



CloudNativeLives

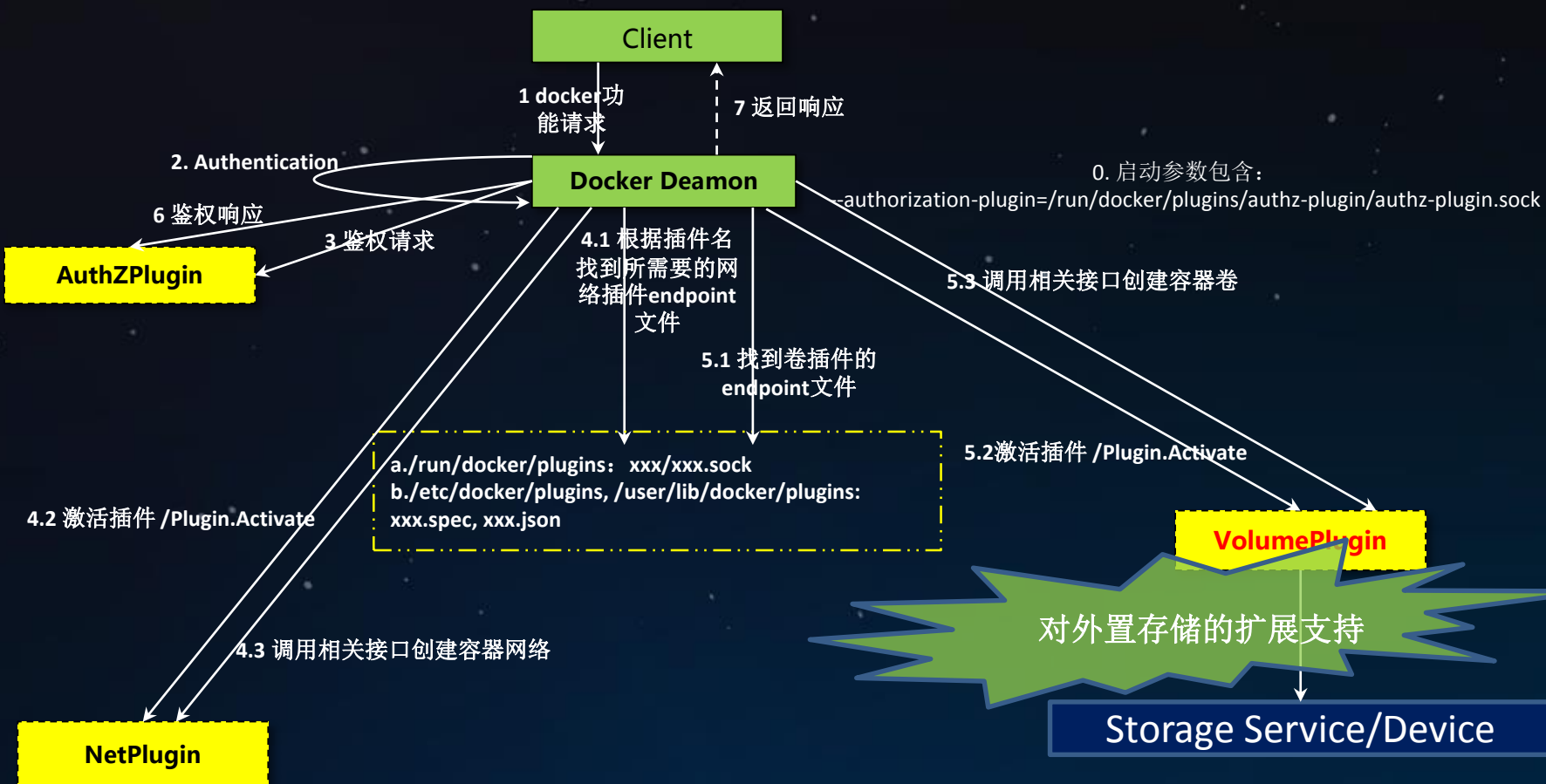
K8S存储原理剖析与实践

华为云容器团队核心架构师 & CNCF社区主要贡献者倾心打造

大纲

- **K8S存储整体框架及原理解析**
- K8S FlexVolume存储扩展机制
- K8S CSI存储扩展机制

Docker插件机制-架构&评价



【评价】

优点：

1) 不重新编译docker的情况下可以提供针对鉴权、网络、卷等功能的扩展；

2) 基于Http JSON-PRC格式的接口与插件进行交互；

3) 支持容器化和非容器化多种部署形式；

4) 支持插件生命周期管理：1.52+ docker plugin命令及其API；<1.52 docker volume/network；

5) 支持基于TLS的安全加固。

约束：

某些插件（认证）的添加需要重启docker daemon

Docker Volume Plugin列表



名称	描述	地址
Azure File Storage plugin	Lets you mount Microsoft Azure File Storage shares to Docker containers as volumes using the SMB 3.0 protocol. Learn more.	https://github.com/Azure/azurefile-dockervolumedriver
BeeGFS Volume Plugin	An open source volume plugin to create persistent volumes in a BeeGFS parallel file system.	https://github.com/RedCoolBeans/docker-volume-beegfs
Blockbridge plugin	A volume plugin that provides access to an extensible set of container-based persistent storage options. It supports single and multi-host Docker environments with features that include tenant isolation, automated provisioning, encryption, secure deletion, snapshots and QoS.	https://github.com/blockbridge/blockbridge-docker-volume
Contiv Volume Plugin	An open source volume plugin that provides multi-tenant, persistent, distributed storage with intent based consumption. It has support for Ceph and NFS.	https://github.com/rancher/convoy
DigitalOcean Block Storage plugin	Integrates DigitalOcean's block storage solution into the Docker ecosystem by automatically attaching a given block storage volume to a DigitalOcean droplet and making the contents of the volume available to Docker containers running on that droplet.	https://github.com/omallo/docker-volume-plugin-dostorage
DRBD plugin	A volume plugin that provides highly available storage replicated by DRBD. Data written to the docker volume is replicated in a cluster of DRBD nodes.	https://www.drbd.org/en/supported-projects/docker
Flocker plugin	A volume plugin that provides multi-host portable volumes for Docker, enabling you to run databases and other stateful containers and move them around across a cluster of machines.	https://clusterhq.com/docker-plugin/
Fuxi Volume Plugin	A volume plugin that is developed as part of the OpenStack Kuryr project and implements the Docker volume plugin API by utilizing Cinder, the OpenStack block storage service.	https://github.com/openstack/fuxi
gce-docker plugin	A volume plugin able to attach, format and mount Google Compute persistent-disks.	https://github.com/mcuadros/gce-docker
GlusterFS plugin	A volume plugin that provides multi-host volumes management for Docker using GlusterFS.	https://github.com/calavera/docker-volume-glusterfs

Docker Volume Plugin列表



名称	描述	地址
Horcrux Volume Plugin	A volume plugin that allows on-demand, version controlled access to your data. Horcrux is an open-source plugin, written in Go, and supports SCP, Minio and Amazon S3.	https://github.com/muthu-r/horcrux
HPE 3Par Volume Plugin	A volume plugin that supports HPE 3Par and StoreVirtual iSCSI storage arrays.	https://github.com/hpe-storage/python-hpedockerplugin/
Infiniit volume plugin	A volume plugin that makes it easy to mount and manage Infiniit volumes using Docker.	https://infiniit.sh/documentation/docker/volume-plugin
IPFS Volume Plugin	An open source volume plugin that allows using an ipfs filesystem as a volume.	http://github.com/vdemeester/docker-volume-ipfs
Keywhiz plugin	A plugin that provides credentials and secret management using Keywhiz as a central repository.	https://github.com/calavera/docker-volume-keywhiz
Local Persist Plugin	A volume plugin that extends the default local driver's functionality by allowing you specify a mountpoint anywhere on the host, which enables the files to always persist, even if the volume is removed via docker volume rm.	https://github.com/CWSpear/local-persist
NetApp Plugin(nDVP)	A volume plugin that provides direct integration with the Docker ecosystem for the NetApp storage portfolio. The nDVP package supports the provisioning and management of storage resources from the storage platform to Docker hosts, with a robust framework for adding additional platforms in the future.	https://github.com/NetApp/netappdvp
Netshare plugin	A volume plugin that provides volume management for NFS 3/4, AWS EFS and CIFS file systems.	https://github.com/ContainX/docker-volume-netshare
Nimble Storage Volume Plugin	A volume plug-in that integrates with Nimble Storage Unified Flash Fabric arrays. The plug-in abstracts array volume capabilities to the Docker administrator to allow self-provisioning of secure multi-tenant volumes and clones.	https://connect.nimblestorage.com/community/app-integration/docker

