

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 —

09 | 存储类型：如何挑选合适的存储插件？

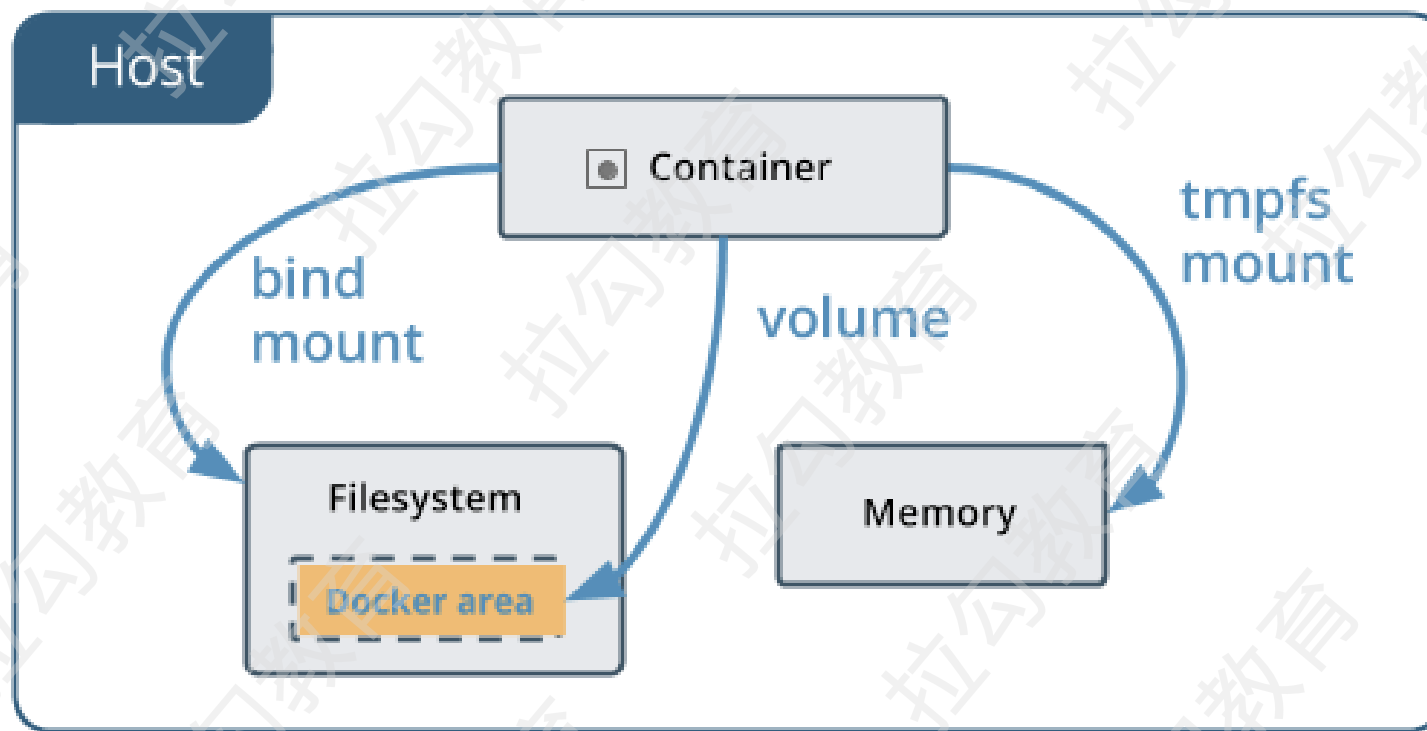
虚拟机时代，大家比较少考虑存储的问题

通过底层 IaaS 平台申请虚拟机的时

