

智能时代的新运维

**CNUTCon**

全球运维技术大会

# 基于 CSI 的 Kubernetes 存储插件开发实践

王欣

青云QingCloud

主办方 **Geekbang** **InfoQ**  
极客邦科技



# TABLE OF CONTENTS 大纲

---

- 背景
- 开发 CSI 插件
- 部署 CSI 插件
- 质量管理
- 展望





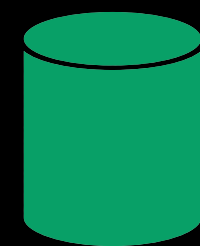
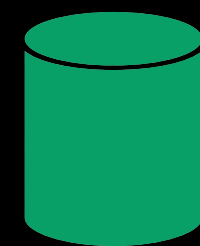
# 背景



Kubernetes

存储插件

存储服务 GCE PD AWS EBS



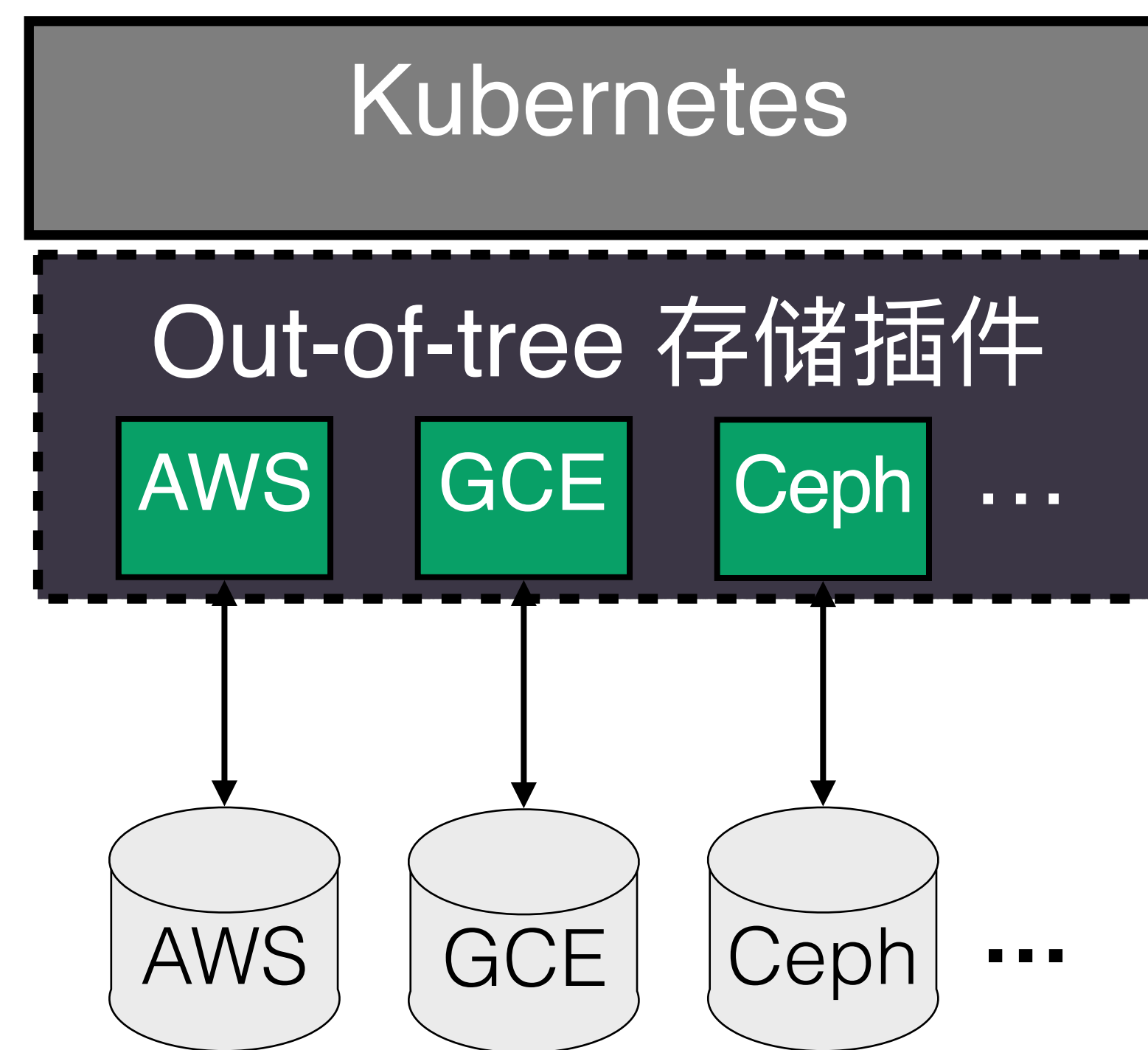
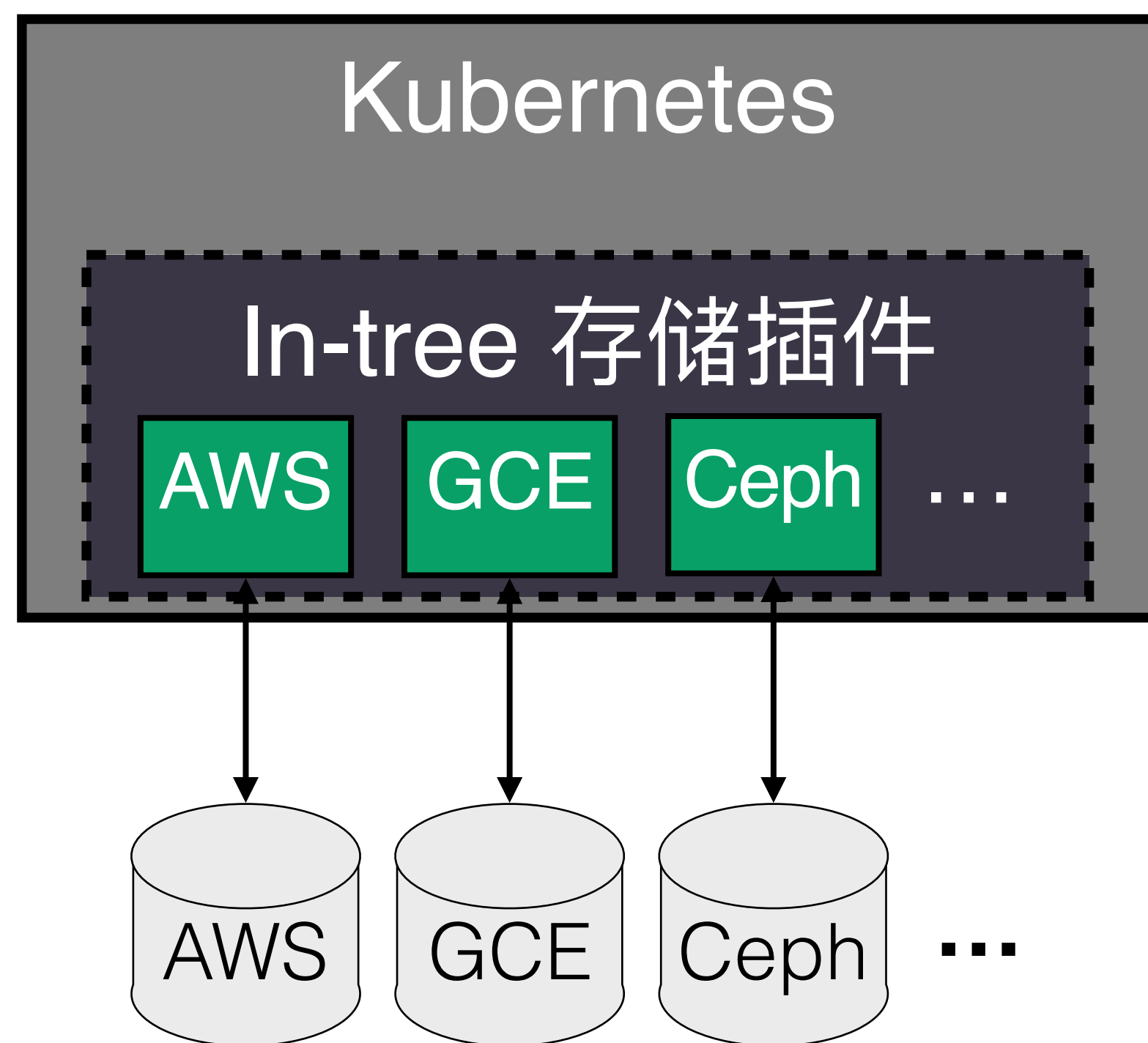
...

