# eSDK Enterprise Storage Plugins V3.0.1

## 用户指南(Kubernetes CSI)

**文档版本** 01

发布日期 2022-08-12





#### 版权所有 © 华为技术有限公司 2022。 保留一切权利。

非经本公司书面许可,任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部,并不得以任何形式传播。

#### 商标声明



HUAWE和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标,由各自的所有人拥有。

#### 注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束,本文档中描述的全部或部分产品、服务或 特性可能不在您的购买或使用范围之内。除非合同另有约定,华为公司对本文档内容不做任何明示或暗示的声 明或保证。

由于产品版本升级或其他原因,本文档内容会不定期进行更新。除非另有约定,本文档仅作为使用指导,本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

## 华为技术有限公司

地址: 深圳市龙岗区坂田华为总部办公楼 邮编: 518129

网址: <a href="https://e.huawei.com">https://e.huawei.com</a>

## 前言

## 读者对象

本文档主要适用于以下读者对象:

- 技术支持工程师
- 运维工程师
- 具备存储和Kubernetes基础知识的工程师

## 符号约定

在本文中可能出现下列标志,它们所代表的含义如下。

| 符号   | 说明  |
|------|---|
| ▲ 危险 | 表示如不避免则将会导致死亡或严重伤害的具有高等级风险的危害。  |
| ▲ 警告 | 表示如不避免则可能导致死亡或严重伤害的具有中等级风险的危害。  |
| ⚠ 注意 | 表示如不避免则可能导致轻微或中度伤害的具有低等级风险的危害。  |
| 须知   | 用于传递设备或环境安全警示信息。如不避免则可能会导致设备<br>损坏、数据丢失、设备性能降低或其它不可预知的结果。<br>"须知"不涉及人身伤害。 |
| □ 说明 | 对正文中重点信息的补充说明。<br>"说明"不是安全警示信息,不涉及人身、设备及环境伤害信<br>息。                       |

## 修改记录

| 文档版本 | 发布日期       | 修改说明     |
|------|------------|----------|
| 01   | 2022-08-12 | 第一次正式发布。 |

## 目录

| <u> </u>                                | ii |
|---|----|
| 1 概述                                    | 1  |
| 2 特性和兼容性                                |    |
| 2.1 兼容性                                 |    |
| 2.2 Kubernetes 特性                       |    |
| 2.3 存储特性                                | 6  |
| 3 安装前准备                                 | 8  |
| 3.1 前提条件                                |    |
| 3.2 获取软件包                               |    |
| 3.3 上传软件包组件                             |    |
| 3.4 制作华为 CSI 镜像                         | 9  |
| 3.5 上传华为 CSI 镜像                         | 10 |
| 3.6 配置主机多路径                             | 11 |
| 3.6.1 安装原生多路径软件                         | 12 |
| 3.6.1.1 安装多路径工具包                        | 12 |
| 3.6.1.2 配置多路径服务                         | 12 |
| 3.6.2 安装华为多路径软件                         | 14 |
| 3.6.2.1 安装华为多路径工具                       | 14 |
| 4 安装华为 CSI                              | 15 |
| 4.1 使用 Helm 安装华为 CSI                    | 15 |
| 4.1.1 安装 Helm                           | 15 |
| 4.1.2 安装华为 CSI                          | 16 |
| 4.2 手动安装华为 CSI                          | 25 |
| 4.2.1 对接企业存储配置                          | 25 |
| 4.2.1.1 通过 iSCSI 对接企业存储 SAN 配置          | 25 |
| 4.2.1.2 通过 FC 对接企业存储 SAN 配置             | 28 |
| 4.2.1.3 通过 NFS 对接企业存储 NAS 配置            | 30 |
| 4.2.1.4 通过 NVMe over RoCE 对接企业存储 SAN 配置 | 32 |
| 4.2.1.5 通过 NVMe over FC 对接企业存储 SAN 配置   | 34 |
| 4.2.2 对接分布式存储配置                         |    |
| 4.2.2.1 通过 SCSI 对接分布式存储 SAN 配置          |    |
| 4.2.2.2 通过 iSCSI 对接分布式存储 SAN 配置         | 38 |

| 4.2.2.3 通过 NFS 对接分布式存储 NAS 配置       |    |
|-------------------------------------|----|
| 4.2.2.4 通过 DPC 对接分布式存储 NAS 配置       |    |
| 4.2.3 启动 huawei-csi 服务              | 44 |
| 5 升级操作                              | 50 |
| 5.1 使用 Helm 升级华为 CSI                | 50 |
| 5.1.1 升级 CSI                        | 50 |
| 5.1.2 回退 CSI                        | 51 |
| 5.2 手动升级华为 CSI                      | 51 |
| 5.2.1 卸载原 CSI                       | 52 |
| 5.2.1.1 卸载 huawei-csi-node 服务       | 52 |
| 5.2.1.2 卸载 huawei-csi-controller 服务 | 52 |
| 5.2.1.3 删除 huawei-csi-configmap 对象  | 52 |
| 5.2.1.4 删除 huawei-csi-secret 对象     | 53 |
| 5.2.1.5 删除 RBAC 权限                  | 53 |
| 5.2.1.6 删除老版本镜像                     | 54 |
| 5.2.2 安装新 CSI                       | 55 |
| 6 使用说明                              | 57 |
| 6.1 (条件必选)管理 StorageClass           | 57 |
| 6.1.1 创建 StorageClass               | 57 |
| 6.1.1.1 创建 LUN StorageClass         | 57 |
| 6.1.1.2 创建文件系统 StorageClass         | 58 |
| 6.1.2 删除 StorageClass               | 61 |
| 6.2 (条件必选)管理 PV                     | 61 |
| 6.2.1 创建 PV                         | 61 |
| 6.2.2 删除 PV                         | 64 |
| 6.3 管理 PVC                          | 64 |
| 6.3.1 创建 PVC                        | 64 |
| 6.3.2 (可选)扩容 PVC                    | 67 |
| 6.3.3 (可选)克隆 PVC                    |    |
| 6.3.4 (可选)从快照创建 PVC                 | 71 |
| 6.3.5 删除 PVC                        |    |
| 6.4 管理 Pod                          | 73 |
| 6.4.1 创建 Pod                        | 73 |
| 6.4.2 删除 Pod                        | 75 |
| 6.5 (可选)管理快照                        |    |
| 6.5.1 安装 Snapshot 依赖组件服务            | 75 |
| 6.5.2 管理 VolumeSnapshotClass        |    |
| 6.5.2.1 创建 VolumeSnapshotClass      |    |
| 6.5.2.2 删除 VolumeSnapshotClass      |    |
| 6.5.3 管理 VolumeSnapshot             |    |
| 6.5.3.1 创建 VolumeSnapshot           |    |
| 6.5.3.2 删除 VolumeSnapshot           | 79 |

| 7 高级特性   | 80  |
|--|-----|
| 7.1 配置多个后端   | 80  |
| 7.1.1 通过 Helm 配置多个后端   | 80  |
| 7.1.2 手动配置多个后端   | 81  |
| 7.2 指定后端创建 PVC   | 82  |
| 7.3 指定存储池创建 PVC  | 82  |
| 7.4 配置 ALUA 特性   | 83  |
| 7.4.1 通过 Helm 配置 ALUA 特性   | 83  |
| 7.4.1.1 通过 Helm 配置 OceanStor V3/V5 系列和 OceanStor Dorado V3 系列 ALUA | 83  |
| 7.4.1.2 通过 Helm 配置 OceanStor Dorado 6.x ALUA                       | 86  |
| 7.4.1.3 通过 Helm 配置分布式存储 ALUA                                       | 89  |
| 7.4.2 手动配置 ALUA 特性   | 90  |
| 7.4.2.1 配置 OceanStor V3/V5 系列和 OceanStor Dorado V3 系列 ALUA         | 90  |
| 7.4.2.2 配置 OceanStor Dorado 6.x ALUA                               | 93  |
| 7.4.2.3 配置分布式存储 ALUA   | 95  |
| 7.5 配置存储拓扑感知   | 97  |
| 7.5.1 通过 Helm 配置存储拓扑感知   | 98  |
| 7.5.2 手动配置存储拓扑感知   | 100 |
| 7.6 企业存储高级特性   | 102 |
| 7.6.1 配置 QoS   | 102 |
| 7.6.2 配置租户   | 104 |
| 7.6.2.1 通过 Helm 配置租户   | 104 |
| 7.6.2.2 手动配置租户   | 105 |
| 7.6.3 配置 NAS 双活  | 106 |
| 7.6.3.1 前提条件   | 107 |
| 7.6.3.2 使用 Helm 配置 NAS 双活  | 109 |
| 7.6.3.3 手动配置 NAS 双活  | 111 |
| 7.6.4 配置应用类型   | 112 |
| 7.7 分布式存储高级特性  | 114 |
| 7.7.1 配置 QoS   | 114 |
| 7.7.2 配置软配额  | 114 |
| 7.7.3 配置帐户   | 115 |
| 7.7.3.1 通过 Helm 配置帐户   | 116 |
| 7.7.3.2 手动配置帐户   | 117 |
| 7.7.4 DPC 场景配置挂载参数   | 118 |
| 8 卸载 CSI   | 120 |
| 8.1 使用 Helm 卸载 huawei-csi  |     |
| 8.2 手动卸载 huawei-csi  |     |
| 8.2.1 卸载 huawei-csi-node 服务  | 121 |
| 8.2.2 卸载 huawei-csi-controller 服务                                  | 121 |
| 8.2.3 删除 huawei-csi-configmap 对象                                   |     |
| 8.2.4 删除 huawei-csi-secret 对象                                      | 122 |

| 8.2.5 删除 RBAC 权限   | 122 |
|--|-----|
| 8.2.6 删除老版本镜像  |     |
| 8.3 (可选)卸载 Snapshot 依赖组件服务   |     |
| 9 常用操作   | 125 |
| 9.1 更新 CSI 上配置的存储用户名或密码  |     |
| 9.2 更新 huawei-csi 的 configmap 对象                                     |     |
| 9.2.1 使用 Helm 更新 configmap 对象  |     |
| 9.2.2 手动更新 configmap 对象  |     |
| 9.3 为 huawei-csi 新增后端  | 129 |
| 9.3.1 使用 Helm 新增后端   | 129 |
| 9.3.2 手动新增后端   | 130 |
| 9.4 更新 huawei-csi-controller 服务                                      | 130 |
| 9.4.1 使用 Helm 更新 controller 服务                                       | 130 |
| 9.4.2 手动更新 controller 服务   | 131 |
| 9.5 更新 huawei-csi-node 服务  | 132 |
| 9.5.1 使用 Helm 更新 node 服务   | 132 |
| 9.5.2 手动更新 node 服务   | 133 |
| 9.6 修改日志输出模式   | 133 |
| 9.6.1 使用 Helm 修改 controller 或 node 服务的日志输出模式                         | 133 |
| 9.6.2 手动修改 huawei-csi-controller 服务的日志输出模式                           | 134 |
| 9.6.3 手动修改 huawei-csi-node 服务的日志输出模式                                 | 135 |
| 9.7 开启 ReadWriteOncePod 功能门  | 136 |
| 9.8 配置非 root 用户访问 Kubernetes 集群                                      | 138 |
| 10 FAQ   | 139 |
| 10.1 查看日志信息  | 139 |
| 10.2 Kubernetes 平台第一次搭建时, iscsi_tcp 服务没有正常启动,导致创建 Pod 失败             | 140 |
| 10.3 启动 huawei-csi-node 失败,提示错误为:"/var/lib/iscsi is not a directory" | 141 |
| 10.4 集群中 worker 节点宕机并恢复后,Pod 完成 failover,但是 Pod 所在源主机出现盘符残留          | 142 |
| 10.5 启动 huawei-csi 服务时,服务启动异常, 状态显示 InvalidImageName                 | 144 |
| 10.6 创建 PVC 时, PVC 的状态为 Pending                                      | 145 |
| 10.7 删除 PVC 前,PVC 的状态为 Pending                                       | 147 |
| 10.8 创建 Pod 时,Pod 的状态为 ContainerCreating                             |     |
| 10.9 创建 Pod 时,Pod 的状态长时间处于 ContainerCreating 状态                      | 149 |
| 11 附录  | 150 |
| 11.1 OceanStor V3/V5 系列和 OceanStor Dorado V3 系列 ALUA 特性配置策略样例        |     |
| 11.2 OceanStor Dorado 6.x ALUA 特性配置策略样例                              | 151 |
| 11.3 分布式存储 ALUA 特性配置策略样例   |     |

● 概述

本文档主要介绍Kubernetes配套华为企业存储和分布式存储产品的CSI插件安装部署和使用,达到利用华为存储向Kubernetes提供持久化卷存储能力的目的。

# 2 特性和兼容性

本章节介绍华为CSI支持的特性和兼容性。

- 2.1 兼容性
- 2.2 Kubernetes特性
- 2.3 存储特性

## 2.1 兼容性

表 2-1 支持的华为存储产品

| 存储产品                | 版本                         |
|---------------------|----------------------------|
| OceanStor Dorado V6 | 6.0.1, 6.1.0, 6.1.2, 6.1.3 |
| OceanStor Dorado V3 | V300R002                   |
| OceanStor V6        | 6.1.3                      |
| OceanStor V5/F V5   | V500R007, V500R007 Kunpeng |
| OceanStor V3/F V3   | V300R006                   |
| FusionStorage       | V100R006C30                |
| FusionStorage Block | 8.0.0, 8.0.1               |
| OceanStor Pacific   | 8.1.0, 8.1.1, 8.1.2        |

表 2-2 支持的容器平台和操作系统

| 容器平台/操作系统  | 版本   |
|------------|--|
| Kubernetes | 1.21, 1.22, 1.23, 1.24                         |
| CentOS     | 7.6 x86_64, 7.7 x86_64, 7.9 x86_64, 8.2 x86_64 |

| 容器平台/操作系统      | 版本  |
|----------------|---|
| SUSE           | 15 SP2 x86_64   |
| Red Hat CoreOS | 4.6 x86_64, 4.7 x86_64, 4.8 x86_64, 4.9 x86_64, 4.10 x86_64 |
| Ubuntu         | 18.04 x86_64, 20.04 x86_64                                  |
| Kylin V10      | SP1 Arm, SP2 Arm  |

#### 须知

如果宿主机操作系统为CoreOS 4.6/4.7/4.8/4.9/4.10,相关的操作请参见《Kubernetes CSI for Red Hat OpenShift 用户指南》。

#### 表 2-3 宿主机操作系统版本与多路径软件版本的配套关系

| 宿主机操作系统版<br>本                               | 原生多路径软件版本            | 华为多路径软件版本(仅企业存储支<br>持)   |
|---|----------------------|--|
| CentOS 7.6/7.7/7.9<br>(x86_64)              | 随OS自带,支持FC/<br>iSCSI | UltraPath 31.1.0,支持FC/iSCSI  |
| CentOS 8.2<br>(x86_64)                      | 随OS自带,支持FC/<br>iSCSI | UltraPath 31.1.0,支持FC/iSCSI<br>UltraPath-NVMe 31.1.RC8,支持<br>NVMe over RoCE/NVMe over FC |
| SUSE 15 SP2<br>(x86_64)                     | 随OS自带,支持FC/<br>iSCSI | UltraPath 31.1.0,支持FC/iSCSI<br>UltraPath-NVMe 31.1.RC8,支持<br>NVMe over RoCE              |
| CoreOS<br>4.6/4.7/4.8/4.9/4.1<br>0 (x86_64) | 随OS自带,支持FC/<br>iSCSI | 不支持  |
| Ubuntu<br>18.04/20.04<br>(x86_64)           | 随OS自带,支持FC/<br>iSCSI | 不支持  |
| Kylin V10 SP1/SP2<br>(Arm)                  | 随OS自带,支持FC/<br>iSCSI | 不支持  |

## 2.2 Kubernetes 特性

表 2-4 企业存储支持的 Kubernetes 特性

| 特性                                | V3  | tor V3/F<br>tor V5/F | OceanS<br>tor<br>Dorado<br>V3 | OceanS       | tor V6 | OceanS<br>Dorado |     |
|-----------------------------------|-----|----------------------|-------------------------------|--------------|--------|------------------|-----|
| -                                 | SAN | NAS                  | SAN                           | SAN          | NAS    | SAN              | NAS |
| Dynamic<br>Provisioning           | V   | $\sqrt{}$            | V                             | V            | V      | V                | V   |
| Static<br>Provisioning            | V   | V                    | V                             | V            | V      | V                | V   |
| Expand<br>Persistent<br>Volume    | V   | V                    | V                             | V            | V      | V                | V   |
| Create<br>VolumeSnaps<br>hot      | V   | V                    | V                             | V            | V      | V                | V   |
| Create<br>Volume from<br>Snapshot | V   | V                    | V                             | V            | ×      | V                | ×   |
| Delete<br>Snapshot                | V   | V                    | V                             | V            | V      | V                | V   |
| CSI Volume<br>Cloning             | V   | V                    | V                             | V            | ×      | V                | ×   |
| CSI Raw Block<br>Volume           | V   | ×                    | V                             | V            | ×      | V                | ×   |
| Topology                          | V   | V                    |                               | $\checkmark$ | V      | V                | V   |

表 2-5 分布式存储支持的 Kubernetes 特性

| 特性                      | FusionStorag<br>e | FusionStorag<br>e Block | OceanStor Pacific |     |
|-------------------------|-------------------|-------------------------|-------------------|-----|
| -                       | -                 | -                       | SAN               | NAS |
| Dynamic<br>Provisioning | V                 | $\checkmark$            | $\checkmark$      | V   |
| Static<br>Provisioning  | V                 | $\checkmark$            | V                 | V   |

| 特性                             | FusionStorag<br>e | FusionStorag<br>e Block | OceanStor Pacific |   |
|--------------------------------|-------------------|-------------------------|-------------------|---|
| Expand<br>Persistent<br>Volume | V                 | V                       | V                 | × |
| Create<br>VolumeSnapsho<br>t   | V                 | V                       | V                 | × |
| Create Volume from Snapshot    | V                 | V                       | V                 | × |
| Delete Snapshot                | V                 | V                       | V                 | × |
| CSI Volume<br>Cloning          | $\checkmark$      | V                       | V                 | × |
| CSI Raw Block<br>Volume        | V                 | V                       | V                 | × |
| Topology                       | V                 | V                       | $\sqrt{}$         | V |

#### 表 2-6 支持的 Kubernetes 特性限制说明

| 特性                             | 限制说明   |
|--------------------------------|--|
| PVC Access Modes               | <ul> <li>RWO/ROX/RWOP: 所有类型卷均支持, RWOP访问模式仅Kubernetes 1.22及以上版本支持。</li> <li>RWX: 仅Raw Block (SAN) 卷和NAS类型的卷支持。</li> </ul> |
| Create VolumeSnapshot          | <ul><li>NAS双活不支持创建快照。</li><li>静态PV创建的PVC不支持创建快照。</li></ul>   |
| Create Volume from<br>Snapshot | <ul><li>源PVC和目标PVC的StorageClass、volumeMode需要相同。</li><li>NAS双活不支持从快照创建卷。</li></ul>  |
| Expand Persistent Volume       | <ul><li>仅支持扩容,不支持缩容。</li><li>访问模式为ROX的PVC不支持扩容。</li><li>静态PV创建的PVC不支持扩容。</li></ul>                                       |
| CSI Volume Cloning             | <ul> <li>源PVC和目标PVC的StorageClass、volumeMode需要相同。</li> <li>静态PV创建的PVC不支持克隆。</li> <li>NAS双活不支持克隆。</li> </ul>               |

## 2.3 存储特性

表 2-7 支持的企业存储特性

| 标准特性                 | OceanSto<br>V3/V3<br>OceanSto<br>V5/V5 |              | OceanS<br>tor<br>Dorado<br>V3 | OceanSt | or V6     | OceanS<br>Dorado |              |
|----------------------|--|--------------|-------------------------------|---------|-----------|------------------|--------------|
| -                    | SAN                                    | NAS          | SAN                           | SAN     | NAS       | SAN              | NAS          |
| QoS                  | $\sqrt{}$                              | V            |                               | V       | V         | V                | $\checkmark$ |
| 应用类型                 | ×                                      | ×            | ×                             | ×       | ×         | V                | $\checkmark$ |
| 双活                   | ×                                      | ×            | ×                             | ×       | V         | ×                | √<br>(6.1.3) |
| 多租户                  | ×                                      | V            | ×                             | ×       | V         | ×                | √<br>(6.1.3) |
| NFS 3.0              | ×                                      | V            | ×                             | ×       | V         | ×                | $\checkmark$ |
| NFS 4.0 <sup>1</sup> | ×                                      | V            | ×                             | ×       | $\sqrt{}$ | ×                | $\sqrt{}$    |
| NFS 4.1 <sup>1</sup> | ×                                      | $\checkmark$ | ×                             | ×       | V         | ×                | $\checkmark$ |

注释1: 部分企业存储不支持开启NFS 4.0或NFS 4.1,请以存储的实际能力为准。

表 2-8 支持的分布式存储特性

| 标准特性    | FusionStorag<br>e | FusionStorage<br>Block | OceanStor Pacific |              |
|---------|-------------------|------------------------|-------------------|--------------|
| -       | -                 | -                      | Block             | File         |
| QoS     | V                 | V                      | V                 | ×            |
| 软配额     | ×                 | ×                      | ×                 |              |
| 账户      | ×                 | ×                      | ×                 | V            |
| DPC     | ×                 | ×                      | ×                 | √<br>(8.1.2) |
| NFS 3.0 | ×                 | ×                      | ×                 | V            |
| NFS 4.1 | ×                 | ×                      | ×                 | √<br>(8.1.2) |

#### 表 2-9 支持的存储特性限制说明

| 特性      | 限制描述  |
|---------|---|
| 双活      | <ul> <li>只能在双活存储均正常情况下创建PVC和Pod,如果单个存储故障,只能保证已经下发的业务正常,不能下发新业务。</li> </ul>  |
|         | <ul><li>如果两个存储均故障,请联系华为工程师处理。</li><li>仅支持文件系统双活域工作模式为双活模式</li></ul>         |
| DPC     | • DPC支持的宿主机操作系统请参考 <b>华为存储兼容性</b> 。   |
| NFS 4.0 | 通过静态PV创建的PVC来创建Pod时,不支持指定NFS<br>4.0协议挂载,但是用户可通过修改主机配置在挂载时<br>默认使用NFS 4.0协议。 |
| NFS 4.1 | 通过静态PV创建的PVC来创建Pod时,不支持指定NFS<br>4.1协议挂载,但是用户可通过修改主机配置在挂载时<br>默认使用NFS 4.1协议。 |

# **3** 安装前准备

- 3.1 前提条件
- 3.2 获取软件包
- 3.3 上传软件包组件
- 3.4 制作华为CSI镜像
- 3.5 上传华为CSI镜像
- 3.6 配置主机多路径

## 3.1 前提条件

- Kubernetes环境部署完成并正常运行。
- 华为存储正常运行。
- Kubernetes集群的所有worker主机上已安装扫盘和文件挂载依赖的驱动程序 (iSCSI, DM-Multipath, UltraPath NVMe等,多路径选择可参考多路径软件选择),如果由于系统工具欠缺导致容器和服务无法正常运行,可先参考10.1 查看日志信息进行日志查看,并自行安装主机侧工具。

## 3.2 获取软件包

#### 操作步骤

步骤1 打开浏览器,访问仓库地址: https://github.com/Huawei/eSDK\_K8S\_Plugin/releases。

**步骤2** 选择对应的版本包,下载*eSDK\_EnterPrise\_Storage\_Plugin\_\*\*\*\*\**zip,*\*\*\*\*\**表示发布版本号(当前文档对应的版本号为3.0.0)。

步骤3 解压压缩包,并获取需要的组件和资料。

----结束

## 3.3 上传软件包组件

#### 操作步骤

**步骤1** 通过解压*eSDK\_EnterPrise\_Storage\_Plugin\_\*.\*.\*\*.*zip,获取到CSI安装和使用需要的软件包和示例文件。软件包组件结构如表3-1所示。

表 3-1 软件包组件描述

| 组件                 | 组件描述                   |
|--------------------|------------------------|
| bin/huawei-csi     | 实现CSI规范接口的服务组件。        |
| bin/secretGenerate | 明文密码加密工具,用于生产secret对象。 |
| bin/secretUpdate   | 明文密码加密工具,用于更新secret对象。 |
| Helm               | Helm组件,用于部署华为CSI。      |
| deploy             | CSI部署过程中,yaml示例文件。     |
| examples           | CSI使用过程中,yaml示例文件。     |
| tools              | 镜像上传脚本。                |

步骤2 使用文件传输工具(例如Xftp),将解压后的文件上传到master节点。

----结束

## 3.4 制作华为 CSI 镜像

Huawei CSI在运行时,是以容器的形态存在。目前Huawei CSI只提供二进制文件(bin/huawei-csi),无法直接使用,因此我们需要将二进制文件制作成CSI镜像,用于启动Huawei CSI服务。

#### 前提条件

已准备好一台**已安装Docker的Linux主机**,且该主机支持访问互联网(仅用于下载镜像包)。

#### 操作步骤

步骤1 登录该Linux主机。

**步骤2** 执行**mkdir image**命令,在该主机上新建一个目录(例如:"image")。
# mkdir image

步骤3 执行cd image命令进入"image"目录。
# cd image

步骤4 拷贝huawei-csi组件到"image"目录下。

#### 步骤5 执行以下命令,创建Dockerfile文件。

# cat <<EOF > ./Dockerfile FROM busybox:stable-glibc

ADD ["huawei-csi", "/"]

RUN ["chmod", "+x", "/huawei-csi"]

ENTRYPOINT ["/huawei-csi"]

EOF

#### 须知

busybox:stable-glibc是基础镜像及其对应的TAG,此处仅为示例,请根据实际情况进行替换。

#### 步骤6 执行docker build -f Dockerfile -t huawei-csi:3.0.0. 命令制作镜像。

# docker build -f Dockerfile -t huawei-csi:3.0.0.

#### 山 说明

"3.0.0"为对应软件包名的插件版本号,此处仅为示例,请根据实际情况替换。如果环境上已经存在相同的镜像,请使用docker image rm <image-id>。

## **步骤7** 执行docker image ls | grep huawei-csi命令,检查镜像是否制作完成。显示如下回显,则表示制作完成。

# docker image ls | grep huawei-csi huawei-csi 3.0.0

c8b5726118ac

About a minute ago 39 MB

#### 步骤8 执行docker save huawei-csi:3.0.0 -o huawei-csi.tar命令,导出镜像。

# docker save huawei-csi:3.0.0 -o huawei-csi.tar

#### □ 说明

"3.0.0"为对应软件包名的插件版本号,此处仅为示例,请根据实际情况替换。

#### ----结束

## 3.5 上传华为 CSI 镜像

本章节用于介绍使用镜像上传脚本将华为CSI镜像上传至所有worker节点并加载到容器运行时中。如果使用镜像仓库,请自行上传镜像至仓库中,并跳过本节。

#### 前提条件

- 执行脚本的主机与需要导入镜像的所有主机之间能进行正常的SSH通信。
- 执行脚本的主机上已经安装expect、sshpass和scp软件包。

#### 操作步骤

**步骤1** 执行**vi** *worker-list.txt*命令,创建worker-list.txt配置文件。

# vi worker-list.txt

**步骤2** 配置worker-list.txt文件,worker-list.txt文件的模板如下所示。按l或lnsert进入编辑状态,添加节点信息。修改完成后,按**Esc**,并输入:**wq!**,保存修改。

# ip 192.168.128.16 192.168.128.17

#### 步骤3 上传并导入镜像。

如果使用的是containerd作为容器运行时,执行./containerd-upload.sh worker-list.txt huawei-csi.tar命令,按提示输入用户名和密码,上传并导入镜像。其中worker-list.txt文件是步骤2中配置的节点信息文件,huawei-csi.tar是步骤8中导出的镜像包。

# ./containerd-upload.sh worker-list.txt huawei-csi.tar

如果使用的是Docker作为容器运行时,执行./docker-upload.sh worker-list.txt huawei-csi.tar命令,按提示输入用户名和密码,上传并导入镜像。其中worker-list.txt文件是步骤2中配置的节点信息文件,huawei-csi.tar是步骤8中导出的镜像包。

# ./docker-upload.sh worker-list.txt huawei-csi.tar

#### 步骤4 检查导入镜像是否成功。

1. 如果脚本最后提示All images are uploaded successfully,则表示已成功将镜像导入到所有节点。

All images are uploaded successfully

2. 如果脚本最后回显是如下的列表,则表示列表中的节点镜像导入失败,此时请仔细检查脚本回显,确认失败原因。

List of nodes to which the image fails to be imported: 192.168.128.16 192.168.128.17

----结束

## 3.6 配置主机多路径

当您使用块存储,并且使用FC/iSCSI/NVMe over RoCE/NVMe over FC协议对存储进行访问时,推荐配置主机多路径,用于提升存储的链路可靠性。当前支持的多路径软件有原生多路径软件(DM-Multipath)和华为多路径软件(UltraPath和UltraPathNVMe)共三种多路径软件。

#### 注意事项

- 企业存储和分布式存储在不同OS类型下支持的主机多路径软件关系,请参考表
   2-3。
- 使用SAN存储时,不同的协议组网下,支持的多路径软件类型请参见表3-2。

#### 表 3-2 SAN 存储不同协议支持的多路径软件类型

| SAN存储组网<br>协议类型   | 不使用多路径<br>软件 | DM-<br>Multipath | UltraPath | UltraPath-<br>NVMe |
|-------------------|--------------|------------------|-----------|--------------------|
| VBS               |              | х                | Х         | х                  |
| iSCSI             | V            | V                | V         | V                  |
| FC                | V            | V                | V         | V                  |
| NVMe over<br>RoCE | V            | x                | x         | $\checkmark$       |

| SAN存储组网         | 不使用多路径 | DM-       | UltraPath | UltraPath-   |
|-----------------|--------|-----------|-----------|--------------|
| 协议类型            | 软件     | Multipath |           | NVMe         |
| NVMe over<br>FC | V      | х         | х         | $\checkmark$ |

### 3.6.1 安装原生多路径软件

原生多路径软件由主机系统自带,如果您需要安装,请参考本章节。

#### 3.6.1.1 安装多路径工具包

本章节介绍如何安装原生多路径工具包。

#### 前提条件

请确保Kubernetes集群的worker节点支持访问互联网(仅用于下载多路径工具包)。

#### 操作步骤

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的worker节点。

步骤2 根据不同的操作系统安装多路径工具包。

- CentOS:
  - yum install -y device-mapper-multipath
- SUSE:

zypper install -y multipath-tools

#### 步骤3 开启主机多路径服务。

CentOS:

/sbin/mpathconf --enable systemctl start multipathd.service systemctl enable multipathd.service systemctl restart multipathd.service

SUSE:

systemctl restart multipath-tools.service chkconfig multipathd on

步骤4 重复执行步骤1到步骤3,将多路径工具安装到所有worker节点。

----结束

#### 3.6.1.2 配置多路径服务

配置多路径是为了提升SAN存储的LUN的链路可靠性,如果多路径配置不当,会造成单条链路故障后I/O错误,导致Kubernetes集群管理的容器的文件系统或磁盘为只读或者故障,最终影响I/O下发。