大多数情况下都会事先预估好需要的容量

方便虚拟机起来后可以稳定的使用这些存储资源

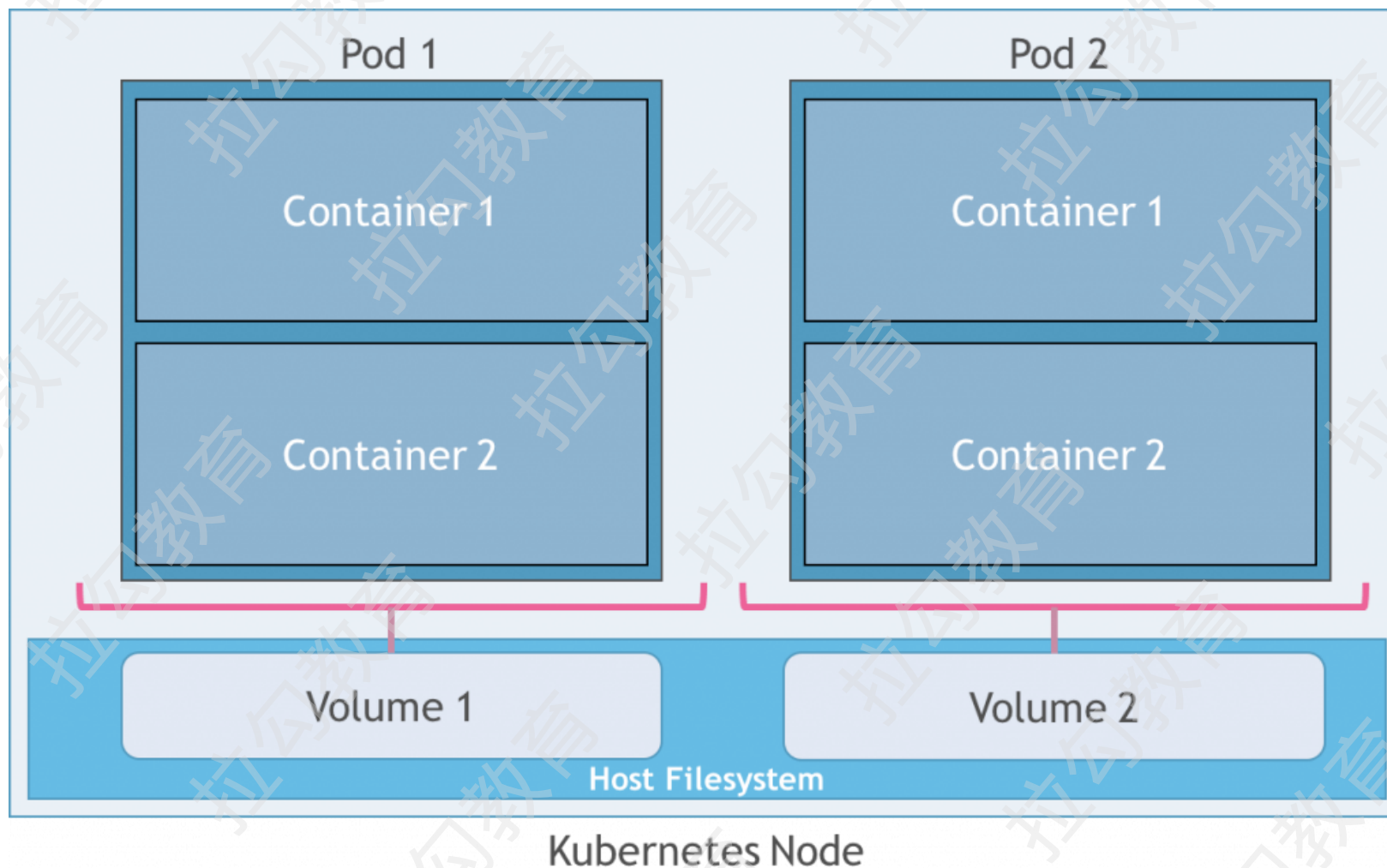




Kubernetes 中的 Volume 是如何设计的?

拉勾教育

— 互联网人实战大学 —



Kubernetes 中的 Volume 是如何设计的?

