

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 —

04 | 核心定义：Kubernetes 是如何搞定“不可变基础设施”的？

上节课了解了 Kubernetes 集群的搭建方式

本节课学习 Kubernetes 中最重要、也最核心的对象——Pod



CNCF 官方定义云原生

云原生技术有利于各组织在公有云、私有云和混合云等新型动态环境中构建和运行可弹性扩展的应用

云原生的代表技术包括容器、服务网格、微服务、**不可变基础设施**和声明式API

这些技术能够构建容错性好、易于管理和便于观察的松耦合系统

结合可靠的自动化手段，云原生技术使工程师能够轻松地对系统作出频繁和可预测的重大变更

怎么理解不可变基础设施？

拉勾教育

— 互联网人实战大学 —

不可变基础设施

这个名词最早由 Chad Fowler 于 2013 年在他的文章

“[Trash Your Servers and Burn Your Code: Immutable Infrastructure and Disposable Components](#)” 中提出来

怎么理解不可变基础设施？

拉勾教育

— 互联网人实战大学 —

跟不可变基础设施相对的，我们称之为**可变基础设施**

怎么理解不可变基础设施？

拉勾教育

— 互联网人实战大学 —

跟不可变基础设施相对的，我们称之为**可变基础设施**

比如 kernel 升级、配置更新、打补丁等

怎么理解不可变基础设施?

拉勾教育

— 互联网人实战大学 —

- 持续的变更修改给服务运行态引入过多的中间态，增加了不可预知的风险
 - 故障发生时，难以及时快速构建出新的服务副本
 - 不易标准化，交付运维过程异常痛苦，虽然可以通过 Ansible、Puppet 等部署工具进行交付
- 但是也很难保证对底层各种异构的环境支持得很好，还有随时会出现的版本漂移问题



