Pod管理与使用

前言

- 本章节主要讲述Pod的概念，如何使用Pod，如何编写Yaml格式文件，如何使用不同类型的Pod等。

目标

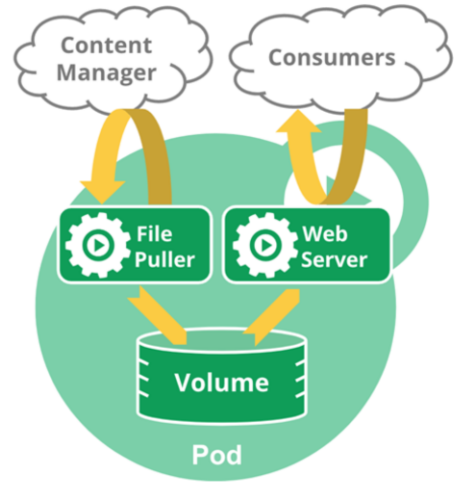
- 学完本课程后，您将能够：
  - 描述Pod特性
  - 区分单容器Pod和多容器Pod
  - 区分运行常驻任务的Pod和运行一次性任务的Pod

目录

1. **Pod基本概念**
2. 使用Pod

Pod

- Pod是Kubernetes管理的最小基础单元。
- 一个Pod中封装了：
  - 一个或多个紧耦合的应用容器
  - 存储资源
  - 独立的IP
  - 容器运行的选项



HUAWEI

- 图中pod为多容器pod。

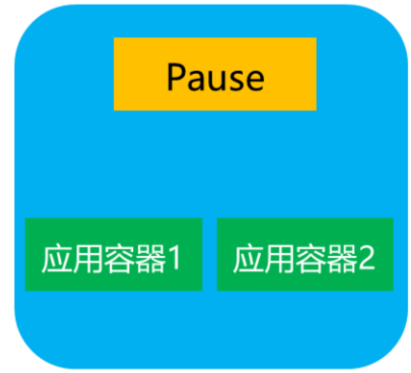
Pod的两种模式

- 只包含一个应用容器的pod
  - “一个pod一个容器”的模式是在Kubernetes中主流的使用场景。
  - 在这种场景中，pod可以被看做是一个“包装纸”包着的容器。Kubernetes不能直接管理容器，而是需要通过管理pod来管理容器
- 包含多个应用的pod
  - 仅当两种容器紧耦合，且需要共享相同的资源时使用这一种模式。
  - 这些在一个pod内的容器形成一个统一的服务单元。例如一个容器从共享卷提供文件，而另一个容器刷新或更新资源。Pod将这些容器和存储资源作为单个可管理的实体包装在一起。

- 尽量使用单容器pod，避免pod管理复杂化

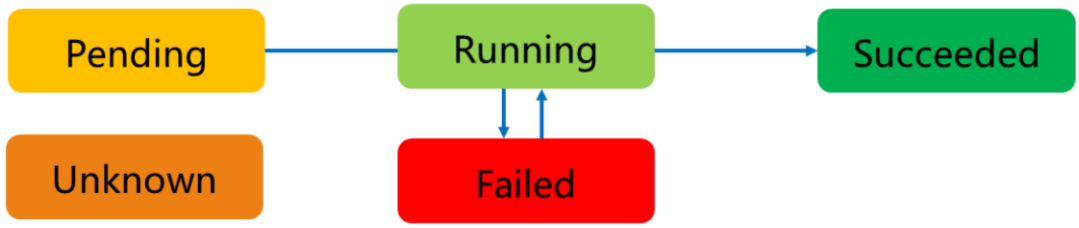
Pod内部结构

- 当一个多容器的Pod中有一个容器故障了，这个pod是故障还是正常?
- 一个pod共享ip，那这个IP应该挂载在容器1上还是容器2上?
- 一个pod中会分配一个pause容器，这也被称为根容器。
  - Pause容器的状态代表整个pod的状态
  - Pod中多个容器共享pause容器的网络，容器间可以通过localhost互访。



Pod生命周期

- Pod一旦被创建，会被master调度到某个具体的node上进行绑定。Pod会呈现出不同的状态。



HUAWEI

- 挂起（Pending）：Pod 已被 Kubernetes 系统接受，但有一个或者多个容器镜像尚未创建。等待时间包括调度 Pod 的时间和通过网络下载镜像的时间，这可能需要花点时间。
- 运行中（Running）：该 Pod 已经绑定到了一个节点上，Pod 中所有的容器都已被创建。至少有一个容器正在运行，或者正处于启动或重启状态。
- 成功（Succeeded）：Pod 中的所有容器都被成功终止，并且不会再重启。
- 失败（Failed）：Pod 中的所有容器都已终止了，并且至少有一个容器是因为失败终止。也就是说，容器以非0状态退出或者被系统终止。
- 未知（Unknown）：因为某些原因无法取得 Pod 的状态，通常是因为与 Pod 所在主机通信失败。