拉勾教育

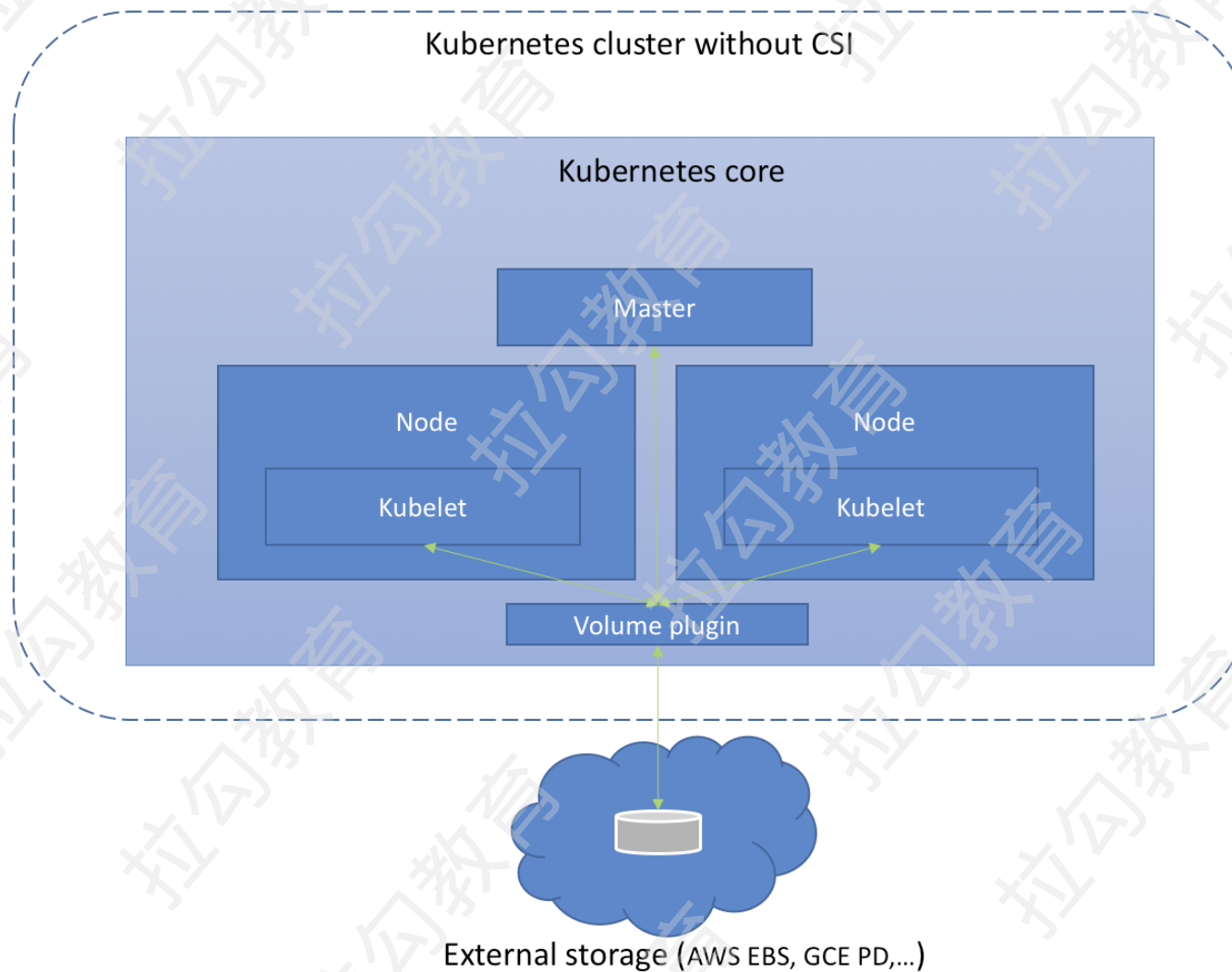
— 互联网人实战大学 —

插件分类	主要用途描述	卷插件	数据是否会随着 Pod 删除而删除
临时存储	主要用于存储一些临时文件，类似于在操作系统中创建的 tempDir	EmptyDir	是
本地存储	用于将一些 Kubernetes 中定义的配置通过 volume 映射到容器中使用	ConfigMap DownwardAPI Secret	是
	使用宿主机上的存储资源	HostPath Local	否
自建存储平台	客户自己搭建的存储平台	CephFS Ginder GlusterFS NFS RBD	否
云厂商插件	一些云厂商提供的插件，供云上的 Kuberentes 使用	awsElasticBlockStore AzureDisk AzureFile GCEPersistentDisk	否

Kubernetes 中的 Volume 是如何设计的?

拉勾教育

— 互联网人实战大学 —



External storage (AWS EBS, GCE PD,...)

