

# 21.ReplicaSet

[ReplicaSet与ReplicationController的区别](#)

[创建ReplicaSet](#)

[ReplicaSet的高级标签选择器](#)

[删除ReplicaSet](#)

在最初的时候，在Kubernetes中用于复制pod以及节点故障后重新调度pod的组件只有ReplicationController。后来引入了ReplicaSet，它是新一代的ReplicationController，用于取代旧有的ReplicationController。

通常我们不会直接创建ReplicaSet，而是通过创建更高级别的Deployment来自动创建ReplicaSet。需要记住的是：我们在以后的工作实践中，一定要使用ReplicaSet，而不是ReplicationController，因为后者很有可能在以后更新的版本中被完全废弃掉。

## ReplicaSet与ReplicationController的区别

ReplicaSet的功能类似ReplicationController，但是拥有表达方式更丰富的标签选择器。

ReplicationController的标签选择器只能匹配拥有某个标签的pod，而ReplicaSet标签选择器除此之外还能匹配缺少某个标签或者包含某个标签key的pod。

例如，一个ReplicationController的标签选择器不能同时匹配拥有env=dev标签的pod和拥有env=test标签的pod。而ReplicaSet则可以同时匹配这两组pod，并且将它们当做一个组。

同样地，ReplicationController无法只通过标签key去匹配pod，而ReplicaSet却可以做到。就像上面提到的情况，ReplicaSet可以通过env=\*来匹配标签key为env的pod。

## 创建ReplicaSet

在上一节中，我们将ReplicationController删掉了，所以当前集群中有三个孤立的、不受管控的pod：

```
[david@dhr-demo root]$ kubectl get po --show-labels
NAME          READY   STATUS    RESTARTS   AGE   LABELS
test-rc-8w759 1/1     Running   0           25h   app=test1,location=A1
test-rc-ph4lb 1/1     Running   0           2d1h  app=test1
test-rc-v4xsq 1/1     Running   0           2d1h  app=test1
[david@dhr-demo root]$
```

现在我们创建一个ReplicaSet来接管这三个pod。

首先创建ReplicaSet的定义文件：

vim test-replicase.yaml

```

apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: test-replicaset
spec:
  replicas: 3
  selector:
    matchLabels:
      app: test1
  template:
    metadata:
      labels:
        app: test1
    spec:
      containers:
        - name: test1
          image: registry.cn-shanghai.aliyuncs.com/david-ns01/test1:1.0
~
~

```

内容如下：

apiVersion: apps/v1

kind: ReplicaSet

metadata:

name: test-replicaset

spec:

replicas: 3

selector:

matchLabels:

app: test1

template:

metadata:

labels:

app: test1

spec:

containers:

– name: test1

image: registry.cn-shanghai.aliyuncs.com/david-ns01/test1:1.0

需要注意的是：

- apiVersion的值为app/v1（apps/v1包含一些通用的应用层的api组合，如：Deployments, RollingUpdates, and ReplicaSets），因为ReplicaSet并不是属于v1 API的部分，而是属于apps API组和v1版本
- 标签选择器部分与ReplicationController稍有不同，标签并不是直接放到selector下面，而是在中间多了一层matchLabels。
- pod模板部分与ReplicationController完全一样。

当前集群中的三个pod拥有标签app=test1，而这个ReplicaSet的标签选择器也是app=test1且replicas期望数为3，所以这三个pod刚好满足ReplicaSet的要求，因此如果我们创建这个ReplicaSet的话，就不会自动创建新的pod了。

与创建ReplicationController一样，我们仍然使用命令kubectl create来创建ReplicaSet：

```
kubectl create -f test-replicaset.yaml
```

```
[david@dhr-demo ~]$ kubectl create -f test-replicaset.yaml
replicaset.apps/test-replicaset created
[david@dhr-demo ~]$ kubectl get rs
NAME          DESIRED  CURRENT  READY  AGE
test-replicaset  3        3        3      13s
```