Docker Volume Plugin列表



名称	描述	地址
OpenStorage Plugin	A cluster-aware volume plugin that provides volume management for file and block storage solutions. It implements a vendor neutral specification for implementing extensions such as CoS, encryption, and snapshots. It has example drivers based on FUSE, NFS, NBD and EBS to name a few.	https://github.com/libopenstorage/openstorage
Portworx Volume Plugin	A volume plugin that turns any server into a scale-out converged compute/storage node, providing container granular storage and highly available volumes across any node, using a shared-nothing storage backend that works with any docker scheduler.	https://github.com/portworx/px-dev
Quobyte Volume Plugin	A volume plugin that connects Docker to Quobyte's data center file system, a general-purpose scalable and fault-tolerant storage platform.	https://github.com/quobyte/docker-volume
REX-Ray plugin	A volume plugin which is written in Go and provides advanced storage functionality for many platforms including VirtualBox, EC2, Google Compute Engine, OpenStack, and EMC.	https://github.com/emccode/rexray
Virtuozzo Storage and Ploop plugin	A volume plugin with support for Virtuozzo Storage distributed cloud file system as well as ploop devices.	https://github.com/virtuozzo/docker-volume-ploop
VMware vSphere Storage Plugin	Docker Volume Driver for vSphere enables customers to address persistent storage requirements for Docker containers in vSphere environments.	https://github.com/vmware/docker-volume-vsphere

支持Kubernetes的平台和存储服务



容器服务/平台



EKS

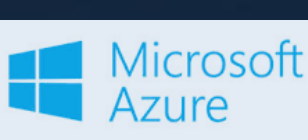


GKS



AKS

基础设施

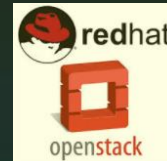
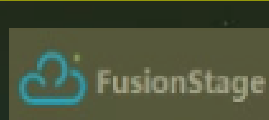


CCE



CCI

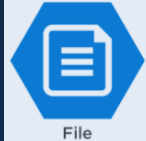
Huawei Cloud



存储服务/设备



Persistent Disk



File



Disk



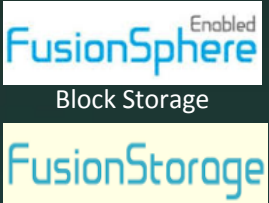
EVS



SFS



OBS



NAS



公有云

私有云

华为：云厂中的跨界王者

K8S 存储能力-Volume概述



- K8S中的普通Volume提供了在容器中挂卷的能力，它不是独立的K8S资源对象，不能通过k8s去管理（创建、删除等），只能在创建Pod时去引用。
- Pod需要设置卷来源（ `spec.volume` ）和挂载点（ `spec.containers.volumeMounts` ）两个信息后才可以使使用相应的Volume

使用AWS EBS的容器应用

```
apiVersion: v1
kind: Pod
metadata:
  name: test-efs
spec:
  containers:
  - image: k8s.gcr.io/test-webserver
    name: test-container
    volumeMounts:
    - mountPath: /test-efs
      name: test-volume
  volumes:
  - name: test-volume
    # This AWS EBS volume must already
    exist.
    awsElasticBlockStore:
      volumeID: <volume-id>
      fsType: ext4
```

使用GCE PD的容器应用

```
apiVersion: v1
kind: Pod
metadata:
  name: test-pd
spec:
  containers:
  - image: k8s.gcr.io/test-webserver
    name: test-container
    volumeMounts:
    - mountPath: /test-pd
      name: test-volume
  volumes:
  - name: test-volume
    # This GCE PD must already exist.
    gcePersistentDisk:
      pdName: my-data-disk
      fsType: ext4
```

使用HostPath的容器应用

```
apiVersion: v1
kind: Pod
metadata:
  name: test-pd
spec:
  containers:
  - image: k8s.gcr.io/test-webserver
    name: test-container
    volumeMounts:
    - mountPath: /test-pd
      name: test-volume
  volumes:
  - name: test-volume
    hostPath:
      # directory location on host
      path: /data
      # this field is optional
      type: Directory
```

使用ConfigMap的容器应用

```
apiVersion: v1
kind: Pod
metadata:
  name: configmap-pod
spec:
  containers:
  - name: test
    image: busybox
    volumeMounts:
    - name: config-vol
      mountPath: /etc/config
  volumes:
  - name: config-vol
    configMap:
      name: log-config
      items:
      - key: log_level
        path: log_level
```


K8S 存储能力 : In-Tree Volume Plugins



K8S的VolumePlugin提供了插件化扩展存储的机制，分为内置插件（In-Tree Plugins）和外置插件（Out-of-Tree） 两种

名称	描述
awsElasticBlockStore	mounts an Amazon Web Services (AWS) EBS Volume (Elastic Block Store)
azureDisk	is used to mount a Microsoft Azure Data Disk into a Pod.
azureFile	is used to mount a Microsoft Azure File Volume (SMB 2.1 and 3.0) into a Pod.
cephfs	allows an existing CephFS volume to be mounted into your pod.
cinder	is used to mount OpenStack Block Storage into a pod.
configMap	The data stored in a ConfigMap object can be referenced in a volume of type configMap and then consumed by containerized applications running in a Pod.
downwardAPI	is used to make downward API data available to applications. It mounts a directory and writes the requested data in plain text files
emptyDir	is first created when a Pod is assigned to a Node, and exists as long as that Pod is running on that node. When a Pod is removed from a node for any reason, the data in the emptyDir is deleted forever.
fc (fibre channel)	allows an existing fibre channel volume to be mounted in a pod
flocker	allows a Flocker dataset to be mounted into a pod.
gcePersistentDisk	mounts a Google Compute Engine (GCE) Persistent Disk into your pod.
gitRepo	mounts an empty directory and clones a git repository into it for your pod to use.
glusterfs	allows a Glusterfs (an open source networked filesystem) volume to be mounted into your pod
hostPath	mounts a file or directory from the host node's filesystem into your pod.
iscsi	allows an existing iSCSI (SCSI over IP) volume to be mounted into your pod
local	represents a mounted local storage device such as a disk, partition or directory. can only be used as a statically created PersistentVolume.

K8S 存储能力-In-Tree Volume Plugins



名称	描述
nfs	allows an existing NFS (Network File System) share to be mounted into your pod
persistentVolumeClaim	is used to mount a PersistentVolume into a pod.
projected	maps several existing volume sources into the same directory.
portworxVolume	can be dynamically created through Kubernetes or it can also be pre-provisioned and referenced inside a Kubernetes pod.
quobyte	allows an existing Quobyte volume to be mounted into your pod.
rbd	allows a Rados Block Device volume to be mounted into your pod.
scaleIO	ScaleIO is a software-based storage platform that can use existing hardware to create clusters of scalable shared block networked storage. The ScaleIO volume plugin allows deployed pods to access existing ScaleIO volumes
secret	is used to pass sensitive information, such as passwords, to pods
storageos	allows an existing StorageOS volume to be mounted into your pod. StorageOS provides block storage to containers, accessible via a file system.
vsphereVolume	used to mount a vSphere VMDK Volume into your Pod.