目录

1. Pod基本概念
2. **使用Pod**

创建一个pod

- 创建一个yaml文件

```
kind: Pod
apiVersion: v1
metadata:
  name: mypod
spec:
  containers:
    - name: mypod
      image: busybox
      args:
      - /bin/sh
      - -c
      - sleep 30000
```

- 使用create命令由该文件创建pod

```
kubectl create -f mypod.yaml
```

- 使用get命令查看

```
[root@k8s-master]# kubectl get pod
NAME    READY   STATUS    RESTARTS   AGE
mypod   1/1     Running   0          39m
```


HUAWEI

- 关于yaml的讲解我们放在下一章节中
- 可以通过kubectl explain pod.spec之类的指令，查看yaml详细参数

Pod管理常用命令

- 查看pod命令时，通过添加“-o=wide”参数获取更多信息

```
[root@k8s-master]# kubectl get pod -o=wide
NAME    READY   STATUS    RESTARTS   AGE   IP             NODE        NOMINATED NODE   READINESS GATES
mypod   1/1     Running   0          40m   192.169.2.42   k8s-node2   <none>           <none>
```

- 使用describe命令，可以看到pod的完整信息

```
kubectl describe pod mypod
```

- 删除Pod，使用delete命令

```
kubectl delete pod mypod
```


- 关于yaml的讲解我们放在下一章节中

Pod管理常用命令

- 进入mypod的命令行，

```
kubectl exec -it mypod /bin/sh
```

  - exec命令用于在容器中执行命令
  - -it参数使得用户可以直接进入容器进行操作。
- 在pod中可能存在多个容器，加入“--container *容器名*“参数指定进入容器。

```
kubectl exec -it mypod --container mypod /bin/sh
```

- 如果希望退出当前所在容器，可以输入exit回车退出。

HUAWEI

- 容器名在yaml文件中定义

Pod内容器查看

- 在之前使用命令查看pod状态时，可以检索到Pod所在的位置。

```
[root@k8s-master]# kubectl get pod -o=wide
NAME    READY   STATUS    RESTARTS   AGE   IP             NODE        NOMINATED NODE   READINESS GATES
mypod   1/1     Running   0          40m   192.169.2.42   k8s-node2   <none>           <none>
```

- 这意味着这个pod被绑定到k8s-node2这台主机上运行，我们可以登录这台主机，通过docker命令查看创建了哪些容器。

```
 docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED       STATUS       NAMES
3ad16a2860dc   busybox   "/bin/sh -c 'sleep 3…"   2 hours ago   Up 2 hours   k8s_mypod_mypod_default……
29d082178153   k8s.gcr.io/pause:3.1   "/pause"   2 hours ago   Up 2 hours   k8s_POD_mypod_default……
```

  - Busybox 这个容器是我们在创建pod时通过yaml文件定义的容器。
  - Pause是kubernetes默认会为pod拉起的容器

- 通过NAME列的参数可以关联到Pod中运行的容器

运行第二个pod

- 使用如下的helloworld.yaml文件创建第二个pod

```
kind: Pod
apiVersion: v1
metadata:
  name: helloworld
spec:
  restartPolicy: Never
  containers:
    - name: helloworld
      image: hello-world
```

- 查看pod状态会发现“helloworld”和“mypod”这两个pod处于不同的状态。

```
kubectl get pods
NAME         READY   STATUS      RESTARTS   AGE
helloworld   0/1     Completed   0          4s
mypod        1/1     Running     0          160m
```

  HUAWEI

- 有一些pod用于执行常驻任务，如mysql，提供文件服务等等，而一些pod用于执行一次性任务，执行完后即关闭。

实验&实训任务

- 实验任务
  - 请按照实验手册2.4章节完成Pod相关实验，包括：
    - 创建Pod
    - 运行一次性Pod
- 实训任务
  - 请灵活使用本章节课程及实验手册中学到的知识，按照实验手册2.4.3章节完成Pod实训任务。

本章总结

- 本章节主要讲述：
  - Pod的概念
  - 如何使用Pod
  - 如何编写Yaml格式文件
  - 如何使用不同类型的Pod


谢 谢
www.huawei.com