上图中，我们使用了kubectl get rs命令查看ReplicaSet列表。

在创建ReplicaSet后，可以看到仍然还是这个三个pod：

```
[david@dhr-demo ~]$ kubectl get po --show-labels
NAME          READY  STATUS   RESTARTS  AGE    LABELS
test-rc-8w759  1/1    Running  0          25h    app=test1,location=A1
test-rc-ph4lb  1/1    Running  0          2d1h   app=test1
test-rc-v4xsq  1/1    Running  0          2d1h   app=test1
[david@dhr-demo ~]$
```

执行如下命令查看这个ReplicaSet的具体信息：

```
kubectl describe rs
```

```
[david@dhr-demo ~]$ kubectl describe rs
Name:         test-replicaset
Namespace:    default
Selector:     app=test1
Labels:       <none>
Annotations:  <none>
Replicas:     3 current / 3 desired
Pods Status:  3 Running / 0 Waiting / 0 Succeeded / 0 Failed
Pod Template:
  Labels:  app=test1
  Containers:
    test1:
      Image:          registry.cn-shanghai.aliyuncs.com/david-ns01/test1:1.0
      Port:           <none>
      Host Port:      <none>
      Environment:    <none>
      Mounts:         <none>
      Volumes:        <none>
  Events:            <none>
[david@dhr-demo ~]$
```

## ReplicaSet的高级标签选择器

ReplicaSet拥有比ReplicationController更加高级的标签选择器。现在我们修改上面的ReplicaSet定义中的matchLabels属性，改用功能更加强大的matchExpressions属性：

```
apiVersion: apps/v1
```

```
kind: ReplicaSet
```

```
metadata:
```

```
  name: test-replicaset
```

```
spec:
```

```
  replicas: 3
```

```

selector:
  matchExpressions:
    - key: app
      operator: In
      values:
        - test1
template:
  metadata:
    labels:
      app: test1
  spec:
    containers:
      - name: test1
        image: registry.cn-shanghai.aliyuncs.com/david-ns01/test1:1.0

```

从上面的YAML定义文件可以看出该ReplicaSet的标签选择器匹配标签key为app、标签value包含test1的pod。如果某个pod拥有标签app=test2，则该ReplicaSet的标签选择器也能与其匹配。

对于matchExpressions属性，每个标签选择器的表达式必须包含一个key、一个operator以及相应的values（取决于operator的类型）。

operator的类型有如下4种：

- In-pod标签的值必须与values中指定的至少一个值匹配
- NotIn-pod标签的值不能是values中指定的任何一个值
- Exists-pod必须拥有一个具有指定key的标签（与值无关）。使用该operator时不能指定values属性。
- DoesNotExists-pod不能拥有具有指定key的标签。使用该operator时不能指定values属性。

如果我们同时指定多个标签选择器表达式，那么只有这些表达式全部为true的情况下才能完全匹配一个pod。用逻辑表达式的概念来表述的话，它们之间不是“或”的关系，而是“与”的关系。

同样地，如果我们既指定了matchLabels，又指定了matchExpressions表达式，那么只有两者所有的条件都满足的情况下才算与某个pod匹配成功。

## 删除ReplicaSet

删除ReplicaSet与删除ReplicationController的方式一样，只需要执行：

```
kubectl delete rs test-replicaset
```

```
[david@dhr-demo root]$ kubectl get po
NAME          READY   STATUS    RESTARTS   AGE
test-rc-8w759 1/1     Running   0           35h
test-rc-ph4lb 1/1     Running   0           2d11h
test-rc-v4xsq 1/1     Running   0           2d11h
[david@dhr-demo root]$ kubectl delete rs test-replicaset
replicaset.apps "test-replicaset" deleted
[david@dhr-demo root]$ kubectl get po
NAME          READY   STATUS    RESTARTS   AGE
test-rc-8w759 1/1     Terminating 0           35h
test-rc-ph4lb 1/1     Terminating 0           2d11h
test-rc-v4xsq 1/1     Terminating 0           2d11h
[david@dhr-demo root]$ kubectl get po
NAME          READY   STATUS    RESTARTS   AGE
test-rc-8w759 1/1     Terminating 0           35h
test-rc-ph4lb 1/1     Terminating 0           2d11h
test-rc-v4xsq 1/1     Terminating 0           2d11h
[david@dhr-demo root]$ kubectl get po
No resources found in default namespace.
[david@dhr-demo root]$
```

可以看到相应的pod也被删除掉了。