

拉勾教育

— 互联网人实战大学 —

《Kubernetes 原理剖析与实战应用》

正范

— 拉勾教育出品 —

29 | Kubernetes 中也有定时任务吗？

Deployment、Statefulset、Daemonset 工作负载

可以在不同的场景下运行长伺型（Long Running）的服务



```
apiVersion: batch/v1beta1
kind: Job
metadata:
  name: pi
spec:
  template:
    spec:
      containers:
      - name: pi
        image: perl
        command: ["perl", "-Mbignum=bpi", "-wle", "print
bpi (2000)"]
      restartPolicy: Never
      backoffLimit: 4
```

```
$ kubectl create -f
https://kubernetes.io/examples/controllers/job.yaml
job.batch/pi created
```

```
kubectl describe jobs/pi
Name:          pi
Namespace:     default
Selector:      controller-uid=4f8027d0-cac1-42ea-b5f8-dbb4d9c9f67a
Labels:        controller-uid=4f8027d0-cac1-42ea-b5f8-dbb4d9c9f67a
               job-name=pi
Annotations:   <none>
Parallelism:   1
Completions:   1
Start Time:    Mon, 02 Dec 2020 15:04:52 +0200
Completed At:  Mon, 02 Dec 2020 15:06:39 +0200
Duration:      65s
Pods Statuses: 0 Running / 1 Succeeded / 0 Failed
Pod Template:
  Labels:  controller-uid=c9948307-e56d-4b5d-8302-ae2d7b7da67c
          job-name=pi
  Containers:
  pi:
    ...
Events:
  Type    Reason            Age   From          Message
  ----    -
  Normal  SuccessfulCreate  4m    job-controller  Created pod: pi-jk2k7
```

```
kubectl get pods --selector=job-name=pi
```

```
kubectl get pods --selector=controller-uid=4f8027d0-cac1-42ea-b5f8-dbb4d9c9f67a
```


NAME	READY	STATUS	RESTARTS
AGE			
pi-ik2k7	0/1	Completed	0

2m

NAME	READY	STATUS	RESTARTS
AGE			
pi-ik2k7	0/1	Completed	0
2m			

Job 控制器会根据 spec.backoffLimit 中定义的数值来限制 Pod 失败的次数

```
apiVersion: batch/v1
kind: Job
metadata:
  name: job-wq-2
spec:
  parallelism: 2
  template:
    metadata:
      name: job-wq-2
    spec:
      containers:
        - name: c
          image: gcr.io/myproject/job-wq-2
      restartPolicy: OnFailure
```

官方文档

<https://kubernetes.io/zh/docs/concepts/workloads/controllers/job/#job-patterns>

```
apiVersion: batch/v1
kind: Job
metadata:
  name: pi-with-ttl
spec:
  ttlSecondsAfterFinished: 100
  template:
    spec:
      containers:
      - name: pi
        image: perl
        command: ["perl", "-Mbignum=bpi", "-wle", "print
bpi (2000)"]
      restartPolicy: Never
```

```
apiVersion: batch/v1beta1
kind: CronJob
metadata:
  name: hello # cronjob 名字
spec:
  schedule: "*/1 * * * *" # job执行的周期, 通过 cron 格式
来标明
  jobTemplate: # job模板
    spec:
      template:
        spec:
          containers:
            - name: hello
              image: busybox
              imagePullPolicy: IfNotPresent
              args:
                - /bin/sh
                - c
                - date; echo Hello from the Kubernetes cluster
          restartPolicy: OnFailure
```

Cron 的基本格式

〈分钟〉 〈小时〉 〈日〉 〈月〉 〈星期〉

Cron

还支持 “*” “/” 等字符

* 是个通配符，可以匹配任何值

/ 表示起始时间触发，然后每隔一个固定时间触发一次



CronJob 会帮助管理 Job

不需要配置TTL 自动清理, CronJob 控制器会自动清理



1

设置为 Allow, 这也是默认的值, 允许并发任务的执行

2

设置为 Forbid, 不允许并发任务执行

3

设置为 Replace, 用新的 Job 来替换当前正在运行的老的 Job

Next: [《30 | Kubectl 命令行工具使用秘笈》](#)

拉勾教育

— 互联网人实战大学 —



关注拉勾「教育公众号」
获取更多课程信息