现有的 Kubernetes 存储插件已经  
**无法满足** 用户日益增长的需求

急需将 **存储服务** 与 **Kubernetes** 对接。



# Kubernetes 存储插件分类



# In-tree vs Out-of-tree

	In-tree 存储插件	Out-of-tree 存储插件
功能性	存储卷创建，删除，挂载	存储卷创建、删除和挂载，快照创建和删除等功能
支持的存储类型	有限，限制用户的选择	多样，开发插件即可对接
易维护性	插件代码在 K8S 代码仓库内，不易维护	插件代码独立于 Kubernetes，可独立构建、发布，易于维护

❖ 结论：开发和使用 Out-of-tree 存储插件

# Flex-volume vs CSI

- ❖ Out-of-tree 存储插件分为 Flex-volume 和 CSI 两类
- ❖ **Flex-volume**: Kubernetes 1.2 开始支持
  - ❖ 部署**复杂**，功能**有限**
- ❖ **CSI (Container Storage Interface)**: Kubernetes 1.9 开始支持
  - ❖ **容器平台**领域工业标准
  - ❖ 部署**简便**，支持**容器化部署**
  - ❖ 功能**强大**，支持存储卷管理，快照管理等功能，在持续快速发展中
- ❖ 结论：开发和使用 CSI 存储插件



目标：开发 CSI 插件，并通过 CSI 插件将存储服务与 Kubernetes 对接





# TABLE OF CONTENTS 大纲

---

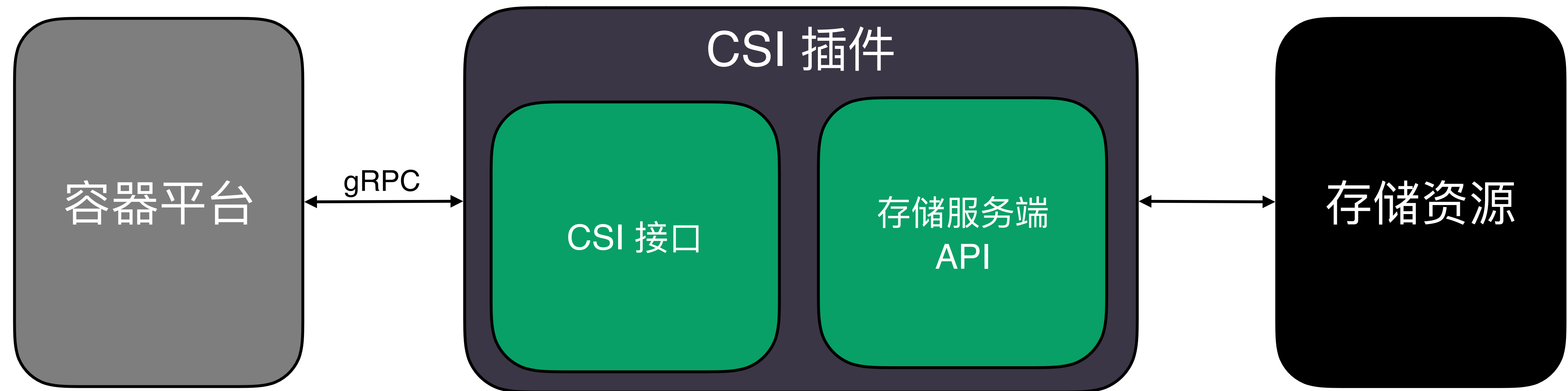
- 背景
- 开发 CSI 插件
- 部署 CSI 插件
- 质量管理
- 展望





# CSI 插件基本原理

- ❖ CSI 插件起到**承上启下**作用，上层对接容器平台，下层操作存储服务端
- ❖ “承上”：实现 CSI 接口 ——> 对接容器平台，为容器平台赋能
- ❖ “启下”：调用存储服务端 API ——> 操作存储资源





# 开发 CSI 插件

CSI 官方规范



<https://github.com/container-storage-interface/spec/blob/master/spec.md>

- ❖ 实现 CSI 接口的功能
- ❖ 实现 CSI 接口的幂等性
- ❖ 符合 CSI 返回值规范





# CSI 接口概述

- ❖ 分为 Identity, Controller, Node 三类接口
- ❖ 分为 Controller, Node 两类服务

	Identity	Controller	Node
功能	描述插件基本信息 检测插件健康状态	向存储服务端发送指令 对存储卷进行管理	对主机上的存储卷进行操作
接口	GetPluginInfo, Probe, GetPluginCapabilities	CreateVolume, DeleteVolume, ControllerPublishVolume, ControllerUnpublishVolume, ControllerGetCapabilities ...	NodePublishVolume, NodeUnpublishVolume, NodeStageVolume. NodeUnstageVolume, NodeGetId, NodeGetInfo, NodeGetCapabilities ...