怎么理解不可变基础设施？

拉勾教育

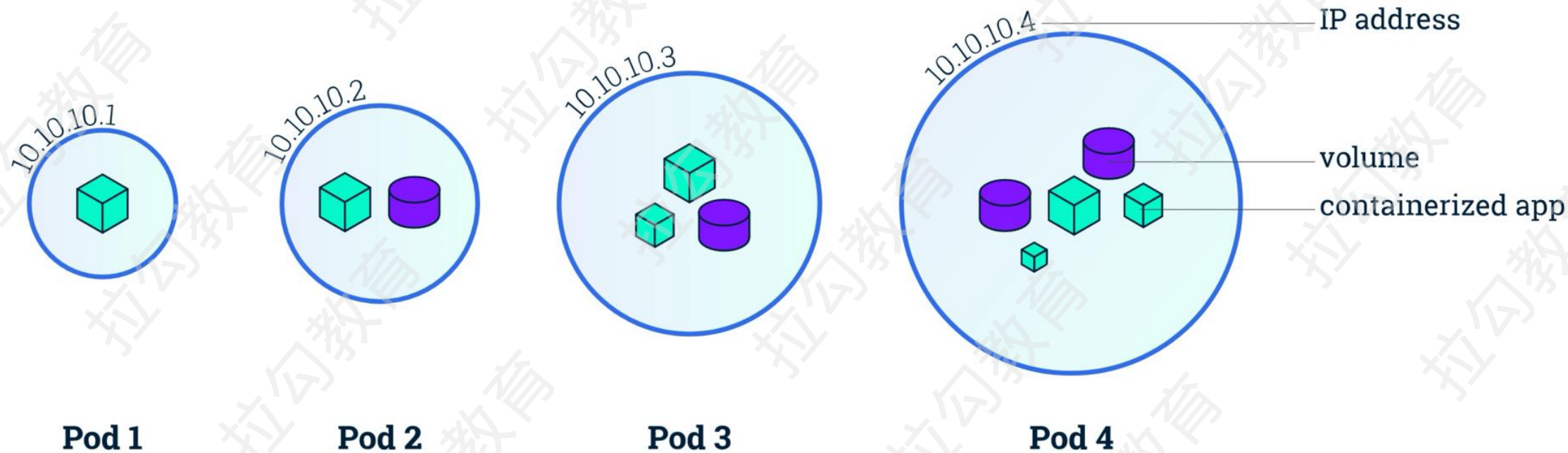
— 互联网人实战大学 —

Kubernetes 中的不可变基础设施就是 Pod

Pod 是什么

拉勾教育

— 互联网人实战大学 —



Pod 是什么

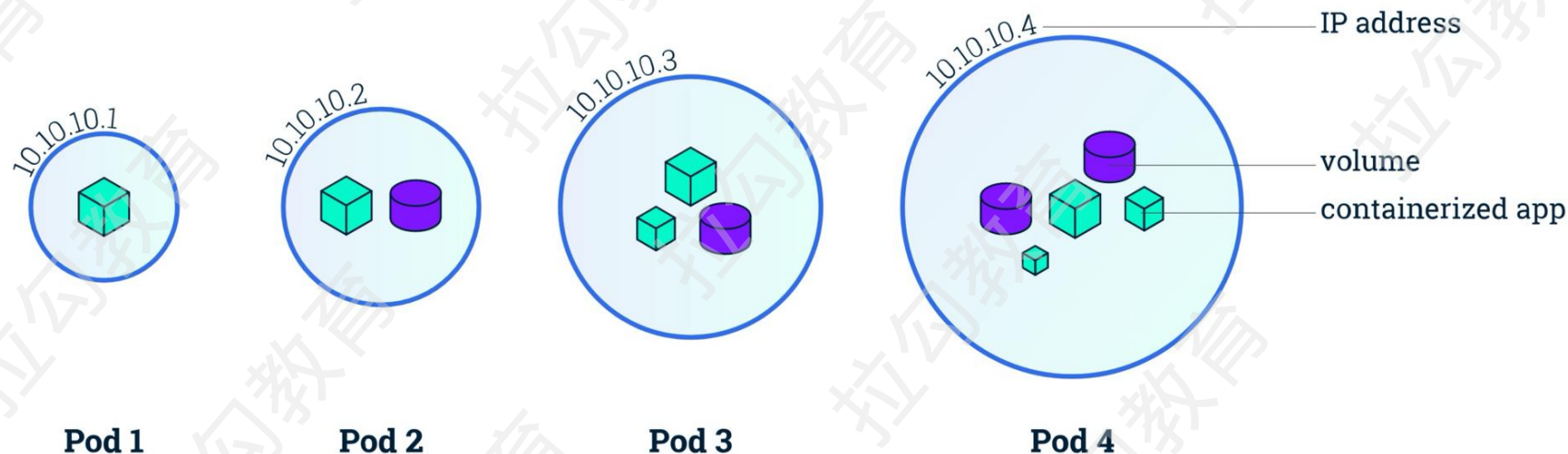
同一个 Pod 中的容器共享网络、存储资源

- 每个 Pod 都会拥有一个独立的网络空间，其内部的所有容器都共享网络资源，即 IP 地址、端口

内部的容器直接通过 localhost 就可以通信

- Pod 可以挂载多个共享的存储卷（Volume）

这时内部的各个容器就可以访问共享的 Volume 进行数据的读写

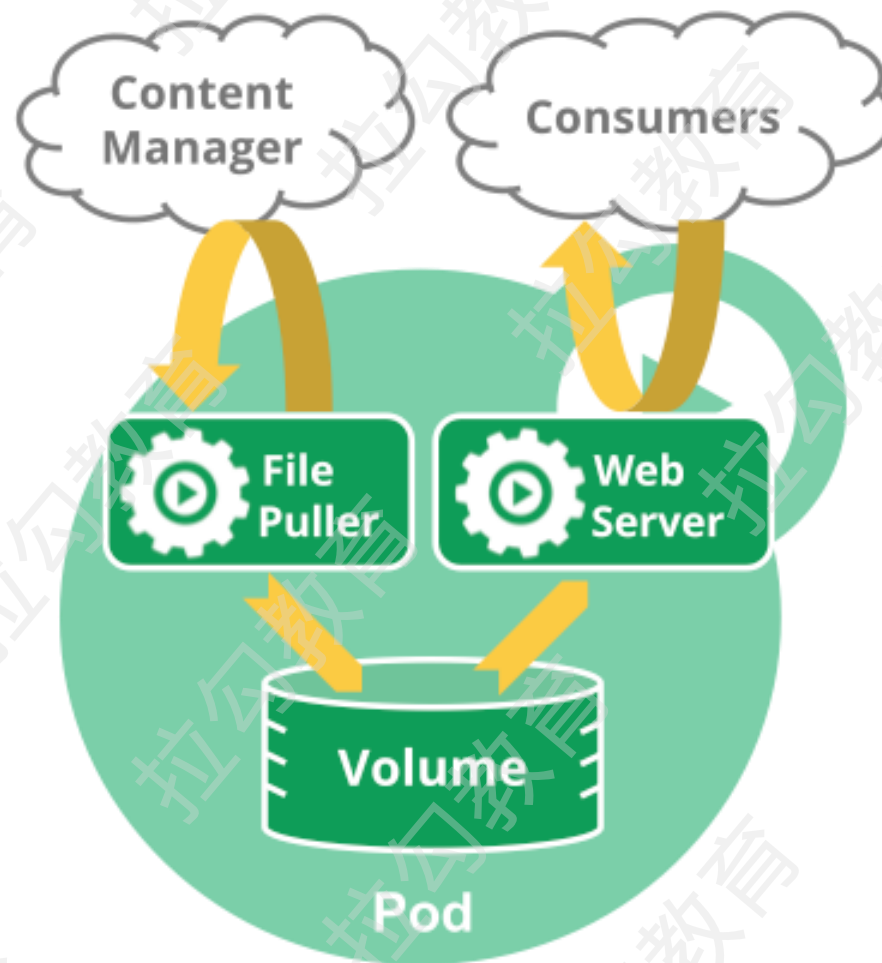


Pod 是什么

拉勾教育

— 互联网人实战大学 —

文档地址见文字版



Pod 是什么

拉勾教育

— 互联网人实战大学 —

容器之间会发生文件交换等

一个写文件，一个读文件

容器之间需要本地通信

比如通过 localhost 或者本地的 Socket

容器之间需要发生频繁的 RPC 调用