K8S 存储能力-PersistentVolume



Kubernetes通过Persistent Volume子系统API对管理员和用户提供了存储资源创建和使用的抽象

容器应用

```
kind: Pod
apiVersion: v1
metadata:
  name: task-pv-pod
spec:
  volumes:
    - name: task-pv-storage
      persistentVolumeClaim:
        claimName: task-pv-claim
  containers:
    - name: task-pv-container
      image: nginx
      ports:
        - containerPort: 80
          name: "http-server"
      volumeMounts:
        - mountPath:
            "/usr/share/nginx/html"
          name: task-pv-storage
```

卷需求模板

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: task-pv-claim
  annotations:
    volume.beta.kubernetes.io/storage
    e-class: "ssd"
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 5Gi
```

数据卷定义

```
kind: PersistentVolume
apiVersion: v1
metadata:
  name: task-pv-volume
  annotations:
    volume.beta.kubernetes.io/storage
    e-class: "ssd"
spec:
  capacity:
    storage: 10Gi
  accessModes:
    - ReadWriteOnce
  flexVolume:
    driver:
      "kubernetes.io/cinder"
    fsType: "ext4"
    options:
      volumeID: "vol1"
      size: "10"
```

数据卷类型

```
kind: StorageClass
apiVersion:
  storage.k8s.io/v1beta1
metadata:
  name: ssd
provisioner:
  kubernetes.io/cinder
parameters:
  type: ssd
  availability: nova
```

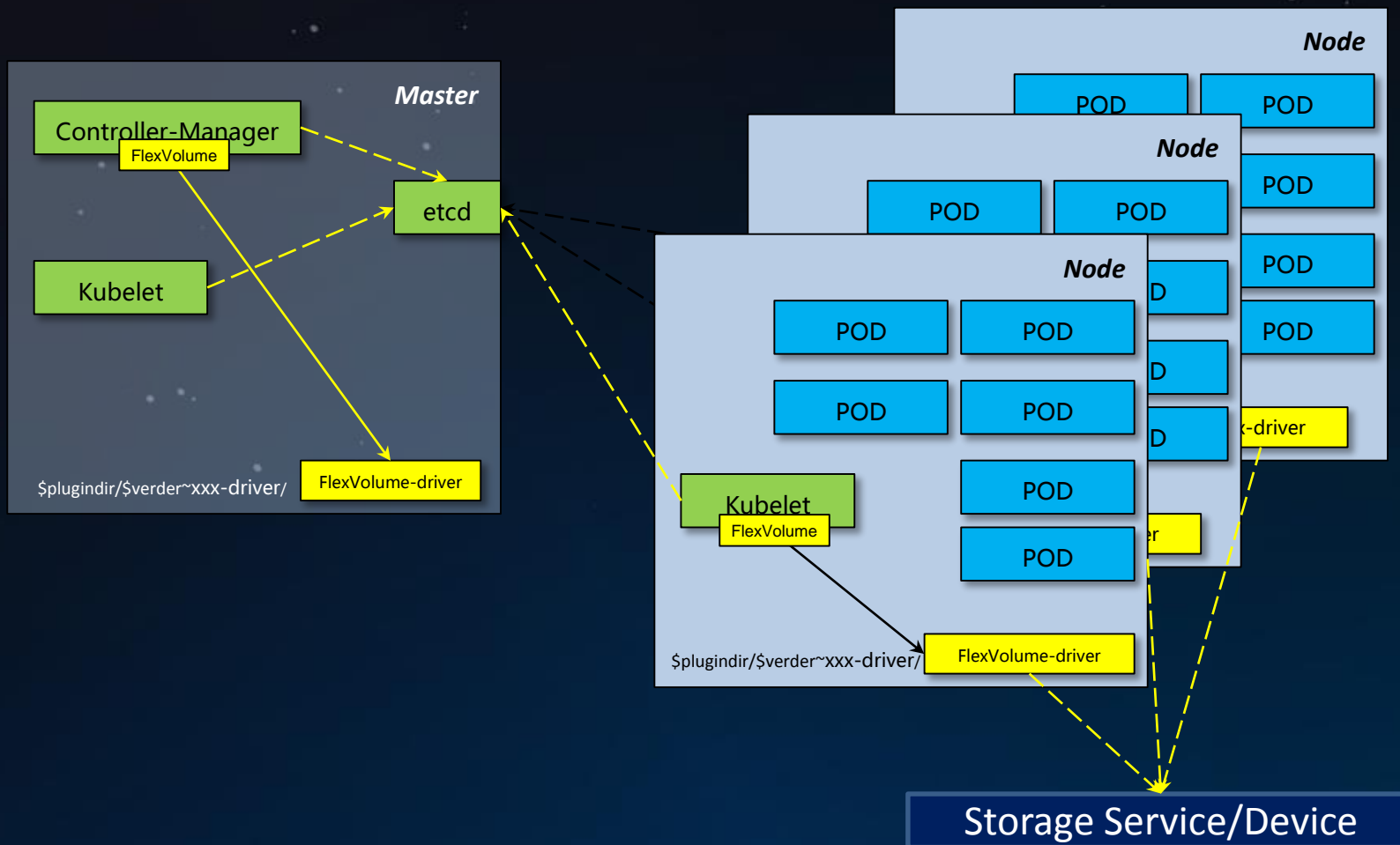
FlexVolume: 此Volume Driver允许不同厂商去开发他们自己的驱动来挂载卷到计算节点

PersistentVolumeClaim: K8提供的资源抽象的Volume Driver, 让用户不用关心具体的Volume的实现细节

大纲

- K8S存储整体框架及原理解析
- **K8S FlexVolume存储扩展机制**
- K8S CSI存储扩展机制

Flex Volume 架构



K8S 1.5引入1.8GA的Out-Of-Tree Volume Plugin :

- driver以二进制命令行形式实现FlexVolume API , 以供Controller-Manager和Kubelet调用 , 对外接口实现容易 ;
- DaemonSet方式部署确保Master和Node上都会将driver安装到插件目录 ;
- Docker镜像+yaml配置的交付形式

Flex Volume Driver部署脚本和配置



部署脚本假设所需要的驱动二进制文件，且已经被至于部署镜像的/\$DRIVER目录下：启动脚本将二进制driver重命名名为.driver，再拷贝到<plugindir>/<vendor~driver>/.driver目录下，接着使用mv将其重命名为driver（确保驱动安装的原子性），最后进入死循环确保容器活着。

```
#!/bin/sh

set -o errexit
set -o pipefail

VENDOR=k8s.io
DRIVER=nfs

# Assuming the single driver file is located at /$DRIVER inside the DaemonSet image.

driver_dir=$VENDOR${VENDOR:+~}${DRIVER}
if [ ! -d "/flexmnt/$driver_dir" ]; then
    mkdir "/flexmnt/$driver_dir"
fi

cp "/$DRIVER" "/flexmnt/$driver_dir/.$DRIVER"
mv -f "/flexmnt/$driver_dir/.$DRIVER" "/flexmnt/$driver_dir/$DRIVER"