# 存储卷的生命周期

**Controller 服务**

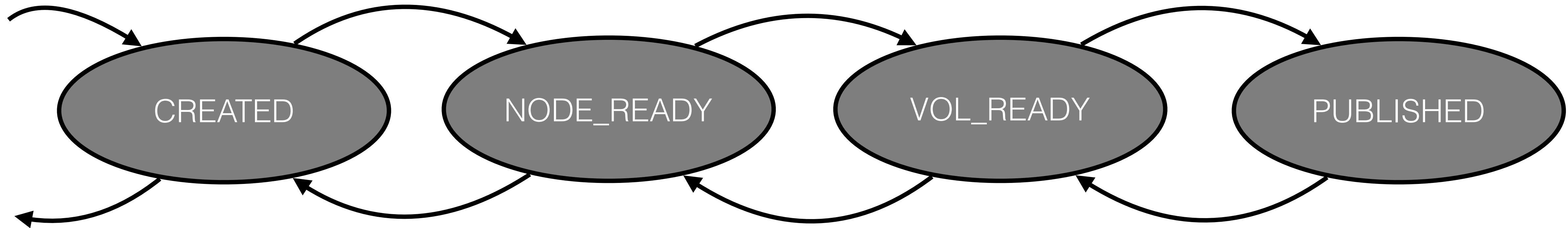
**Node 服务**

**CreateVolume**

**ControllerPublishVolume**

**NodeStageVolume**

**NodePublishVolume**



**DeleteVolume**

**ControllerUnpublishVolume**

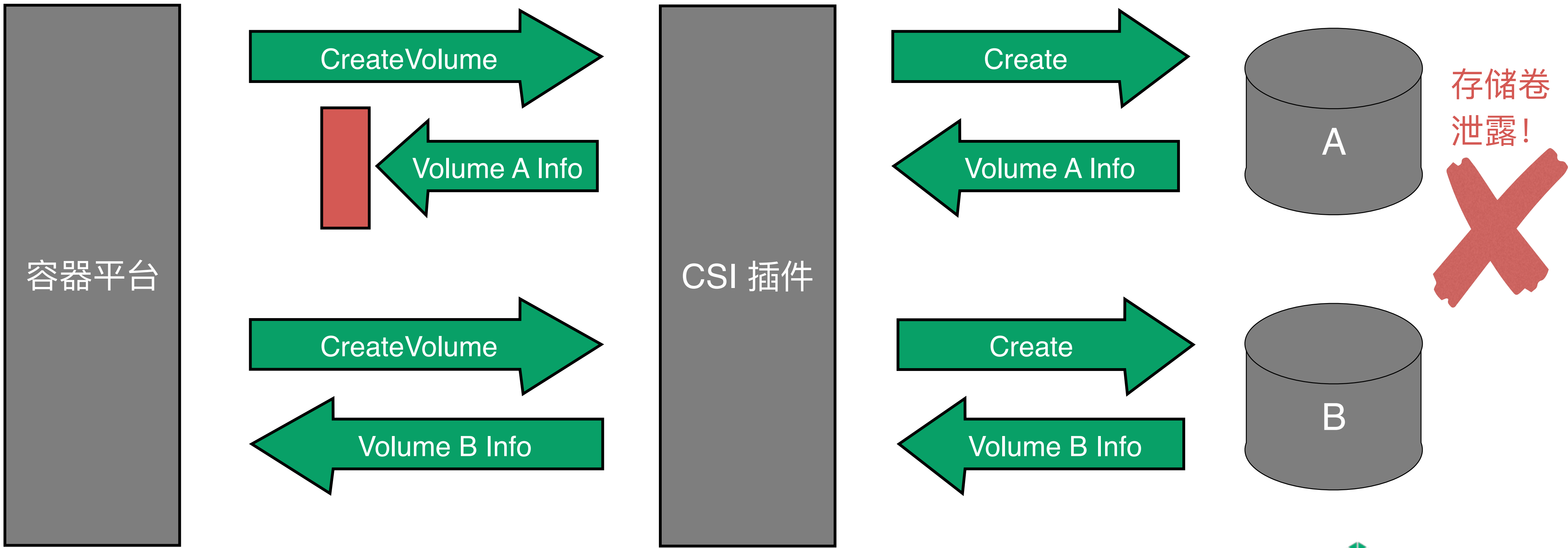
**NodeUnstageVolume**

**NodeUnpublishVolume**



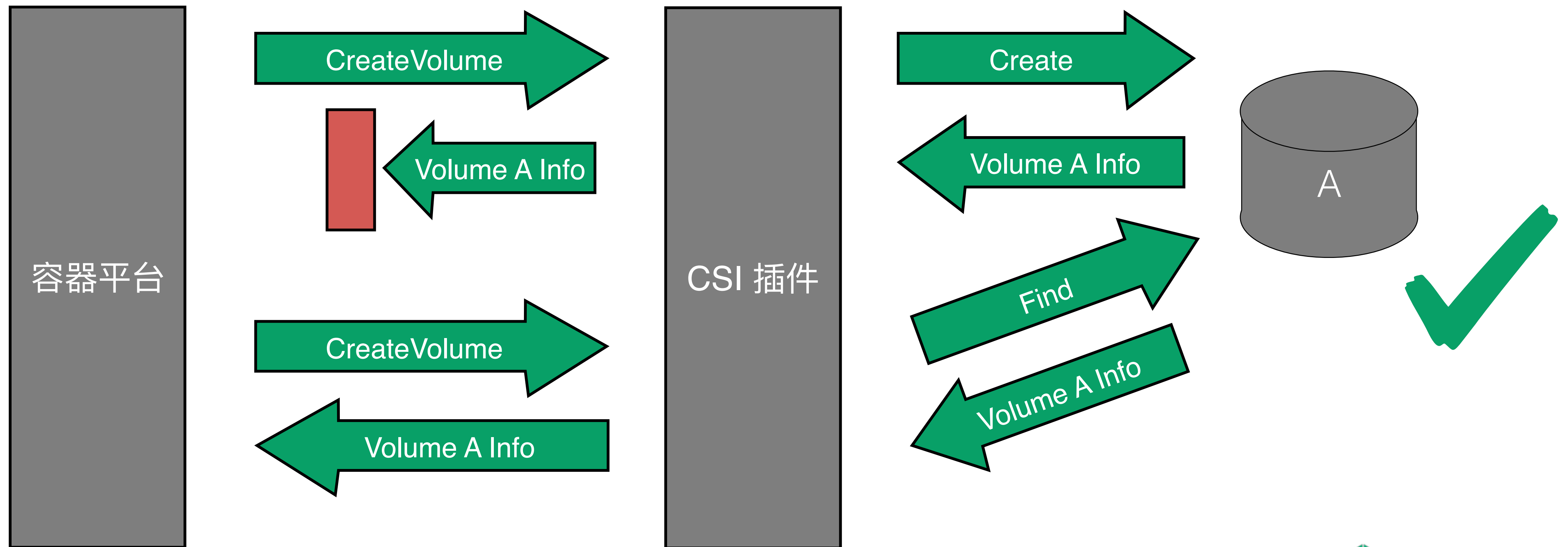


# 存储卷泄露



# 幂等性

❖ 幂等性：使用相同参数对某个接口调用多次与调用一次的结果相同。





# 规范返回值

❖ CSI 的 CreateVolume 接口错误码规范

Condition	gRPC Code
Missing required field	3 INVALID_ARGUMENT
Invalid or unsupported field in the request	3 INVALID_ARGUMENT
Operation pending for volume	10 ABORTED
Volume already exists but is incompatible	6 ALREADY_EXISTS
Unable to provision in accessible_topology	8 RESOURCE_EXHAUSTED
Unsupported capacity_range	11 OUT_OF_RANGE
Call not implemented	12 UNIMPLEMENTED