出于性能的考量，将它们放在一个 Pod 内

希望为应用添加其他功能

比如日志收集、监控数据采集、配置中心、路由及熔断等功能

为什么 Kubernetes 不直接管理容器，而用 Pod 来管理呢？

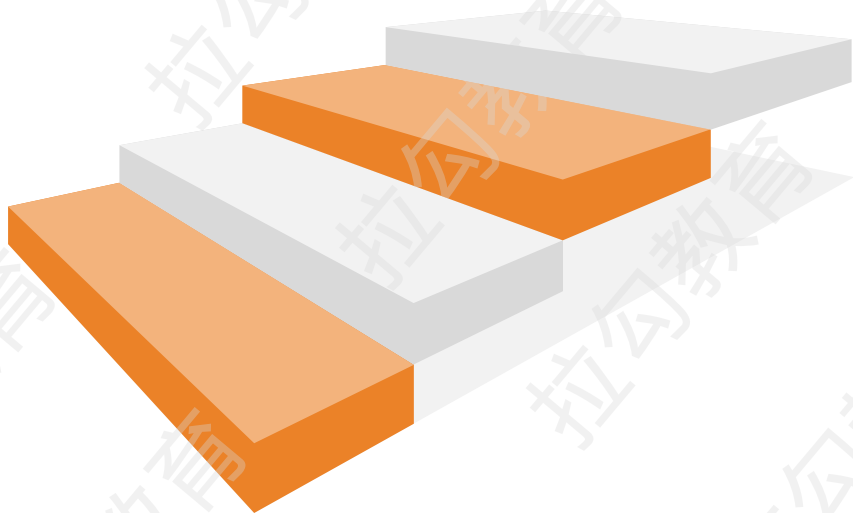
Pod 背后的设计理念

拉勾教育

— 互联网人实战大学 —

直接管理一个容器看起来更简单，为了能够更好地管理容器

Kubernetes 在容器基础上做了更高层次的抽象，即 **Pod**

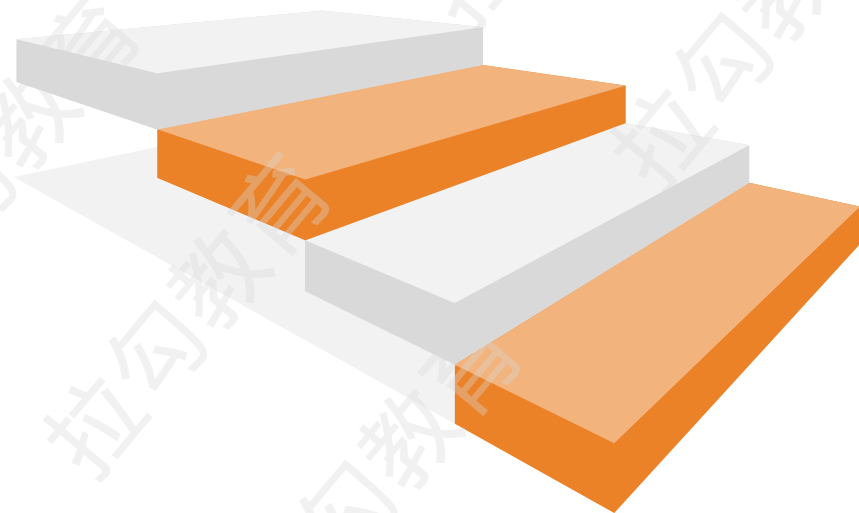


Pod 背后的设计理念

拉勾教育

— 互联网人实战大学 —

- 存活探针（Liveness Probe）可以从应用程序的角度去探测一个进程是否还活着
在容器出现问题之前，就可以快速检测到问题
- 容器启动后和终止前可以进行的操作
比如在容器停止前，可能需要做一些清理工作，或者不能马上结束进程
- 定义了容器终止后要采取的策略，比如始终重启、正常退出才重启等



为什么要允许一个 Pod 内可以包含多个容器？

为什么要允许一个 Pod 内可以包含多个容器？

为什么不直接在单个容器里运行多个程序？

容器实际上是一个“单进程”的模型

如果在容器里启动多个进程

不仅它们的日志记录会混在一起，它们各自的生命周期也无法管理





很多公司在刚开始容器化改造时都会这么去使用容器

把容器当作 VM 来使用，有时也叫作**富容器模式**

这是一种非常不好的尝试，也不符合不可变基础设施的理念

为什么要允许一个 Pod 内可以包含多个容器？

为什么不直接在单个容器里运行多个程序？

用一个 Pod 管理多个容器，既能够保持容器之间的隔离性，还能保证相关容器的环境一致性

如何声明一个 Pod

拉勾教育

— 互联网人实战大学 —

元数据 (metadata)

规范 (spec)

状态 (status)

如何声明一个 Pod

拉勾教育

— 互联网人实战大学 —

元数据 (metadata)



Namespace
(命名空间)



Name
(对象名)



Uid
(对象 ID)

如何声明一个 Pod

拉勾教育

— 互联网人实战大学 —

元数据 (metadata)



Namespace
(命名空间)

是对一组资源和对象的抽象集合

主要用于逻辑上的隔离

如何声明一个 Pod

拉勾教育

— 互联网人实战大学 —

元数据 (metadata)



Namespace
(命名空间)