while : ; do
    sleep 3600
done
```

```
apiVersion: extensions/v1beta1
kind: DaemonSet
metadata:
  name: flex-set
spec:
  template:
    metadata:
      name: flex-deploy
      labels:
        app: flex-deploy
    spec:
      containers:
        - image: <deployment_image>
          name: flex-deploy
          securityContext:
            privileged: true
          volumeMounts:
            - mountPath: /flexmnt
              name: flexvolume-mount
      volumes:
        - name: flexvolume-mount
          hostPath:
            path: <host_driver_directory>
```

Flex Volume CLI API



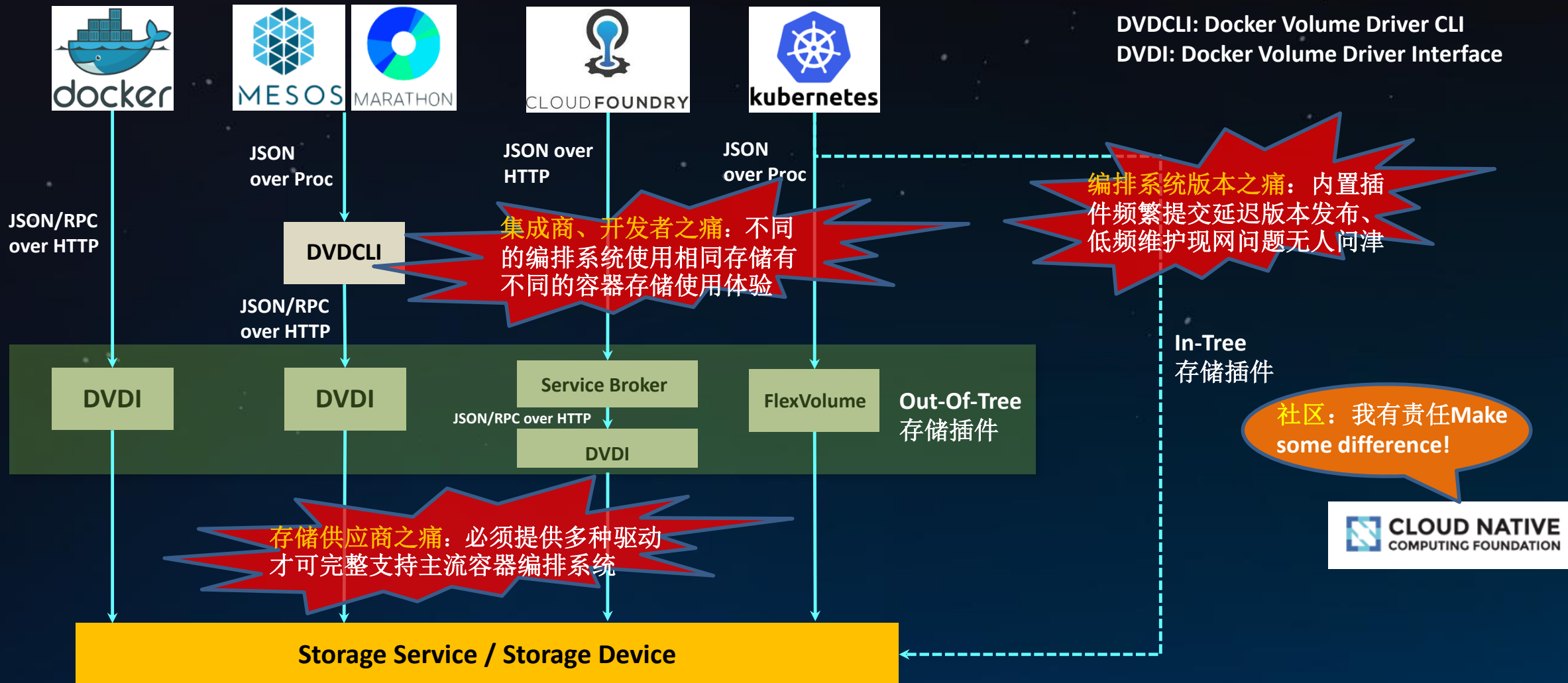
步骤	命令	描述
Init	<driver executable> init	初始化驱动。在Kubelet和Controller-Manager初始化时被调用。若调用成功则需要返回一个展示对应驱动所支持的FlexVolume能力的map，现在只包含一个必填字段attach，用于表明本驱动是否需要attach和detach操作。为向后兼容该字段一般默认值设为true。
Attach	<driver executable> attach <json options> <node name>	将给定规格的卷添加到给定的主机上。若调用成功则返回存储设备添加到该主机的路径。Kubelet和Controller-Manager都需要调用该方法。