# TABLE OF CONTENTS 大纲

---

- 背景
- 开发 CSI 插件
- 部署 CSI 插件
- 质量管理
- 展望





# Kubernetes 部署 CSI 插件

## Kubernetes CSI 推荐部署方式

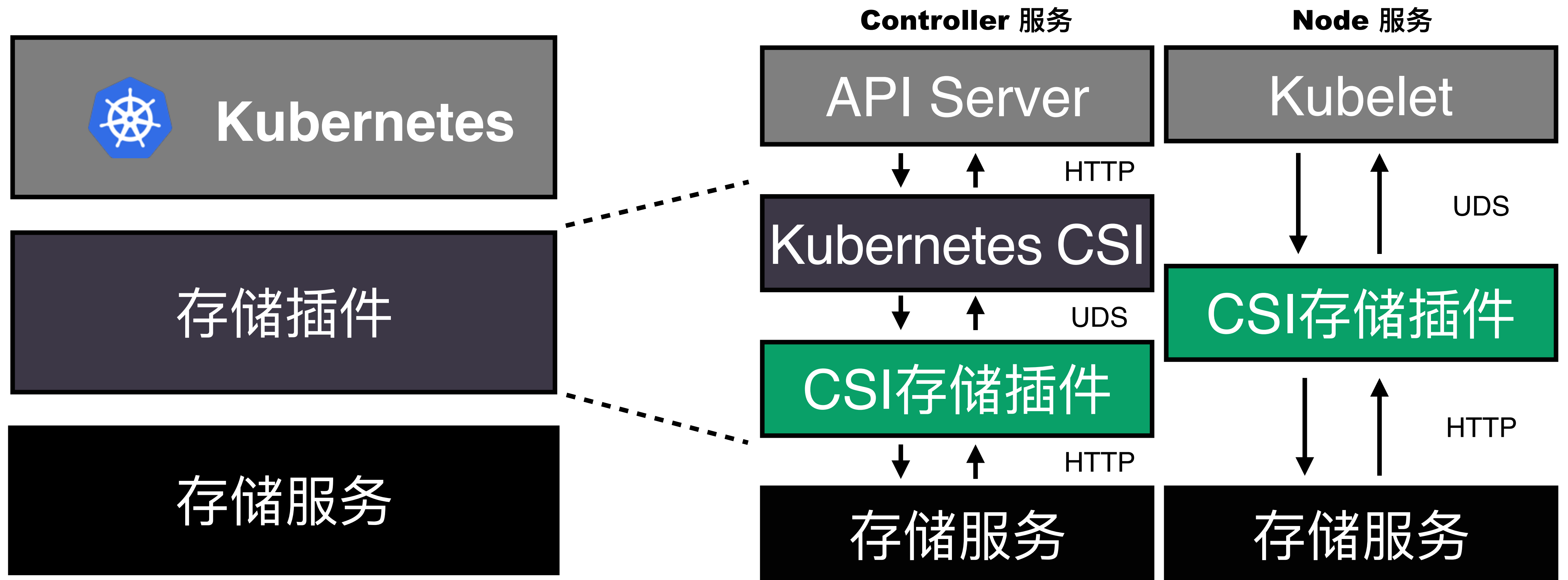


<https://github.com/kubernetes/community/blob/master/contributors/design-proposals/storage/container-storage-interface.md>

- ❖ 通信：使用 UDS (Unix Domain Socket) 和 HTTP
- ❖ 部署：使用 Kubernetes YAML 文件容器化部署
- ❖ 权限控制：使用 RBAC 限制 CSI 插件权限