#### 操作步骤

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的worker节点。

步骤2 执行vi /etc/multipath.conf命令,修改"multipath.conf"文件(如果文件不存在,请参考存储主机连通性指南进行配置或生成)。按l或lnsert进入编辑状态,修改相关参数。修改完成后,按Esc,并输入:wq!,保存修改。以Red Hat系统为例,进行介绍,其余请参考存储主机连通性指南。

#### 须知

负载均衡模式:进行业务读写时,主机到存储上所有控制器的IO的路径是一样的。详情可参考《华为SAN存储在Red Hat系统下的主机连通性指南》中的"配置多路径>常用概念"章节。

本端优选模式:进行业务读写时,主机下发IO到控制器时,由于不同存储之间存在业务链路距离差异,访问性能更优的存储。详情可参考《华为SAN存储在Red Hat系统下的主机连通性指南》中的"配置多路径>常用概念"章节。

如果是企业存储,且采用负载均衡模式,编辑多路径配置文件(/etc/multipath.conf),推荐在devices字段里添加如下内容,详情可参考《OceanStor Dorado在Red Hat下的主机连通性指南》和《OceanStor Dorado在SUSE下的主机连通性指南》。

```
defaults {
     user_friendly_names yes
     find_multipaths no
devices {
  device {
          vendor
                               "HUAWEI"
          product
                               "XSG1"
          path_grouping_policy
                                   multibus
          path_checker
                                 tur
          prio
                             const
          path_selector
                                "service-time 0"
          failback
                              immediate
          no_path_retry
                                 15
      }
```

如果是企业存储,且采用本端优选模式,编辑多路径配置文件(/etc/multipath.conf),推荐在devices字段里添加如下内容(详情可参考《华为SAN存储在Red Hat系统下的主机连通性指南》和《华为SAN存储在SUSE系统下的主机连通性指南》):

```
defaults {
     user_friendly_names yes
     find_multipaths no
}
devices {
  device {
         vendor
                            "HUAWEI"
                            "XSG1"
         product
         path_grouping_policy group_by_prio
         path_checker
                             tur
         prio
                          alua
         path_selector
                             "round-robin 0"
         failback
                           immediate
         no_path_retry
                              15
}
```

● 如果是分布式存储,编辑多路径配置文件(/etc/multipath.conf),推荐在devices字段里添加如下内容(针对不同的OS,配置会存在差异,详情可参考《FusionStorage 8.0.1 块存储基础业务配置指南 08》中应用服务器配置多路径(Red Hat/CentOS)):

```
user_friendly_names yes
     find_multipaths no
devices {
  device {
                               "Huawei"
          vendor
          product
                               "VBS fileIO"
          path_grouping_policy
                                  multibus
          path_checker
                                 tur
          prio
                             const
          path_selector
                                "service-time 0"
          failback
                              immediate
          no_path_retry
                                 "10"
```

步骤3 配置完成后,执行以下命令,重启multipathd服务。

systemctl restart multipathd.service

步骤4 重复执行步骤1到步骤3,为所有worker节点配置多路径服务。

----结束

#### 3.6.2 安装华为多路径软件

华为多路径软件由华为提供,如果您需要安装,请参考本章节。

#### 3.6.2.1 安装华为多路径工具

本章节介绍如何安装华为多路径工具包。

#### 前提条件

根据主机操作系统,组网和版本配套关系选择合适的华为多路径软件(企业用户登录 https://support.huawei.com/enterprise,运营商用户登录https://support.huawei.com,搜索"UltraPath"获取软件包和用户指南)。

#### 安装步骤

根据获取的华为多路径软件用户指南,安装华为多路径软件。

#### 配置步骤

根据获取的华为多路径软件用户指南,配置华为多路径软件。

#### □ 说明

对于多种多路径软件共存的场景,兼容性和配置方法请参考华为多路径软件用户指南。

## **4** 安装华为 CSI

本章节介绍如何安装华为CSI,推荐使用Helm安装华为CSI,如果需要手动安装华为CSI,请参考**4.2 手动安装华为CSI**。当前版本会保留这两种安装方式,计划在V3.2.0版本删除手动安装方式。

华为CSI的安装支持root用户和非root用户。使用非root用户安装华为CSI时,需要保证当前用户能够访问Kubernetes集群的API Server,配置非root用户访问Kubernetes集群请参考9.8 配置非root用户访问Kubernetes集群。

华为CSI必须在root用户权限下运行。

- 4.1 使用Helm安装华为CSI
- 4.2 手动安装华为CSI

## 4.1 使用 Helm 安装华为 CSI

本章节介绍如何使用Helm 3安装华为CSI。

## 4.1.1 安装 Helm

#### 前提条件

• 确保Kubernetes集群中的master节点可以访问Internet。

#### 操作步骤

步骤1 执行以下命令,下载Helm 3的安装脚本。

# curl -fsSL -o get\_helm.sh https://raw.githubusercontent.com/helm/helm/master/scripts/get-helm-3

步骤2 执行以下命令,修改Helm 3的安装脚本权限。

# chmod 700 get\_helm.sh

步骤3 根据Helm与Kubernetes版本配套关系确认需要安装的Helm版本,配套关系请参考 Helm Version Support Policy,执行以下命令,修改DESIRED\_VERSION环境变量为 需要安装的Helm版本,并执行安装命令。

# DESIRED\_VERSION=v3.9.0 ./get\_helm.sh

步骤4 执行以下命令,查看指定版本的Helm 3是否安装成功。

# helm version version:"v3.9.0", GitCommit:"7ceeda6c585217a19a1131663d8cd1f7d641b2a7", GitTreeState:"clean", GoVersion:"qo1.17.5"}

----结束

#### 4.1.2 安装华为 CSI

#### 组件说明

Helm在安装huawei-csi-controller时,将在指定命名空间的Deployment类型Pod中安装以下组件:

- huawei-csi-driver,华为CSI驱动
- Kubernetes External Provisioner, 用于提供卷
- Kubernetes External Attacher,用于挂载卷
- (可选)Kubernetes External Snapshotter,提供快照支持(作为CRD安装)
- (可选)Kubernetes External Resizer,用于扩容卷

Helm在安装huawei-csi-node时,将在指定命名空间的DaemonSet类型Pod中安装以下组件:

- huawei-csi-driver, 华为CSI驱动
- Kubernetes Node Registrar, 处理驱动程序注册

#### 前提条件

- huawei-csi镜像已制作完成并在所有worker节点上导入(或者通过镜像仓库在线 拉取)。
- 安装过程中可能涉及到sidecar镜像下载,需要Kubernetes集群的worker节点能够 访问外网。如果是内网环境,请通过其他方式获取相关镜像包并手动导入到所有 worker节点,镜像包列表请参考表4-23。
- 已向管理员获取Kubernetes集群中任意master节点的IP地址、登录帐号和密码。
- 已向管理员获取存储设备的用户名和密码。
- Kubernetes的所有节点与待接入的存储设备管理IP通信正常。
- Kubernetes的所有worker节点与待接入的存储设备业务IP通信正常(华为CSI使用ping命令判断)。
- Kubernetes的所有worker节点已安装对应协议所需要的软件客户端,如iSCSI客户端、NFS客户端等。
- 不同存储设备连接时支持的用户类型和要求请参考表4-21。
- 如果是多路径组网,请确保所有worker节点上已安装多路径软件,详细操作请参考3.6 配置主机多路径。

#### 操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。
- 步骤2 将Kubernetes CSI组件包中的"helm"目录拷贝到master节点的任意目录下。Helm工具路径请参见3.3 上传软件包组件。

**步骤3** 执行**kubectl create namespace** *huawei-csi*命令创建部署华为CSI的命名空间,huawei-csi为自定义的命名空间,如果华为CSI与Helm部署在相同的命名空间中,则可跳过该步骤,通过步骤7创建命名空间。

# kubectl create namespace huawei-csi

步骤4 执行cd helm/esdk命令,进入到helm/esdk的工作目录下。

# cd helm/esdk

步骤5 执行vi values.yaml命令,修改values.yaml文件。

# vi values.yaml

**步骤6** 配置values.yaml文件,参数说明请参考**表4-1**,无默认值的必选参数需要用户根据实际情况进行配置,有默认值的参数可不做修改。

如果需要配置多个后端存储,请在backends列表中并列添加。不同存储类型和协议的后端配置示例请参考表4-2。

```
# An array of storages with the access info
backends:
 storage: "oceanstor-nas"
  # support upper&lower characters, numeric and [-_].
  name: "nfs-155"
  urls:
    - "https://192.168.129.155:8088"
    - "StoragePool001"
  parameters:
   protocol: "nfs"
    portals:
     - "192.168.128.155"
  - storage: "fusionstorage-san"
  # support upper&lower characters, numeric and [-_].
  name: "iscsi-156"
  urls:
    - "https://192.168.129.156:8088"
  pools:
    - "StoragePool001"
  parameters:
    protocol: "iscsi"
    portals:
     - "192.168.128.160"
     - "192.168.128.161"
kubernetes:
 namespace: huawei-csi
images:
 # The image name and tag for the attacher, provisioner and registrar sidecars. These must match the
appropriate Kubernetes version.
 sidecar:
  csiAttacher: k8s.gcr.io/sig-storage/csi-attacher:v3.3.0
  csiProvisioner: k8s.gcr.io/sig-storage/csi-provisioner:v3.0.0
  csiResizer: k8s.gcr.io/sig-storage/csi-resizer:v1.3.0
  registrar: k8s.gcr.io/sig-storage/csi-node-driver-registrar:v2.3.0
  livenessProbe: k8s.gcr.io/sig-storage/livenessprobe:v2.5.0
  csiSnapshotter: k8s.gcr.io/sig-storage/csi-snapshotter:v4.2.1
  snapshotController: k8s.gcr.io/sig-storage/snapshot-controller:v4.2.1
 # The image name and tag for the Huawei CSI Service container
 # Replace the appropriate tag name
 huaweiCSIService: huawei-csi:3.0.0
# The CSI driver parameter configuration
 driverName: csi.huawei.com # It is strongly recommended not to modify this parameter
 endpoint: /csi/csi.sock # It is strongly recommended not to modify this parameter
 connectorThreads: 4
 volumeUseMultipath: true # Flag to enable or disable volume multipath access
```

```
scsiMultipathType: DM-multipath # Required, if volume-use-multipath is set to TRUE
 nvmeMultipathType: HW-UltraPath-NVMe # Required, if volume-use-multipath is set to TRUE
 scanVolumeTimeout: 3
 backendUpdateInterval: 60
 controllerLogging:
  module: file
  level: info
  fileDir: /var/log/huawei
  fileSize: 20M
  maxBackups: 9
 nodeLogging:
  module: file
  level: info
  fileDir: /var/log/huawei
  fileSize: 20M
  maxBackups: 9
huaweiCsiController:
 replicas: 1 # Currently, the value can only be set to 1.
# Default image pull policy for sidecar container images
sidecarImagePullPolicy: "IfNotPresent"
# Default image pull policy for Huawei plugin container images
huaweiImagePullPolicy: "IfNotPresent"
# Flag to enable or disable snapshot (Optional)
snapshot:
 enable: true
# Flag to enable or disable resize (Optional)
resizer:
 enable: true
```

#### 表 4-1 values.yaml 参数说明

| 参数               | 描述   | 必选参数 | 默认值 |
|------------------|--|------|-----|
| backends.storage | 存储设备类型。                                      | 是    | -   |
|                  | ● 企业SAN存储固定填写 oceanstor-san。                 |      |     |
|                  | 企业NAS存储固定填写 oceanstor-nas。                   |      |     |
|                  | 分布式SAN存储固定填写 fusionstorage-san。              |      |     |
|                  | 分布式NAS存储固定填写 fusionstorage-nas。              |      |     |
| backends.name    | 存储后端名称。支持大小写<br>字母、数字和[]。                    | 是    | -   |
|                  | 如果需要配置多个存储后<br>端,请保证存储后端名唯<br>一。             |      |     |
| backends.urls    | 存储设备的管理URL。参数<br>格式为列表。                      | 是    | -   |
|                  | 支持同一存储设备的一个或<br>多个管理URL(多控制<br>器),当前仅支持IPv4。 |      |     |

| 参数                                | 描述   | 必选参数 | 默认值   |
|-----------------------------------|--|------|---|
| backends.pools                    | 存储设备的存储池。参数格<br>式为列表。<br>支持同一存储设备上的一个<br>或多个存储池。   | 是    | -   |
| backends.paramet<br>ers.protocol  | 存储协议。参数格式为字符串。  • 企业SAN存储支持配置iscsi、fc、roce、fc-nvme • 企业NAS存储支持配置nfs • 分布式SAN存储支持配置iscsi、scsi • 分布式NAS存储支持配置nfs、dpc | 是    | -   |
| backends.paramet<br>ers.portals   | 业务端口。参数格式为字符串。 iscsi,roce协议支持配置多个端口,nfs协议仅支持配置一个端口,fc、fc-nvme、dpc协议无需配置业务端口,scsi协议的端口形式为字典格式,key为主机名称,value为IP地址。  | 条件必选 | -   |
| kubernetes.names<br>pace          | 命名空间,支持用户自定<br>义。<br>名称必须使用字母或者数字<br>开头和结尾,且包含字母、<br>数字或中划线(- )。例如:<br>my-name、123-abc。                            | 是    | huawei-csi  |
| images.sidecar.live<br>nessProbe  | <b>livenessprobe</b> sidecar镜<br>像。  | 是    | k8s.gcr.io/sig-<br>storage/<br>livenessprobe:v2.5<br>.0 |
| images.sidecar.csi<br>Provisioner | <b>csi-provisioner</b> sidecar镜<br>像。  | 是    | k8s.gcr.io/sig-<br>storage/csi-<br>provisioner:v3.0.0   |
| images.sidecar.csi<br>Attacher    | <b>csi-attacher</b> sidecar镜像。   | 是    | k8s.gcr.io/sig-<br>storage/csi-<br>attacher:v3.3.0      |
| images.sidecar.csi<br>Resizer     | <b>csi-resizer</b> sidecar镜像。  | 是    | k8s.gcr.io/sig-<br>storage/csi-<br>resizer:v1.3.0       |

| 参数                                    | 描述   | 必选参数   | 默认值   |
|---------------------------------------|--|--|---|
| images.sidecar.csi<br>Snapshotter     | <b>csi-snapshotter</b> sidecar镜<br>像。                                      | 是  | k8s.gcr.io/sig-<br>storage/csi-<br>snapshotter:v4.2.1               |
| images.sidecar.sna<br>pshotController | <b>snapshot-controller</b><br>sidecar镜像。                                   | 是  | k8s.gcr.io/sig-<br>storage/snapshot-<br>controller:v4.2.1           |
| images.sidecar.reg<br>istrar          | <b>csi-node-driver-registrar</b><br>sidecar镜像。                             | 是  | k8s.gcr.io/sig-<br>storage/csi-node-<br>driver-<br>registrar:v2.3.0 |
| images.huaweiCSI<br>Service           | huawei-csi镜像。  | 是  | -   |
| csi_driver.endpoint                   | 通信端点,建议不修改。  | 是  | /csi/csi.sock   |
| csi_driver.driverNa<br>me             | 注册的驱动名称,建议不修 改。  | 是  | csi.huawei.com  |
| csi_driver.connect<br>orThreads       | 最大并发扫盘/卸盘数。参数<br>格式为整型,支持范围为<br>1~10。                                      | 是  | 4   |
| csi_driver.volumeU<br>seMultipath     | 是否使用多路径软件。参数<br>格式为布尔值。  | 是  | true  |
| csi_driver.scsiMulti<br>pathType      | fc/iscsi使用的多路径软件。<br>支持配置如下参数: DM-multipath HW-UltraPath HW-UltraPath-NVMe | 当<br>volumeU<br>seMultip<br>ath为<br>TRUE时<br>必填。 | DM-multipath  |
| csi_driver.nvmeMu<br>ltipathType      | roce/fc-nvme 使用的多路径<br>软件。仅支持配置HW-<br>UltraPath-NVMe。                      | 当<br>volumeU<br>seMultip<br>ath为<br>TRUE时<br>必填。 | HW-UltraPath-<br>NVMe   |
| csi_driver.scanVolu<br>meTimeout      | 在主机上使用DM-multipath<br>多路径时,等待多路径聚合<br>的超时时间,支持范围为<br>1~600,单位秒。            | 是  | 3   |
| csi_driver.backend<br>UpdateInterval  | 后端能力的更新时间间隔,<br>支持范围60~600,单位秒。  | 是  | 60  |

| 参数  | 描述                                  | 必选参数 | 默认值             |
|---|-------------------------------------|------|-----------------|
| csi_driver.controlle<br>rLogging.module         | controller日志记录类型。支<br>持配置如下参数:      | 是    | file            |
|   | • file                              |      |                 |
|   | • console                           |      |                 |
| csi_driver.controlle<br>rLogging.level          | controller日志输出级别。支<br>持配置如下参数:      | 是    | info            |
|   | • debug                             |      |                 |
|   | • info                              |      |                 |
|   | warning                             |      |                 |
|   | • error                             |      |                 |
|   | • fatal                             |      |                 |
| csi_driver.controll<br>erLogging.fileDir        | controller日志在file输出模<br>式下的日志目录。    | 是    | /var/log/huawei |
| csi_driver.controll<br>erLogging.fileSize       | controller日志在file输出模<br>式下单个日志文件大小。 | 是    | 20M             |
| csi_driver.controlle<br>rLogging.maxBack<br>ups | controller日志在file输出模<br>式下日志文件备份上限。 | 是    | 9               |
| csi_driver.nodeLog<br>ging.module               | node日志记录类型。支持配<br>置如下参数:            | 是    | file            |
|   | • file                              |      |                 |
|   | • console                           |      |                 |
| csi_driver.nodeLog<br>ging.level                | node日志输出级别。支持配<br>置如下参数:            | 是    | info            |
|   | • debug                             |      |                 |
|   | • info                              |      |                 |
|   | warning                             |      |                 |
|   | • error                             |      |                 |
|   | • fatal                             |      |                 |
| csi_driver.nodeLog<br>ging.fileDir              | node日志在file输出模式下<br>的日志目录。          | 是    | /var/log/huawei |
| csi_driver.nodeLog<br>ging.fileSize             | node日志在file输出模式下<br>单个日志文件大小。       | 是    | 20M             |
| csi_driver.nodeLog<br>ging.maxBackups           | node日志在file输出模式下<br>日志文件备份上限。       | 是    | 9               |
| sideCarImagePull<br>Policy                      | sidecar镜像的拉取策略。                     | 是    | IfNotPresent    |

| 参数                               | 描述   | 必选参数 | 默认值          |
|----------------------------------|--|------|--------------|
| huweiCSIImagePu<br>llPolicy      | huawei-csi镜像的拉取策<br>略。   | 是    | IfNotPresent |
| huaweiCsiControll<br>er.replicas | huawei-csi-controller的个<br>数,仅支持配置为1。  | 是    | 1            |
| snapshot.enable                  | 是否开启快照特性。 Kubernetes卷快照CRD放置在"helm/crd"目录下,当该参数配置为true时,在执行helm install命令时将安装快照CRD资源。 <b>须知</b> Helm不负责CRD生命周期。如果您需要更新CRD版本,则必须手动更新。 | 是    | true         |
| resizer.enable                   | 是否开启扩容特性。  | 是    | true         |

#### 表 4-2 backends 配置参考模板

| 存储<br>类型        | 协议    | 后端配置参考模板  | 备注                                    |
|-----------------|-------|---|---------------------------------------|
| 企业<br>存储<br>SAN | iSCSI | backends: - storage: "oceanstor-san" # support upper&lower characters, numeric and []. name: "iscsi-155" urls: - "https://192.168.129.155:8088" - "https://192.168.129.156:8088" pools: - "StoragePool001" parameters: protocol: "iscsi" portals: - "192.168.128.120" - "192.168.128.121" | 请确保所有<br>worker节点<br>已安装iSCSI<br>客户端。 |
| 企业<br>存储<br>SAN | FC    | backends: - storage: "oceanstor-san" # support upper&lower characters, numeric and []. name: "fc-155" urls: - "https://192.168.129.155:8088" - "https://192.168.129.156:8088" pools: - "StoragePool001" parameters: protocol: "fc"  | -                                     |

| 存储<br>类型             | 协议                | 后端配置参考模板  | 备注  |
|----------------------|-------------------|---|---|
| 企业<br>存储<br>NAS      | NFS               | backends: - storage: "oceanstor-nas" # support upper&lower characters, numeric and []. name: "nfs-155" urls: - "https://192.168.129.155:8088" - "https://192.168.129.156:8088" pools: - "StoragePool001" parameters: protocol: "nfs" portals: - "192.168.128.155"                         | 请确保所有<br>worker节点<br>已安装NFS<br>客户端工<br>具。                 |
| 企业<br>存储<br>SAN      | NVMe over<br>RoCE | backends: - storage: "oceanstor-san" # support upper&lower characters, numeric and []. name: "roce-155" urls: - "https://192.168.129.155:8088" - "https://192.168.129.156:8088" pools: - "StoragePool001" parameters: protocol: "roce" portals: - "192.168.128.120" - "192.168.128.121"   | 请确保所有<br>worker节点<br>已安装<br>nvme-cli工<br>具,且版本<br>不低于1.9。 |
| 企业<br>存储<br>SAN      | NVMe over<br>FC   | backends: - storage: "oceanstor-san" # support upper&lower characters, numeric and []. name: "fc-nvme-155" urls: - "https://192.168.129.155:8088" - "https://192.168.129.156:8088" pools: - "StoragePool001" parameters: protocol: "fc-nvme"  | 请确保所有<br>worker节点<br>已安装<br>nvme-cli工<br>具,且版本<br>不低于1.9。 |
| 分布<br>式存<br>储<br>SAN | SCSI              | backends: - storage: "fusionstorage-san" # support upper&lower characters, numeric and []. name: "scsi-155" urls: - "https://192.168.129.155:28443" pools: - "StoragePool001" parameters: protocol: "scsi" portals: - {"hostname01": "192.168.125.21"} - {"hostname02": "192.168.125.22"} | 请确保所有<br>worker节点<br>已安装分布<br>式存储VBS客<br>户端。              |

| 存储<br>类型             | 协议    | 后端配置参考模板  | 备注  |
|----------------------|-------|---|---|
| 分布<br>式存<br>储<br>SAN | iSCSI | backends: - storage: "fusionstorage-san" # support upper&lower characters, numeric and []. name: "iscsi-155" urls: - "https://192.168.129.155:28443" - "https://192.168.129.156:28443" pools: - "StoragePool001" parameters: protocol: "iscsi" portals: - "192.168.128.120" - "192.168.128.121" | 请确保所有<br>worker节点<br>已安装iSCSI<br>客户端。                 |
| 分布<br>式存<br>储<br>NAS | NFS   | backends: - storage: "fusionstorage-nas" # support upper&lower characters, numeric and []. name: "nfs-155" urls: - "https://192.168.129.155:28443" - "https://192.168.129.156:28443" pools: - "StoragePool001" parameters: protocol: "nfs" portals: - "192.168.128.120"                         | 请确保所有<br>worker节点<br>已安装NFS<br>客户端工<br>具。             |
| 分布<br>式存<br>储<br>NAS | DPC   | backends: - storage: "fusionstorage-nas" # support upper&lower characters, numeric and []. name: "dpc-155" urls: - "https://192.168.129.155:28443" - "https://192.168.129.156:28443" pools: - "StoragePool001" parameters: protocol: "dpc"  | 请确保所有<br>worker节点<br>均已在待接<br>入存储上添<br>加为DPC计<br>算节点。 |

**步骤7** 执行helm install helm-release-name ./ -n huawei-csi --create-namespace命令安 装华为CSI。

其中,helm-release-name为自定义的chart名称,./表示使用当前目录下的Helm工程,huawei-csi为自定义的命名空间。

# helm install helm-huawei-csi ./ -n huawei-csi --create-namespace

NAME: helm-huawei-csi

LAST DEPLOYED: Wed Jun 8 11:50:28 2022

NAMESPACE: huawei-csi STATUS: deployed REVISION: 1 TEST SUITE: None

步骤8 将华为CSI组件包中的secretGenerate工具拷贝到master节点的任意目录下,工具路径参见3.3 上传软件包组件。

步骤9 使用加密工具填写存储设备的用户名和密码。

1. 执行**chmod +x secretGenerate**命令,给secretGenerate添加可执行权限。 # chmod +x secretGenerate  执行./secretGenerate --namespace=huawei-csi --logFileDir=/var/log/huawei 命令,运行secretGenerate工具,namespace的参数可修改为实际使用的命名空 间,不使用该参数则默认使用huawei-csi命名空间,logFileDir参数可修改为实际 使用的日志目录,不使用该参数则默认使用/var/log/huawei目录,根据界面提示 输入需要配置backend的编号。Configured为false时表示这个backend尚未配置, 为true时表示这个backend已配置。

# ./secretGenerate huawei-csi

Getting backend configuration information.....

Number Configured BackendName Urls

1 false iscsi-25 [https://192.168.125.25:8088] 2 false nfs-26 [https://192.168.125.26:8088] 3 false dpc-27 [https://192.168.125.27:8088]

Diagram and the backward musches to a self-run. (Fates levit) to a sith 2

Please enter the backend number to configure (Enter 'exit' to exit): 3

3. 根据界面提示输入用户名和密码来创建secret对象。

Name:strage-backend-03

Urls:[https://192.168.125.27:8088]

Please enter this backend user name: admin

Please enter this backend password:

Verifying user name and password. Please wait.....

The acount information of the backend strage-backend-03 has been configured successfully.

4. 配置完成后,输入exit退出并保存配置。

Please enter the backend number to configure (Enter 'exit' to exit): *exit* Saving configuration. Please wait......

The configuration is saved successfully.

5. 使用kubectl get secret -n huawei-csi | grep huawei-csi-secret命令检查 secret对象是否已创建成功。

# kubectl -n huawei-csi get secret huawei-csi-secret

NAME TYPE DATA AGE huawei-csi-secret Opaque 1 8d

步骤10 完成huawei-csi服务部署后,可执行kubectl get pod -n huawei-csi命令检查服务是否启动。

# kubectl get pod -n huawei-csi

NAME READY STATUS RESTARTS AGE huawei-csi-controller-764bd64c97-kr2nm 7/7 Running 0 144m huawei-csi-node-7m48s 3/3 Running 0 144m

----结束

## 4.2 手动安装华为 CSI

本章节主要介绍如何手动安装华为CSI。

## 4.2.1 对接企业存储配置

本章节描述如何配置huawei-csi插件对接华为企业存储。

#### 限制说明

当使用同一SAN存储对接Kubernetes时,不支持同一Worker节点配置多个数据协议(iSCSI、FC、NVMe over RoCE、NVMe over FC)。

#### 4.2.1.1 通过 iSCSI 对接企业存储 SAN 配置

当您需要通过iSCSI对接企业存储SAN配置时,执行此操作。

#### 前提条件

- Kubernetes的所有worker节点已安装iSCSI客户端。
- Kubernetes的所有节点与待接入的存储设备管理IP通信正常(华为CSI使用ping命令判断)。
- Kubernetes的所有worker节点与待接入的存储设备业务IP通信正常(华为CSI使用ping命令判断)。
- 如果是多路径组网,请确保所有worker节点上已安装多路径软件,详细操作请参考3.6 配置主机多路径。
- 已向管理员获取Kubernetes集群中任意master节点的IP地址、登录帐号和密码。

#### 操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。
- 步骤2 执行**vi** *huawei-csi-configmap.yaml*命令,创建huawei-csi-configmap.yaml文件。 # vi huawei-csi-configmap.yaml
- 步骤3 配置huawei-csi-configmap.yaml文件, huawei-csi-configmap.yaml文件的模板如回显所示(您也可以参考软件包中deploy/huawei-csi-configmap/huawei-csi-configmap-oceanstor-iscsi.yaml示例文件 )。请根据实际情况设置相关参数,并保存为yaml格式的文件,详细请参考表4-3。

#### 表 4-3 配置项描述

| 配置项                      | 参数格式 | 描述                      | 备注   |
|--------------------------|------|-------------------------|--|
| data."csi.json".backends | 列表   | 必填,待接入<br>后端存储设备<br>列表。 | 配置的后端存储设备数量没有限制。<br>单个后端存储设备支持配置的字段,请参考表4-4。 |

表 4-4 后端存储设备配置项描述

| 配置项        | 参数格式 | 描述                             | 备注   |
|------------|------|--------------------------------|--|
| storage    | 字符串  | 必填,待接入存储<br>设备的类型。             | 对接企业存储SAN场<br>景,固定填写<br>"oceanstor-san"。   |
| name       | 字符串  | 存储后端名。                         | 自定义字符串,支持大小写字母、数字、中划线组合。<br><b>说明</b> 如果需要配置多个存储后端,请保证存储后端名唯一。   |
| urls       | 列表   | 必填,待接入的存储设备的管理URL。             | 支持同一存储设备的一个或多个管理URL,用逗号分割,当前仅支持IPv4。例如: https://192.168.125.20:8088  |
| pools      | 列表   | 必填,被使用的待<br>接入存储设备上的<br>存储池名称。 | 支持同一存储设备上的<br>一个或多个存储池, 用逗<br>号分割。<br>可通过登录<br>DeviceManager获取支<br>持块存储服务的存储<br>池。   |
| parameters | 字典   | 必填,iSCSI场景的可变参数。               | iSCSI场景protocol参数<br>固定填写"iscsi"。<br>portals参数填写待接入<br>存储设备的iSCSI业务IP<br>地址,多个业务IP地址<br>请用逗号分割。<br>iSCSI业务IP地址可通过<br>登录DeviceManager获<br>取。以OceanStor<br>Dorado 6.x系列为例:<br>在DeviceManager管理<br>界面,选择"服务 > 网<br>络 > 逻辑端口",获取<br>数据协议类型为iSCSI的<br>IP(其它系列请参照相<br>应的操作说明进行获<br>取)。 |

**步骤4** 执行**kubectl create -f** *huawei-csi-configmap.yaml*命令,创建huawei-csi-configmap。

# kubectl create -f huawei-csi-configmap.yaml

步骤5 创建完成后,执行kubectl get configmap -n huawei-csi | grep huawei-csi | configmap命令,查看是否创建成功。显示如下回显时,则表示创建成功。

```
# kubectl get configmap -n huawei-csi | grep huawei-csi-configmap
huawei-csi-configmap 1 5s
```

----结束

#### 4.2.1.2 通过 FC 对接企业存储 SAN 配置

当您需要通过FC对接企业存储SAN配置时,请执行此操作。

#### 限制条件

当您需要通过FC对接企业存储SAN配置时,需要保证主机侧没有盘符残留,如果有盘符残留,请参考10.4 集群中worker节点宕机并恢复后,Pod完成failover,但是Pod所在源主机出现盘符残留进行盘符清理。

#### 前提条件

- Kubernetes的所有节点与待接入的存储设备管理IP通信正常(华为CSI使用ping命令判断)。
- Kubernetes的所有worker节点能通过FC与待接入的存储设备通信(华为CSI使用ping命令判断)。
- 如果是多路径组网,请确保Kubernetes的所有worker节点上已安装多路径软件, 详细请参见**3.6 配置主机多路径**。
- 已向管理员获取Kubernetes集群中任意master节点的IP地址、登录帐号和密码。