Detach	<driver executable> detach <mount device> <node name>	卸载给定主机上的指定卷。Kubelet和Controller-Manager都需要调用该方法。
Wait for attach	<driver executable> waitforattach <mount device> <json options>	等待卷被添加到远程节点。若调用成功则将返回设备路径。Kubelet和Controller-Manager都需要调用该方法。
Volume is Attached	<driver executable> isattached <json options> <node name>	检查卷是否已被添加到节点上。Kubelet和Controller-Manager都需要调用该方法。
Mount device	<driver executable> mountdevice <mount dir> <mount device> <json options>	将存储设备挂载到一个将被pod使用的全局路径上。Kubelet需要调用该方法。
Unmount device	<driver executable> unmountdevice <mount device>	将存储设备卸载。This is called once all bind mounts have been unmounted. Kubelet需要调用该方法。
Mount	<driver executable> mount <mount dir> <json options>	将卷挂载到指定目录。Kubelet需要调用该方法。
Unmount	<driver executable> unmount <mount dir>	将卷进行卸载。Kubelet需要调用该方法。

大纲

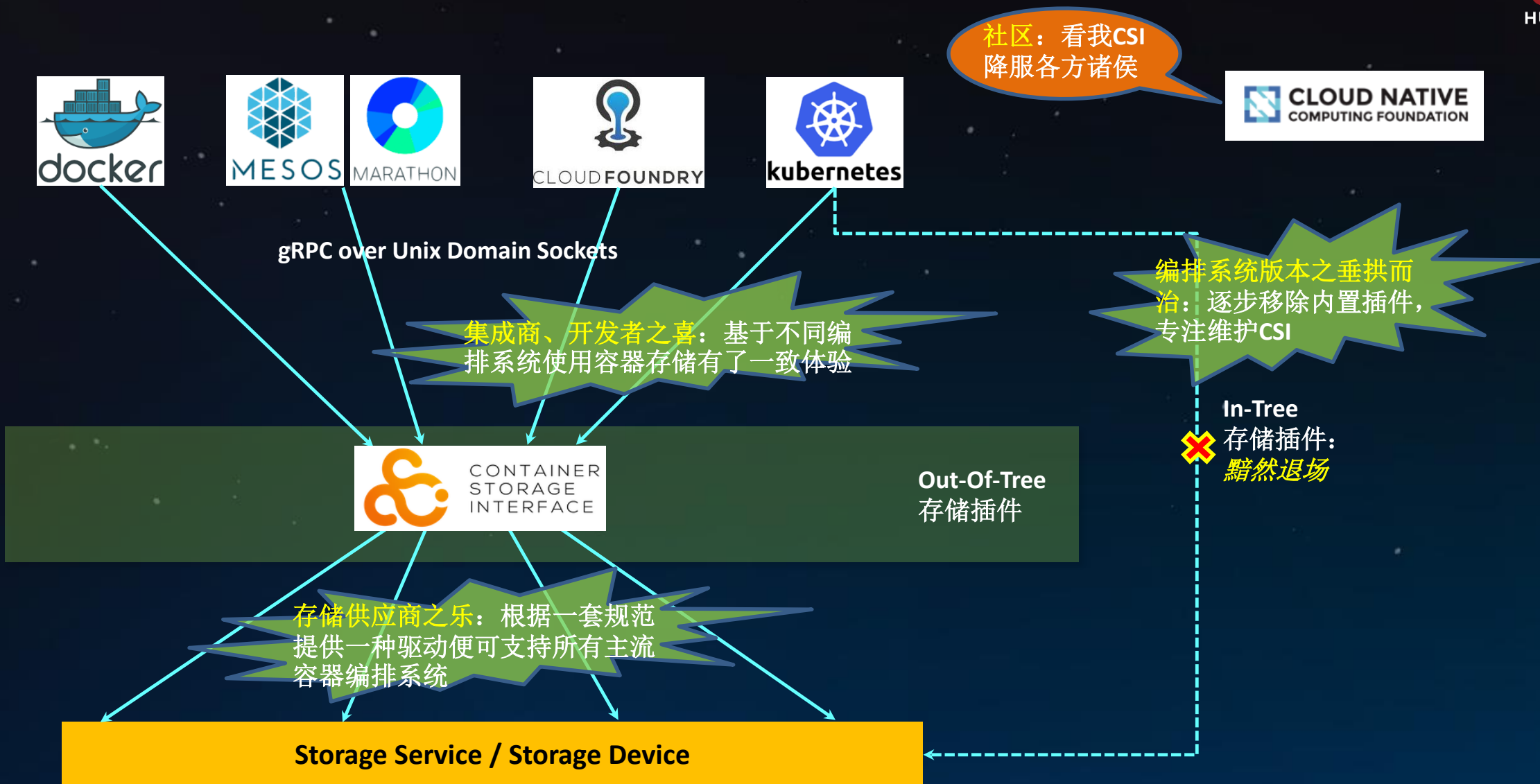
- K8S存储整体框架及原理解析
- K8S FlexVolume存储扩展机制
- **K8S CSI存储扩展机制**

术语	含义
CO	容器编排系统（Container Orchestrator），使用CSI gRPC服务来与插件通信
RPC	远程方法调用（Remote Procedure Call）
Plugin	插件实现，实现CSI服务的gRPC访问端点
