

k8s集群核心概念

Controller(控制器)

一、场景

通过前面课程的学习，播仔已经掌握了Pod管理操作，我们会发现，当我们删除Pod时是可以直接删除的，如果在生产过程中误操作，Pod同样也会轻易被删除的，因此，我们需要在k8s集群中引入另一种概念Controller(控制器)，用于在k8s集群中以loop方式监视Pod状态，如果其发现Pod被删除，将会重新拉起一个Pod，以让Pod一直保持在用户期望状态。

二、学习目标

- ❑ 了解Controller作用
- ❑ 了解Controller分类
- ❑ 了解Deployment控制器作用
- ❑ 掌握创建Deployment控制器类型应用方法
- ❑ 掌握删除Deployment控制器类型应用方法

三、学习步骤

| 序号 | 步骤 | 备注 |
|----|---------------------|----|
| 1 | Controller介绍 | |
| 2 | Controller分类 | |
| 3 | Deployment控制器介绍 | |
| 4 | 创建Deployment控制器类型应用 | |
| 5 | 删除Deployment控制器类型应用 | |

四、课程内容

4.1 Controller介绍

- 控制器
- 用于对应用运行的资源对象进行监控
- 当Pod出现问题时，会把Pod重新拉起，以达到用户的期望状态

4.2 Controller分类

常见Pod控制器：

| 控制器名称 | 作用 |
|-------------|--------------------------------------|
| Deployment | 声明式更新控制器，用于发布无状态应用 |
| ReplicaSet | 副本集控制器，用于对Pod进行副本规模扩大或剪裁 |
| StatefulSet | 有状态副本集，用于发布有状态应用 |
| DaemonSet | 在k8s集群每一个Node上运行一个副本，用于发布监控或日志收集类等应用 |
| Job | 运行一次性作业任务 |
| CronJob | 运行周期性作业任务 |

4.3 Deployment控制器介绍

- 具有上线部署、滚动升级、创建副本、回滚到以前某一版本（成功/稳定）等功能
- Deployment包含ReplicaSet，除非需要自定义升级功能或者根本不需要升级Pod，否则还是建议使用Deployment而不直接使用Replica Set。

4.4 创建Deployment控制器类型应用

4.4.1 通过kubectl命令行创建

```
1 命令
2 [root@master1 ~]# kubectl run nginx-app --
  image=nginx:latest --image-pull-
  policy=IfNotPresent --replicas=2
```

```
1 输出
2 deployment.apps/nginx-app created
```

```
1 说明
2 nginx-app 是deployment控制器类型的应用名称
3 nginx:latest 是应用运行的Pod中的Container所使用的
  镜像
4 IfNotPresent 是Container容器镜像下载策略，如果本
  地有镜像，使用本地，如果本地没有镜像，下载镜像
5 --replicas=2 是指应用运行的Pod共计2个副本，这是用
  户的期望值，Deployment控制器中的ReplicaSet控制器会
  一直监控此应用运行的Pod副本状态，如果数量达不到用户期
  望，就会重新拉起一个新的Pod，会让Pod数量一直维持在用
  户期望值数量。
```

```
1 命令
2 查看Deployment控制器类型应用
3 [root@master1 ~]# kubectl get deployment.apps
```

```
1 输出
2 NAME                READY    UP-TO-DATE    AVAILABLE
   AGE
3 nginx-app           2/2      2              2
   2m46s
```

```
1 命令
2 查看ReplicaSet控制器类型
3 [root@master1 ~]# kubectl get replicaset
```

```
1 输出
2 NAME                DESIRED    CURRENT
   READY    AGE
3 nginx-app-84c4bb6478 2           2           2
   3m14s
```

```
1 命令
2 查看Pod状态
3 [root@master1 ~]# kubectl get pods
```

```

1 输出
2  NAME                                READY   STATUS
   RESTARTS   AGE
3  nginx-app-84c4bb6478-c72fj         1/1     Running
   0           3m37s
4  nginx-app-84c4bb6478-cxjqn         1/1     Running
   0           3m37s

```

```

1 命令
2 查看Pod状态，显示更宽泛信息
3 [root@master1 ~]# kubectl get pods -o wide

```

```

1 输出
2  NAME                                READY   STATUS   NODE
   RESTARTS   AGE      IP             READINESS GATES
3  nginx-app-84c4bb6478-c72fj         1/1     Running  work1
   0           5m52s    172.16.215.5
   <none>      <none>
4  nginx-app-84c4bb6478-cxjqn         1/1     Running  work2
   0           5m52s    172.16.123.56
   <none>      <none>

```

```
1 命令
2 访问Pod
3 [root@master1 ~]# curl http://172.16.215.5
4
5 [root@master1 ~]# curl http://172.16.123.56
```

```
1 输出
2 访问Pod输出web页面信息
3 <!DOCTYPE html>
4 <html>
5 <head>
6 <title>welcome to nginx!</title>
7 <style>
8     body {
9         width: 35em;
10        margin: 0 auto;
11        font-family: Tahoma, Verdana, Arial,
12        sans-serif;
13    }
14 </style>
15 </head>
16 <body>
17 <h1>welcome to nginx!</h1>
18 <p>If you see this page, the nginx web
19 server is successfully installed and
20 working. Further configuration is required.
    </p>
    <p>For online documentation and support
    please refer to
```

```
21 <a href="http://nginx.org/">nginx.org</a>.  
    <br/>  
22 Commercial support is available at  
23 <a href="http://nginx.com/">nginx.com</a>.  
    </p>  
24  
25 <p><em>Thank you for using nginx.</em></p>  
26 </body>  
27 </html>
```