常见的几种内置 Volume 插件

拉勾教育

— 互联网人实战大学 —

ConfigMap 和 Secret

都可以通过 Volume 形式挂载到 Pod 内

常见的几种内置 Volume 插件

拉勾教育

— 互联网人实战大学 —

Downward API

```
$ cat downwardapi-volume-demo.yaml
apiVersion: v1
kind: Pod
metadata:
  name: downwardapi-volume-demo
  namespace: demo
  labels:
    zone: us-east-coast
    cluster: downward-api-test-cluster1
    rack: rack-123
  annotations:
    annotation1: "345"
    annotation2: "456"
spec:
  containers:
  - name: volume-test-container
    image: busybox:1.28
    command: ["sh", "-c"]
    args:
    - while true; do
      if [[ -e /etc/podinfo/labels ]]; then
        echo -en '\n\n'; cat /etc/podinfo/labels; fi;
```

常见的几种内置 Volume 插件

拉勾教育

— 互联网人实战大学 —

Downward API

```
command: ["sh", "-c"]
args:
- while true; do
  if [[ -e /etc/podinfo/labels ]]; then
    echo -en '\n\n'; cat /etc/podinfo/labels; fi;
  if [[ -e /etc/podinfo/annotations ]]; then
    echo -en '\n\n'; cat /etc/podinfo/annotations; fi;
  sleep 5;
done;
volumeMounts:
- name: podinfo
  mountPath: /etc/podinfo
volumes:
- name: podinfo
  downwardAPI:
    items:
    - path: "labels"
      fieldRef:
        fieldPath: metadata.labels
    - path: "annotations"
      fieldRef:
        fieldPath: metadata.annotations
```

常见的几种内置 Volume 插件

Downward API

```
$ kubectl create -f downwardapi-volume-demo.yaml
pod/downwardapi-volume-demo created
$ kubectl get pod -n demo
NAME                READY  STATUS   RESTARTS  AGE
downwardapi-volume-demo 1/1    Running  0         5s
$ kubectl logs -n demo -f downwardapi-volume-demo
```

```
cluster="downward-api-test-cluster1"
rack="rack-123"
zone="us-east-coast"
```

```
annotation1="345"
annotation2="456"
kubernetes.io/config.seen="2020-09-03T12:01:58.1728583Z"
kubernetes.io/config.source="api"
```

```
cluster="downward-api-test-cluster1"
```

常见的几种内置 Volume 插件

拉勾教育

— 互联网人实战大学 —

Downward API

```
cluster="downward-api-test-cluster1"  
rack="rack-123"  
zone="us-east-coast"
```

```
annotation1="345"  
annotation2="456"  
kubernetes.io/config.seen="2020-09-03T12:01:58.1728583Z"  
kubernetes.io/config.source="api"
```

```
cluster="downward-api-test-cluster1"  
rack="rack-123"  
zone="us-east-coast"
```

```
annotation1="345"  
annotation2="456"  
kubernetes.io/config.seen="2020-09-03T12:01:58.1728583Z"  
kubernetes.io/config.source="api"
```

常见的几种内置 Volume 插件

EmptyDir

拉勾教育

— 互联网人实战大学 —

```
apiVersion: v1
kind: Pod
metadata:
  name: empty-dir-vol-demo
  namespace: demo
spec:
  containers:
    - image: busybox:1.28
      name: volume-test-container
      volumeMounts:
        - mountPath: /cache
          name: cache-volume
  volumes:
    - name: cache-volume
      emptyDir: {}
```

常见的几种内置 Volume 插件

拉勾教育

— 互联网人实战大学 —

EmptyDir

```
apiVersion: v1
kind: Pod
metadata:
  name: empty-dir-vol-memory-demo
  namespace: demo
spec:
  containers:
  - image: busybox:1.28
    imagePullPolicy: IfNotPresent
    name: myvolumes-container
    command: ['sh', '-c', 'echo container is Running ; df -h ; sleep 3600']
    volumeMounts:
    - mountPath: /demo
      name: demo-volume
  volumes:
  - name: demo-volume
    emptyDir:
      medium: Memory
```

HostPath

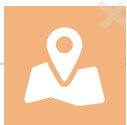
和 EmptyDir 一样，都是利用宿主机的存储为容器分配资源

两者的区别就是 HostPath 中的数据并不会随着 Pod 被删除而删除，而是会持久地存放在该节点上

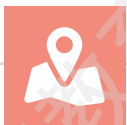
常见的几种内置 Volume 插件

拉勾教育

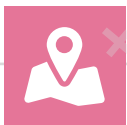
— 互联网人实战大学 —



避免通过容器恶意修改宿主机上的文件内容



避免容器恶意占用宿主机上的存储资源而打爆宿主机



要考虑到 Pod 自身的声明周期，而且 Pod 是会“漂移”重新“长”到别的节点上的
所以要避免过度依赖本地的存储

常见的几种内置 Volume 插件

拉勾教育

— 互联网人实战大学 —

```
apiVersion: v1
kind: Pod
metadata:
  name: hostpath-demo
  namespace: demo
spec:
  containers:
  - image: nginx:1.19.2
    name: container-demo
    volumeMounts:
    - mountPath: /test-pd
      name: hostpath-volume
  volumes:
  - name: hostpath-volume
    hostPath:
      path: /data #对应宿主机上的绝对路径
      type: Directory #可选字段，默认是 Directory
```