SP	存储提供商（Storage Provider），负责提供CSI插件实现
Volume	卷，CO管理的容器可使用的存储单元
Block Volume	块设备卷
Mounted Volume	使用指定文件系统挂载到容器的卷，并显示为容器内的一个目录
Workload	工作负载，是CO任务调度的基本单元，可以是一个或一组容器
Node	用户运行工作负载的主机，从插件的角度通过节点 ID来进行唯一标识
In-Tree	内置的，存在于K8S核心代码仓库内的代码
Out-Of-Tree	外置的，存在于K8S核心代码仓库外的代码
CSI Volume Plugin	一个新的内置卷插件，作为一个适配器来使得外置的第三方CSI卷驱动可以被K8S所使用
CSI Volume Driver	一个外置的CSI兼容的卷插件驱动，可通过K8S卷插件被K8S所使用

业界容器编排系统的存储接口之痛



容器存储界的福音：Container Storage Interface (CSI)



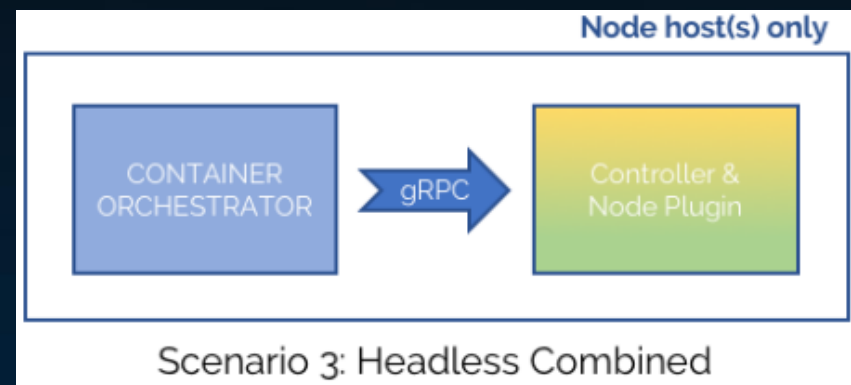
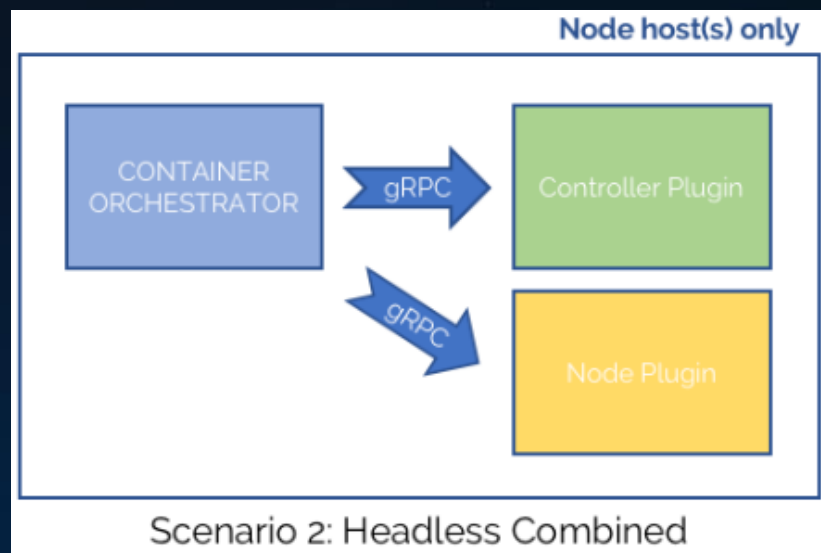
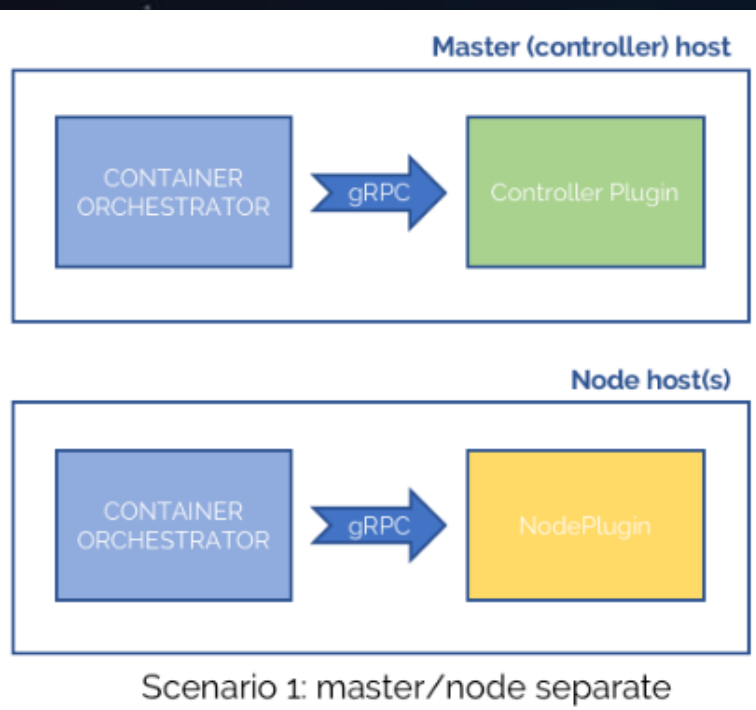
CSI通用架构



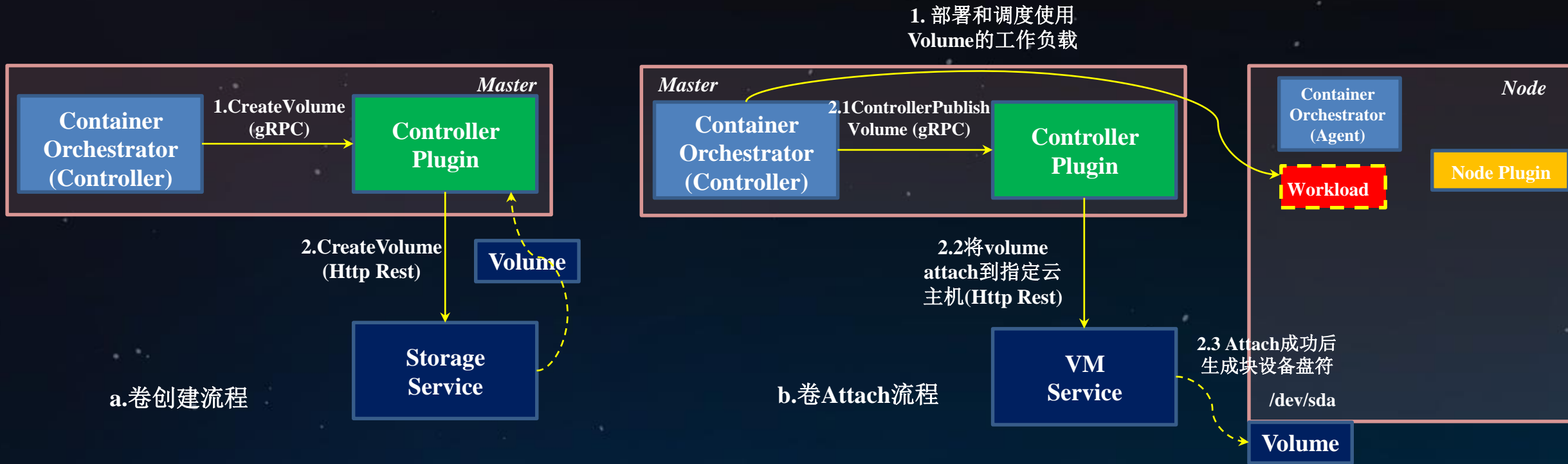
CO通过gRPC与插件交互，每个SP必须实现以下两个plugin：

- **Node Plugin**：需要运行在使用Volume的Node上，主要负责Volume Mount/Unmount等操作
- **Controller Plugin**：可以运行在任何节点上，主要负责Volume Creation/Deletion、Attach/Detach等操作

CSI有以下3种可供选择的插件部署架构：

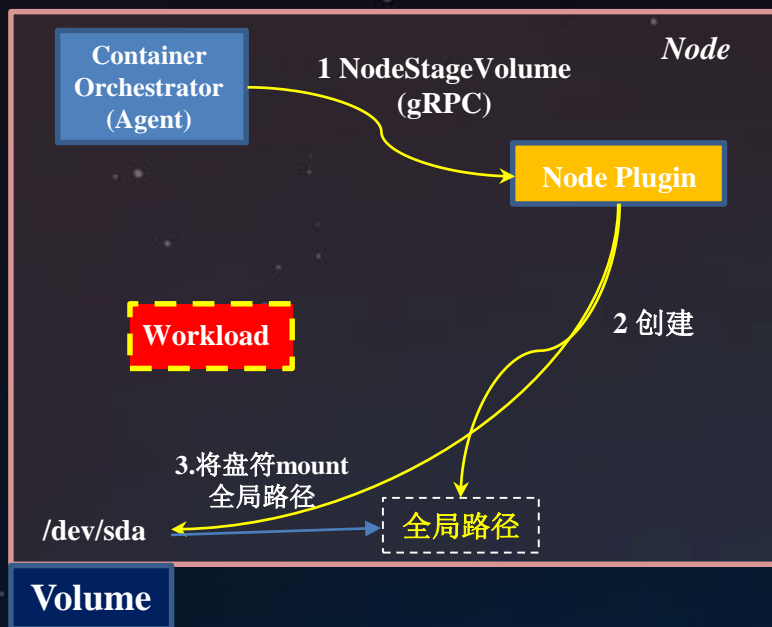


CO与Plugin的交互：01.卷的创建和Attach

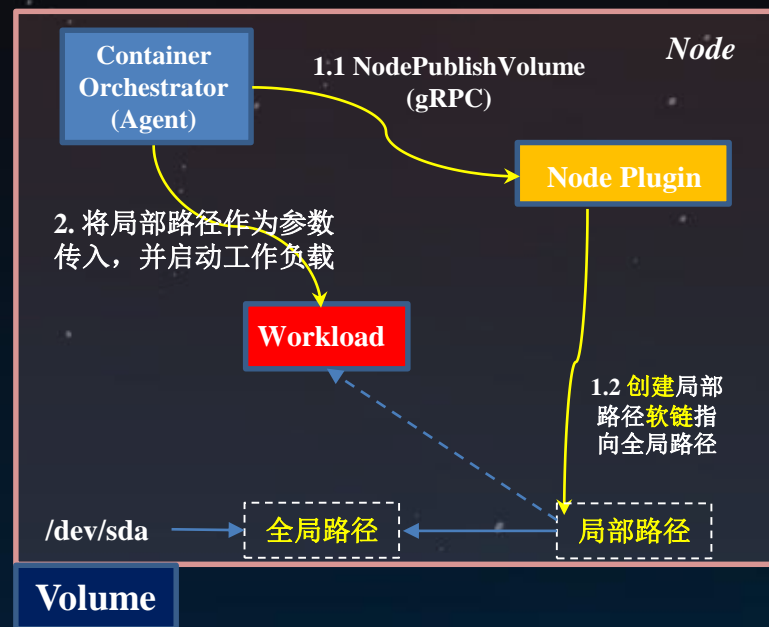


卷的Attach是由Workload的调度和启动所触发的

CO与Plugin的交互：02.将卷mount到全局路径、workload挂载路径