4.4.2 通过资源清单文件创建

4.4.2.1 编写用于创建Deployment控制器类型应用资源清单

```
1 [root@master1 ~]# cat 03-creat-deployment-  
  app.yaml  
2 apiVersion: apps/v1  
3 kind: Deployment  
4 metadata:  
5   name: nginx-apps  
6   labels:  
7     app: nginx  
8 spec:  
9   replicas: 1  
10  selector:  
11    matchLabels:
```



```
12     app: nginx
13   template:
14     metadata:
15       labels:
16         app: nginx
17     spec:
18       containers:
19       - name: nginxapp
20         image: nginx:latest
21         imagePullPolicy: IfNotPresent
22         ports:
23         - containerPort: 80
```

4.4.2.2 应用创建Deployment控制器类型应用资源清单

```
1 命令
2 [root@master1 ~]# kubectl apply -f 03-creat-
  deployment-app.yaml
```

```
1 输出
2 deployment.apps/nginx-apps created
```

4.4.2.3 验证资源清单应用后结果

```
1 命令
2 查看Deployment控制器类型应用
3 [root@master1 ~]# kubectl get deployment.apps
```

```
1 输出
2 NAME                READY    UP-TO-DATE    AVAILABLE
   AGE
3 nginx-apps          1/1      1              1
   78s
```

```
1 命令
2 查看ReplicaSet控制器类型应用
3 [root@master1 ~]# kubectl get replicaset.apps
```

```
1 输出
2 NAME                DESIRED    CURRENT
   READY    AGE
3 nginx-apps-66dcfbcd96  1          1          1
   105s
4
```

```
1 命令
2 查看Pod状态
3 [root@master1 ~]# kubectl get pods
```

```
1 输出
2  NAME                                READY   STATUS
   RESTARTS   AGE
3  nginx-apps-66dcfbcd96-c8bjv        1/1     Running
   0                2m9s
```

4.5 删除Deployment控制器类型应用

带有控制器类型的Pod不能随便删除，如果必须删除，请删除控制器类型的应用名称。

4.5.1 通过kubectl命令行删除

```
1 命令
2 [root@master1 ~]# kubectl get deployment.apps
```

```
1 输出
2 NAME                READY    UP-TO-DATE    AVAILABLE
  AGE
3 nginx-app           2/2      2              2
  20m
```

```
1 命令
2 [root@master1 ~]# kubectl delete
  deployment.apps nginx-app
```

```
1 输出
2 deployment.apps "nginx-app" deleted
```

4.5.2 通过kubectl应用资源清单文件删除

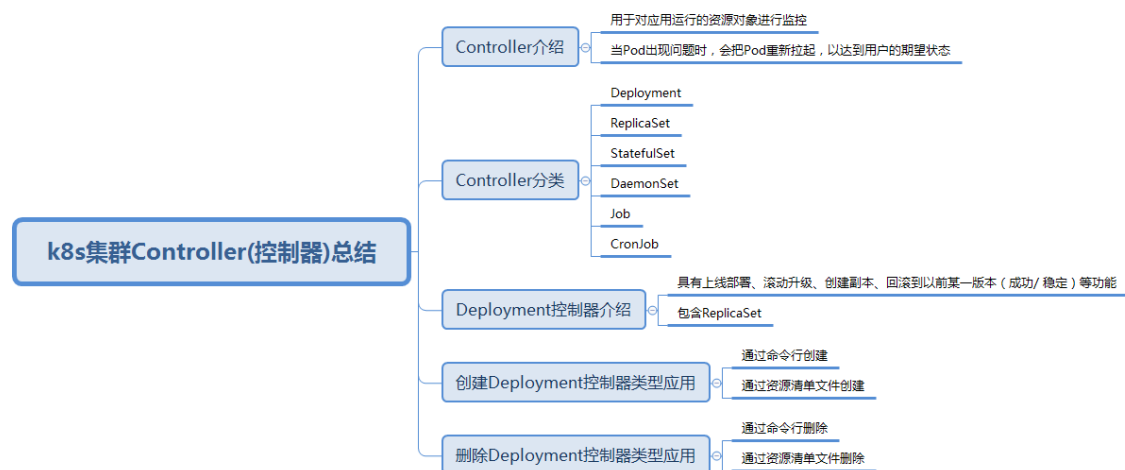
```
1 命令
2 [root@master1 ~]# kubectl get deployment.apps
```

```
1 输出
2 NAME                READY    UP-TO-DATE    AVAILABLE
  AGE
3 nginx-apps          1/1      1              1
  8m15s
```

```
1 命令
2 [root@master1 ~]# kubectl delete -f 03-creat-
  deployment-app.yaml
```

```
1 输出
2 deployment.apps "nginx-apps" deleted
```

五、学习总结



六、课程预约

深入学习Controller(控制器)，可以预约《kubernetes集群从入门到企业应用实战》相关课程。