#### 操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意master节点。
- 步骤2 执行**vi** *huawei-csi-configmap.yaml*命令,创建huawei-csi-configmap.yaml文件。 # vi huawei-csi-configmap.yaml
- 步骤3 配置huawei-csi-configmap.yaml文件, huawei-csi-configmap.yaml文件的模板如回显所示(您也可以参考软件包中deploy/huawei-csi-configmap/huawei-csi-configmap-oceanstor-fc.yaml示例文件 )。请根据实际情况设置相关参数,并保存为yaml格式的文件,详细请参考表4-5。

```
"parameters": {"protocol": "fc"}
}
]
}
```

#### 表 4-5 配置项描述

| 配置项                      | 参数格式 | 描述                      | 备注                                       |
|--------------------------|------|-------------------------|--|
| data."csi.json".backends | 列表   | 必填,待接入后<br>端存储设备列<br>表。 | 配置的后端存储设备数量没有限制。单个后端存储设备支持配置的字段,请参考表4-6。 |

#### 表 4-6 后端存储设备配置项描述

| 配置项        | 参数格式 | 描述                             | 备注   |
|------------|------|--------------------------------|--|
| storage    | 字符串  | 必填,待接入存储设<br>备的类型。             | 对接企业存储SAN场<br>景,固定填写<br>"oceanstor-san"。   |
| name       | 字符串  | 存储后端名。                         | 自定义字符串,支持大小写字母、数字、中划线组合。<br><b>说明</b> 如果需要配置多个存储后端,请保证存储后端名唯一。   |
| urls       | 列表   | 必填,待接入的存储<br>设备的管理URL。         | 支持同一存储设备的一个或多个管理URL,用逗号分割,当前仅支持IPv4。例如:https://192.168.125.20:8088说明 1个存储设备有多个控制器,每个控制器有一个管理URL,所以1个存储设备存在多个管理URL。 |
| pools      | 列表   | 必填,被使用的待接<br>入存储设备上的存储<br>池名称。 | 支持同一存储设备上的<br>一个或多个存储池, 用逗<br>号分割。<br>可通过登录<br>DeviceManager获取支<br>持块存储服务的存储<br>池。                                 |
| parameters | 字典   | 必填,FC场景的可<br>变参数。              | FC场景protocol参数固定填写"fc"。  |

**步骤4** 执行**kubectl create -f** *huawei-csi-configmap.yaml*命令,创建huawei-csi-configmap。

# kubectl create -f huawei-csi-configmap.yaml

步骤5 创建完成后,执行kubectl get configmap -n huawei-csi | grep huawei-csi | configmap命令,查看是否创建成功。显示如下回显时,则表示创建成功。

```
# kubectl get configmap -n huawei-csi | grep huawei-csi-configmap
huawei-csi-configmap 1 5s
```

----结束

# 4.2.1.3 通过 NFS 对接企业存储 NAS 配置

当您需要通过NFS对接企业存储NAS配置时,执行此操作。

# 前提条件

- Kubernetes的所有worker节点已安装NFS客户端工具。
- Kubernetes的所有节点与待接入的存储设备管理IP通信正常(华为CSI使用ping命令判断)。
- Kubernetes的所有worker节点与待接入存储设备的NFS逻辑端口通信正常(华为 CSI使用ping命令判断)。
- 已向管理员获取Kubernetes集群中任意master节点的IP地址、登录帐号和密码。

# 操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。
- **步骤2** 执行**vi** *huawei-csi-configmap.yaml*命令,创建huawei-csi-configmap.yaml文件。 # vi huawei-csi-configmap.yaml
- **步骤3** 配置huawei-csi-configmap.yaml文件, huawei-csi-configmap.yaml文件的模板如回显所示(您也可以参考软件包中deploy/huawei-csi-configmap/huawei-csi-configmap-oceanstor-nfs.yaml示例文件 )。请根据实际情况设置相关参数,并保存为yaml格式的文件,详细请参考表4-7。

# 表 4-7 配置项描述

| 配置项                      | 参数格式 | 描述                      | 备注   |
|--------------------------|------|-------------------------|--|
| data."csi.json".backends | 列表   | 必填,待接入后<br>端存储设备列<br>表。 | 配置的后端存储设备<br>数量没有限制。<br>单个后端存储设备支<br>持配置的字段,请参<br>考表4-8。 |

# 表 4-8 后端存储设备配置项描述

| 配置项     | 参数格式 | 描述                     | 备注   |
|---------|------|------------------------|--|
| storage | 字符串  | 必填,待接入存储<br>设备的类型。     | 对接企业存储NAS场景<br>固定填写"oceanstor-<br>nas"。  |
| name    | 字符串  | 存储后端名。                 | 自定义字符串,支持大<br>小写字母、数字、中划<br>线组合。<br><b>说明</b><br>如果需要配置多个存储后<br>端,请保证存储后端名唯<br>一。                                  |
| urls    | 列表   | 必填,待接入的存储设备的管理URL。     | 支持同一存储设备的一个或多个管理URL,用逗号分割,当前仅支持IPv4。例如:https://192.168.125.20:8088说明 1个存储设备有多个控制器,每个控制器有一个管理URL,所以1个存储设备存在多个管理URL。 |
| pools   | 列表   | 必填,被使用的待接入存储设备上的存储池名称。 | 支持同一存储设备上的<br>一个或多个存储池, 用逗<br>号分割。<br>可通过登录<br>DeviceManager获取支<br>持文件存储服务的存储<br>池。                                |

| 配置项        | 参数格式 | 描述                 | 备注  |
|------------|------|--------------------|---|
| parameters | 字典   | 必填,NFS场景的可<br>变参数。 | protocol参数固定填写<br>"nfs"。  |
|            |      |                    | portals:指定存储的逻<br>辑端口IP或者DNS<br>Zone,只支持配置一<br>个。   |
|            |      |                    | 逻辑端口IP地址可通过<br>登录DeviceManager获<br>取。以OceanStor<br>Dorado 6.x系列为例:<br>在DeviceManager管理<br>界面,选择"服务 > 网<br>络 > 逻辑端口",获取<br>数据协议类型为NFS的 |
|            |      |                    | IP(其它系列请参照相<br>应的操作说明进行获<br>取)。   |

**步骤4** 执行**kubectl create -f** *huawei-csi-configmap.yaml*命令,创建huawei-csi-configmap。

# kubectl create -f huawei-csi-configmap.yaml

步骤5 创建完成后,执行kubectl get configmap -n huawei-csi | grep huawei-csi | configmap命令,查看是否创建成功。显示如下回显时,则表示创建成功。

# kubectl get configmap -n huawei-csi | grep huawei-csi-configmap huawei-csi-configmap 1 5s

----结束

# 4.2.1.4 通过 NVMe over RoCE 对接企业存储 SAN 配置

当您需要通过NVMe over RoCE对接企业存储SAN配置时,执行此操作。

## 前提条件

- Kubernetes的所有节点与待接入的存储设备管理IP通信正常(华为CSI使用ping命令判断)。
- Kubernetes的所有worker节点与待接入的存储设备业务IP通信正常(华为CSI使用ping命令判断)。
- Kubernetes的所有worker节点已安装nvme-cli工具,且版本不低于1.9。
- 已向管理员获取Kubernetes集群中任意master节点的IP地址、登录帐号和密码。
- 如果是多路径组网,请确保Kubernetes的所有worker节点上已安装多路径软件, 详细请参见**3.6 配置主机多路径**。

## 操作步骤

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。

- **步骤2** 执行**vi** *huawei-csi-configmap.yaml*命令,创建huawei-csi-configmap.yaml文件。
  # vi huawei-csi-configmap.yaml
- 步骤3 配置huawei-csi-configmap.yaml文件, huawei-csi-configmap.yaml文件的模板如回显所示(您也可以参考软件包中deploy/huawei-csi-configmap/huawei-csi-configmap-oceanstor-roce.yaml示例文件 )。请根据实际情况设置相关参数,并保存为yaml格式的文件,详细请参考表4-9。

#### 表 4-9 配置项描述

| 配置项                          | 参数格式 | 描述                      | 备注  |
|------------------------------|------|-------------------------|---|
| data."csi.json".b<br>ackends | 列表   | 必填,待接入<br>后端存储设备<br>列表。 | 配置的后端存储设备数量没有限制。<br>单个后端存储设备支持配置的字段,请参考表4-10。 |

#### 表 4-10 后端存储设备配置项描述

| 配置项         | 参数<br>格式 | 描述                         | 备注  |
|-------------|----------|----------------------------|---|
| stor<br>age | 字符串      | 必填,待接入存<br>储设备的类型。         | 对接企业存储SAN场景固定填写"oceanstor-<br>san"。   |
| na<br>me    | 字符串      | 存储后端名。                     | 自定义字符串,支持大小写字母、数字、中划线组合。<br><b>说明</b><br>如果需要配置多个存储后端,请保证存储后端名唯一。             |
| urls        | 列表       | 必填,待接入的<br>存储设备的管理<br>URL。 | 支持同一存储设备的一个或多个管理URL,用逗号分割,当前仅支持IPv4。例如: https://<br>192.168.125.20:8088<br>说明 |
|             |          |                            | 1个存储设备有多个控制器,每个控制器有一个管理<br>URL,所以1个存储设备存在多个管理URL。                             |

| 配置项                    | 参数格式 | 描述                                 | 备注  |
|------------------------|------|------------------------------------|---|
| poo<br>ls              | 列表   | 必填,被使用的<br>待接入存储设备<br>上的存储池名<br>称。 | 支持同一存储设备上的一个或多个存储池, 用逗号分割。<br>可通过登录DeviceManager获取存储池。  |
| par<br>am<br>eter<br>s | 字典   | 必填,NVMe<br>over RoCE场景<br>的可变参数。   | NVMe over RoCE场景protocol参数固定填写 "roce"。 portals参数填写存储的数据协议类型为NVMe over RoCE的逻辑端口IP,用逗号分割。逻辑端口IP地址可通过登录DeviceManager获取。以OceanStor Dorado 6.x系列为例:在 DeviceManager管理界面,选择"服务 > 网络 > 逻辑端口",获取数据协议类型为NVMe over RoCE的IP(其它系列请参照相应的操作说明进行获取)。 |

**步骤4** 执行**kubectl create -f** *huawei-csi-configmap.yaml*命令,创建huawei-csi-configmap。

# kubectl create -f huawei-csi-configmap.yaml

步骤5 创建完成后,执行kubectl get configmap -n huawei-csi | grep huawei-csi | configmap命令,查看是否创建成功。显示如下回显时,则表示创建成功。

# kubectl get configmap -n huawei-csi | grep huawei-csi-configmap huawei-csi-configmap 1 5s

----结束

# 4.2.1.5 通过 NVMe over FC 对接企业存储 SAN 配置

当您需要通过NVMe over FC对接企业存储SAN配置时,请执行此操作。

#### 限制条件

当您需要通过NVMe over FC对接企业存储SAN配置时,需要保证主机侧没有盘符残留,如果有盘符残留,请参考10.4 集群中worker节点宕机并恢复后,Pod完成failover,但是Pod所在源主机出现盘符残留进行盘符清理。

# 前提条件

- Kubernetes的所有节点与待接入的存储设备管理IP通信正常(华为CSI使用ping命令判断)。
- Kubernetes的所有worker节点能通过NVMe over FC与待接入的存储设备通信 (华为CSI使用ping命令判断)。
- Kubernetes的所有worker节点已安装nvme-cli工具,且版本不低于1.9。
- 如果是多路径组网,请确保Kubernetes的所有worker节点上已安装多路径软件, 详细请参见**3.6 配置主机多路径**。
- 已向管理员获取Kubernetes集群中任意master节点的IP地址、登录帐号和密码。

# 操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。
- **步骤2** 执行**vi** *huawei-csi-configmap.yaml*命令,创建huawei-csi-configmap.yaml文件。
  # vi huawei-csi-configmap.yaml
- 步骤3 配置huawei-csi-configmap.yaml文件, huawei-csi-configmap.yaml文件的模板如回显所示(您也可以参考软件包中deploy/huawei-csi-configmap/huawei-csi-configmap-oceanstor-fc-nvme.yaml示例文件 )。请根据实际情况设置相关参数,并保存为yaml格式的文件,详细请参考表4-11。

#### 表 4-11 配置项描述

| 配置项          | 参数<br>格式 | 描述     | 备注                                 |
|--------------|----------|--------|------------------------------------|
| data."csi.js | 列表       | 必填,待接入 | 配置的后端存储设备数量没有限制。                   |
| on".backe    |          | 后端存储设备 | 单个后端存储设备支持配置的字段,请参考 <mark>表</mark> |
| nds          |          | 列表。    | 4-12。                              |

#### 表 4-12 后端存储设备配置项描述

| 配置项         | 参数格式 | 描述                 | 备注  |
|-------------|------|--------------------|---|
| stor<br>age | 字符串  | 必填,待接入存储<br>设备的类型。 | 对接企业存储SAN场景,固定填写"oceanstor-<br>san"。                              |
| nam<br>e    | 字符串  | 存储后端名。             | 自定义字符串,支持大小写字母、数字、中划线组合。<br><b>说明</b><br>如果需要配置多个存储后端,请保证存储后端名唯一。 |

| 配置项                | 参数格式   | 描述                             | 备注  |
|--------------------|--------|--------------------------------|---|
| urls               | 列表     | 必填,待接入的存储设备的管理URL。             | 支持同一存储设备的一个或多个管理URL,用逗号分割,当前仅支持IPv4。例如: https://192.168.125.20:8088  说明 1个存储设备有多个控制器,每个控制器有一个管理URL,所以1个存储设备存在多个管理URL。 |
| pool<br>s          | 列<br>表 | 必填,被使用的待<br>接入存储设备上的<br>存储池名称。 | 支持同一存储设备上的一个或多个存储池, 用逗号分割。<br>可通过登录DeviceManager获取支持块存储服务的存储池。  |
| para<br>met<br>ers | 字<br>典 | 必填,NVMe over<br>FC场景的可变参<br>数。 | NVMe over FC场景protocol参数固定填写"fc-nvme"。  |

**步骤4** 执行**kubectl create -f** *huawei-csi-configmap.yaml*命令,创建huawei-csi-configmap。

# kubectl create -f huawei-csi-configmap.yaml

步骤5 创建完成后,执行kubectl get configmap -n huawei-csi | grep huawei-csi | configmap命令,查看是否创建成功。显示如下回显时,则表示创建成功。

# kubectl get configmap -n huawei-csi | grep huawei-csi-configmap huawei-csi-configmap 1 5s

----结束

# 4.2.2 对接分布式存储配置

本章节描述如何配置huawei-csi插件对接华为分布式存储。

## 限制说明

当使用同一SAN存储对接Kubernetes时,不支持同一Worker节点配置多个数据协议(SCSI、iSCSI)。

# 4.2.2.1 通过 SCSI 对接分布式存储 SAN 配置

当您需要通过SCSI对接分布式存储SAN配置,执行此操作。

# 前提条件

- Kubernetes的所有worker节点已安装分布式存储VBS客户端。
- Kubernetes的所有节点与待接入的存储设备管理IP通信正常(华为CSI使用ping命令判断)。
- 已向管理员获取Kubernetes集群中任意master节点的IP地址、登录帐号和密码。

# 操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。
- 步骤2 执行**vi** *huawei-csi-configmap.yaml*命令,创建huawei-csi-configmap.yaml文件。 # vi huawei-csi-configmap.yaml
- 步骤3 配置huawei-csi-configmap.yaml文件,huawei-csi-configmap.yaml文件的模板如回显所示(您也可以参考软件包中deploy/huawei-csi-configmap/huawei-csi-configmap-fusionstorage-scsi.yaml示例文件)。请根据实际情况设置相关参数,并保存为yaml格式的文件,详细请参考表4-13。

#### 表 4-13 配置项描述

| 配置项                      | 参数格式 | 描述                  | 备注  |
|--------------------------|------|---------------------|---|
| data."csi.json".backends | 列表   | 必填,待接入后端<br>存储设备列表。 | 配置的后端存储设备数量没有限制。单个后端存储设备支持配置的字段,请参考表4-14。 |

#### 表 4-14 后端存储设备配置项描述

| 配置项     | 参数格式 | 描述                 | 备注  |
|---------|------|--------------------|---|
| storage | 字符串  | 必填,待接入存<br>储设备的类型。 | 对接分布式存储SAN场景固定填写"fusionstorage-san"。      |
| name    | 字符串  | 存储后端名。             | 自定义字符串,支持大小写字<br>母、数字、中划线组合。              |
|         |      |                    | <b>说明</b><br>如果需要配置多个存储后端,请保<br>证存储后端名唯一。 |

| 配置项        | 参数格式 | 描述                                 | 备注  |
|------------|------|------------------------------------|---|
| urls       | 列表   | 必填,待接入的<br>存储设备的管理<br>URL。         | FusionStorage的管理URL, 只支持配置一个。   |
| pools      | 列表   | 必填,被使用的<br>待接入存储设备<br>上的存储池名<br>称。 | 支持同一存储设备上的一个或多<br>个存储池, 用逗号分割。<br>可通过登录DeviceManager获<br>取存储池。  |
| parameters | 字典   | 必填,SCSI场景<br>的可变参数。                | protocol参数固定填写<br>"scsi"。   |
|            |      |                                    | portals参数填写主机名称和VBS<br>节点IP Pair列表,参数格式为<br>[{"hostname":"*.*.*.*"}],其中<br>hostname为对应的worker节点<br>主机名称,"*.*.*."为分布式存储<br>块客户端对应的管理IP地址(当<br>前仅支持IPv4)。 |
|            |      |                                    | 如果存在多个worker节点,则<br>相应地以字典格式配置多个,以<br>逗号隔开。   |
|            |      |                                    | 例如示例中hostname01为<br>Kubernetes中Worker节点的主<br>机名称,192.168.125.21为该<br>Kubernetes中Worker节点创建<br>VBS后,在VBS中呈现的管理<br>IP。                                    |

**步骤4** 执行**kubectl create -f** *huawei-csi-configmap.yaml*命令,创建huawei-csi-configmap。

# kubectl create -f huawei-csi-configmap.yaml

# kubectl get configmap -n huawei-csi | grep huawei-csi-configmap huawei-csi-configmap 1 5s

----结束

# 4.2.2.2 通过 iSCSI 对接分布式存储 SAN 配置

当您需要通过iSCSI对接分布式存储SAN配置时,请执行此操作。

# 前提条件

- Kubernetes的所有worker节点已安装iSCSI客户端。
- Kubernetes的所有节点与待接入的存储设备管理IP通信正常(华为CSI使用ping命令判断)。
- Kubernetes的所有worker节点与待接入的存储设备业务IP通信正常(华为CSI使用ping命令判断)。

- 已向管理员获取Kubernetes集群中任意master节点的IP地址、登录帐号和密码。
- 如果是多路径组网,请确保所有worker节点上安装多路径软件。

# 注意事项

- Kubernetes的worker节点主机名称由数字、字母、"\_"、"-"、"-"、"."和":"组成,首字符只能是数字、字母或"\_",名称长度不超过31个字符。
- 仅FusionStorage 8.0.0及其以后版本支持iSCSI组网配置。

# 操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。
- 步骤2 执行**vi** *huawei-csi-configmap.yaml*命令,创建huawei-csi-configmap.yaml文件。 # vi huawei-csi-configmap.yaml
- 步骤3 配置huawei-csi-configmap.yaml文件, huawei-csi-configmap.yaml文件的模板如回显所示(您也可以参考软件包中deploy/huawei-csi-configmap/huawei-csi-configmap-fusionstorage-iscsi.yaml示例文件)。请根据实际情况设置相关参数,并保存为yaml格式的文件,详细请参考表4-15。

#### 表 4-15 配置项描述

| 配置项                      | 参数格式 | 描述                      | 备注  |
|--------------------------|------|-------------------------|---|
| data."csi.json".backends | 列表   | 必填,待接入<br>后端存储设备<br>列表。 | 配置的后端存储设备数量没有限制。单个后端存储设备支持配置的字段,请参考表4-16。 |

表 4-16 后端存储设备配置项描述

| 配置项        | 参数格式 | 描述                             | 备注   |
|------------|------|--------------------------------|--|
| storage    | 字符串  | 必填,待接入存储<br>设备的类型。             | 对接分布式存储SAN场<br>景固定填写<br>"fusionstorage-san"。   |
| name       | 字符串  | 存储后端名。                         | 自定义字符串,支持大小写字母、数字、中划线组合。<br><b>说明</b> 如果需要配置多个存储后端,请保证存储后端名唯一。   |
| urls       | 列表   | 必填,待接入的存<br>储设备的管理URL。         | FusionStorage的管理<br>URL, 只支持配置一个。  |
| pools      | 列表   | 必填,被使用的待<br>接入存储设备上的<br>存储池名称。 | 支持同一存储设备上的<br>一个或多个存储池, 用逗<br>号分割。<br>可通过登录<br>DeviceManager获取存<br>储池。   |
| parameters | 字典   | 必填,iSCSI场景的可变参数。               | iSCSI场景protocol参数<br>固定填写"iscsi"。<br>portals参数填写指定存储设备的iSCSI业务IP地址,用逗号分割,可通过登录DeviceManager获取。<br>iSCSI业务IP地址可通过登录DeviceManager获取。以OceanStorPacific系列为例:在DeviceManager管理界面,选择"资源》;访问》,业务网络"获取(其它系列请参照相应的操作说明进行获取)。 |

**步骤4** 执行**kubectl create -f** *huawei-csi-configmap.yaml*命令创建huawei-csi-configmap。

# kubectl create -f huawei-csi-configmap.yaml

# kubectl get configmap -n huawei-csi | grep huawei-csi-configmap huawei-csi-configmap 1 5s

-----结束

# 4.2.2.3 通过 NFS 对接分布式存储 NAS 配置

当您需要通过NFS对接分布式存储NAS配置时,请执行此步骤。

# 前提条件

- Kubernetes的所有worker节点已安装NFS客户端工具。
- Kubernetes的所有节点与待接入的存储设备管理IP通信正常(华为CSI使用ping命令判断)。
- Kubernetes的所有worker节点与待接入的存储设备的NFS逻辑端口IP通信正常(华为CSI使用ping命令判断)。
- 已向管理员获取Kubernetes集群中任意master节点的IP地址、登录帐号和密码。

# 操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。
- **步骤2** 执行**vi** *huawei-csi-configmap.yaml*命令,创建huawei-csi-configmap.yaml文件。 # vi huawei-csi-configmap.yaml
- 步骤3 配置huawei-csi-configmap.yaml文件, huawei-csi-configmap.yaml文件的模板如回显所示(您也可以参考软件包中deploy/huawei-csi-configmap/huawei-csi-configmap-fusionstorage-nfs.yaml示例文件)。请根据实际情况设置相关参数,并保存为yaml格式的文件,详细请参考表4-17。

#### 表 4-17 配置项描述

| 配置项                      | 参数格式 | 描述                  | 备注   |
|--------------------------|------|---------------------|--|
| data."csi.json".backends | 列表   | 必填,待接入后端<br>存储设备列表。 | 配置的后端存储设<br>备数量没有限制。                             |
|                          |      |                     | 单个后端存储设备<br>支持配置的字段,<br>请参考 <mark>表4-18</mark> 。 |

表 4-18 后端存储设备配置项描述

| 配置项        | 参数格式 | 描述                             | 备注   |
|------------|------|--------------------------------|--|
| storage    | 字符串  | 必填,待接入存储<br>设备的类型。             | 对接分布式存储NAS场<br>景固定填写<br>"fusionstorage-nas"。   |
| name       | 字符串  | 存储后端名。                         | 自定义字符串,支持大<br>小写字母、数字、中划<br>线组合。<br><b>说明</b><br>如果需要配置多个存储后<br>端,请保证存储后端名唯<br>一。  |
| urls       | 列表   | 必填,待接入的存<br>储设备的管理URL。         | FusionStorage的管理<br>URL, 只支持配置一个。  |
| pools      | 列表   | 必填,被使用的待<br>接入存储设备上的<br>存储池名称。 | 支持同一存储设备上的<br>一个或多个存储池, 用逗<br>号分割。<br>可通过登录<br>DeviceManager获取存<br>储池。   |
| parameters | 字典   | 必填,NFS场景的可变参数。                 | portals: 指定存储的逻辑<br>端口IP,可通过登录<br>DeviceManager获取,<br>当前只支持配置一个。<br>逻辑端口IP地址可通过<br>登录DeviceManager获取。以OceanStor<br>Pacific系列为例: 在<br>DeviceManager管 > 位<br>DeviceManager管 > 位<br>DeviceManager管 > 位<br>DeviceManager管 > 位<br>证别,选择"资单击一个<br>之one的名称。在弹出的<br>页面,单击"IP地址/掩<br>码"列即为逻辑端口IP<br>地址。其它系列请手<br>相应的操作说明进行获取。 |

**步骤4** 执行**kubectl create -f** *huawei-csi-configmap.yaml*命令,创建huawei-csi-configmap。

# kubectl create -f huawei-csi-configmap.yaml

# kubectl get configmap -n huawei-csi | grep huawei-csi-configmap huawei-csi-configmap 1 5s

----结束

# 4.2.2.4 通过 DPC 对接分布式存储 NAS 配置

当您需要通过DPC对接分布式存储NAS配置时,请执行此步骤。

## 前提条件

- Kubernetes的所有worker节点均已在待接入存储上添加为DPC计算节点。
- Kubernetes的所有节点与待接入的存储设备管理IP通信正常(华为CSI使用ping命令判断)。
- 已向管理员获取Kubernetes集群中任意master节点的IP地址、登录帐号和密码。

# 操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。
- **步骤2** 执行**vi** *huawei-csi-configmap.yaml*命令,创建huawei-csi-configmap.yaml文件。 # vi huawei-csi-configmap.yaml
- 步骤3 配置huawei-csi-configmap.yaml文件, huawei-csi-configmap.yaml文件的模板如回显所示(您也可以参考软件包中deploy/huawei-csi-configmap/huawei-csi-configmap-fusionstorage-nfs.yaml示例文件)。请根据实际情况设置相关参数,并保存为yaml格式的文件,详细信息请参考表4-19。

#### 表 4-19 配置项描述

| 配置项                      | 参数格式 | 描述                  | 备注  |
|--------------------------|------|---------------------|---|
| data."csi.json".backends | 列表   | 必填,待接入后端<br>存储设备列表。 | 配置的后端存储设备数量没有限制。单个后端存储设备支持配置的字段,请参考表4-20。 |

| 配置项        | 参数格式 | 描述                             | 备注   |
|------------|------|--------------------------------|--|
| storage    | 字符串  | 必填,待接入存储<br>设备的类型。             | 对接分布式存储NAS场<br>景固定填写<br>"fusionstorage-nas"。                             |
| name       | 字符串  | 存储后端名。                         | 自定义字符串,支持大小写字母、数字、中划线组合。<br><b>说明</b> 如果需要配置多个存储后端,请保证存储后端名唯一。           |
| urls       | 列表   | 必填,待接入的存<br>储设备的管理URL。         | FusionStorage的管理<br>URL, 只支持配置一个。  |
| pools      | 列表   | 必填,被使用的待<br>接入存储设备上的<br>存储池名称。 | 支持同一存储设备上的<br>一个或多个存储池, 用逗<br>号分割。<br>可通过登录<br>DeviceManager获取存<br>储池名称。 |
| parameters | 字典   | 必填,DPC场景的<br>可变参数。             | DPC场景protocol参数固<br>定填写为"dpc"。   |

**步骤4** 执行**kubectl create -f** *huawei-csi-configmap.yaml*命令,创建huawei-csi-configmap。

# kubectl create -f huawei-csi-configmap.yaml

步骤5 创建完成后,执行kubectl get configmap -n huawei-csi | grep huawei-csi | configmap命令,查看是否创建成功。显示如下回显时,则表示创建成功。

# kubectl get configmap -n huawei-csi | grep huawei-csi-configmap huawei-csi-configmap 1 5s

----结束

# 4.2.3 启动 huawei-csi 服务

本章节介绍如何启动huawei-csi服务。

# 注意事项

步骤过程中可能涉及到镜像下载,需要Kubernetes集群的worker节点能够访问外网。如果是内网环境,请通过其他方式获取相关镜像包并手动导入到所有worker节点,镜像包列表请参考表4-23。

# 前提条件

- 已向管理员获取存储的用户名和密码。
- 对接不同存储时,支持的用户类型和要求详见表4-21。

#### □ 说明

当使用租户用户时,仅支持对接NAS存储,具体支持的存储型号参考表2-7。

表 4-21 不同存储对接 CSI 时支持的用户详情

| 存储类型                    | 用户类型 | 角色    | 级别  | 类型   |
|-------------------------|------|-------|-----|------|
| OceanStor V3/V5         | 系统用户 | 管理员   | 管理员 | 本地用户 |
|                         | 租户用户 | 租户管理员 | 管理员 | 本地用户 |
| OceanStor<br>Dorado V3  | 系统用户 | 管理员   | 管理员 | 本地用户 |
| OceanStor 6.1           | 系统用户 | 管理员   | N/A | 本地用户 |
| OceanStor               | 系统用户 | 管理员   | N/A | 本地用户 |
| Dorado 6.1.3            | 租户用户 | 租户管理员 | N/A | 本地用户 |
| OceanStor Pacific<br>系列 | 系统用户 | 管理员   | N/A | 本地用户 |

# 操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意master节点。
- **步骤2** 将Kubernetes CSI组件包中的secretGenerate工具拷贝到master节点的任意目录下,工具路径参见**3.3** 上传软件包组件。

步骤3 使用加密工具填写存储设备的用户名和密码。

- 1. 执行**chmod +x secretGenerate**命令,给secretGenerate添加可执行权限。 # chmod +x secretGenerate
- 2. 执行./secretGenerate --namespace=huawei-csi --logFileDir=/var/log/huawei 命令,运行secretGenerate工具,namespace的参数可修改为实际使用的命名空间,不使用该参数则默认使用huawei-csi命名空间,logFileDir参数可修改为实际使用的日志目录,不使用该参数则默认使用/var/log/huawei目录,根据界面提示输入需要配置backend的编号。Configured为false时表示这个backend尚未配置,为true时表示这个backend已配置。

# ./secretGenerate huawei-csi Getting backend configuration information..... Number Configured BackendName Urls false strage-backend [https://192.168.125.25:8088] strage-backend-02 [https://192.168.125.26:8088] false false strage-backend-03 [https://192.168.125.27:8088] 3 strage-backend-04 [https://192.168.125.28:8088] strage-backend-05 [https://192.168.125.29:28443] false false strage-backend-06 [https://192.168.125.30:28443] Please enter the backend number to configure (Enter 'exit' to exit):3

3. 根据界面提示输入用户名和密码来创建secret对象。

Please enter this backend user name: admin
Please enter this backend password:
Verifying user name and password. Please wait......
The acount information of the backend strage-backend-03 has been configured successfully.

4. 配置完成后,输入exit退出并保存配置。

Please enter the backend number to configure (Enter 'exit' to exit): *exit*Saving configuration. Please wait........
The configuration is saved successfully.