是对一组资源和对象的抽象集合

主要用于逻辑上的隔离

内置的 namespace:

default

kube-system

kube-public

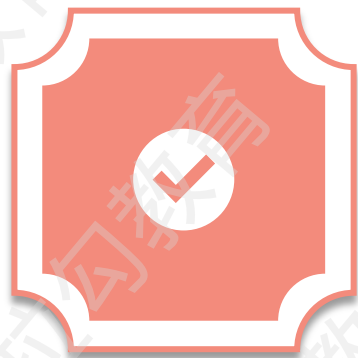
kube-node-lease

如何声明一个 Pod

拉勾教育

— 互联网人实战大学 —

元数据 (metadata)



Name
(对象名)

是用来标识对象的名称，在 namespace 内具有**唯一性**

在不同的 namespace 下，可以创建相同名字的对象

如何声明一个 Pod

拉勾教育

— 互联网人实战大学 —

元数据 (metadata)



Uid
(对象 ID)

由系统自动生成的，主要用于 Kubernetes 内部标识使用

如何声明一个 Pod

拉勾教育

— 互联网人实战大学 —

元数据 (metadata)



Uid
(对象 ID)

还可以在 metadata 里面用各种标签 (labels) 和注释 (annotations)

来标识和匹配不同的对象

用标签 **env=dev** 来标识开发环境

用 **env=testing** 来标识测试环境

如何声明一个 Pod

规范 (spec)

在 Spec 中描述了该对象的详细配置信息，即用户希望的状态 (Desired State)

Kubernetes 中的各大组件会根据这个配置进行一系列的操作，将这种定义从“抽象”变为“现实”

称之为**调和 (Reconcile)**

如何声明一个 Pod

状态 (status)

包含了该对象的一些状态信息，会由各个控制器定期进行更新

也是不同控制器之间进行相互通信的一个渠道

像 Node 的 status 就记录了该节点的一些状态信息

其他的控制器就可以通过 status 知道该 Node 的情况，做一些操作，比如节点宕机修复、可分配资源等

如何声明一个 Pod

拉勾教育

— 互联网人实战大学 —

```
apiVersion: v1 #指定当前描述文件遵循v1版本的Kubernetes API
kind: Pod #我们在描述一个pod
metadata:
  name: twocontainers #指定pod的名称
  namespace: default #指定当前描述的pod所在的命名空间
  labels: #指定pod标签
    app: twocontainers
  annotations: #指定pod注释
    version: v0.5.0
    releasedBy: david
    purpose: demo
spec:
  containers:
    - name: sise #容器的名称
      image: quay.io/openshiftlabs/simple-service:0.5.0 #创建容器所使用的镜像
      ports:
        - containerPort: 9876 #应用监听的端口
```

如何声明一个 Pod

拉勾教育

— 互联网人实战大学 —

```
app: twocontainers
annotations: #指定pod注释
version: v0.5.0
releasedBy: david
purpose: demo
spec:
  containers:
  - name: sise #容器的名称
    image: quay.io/openshiftlabs/simple-service:0.5.0 #创建容器所使用的镜像
    ports:
    - containerPort: 9876 #应用监听的端口
  - name: shell #容器的名称
    image: centos:7 #创建容器所使用的镜像
    command: #容器启动命令
    - "bin/bash"
    - "-c"
    - "sleep 10000"
```


如何声明一个 Pod

拉勾教育

— 互联网人实战大学 —

```
$ kubectl create -f ./twocontainers.yaml
```

```
kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
twocontainers	2/2	Running	0	7s

如何声明一个 Pod

拉勾教育

— 互联网人实战大学 —

通过 exec 进入 shell 这个容器，访问 sise 服务

```
$ kubectl exec twocontainers -c shell -i -t -- bash  
[root@twocontainers /]# curl -s localhost:9876/info  
{"host": "localhost:9876", "version": "0.5.0", "from": "127.0.0.1"}
```

Pod 是 Kubernetes 项目中实现“容器设计模式”的最佳实践之一
也是 Kubernetes 进行复杂应用编排的基础依赖

Next: 《05 | K8s Pod: 最小调度单元的使用进阶及实践》

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