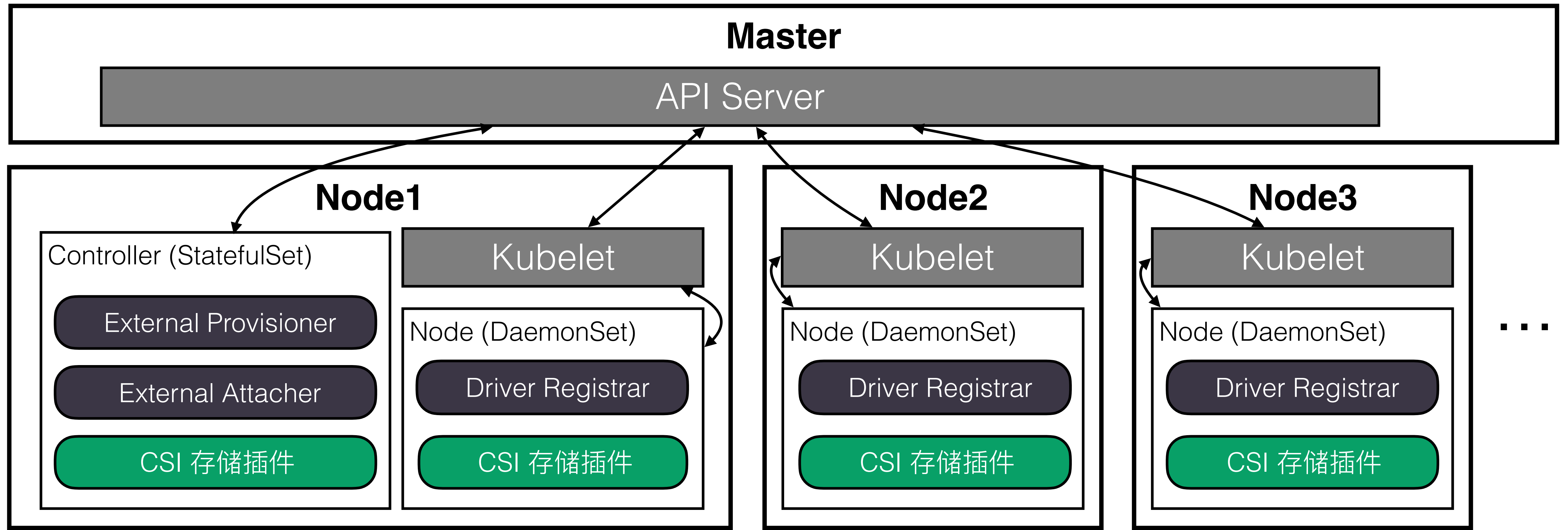


# 通信





# 部署架构



Legend: K8S 核心组件 K8S CSI 对接组件 - Kubernetes 团队 CSI 插件 - 存储厂商



# 部署 YAML 文件

- ❖ 声明式的 Kubernetes YAML 定义文件，便于用户在不同集群**快速和标准化**地部署插件。

 [config.yaml](#)

 [csi-controller-rbac.yaml](#)

 [csi-controller-sts.yaml](#)

 [csi-node-ds.yaml](#)

 [csi-node-rbac.yaml](#)

 [csi-secret.yaml](#)





# RBAC 权限控制

- ❖ 存储插件仅需读写 Kubernetes 集群的部分对象，如：PV，Node 等对象。
- ❖ 通过 RBAC，限制存储插件的权限，维护集群安全。

```
kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: csi-qingcloud-attacher
  labels:
    qingcloud.com/csi-controller: "true"
    csi: "v0.2.0"
    owner: "yunify"
rules:
- apiGroups: [""]
  resources: ["events"]
  verbs: ["get", "list", "watch", "update"]
- apiGroups: [""]
  resources: ["persistentvolumes"]
  verbs: ["get", "list", "watch", "update"]
- apiGroups: [""]
  resources: ["nodes"]
  verbs: ["get", "list", "watch"]
- apiGroups: ["storage.k8s.io"]
  resources: ["volumeattachments"]
  verbs: ["get", "list", "watch", "update"]
```