5. 使用kubectl get secret -n huawei-csi | grep huawei-csi-secret命令检查 secret对象是否已创建成功。

# kubectl -n huawei-csi get secret huawei-csi-secret NAME TYPE DATA AGE huawei-csi-secret Opaque 1 8d

#### 步骤4 执行以下命令,创建RBAC权限。

# kubectl apply -f huawei-csi-rbac.yaml

## 步骤5 启动controller服务。

- 1. 执行以下命令,部署snapshot-crd服务。 # kubectl apply -f huawei-csi-snapshot-crd.yaml
- 2. 执行*vi huawei-csi-controller.yaml*命令,修改yaml文件。按l或Insert进入编辑 状态,修改以下相关参数。修改完成后,按Esc,并输入:wq!,保存修改。

#### □ 说明

- (必选)示例yaml文件中huawei-csi-driver的参数image配置项,修改huawei-csi:\*.\*.\* 为上一章节中**制作并上传**的华为CSI镜像<*名称*>:<*版本号*>,以docker为例。 containers:

 name: huawei-csi-driver image: huawei-csi:3.0.0

- (可选)示例yaml文件中metadata的参数namespace配置项表示huawei-csi-controller服务安装的命名空间,如需修改,请确保config.yaml、rbac.yaml、node.yaml文件中命名空间保持一致,修改方式请参考如下。metadata:

...
name: huawei-csi-controller
namespace: huawei-csi

3. 执行以下命令,启动controller服务。 # kubectl apply -f huawei-csi-controller.yaml

#### 步骤6 启动node服务。

 执行vi huawei-csi-node.yaml命令,修改yaml文件。按l或Insert进入编辑状态, 修改相关参数。修改完成后,按Esc,并输入:wq!,保存修改。

#### □ 说明

– (必选)示例yaml文件中huawei-csi-driver的参数image配置项,修改huawei-csi:\*.\*.\* 为上一章节中<mark>制作并上传</mark>的华为CSI镜像<*名称*>:<*版本号*>,以docker为例。

#### containers:

- name: huawei-csi-driver image: huawei-csi:3.0.0
- (可选)示例yaml文件中metadata的参数namespace配置项表示huawei-csi-node服务安装的命名空间,如需修改,请确保config.yaml、rbac.yaml、controller.yaml文件中命名空间保持一致,修改方式请参考如下。

#### metadata:

name: huawei-csi-controller namespace: huawei-csi

- (可选)示例yaml文件中huawei-csi-driver配置参数args:--volume-use-multipath 表示默认开启多路径,如需修改,请参考如下。

#### args:

- "--endpoint=/csi/csi.sock"
- "--containerized"
- "--driver-name=csi.huawei.com"
- "--volume-use-multipath=false"
- (可选)示例yaml文件中huawei-csi-driver配置参数args: --connector-threads 表示 在主机上对盘符操作的并发量,整型,默认值为4,支持范围为1~10,如需修改,请参 考如下。

#### args:

- "--endpoint=/csi/csi.sock"
- "--containerized"
- "--driver-name=csi.huawei.com"
- "--volume-use-multipath=true"
- "--connector-threads=5"
- (可选)示例yaml文件中huawei-csi-driver配置参数args: --scan-volume-timeout 表示在主机上使用DM-multipath多路径时,等待多路径聚合的超时时间,整型,默认值为3,支持范围为1~600,如需修改,请参考如下。

#### args:

- "--endpoint=/csi/csi.sock"
- "--containerized"
- "--driver-name=csi.huawei.com"
- "--volume-use-multipath=true"
- "--connector-threads=4"
- "--scan-volume-timeout=3"
- (可选)示例yaml文件中huawei-csi-driver配置参数args:对于企业存储,当--volume-use-multipath设置为true时,可针对不同组网进行多路径类型配置,详情可参考表4-22。

#### args:

- "--endpoint=/csi/csi.sock"
- "--containerized"
- "--driver-name=csi.huawei.com"
- "--connector-threads=4"
- "--volume-use-multipath=true"
- "--scsi-multipath-type=DM-multipath"
- "--nvme-multipath-type=HW-UltraPath-NVMe"

| 表 4-22  | 企业存储多路径配置参数说明 |
|---------|---------------|
| 4X T 44 |               |

| 存储协议                              | 参数名称                    | 说明   | 备注   |
|-----------------------------------|-------------------------|--|--|
| iSCSI/FC                          | scsi-multipath-<br>type | 取值可为: - DM-multipath - HW-UltraPath - HW- UltraPath- NVMe 默认值为DM- multipath。 | <ul> <li>DM-<br/>multipath:操<br/>作系统原生多<br/>路径软件</li> <li>HW-<br/>UltraPath:华<br/>为UltraPath多<br/>路径软件</li> <li>HW-</li> </ul> |
| NVMe over<br>RoCE/NVMe over<br>FC | nvme-<br>multipath-type | 默认值为HW-<br>UltraPath-<br>NVMe, 仅支持<br>配置HW-<br>UltraPath-<br>NVMe。           | UltraPath-<br>NVMe: 华为<br>UltraPath-<br>NVMe多路径<br>软件  |

2. 执行以下命令,启动node服务。 # kubectl apply -f huawei-csi-node.yaml

## **步骤7** 完成huawei-csi服务部署后,可执行**kubectl get pod -A | grep huawei**命令检查服务 是否启动 。

# kubectl get pod -A | grep huawei huawei-csi huawei-cs

#### □ 说明

在 *huawei-csi-controller-*695b84b4d8-tg64l 的Pod中,一共有7个容器,包含:liveness-probe、csi-provisioner、csi-attacher、csi-resizer、csi-snapshotter、shapshot-controller和 huawei-csi-driver。每个容器都有自己的镜像仓库和功能。关于容器更多说明详情,可参考表4-23。

在*huawei-csi-node*-*g6f7z*的Pod中,一共有3个容器,包含:liveness-probe、csi-node-driver-registrar和huawei-csi-driver。每个容器都有自己的镜像仓库和功能。关于容器更多说明详情,可参考<mark>表4-23</mark>。

#### 表 4-23 容器详细说明

| 容器名称               | 容器镜像  | 功能描述  | 备注   |
|--------------------|---|---|------|
| liveness-<br>probe | k8s.gcr.io/sig-<br>storage/<br>livenessprobe:v2.5.0 | 用于监控CSI的健康状态,并上<br>报给Kubernetes,使<br>Kubernetes能够自动检测CSI程<br>序的问题并重启Pod尝试修改该<br>问题。 | 查看详情 |

| 容器名称                              | 容器镜像  | 功能描述  | 备注         |
|-----------------------------------|---|---|------------|
| csi-<br>provisioner               | k8s.gcr.io/sig-<br>storage/csi-<br>provisioner:v3.0.0                   | <ul> <li>在创建PVC时,调用CSI<br/>Controller服务在存储上创建<br/>LUN/文件系统作为PV,并将<br/>PV绑定至PVC。</li> <li>在删除PVC时,调用CSI<br/>Controller服务解除PV至PVC<br/>的绑定,然后在存储上删除<br/>该PV对应的LUN/文件系统。</li> </ul> | 查看详情       |
| csi-attacher                      | k8s.gcr.io/sig-<br>storage/csi-<br>attacher:v3.3.0                      | 在创建/删除Pod时,调用CSI<br>Controller服务执行Publish/<br>Unpublish Volume操作。   | 查看详情       |
| csi-resizer                       | k8s.gcr.io/sig-<br>storage/csi-<br>resizer:v1.3.0                       | 在扩容PVC时,调用CSI给PVC提供更多的存储容量空间。   | 查看详情       |
| csi-<br>snapshotter               | k8s.gcr.io/sig-<br>storage/csi-<br>snapshotter:v4.2.1                   | 在创建/删除VolumeSnapshot<br>时,调用CSI在存储侧完成快照<br>的创建和删除。  | 查看详情       |
| shapshot-<br>controller           | k8s.gcr.io/sig-<br>storage/snapshot-<br>controller:v4.2.1               | 在创建/删除VolumeSnapshot<br>时,监听Kubernetes API中关于<br>VolumeSnapshot和<br>VolumeSnapshotContent的对<br>象,并触发csi-snapshotter在存<br>储上完成快照的创建。   | 查看详情       |
| csi-node-<br>driver-<br>registrar | k8s.gcr.io/sig-<br>storage/csi-node-<br>driver-<br>registrar:v2.3.0     | 用于获取CSI信息,并通过<br>kubelet的插件注册机制将节点<br>注册到kubelet中,从而<br>Kubernetes能够感知该节点与华<br>为存储的对接。  | 查看详情       |
| huawei-csi-<br>driver             | huawei-csi-driver的<br>名称和TAG为 <b>3.4 制</b><br><b>作华为CSI镜像</b> 中指<br>定的。 | 用于对接Kubernetes平台,提供<br>华为存储(集中式存储/分布式<br>存储)资源给容器使用。  | 版本配套<br>关系 |

----结束

# **5** 升级操作

#### 须知

- 在升级过程中,不能使用CSI下发新的资源。
- 在升级过程中,请勿卸载Snapshot依赖组件服务。
- CSI升级对已经下发的PVC/Snapshot/Pod等资源没有影响。
- 5.1 使用Helm升级华为CSI
- 5.2 手动升级华为CSI

# 5.1 使用 Helm 升级华为 CSI

#### □ 说明

使用Helm安装的华为CSI可以参考本章节进行升级/回退操作,手动安装的华为CSI请参考**5.2 手动升级华为CSI**进行升级(卸载/安装)操作。

# 5.1.1 升级 CSI

## 前提条件

在对CSI进行升级操作时,请确保values.yaml文件中backends字段的参数与之前安装华为CSI时配置的huawei-csi-configmap.yaml中的参数保持一致,否则将会导致升级后的CSI无法对之前发放的PVC、Pod等资源进行管理。

## 操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。
- 步骤2 进入Helm工程的目录下,如果无法找到之前的Helm工程,则将组件包中的helm目录 拷贝到master节点的任意目录下,组件包路径请参考3.3 上传软件包组件。
- **步骤3** 备份安装CSI时使用的values.yaml文件,如果上次安装时使用的values.yaml文件无法找到,可通过**helm get values** *helm-huawei-csi* **-n** *huawei-csi* **-a**命令查询。

# cp values.yaml values.yaml.bak

**步骤4** 执行**vi** *values.yaml*命令打开文件,根据升级需要更新参数值,修改完成后,按**Esc**,并输入:**wq!**,保存修改。您可以更改**表4-1**中提供的所有参数,后端参数请谨慎修改。 # vi values.yaml

步骤5 执行helm upgrade helm-huawei-csi ./ -n huawei-csi -f values.yamt命令升级Helm chart。升级命令将更新Deployment、daemonset和RBAC资源,但是secret资源不会 更新,如果后端信息发生更改,则必须手动更新secret,更新操作请参考9.1 更新CSI 上配置的存储用户名或密码。

其中,helm-huawei-csi为自定义的chart名称,huawei-csi为自定义的命名空间。

# helm upgrade helm-huawei-csi ./ -n huawei-csi

Release "helm-huawei-csi" has been upgraded. Happy Helming!

NAME: helm-huawei-csi

LAST DEPLOYED: Thu Jun 9 07:58:15 2022

NAMESPACE: huawei-csi STATUS: deployed REVISION: 2 TEST SUITE: None

#### ----结束

# 5.1.2 回退 CSI

Helm回滚命令将当前安装的版本回滚到指定的版本。如果未指定版本号,它将回滚到上一个版本。

# 操作步骤

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。

步骤2 将Kubernetes CSI组件包中的"helm"目录拷贝到master节点的任意目录下。Helm工具路径请参见3.3 上传软件包组件。

步骤3 查看当前使用Helm安装的华为CSI版本,使用**helm list -A**命令可以查询所有命名空间中的所有Helm版本;使用**helm list -n** *huawei-csi*可以查询出指定命名空间中的Helm版本。

# helm list -A NAMESPACE NAME **REVISION UPDATED STATUS CHART** APP VERSION helm-test huawei-csi 2022-06-08 08:48:30.038729177 +0000 UTC deployed esdk-1.0.0 300 # helm list -n huawei-csi NAME NAMESPACE **REVISION UPDATED STATUS** CHART APP VERSION helm-test huawei-csi 2022-06-08 08:48:30.038729177 +0000 UTC deployed esdk-1.0.0 3.0.0

步骤4 执行helm rollback release-name revision-number -n huawei-csi命令将华为CSI回退到指定版本。如下示例中将回退到"1"版本。

其中,release-name为需要回退的chart名称,revision-number为需要回退的目标版本,huawei-csi为chart所在的命名空间。

# helm rollback helm-test 1 -n huawei-csi Rollback was a success! Happy Helming!

#### ----结束

# 5.2 手动升级华为 CSI

# 5.2.1 卸载原 CSI

当您需要卸载CSI时,执行此操作。

# 准备步骤

在卸载CSI前,请先执行**kubectl get configmap huawei-csi-configmap -n huawei-csi -o yaml >> huawei-csi-configmap.yaml.bak**命令,备份huawei-csi-configmap的内容(在升级CSI时huawei-csi-configmap.yaml中的backends参数需与现有的configmap配置保持一致)。

# 5.2.1.1 卸载 huawei-csi-node 服务

# 操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意master节点。
- **步骤2** 执行**kubectl delete daemonset huawei-csi-node -n** *huawei-csi*命令卸载huawei-csi-node服务,huawei-csi替换为实际的命名空间。

# kubectl delete daemonset huawei-csi-node -n huawei-csi

步骤3 执行以下命令检查服务是否已成功卸载(如果提示NotFound错误,表示已成功卸载)。

# kubectl get daemonset huawei-csi-node -n huawei-csi

----结束

# 5.2.1.2 卸载 huawei-csi-controller 服务

## 操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意master节点。
- 步骤2 执行kubectl delete deployment huawei-csi-controller -n huawei-csi命令卸载 huawei-csi-controller服务,huawei-csi替换为实际的命名空间。

# kubectl delete deployment huawei-csi-controller -n huawei-csi

步骤3 执行以下命令检查服务是否已成功卸载(如果提示NotFound错误,表示已成功卸载)。

# kubectl get deployment huawei-csi-controller -n huawei-csi

----结束

# 5.2.1.3 删除 huawei-csi-configmap 对象

# 操作步骤

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意master节点。

步骤2 执行kubectl delete configmap huawei-csi-configmap -n huawei-csi命令删除 configmap对象,其中huawei-csi-configmap是configmap的名称,huawei-csi是对象所在的命名空间。

# kubectl delete configmap huawei-csi-configmap -n huawei-csi

步骤3 执行以下命令检查对象是否已成功删除(如果提示NotFound错误,表示已成功删除)。

# kubectl get configmap huawei-csi-configmap -n huawei-csi

----结束

# 5.2.1.4 删除 huawei-csi-secret 对象

# 操作步骤

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意master节点。

**步骤2** 执行**kubectl delete secret** *huawei-csi-secret* **-n** *huawei-csi*命令删除secret对象,其中huawei-csi-secret是secret对象的名称,huawei-csi是secret对象所在的命名空间。
# kubectl delete secret huawei-csi-secret -n huawei-csi

**步骤3** 执行以下命令检查secret对象是否删除成功,如果命令回显提示"NotFound"表示huawei-csi-secret对象已成功删除。

# kubectl get secret huawei-csi-secret -n huawei-csi
Error from server (NotFound): secrets "huawei-csi-secret" not found

----结束

# 5.2.1.5 删除 RBAC 权限

### 操作步骤

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。

步骤2 删除RBAC权限。

● 当huawei-csi版本号高于2.2.15,执行以下命令进行删除,-n参数为命名空间,请按实际修改。

# kubectl -n huawei-csi -l provisioner=csi.huawei.com delete ServiceAccount,role,rolebinding,ClusterRole,ClusterRoleBinding

- 当huawei-csi版本号低于或等于2.2.15,按照以下步骤进行删除。
  - a. 执行以下命令创建delete-huawei-csi-rbac.sh文件,-n参数为命名空间,请按 实际修改。

# cat <<EOF > delete-huawei-csi-rbac.sh

kubectl delete ServiceAccount huawei-csi-controller -n huawei-csi

kubectl delete ServiceAccount huawei-csi-node -n huawei-csi

kubectl delete ClusterRole huawei-csi-attacher-runner -n huawei-csi

kubectl delete ClusterRole huawei-csi-driver-registrar-runner -n huawei-csi

kubectl delete ClusterRole huawei-csi-provisioner-runner -n huawei-csi

kubectl delete ClusterRole huawei-csi-resizer-runner -n huawei-csi kubectl delete ClusterRole huawei-csi-snapshotter-runner -n huawei-csi

kubectl delete ClusterRole snapshot-controller-runner -n huawei-csi

kubectl delete ClusterRoleBinding huawei-csi-attacher-role -n huawei-csi

kubectl delete ClusterRoleBinding huawei-csi-driver-registrar-role -n huawei-csi

kubectl delete ClusterRoleBinding huawei-csi-provisioner-role -n huawei-csi

kubectl delete ClusterRoleBinding huawei-csi-resizer-role -n huawei-csi

kubectl delete ClusterRoleBinding huawei-csi-snapshotter-role -n huawei-csi kubectl delete ClusterRoleBinding snapshot-controller-role -n huawei-csi kubectl delete Role huawei-csi-resizer-cfg -n huawei-csi kubectl delete Role huawei-csi-snapshotter-leaderelection -n huawei-csi kubectl delete Role snapshot-controller-leaderelection -n huawei-csi kubectl delete RoleBinding huawei-csi-resizer-role-cfg -n huawei-csi kubectl delete RoleBinding huawei-csi-snapshotter-leaderelection -n huawei-csi kubectl delete RoleBinding snapshot-controller-leaderelection -n huawei-csi

b. 执行以下命令删除RBAC权限。如果报NotFound错误,请忽略。 # sh delete-huawei-csi-rbac.sh

#### 步骤3 按照以下步骤检查RBAC权限是否已成功删除。

当huawei-csi版本号高于2.2.15,执行以下命令,-n参数为命名空间,请按实际修改。如果提示No resources found,则表示已成功删除。

# kubectl -n huawei-csi -l provisioner=csi.huawei.com get ServiceAccount,role,rolebinding,ClusterRole,ClusterRoleBinding

- 当huawei-csi版本号低于或等于2.2.15,按照以下步骤检查RBAC权限是否已成功 删除。
  - a. 执行以下命令创建check-huawei-csi-rbac.sh文件,-n参数为命名空间,请按实际修改。

# cat <<EOF > check-huawei-csi-rbac.sh kubectl get ServiceAccount -n huawei-csi | grep huawei-csi kubectl get ClusterRole -n huawei-csi | grep huawei-csi

kubectl get ClusterRoleBinding -n huawei-csi | grep huawei-csi

kubectl get Role -n huawei-csi | grep huawei-csi

kubectl get RoleBinding -n huawei-csi | grep huawei-csi

kubectl get ClusterRole snapshot-controller-runner -n huawei-csi --ignore-not-found=true kubectl get ClusterRoleBinding snapshot-controller-role -n huawei-csi --ignore-not-found=true

kubecti get Clusterkolebinding snapshot-controller-role -n nuawei-csi --ignore-not-found=true

kubectl get RoleBinding snapshot-controller-leaderelection -n huawei-csi --ignore-not-found=true

FOF

b. 执行以下命令,如果没有回显,表示RBAC权限已成功删除。 # sh check-huawei-csi-rbac.sh

#### ----结束

## 5.2.1.6 删除老版本镜像

如果需要在集群中删除huawei-csi镜像,需要在所有的worker节点都执行一遍删除操作。

以删除单个节点的镜像为例,操作步骤如下。

# 前提条件

依赖该镜像的容器服务已经被停止,否则无法删除。

# 操作步骤

步骤1 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录该worker节点。

步骤2 执行以下命令,查看现存的所有版本。

● 如果使用的docker,执行**docker image ls | grep huawei-csi**命令。

# docker image ls | grep huawei-csi

REPOSITORY TAG IMAGE ID CREATED SIZE

```
huawei-csi 2.2.15 b30b3a8b5959 2 weeks ago 79.7MB
huawei-csi 3.0.0 14b854dba227 2 weeks ago 79.6MB
```

● 如果使用的containerd,执行**crictl image ls | grep huawei-csi**命令。

```
# crictl image ls | grep huawei-csi
REPOSITORY TAG IMAGE ID CREATED SIZE
docker.io/library/huawei-csi 2.2.15 b30b3a8b5959 2 weeks ago 79.7MB
docker.io/library/huawei-csi 3.0.0 14b854dba227 2 weeks ago 79.6MB
```

#### 步骤3 执行以下命令,删除老版本镜像:

- 如果使用的docker, 执行docker rmi <REPOSITORY>:<TAG>命令。。# docker rmi huawei-csi:2.2.15
- 如果使用的containerd,执行crictl rmi <REPOSITORY>:<TAG>命令。
   # crictl rmi huawei-csi:2.2.15

**步骤4** 再次执行以下命令,检查镜像是否已成功删除。如果已没有要删除版本的记录,表示已成功删除该版本。

```
● 如果使用的docker,执行docker image ls | grep huawei-csi命令。
# docker image ls | grep huawei-csi
huawei-csi 3.0.0 14b854dba227 10 minutes ago 80MB
```

如果使用的containerd,执行crictl image ls | grep huawei-csi命令。
 # crictl image ls | grep huawei-csi docker.io/library/huawei-csi
 3.0.0
 14b854dba2273
 93.1MB

----结束

# 5.2.2 安装新 CSI

卸载完成后需要重新安装CSI。

# 前提条件

已备份原CSI中的huawei-csi-configmap.yaml内容。

## 注意事项

如果huawei-csi-configmap.yaml的模板变更,请确保以下参数配置与升级前一致,**否则会导致huawei-csi服务无法启动,并且无法管理已创建资源**。

- "storage", "name", "pools" 参数与5.2.1 卸载原CSI的前提条件中已备份的huawei-csi-configmap.yaml.bak文件保持一致。
- "urls"和"parameters"参数请参考5.2.1 卸载原CSI的前提条件中已备份的huawei-csi-configmap.yaml.bak文件中的信息,按照当前版本的huawei-csi-configmap.yaml模板填写,模板详细请参考4.2.1 对接企业存储配置和4.2.2 对接分布式存储配置。以下回显仅为示例。

```
"backends": [

{
    "storage": "oceanstor-san",
    "name": "***",
    "urls": ["https://*.*.*:8088", "https://*.*.*:8088"],
    "pools": ["***", "***"],
    "parameters": {"protocol": "iscsi", "portals": ["*.*.*", "*.*.*"]}
  }
}
```

## 操作步骤

步骤1 获取CSI新版本软件包,请参见3.2 获取软件包。

步骤2 制作CSI新版本镜像,请参见3.4 制作华为CSI镜像。

步骤3 创建huawei-csi-configmap,请参见4.2.1 对接企业存储配置或4.2.2 对接分布式存储配置。

步骤4 启动huawei-csi服务,请参见4.2.3 启动huawei-csi服务。

----结束

# **6** 使用说明

本章节主要介绍如何使用华为CSI。

- 6.1 (条件必选)管理StorageClass
- 6.2 (条件必选)管理PV
- 6.3 管理PVC
- 6.4 管理Pod
- 6.5 (可选)管理快照

# 6.1 (条件必选)管理 StorageClass

关于PV的发放有两种方式:静态发放和动态发放。静态发放是指在存储设备上预先完成存储资源(LUN/共享)的创建,然后通过静态发放提供给Kubernetes使用。动态发放是指通过CSI自动化地完成对存储资源(LUN/共享)的创建。

当您需要使用动态PV功能时,执行此操作。

# 6.1.1 创建 StorageClass

StorageClass是在申请存储资源时,可供选择的一组能力定义。Kubernetes集群用户可基于StorageClass创建PVC。

# 6.1.1.1 创建 LUN StorageClass

本章节介绍如何创建LUN StorageClass。

# 操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意master节点。
- **步骤2** 执行**vi** *StorageClass.yaml*命令,创建StorageClass.yaml文件。 # vi StorageClass.yaml
- **步骤3** 按l或Insert进入编辑状态,在StorageClass.yaml文件中输入以下内容。修改完成后,按Esc,并输入:wq!,保存修改。

StorageClass.yaml文件的模板如回显所示(您也可以参考软件包中examples/lun-sc-for-csi-example.yaml示例文件)。请根据实际情况设置相关参数,并保存为yaml格式的文件,详细参考表6-1

kind: StorageClass apiVersion: storage.k8s.io/v1 metadata: name: "mysc" provisioner: "csi.huawei.com" parameters: volumeType: "lun" allocType: "thin"

#### 表 6-1 参数说明

| 参数                          | 说明                           | 备注  |
|-----------------------------|------------------------------|---|
| metadata.name               | 自定义的<br>StorageClass对<br>象名称 | 以Kubernetes v1.22.1为例,支持数字、小写字母、中划线(-)和点(.)的组合,并且必须以字母数字字符开头和结尾。              |
| provisioner                 | provisioner标识                | 固定填写"csi.huawei.com"。   |
| parameters.volu<br>meType   | 待创建卷类型                       | 固定填写"lun"。  |
| parameters.alloc<br>Type    | 待创建卷的分配<br>类型                | 可选,支持thin/thick,默认为thin。  |
| parameters.clone<br>Speed   | 指定克隆速度                       | 可选,默认值3,支持1~4,4速度最快。配置克隆PVC或从快照创建PVC时生效,参考6.3.3(可选)克隆PVC或6.3.4(可选)从快照创建PVC。   |
| parameters.fsTyp<br>e       | 指定文件系统类型                     | 可选,支持ext2/ext3/ext4/xfs,默认为ext4。 <b>须知</b> CSI不会对fsType字段的值做有效性校验,请保证填写值的准确性。 |
| parameters.fsPer<br>mission | 指定文件系统权<br>限                 | 可选,配置格式如:"755"。<br>仅当PVC的volumeMode配置为Filesystem<br>时生效。                      |

## 步骤4 执行以下命令,基于该yaml文件创建StorageClass。

# kubectl create -f StorageClass.yaml

# 步骤5 执行以下命令,查看当前已经创建的StorageClass信息。

# kubectl get sc NAME PROVISIONER RECLAIMPOLICY VOLUMEBINDINGMODE ALLOWVOLUMEEXPANSION AGE mysc csi.huawei.com Delete Immediate false 87s

#### ----结束

# 6.1.1.2 创建文件系统 StorageClass

本章节介绍如何创建文件系统StorageClass。

# 操作步骤

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。

**步骤2** 执行**vi** *StorageClass.yaml*命令,创建StorageClass.yaml文件。 # vi StorageClass.yaml

**步骤3** 按l或Insert进入编辑状态,在StorageClass.yaml文件中输入以下内容。修改完成后,按Esc,并输入:wq!,保存修改。

StorageClass.yaml文件的模板如回显所示(您也可以参考软件包中examples/fs-sc-for-csi-example.yaml示例文件)。请根据实际情况设置相关参数,并保存为yaml格式的文件,详细参考表6-2

kind: StorageClass apiVersion: storage.k8s.io/v1 metadata: name: "mysc" provisioner: "csi.huawei.com" mountOptions: - nfsvers=3 parameters: volumeType: "fs" allocType: "thin" authClient: "\*"

#### 表 6-2 参数说明

| 参数                            | 说明                       | 备注   |
|-------------------------------|--------------------------|--|
| metadata.<br>name             | 自定义的StorageClass<br>对象名称 | 以Kubernetes v1.22.1为例,支持数字、小写字母、中划线(-)和点(.)的组合,并且必须以字母数字字符开头和结尾。   |
| provisioner                   | provisioner标识            | 固定填写"csi.huawei.com"。  |
| mountOpti<br>ons              | 挂载选项                     | 在主机执行mount命令时-o参数后的可选选项。列表格式。<br>指定NFS版本挂载时,使用nfsvers=*参数,当前支持NFS 3/4.0/4.1协议(需存储设备支持且开启)。当配置参数为nfsvers=4时,因为操作系统配置的不同,实际挂载可能为NFS 4的最高版本协议,如4.1,当需要使用4.0协议时,建议配置nfsvers=4.0。 |
| parameters<br>.volumeTy<br>pe | 待创建卷类型                   | 固定填写"fs"   |

| 参数                            | 说明   | 备注  |
|-------------------------------|--|---|
| parameters<br>.authClient     | 指定可访问该卷的客户<br>端                                | 必选。以OceanStor Dorado 6.x系列为例,<br>支持输入客户端主机名称(建议使用全称域<br>名)、客户端IP地址、客户端IP地址段或使<br>用"*"表示全部客户端IP地址。 |
|                               |  | IP地址支持IPv4、IPv6地址或两者的混合IP<br>地址。  |
|                               |  | 您可以同时输入多个主机名称、IP地址或IP<br>地址段,以英文分号,空格或按回车键隔<br>开。   |
|                               |  | 示例: 192.168.0.10;192.168.0.0/24;*   |
| parameters<br>.allocType      | 待创建卷的分配类型                                      | 可选,支持thin/thick,默认为thin。  |
| parameters<br>.cloneSpee<br>d | 指定克隆速度   | 可选,默认值3,支持1~4,4速度最快。配置克隆PVC或从快照创建PVC时生效,参考6.3.3(可选)克隆PVC或6.3.4(可选)从快照创建PVC。                       |
| parameters<br>.fsType         | 指定文件系统类型                                       | 可选,支持ext2/ext3/ext4/xfs,默认为ext4。 <b>须知</b> CSI不会对fsType字段的值做有效性校验,请保证                             |
|                               |  | 填写值的准确性。  |
| parameters<br>.fsPermissi     | 指定文件系统权限                                       | 可选,配置格式如: "755"。  |
| on                            |  | 企业存储仅支持OceanStor V6及OceanStor<br>Dorado V6。   |
|                               |  | 分布式存储仅支持OceanStor Pacific<br>8.1.2/8.1.3  |
| parameters                    | 设置是否保留共享目录<br>的UID(User ID)和<br>GID(Group ID)。 | 可选,支持配置以下参数:  |
| .allSquash                    |  | ● "0":all_squash,共享文件的UID和GID<br>映射给nobody用户,适合公共目录。  |
|                               |  | ● "1": no_all_squash,保留共享文件的<br>UID和GID。  |
| parameters<br>.rootSquas<br>h | 设置是否允许客户端的<br>root权限。                          | 可选,支持配置以下参数:  |
|                               |  | <ul><li>"0": root_squash,表示不允许客户端以<br/>root用户访问共享目录,客户端使用root<br/>用户访问时映射为匿名用户。</li></ul>         |
|                               |  | ● "1": no_root_squash,允许客户端以<br>root用户访问共享目录,root用户具有共<br>享目录的完全控制访问权限。                           |

步骤4 执行以下命令,基于该yaml文件创建StorageClass。

# kubectl create -f StorageClass.yaml

步骤5 执行以下命令,查看当前已经创建的StorageClass信息。

# kubectl get sc

NAME PROVISIONER RECLAIMPOLICY VOLUMEBINDINGMODE ALLOWVOLUMEEXPANSION AGE mysc csi.huawei.com Delete Immediate false 34s

----结束

# 6.1.2 删除 StorageClass

本章节介绍如何删除StorageClass。

## 操作步骤

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。

步骤2 执行以下命令, 查询集群中的StorageClass。

# kubectl get sc

NAME PROVISIONER RECLAIMPOLICY VOLUMEBINDINGMODE ALLOWVOLUMEEXPANSION

AGE

huawei-nas csi.huawei.com Delete Immediate false 3s mysc csi.huawei.com Delete Immediate false 16s