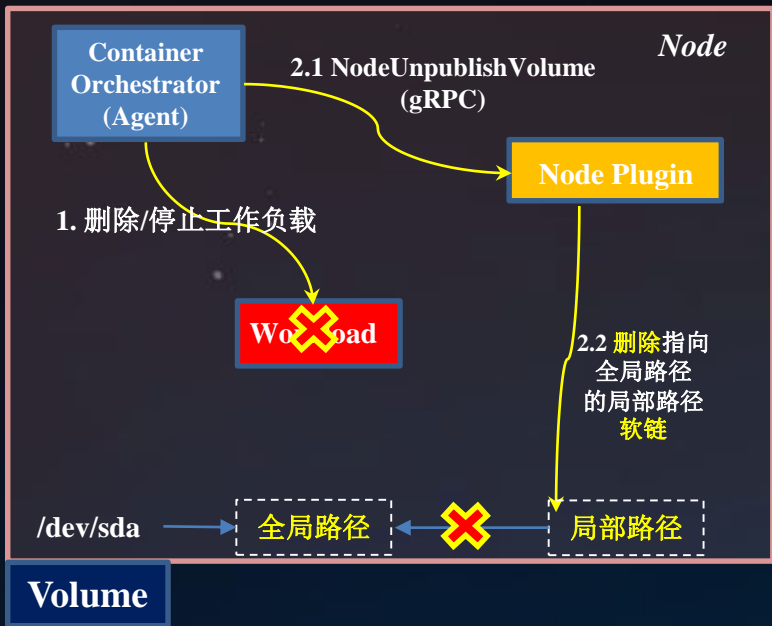
c.卷mount到全局路径流程



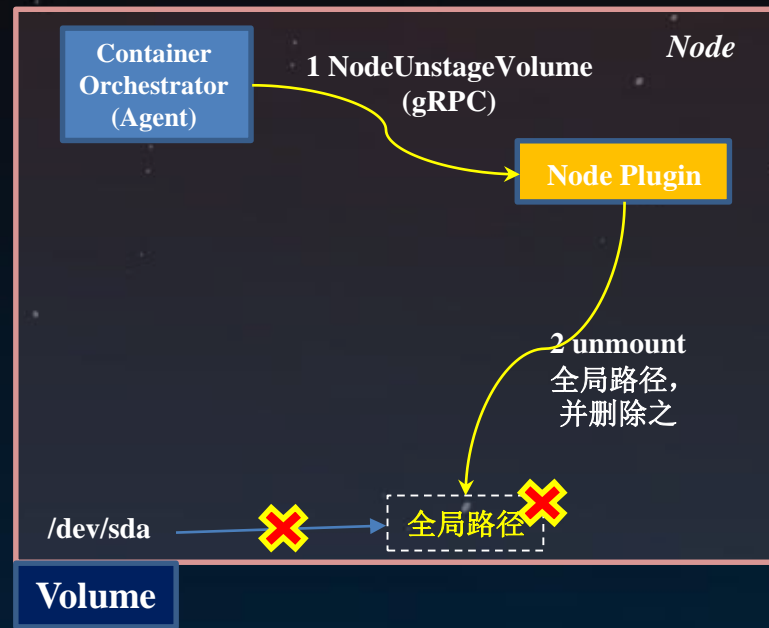
d.卷mount到workload挂载路径流程

- 卷的**Mount**系列操作是由**Workload**的启动所触发的
- K8S中volume的**全局挂载路径**（存储挂载点）格式样例：
`/var/lib/kubelet/plugins/kubernetes.io/$volume_plugin/mounts/$volume_name`
- K8S中volume的**workload挂载路径**（软链）格式样例：
`/var/lib/kubelet/pods/$pod_id/volumes/$volume_plugin/$volume_name`

CO与Plugin的交互：03.将卷从workload挂载路径、全局路径unmount



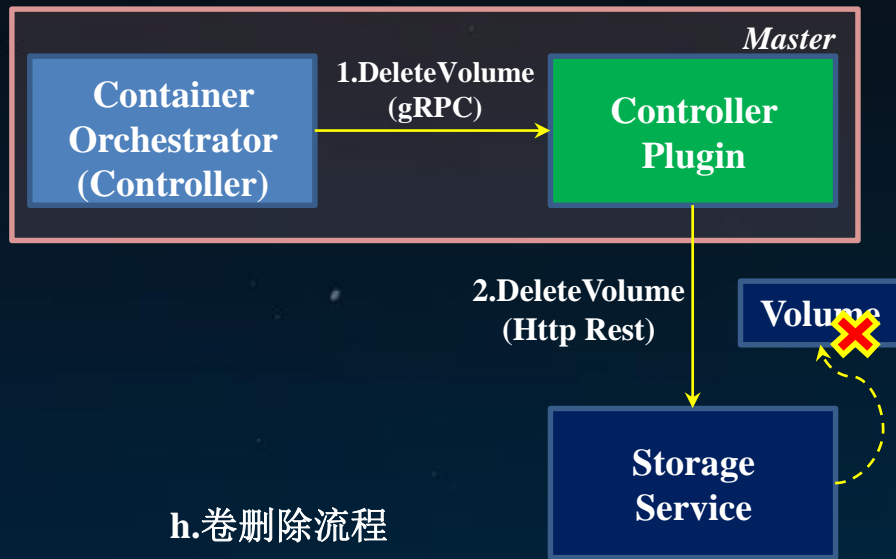
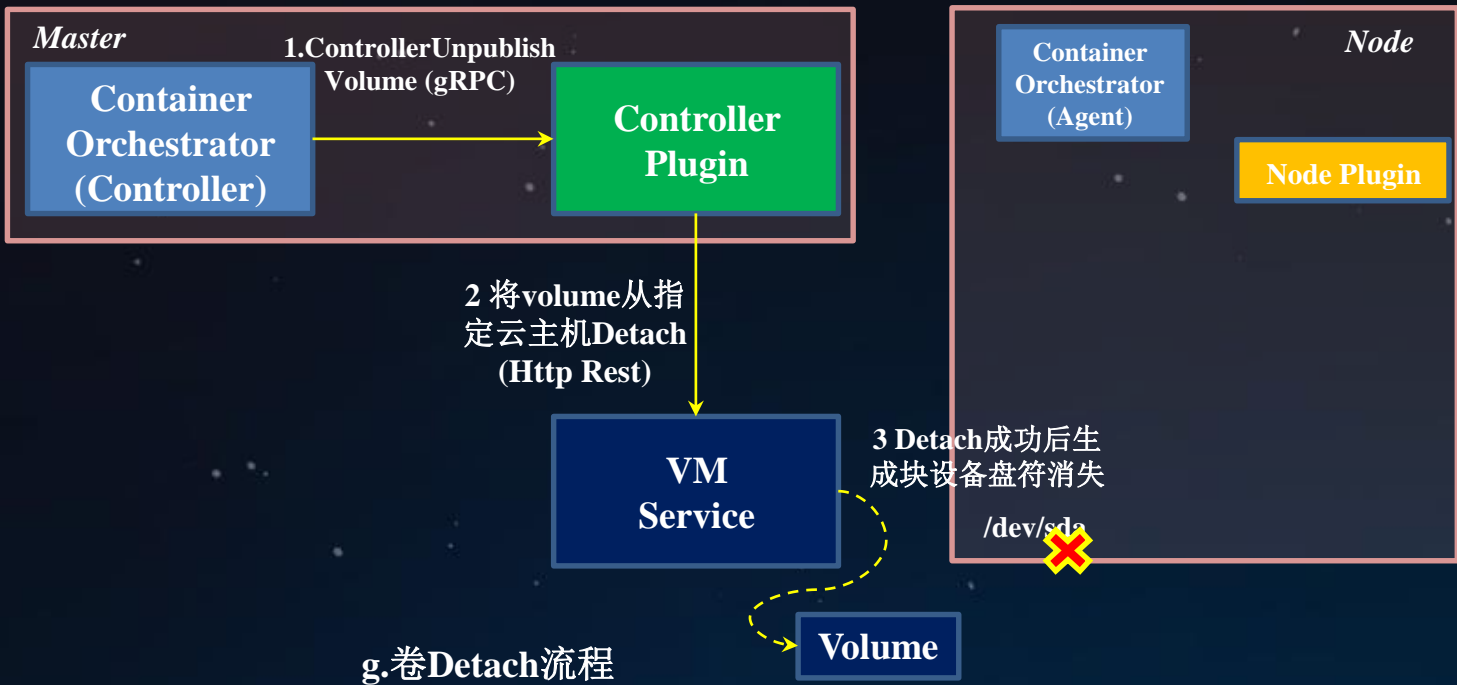
e. workload停止，将卷从局部路径unmount(删除指向全局路径的局部路径软链)流程



f.将卷从全局路径unmount流程

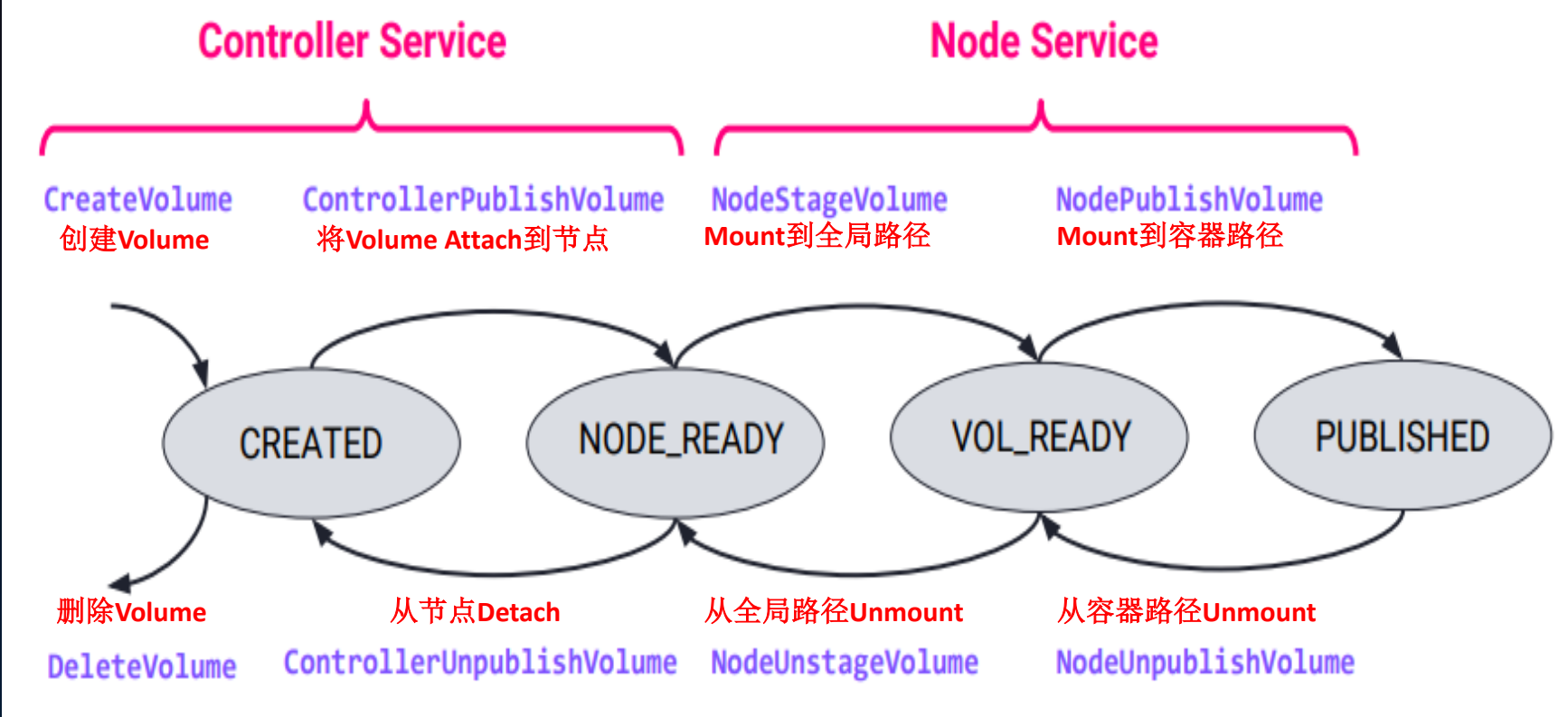
卷的unmount系列操作是由Workload的重启、迁移、删除所触发的

CO与Plugin的交互：04.卷的Detach和删除



卷的Detach是由Workload的重启、迁移、删除所触发的

卷的生命周期



RPC接口集合-Identity



CSI规范定义了**3类RPC集合**：

- **Identity Service**：Node Plugin和Controller Plugin都需要实现的RPC集合
- **Controller Service**：Controller Plugin需要实现的RPC集合
- **Node Service**：Node Plugin需要实现的RPC集合

Identity Service RPC：身份服务RPC允许CO查询插件的功能，健康状况和其他元数据。