# TABLE OF CONTENTS 大纲

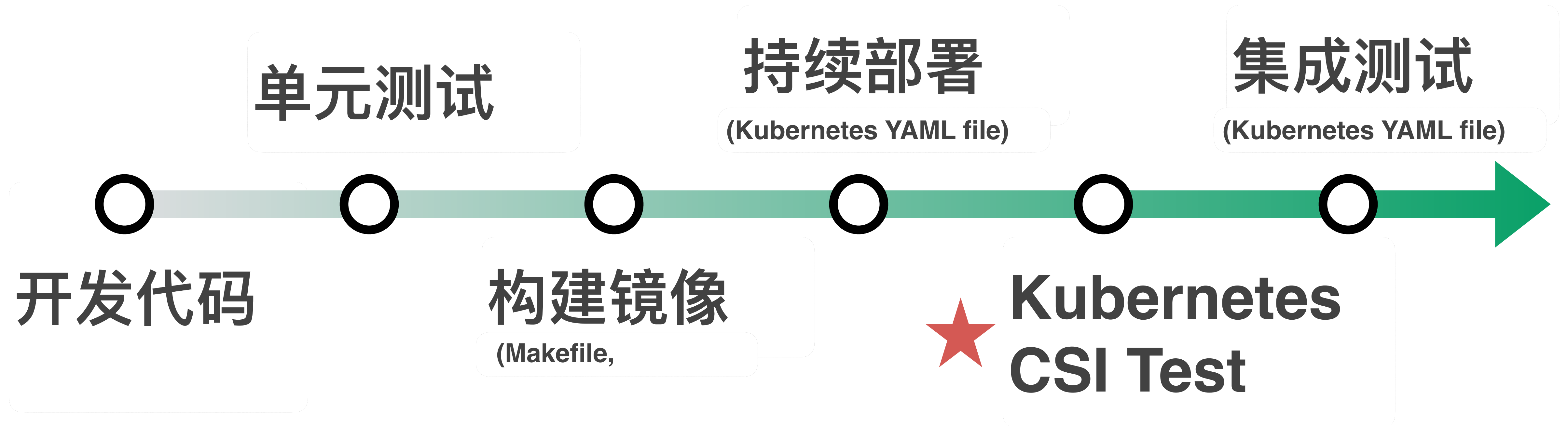
---

- 背景
- 开发 CSI 插件
- 部署 CSI 插件
- 质量管理
- 展望



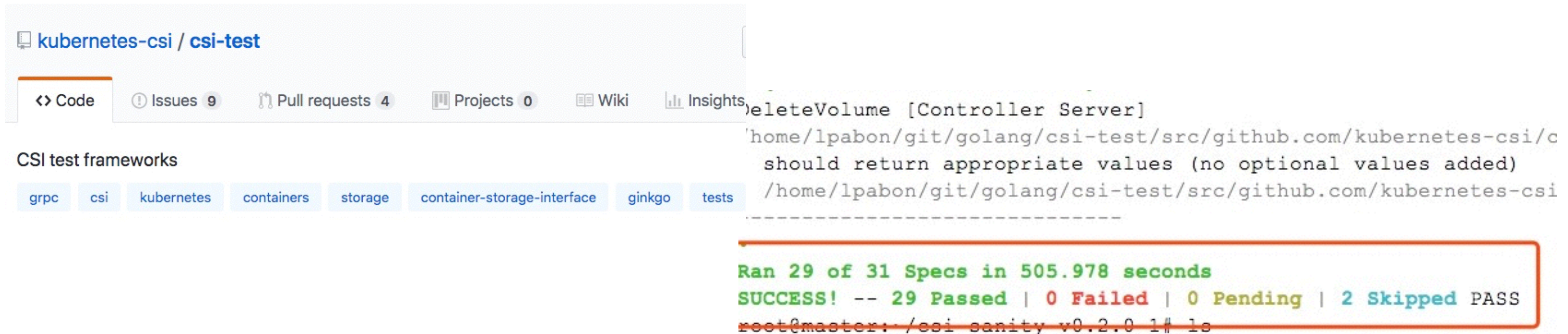


# 持续集成



# Kubernetes CSI Test

- ❖ CSI Test 是 Kubernetes CSI 官方开发的开源项目，旨在测试 CSI 存储插件是否符合 CSI 官方规范。



The image shows a screenshot of the Kubernetes CSI Test GitHub repository page. The repository is named 'kubernetes-csi / csi-test'. It has 9 issues, 4 pull requests, 0 projects, a Wiki, and Insights. Below the repository name, there are tags for 'CSI test frameworks' including 'grpc', 'csi', 'kubernetes', 'containers', 'storage', 'container-storage-interface', 'ginkgo', and 'tests'. To the right, a terminal window shows the output of a test run. The output indicates that 29 of 31 specs passed in 505.978 seconds, with 0 failed, 0 pending, and 2 skipped. The test results are highlighted with a red box.

```
deleteVolume [Controller Server]
/home/lpabon/git/golang/csi-test/src/github.com/kubernetes-csi/c
should return appropriate values (no optional values added)
/home/lpabon/git/golang/csi-test/src/github.com/kubernetes-csi
-----
Ran 29 of 31 Specs in 505.978 seconds
SUCCESS! -- 29 Passed | 0 Failed | 0 Pending | 2 Skipped PASS
root@master:~/csi-sanity v0.2.0 1# ls
```





# TABLE OF CONTENTS 大纲

---

- 背景
- 开发 CSI 插件
- 部署 CSI 插件
- 质量管理
- 展望





# 展望

- ❖ 快照功能完善
- ❖ 存储卷监控
- ❖ 存储卷扩容
- ❖ In-tree 存储插件向 CSI 迁移





## 相关资源

CSI 定义规范: <https://github.com/container-storage-interface/spec/blob/master/spec.md>

青云云平台 CSI 存储插件: <https://github.com/yunify/qingcloud-csi>

NeonSAN CSI 存储插件: <https://github.com/yunify/qingstor-csi>

## Q&A





智能时代的新运维

**CNUTCon**  
全球运维技术大会

# THANKS

主办方 **Geekbang** **InfoQ**  
极客邦科技





AiCon

2018.12.20-23 / 北京·国际会议中心

# AI商业化下的技术演进实战干货分享

京东：智能金融

景驰科技：自动驾驶

阿里巴巴：NLP

清华人工智能研究院：机器学习

今日头条：机器学习

Twitter：搜索推荐

AWS：计算机视觉

Netflix：机器学习



扫码了解详情



在不同业务场景下架构师如何升级？

# 听100+位讲师的优质 架构实践分享

来自Google、Netflix、LinkedIn、BAT等...

2018年12月7-8日 北京·国际会议中心



扫码了解大会详情



# QCon

## 全球软件开发大会

### 北京·2019

更多技术干货分享，北京站精彩继续  
提前参与，还能享受更多优惠

识别二维码  
查看了解更多

[2019.qconbeijing.com](http://2019.qconbeijing.com)





# 深入剖析 Kubernetes

Kubernetes 原来可以如此简单



扫码了解更多

张磊

Kubernetes 社区

资深成员与项目维护者