步骤3 执行以下命令,删除StorageClass。例如删除名称为mysc的StorageClass。

# kubectl delete sc *mysc* storageclass.storage.k8s.io "*mysc*" deleted

**步骤4** 执行以下命令, 查询集群中的StorageClass。回显中不包含需要删除的StorageClass 名称,则表示删除成功。

# kubectl get sc

NAME PROVISIONER RECLAIMPOLICY VOLUMEBINDINGMODE ALLOWVOLUMEEXPANSION

AGE

huawei-nas csi.huawei.com Delete Immediate false 3s

----结束

# 6.2 (条件必选)管理 PV

关于PV的发放有两种方式:静态发放和动态发放。静态发放是指在存储设备上预先完成存储资源(LUN/共享)的创建,然后通过静态发放提供给Kubernetes使用。动态发放是指通过CSI自动化地完成对存储资源(LUN/共享)的创建。

当您需要使用静态PV功能时,执行此操作。

# 6.2.1 创建 PV

## 操作步骤

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。

步骤2 执行vi PersistentVolume.yaml命令, 创建 PersistentVolume.yaml文件。

# vi PersistentVolume.yaml

步骤3 按l或Insert进入编辑状态,在PersistentVolume.yaml文件中输入以下内容。修改完成后,按Esc,并输入:wq!,保存修改。

PersistentVolume.yaml文件的模板如回显所示(您也可以参考软件包中examples/static-pv-example.yaml示例文件)。请根据实际情况设置相关参数,并保存为yaml格式的文件,详细参考表6-3。

kind: PersistentVolume
apiVersion: v1
metadata:
name: mypv
spec:
volumeMode: Block
storageClassName: ""
accessModes:
- ReadWriteOnce
csi:
driver: csi.huawei.com
volumeHandle: <backendName>.<volume-name>
fsType: <string>
capacity:
storage: 100Gi

## 表 6-3 参数说明

| 参数   | 说明                       | 备注   |
|--|--------------------------|--|
| metadata.nam<br>e                          | 自定义的<br>PV对象<br>名称       | 以Kubernetes v1.22.1为例,支持数字、小写字母、中划线(-)和点(.)的组合,并且必须以字母数字字符开头和结尾。   |
| spec.volumeM<br>ode                        | 卷模式                      | 可选, 支持Filesystem或Block, 默认为Filesystem。<br>该参数在创建Pod时生效,其中Filesystem表示在PVC<br>上创建一个文件系统访问存储, Block表示使用裸卷<br>的方式访问存储。                        |
| spec.storageCla<br>ssName                  | StorageC<br>lass对象<br>名称 | 必选,此处须设置为空字符串(即输入"")。  |
| spec.persistent<br>VolumeReclai<br>mPolicy | 卷的回收<br>策略               | 可选,枚举值为Retain手动回收,Delete删除已关联的存储资源,默认值为Retain,详见 <mark>表6-5</mark> 。   |
| spec.accessMo<br>des                       | 指定卷访<br>问模式              | 卷模式为Filesystem的LUN卷支持ReadWriteOnce、<br>ReadOnlyMany和ReadWriteOncePod。  |
|  |                          | 卷模式为Block的LUN卷支持ReadWriteOnce、<br>ReadOnlyMany、ReadWriteMany和<br>ReadWriteOncePod,当配置为ReadWriteMany时,<br>需要由Pod业务保证数据一致性。                  |
|  |                          | 文件系统卷支持ReadWriteOnce,ReadOnlyMany、<br>ReadWriteMany和ReadWriteOncePod。  |
|  |                          | <b>须知</b><br>由于ReadWriteOncePod访问模式是Kubernetes v1.22+版本<br>的alpha特性,需要先为集群开启ReadWriteOncePod特性才<br>能使用,详见 <b>9.7 开启ReadWriteOncePod功能门</b> |
| spec.csi.driver                            | CSI驱动<br>名称              | 固定填写csi.huawei.com。  |

| 参数                        | 说明           | 备注   |
|---------------------------|--------------|--|
| spec.csi.volume           |              | 由两部分构成:  |
| Handle                    |              | <ul> <li><backendname>: 指的是configmap中每个后端的名称,可使用如下命令获取configmap:</backendname></li> <li>kubectl get configmap huawei-csi-configmap-n huawei-csi -o yaml</li> </ul> |
|                           |              | <ul><li><volume-name>: 指的是存储上资源(LUN/文件系统)的名称,可通过DeviceManager查看获取。</volume-name></li></ul>   |
| spec.csi.fsType           | 指定文件<br>系统类型 | 可选,支持ext2/ext3/ext4/xfs,默认为ext4。当<br>volumeMode为Filesystem时生效。   |
|                           |              | <b>须知</b><br>CSI不会对fsType字段的值做有效性校验,请保证值的准确<br>性。  |
| spec.capacity.st<br>orage | 指定卷大小        | 在创建PV时,请确保与存储上对应资源的容量保持一致,(因为Kubernetes并不会调用CSI检查此字段值的正确性,所以在PV容量与存储上对应资源的容量不一致也能被成功创建)。   |

# 步骤4 执行以下命令,基于该yaml文件创建PV。

# kubectl create -f PersistentVolume.yaml

# 步骤5 等待一段时间后,执行以下命令,查看已经创建的PV信息。

# kubectl get pv

NAME CAPACITY ACCESS MODES RECLAIM POLICY STATUS CLAIM STORAGECLASS

REASON AGE

mypv 100Gi RWO Retain Available 4s

## 表 6-4 PV 状态说明

| 状态              | 说明  |
|-----------------|---|
| Available       | 尚未绑定到PVC。                                       |
| Bound           | 已绑定到PVC。  |
| Released        | PVC已被删除,但资源尚未被集群回收。                             |
| Failed          | 卷的自动回收操作失败。                                     |
| Terminatin<br>g | 该PV对象已被Kubernetes 控制器标记为删除, 但因为该PV正被使用 而无法成功删除。 |

#### 表 6-5 回收策略说明

| 回收策略   | 说明  |
|--------|---|
| Delete | ● 当PVC对象被删除后,PV卷仍然存在,对应的数据卷状态为<br>"Failed",该PV卷将无法绑定其他PVC。                                |
|        | ● 删除PV对象后,后端存储设备中对应的文件系统或LUN也将<br>被删除。  |
| Retain | ● 用户可以手动回收资源。当PVC对象被删除后,PV卷仍然存在,对应的数据卷状态为"Released"。由于PV卷上仍然存在前一个PVC的数据,因此该PV卷将无法绑定其他PVC。 |
|        | ● 删除PV对象后,后端存储设备中对应的文件系统或LUN将保留。  |

#### ----结束

# 6.2.2 删除 PV

本章节介绍如何删除PV。

# 操作步骤

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。

步骤2 执行以下命令,查询集群中的PV。

# kubectl get pv

NAME CAPACITY ACCESS MODES RECLAIM POLICY STATUS CLAIM STORAGECLASS REASON AGE
mypv 100Gi RWO Retain Available 11m

步骤3 执行以下命令,删除PV。例如删除名称为mypv的PV。

# kubectl delete pv *mypv* persistentvolume "mypv" deleted

**步骤4** 执行以下命令,查询集群中的PV。回显中不包含需要删除的PV名称,则表示删除成功。

# kubectl get pv

No resources found in default namespace.

#### ----结束

# 6.3 管理 PVC

# 6.3.1 创建 PVC

本章节用于介绍如何创建PVC。

# 限制说明

存储限制RESTful请求并发量为100,因此建议每批次最多批量创建/删除100个PVC。

# 操作步骤

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意master节点。

步骤2 执行**vi** *PersistentVolumeClaim.yaml*命令,创建 PersistentVolumeClaim.yaml文件。
# vi PersistentVolumeClaim.yaml

**步骤3** 按l或lnsert进入编辑状态,在PersistentVolumeClaim.yaml文件中输入以下内容。修改完成后,按Esc,并输入:wq!,保存修改。

PersistentVolumeClaim.yaml文件的模板如回显所示(您也可以参考软件包中 examples/pvc-example.yaml示例文件 )。请根据实际情况设置相关参数,并保存为 yaml格式的文件,详细参考表6-6

kind: PersistentVolumeClaim apiVersion: v1 metadata: name: "mypvc" spec: accessModes: - ReadWriteOnce volumeMode: Filesystem volumeName: mypv storageClassName: "mysc" resources: requests: storage: 100Gi

#### 表 6-6 参数说明

| 参数                            | 说明                   | 备注   |  |
|-------------------------------|----------------------|--|--|
| metadata.<br>name             | 自定义的<br>PVC对象名<br>称  | 以Kubernetes v1.22.1为例,支持数字、小写字母、中划线( - )和点( . )的组合,并且必须以字母数字字符开头和结尾。   |  |
| spec.volu<br>meMode           | 卷模式                  | 可选, 支持Filesystem或Block, 默认为Filesystem。该<br>参数在创建Pod时生效,其中Filesystem表示在PVC上创<br>建一个文件系统访问存储, Block表示使用裸卷的方式<br>访问存储。    |  |
| spec.volu<br>meName           | PV对象名<br>称           | 仅当静态创建PVC时必选。  |  |
| spec.stora<br>geClassNa<br>me | StorageCla<br>ss对象名称 | <ul> <li>当动态创建PVC时,填写6.1(条件必选)管理<br/>StorageClass创建的StorageClass对象名称。</li> <li>当静态创建PVC时,此处须设置为空字符串(即输入"")。</li> </ul> |  |

| 参数                       | 说明           | 备注  |
|--------------------------|--------------|---|
| spec.resou<br>rces.reque | 指定待创建<br>卷大小 | 格式为***Gi,单位为GiB。  |
| sts.storage              |              | <ul> <li>PVC容量的规格取决于存储规格限制和主机规格限制,以OceanStor Dorado 6.1.2/OceanStor Pacific系列 8.1.0对接CentOS 7为例,示例表6-7和表6-8。</li> </ul>   |
|                          |              | ● 其余存储和主机请根据 <mark>StorageClass</mark> 中<br>"VolumeType"类型进行规格查看。   |
|                          |              | <ul> <li>如果 "volumeType" 类型为 "lun",需要参考存储自身规格,详情请参考https://info.support.huawei.com/storage/spec/#/home。同时还需要主机连通性指南,详情请参考https://support.huawei.com/enterprise/zh/doc/EDOC1100112792/87e2c8fd。</li> </ul> |
|                          |              | <ul> <li>如果 "volumeType"类型为 "fs",参考存储自身<br/>支持的规格,详情请参考https://<br/>info.support.huawei.com/storage/spec/#/<br/>home。</li> </ul>  |
|                          |              | <ul><li>如果PVC容量不在规格范围内,可能会由于存储规格<br/>限制或主机文件系统规格限制导致创建PVC或Pod失<br/>败。</li></ul>   |
|                          |              | <ul><li>在通过静态PV创建PVC时,若PVC容量小于绑定PV容量,最终PVC容量大小为绑定PV容量,若PVC容量大于绑定PV容量,PVC将无法被创建。</li></ul>   |
| spec.acces<br>sModes     | 指定卷访问<br>模式  | 卷模式为Filesystem的LUN卷支持ReadWriteOnce、<br>ReadOnlyMany和ReadWriteOncePod。   |
|                          |              | 卷模式为Block的LUN卷支持ReadWriteOnce、<br>ReadOnlyMany、ReadWriteMany和<br>ReadWriteOncePod,当配置为ReadWriteMany时,需<br>要由Pod业务保证数据一致性。   |
|                          |              | 文件系统卷支持ReadWriteOnce、ReadOnlyMany、<br>ReadWriteMany和ReadWriteOncePod。   |
|                          |              | <b>须知</b><br>由于ReadWriteOncePod访问模式是Kubernetes v1.22+版本的<br>alpha特性,需要先为集群开启ReadWriteOncePod特性才能使<br>用,详见 <b>9.7 开启ReadWriteOncePod功能门</b>  |

## 表 6-7 PVC 容量的规格(ext4)

| volumeTyp<br>e类型 | 存储类型                      | 存储规格限制          | ext4规格限制 | CSI规格限制    |
|------------------|---------------------------|-----------------|----------|------------|
| lun              | OceanStor Dorado<br>6.1.2 | 512Ki~256<br>Ti | 50Ti     | 512Ki~50Ti |
|                  | OceanStor Pacific系列 8.1.0 | 64Mi~512T<br>i  | 50Ti     | 64Mi~50Ti  |

| volumeTyp<br>e类型 | 存储类型                      | 存储规格限制    | ext4规格限制 | CSI规格限制     |
|------------------|---------------------------|-----------|----------|-------------|
| fs               | OceanStor Dorado<br>6.1.2 | 1Gi~32Pi  | N/A      | 1Gi~32Pi    |
|                  | OceanStor Pacific系列 8.1.0 | 1Ki~256Pi | N/A      | 1Ki ~ 256Pi |

#### 表 6-8 PVC 容量的规格(xfs)

| volumeTyp<br>e类型 | 存储类型                          | 存储规格限制          | xfs规格限制 | CSI规格限制     |
|------------------|-------------------------------|-----------------|---------|-------------|
| lun              | OceanStor Dorado<br>6.1.2     | 512Ki~256<br>Ti | 500Ti   | 512Ki~500Ti |
|                  | OceanStor Pacific系列 8.1.0     | 64Mi~512T<br>i  | 500Ti   | 64Mi~500Ti  |
| fs               | OceanStor Dorado<br>6.1.2     | 1Gi~32Pi        | N/A     | 1Gi~32Pi    |
|                  | OceanStor Pacific系<br>列 8.1.0 | 1Ki~256Pi       | N/A     | 1Ki~256Pi   |

### 步骤4 执行以下命令,基于该yaml文件创建PVC。

# kubectl create -f PersistentVolumeClaim.yaml

### 步骤5 等待一段时间后,执行以下命令,查看已经创建的PVC信息。

# kubectl get pvc

NAME STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS AGE mypvc Bound pvc-840054d3-1d5b-4153-b73f-826f980abf9e 100Gi RWX mysc 12s

#### (一) 沿阳

完成创建PVC操作后,如果PVC的状态是Pending,请参考**10.6 创建PVC时, PVC的状态为** Pending。

### ----结束

## 6.3.2 (可选)扩容 PVC

本章节介绍如何扩容PVC。

### 前提条件

- 已经创建PVC,且PVC所在的backend存在支持扩容。支持扩容的存储请参考表 2-4和表2-5。
- huawei-csi服务状态正常。

# kubectl get pod -A | grep huawei

huawei-csi **huawei-csi-controller**-fd5f97768-qlldc **7/7** Running 0 16s huawei-csi **huawei-csi-node**-25txd **3/3** Running 0 15s

### 操作步骤

使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。

步骤2 检查 huawei-csi-controller是否启用了csi-resizer服务。

# kubectl describe deploy huawei-csi-controller -n huawei-csi | grep csi-resizer

k8s.gcr.io/sig-storage/csi-resizer:v1.3.0 Image:

- 如果结果如上所示,表示csi-resizer服务已经启用,直接执行步骤3。
- 否则,表示csi-resizer服务未启用,则需要先将huawei-csi版本升级到2.2.15以 上。
- 步骤3 执行kubectl get pvc mypvc命令,查询PVC的StorageClass名称。其中,mypvc为需 要扩容的PVC名称。

# kubectl get pvc mypvc

NAME STATUS VOLUME

CAPACITY ACCESS MODES

STORAGECLASS AGE

mypvc Bound pvc-3383be36-537c-4cb1-8f32-a415fa6ba384 2Gi **RWX** 

mysc

步骤4 执行kubectl get sc mysc命令,查询StorageClass是否支持扩容。其中,mysc为需要 查看的StorageClass名称。

# kubectl get sc mysc

PROVISIONER RECLAIMPOLICY VOLUMEBINDINGMODE NAME

**ALLOWVOLUMEEXPANSION** AGE

mysc csi.huawei.com Delete Immediate true 172m

- 如果ALLOWVOLUMEEXPANSION的值为false,表示当前StorageClass不支持扩 容,请执行<mark>步骤5</mark>。
- 如果ALLOWVOLUMEEXPANSION的值为true,表示当前StorageClass已经支持扩 容,请执行步骤6。
- 步骤5 (可选)执行以下命令,将"allowVolumeExpansion"的值修改为"true"。其中, mysc为需要修改的StorageClass名称。

# kubectl patch sc mysc --patch '{"allowVolumeExpansion":true}'

步骤6 执行以下命令进行扩容。

# kubectl patch pvc mypvc -p '{"spec":{"resources":{"requests":{"storage":"120Gi"}}}}'

其中,"mypvc"是需要扩容的PVC名称,"120Gi"是扩容后的容量大小。请根据实际 情况进行替换。

#### □ 说明

- PVC容量的规格取决于存储规格限制和主机规格限制,以OceanStor Dorado 6.1.2或 OceanStor Pacific系列 8.1.0对接CentOS 7为例,参考表6-9。
- 其余存储和主机请根据6.1.1 创建StorageClass中"VolumeType"的值进行规格查看。
  - 如果 "volumeType" 的值为 "lun", 需要参考存储自身规格, 详情请参考https:// info.support.huawei.com/storage/spec/#/home。同时还需要参考主机连通性指 南,详情请参考https://support.huawei.com/enterprise/zh/doc/ EDOC1100112792/87e2c8fd。
  - 如果"volumeType"的值为"fs",参考存储自身支持的规格,详情请参考https:// info.support.huawei.com/storage/spec/#/home。
- 如果PVC容量不在规格范围内,可能会由于存储规格限制或主机文件系统规格限制导致创建 PVC或Pod失败。

表 6-9 PVC 容量的规格(ext4)

| volumeTyp<br>e类型 | 存储类型                      | 存储规格限制          | ext4规格限制 | CSI规格限制     |
|------------------|---------------------------|-----------------|----------|-------------|
| lun              | OceanStor Dorado<br>6.1.2 | 512Ki~256<br>Ti | 50Ti     | 512Ki~50Ti  |
|                  | OceanStor Pacific系列 8.1.0 | 64Mi~512T<br>i  | 50Ti     | 64Mi~50Ti   |
| fs               | OceanStor Dorado<br>6.1.2 | 1Gi~32Pi        | N/A      | 1Gi~32Pi    |
|                  | OceanStor Pacific系列 8.1.0 | 1Ki~256Pi       | N/A      | 1Ki ~ 256Pi |

#### 表 6-10 PVC 容量的规格(xfs)

| volumeTyp<br>e类型 | 存储类型                          | 存储规格限<br>制      | xfs规格限制 | CSI规格限制     |
|------------------|-------------------------------|-----------------|---------|-------------|
| lun              | OceanStor Dorado<br>6.1.2     | 512Ki~256<br>Ti | 500Ti   | 512Ki~500Ti |
|                  | OceanStor Pacific系列 8.1.0     | 64Mi~512T<br>i  | 500Ti   | 64Mi~500Ti  |
| fs               | OceanStor Dorado<br>6.1.2     | 1Gi~32Pi        | N/A     | 1Gi~32Pi    |
|                  | OceanStor Pacific系<br>列 8.1.0 | 1Ki~256Pi       | N/A     | 1Ki~256Pi   |

### **步骤7** 执行以下命令,查看容量是否发生改变。

# kubectl get pvc

NAME STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS AGE mypvc Bound pvc-840054d3-1d5b-4153-b73f-826f980abf9e 120Gi RWX mysc 24s

----结束

## 6.3.3 (可选)克隆 PVC

当您需要克隆Kubernetes上已有的PVC时,执行此操作。

## 操作步骤

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意master节点。

**步骤2** 执行**vi** *clone.yaml*命令,创建 clone.yaml文件。 # vi clone.yaml

步骤3 配置clone.yaml文件。examples/clone.yaml模板文件如回显所示。请根据实际情况设置相关参数,并保存为yaml格式的文件,详细请参考表6-11。

kind: PersistentVolumeClaim apiVersion: v1 metadata: name: myclone spec: storageClassName: mysc dataSource: name: mypvc kind: PersistentVolumeClaim volumeMode: Filesystem accessModes: - ReadWriteOnce resources: requests: storage: 2Gi

### 表 6-11 参数说明

| 参数                                      | 说明                    | 备注   |
|---|-----------------------|--|
| metadata.na<br>me                       | 自定义的新<br>建PVC对象<br>名称 | 以Kubernetes v1.22.1为例,支持数字、小写字<br>母、中划线(- )和点(. )的组合,并且必须以字<br>母数字字符开头和结尾。   |
| spec.storageCl<br>assName               | StorageClas<br>s对象名称  | 填写 <b>6.1(条件必选)管理StorageClass</b> 创建的<br>StorageClass对象名称,与dataSource中的对象的<br>StorageClass需一致  |
| spec.dataSour<br>ce.name                | 源PVC对象<br>名称          | -  |
| spec.volumeM<br>ode                     | 卷模式                   | 可选, 支持Filesystem或Block, 默认为<br>Filesystem,必须与源PVC对象的volumeMode值<br>保持一致。该参数在创建Pod时生效,其中<br>Filesystem表示在PVC上创建一个文件系统访问存<br>储, Block表示使用裸卷的方式访问存储。  |
| spec.accessMo<br>des                    | 指定卷访问<br>模式           | 卷模式为Filesystem的LUN卷支持 ReadWriteOnce、ReadOnlyMany和 ReadWriteOncePod。 卷模式为Block的LUN卷支持ReadWriteOnce、 ReadOnlyMany、ReadWriteMany和 ReadWriteOncePod,当配置为ReadWriteMany 时,需要由Pod业务保证数据一致性。 文件系统卷支持ReadWriteOnce、 ReadOnlyMany、ReadWriteMany和 ReadWriteOncePod。  须知 由于ReadWriteOncePod访问模式是Kubernetes v1.22+版 本的alpha特性,需要先为集群开启ReadWriteOncePod 特性才能使用,详见9.7 开启ReadWriteOncePod功能门 |
| spec.resources<br>.requests.stora<br>ge | 指定待创建<br>新卷大小         | 不小于源PVC,格式为***Gi,单位为GiB  |

步骤4 执行以下命令,基于该yaml文件创建PVC。

# kubectl create -f clone.yaml

#### ----结束

## 6.3.4 (可选)从快照创建 PVC

当您需要对Kubernetes上已有的Snapshot创建新的PVC时,执行此操作。

## 前提条件

已经创建Snapshot,详细请参考6.5 (可选)管理快照。

## 操作步骤

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意master节点。

步骤2 执行vi restore.yaml命令,创建 restore.yaml文件。

# vi restore.yaml

步骤3 配置restore.yaml文件。examples/<Kubernetes版本号>/restore.yaml模板文件如回显所示。请根据实际情况设置相关参数,并保存为yaml格式的文件,详细请参考表6-12。

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
name: ***
spec:
storageClassName: ***
dataSource:
name: ***
kind: VolumeSnapshot
apiGroup: snapshot.storage.k8s.io
volumeMode: Filesystem
accessModes:
- ReadWriteOnce
resources:
requests:
storage: ***Gi
```

### 表 6-12 参数说明

| 参数                        | 说明                              | 备注  |
|---------------------------|---------------------------------|---|
| metadata.nam<br>e         | 自定义的新<br>建PVC对象<br>名称           | 以Kubernetes v1.22.1为例,支持数字、小写字<br>母、中划线(- )和点(. )的组合,并且必须以字<br>母数字字符开头和结尾。                        |
| spec.storageCla<br>ssName | StorageClas<br>s对象名称            | 填写 <b>6.1(条件必选)管理StorageClass</b> 创建的<br>StorageClass对象名称,与dataSource中快照的原<br>PVC的StorageClass需一致 |
| spec.dataSourc<br>e.name  | 源<br>VolumeSna<br>pshot对象名<br>称 | -   |

| 参数                                  | 说明            | 备注   |
|-------------------------------------|---------------|--|
| spec.volumeMo<br>de                 | 卷模式           | 可选, 支持Filesystem或Block, 默认为<br>Filesystem,必须与源PVC对象的volumeMode值<br>保持一致。该参数在创建Pod时生效,其中<br>Filesystem表示在PVC上创建一个文件系统访问存<br>储, Block表示使用裸卷的方式访问存储。    |
| spec.accessMod<br>es                | 指定卷访问<br>模式   | 卷模式为Filesystem的LUN卷支持<br>ReadWriteOnce、ReadOnlyMany和<br>ReadWriteOncePod。  |
|                                     |               | 卷模式为Block的LUN卷支持ReadWriteOnce、<br>ReadOnlyMany、ReadWriteMany和<br>ReadWriteOncePod,当配置为ReadWriteMany<br>时,需要由Pod业务保证数据一致性。                            |
|                                     |               | 文件系统卷支持ReadWriteOnce、<br>ReadOnlyMany、ReadWriteMany和<br>ReadWriteOncePod。  |
|                                     |               | <b>须知</b><br>由于ReadWriteOncePod访问模式是Kubernetes<br>v1.22+版本的alpha特性,需要先为集群开启<br>ReadWriteOncePod特性才能使用,详见 <b>9.7 开启</b><br><b>ReadWriteOncePod功能门</b> |
| spec.resources.r<br>equests.storage | 指定待创建<br>新卷大小 | 不小于源VolumeSnapshot,格式为***Gi,单位为<br>GiB   |

### 步骤4 执行以下命令,基于该yaml文件创建PVC

# kubectl create -f restore.yaml

----结束

## 6.3.5 删除 PVC

本章节介绍如何删除PVC。

### 操作步骤

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意master节点。

步骤2 执行以下命令,查询集群中的PVC。

# kubectl get pvc

NAME STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS AGE mypvc Bound pvc-840054d3-1d5b-4153-b73f-826f980abf9e 100Gi RWX mysc 12s

#### □ 说明

删除PVC前,如果PVC的状态是Pending,不建议直接删除,如果需要删除,请参考**10.7 删除** PVC前,PVC的状态为Pending。

步骤3 执行以下命令,删除PVC。例如删除名称为mypvc的PVC。

# kubectl delete pvc *mypvc* persistentvolumeclaim "*mypvc*" deleted

**步骤4** 执行以下命令,查询集群中的PVC。回显中不包含需要删除的PVC名称,则表示删除成功。

# kubectl get pvc No resources found in default namespace.

----结束

## 6.4 管理 Pod

## 6.4.1 创建 Pod

本章节介绍如何创建Pod。Pod是存储系统提供的原始存储容量池或存储功能集合,它们作为虚拟卷的容器,虚拟卷只能从存储容器中分配空间。通过该操作,您可以快速的获取指定的存储资源。

### 限制说明

建议每批次最多批量创建/删除100个Pod。

## 操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。
- **步骤2** 执行**vi** *pod.yaml*命令,创建pod.yaml文件。

# vi pod.yaml

- 步骤3 按l或Insert进入编辑状态,在pod.yaml文件中输入以下内容。修改完成后,按Esc,并输入:wq!,保存修改。
  - 如果步骤3中参数volumeMode设置为Filesystem, pod.yaml文件的模板如回显所示(您也可以参考软件包中examples/pod-example.yaml示例文件)。请根据实际情况设置相关参数,并保存为yaml格式的文件,详细参考表6-13。

```
kind: Pod
apiVersion: v1
metadata:
name: "mypod"
spec:
containers:
- name: "mycontainer"
image: "****"
volumeMounts:
- name: mypv
mountPath: "/mnt/path/in/container"
volumes:
- name: mypv
persistentVolumeClaim:
claimName: "mypvc"
```

表 6-13 参数说明

| 参数   | 说明                  | 备注  |
|--|---------------------|---|
| metadata.name  | 自定义的<br>Pod对象名<br>称 | 以Kubernetes v1.22.1为例,支持数字、小写字母、中划线(- )和点(.)<br>的组合,并且必须以字母数字字符开头和结尾。 |
| spec.containers.name                                 | 自定义的容<br>器名称        | 以Kubernetes v1.22.1为例,支持数字、小写字母、中划线(- )和点(.)<br>的组合,并且必须以字母数字字符开头和结尾。 |
| spec.containers.image                                | 指定容器镜像              | 镜像根据实际情况填写。   |
| spec.containers.image.<br>volumeMounts.mount<br>Path | 容器内卷的挂载路径           | -   |
| spec.volumes.persisten<br>tVolumeClaim.claimNa<br>me | PVC对象名<br>称         | 填写6.3.1 创建PVC创建的PVC对象名称。  |

如果步骤3中参数volumeMode设置为Block, pod.yaml文件的模板如回显所示(您也可以参考软件包中examples/pod-rbd-example.yaml示例文件)。请根据实际情况设置相关参数,并保存为yaml格式的文件,详细参考表6-14。

