokerName=broker-0
brokerId=0
deleteWhen=04
fileReservedTime=48
brokerRole=SYNC_MASTER
flushDiskType=ASYNC_FLUSH
[root@mybrokercluster-0-0 bin]# exit
exit
[root@k8s-master deploy]# kubectl exec -it mybrokercluster-0-1 -n rocketmq-oper
[root@mybrokercluster-0-1 bin]# cat ../conf/broker.conf
#This file was auto generated by rocketmq-operator. DO NOT EDIT.
brokerClusterName=DefaultCluster
brokerName=broker-0
brokerId=1
deleteWhen=04
fileReservedTime=48
brokerRole=SLAVE
flushDiskType=ASYNC_FLUSH
[root@mybrokercluster-0-1 bin]# exit
exit
```

```
[root@k8s-master deploy]# kubectl exec -it mybrokercluster-1-0 -n rocketmq-operator bash
[root@mybrokercluster-1-0 bin]# cat ../conf/broker.conf
#This file was auto generated by rocketmq-operator. DO NOT EDIT.
brokerClusterName=DefaultCluster
brokerName=broker-1
brokerId=0
deleteWhen=04
fileReservedTime=48
brokerRole=SYNC_MASTER
flushDiskType=ASYNC_FLUSH
[root@mybrokercluster-1-0 bin]# exit
exit
[root@k8s-master deploy]# kubectl exec -it mybrokercluster-1-1 -n rocketmq-operator bash
[root@mybrokercluster-1-1 bin]# cat ../conf/broker.conf
#This file was auto generated by rocketmq-operator. DO NOT EDIT.
brokerClusterName=DefaultCluster
brokerName=broker-1
brokerId=1
deleteWhen=04
fileReservedTime=48
brokerRole=SLAVE
flushDiskType=ASYNC_FLUSH
```

RocketMQ Operator Tutorial

Verify the cluster

Below is the command line output for a cluster with 2 masters and 2 slaves.

```
[root@mybrokercluster-1-1 bin]# ./mqadmin clusterList -n 10.10.103.181:9876 10.10.101.98:9876
OpenJDK 64-Bit Server VM warning: ignoring option PermSize=128m; support was removed in 8.0
OpenJDK 64-Bit Server VM warning: ignoring option MaxPermSize=128m; support was removed in 8.0
#Cluster Name     #Broker Name          #BID    #Addr                    #Version          #InTPS(L(
DefaultCluster    mybrokercluster-0-0     0      192.168.122.145:10911    V4_3_0                    0.00(0,(
DefaultCluster    mybrokercluster-0-1     0      192.168.196.17:10911     V4_3_0                    0.00(0,(
DefaultCluster    mybrokercluster-1-0     0      192.168.196.16:10911     V4_3_0                    0.00(0,(
DefaultCluster    mybrokercluster-1-1     0      192.168.122.146:10911    V4_3_0                    0.00(0,(
```

Project Improvements Plan

- More features are WIP

- Welcome to submit issue/ PR

- Github - https://github.com/huanwei/rocketmq-operator

Thanks!