```
service Identity {  
  rpc GetPluginInfo(GetPluginInfoRequest)  
    returns (GetPluginInfoResponse) {}  
  
  rpc GetPluginCapabilities(GetPluginCapabilitiesRequest)  
    returns (GetPluginCapabilitiesResponse) {}  
  
  rpc Probe (ProbeRequest)  
    returns (ProbeResponse) {}  
}
```

RPC接口集合-Controller



```
service Controller {  
    rpc CreateVolume (CreateVolumeRequest)  
        returns (CreateVolumeResponse) {}  
  
    rpc DeleteVolume (DeleteVolumeRequest)  
        returns (DeleteVolumeResponse) {}  
  
    rpc ControllerPublishVolume (ControllerPublishVolumeRequest)  
        returns (ControllerPublishVolumeResponse) {}  
  
    rpc ControllerUnpublishVolume (ControllerUnpublishVolumeRequest)  
        returns (ControllerUnpublishVolumeResponse) {}  
  
    rpc ValidateVolumeCapabilities (ValidateVolumeCapabilitiesRequest)  
        returns (ValidateVolumeCapabilitiesResponse) {}  
  
    rpc ListVolumes (ListVolumesRequest)  
        returns (ListVolumesResponse) {}  
  
    rpc GetCapacity (GetCapacityRequest)  
        returns (GetCapacityResponse) {}  
  
    rpc ControllerGetCapabilities (ControllerGetCapabilitiesRequest)  
        returns (ControllerGetCapabilitiesResponse) {}  
  
    rpc CreateSnapshot (CreateSnapshotRequest)  
        returns (CreateSnapshotResponse) {}  
  
    rpc DeleteSnapshot (DeleteSnapshotRequest)  
        returns (DeleteSnapshotResponse) {}  
  
    rpc ListSnapshots (ListSnapshotsRequest)  
        returns (ListSnapshotsResponse) {}  
}
```

Controller Service RPC : 控制服务RPC提供卷的创建、删除、Attach、Detach、查询等功能，以及卷快照的创建、删除、查询等功能

RPC接口集合-Node



```
service Node {
  rpc NodeStageVolume (NodeStageVolumeRequest)
    returns (NodeStageVolumeResponse) {}

  rpc NodeUnstageVolume (NodeUnstageVolumeRequest)
    returns (NodeUnstageVolumeResponse) {}

  rpc NodePublishVolume (NodePublishVolumeRequest)
    returns (NodePublishVolumeResponse) {}

  rpc NodeUnpublishVolume (NodeUnpublishVolumeRequest)
    returns (NodeUnpublishVolumeResponse) {}

  // NodeGetId is being deprecated in favor of NodeGetInfo and will be
  // removed in CSI 1.0. Existing drivers, however, may depend on this
  // RPC call and hence this RPC call MUST be implemented by the CSI
  // plugin prior to v1.0.
  rpc NodeGetId (NodeGetIdRequest)
    returns (NodeGetIdResponse) {
    option deprecated = true;
  }

  rpc NodeGetCapabilities (NodeGetCapabilitiesRequest)
    returns (NodeGetCapabilitiesResponse) {}

  // Prior to CSI 1.0 - CSI plugins MUST implement both NodeGetId and
  // NodeGetInfo RPC calls.
  rpc NodeGetInfo (NodeGetInfoRequest)
    returns (NodeGetInfoResponse) {}
}
```

将卷mount到指定全局路径，此方法调用必须在NodePublishVolume之前，该方法是每个卷/每个节点执行一次

将卷从指定全局路径unmount

将卷从指定全局路径mount到目标路径，此方法调用必须在NodePublishVolume之前，该方法是每个卷/每个工作负载执行一次

将卷从指定目标路径unmount

返回节点Id

返回节点插件的功能

返回插件所在节点信息

K8S CSI架构



K8S 1.9实现了CSI plugin alpha版本，1.11版本已升至Beta

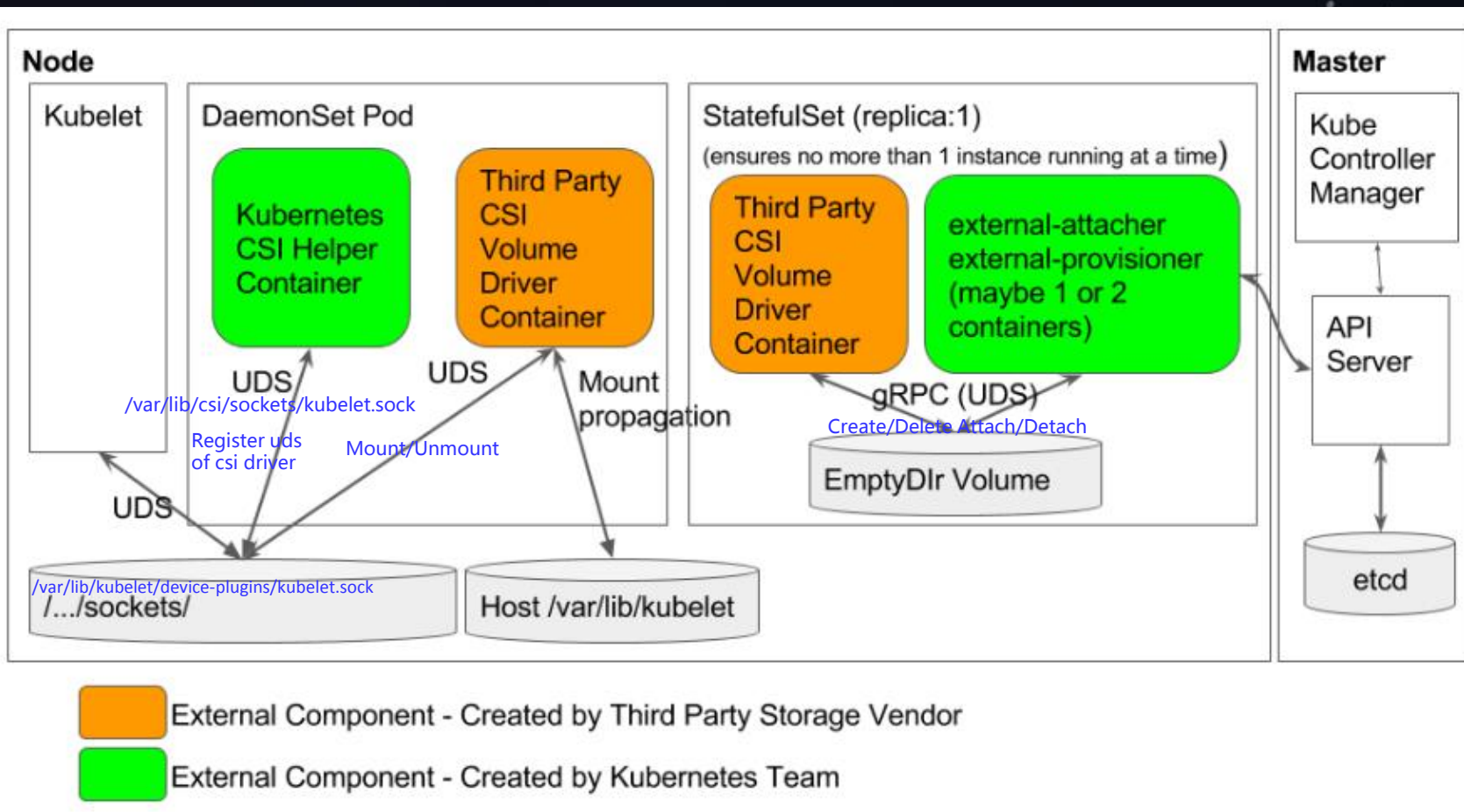
为了部署一个容器化的第三方CSI volume driver，存储提供商需要执行如下操作：

1. 创建一个实现CSI规范描述的插件功能，并通过Unix套接字来暴露gRPC访问接口的“CSI volume driver”容器；
2. 结合使用K8S团队提供的帮助容器来部署CSI volume driver，具体需要创建如下两类K8S对象：

1) StatefulSet：用于与K8S控制器进行交互，实例数1，包含3个容器（CSI volume driver、external-attacher、external-provisioner），需要挂载一个挂载点为/var/lib/csi/sockets/pluginproxy/的emptyDir volume

2) DaemonSet：包含2个容器（CSI volume driver、K8S CSI Helper），挂载3个hostpath volume

3. 集群管理员为存储系统在K8S集群中部署上述StatefulSet和DaemonSet



Thank You

<http://zhibo.huaweicloud.com/watch/2174406>

直播 每周四 晚20:00

扫码加群技术交流



扫码报名厦门meetup