```
kind: Pod
apiVersion: v1
metadata:
name: "mypod"
spec:
containers:
- name: "mycontainer"
image: "****"
volumeDevices:
- name: mypv
devicePath: "/dev/xvda"
volumes:
- name: mypv
persistentVolumeClaim:
claimName: "mypvc"
```

### 表 6-14 参数说明

| 参数                        | 说明                  | 备注   |
|---------------------------|---------------------|--|
| metadata.name             | 自定义<br>的Pod对<br>象名称 | 以Kubernetes v1.22.1为例,支持数字、小写字母、中划线(- )和点(. )的组合,并且必须以字母数字字符开头和结尾。 |
| spec.containers.nam<br>e  | 自定义<br>的容器<br>名称    | 以Kubernetes v1.22.1为例,支持数字、小写字母、中划线(- )和点(. )的组合,并且必须以字母数字字符开头和结尾。 |
| spec.containers.ima<br>ge | 指定容<br>器镜像          | 镜像根据实际情况填写。  |

| 参数   | 说明                | 备注                       |
|--|-------------------|--------------------------|
| spec.containers.ima<br>ge.volumeDevices.d<br>evicePath | 容器内<br>卷的设<br>备路径 | -                        |
| spec.volumes.persist<br>entVolumeClaim.clai<br>mName   | PVC对<br>象名称       | 填写6.3.1 创建PVC创建的PVC对象名称。 |

步骤4 执行以下命令,基于该yaml文件创建Pod。

# kubectl create -f pod.yaml

步骤5 执行以下命令,查看已经创建的Pod。

# kubectl get pod

NAME READY STATUS RESTARTS AGE mypod 1/1 Running 0 37s

----结束

## 6.4.2 删除 Pod

本章节介绍如何删除Pod。

## 操作步骤

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意master节点。

步骤2 执行以下命令,查询集群中的Pod。

# kubectl get pod

NAME READY STATUS RESTARTS AGE mypod 1/1 Running 0 14h

步骤3 执行以下命令,删除Pod。例如删除名称为mypod的Pod。

# kubectl delete pod *mypod* pod "*mypod*" deleted

**步骤4** 执行以下命令,查询集群中的Pod。回显中不包含需要删除的Pod名称,则表示删除成功。

# kubectl get pod

No resources found in default namespace.

----结束

## 6.5 (可选)管理快照

## 6.5.1 安装 Snapshot 依赖组件服务

### Helm 安装操作步骤

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。

**步骤2** 执行kubectl api-resources | grep snapshot | awk '{print \$1}'命令,查看snapshot 相关资源服务安装详情。

如果回显如下,则说明快照依赖组件服务已安装,跳过本章节。如果缺少下列回显中任意一项服务,请继续执行后续步骤进行安装。

# kubectl api-resources | grep snapshot | awk '{print \$1}'

volumesnapshotclasses volumesnapshotcontents volumesnapshots

步骤3 进入Helm工程的目录下(若无法找到之前的Helm工程,则将组件包中的helm目录拷贝到master节点的任意目录下。参见3.3 上传软件包组件)

**步骤4** 备份安装CSI时使用的values.yaml文件,如果上次安装时使用的values.yaml文件无法 找到,可通过**helm get values** *helm-huawei-csi* **-n** *huawei-csi* **-a**命令查询。

# cp values.yaml values.yaml.bak

**步骤5** 执行**vi** *values.yaml*命令打开文件,将snapshot.enable参数配置为true,修改完成后,按**Esc**,并输入:**wq!**,保存修改。

# vi values.yaml

步骤6 执行helm upgrade helm-huawei-csi./ -n huawei-csi命令升级Helm chart。

其中,helm-huawei-csi为需要升级的chart名称,./为使用当前目录下Helm工程,huawei-csi为chart所在的命名空间。

# helm upgrade helm-huawei-csi ./ -n huawei-csi

Release "helm-huawei-csi" has been upgraded. Happy Helming!

NAME: helm-huawei-csi

LAST DEPLOYED: Thu Jun 9 07:58:15 2022

NAMESPACE: huawei-csi STATUS: deployed REVISION: 2 TEST SUITE: None

### ----结束

## 手动安装操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意master节点。
- **步骤2** 执行kubectl api-resources | grep snapshot | awk '{print \$1}'命令,查看snapshot 相关资源服务安装详情。

如果回显如下,则说明快照依赖组件服务已安装,跳过本章节。如果缺少下列回显中 任意一项服务,请继续执行后续步骤进行安装。

# kubectl api-resources | grep snapshot | awk '{print \$1}'

volumesnapshotclasses volumesnapshotcontents volumesnapshots

步骤3 将Kubernetes CSI组件包中的"deploy"目录拷贝到任意目录下,组件包路径请参考 3.3 上传软件包组件。

步骤4 进入deploy目录,执行以下命令,安装Snapshot依赖组件服务。

# kubectl apply -f huawei-csi-snapshot-crd.yaml

### ----结束

## 6.5.2 管理 VolumeSnapshotClass

## 6.5.2.1 创建 VolumeSnapshotClass

本章节介绍如何创建VolumeSnapshotClass。

## 操作步骤

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。

**步骤2** 执行**vi** *volume-snapshot-class.yaml*命令,创建*volume-snapshot-class.*yaml文件 # vi volume-snapshot-class.yaml

**步骤3** 按l或Insert进入编辑状态,在*volume-snapshot-class*.yaml文件中输入以下内容。修改完成后,按**Esc**,并输入:**wq!**,保存修改。

volume-snapshot-class.yaml文件的模板如回显所示(您也可以参考软件包中 examples/volume-snapshot-class.yaml示例文件 )。请根据实际情况设置相关参数,并保存为yaml格式的文件,详细参考<mark>表6-15</mark>

apiVersion: snapshot.storage.k8s.io/v1beta1 kind: VolumeSnapshotClass metadata: name: mysnapclass driver: csi.huawei.com

### 表 6-15 参数说明

deletionPolicy: Delete

| 参数                 | 说明  | 备注   |
|--------------------|---|--|
| metadata.n<br>ame  | 自定义的<br>VolumeSnapshotClass对象<br>名称                     | 以Kubernetes v1.22.1为例,支持数字、小写字母、中划线(-)和点(.)<br>的组合,并且必须以字母数字字符开<br>头和结尾。 |
| driver             | driver标识  | 固定填写"csi.huawei.com"   |
| deletionPoli<br>cy | VolumeSnapshot删除时,<br>处理<br>VolumeSnapshotContent策<br>略 | 必填, 可选值为Delete或Retain  |

步骤4 执行以下命令,基于该yaml文件创建VolumeSnapshotClass

# kubectl create -f volume-snapshot-class.yaml

步骤5 执行以下命令,查看已创建的VolumeSnapshotClass信息

# kubectl get volumesnapshotclass

NAME DRIVER DELETIONPOLICY AGE mysnapclass csi.huawei.com Delete 25s

----结束

## 6.5.2.2 删除 VolumeSnapshotClass

本章节介绍如何删除VolumeSnapshotClass。

## 操作步骤

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。

步骤2 执行以下命令, 查询集群中的VolumeSnapshotClass。

# kubectl get volumesnapshotclass

NAME DRIVER DELETIONPOLICY AGE mysnapclass csi.huawei.com Delete 52s

**步骤3** 执行以下命令, 删除VolumeSnapshotClass。例如删除名称为mysnapclass的 VolumeSnapshotClass。

# kubectl delete volumesnapshotclass *mysnapclass* volumesnapshotclass.snapshot.storage.k8s.io "*mysnapclass*" deleted

**步骤4** 执行以下命令, 查询集群中的VolumeSnapshotClass。回显中不包含需要删除的 VolumeSnapshotClass名称,则表示删除成功。

# kubectl get volumesnapshotclass No resources found in default namespace.

----结束

## 6.5.3 管理 VolumeSnapshot

## 6.5.3.1 创建 VolumeSnapshot

本章节介绍如何创建VolumeSnapshot。

## 操作步骤

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。

**步骤2** 执行**vi** *volume-snapshot.yaml*命令,创建volume-snapshot.yaml文件 # vi volume-snapshot.yaml

**步骤3** 按l或Insert进入编辑状态,在volume-snapshot.yaml文件中输入以下内容。修改完成后,按Esc,并输入:wq!,保存修改。

volume-snapshot.yaml文件的模板如回显所示(您也可以参考软件包中examples/volume-snapshot.yaml示例文件)。请根据实际情况设置相关参数,并保存为yaml格式的文件,详细参考表6-16

apiVersion: snapshot.storage.k8s.io/v1beta1

kind: VolumeSnapshot

metadata:

name: mysnapshot

spec:

 $volume Snapshot Class Name: \ mysnap class$ 

source:

persistentVolumeClaimName: mypvc

### 表 6-16 参数说明

| 参数                                     | 说明                             | 备注   |
|--|--------------------------------|--|
| metadata.name                          | 自定义的<br>VolumeSnapshot对象<br>名称 | 以Kubernetes v1.22.1为例,<br>支持数字、小写字母、中划线<br>(-)和点(.)的组合,并且<br>必须以字母数字字符开头和结<br>尾。 |
| spec.volumeSnapshotCl<br>assName       | VolumeSnapshotClass<br>对象名称    | 填写 <b>6.5.2.1 创建</b> VolumeSnapshotClass创建 的VolumeSnapshotClass对象 名称             |
| spec.source.persistentV olumeClaimName | 源PVC对象名称                       | 填写 <b>6.3.1 创建PVC</b> 创建的PVC<br>对象名称   |

步骤4 执行以下命令,基于该yaml文件创建VolumeSnapshot。

# kubectl create -f volume-snapshot.yaml

步骤5 执行以下命令,查看已创建的VolumeSnapshot信息。

# kubectl get volumesnapshot

NAME READYTOUSE SOURCEPVC SOURCESNAPSHOTCONTENT RESTORESIZE SNAPSHOTCLASS SNAPSHOTCONTENT CREATIONTIME AGE mysnapshot **true** mypvc 100Gi mysnapclass snapcontent-1009af0a-24c2-4435-861c-516224503f2d <invalid> 78s

----结束

## 6.5.3.2 删除 VolumeSnapshot

本章节介绍如何删除VolumeSnapshot。

## 操作步骤

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。

步骤2 执行以下命令,查询集群中的VolumeSnapshot。

# kubectl get volumesnapshot

NAME READYTOUSE SOURCEPVC SOURCESNAPSHOTCONTENT RESTORESIZE SNAPSHOTCLASS SNAPSHOTCONTENT CREATIONTIME AGE mysnapshot true mypvc 100Gi mysnapclass snapcontent-1009af0a-24c2-4435-861c-516224503f2d <invalid> 78s

**步骤3** 执行以下命令, 删除VolumeSnapshot。例如删除名称为mysnapshot的 VolumeSnapshot。

# kubectl delete volumesnapshot *mysnapshot* volumesnapshot.snapshot.storage.k8s.io "*mysnapshot*" deleted

**步骤4** 执行以下命令,查询集群中的VolumeSnapshot。回显中不包含需要删除的 VolumeSnapshot名称,则表示删除成功。

# kubectl get volumesnapshot No resources found in default namespace.

----结束

# **了** 高级特性

本章节介绍如何配置华为存储的高级特性。

- 7.1 配置多个后端
- 7.2 指定后端创建PVC
- 7.3 指定存储池创建PVC
- 7.4 配置ALUA特性
- 7.5 配置存储拓扑感知
- 7.6 企业存储高级特性
- 7.7 分布式存储高级特性

## 7.1 配置多个后端

Huawei CSI是可以支持配置多个后端。当您需要配置多个后端时,执行此操作。

## 7.1.1 通过 Helm 配置多个后端

## 操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意master节点。
- 步骤2 进入Helm工程的目录下,如果无法找到之前的Helm工程,则将组件包中的helm目录 拷贝到master节点的任意目录下,组件包路径请参考3.3 上传软件包组件。
- **步骤3** 备份安装CSI时使用的values.yaml文件,如果上次安装时使用的values.yaml文件无法找到,可通过**helm get values** *helm-huawei-csi* **-n** *huawei-csi* **-a**命令查询。
  - # cp values.yaml values.yaml.bak
- **步骤4** 执行**vi** *values.yaml*命令打开文件,按需求配置多个后端,示例如下所示。修改完成后,按**Esc**,并输入:**wq!**,保存修改。

```
backends:
- storage: "oceanstor-san"
name: "***"
urls:
- "https://*.*.*:8088"
```

```
pools:
_ "***"

parameters:
  protocol: "iscsi"
  portals:
_ "****"

- storage: "oceanstor-nas"
  name: "***"
  urls:
_ "https://*.*.*.*8088"
  pools:
_ "***"

parameters:
  protocol: "nfs"
  portals:
_ "****"
```

步骤5 执行helm upgrade helm-huawei-csi ./ -n huawei-csi -f values.yaml命令升级Helm chart。升级命令将更新Deployment、daemonset和RBAC资源,但是secret资源不会更新,如果后端信息发生更改,则必须手动更新secret,更新操作请参考9.1 更新CSI上配置的存储用户名或密码。

其中,helm-huawei-csi为自定义的chart名称,huawei-csi为自定义的命名空间。

```
# helm upgrade helm-huawei-csi / -n huawei-csi
Release "helm-huawei-csi" has been upgraded. Happy Helming!
NAME: helm-huawei-csi
LAST DEPLOYED: Thu Jun 9 07:58:15 2022
NAMESPACE: huawei-csi
STATUS: deployed
REVISION: 2
TEST SUITE: None
```

----结束

## 7.1.2 手动配置多个后端

### 操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。
- 步骤2 配置huawei-csi-configmap.yaml文件, huawei-csi-configmap.yaml文件的模板如回显所示。请根据实际情况设置相关参数,并保存为yaml格式的文件。

多个后端间使用","进行分割,每个后端内容可以参考**4.2.1 对接企业存储配置**或 **4.2.2 对接分布式存储配置**。

```
}
]
}
```

**步骤3** 执行**kubectl create -f** *huawei-csi-configmap.yaml*命令,创建huawei-csi-configmap。

# kubectl create -f huawei-csi-configmap.yaml

```
# kubectl get configmap -n huawei-csi | grep huawei-csi-configmap huawei-csi-configmap 1 5s
```

步骤5 启动huawei-csi服务,详细请参见4.2.3 启动huawei-csi服务。

----结束

## 7.2 指定后端创建 PVC

配置多后端场景后,可以通过以下操作为指定后端创建PVC。

## 操作步骤

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意master节点。

**步骤2** 执行**vi** *StorageClass.yaml*命令,修改yaml文件。按**I**或**Insert**进入编辑状态,修改下列字段的参数。修改完成后,按**Esc**,并输入:**wq!**,保存修改。

- "parameters"配置项下添加backend配置项。
- metadata.name参数值是自定义的StorageClass对象名称。 以Kubernetes v1.22.1为例,支持数字、小写字母、中划线(-)和点(.)的组合,并且必须以字母数字字符开头和结尾。
- parameters.backend参数值是huawei-csi-configmap.yaml后端配置中的某个后端 名称。

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
name: "***"
provisioner: "csi.huawei.com"
parameters:
...
backend: "***"
```

步骤3 执行以下命令,基于该yaml文件创建StorageClass。

# kubectl create -f StorageClass.yaml

----结束

## 7.3 指定存储池创建 PVC

配置多存储池场景后,可以通过以下操作为指定存储池创建PVC。

### 操作步骤

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。

**步骤2** 执行**vi** *StorageClass.yaml*命令,修改yaml文件。按**l**或**Insert**进入编辑状态,修改下列字段的参数。修改完成后,按**Esc**,并输入:**wg!**,保存修改。

- metadata.name参数值是自定义的StorageClass对象名称。
   以Kubernetes v1.22.1为例,支持数字、小写字母、中划线(-)和点(.)的组合,并且必须以字母数字字符开头和结尾。
- "parameters"配置项下添加pool配置项。
- "pool"参数值是huawei-csi-configmap.yaml后端配置中的某个存储池名称。

#### □□ 说明

使用该StorageClass新创建的卷即会在指定存储池上创建,已经存在的PVC不会更改存储池信息。

kind: StorageClass apiVersion: storage.k8s.io/v1 metadata: name: "\*\*\*" provisioner: "csi.huawei.com" parameters: ... pool: "\*\*\*"

步骤3 执行以下命令,基于该yaml文件创建StorageClass。

# kubectl create -f StorageClass.yaml

----结束

## 7.4 配置 ALUA 特性

## 7.4.1 通过 Helm 配置 ALUA 特性

本章节介绍通过Helm配置ALUA特性。

## 7.4.1.1 通过 Helm 配置 OceanStor V3/V5 系列和 OceanStor Dorado V3 系列 ALUA

### 操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。
- 步骤2 进入Helm工程的目录下,如果无法找到之前的Helm工程,则将组件包中的helm目录 拷贝到master节点的任意目录下,组件包路径请参考3.3 上传软件包组件。
- **步骤3** 备份安装CSI时使用的values.yaml文件,如果上次安装时使用的values.yaml文件无法 找到,可通过**helm get values** *helm-huawei-csi* **-n** *huawei-csi* **-a**命令查询。 # cp values.yaml values.yaml.bak
- **步骤4** 执行**vi** *values.yaml*命令,修改yaml文件。按**l**或**lnsert**进入编辑状态,修改相关参数。 修改完成后,按**Esc**,并输入:**wq!**,保存修改。

### "parameters"配置项下增加ALUA配置,参数配置请参见表7-1。

```
- storage: "oceanstor-san"
 name: "storage"
 urls:
  - "https://*.*.*:8088"
- "https://*.*.*:8088"
 pools:
  - "***"
- "***"
 parameters:
  protocol: "iscsi"
  portals:
  ALUA:
    HostName1:
     MULTIPATHTYPE: "*"
     FAILOVERMODE: "*"
     SPECIALMODETYPE: "*"
     PATHTYPE: "*"
    HostName2:
     MULTIPATHTYPE: "*"
     FAILOVERMODE: "*"
     SPECIALMODETYPE: "*"
     PATHTYPE: "*"
```

### 表 7-1 ALUA 参数说明

| 参数名             | 参数描述  | 备注   |
|-----------------|---|--|
| HostName        | HostName的值为worker节<br>点的主机名,如<br>HostName1、<br>HostName2。       | 主机名通常使用<br>cat /etc/hostname 可获<br>取,支持正则表达式方<br>式匹配(配置规则和优先<br>级详见说明) |
| MULTIPATHTYPE   | 多路径类型。取值为:  | -  |
| FAILOVERMODE    | 启动器的切换模式。取值为:   | 当使用第三方多路径时<br>该参数才需要指定,所<br>有OceanStor V5型号均<br>不支持旧版本ALUA。            |
| SPECIALMODETYPE | 启动器的特殊模式类型。取值为:  • 0: 特殊模式0  • 1: 特殊模式1  • 2: 特殊模式2  • 3: 特殊模式3 | 当启动器的切换模式为<br>特殊模式ALUA时该参数<br>才需要指定。                                   |

| 参数名      | 参数描述                                 | 备注                      |
|----------|--------------------------------------|-------------------------|
| PATHTYPE | 启动器的路径类型。取值为:  • 0: 优选路径  • 1: 非优选路径 | 当使用第三方多路径时<br>该参数才需要指定。 |

#### □ 说明

- 针对不同的操作系统,ALUA配置可能有所不同。进入https://support.huawei.com/enterprise/zh/index.html,在搜索输入框中输入"主机连通性指南",点击搜索。在搜索结果中,选择对应操作系统的主机连通性指南,根据指南的推荐配置进行ALUA配置。
- 已经发放的Pod的节点不会主动更改ALUA信息,需要通过在该节点重新发放Pod才会变更主机ALUA配置。
- HostName为正则表达式,配置方式参考《正则表达式》

当HostName= "\*"时,表示的是通用配置,该条配置对任意主机名的主机生效, 其余 HostName的配置均视为一般配置。配置HostName时,主机连接数有限制,限制值参考<mark>存 储规格查询</mark>,规格项:最大iSCSI连接数/控制框。主机连接数小于等于限制值,建议使用通 用配置;主机连接数大于限制值,使用一般配置。

#### 配置策略的规则:

- 优先级: 主机名一般配置 > 主机名通用配置(详见11.1 OceanStor V3/V5系列和 OceanStor Dorado V3系列ALUA特性配置策略样例例1)
- 在一般配置中,采用满足配置策略的第一条ALUA配置段(详见11.1 OceanStor V3/V5 系列和OceanStor Dorado V3系列ALUA特性配置策略样例例2)
- 在一般配置中,如果需要精确匹配某个主机,参考11.1 OceanStor V3/V5**系列和** OceanStor Dorado V3**系列ALUA特性配置策略样例**例3
- OceanStor V3/V5系列和OceanStor Dorado V3系列使用此种配置方式,详细参数请参见表 7-2

表 7-2 OceanStor V3/V5 系列和 OceanStor Dorado V3 系列推荐 ALUA 参数配置

| 使用场景   | 主机类型                 | 是否为优选路<br>径存储 | ALUA配置推荐值  |
|--------|----------------------|---------------|--|
| 双活存储场景 | CentOS/RHEL系列的主机     | 是             | ALUA="1" FAILOVERMODE="3" SPECIALMODETYPE="0" PATHTYPE="0" |
|        |                      | 否             | ALUA="1" FAILOVERMODE="3" SPECIALMODETYPE="0" PATHTYPE="1" |
|        | SUSE/Debian系<br>列的主机 | 是             | ALUA="1"<br>FAILOVERMODE="1"<br>PATHTYPE="0"               |
|        |                      | 否             | ALUA="1"<br>FAILOVERMODE="1"<br>PATHTYPE="1"               |

| 使用场景        | 主机类型                 | 是否为优选路<br>径存储 | ALUA配置推荐值  |
|-------------|----------------------|---------------|--|
| 非双活存储<br>场景 |                      | N/A           | ALUA="1" FAILOVERMODE="3" SPECIALMODETYPE="0" PATHTYPE="0" |
|             | SUSE/Debian系<br>列的主机 | N/A           | ALUA="1"<br>FAILOVERMODE="1"<br>PATHTYPE="0"               |

步骤5 执行helm upgrade helm-huawei-csi ./ -n huawei-csi -f values.yaml命令升级Helm chart。升级命令将更新Deployment、daemonset和RBAC资源,但是secret资源不会更新,如果后端信息发生更改,则必须手动更新secret,更新操作请参考9.1 更新CSI上配置的存储用户名或密码。

其中,helm-huawei-csi为自定义的chart名称,huawei-csi为自定义的命名空间。

# helm upgrade helm-huawei-csi ./ -n huawei-csi

Release "helm-huawei-csi" has been upgraded. Happy Helming!

NAME: helm-huawei-csi

LAST DEPLOYED: Thu Jun 9 07:58:15 2022

NAMESPACE: huawei-csi STATUS: deployed REVISION: 2 TEST SUITE: None

----结束

### 7.4.1.2 通过 Helm 配置 OceanStor Dorado 6.x ALUA

## 操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。
- 步骤2 进入Helm工程的目录下,如果无法找到之前的Helm工程,则将组件包中的helm目录 拷贝到master节点的任意目录下,组件包路径请参考3.3 上传软件包组件。
- **步骤3** 备份安装CSI时使用的values.yaml文件,如果上次安装时使用的values.yaml文件无法找到,可通过**helm get values** *helm-huawei-csi* **-n** *huawei-csi* **-a**命令查询。
  # cp values.yaml values.yaml.bak
- **步骤4** 执行**vi** *huawei-csi-configmap.yaml*命令,修改yaml文件。按**l**或**Insert**进入编辑状态,修改相关参数。修改完成后,按**Esc**,并输入**:wq!**,保存修改。
  - "parameters"配置项下增加ALUA配置,参数配置请参见表7-3。

```
backends:
- storage: "oceanstor-san"
name: "storage"
urls:
- "https://*.*.*.8088"
- "https://*.*.*:8088"
pools:
- "***"
- "***"
parameters:
protocol: "iscsi"
```

```
portals:
_ "****"
ALUA:
HostName1:
accessMode: "*"
hyperMetroPathOptimized: "*"
HostName2:
accessMode: "*"
hyperMetroPathOptimized: "*"
```

#### 表 7-3 ALUA 参数说明

| 参数名                         | 参数描述  | 备注  |
|-----------------------------|---|---|
| <hostname></hostname>       | HostName的值为worker<br>节点的主机名,如<br>HostName1、<br>HostName2。 | 主机名通常使用 cat /etc/<br>hostname 可获取,支持正则<br>表达式方式匹配(配置规则和<br>优先级详见说明) |
| accessMode                  | 主机访问模式。取值为:  • 0: 均衡模式  • 1: 非对称模式                        |   |
| hyperMetroPathOpti<br>mized | 双活场景下,主机在当前<br>阵列的路径是否优选。取<br>值为:                         | 当主机访问模式设置为非对称<br>模式时,才需要配置该参数。                                      |
|                             | ● 1: 是  |   |
|                             | ● 0: 否  |   |

#### 山 说明

- 针对不同的操作系统,ALUA配置可能有所不同。进入https://support.huawei.com/enterprise/zh/index.html,在搜索输入框中输入"主机连通性指南",点击搜索。在搜索结果中,选择对应操作系统的主机连通性指南,根据指南的推荐配置进行ALUA配置。
- 已经发放的Pod的节点不会主动更改ALUA 信息,需要通过在该节点重新发放Pod才会变更主机ALUA配置。
- HostName为正则表达式,配置方式参考《正则表达式》

当HostName= "\*"时,表示的是通用配置,该条配置对任意主机名的主机生效, 其余 HostName的配置均视为一般配置。配置HostName时,主机连接数有限制,限制值参考<mark>存 储规格查询</mark>,规格项:最大iSCSI连接数/控制框。主机连接数小于等于限制值,建议使用通 用配置;主机连接数大于限制值,使用一般配置。

#### 配置策略的规则:

- 优先级: 主机名一般配置 > 主机名通用配置 (详见11.2 OceanStor Dorado 6.x ALUA特性配置策略样例例1)
- 在一般配置中,采用满足配置策略的第一条ALUA配置段(详见11.2 OceanStor Dorado 6.x ALUA特性配置策略样例例2)
- 在一般配置中,如果需要精确匹配某个主机,参考11.2 OceanStor Dorado 6.x ALUA 特性配置策略样例例3
- 当主机仅使用OceanStor Dorado 6.x固态存储,详细参数请参见表7-4。
- 当同一主机先使用OceanStor融合存储,再新增OceanStor Dorado 6.x固态存储时,详细参数请参见表7-5。

| 表 7.  | 4 Ocea | nStor   | Dorado | 6 x | 固态存储推荐        | ALUA | 参数配置 |
|-------|--------|---------|--------|-----|---------------|------|------|
| 双 / - | 4 000  | IIISLOI | Dolado | U.X | 101かぶ1十1泊1比1子 | ALUA | 多女的巨 |

| 使用场景        | 主机类型                                     | 主机访问模式            | ALUA配置推荐值   |
|-------------|--|-------------------|---|
| 双活存储场       | CentOS/                                  | 负载均衡模式            | 不需要配置ALUA   |
| 景           | RHEL/SUSE/<br>Debian系列的<br>主机            | 非对称模式+优选<br>路径存储  | ACCESSMODE="1"<br>HYPERMETROPATHOPTIMIZ<br>ED="1" |
|             |  | 非对称模式+非优<br>选路径存储 | ACCESSMODE="1"<br>HYPERMETROPATHOPTIMIZ<br>ED="0" |
| 非双活存储<br>场景 | CentOS/<br>RHEL/SUSE/<br>Debian系列的<br>主机 | N/A               | 不需要配置ALUA   |

表 7-5 OceanStor V3/V5、OceanStor Dorado V3、OceanStor Dorado 6.x 存储共存 场景推荐 ALUA 参数配置

| 使用场景        | 主机类型                             | 主机访问模式                | ALUA配置推荐值   |
|-------------|----------------------------------|-----------------------|---|
| 双活存储<br>场景  | CentOS/RHEL/SUSE/<br>Debian系列的主机 | 负载均衡模式                | ACCESSMODE="1"<br>HYPERMETROPATHOPTIMIZ<br>ED="1" |
|             |                                  | 非对称模式<br>+优选路径存<br>储  | ACCESSMODE="1"<br>HYPERMETROPATHOPTIMIZ<br>ED="1" |
|             |                                  | 非对称模式<br>+非优选路径<br>存储 | ACCESSMODE="1"<br>HYPERMETROPATHOPTIMIZ<br>ED="0" |
| 非双活存<br>储场景 | CentOS/RHEL/SUSE/<br>Debian系列的主机 | N/A                   | ACCESSMODE="1"<br>HYPERMETROPATHOPTIMIZ<br>ED="1" |

步骤5 执行helm upgrade helm-huawei-csi ./ -n huawei-csi -f values.yaml命令升级Helm chart。升级命令将更新Deployment、daemonset和RBAC资源,但是secret资源不会 更新,如果后端信息发生更改,则必须手动更新secret,更新操作请参考9.1 更新CSI 上配置的存储用户名或密码。

其中,helm-huawei-csi为自定义的chart名称,huawei-csi为自定义的命名空间。

# helm upgrade helm-huawei-csi / -n huawei-csi Release "helm-huawei-csi" has been upgraded. Happy Helming!

NAME: helm-huawei-csi

LAST DEPLOYED: Thu Jun 9 07:58:15 2022

NAMESPACE: huawei-csi STATUS: deployed **REVISION: 2** TEST SUITE: None

----结束

## 7.4.1.3 通过 Helm 配置分布式存储 ALUA

### 操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。
- 步骤2 进入Helm工程的目录下,如果无法找到之前的Helm工程,则将组件包中的helm目录 拷贝到master节点的任意目录下,组件包路径请参考3.3 上传软件包组件。
- **步骤3** 备份安装CSI时使用的values.yaml文件,如果上次安装时使用的values.yaml文件无法找到,可通过**helm get values** *helm-huawei-csi* **-n** *huawei-csi* **-a**命令查询。

# cp values.yaml values.yaml.bak

**步骤4** 执行**vi** *values.yaml*命令,修改yaml文件。按**I**或**Insert**进入编辑状态,修改相关参数。 修改完成后,按**Esc**,并输入:**wq!**,保存修改。

"parameters"配置项下增加ALUA配置,参数配置请参见表7-6。

```
- storage: "oceanstor-san"
name: "storage"
 - "https://*.*.*:8088"
  - "https://*.*.*:8088"
pools:
 - "***"
parameters:
 protocol: "iscsi"
  portals:
   _ ''* * * *'
  ALUA:
   HostName1:
    switchoverMode: "*"
    pathType: "*"
   HostName2:
    switchoverMode: "*"
    pathType: "*"
```

### 表 7-6 ALUA 参数说明

| 参数名                   | 参数描述  | 备注  |
|-----------------------|---|---|
| <hostname></hostname> | HostName的值为worker节<br>点的主机名,如<br>HostName1、<br>HostName2。 | 主机名通常使用 cat /etc/<br>hostname 可获取,支持正<br>则表达式方式匹配(配置规则<br>和优先级详见说明) |
| switchoverMode        | 切换模式。取值为:   | -   |
|                       | ● Disable_alua:禁用<br>ALUA                                 |   |
|                       | ● Enable_alua: 启用<br>ALUA                                 |   |

| 参数名      | 参数描述                          | 备注 |
|----------|-------------------------------|----|
| pathType | 路径类型。取值为:                     | -  |
|          | ● optimal_path: 优选路<br>径      |    |
|          | ● non_optimal_path: 非<br>优选路径 |    |

#### □ 说明

- 仅支持分布式存储iSCSI场景。
- 已经发放的Pod的节点不会主动更改ALUA 信息,需要通过在该节点重新发放Pod才会变更主机ALUA配置。
- 当HostName为正则表达式,配置方式参考《正则表达式》
   当HostName= "\*"时,表示的是通用配置,该条配置对任意主机名的主机生效,其余HostName的配置均视为一般配置。

#### 配置策略的规则:

- 优先级: 主机名一般配置 > 主机名通用配置( 详见**11.3 分布式存储ALUA特性配置策** 略样**例**例1)
- 在一般配置中,采用满足配置策略的第一条ALUA配置段(详见**11.3 分布式存储ALUA** 特性配置策略样例例2)
- 在一般配置中,如果需要精确匹配某个主机,参考**11.3 分布式存储ALUA特性配置策略** 样**例**例3

步骤5 执行helm upgrade helm-huawei-csi./ -n huawei-csi-f values.yaml命令升级Helm chart。升级命令将更新Deployment、daemonset和RBAC资源,但是secret资源不会更新,如果后端信息发生更改,则必须手动更新secret,更新操作请参考9.1 更新CSI上配置的存储用户名或密码。

其中,helm-huawei-csi为自定义的chart名称,huawei-csi为自定义的命名空间。

# helm upgrade helm-huawei-csi ./ -n huawei-csi

Release "helm-huawei-csi" has been upgraded. Happy Helming!