为什么社区要采用 CSI

拉勾教育

— 互联网人实战大学 —

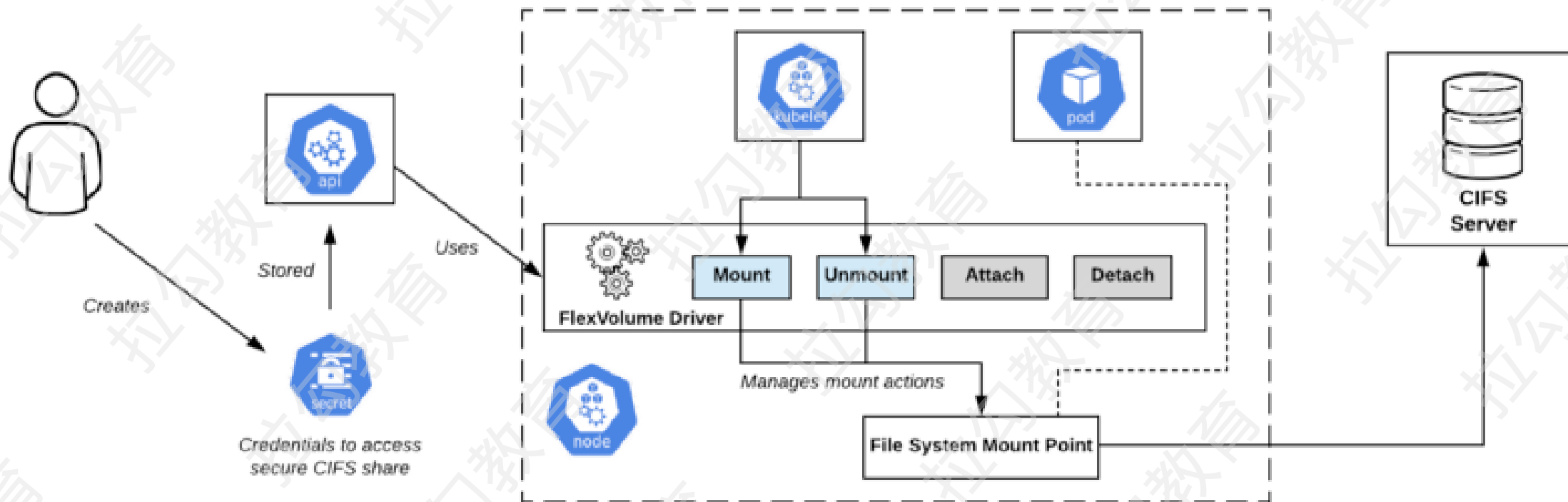


1. 这些插件对 Kubernetes 代码本身的稳定性以及安全性引入了很多未知的风险
一个很小的 Bug 都有可能导致集群受到攻击或者无法工作
2. 这些插件的维护和 Kubernetes 的正常迭代紧密耦合在一起，一起打包和编译
即便是某个单一插件出现了 Bug，都需要通过升级 Kubernetes 的版本来修复
3. 社区需要维护所有的 volume plugin，并且要经过完整的测试验证流程
来保证可用性，这给社区的正常迭代平添了很多麻烦
4. 各个卷插件依赖的包也都要算作 Kubernetes 项目的一部分
这会让 Kubernetes 的依赖变得臃肿
5. 开发者被迫要将这些插件代码进行开源

为什么社区要采用 CSI

拉勾教育

— 互联网人实战大学 —



为什么社区要采用 CSI

拉勾教育

— 互联网人实战大学 —

官方清单列表

<https://kubernetes-csi.github.io/docs/drivers.html>

官方开发文档

<https://kubernetes-csi.github.io/docs/introduction.html>

Next: 《10 | 存储管理：怎样对业务数据进行持久化存储？》

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