## 使用 descheduler 平衡 pod 在 worker 上的分布

## 介绍

k8s 本身是缺少对 pod 再调度的功能的,假设当前环境里有 2 个 worker,通过 deployment 创建了 6 个副本,那么这 6 个副本会分布再这两个 worker 上运行。

如果此时增加了一个新的 worker,那么已经运行的 pod 是没法调度到第三个节点上的,除非是手动删除这些 pod 让其重新调度。

descheduler 解决的就是 pod 再调度的问题的。当集群中新增加了节点,descheduler 会平衡一下 worker 上的 pod 数,即把一些已经存在的 pod 调度到新的 worker 上。

## 环境准备

```
[root@vms61 2-pod]# kubectl get nodes
NAME
                 STATUS
                                           VERSION
                          ROLES
                                     AGE
vms61.rhce.cc
                 Ready
                                     10d
                                           v1.18.2
                          master
                                           v1.18.2
vms62.rhce.cc
                 Ready
                          worker1
                                     10d
vms63.rhce.cc
                 Ready
                          worker2
                                     10d
                                           v1.18.2
[root@vms61 2-pod]#
```

先把 vms63 设置为不可用:

```
[root@vms61 2-pod]# kubectl cordon vms63.rhce.cc
node/vms63.rhce.cc cordoned
[root@vms61 2-pod]# kubectl get nodes
                STATUS
                                             ROLES
                                                       AGE
                                                             VERSION
vms61.rhce.cc
                                                       10d
                                                             v1.18.2
                Ready
                                             master
vms62.rhce.cc
                                                       10d
                                                             v1.18.2
                Ready
                                             worker1
                Ready, Scheduling Disabled
                                                             v1.18.2
vms63.rhce.cc
                                            worker2
                                                       10d
[root@vms61 2-pod]#
```

创建含有 6 个副本的 deployment,这 6 个副本应该都是在 vms62 上运行的:

```
[root@vms61 2-pod]# kubectl apply -f web1.yaml
deployment.apps/web1 created
[root@vms61 2-pod]#
[root@vms61 2-pod]# kubectl get pods -o wide
                       READY
                                         RESTARTS AGE IP
NAME
                                                                           NODE
                               STATUS
GATES
web1-7c7c8cdd9d-d2xcb
                       1/1
                               Running
                                                    23s
                                                          10.244.118.231
                                                                           vms62.rhce.cc
web1-7c7c8cdd9d-fcvbf
                       1/1
                               Running
                                                    23s
                                                          10.244.118.227
                                                                           vms62.rhce.cc
web1-7c7c8cdd9d-gvwjj
                       1/1
                               Running
                                         0
                                                    23s
                                                          10.244.118.226
                                                                           vms62.rhce.cc
                                                    23s
web1-7c7c8cdd9d-lmpmt
                       1/1
                               Running
                                                          10.244.118.230
                                                                           vms62.rhce.cc
web1-7c7c8cdd9d-wjnpp
                       1/1
                               Running
                                         0
                                                    23s
                                                          10.244.118.229
                                                                           vms62.rhce.cc
web1-7c7c8cdd9d-x7m4d
                                                          10.244.118.228
                               Running
                                                                           vms62.rhce.cc
[root@vms61 2-pod]#
```

现在把 vms63 设置为可用:

```
[root@vms61 2-pod]# kubectl uncordon vms63.rhce.cc
node/vms63.rhce.cc uncordoned
[root@vms61 2-pod]# kubectl get nodes
NAME
                STATUS
                         ROLES
                                    AGE
                                          VERSION
vms61.rhce.cc
                                          v1.18.2
                Ready
                                    10d
                         master
vms62.rhce.cc
                         worker1
                                    10d
                                          v1.18.2
                Ready
vms63.rhce.cc
                         worker2
                                          v1.18.2
                Ready
                                    10d
[root@vms61 2-pod]#
```

这 6 个副本依然是不会运行在 vms63 上的。

## 部署 descheduler

descheduler 的本质就是利用计划任务定期去检测 pod 在节点的分布,然后根据自己的算去平衡每个节点上的 pod 数。

项目地址 https://github.com/kubernetes-sigs/descheduler 下载地址 https://codeload.github.com/kubernetes-sigs/descheduler/zip/master 下载解压之后,进入目录:

```
[root@vms61 kubernetes]# pwd
/root/descheduler-master/kubernetes
[root@vms61 kubernetes]# ls
configmap.yaml cronjob.yaml job.yaml rbac.yaml
[root@vms61 kubernetes]#
```