NAME: helm-huawei-csi

LAST DEPLOYED: Thu Jun 9 07:58:15 2022

NAMESPACE: huawei-csi STATUS: deployed REVISION: 2 TEST SUITE: None

#### ----结束

## 7.4.2 手动配置 ALUA 特性

本章节介绍如何手动配置ALUA特性。

## 7.4.2.1 配置 OceanStor V3/V5 系列和 OceanStor Dorado V3 系列 ALUA

在对接块存储时,如果有使用多路径,可以通过本章节配置ALUA。

### 操作步骤

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。

**步骤2** 执行**vi** *huawei-csi-configmap.yaml*命令,修改yaml文件。按**l**或**Insert**进入编辑状态,修改相关参数。修改完成后,按**Esc**,并输入**:wq!**,保存修改。

多个后端间使用","进行分割,每个后端内容可以参考**4.2.1 对接企业存储配置**或 **4.2.2 对接分布式存储配置**。

"parameters"配置项下增加ALUA配置,参数配置请参见表7-7。

```
{
  "backends": [
  {
    "storage": "oceanstor-san",
    ...
    "parameters": {..., "ALUA": {"<HostName>": {"MULTIPATHTYPE": "*", "FAILOVERMODE": "*",
    "SPECIALMODETYPE": "*", "PATHTYPE": "*"}, "<HostName>": {...}}}
  }
}
```

### 表 7-7 ALUA 参数说明

| 参数名                   | 参数描述  | 备注   |  |
|-----------------------|---|--|--|
| <hostname></hostname> | HostName的值为worker节<br>点的主机名   | 主机名通常使用<br>cat /etc/hostname 可获<br>取,支持正则表达式方<br>式匹配(配置规则和优先<br>级详见说明) |  |
| MULTIPATHTYPE         | 多路径类型。取值为:  • 0: 默认  • 1: 第三方多路径                                      | -  |  |
| FAILOVERMODE          | 启动器的切换模式。取值为:  • 0: 旧版本ALUA  • 1: 通用ALUA  • 2: 不使用ALUA  • 3: 特殊模式ALUA | 当使用第三方多路径时<br>该参数才需要指定,所<br>有OceanStor V5型号均<br>不支持旧版本ALUA。            |  |
| SPECIALMODETYPE       | 启动器的特殊模式类型。取值为:  • 0: 特殊模式0  • 1: 特殊模式1  • 2: 特殊模式2  • 3: 特殊模式3       | 当启动器的切换模式为<br>特殊模式ALUA时该参数<br>才需要指定。                                   |  |
| PATHTYPE              | 启动器的路径类型。取值为:  • 0: 优选路径  • 1: 非优选路径                                  | 当使用第三方多路径时<br>该参数才需要指定。  |  |

#### □ 说明

- 针对不同的操作系统,ALUA配置可能有所不同。进入https://support.huawei.com/enterprise/zh/index.html,在搜索输入框中输入"主机连通性指南",点击搜索。在搜索结果中,选择对应操作系统的主机连通性指南,根据指南的推荐配置进行ALUA配置。
- 已经发放的Pod的节点不会主动更改ALUA信息,需要通过在该节点重新发放Pod才会变更主机ALUA配置。
- HostName为正则表达式,配置方式参考《正则表达式》

当HostName= "\*"时,表示的是通用配置,该条配置对任意主机名的主机生效, 其余 HostName的配置均视为一般配置。配置HostName时,主机连接数有限制,限制值参考**存储规格查询**,规格项:最大iSCSI连接数/控制框。主机连接数小于等于限制值,建议使用通用配置;主机连接数大于限制值,使用一般配置。

#### 配置策略的规则:

- 优先级: 主机名一般配置 > 主机名通用配置( 详见**11.1 OceanStor V3/V5<u>系</u>列和** OceanStor Dorado V3**系列ALUA特性配置策略样例**例1 )
- 在一般配置中,采用满足配置策略的第一条ALUA配置段(详见11.1 OceanStor V3/V5 系列和OceanStor Dorado V3系列ALUA特性配置策略样例例2)
- 在一般配置中,如果需要精确匹配某个主机,参考11.1 OceanStor V3/V5<mark>系列和</mark> OceanStor Dorado V3**系列**ALUA**特性配置策略样例**例3
- OceanStor V3/V5系列和OceanStor Dorado V3系列使用此种配置方式,详细参数请参见表7-8

表 7-8 OceanStor V3/V5 系列和 OceanStor Dorado V3 系列推荐 ALUA 参数配置

| 使用场景                 | 主机类型                 | 是否为优选路<br>径存储                                | ALUA配置推荐值  |
|----------------------|----------------------|--|--|
| 双活存储场景               | CentOS/RHEL系列的主机     | 是  | ALUA="1" FAILOVERMODE="3" SPECIALMODETYPE="0" PATHTYPE="0" |
|                      |                      | 否  | ALUA="1" FAILOVERMODE="3" SPECIALMODETYPE="0" PATHTYPE="1" |
| SUSE/Debian系<br>列的主机 | 是                    | ALUA="1"<br>FAILOVERMODE="1"<br>PATHTYPE="0" |  |
|                      |                      | 否  | ALUA="1"<br>FAILOVERMODE="1"<br>PATHTYPE="1"               |
| 非双活存储<br>场景          | CentOS/RHEL系<br>列的主机 | N/A  | ALUA="1" FAILOVERMODE="3" SPECIALMODETYPE="0" PATHTYPE="0" |
|                      | SUSE/Debian系<br>列的主机 | N/A  | ALUA="1"<br>FAILOVERMODE="1"<br>PATHTYPE="0"               |

**步骤3** 执行**kubectl create -f** *huawei-csi-configmap.yaml*命令,创建huawei-csi-configmap。

# kubectl create -f huawei-csi-configmap.yaml

```
# kubectl get configmap -n huawei-csi | grep huawei-csi-configmap huawei-csi-configmap 1 5s
```

步骤5 启动huawei-csi服务,详细请参见4.2.3 启动huawei-csi服务。

----结束

### 7.4.2.2 配置 OceanStor Dorado 6.x ALUA

在对接块存储时,如果有使用多路径,可以通过本章节配置ALUA。

## 操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。
- **步骤2** 执行**vi** *huawei-csi-configmap.yaml*命令,修改yaml文件。按**l**或**lnsert**进入编辑状态,修改相关参数。修改完成后,按**Esc**,并输入**:wq!**,保存修改。

多个后端间使用","进行分割,每个后端内容可以参考**4.2.1 对接企业存储配置**或 **4.2.2 对接分布式存储配置**。

"parameters"配置项下增加ALUA配置,参数配置请参见表7-9。

```
{
  "backends": [
  {
    "storage": "oceanstor-san",
    ...
    "parameters": {..., "ALUA": {"<HostName>": {"accessMode": "*", "hyperMetroPathOptimized": "*"},
  "<HostName>": {...}}}
  }
}
}
```

#### 表 7-9 ALUA 参数说明

| 参数名                   | 参数描述                               | 备注  |
|-----------------------|------------------------------------|---|
| <hostname></hostname> | HostName的值为worker<br>节点的主机名        | 主机名通常使用 cat /etc/<br>hostname 可获取,支持正则<br>表达式方式匹配(配置规则和<br>优先级详见说明) |
| accessMode            | 主机访问模式。取值为:  • 0: 均衡模式  • 1: 非对称模式 |   |

| 参数名                         | 参数描述                              | 备注                             |
|-----------------------------|-----------------------------------|--------------------------------|
| hyperMetroPathOpti<br>mized | 双活场景下,主机在当前<br>阵列的路径是否优选。取<br>值为: | 当主机访问模式设置为非对称<br>模式时,才需要配置该参数。 |
|                             | ● 1: 是                            |                                |
|                             | ● 0: 否                            |                                |

### □ 说明

- 针对不同的操作系统,ALUA配置可能有所不同。进入https://support.huawei.com/enterprise/zh/index.html,在搜索输入框中输入"主机连通性指南",点击搜索。在搜索结果中,选择对应操作系统的主机连通性指南,根据指南的推荐配置进行ALUA配置。
- 已经发放的Pod的节点不会主动更改ALUA 信息,需要通过在该节点重新发放Pod才会变更主机ALUA配置。
- HostName为正则表达式,配置方式参考《正则表达式》

当HostName="\*"时,表示的是通用配置,该条配置对任意主机名的主机生效, 其余 HostName的配置均视为一般配置。配置HostName时,主机连接数有限制,限制值参考<mark>存 储规格查询</mark>,规格项:最大iSCSI连接数/控制框。主机连接数小于等于限制值,建议使用通 用配置;主机连接数大于限制值,使用一般配置。

#### 配置策略的规则:

- 优先级: 主机名一般配置 > 主机名通用配置 (详见11.2 OceanStor Dorado 6.x ALUA特性配置策略样例例1)
- 在一般配置中,采用满足配置策略的第一条ALUA配置段(详见11.2 OceanStor Dorado 6.x ALUA特性配置策略样例例2)
- 在一般配置中,如果需要精确匹配某个主机,参考11.2 OceanStor Dorado 6.x ALUA 特性配置策略样例例3
- 当主机仅使用OceanStor Dorado 6.x固态存储,详细参数请参见表7-10。
- 当同一主机先使用OceanStor融合存储,再新增OceanStor Dorado 6.x固态存储时,详细参数请参见表7-11。

#### 表 7-10 OceanStor Dorado 6.x 固态存储推荐 ALUA 参数配置

| 使用场景        | 主机类型  | 主机访问模式            | ALUA配置推荐值   |
|-------------|---|-------------------|---|
| 双活存储场       | 双活存储场<br>景 CentOS/<br>RHEL/SUSE/<br>Debian系列的<br>主机 | 负载均衡模式            | 不需要配置ALUA   |
| 京           |   | 非对称模式+优选<br>路径存储  | ACCESSMODE="1"<br>HYPERMETROPATHOPTIMIZ<br>ED="1" |
|             |   | 非对称模式+非优<br>选路径存储 | ACCESSMODE="1"<br>HYPERMETROPATHOPTIMIZ<br>ED="0" |
| 非双活存储<br>场景 | CentOS/<br>RHEL/SUSE/<br>Debian系列的<br>主机            | N/A               | 不需要配置ALUA   |

| 表 7-11 OceanStor V3/V5、 | OceanStor Dorado V3、 | OceanStor Dorado | 6.x 存储共存 |
|-------------------------|----------------------|------------------|----------|
| 场景推荐 ALUA 参数配置          |                      |                  |          |

| 使用场景        | 主机类型                             | 主机访问模式                | ALUA配置推荐值   |
|-------------|----------------------------------|-----------------------|---|
| 双活存储<br>场景  |                                  | 负载均衡模式                | ACCESSMODE="1"<br>HYPERMETROPATHOPTIMIZ<br>ED="1" |
|             |                                  | 非对称模式<br>+优选路径存<br>储  | ACCESSMODE="1"<br>HYPERMETROPATHOPTIMIZ<br>ED="1" |
|             |                                  | 非对称模式<br>+非优选路径<br>存储 | ACCESSMODE="1"<br>HYPERMETROPATHOPTIMIZ<br>ED="0" |
| 非双活存<br>储场景 | CentOS/RHEL/SUSE/<br>Debian系列的主机 | N/A                   | ACCESSMODE="1"<br>HYPERMETROPATHOPTIMIZ<br>ED="1" |

**步骤3** 执行**kubectl create -f** *huawei-csi-configmap.yaml*命令,创建huawei-csi-configmap。

# kubectl create -f huawei-csi-configmap.yaml

```
# kubectl get configmap -n huawei-csi | grep huawei-csi-configmap huawei-csi-configmap 1 5s
```

步骤5 启动huawei-csi服务,详细请参见4.2.3 启动huawei-csi服务。

----结束

## 7.4.2.3 配置分布式存储 ALUA

在对接块存储时,如果有使用多路径,可以通过本章节配置ALUA。

### 操作步骤

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。

**步骤2** 执行**vi** *huawei-csi-configmap.yaml*命令,修改yaml文件。按**l**或**lnsert**进入编辑状态,修改相关参数。修改完成后,按**Esc**,并输入:**wq!**,保存修改。

多个后端间使用","进行分割,每个后端内容可以参考**4.2.1 对接企业存储配置**或 **4.2.2 对接分布式存储配置**。

"parameters"配置项下增加ALUA配置,参数配置请参见<mark>表7-12</mark>。

```
{
    "backends": [
    {
        "storage": "fusionstorage-san",
        ...
        "parameters": {..., "ALUA": {"<HostName>": {"switchoverMode": "*", "pathType": "*"},
    "<HostName>": {...}}}
```

} ]

### 表 7-12 ALUA 参数说明

| 参数名                   | 参数描述                         | 备注  |
|-----------------------|------------------------------|---|
| <hostname></hostname> | HostName的值为worker节<br>点的主机名  | 主机名通常使用 cat /etc/<br>hostname 可获取,支持正<br>则表达式方式匹配(配置规则<br>和优先级详见说明) |
| switchoverMode        | 切换模式。取值为:                    |   |
|                       | ● Disable_alua:禁用<br>ALUA    |   |
|                       | ● Enable_alua:启用<br>ALUA     |   |
| pathType              | 路径类型。取值为:                    |   |
|                       | ● optimal_path: 优选路<br>径     |   |
|                       | ● non_optimal_path:非<br>优选路径 |   |

### 山 说明

- 仅支持分布式存储iSCSI场景。
- 已经发放的Pod的节点不会主动更改ALUA 信息,需要通过在该节点重新发放Pod才会变更主机ALUA配置。
- 当HostName为正则表达式,配置方式参考《正则表达式》
   当HostName= "\*"时,表示的是通用配置,该条配置对任意主机名的主机生效,其余HostName的配置均视为一般配置。

#### 配置策略的规则:

- 优先级: 主机名一般配置 > 主机名通用配置( 详见**11.3 分布式存储ALUA特性配置策** 略样**例**例1)
- 在一般配置中,采用满足配置策略的第一条ALUA配置段(详见**11.3 分布式存储ALUA** 特性配置策略样例例2)
- 在一般配置中,如果需要精确匹配某个主机,参考**11.3 分布式存储ALUA特性配置策略 样例**例3
- **步骤3** 执行**kubectl create -f** *huawei-csi-configmap.yaml*命令,创建huawei-csi-configmap。

# kubectl create -f huawei-csi-configmap.yaml

步骤4 创建完成后,执行kubectl get configmap -n huawei-csi | grep huawei-csi | configmap命令,查看是否创建成功。显示如下回显时,则表示创建成功。

# kubectl get configmap -n huawei-csi | grep huawei-csi-configmap huawei-csi-configmap 1 5s

步骤5 启动huawei-csi服务,详细请参见4.2.3 启动huawei-csi服务。

----结束

## 7.5 配置存储拓扑感知

在Kubernetes集群中,可以根据节点的拓扑标签以及存储后端支持的拓扑能力调度和 发放资源。

## 前提条件

需要在集群中的worker节点完成拓扑的标签配置,标签配置方法如下:

- 1. 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意master节点。
- 2. 执行kubectl get node命令,查看当前集群中的worker节点信息。

# kubectl get node

NAME STATUS ROLES AGE VERSION

node01 Ready controlplane,etcd,worker 42d v1.22.3

node03 Ready worker 42d v1.22.3

node03 Ready worker 42d v1.22.3

 执行kubectl label node <nodename> topology.kubernetes.io/ <key>= <value>命令,给worker节点配置拓扑标签。其中<nodename>为worker 节点名称, key和value参数说明请参考表7-13。

# kubectl label node node01 topology.kubernetes.io/zone=ChengDu node/node01 labeled

#### 表 7-13 参数说明

| 参数名             | 参数描述           | 备注  |
|-----------------|----------------|---|
| <key></key>     | 拓扑标签的唯一标<br>识。 | 可支持配置: zone, region,<br>protocol. <i><protocol></protocol></i><br>其中 <i><protocol></protocol></i> 可支持配置iscsi, nfs, fc,<br>roce。     |
| <value></value> | 拓扑标签的参数值。      | "key"如果是"zone","region",<br>"value"值为自定义参数。<br>"key"如果是protocol. <i><protocol></protocol></i> ,<br>"value"值固定为<br>"csi.huawei.com"。 |

#### □ 说明

- 拓扑标签必须以topology.kubernetes.io开头。拓扑标签示例:
  - 示例1: topology.kubernetes.io/region=China-west
  - 示例2: topology.kubernetes.io/zone=ChengDu
  - 示例3: topology.kubernetes.io/protocol.iscsi=csi.huawei.com
  - 示例4: topology.kubernetes.io/protocol.fc=csi.huawei.com
- 同一节点上拓扑标签中同一个key只能支持一个value值。
- 如果同一节点上拓扑标签中同时配置多个protocol,选择后端时,只需要满足其中一个 protocol即可。
- 如果同一节点上拓扑标签中同时配置region和zone,选择后端时,需要满足全部。

4. 执行kubectl get nodes -o=json path='{range .items[\*]}[{.metadata.name}, {.metadata.labels}]{"\n"}{end}' | grep --color "topology.kubernetes.io"命令,查看当前集群中所有worker节点的标签信息。

# kubectl get nodes -o=jsonpath='{range .items[\*]}[{.metadata.name}, {.metadata.labels}]{"\n"}{end}' | grep --color "topology.kubernetes.io" [node01, {"beta.kubernetes.io/arch":"amd64","beta.kubernetes.io/os":"linux","kubernetes.io/arch":"amd64","kubernetes.io/os":"linux","node-role.kubernetes.io/controlplane":"true","node-role.kubernetes.io/etcd":"true","node-role.kubernetes.io/worker":"true","topology.kubernetes.io/zone":"ChengDu"}]

## 7.5.1 通过 Helm 配置存储拓扑感知

## 操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。
- 步骤2 进入Helm工程的目录下,如果无法找到之前的Helm工程,则将组件包中的helm目录 拷贝到master节点的任意目录下,组件包路径请参考3.3 上传软件包组件。
- 步骤3 备份安装CSI时使用的values.yaml文件,如果上次安装时使用的values.yaml文件无法找到,可通过helm get values helm-huawei-csi -n huawei-csi -a命令查询。

# cp values.yaml values.yaml.bak

**步骤4** 执行**vi** *values.yaml*命令打开文件,按需求配置多个后端,示例如下所示。修改完成后,按**Esc**,并输入:**wq!**,保存修改。

```
backends:
- storage: "oceanstor-nas"
name: "storage"
urls:
- "https://*.*.*:8088"
- "https://*.*.*:8088"
pools:
- "***"
- "***"
parameters:
protocol: "nfs"
portals:
- "*.*.*"
supportedTopologies:
- {"topology.kubernetes.io/region": "China-west", "topology.kubernetes.io/zone": "ChengDu"}
- {"topology.kubernetes.io/region": "China-south","topology.kubernetes.io/zone": "ShenZhen"}
```

步骤5 执行helm upgrade helm-huawei-csi./ -n huawei-csi-f values.yaml命令升级Helm chart。升级命令将更新Deployment、daemonset和RBAC资源,但是secret资源不会 更新,如果后端信息发生更改,则必须手动更新secret,更新操作请参考9.1 更新CSI上配置的存储用户名或密码。

```
其中,helm-huawei-csi为自定义的chart名称,huawei-csi为自定义的命名空间。
# helm upgrade helm-huawei-csi ./ -n huawei-csi
Release "helm-huawei-csi" has been upgraded. Happy Helming!
NAME: helm-huawei-csi
LAST DEPLOYED: Thu Jun 9 07:58:15 2022
NAMESPACE: huawei-csi
STATUS: deployed
REVISION: 2
TEST SUITE: None
```

**步骤6** 执行**vi** *StorageClass.yaml*命令,修改yaml文件。按**I**或**Insert**进入编辑状态,在yaml 文件下增加相关参数,详细参数说明请参见<mark>表</mark>7-14。修改完成后,按**Esc**,并输入:**wq!**,保存修改。

在StorageClass.yaml文件中添加以下配置项。

### • 示例1: 在StorageClass中配置zone和region信息

kind: StorageClass

apiVersion: storage.k8s.io/v1

metadata:

name: example-storageclass

provisioner: csi.huawei.com

parameters: volumeType: lun allocType: thin

#### volumeBindingMode: WaitForFirstConsumer

### allowedTopologies:

- matchLabelExpressions:
- key: topology.kubernetes.io/zone values:
- ChengDu
- key: topology.kubernetes.io/region

values:

- China-west

### • 示例2: 在StorageClass中配置协议信息

kind: StorageClass

apiVersion: storage.k8s.io/v1

metadata:

name: protocol-example-storageclass

provisioner: csi.huawei.com

parameters: volumeType: lun allocType: thin

volumeBindingMode: WaitForFirstConsumer

#### allowedTopologies:

- matchLabelExpressions:
- key: topology.kubernetes.io/protocol.iscsi values:
- csi.huawei.com

### 表 7-14 参数说明

| 参数名   | 参数描述  | 备注   |
|---|---|--|
| volumeBindin<br>gMode                               | PersistentVolume<br>绑定方式,用于控<br>制何时进行<br>PersistentVolume<br>动态资源调配和绑<br>定。       | 可配置 "WaitForFirstConsumer"或<br>"Immediate"<br>"WaitForFirstConsumer":表示延迟<br>PersistentVolume的绑定和调配,直到创建<br>使用PVC的Pod。<br>"Immediate":表示创建PVC后,立即发<br>生PersistentVolume绑定和调配。                                      |
| allowedTopol<br>ogies.matchLa<br>belExpression<br>s | 拓扑信息标签,用于过滤CSI后端和Kubernetes节点。如果匹配失败,会导致PVC或Pod无法创建。配置时需要同时按照固定格式配置"key"和"value". | "key":可支持配置<br>"topology.kubernetes.io/zone",<br>"topology.kubernetes.io/region",<br>topology.kubernetes.io/<br>protocol. <i><protocol></protocol></i> , 其中 <i><protocol></protocol></i> 为协议类型,例如:iscsi, fc, nfs等。 |

| 参数名 | 参数描述 | 备注  |
|-----|------|---|
|     |      | "value": "key"如果是"topology.kubernetes.io/ zone","topology.kubernetes.io/ region","value"值需要和 <mark>前提条件中设</mark><br>置的拓扑标签保持一致。 "key"如果是topology.kubernetes.io/ protocol. <pre>rotocol&gt;,"value"值固定为 "csi.huawei.com"</pre> |

步骤7 执行以下命令,基于该yaml文件创建StorageClass。

# kubectl create -f StorgeClass.yaml

步骤8 使用该StorageClass创建具有拓扑能力的PVC,详细操作请参考6.3.1 创建PVC。

步骤9 使用该PVC创建Pod,详细操作请参考6.4.1 创建Pod。

----结束

## 7.5.2 手动配置存储拓扑感知

## 操作步骤

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。

**步骤2** 执行**vi** *huawei-csi-configmap.yaml*命令,修改yaml文件。按**l**或**Insert**进入编辑状态,修改相关参数。修改完成后,按**Esc**,并输入**:wq!**,保存修改。

多个后端间使用","进行分割,每个后端内容可以参考**4.2.1 对接企业存储配置**或 **4.2.2 对接分布式存储配置**。

在huawei-csi-configmap.yaml文件的backends配置段中,添加**supportedTopologies** 字段,用于配置每个后端支持的拓扑信息。以下是后端示例。

```
{
    "backends":[
    {
        "storage": "oceanstor-san",
        ...
        "parameters": {"protocol": "iscsi", "portals": ["192.168.125.22", "192.168.125.23"]},
        "supportedTopologies": [
        "topology.kubernetes.io/region": "China-west", "topology.kubernetes.io/zone": "ChengDu"},
        {"topology.kubernetes.io/region": "China-south", "topology.kubernetes.io/zone": "ShenZhen"}]
    }
}
```

#### □ 说明

- "supportedTopologies"是一个列表,列表中的每个元素是一个字典。
- 列表中的每个元素只支持配置"topology.kubernetes.io/region"或 "topology.kubernetes.io/zone",参数值需要保持和前提条件中设置的拓扑标签一致。 (不需要配置"topology.kubernetes.io/protocol.<protocol>")

**步骤3** 执行**kubectl create -f** *huawei-csi-configmap.yaml*命令,创建huawei-csi-configmap。

# kubectl create -f huawei-csi-configmap.yaml

步骤4 创建完成后,执行kubectl get configmap -n huawei-csi | grep huawei-csiconfigmap命令,查看是否创建成功。显示如下回显时,则表示创建成功。

# kubectl get configmap -n huawei-csi | grep huawei-csi-configmap huawei-csi-configmap

步骤5 启动huawei-csi服务,详细请参见4.2.3 启动huawei-csi服务。

步骤6 执行vi StorageClass.yaml命令,修改yaml文件。按l或Insert进入编辑状态,在yaml 文件下增加相关参数,详细参数说明请参见表7-15。修改完成后,按Esc,并输 入:wq!,保存修改。

在StorageClass.yaml文件中添加以下配置项。

示例1: 在StorageClass中配置zone和region信息

kind: StorageClass

apiVersion: storage.k8s.io/v1

metadata:

name: example-storageclass

provisioner: csi.huawei.com

parameters:

volumeType: lun

allocType: thin

volumeBindingMode: WaitForFirstConsumer

#### allowedTopologies:

- matchLabelExpressions:
- key: topology.kubernetes.io/zone values:

  - ChengDu
- key: topology.kubernetes.io/region values:
- China-west
- 示例2: 在StorageClass中配置协议信息

kind: StorageClass

apiVersion: storage.k8s.io/v1

metadata:

name: protocol-example-storageclass

provisioner: csi.huawei.com

parameters:

volumeType: lun

allocType: thin

volumeBindingMode: WaitForFirstConsumer

#### allowedTopologies:

- matchLabelExpressions:
- key: topology.kubernetes.io/protocol.iscsi values:
- csi.huawei.com

### 表 7-15 参数说明

| 参数名                   | 参数描述  | 备注   |
|-----------------------|---|--|
| volumeBindin<br>gMode | PersistentVolume<br>绑定方式,用于控<br>制何时进行<br>PersistentVolume<br>动态资源调配和绑<br>定。 | 可配置"WaitForFirstConsumer"或<br>"Immediate"<br>"WaitForFirstConsumer":表示延迟<br>PersistentVolume的绑定和调配,直到创建<br>使用PVC的Pod。<br>"Immediate":表示创建PVC后,立即发<br>生PersistentVolume绑定和调配。 |

| 参数名   | 参数描述  | 备注   |
|---|---|--|
| allowedTopol<br>ogies.matchLa<br>belExpression<br>s | 拓扑信息标签,用于过滤CSI后端和Kubernetes节点。如果匹配失败,会导致PVC或Pod无法创建。配置时需要同时按照固定格式配置"key"和"value". | "key":可支持配置 "topology.kubernetes.io/zone", "topology.kubernetes.io/region", topology.kubernetes.io/ protocol. <pre>/protocol&gt;, 其中<pre>protocol&gt;为协议类型,例如: iscsi, fc, nfs等。  "value": "key"如果是"topology.kubernetes.io/zone", "topology.kubernetes.io/region", "value"值需要和前提条件中设置的拓扑标签保持一致。 "key"如果是topology.kubernetes.io/protocol.</pre>/ 如果是可以是是一个专家的一个专家的一个专家的一个专家的一个专家的一个专家的一个专家的一个专家的</pre> |

步骤7 执行以下命令,基于该yaml文件创建StorageClass。

# kubectl create -f StorgeClass.yaml

步骤8 使用该StorageClass创建具有拓扑能力的PVC,详细操作请参考6.3.1 创建PVC。

步骤9 使用该PVC创建Pod,详细操作请参考6.4.1 创建Pod。

----结束

## 7.6 企业存储高级特性

# 7.6.1 配置 QoS

本章节介绍如何创建支持QoS的LUN/文件系统卷。

#### 注意事项

- QoS特性不是Kubernetes的标准特性,由存储厂商自定义。
- QoS策略只能在创建StorageClass时指定,一旦完成创建,由于Kubernetes中 StorageClass不支持修改,QoS策略也不支持修改。

### 操作步骤

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。

**步骤2** 执行**vi** *StorageClass.yaml*命令,修改yaml文件。按**l**或**Insert**进入编辑状态,修改相关参数。修改完成后,按**Esc**,并输入:**wq!**,保存修改。

- metadata.name参数值是自定义的StorageClass对象名称。自定义字符串,支持 大小写字母、数字、中划线组合。
- "parameters"配置项下添加qos配置项。关于parameters的其它内容可以参考
   6.1.1.1 创建LUN StorageClass或6.1.1.2 创建文件系统StorageClass。

- qos配置项值是字典格式的json字符串(字符串两边由单引号修饰,字典key由双引号修饰)。
  - OceanStor V3/OceanStor V5系列存储详细参数说明请参见表7-16。
  - OceanStor Dorado V3系列存储详细参数说明请参见表7-17。
  - OceanStor Dorado 6.x/OceanStor 6.x系列存储详细参数说明请参见表 7-18。

kind: StorageClass apiVersion: storage.k8s.io/v1

metadata: name: "\*\*\*"

provisioner: "csi.huawei.com"

parameters:

qos: '{"IOTYPE": 2, "MINIOPS": 1000}'

#### 表 7-16 OceanStor V3/OceanStor V5 系列 QoS 支持的参数说明

| 参数名          | 参数描述       | 备注  |
|--------------|------------|---|
| IOTYPE       | 控制读写类型     | 可选(未明确指定将使用后端存储默认值,具体参考相关存储资料)有效值如下: ① 0:读IO ② 1:写IO ② 2:读写IO |
| MAXBANDWIDTH | 最大带宽限制策略   | 单位MB/s,有效值为>0的整数  |
| MINBANDWIDTH | 最小带宽保护策略   | 单位MB/s,有效值为>0的整数  |
| MAXIOPS      | 最大IOPS限制策略 | 有效值为>0的整数   |
| MINIOPS      | 最小IOPS保护策略 | 有效值为>0的整数   |
| LATENCY      | 最大时延保护策略   | 单位ms,有效值为>0的整数  |

#### 表 7-17 OceanStor Dorado V3 系列 QoS 支持的参数说明

| 参数名          | 参数描述       | 备注                           |
|--------------|------------|------------------------------|
| IOTYPE       | 控制读写类型     | 有效值如下:                       |
|              |            | ● 2: 读写IO                    |
| MAXBANDWIDTH | 最大带宽限制策略   | 单位MB/s,整数, 范围<br>1~999999999 |
| MAXIOPS      | 最大IOPS限制策略 | 类型为整数, 范围<br>100~99999999    |

| 参数名          | 参数描述     | 备注                              |
|--------------|----------|---------------------------------|
| IOTYPE       | 控制读写类型   | 有效值如下:<br>● 2: 读写IO             |
| MAXBANDWIDTH | 最大带宽策略   | 单位MB/s,类型为整数, 范围<br>1~999999999 |
| MINBANDWIDTH | 最小带宽策略   | 单位MB/s,类型为整数, 范围<br>1~999999999 |
| MAXIOPS      | 最大IOPS策略 | 类型为整数, 范围<br>100~99999999       |
| MINIOPS      | 最小IOPS策略 | 类型为整数, 范围<br>100~99999999       |
| LATENCY      | 最大时延策略   | 单位ms,仅支持配置0.5或1.5               |

表 7-18 OceanStor Dorado 6.x/OceanStor 6.x 系列 QoS 支持的参数说明

#### 山 说明

- 同属保护策略或者同属限制策略的多个参数可以同时指定,但是保护策略和限制策略参数不能同时指定。
- 租户用户不支持配置QoS策略。
- 配置QoS后只能在新建的PVC上生效;对于同名StorageClass已经发放的PVC,不能自动添加 OoS

步骤3 执行以下命令,基于该yaml文件创建StorageClass。

# kubectl create -f StorgeClass.yaml

步骤4 使用该StorageClass创建具有QoS能力的PVC,详细操作请参考6.3.1 创建PVC。

----结束

## 7.6.2 配置租户

本章节介绍如何配置租户。

#### 注意事项

该特性不支持SAN存储,具体支持型号请参考表2-7。

#### 7.6.2.1 通过 Helm 配置租户

#### 操作步骤

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。

步骤2 进入Helm工程的目录下,如果无法找到之前的Helm工程,则将组件包中的helm目录 拷贝到master节点的任意目录下,组件包路径请参考3.3 上传软件包组件。 步骤3 备份安装CSI时使用的values.yaml文件,如果上次安装时使用的values.yaml文件无法找到,可通过helm get values helm-huawei-csi -n huawei-csi -a命令查询。

