59Kubernetes 系列（五十六）如何在 Kubernetes 中实现排他锁

Kubernetes 系列（五十六）如何在 Kubernetes 中实现排他锁

Kubernetes 集群中有一个应用程序在运行。目标是保护该应用程序不受任何操作修改，除非这些修改来自预定义的参与者。

配置要求

首先，让我们设置Kind集群和 Klock 。我想你已经安装了 Kind 。现在让我们创建一个集群并安装所有依赖:

1. 创建集群: `kind Create cluster`

2. 安装cert-manager: `kubectl apply -f [https://github.com/cert-manager/cert-manager/releases/download/v1.8.2/cert-manager.yaml](https://github.com/cert-manager/cert-manager/releases/download/v1.8.2/cert-manager.yaml)`

3.安装Klock:

```
helm repo add rnemet https://rnemet.dev/helm-charts
helm repo update
helm install klock rnemet/klock
```

设置场景

我的应用程序有一个 Pod 和一个 ConfigMap。默认情况下，Klock(https://github.com/robert-nemet/klock)支持锁定 Deployment，Pods, Secrets和configmap。我就讲到这里了。

让我们部署我们的应用程序，一个Pod和一个ConfigMap:

```
kubectl run my-pod --image nginx --labels aura=red
kubectl create configmap my-cm --from-literal nikola=tesla
kubectl label cm my-cm aura=red
```

增加一个新的 Pod:

```
kubectl run other-pod --image nginx
```

检查 default 命名空间下我们具备的资源:

```
⎈ kubectl get pods,cm
NAME            READY   STATUS    RESTARTS   AGE
pod/my-pod      1/1     Running   0          5m52s
pod/other-pod   1/1     Running   0          112s
NAME                          DATA   AGE
configmap/kube-root-ca.crt    1      18m
configmap/my-cm               1      4m20s
```

而我们的应用是:

```
⎈ kubectl get pod,cm -A --selector aura=red
NAMESPACE   NAME         READY   STATUS    RESTARTS   AGE
default     pod/my-pod   1/1     Running   0          7m2s
NAMESPACE   NAME              DATA   AGE
default     configmap/my-cm   1      5m30s
```

让我们添加一个 Service Account(SA)，用于我们所谓的操作员管理我们的应用程序:

```
kubectl create serviceaccount jonny-op
```

如果你试着用jonny-op列出 Pod:

```
⎈ kubectl run operator  --image roffe/kubectl --overrides='{"apiVersion":"v1","spec":{"serviceAccount":"jo
No resources found.
Error from server (Forbidden): pods is forbidden: User "system:serviceaccount:default:jonny-op" cannot lis
pod default/operator terminated (Error)
```

在这里，我试图用我们的假操作符列出默认名称空间中的所有pod。当操作员使用SA jonny-op运行时，它不能列出Pod。我们的应用程序如何:

```
kubectl delete pod operator
⎈ kubectl run operator  --image roffe/kubectl --overrides='{"apiVersion":"v1","spec":{"serviceAccount":"jo
No resources found.
Error from server (Forbidden): pods is forbidden: User "system:serviceaccount:default:jonny-op" cannot lis
Error from server (Forbidden): configmaps is forbidden: User "system:serviceaccount:default:jonny-op" cann
pod default/operator terminated (Error)
```

让我们通过创建Role和RoleBinding来解决这个问题:

```
kubectl create role app-op --verb list --verb get --verb create --verb update --verb delete --verb patch -
kubectl create rolebinding jonny-app --role app-op --serviceaccount default:jonny-op
```

然后进行验证:

```
⎈ kubectl delete pod operator
pod "operator" deleted
⎈ kubectl run operator  --image roffe/kubectl --overrides='{"apiVersion":"v1","spec":{"serviceAccount":"jo
If you don't see a command prompt, try pressing enter.
NAME         READY     STATUS    RESTARTS   AGE
pod/my-pod   1/1       Running   0          26m
NAME              DATA      AGE
configmap/my-cm   1         25m
```

注意:由于 operator pod不能更新，每次需要删除 operator pod。

运行场景

Me 作为外部 actor 可以修改Pod **my-pod** 和ConfigMap **my-cm** 。我创建了集群，在这种情况下我是管理员，所以这是很自然的。如所示，我们可以在 *default* 命名空间中使用SA **jonny-op** 或operator运行工作负载。但我的目标是公正的，只有 **jonny_op** 可以做到。而且只针对我的申请。

我的应用程序中的所有资源都以标签“aura=red”分组。那么，让我们创建一个锁:

```
kubectl apply -f - <<EOF
apiVersion: klock.rnemet.dev/v1
kind: Lock
metadata:
  name: lockred
spec:
  operations:
    - UPDATE
    - DELETE
  matcher:
    aura: red
  exclusive:
    name: jonny-op
EOF
```

现在，尝试去更新 Pod my-pod:

```
⎈ kubectl label pod my-pod aura=blue --overwrite
Error from server (denied, there is a lock: map[aura:red]): admission webhook "klocks.rnemet.dev" denied t
```

好的... **other-pod 又怎样呢** :

```
⎈ kubectl label pod other-pod aura=blue --overwrite
pod/other-pod labeled
```

所以，我作为管理员可以标记(更新) 没有标记为: `aura:red` 的Pod。那么SA jonny-op呢?

```
⎈ kubectl run operator  --image roffe/kubectl --overrides='{"apiVersion":"v1","spec":{"serviceAccount":"jo
pod/my-pod labeled
⎈ k get pods --show-labels
NAME        READY   STATUS      RESTARTS   AGE   LABELS
my-pod      1/1     Running     0          91m   aura=red,here=was-jonny
operator    0/1     Completed   0          25s   run=operator
other-pod   1/1     Running     0          87m   aura=blue,run=other-pod
```

ConfigMap **my-cm 呢** ?

```
⎈ kubectl label cm my-cm admin=was-here
Error from server (denied, there is a lock: map[aura:red]): admission webhook "klocks.rnemet.dev" denied t
⎈ k delete pods operator
pod "operator" deleted
⎈ kubectl run operator  --image roffe/kubectl --overrides='{"apiVersion":"v1","spec":{"serviceAccount":"jo
configmap/my-cm labeled
⎈ kubectl get cm --show-labels
NAME               DATA   AGE    LABELS
kube-root-ca.crt   1      108m   <none>
my-cm              1      94m    aura=red,here=was-jonny
```

相同。因此，我设法保护我的应用程序不受集群中的其他参与者的影响，而SA jonny-op可以对它进行操作。

总结

即使这是一个更加手工的例子，它也表明了在Kubernetes中使用Klock实现排他锁是可能的。

谢谢你阅读我的博客!

欢迎关注我的公众号“**云原生拓展**”，原创技术文章第一时间推送。