在所有节点上下载镜像 us.gcr.io/k8s-artifacts-prod/descheduler/descheduler:v0.18.0 修改 cronjob.yaml 的内容,把镜像下载策略设置为 IfNotPresent,并把 cronjob 的间隔设置为 1 分钟

```
spec:
    schedule: "*/1 * * * *"
    concurrencyPolicy: "Forbid"
    jobTemplate:
```

保存退出。

检查 kube-system 命名空间里 cronjob,当此 cronjob 运行之后:

[root@vms61 ~]# kubec	tl get cj -n k	ube-system			
NAME	SCHEDULE	SUSPEND	ACTIVE	LAST SCHEDULE	AGE
descheduler-cronjob	*/1 * * * *	False	0	<none></none>	50s
[root@vms61 ~]# kubec	tl get cj -n k	ube-system			
NAME	SCHEDULE	SUSPEND	ACTIVE	LAST SCHEDULE	AGE
descheduler-cronjob	*/1 * * * *	False	1	4s	55s
[root@vms61 ~]#					

descheduler 就开始重新调度 pod 的运行:

	-					
[root@vms61 2-pod]# kul NAME NESS GATES	READY	STATUS	RESTARTS	AGE	IP	NODE
web1-7c7c8cdd9d-8r464	1/1	Running	0	85	10.244.85.63	vms63.rhce.cc
web1-7c7c8cdd9d-cnvfm	1/1	Running	0	9s	10.244.85.58	vms63.rhce.cc
web1-7c7c8cdd9d-d2xcb	1/1	Running	0	11m	10.244.118.231	vms62.rhce.cc
web1-7c7c8cdd9d-gvwjj	0/1	Terminating	0	11m	<none></none>	vms62.rhce.cc
web1-7c7c8cdd9d-jkmgf	1/1	Running	0	85	10.244.85.59	vms63.rhce.cc
web1-7c7c8cdd9d-lmpmt	0/1	Terminating	0	11m	<none></none>	vms62.rhce.cc
web1-7c7c8cdd9d-psgl2	1/1	Running	0	8s	10.244.85.61	vms63.rhce.cc
web1-7c7c8cdd9d-q8hsq >	1/1	Running	0	8s	10.244.85.62	vms63.rhce.cc
[root@vms61 2-pod]#						

[root@vms61 2-pod]# ku NAME	bectl ge READY	t pods -o STATUS	wide RESTARTS	AGE	IP	NODE
GATES web1-7c7c8cdd9d-8r464 web1-7c7c8cdd9d-cnvfm web1-7c7c8cdd9d-d2xcb web1-7c7c8cdd9d-jkmgf web1-7c7c8cdd9d-psgl2 web1-7c7c8cdd9d-q8hsq	1/1 1/1 1/1 1/1 1/1 1/1	Running Running Running Running Running Running	0 0 0 0	11s 12s 11m 11s 11s	10.244.85.63 10.244.85.58 10.244.118.231 10.244.85.59 10.244.85.61 10.244.85.62	vms63.rhce.cc vms63.rhce.cc vms62.rhce.cc vms63.rhce.cc vms63.rhce.cc vms63.rhce.cc
<pre>[root@vms61 2-pod]# [root@vms61 2-pod]# kub NAME</pre>	ectl get READY	pods -o w STATUS	vide RESTARTS	AGE	IP	NODE
web1-7c7c8cdd9d-2685v web1-7c7c8cdd9d-795p2 web1-7c7c8cdd9d-7n82s web1-7c7c8cdd9d-8r464 web1-7c7c8cdd9d-d2xcb	1/1 1/1 1/1 1/1 1/1	Running Running Running Running Running	0 0 0 0	57s 57s 57s 2m56s 14m	10.244.85.2 10.244.85.5 10.244.118.235 10.244.85.63 10.244.118.231	vms63.rhce.cc vms63.rhce.cc vms62.rhce.cc vms63.rhce.cc vms62.rhce.cc
web1-7c7c8cdd9d-lg2pn [root@vms61 2-pod]#	1/1	Running	0	57s	10.244.85.6	vms63.rhce.cc

可以看到 pod 已经被调度到不同的节点上运行了。