# cp values.yaml values.yaml.bak

- **步骤4** 执行**vi** *values.yaml*命令打开文件,修改相关参数,修改完成后,按**Esc**,并输入:**wq!**,保存修改。
  - 如果租户使用的后端是OceanStor V3/V5存储,则在该后端配置中增加 "vstoreName"参数,参数值填写存储设备上的租户名称,示例如下。

```
packends:
- storage: "oceanstor-san"
name: "***"

vstoreName: "***"

urls:
- "https://*.*.*.*8088"

pools:
- "***"

parameters:
protocol: "iscsi"
portals:
- "****"
```

 如果租户使用的后端是OceanStor 6.1或OceanStor Dorado 6.x存储,则参考4.1.2 安装华为CSI节配置后端,请注意"urls"为租户的逻辑管理端口,"pools"和 "portals"需要确保是当前租户可用的存储池和数据逻辑端口。

```
backends:
- storage: "oceanstor-san"
name: "***"
urls:
- "https://*.*.*.*8088"
pools:
_ "***"
parameters:
protocol: "iscsi"
portals:
_ "****"
```

步骤5 执行helm upgrade helm-huawei-csi./ -n huawei-csi-f values.yaml命令升级Helm chart。升级命令将更新Deployment、daemonset和RBAC资源,但是secret资源不会 更新,如果后端信息发生更改,则必须手动更新secret,更新操作请参考9.1 更新CSI 上配置的存储用户名或密码。

```
其中,helm-huawei-csi为自定义的chart名称,huawei-csi为自定义的命名空间。
```

```
# helm upgrade helm-huawei-csi ./ -n huawei-csi
Release "helm-huawei-csi" has been upgraded. Happy Helming!
NAME: helm-huawei-csi
LAST DEPLOYED: Thu Jun 9 07:58:15 2022
NAMESPACE: huawei-csi
STATUS: deployed
REVISION: 2
TEST SUITE: None
```

----结束

## 7.6.2.2 手动配置租户

### 操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。
- **步骤2** 执行**vi** *huawei-csi-configmap.yaml*命令,修改yaml文件。按**l**或**Insert**进入编辑状态,修改相关参数。修改完成后,按**Esc**,并输入:**wq!**,保存修改。

多个后端间使用","进行分割,每个后端内容可以参考4.2.1 对接企业存储配置。

● 如果租户使用的后端是OceanStor V3/V5存储,则在该后端配置中增加 "vstoreName"参数,参数值填写存储设备上的租户名称。

● 如果租户使用的后端是OceanStor 6.1或OceanStor Dorado 6.x存储,则参考**4.2.1** 对接企业存储配置节配置后端,请注意"urls"为租户的逻辑管理端口, "pools"和"portals"需要确保是当前租户可用的存储池和数据逻辑端口。

#### 山 说明

配置完成huawei-csi-configmap.yaml后,需要重启huawei-csi-controller和huawei-csi-node,否则不生效。

**步骤3** 执行**kubectl create -f** *huawei-csi-configmap.yaml*命令,创建huawei-csi-configmap。

# kubectl create -f huawei-csi-configmap.yaml

步骤4 执行kubectl get configmap -n huawei-csi | grep huawei-csi-configmap命令,查看是否创建成功。显示如下回显时,则表示创建成功。

```
# kubectl get configmap -n huawei-csi | grep huawei-csi-configmap
huawei-csi-configmap 1 5s
```

步骤5 启动huawei-csi服务,详细请参见4.2.3 启动huawei-csi服务。

#### □ 说明

- 启动huawei-csi服务时,**4.2.3 启动huawei-csi服务**中的步骤**3**请填写存储设备租户的用户名和密码。
- 该特性支持的存储型号请参考<mark>表2-7</mark>。

#### ----结束

## 7.6.3 配置 NAS 双活

当您需要配置NAS双活时,执行此操作。

#### 注意事项

支持NAS双活的资源对象以及特性描述,请参见表7-19。

#### 表 7-19 特性描述

| 资源对象 | 操作 | 是否支持 | 备注            |
|------|----|------|---------------|
| PVC  | 创建 | 是    | 支持和其他特性(除远程复  |
|      | 删除 | 是    | 制)配合使用,例如QoS。 |
|      | 扩容 | 是    | -             |

| 资源对象           | 操作       | 是否支持 | 备注                                   |
|----------------|----------|------|--------------------------------------|
|                | 同步双活Pair | 否    | 存储本身能够支持该操作,配                        |
|                | 暂停双活Pair | 否    | 套Kubernetes时,由于<br>Kubernetes无法感知到双活 |
|                | 优先站点切换   | 否    | Pair,因此无法支持该操作。                      |
| Pod            | 创建       | 是    | 支持同时将主端和从端文件系                        |
|                | 删除       | 是    | 统挂载给Pod使用。<br>                       |
| VolumeSnapshot | 创建       | 是    | 仅支持对存储主端文件系统执                        |
|                | 删除       | 是    | 行快照。<br>                             |

#### 7.6.3.1 前提条件

#### 配置存储双活关系

配置NAS双活前,需要在两台存储设备之间配置双活关系,包含远端设备、双活域等,仅支持文件系统双活域工作模式为双活模式,配置操作请参考对应存储型号的产品文档。

#### 须知

- 创建双活租户Pair可能会导致租户下本地非双活文件系统业务中断。
- 创建双活租户Pair后,租户下本地非双活文件系统可以继续使用,不会自动转换为 双活文件系统。
- 不能在已经创建为双活租户Pair的租户下创建非双活文件系统。

## 配置双活租户信息

配置NAS双活前,需要在两台存储设备上创建租户,租户Pair,租户用户和租户对应的逻辑端口,配置操作请参考对应存储型号的产品文档。本节以OceanStor Dorado 6.1.3为例,操作如下:

步骤1 在双活域主端存储创建租户。

- 1. 登录双活域主端存储DeviceManager管理界面。
- 2. 选择"服务 > 租户服务 > 多租户"。
- 3. 单击"创建",设置"名称"(例如vStoreA),单击"确定"。

步骤2 在双活域从端存储创建租户。

- 1. 登录双活域从端存储DeviceManager管理界面。
- 2. 选择"服务 > 租户服务 > 多租户"。
- 3. 单击"创建",设置"名称"(例如vStoreB),单击"确定"。

步骤3 在双活域主端存储创建双活租户Pair。

- 1. 登录双活域主端存储DeviceManager管理界面。
- 2. 选择"数据保护 > 对象 > 多租户 > 多租户"。
- 3. 选择要创建双活租户Pair的租户(例如<mark>步骤1</mark>中创建的vStoreA),单击"创建双活租户Pair"。
- 4. 选择"Pair创建模式"为"手动",选择"远端租户"为从端存储租户(例如步骤2中创建的vStoreB),根据实际情况填写剩余参数,单击"确定"。

#### 步骤4 在双活域主端存储创建本端设备的管理逻辑端口。

- 对于使用系统租户创建的双活Pair,直接使用本端存储DeviceManager管理界面的IP地址作为本端设备的管理逻辑端口。
- 对于使用普通租户创建的双活Pair,创建本端设备的管理逻辑端口。
  - a. 登录双活域主端存储DeviceManager管理界面。
  - b. 选择"服务 > 网络 > 逻辑端口"。
  - c. 单击"创建",输入"名称"(例如manageA),选择"角色"为"管理",选择"所属租户"为"vStoreA",选择"归属站点"为"本端设备",根据实际情况填写剩余参数,单击"确定"。

#### 步骤5 在双活域主端存储创建远端设备的管理逻辑端口。

- 对于使用系统租户创建的双活Pair,直接使用远端存储DeviceManager管理界面的IP地址作为远端设备的管理逻辑端口。
- 对于使用普通租户创建的双活Pair,创建远端设备的管理逻辑端口。
  - a. 登录双活域主端存储DeviceManager管理界面。
  - b. 选择"服务 > 网络 > 逻辑端口"。
  - c. 单击"创建",输入"名称"(例如manageB),选择"角色"为"管理",选择"所属租户"为"vStoreA",选择"归属站点"为"远端设备",根据实际情况填写剩余参数,单击"确定"。
  - d. 在双活域从端存储的DeviceManager管理界面检查该逻辑端口的正确性。 例如该逻辑端口中的"主端口"参数和双活域主端存储对应逻辑端口中"主端口"参数处于同一网络平面。

#### 步骤6 在双活域主端存储创建本端设备的业务逻辑端口。

- 1. 登录双活域主端存储DeviceManager管理界面。
- 2. 选择"服务 > 网络 > 逻辑端口"。
- 3. 单击"创建",输入"名称"(例如dataA ),选择"角色"为"数据",选择 "数据协议"为"NFS",选择"所属租户"为"vStoreA",选择"归属站点" 为"本端设备",根据实际情况填写剩余参数,单击"确定"。

#### **步骤7** 在双活域主端存储创建远端设备的业务逻辑端口。

- 1. 登录双活域主端存储DeviceManager管理界面。
- 2. 选择"服务 > 网络 > 逻辑端口"。
- 3. 单击"创建",输入"名称"(例如dataB),选择"角色"为"数据",选择 "数据协议"为"NFS",选择"所属租户"为"vStoreA",选择"归属站点" 为"远端设备",根据实际情况填写剩余参数,单击"确定"。
- 4. 在双活域从端存储的DeviceManager管理界面检查该逻辑端口的正确性。 例如该逻辑端口中的"主端口"参数和双活域主端存储对应逻辑端口中"主端口"参数处于同一网络平面。

步骤8 在双活域主端存储创建租户用户。

- 1. 登录双活域主端存储DeviceManager管理界面。
- 2. 选择"服务 > 租户服务 > 多租户"。
- 3. 单击租户"vStoreA"的名称,单击"用户管理",单击"创建",选择"类型"为"本地用户",选择"角色"为"租户管理员",根据实际情况填写剩余参数,单击"确定"。

步骤9 在双活域从端存储创建租户用户。

- 1. 登录双活域从端存储DeviceManager管理界面。
- 2. 选择"服务 > 租户服务 > 多租户"。
- 3. 单击租户"vStoreB"的名称,单击"用户管理",单击"创建",选择"类型"为"本地用户",选择"角色"为"租户管理员",根据实际情况填写剩余参数,单击"确定"。

----结束

#### 7.6.3.2 使用 Helm 配置 NAS 双活

#### 操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。
- 步骤2 进入Helm工程的目录下,如果无法找到之前的Helm工程,则将组件包中的helm目录 拷贝到master节点的任意目录下,组件包路径请参考3.3 上传软件包组件。
- **步骤3** 备份安装CSI时使用的values.yaml文件,如果上次安装时使用的values.yaml文件无法找到,可通过**helm get values** *helm-huawei-csi* **-n** *huawei-csi* **-a** 命令查询。

# cp values.yaml values.yaml.bak

**步骤4** 执行**vi** *values.yaml*命令打开文件,新增或修改backends中双活相关的参数,参数说明 参考**表7-20**。修改完成后,按**Esc**,并输入:**wq!**,保存修改。

```
# An array of storages with the access info
backends:
 - storage: "oceanstor-nas"
  name: "hyperMetro1"
  urls:
   - "https://*.*.*:8088"
  pools:
 vstoreName: "vStore1"
 metrovStorePairID: "f09838237b93c000"
 metroBackend: "hyperMetro2"
  parameters:
   protocol: "nfs"
   portals:
     - "*.*.*.
 - storage: "oceanstor-nas"
  name: "hyperMetro2"
  urls:
    - "https://*.*.*:8088"
  pools:
 vstoreName: "vStore1"
 metrovStorePairID: "f09838237b93c000"
 metroBackend: "hyperMetro1"
  parameters:
  protocol: "nfs"
```

portals:

#### 表 7-20 后端存储设备双活配置项描述

| 配置项                   | 参数格式 | 描述   | 备注   |
|-----------------------|------|--|--|
| vstoreName            | 字符串  | 条件必填<br>(OceanStor<br>V3/V5存储必<br>填),租户名称。 | 仅租户用户支持配置NAS双活,租户用户的配置可参考 <b>7.6.3.1</b> 前提条件。                                   |
| metrovStor<br>ePairID | 字符串  | 必填,租户归属的<br>双活租户Pair ID。                   | 以OceanStor Dorado 6.x/<br>OceanStor 6.1为例,该值在存<br>储设备DeviceManager界面显示<br>为"ID"。 |
| metroBacke<br>nd      | 字符串  | 必填,双活对端的<br>后端名,两个双活<br>后端相互构成双活<br>关系。    | 后端hyperMetro1的双活对端是<br>hyperMetro2,相应的,<br>hyperMetro2的双活对端是<br>hyperMetro1。     |

步骤5 执行helm upgrade helm-huawei-csi./ -n huawei-csi-f values.yaml命令升级Helm chart。升级命令将更新Deployment、daemonset和RBAC资源,但是secret资源不会 更新,如果后端信息发生更改,则必须手动更新secret,更新操作请参考9.1 更新CSI 上配置的存储用户名或密码。

其中,helm-huawei-csi为自定义的chart名称,huawei-csi为自定义的命名空间。

# helm upgrade helm-huawei-csi ./ -n huawei-csi

Release "helm-huawei-csi" has been upgraded. Happy Helming!

NAME: helm-huawei-csi

LAST DEPLOYED: Thu Jun 9 07:58:15 2022

NAMESPACE: huawei-csi STATUS: deployed REVISION: 2

TEST SUITE: None

**步骤6** 执行**vi** *StorageClass.yaml*命令,修改yaml文件。按**I**或**Insert**进入编辑状态,在yaml 文件的"parameters"配置项下增加"hyperMetro"参数,详细参数说明请参见表7-21。修改完成后,按**Esc**,并输入:**wq!**,保存修改。

kind: StorageClass

apiVersion: storage.k8s.io/v1

metadata: name: "\*\*\*"

provisioner: "csi.huawei.com"

parameters:

...
volumeType: fs
hyperMetro: "true"

#### 表 7-21 参数说明

| 参数名                   | 参数描述          | 备注   |
|-----------------------|---------------|--|
| parameters.hyperMetro | 指定是否创建双活<br>卷 | 配置为true,表示需要创建双活<br>卷;不配置或配置为false,表示<br>不创建双活卷。 |

步骤7 执行以下命令,基于该yaml文件创建StorageClass。

# kubectl create -f StorgeClass.yaml

步骤8 使用该StorageClass创建具有NAS双活能力的PVC,详细操作请参考6.3.1 创建PVC。

----结束

#### 7.6.3.3 手动配置 NAS 双活

#### 操作步骤

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。

**步骤2** 执行**vi** *huawei-csi-configmap.yaml*命令,修改yaml文件。按**l**或**lnsert**进入编辑状态,修改相关参数。修改完成后,按**Esc**,并输入:**wq!**,保存修改。

在huawei-csi-configmap.yaml文件的backends配置段中,添加构成双活关系的两个后端,每个后端存储设备配置项可以参考**4.2.1 对接企业存储配置**。还需要额外增加双活场景下一些配置参数,详情请参考**表7-22**。

```
kind: ConfigMap
apiVersion: v1
metadata:
 name: huawei-csi-configmap
 namespace: huawei-csi
data:
 csi.json: |
       "backends": [
             "storage": "oceanstor-nas",
             "name": "hyperMetro1",
             "urls": ["https://192.168.125.20:8088", "https://192.168.125.21:8088"],
             "pools": ["storagepool01", "storagepool02"],
"parameters": {"protocol": "nfs", "portals": ["192.168.125.22"]},
"vstoreName": "vStore1",
             "metrovStorePairID": "f09838237b93c000",
             "metroBackend": "hyperMetro2"
             "storage": "oceanstor-nas",
             "name": "hyperMetro2",
"urls": ["https://192.168.125.24:8088", "https://192.168.125.25:8088"],
             "pools": ["storagepool01", "storagepool02"],
"parameters": {"protocol": "nfs", "portals": ["192.168.125.26"]},
"vstoreName": "vStore1",
             "metrovStorePairID": "f09838237b93c000",
             "metroBackend": "hyperMetro1"
      ]
  }
```

#### 表 7-22 后端存储设备双活配置项描述

| 配置项        | 参数格式 | 描述   | 备注   |
|------------|------|--|--|
| vstoreName | 字符串  | 条件必填<br>(OceanStor<br>V3/V5存储必<br>填),租户名称。 | 仅租户用户支持配置NAS双活,租户用户的配置可参考 <b>7.6.3.1</b> 前提条件。 |

| 配置项                   | 参数格式 | 描述                                      | 备注   |
|-----------------------|------|---|--|
| metrovStor<br>ePairID | 字符串  | 必填,租户归属的<br>双活租户Pair ID。                | 以OceanStor Dorado 6.x/<br>OceanStor 6.1为例,该值在存<br>储设备DeviceManager界面显示<br>为"ID"。 |
| metroBacke<br>nd      | 字符串  | 必填,双活对端的<br>后端名,两个双活<br>后端相互构成双活<br>关系。 | 后端hyperMetro1的双活对端是<br>hyperMetro2,相应的,<br>hyperMetro2的双活对端是<br>hyperMetro1。     |

步骤3 执行kubectl create -f huawei-csi-configmap.yaml命令,创建huawei-csiconfigmap.

# kubectl create -f huawei-csi-configmap.yaml

步骤4 创建完成后,执行kubectl get configmap -n huawei-csi | grep huawei-csiconfigmap命令,查看是否创建成功。显示如下回显时,则表示创建成功。

# kubectl get configmap -n huawei-csi | grep huawei-csi-configmap huawei-csi-configmap 1 5s

- 步骤5 启动huawei-csi服务,详细请参见4.2.3 启动huawei-csi服务(使用7.6.3.1 前提条件 中创建的租户用户进行配置)。
- 步骤6 执行vi StorageClass.yaml命令,修改yaml文件。按l或Insert进入编辑状态,在yaml 文件的"parameters"配置项下增加"hyperMetro"参数,详细参数说明请参见表 7-23。修改完成后,按Esc,并输入:wq!,保存修改。

kind: StorageClass apiVersion: storage.k8s.io/v1 metadata: name: "\*\*\*" provisioner: "csi.huawei.com"

parameters:

volumeType: fs hyperMetro: "true"

#### 表 7-23 参数说明

| 参数名                   | 参数描述          | 备注   |
|-----------------------|---------------|--|
| parameters.hyperMetro | 指定是否创建双活<br>卷 | 配置为true,表示需要创建双活<br>卷;不配置或配置为false,表示<br>不创建双活卷。 |

步骤7 执行以下命令,基于该yaml文件创建StorageClass。

# kubectl create -f StorgeClass.yaml

步骤8 使用该StorageClass创建具有NAS双活能力的PVC,详细操作请参考6.3.1 创建PVC。

----结束

## 7.6.4 配置应用类型

本章节介绍如何创建支持不同应用类型的LUN/文件系统卷。

#### 注意事项

- 应用类型特性不是Kubernetes的标准特性,由存储厂商自定义。
- 应用类型只能在创建PVC时指定。
- 针对已经创建好的PVC, Kubernetes无法对其进行修改。

#### 操作步骤

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。

**步骤2** 执行**vi** *StorageClass.yaml*命令,修改yaml文件。按**l**或**Insert**进入编辑状态,修改相关参数。修改完成后,按**Esc**,并输入:**wq!**,保存修改。

"parameters"配置项下添加applicationType配置项,applicationType配置项值是字符串,详细参数说明请参见表7-24。关于parameters的其它内容可以参考6.1.1.1 创建LUN StorageClass或6.1.1.2 创建文件系统StorageClass。

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
name: "***"
provisioner: "csi.huawei.com"
parameters:
...
volumeType: "***"
applicationType: "***"
```

表 7-24 applicationType 参数说明

| 参数名                            | 参数描述                   | 备注  |
|--------------------------------|------------------------|---|
| parameters.app<br>licationType | 字符串格式, 指存储设备 上的应用类型名称。 | "volumeType"为"lun"时,在DeviceManager管理界面,选择<br>"服务 > 块服务 > LUN组 > LUN ><br>创建 > 应用类型",获取应用类型<br>名称。<br>"volumeType"为"fs"时,在DeviceManager管理界面,选择<br>"服务 > 文件服务 > 文件系统 > 创建 > 应用类型",获取应用类型名称。 |

#### 山 说明

该特性仅支持OceanStor Dorado 6.x系列存储。

步骤3 执行以下命令,基于该yaml文件创建StorageClass。

# kubectl create -f StorgeClass.yaml

步骤4 使用该StorageClass创建具有应用类型能力的PVC,详细操作请参考6.3.1 创建PVC。

----结束

# 7.7 分布式存储高级特性

## 7.7.1 配置 QoS

本章节介绍如何创建支持QoS的LUN卷。

## 注意事项

- QoS特性不是Kubernetes的标准特性,由存储厂商自定义。
- QoS策略只能在创建StorageClass时指定,一旦完成创建,由于Kubernetes中 StorageClass不支持修改,QoS策略也不支持修改。

#### 操作步骤

步骤1 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。

步骤2 执行vi StorageClass.yaml命令,修改yaml文件。按l或Insert进入编辑状态,修改相关 参数。修改完成后,按Esc,并输入:wq!,保存修改。

- "parameters"配置项下添加gos配置项。关于parameters的其它内容可以参考 6.1.1.1 创建LUN StorageClass。
- qos配置项值是字典格式的json字符串(字符串两边由单引号修饰,字典key由双 引号修饰),详细参数说明请参见表7-25。

kind: StorageClass apiVersion: storage.k8s.io/v1 metadata: name: "\*\*\*"

provisioner: "csi.huawei.com"

parameters:

gos: '{"maxMBPS": 999, "maxIOPS": 999}'

#### 表 7-25 qos 支持的参数说明

| 参数名     | 参数描述   | 备注                      |
|---------|--------|-------------------------|
| maxMBPS | 最大带宽   | 必填。有效值为>0的整数,单位<br>MB/s |
| maxIOPS | 最大IOPS | 必填。有效值为>0的整数            |

步骤3 执行以下命令,基于该yaml文件创建新的PVC。

# kubectl create -f StorageClass.yaml

----结束

## 7.7.2 配置软配额

本章节介绍如何创建支持软配额的PVC。

#### 注意事项

- 该特性要求存储版本OceanStor Pacific系列8.1.0 及其之后版本。
- 该特性仅支持在对接文件系统类型的存储池时设置。

#### 操作步骤

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。

**步骤2** 执行**vi** *StorageClass.yaml*命令,修改yaml文件。按**l**或**Insert**进入编辑状态,修改相关参数。修改完成后,按**Esc**,并输入:**wq!**,保存修改。

- "parameters"配置项下添加storageQuota配置项。关于parameters的其它内容可以参考**6.1.1.2 创建文件系统StorageClass**。
- storageQuota配置项值是字典格式的json字符串(字符串两边由单引号修饰,字 典key由双引号修饰),详细参数说明请参见表7-26。

kind: StorageClass apiVersion: storage.k8s.io/v1 metadata: name: "\*\*\*" provisioner: "csi.huawei.com" parameters: volumeType: "fs"

storageQuota: '{"spaceQuota": "softQuota", "gracePeriod": 100}'

#### 表 7-26 storageQuota 支持的参数说明

| 参数名         | 参数描述            | 备注  |
|-------------|-----------------|---|
| spaceQuota  | 文件配额类型          | 必选。仅支持配置"softQuota"或者"hardQuota"  |
| gracePeriod | 配置软配额时,允许的 超限天数 | 条件可选,当"spaceQuota"配置<br>为"softQuota"时可选。<br>类型为整数,支持范围为0~<br>4294967294 |

步骤3 执行以下命令,基于该yaml文件创建新的PVC。

# kubectl create -f StorageClass.yaml

步骤4 根据6.3.1 创建PVC,在PVC中配置该StorageClass,完成PVC的创建

----结束

## 7.7.3 配置帐户

本章节介绍如何配置指定帐户的后端并指定该后端创建PVC。

## 注意事项

该特性仅支持分布式NAS系列存储。

#### 7.7.3.1 通过 Helm 配置帐户

#### 操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。
- 步骤2 进入Helm工程的目录下,如果无法找到之前的Helm工程,则将组件包中的helm目录 拷贝到master节点的任意目录下,组件包路径请参考3.3 上传软件包组件。
- **步骤3** 备份安装CSI时使用的values.yaml文件,如果上次安装时使用的values.yaml文件无法找到,可通过**helm get values** *helm-huawei-csi* -**n** *huawei-csi* -**a**命令查询。

# cp values.yaml values.yaml.bak

**步骤4** 执行**vi** *values.yaml*命令打开配置文件,在backends字段下添加accountName字段,参数说明参考**表7-27**。并且确保portals必须配置为帐户所拥有的逻辑端口,示例如下所示。修改完成后,按**Esc**,并输入:**wq!**,保存修改。

```
backends:
- storage: "fusionstorage-nas"
name: "***"

accountName: "***"
urls:
- "https://*.*.*.*:28443"
pools:
- "***"
parameters:
protocol: "nfs"
portals:
- "*.*.*"
```

#### 表 7-27 accountName 参数说明

| 参数名                      | 参数描述                  | 备注  |
|--------------------------|-----------------------|---|
| backends.accou<br>ntName | 字符串格式, 需要指定<br>的帐户名称。 | <ul> <li>"accountName"配置为</li> <li>"system"或不配置该字段时,用该后端创建的PVC将创建在系统帐户(用户名: system)下。</li> <li>"accountName"配置为其他自建用户时,用该后端创建的PVC将只会创建在该用户下。</li> </ul> |

步骤5 执行helm upgrade helm-huawei-csi./ -n huawei-csi-f values.yaml命令升级Helm chart。升级命令将更新Deployment、daemonset和RBAC资源,但是secret资源不会更新,如果后端信息发生更改,则必须手动更新secret,更新操作请参考9.1 更新CSI上配置的存储用户名或密码。

其中,helm-huawei-csi为自定义的chart名称,huawei-csi为自定义的命名空间。

# helm upgrade helm-huawei-csi ./ -n huawei-csi
Release "helm-huawei-csi" has been upgraded. Happy Helming!
NAME: helm-huawei-csi
LAST DEPLOYED: Thu Jun 9 07:58:15 2022
NAMESPACE: huawei-csi
STATUS: deployed
REVISION: 2
TEST SUITE: None

#### ----结束

#### 7.7.3.2 手动配置帐户

#### 操作步骤

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。

**步骤2** 执行**vi** *huawei-csi-configmap.yaml*命令,修改yaml文件。按**l**或**lnsert**进入编辑状态,修改相关参数。修改完成后,按**Esc**,并输入:**wq!**,保存修改。

后端内容配置参考**4.2.2.3 通过NFS对接分布式存储NAS配置**,在huawei-csi-configmap.yaml文件的backends配置字段中,添加"accountName"配置项,"accountName"配置项值是字符串,内容为需要指定的帐户名称,详细参数说明请参考**表7-28**,portals必须配置为帐户所拥有的逻辑端口。

```
{
  "backends":[
  {
    "storage": "fusionstorage-nas",
    "name": "***",
    ...
    "accountName": "***",
    "parameters": {"protocol": "nfs", "portals": ["*.*.*.*"]}
  }
}
```

#### □ 说明

"portals"必须是配置的帐户下的IP地址,可在存储DeviceManager中的"资源 > 访问 > 业务网络"中查看。

#### 表 7-28 accountName 参数说明

| 参数名                      | 参数描述                  | 备注  |
|--------------------------|-----------------------|---|
| backends.accou<br>ntName | 字符串格式, 需要指定<br>的帐户名称。 | <ul> <li>"accountName"配置为<br/>"system"或不配置该字段时,用<br/>该后端创建的PVC将创建在系统帐户(用户名: system)下。</li> <li>"accountName"配置为其他自建<br/>用户时,用该后端创建的PVC将只<br/>会创建在该用户下。</li> </ul> |

**步骤3** 执行**kubectl create -f** *huawei-csi-configmap.yaml*命令,创建huawei-csi-configmap。

# kubectl create -f huawei-csi-configmap.yaml

步骤4 执行以下命令,重启huawei-csi-controller服务。

# kubectl get deployment huawei-csi-controller -o yaml -n=huawei-csi | kubectl replace --force -f -

步骤5 执行以下命令,重启huawei-csi-node服务。

# kubectl get daemonset huawei-csi-node -o yaml -n=huawei-csi | kubectl replace --force -f -

**步骤6** 执行**vi** *StorageClass.yaml*命令,修改yaml文件。按**l**或**Insert**进入编辑状态,修改相关参数。修改完成后,按**Esc**,并输入**:wg!**,保存修改。

"parameters"配置项下添加"backend"配置项,"backend"配置项值是字符串,内容填写为步骤2中指定帐户的后端名称。关于"parameters"配置项的其它内容,请

# 参考6.1.1.2 创建文件系统StorageClass。关于指定后端创建PVC请参考7.2 指定后端创建PVC章节。

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
name: "***"
provisioner: "csi.huawei.com"
parameters:
...
volumeType: "***"
backend: "***"
```

步骤7 执行以下命令,基于该yaml文件创建StorageClass。

# kubectl create -f StorgeClass.yaml

步骤8 使用该StorageClass创建指定用户下的PVC,详细操作请参考6.3.1 创建PVC。

----结束

## 7.7.4 DPC 场景配置挂载参数

本章节介绍DPC场景下配置挂载参数。

#### 注意事项

该特性仅支持使用DPC协议的分布式NAS系列存储。

### 操作步骤

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意master节点。

**步骤2** 执行**vi** *StorageClass.yaml*命令,修改yaml文件。按**l**或**Insert**进入编辑状态,修改相关参数。修改完成后,按**Esc**,并输入:**wq!**,保存修改。

- 添加mountOptions配置项。关于StorageClass.yaml的其它内容可以参考**6.1.1.1 创建LUN StorageClass**或**6.1.1.2 创建文件系统StorageClass**。
- mountOptions配置项值是列表格式,详细参数说明请参见表7-29。

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
name: "***"
provisioner: "csi.huawei.com"
mountOptions:
- ***
- ***
parameters:
...
```

表 7-29 mountOptions 支持的参数说明

| 参数名          | 参数描述   | 备注   |
|--------------|--|--|
| acl          | 命名空间支持ACL功能,DPC客<br>户端支持POSIX ACL、NFSv4<br>ACL、NT ACL的鉴权行为。  | acl、aclonlyposix、cnflush、cflush参数描述仅供参考,详细参数说明请参考《OceanStor Pacific系列产品文档》> 配置 > 文件服务基础业务配置指南 > 配置基础业务(DPC场景)> 客户端访问DPC共享 > 步骤2。 |
| aclonlyposix | 命名空间支持POSIX ACL功能,<br>DPC客户端支持POSIX ACL的鉴<br>权行为。<br>支持POSIX ACL的协议有:<br>DPC、NFSv3、HDFS。如使用<br>NFSv4 ACL或NT ACL,会导致<br>DPC客户端无法识别该类型的<br>ACL,从而导致该类型的ACL不<br>会生效。  | aclonlyposix与acl参数同时使用时,仅acl参数生效,即命名空间支持ACL功能。   |
| cnflush      | 异步刷盘模式,即关闭命名空间下的文件时不会立即刷盘。<br>异步刷盘模式,当文件关闭时不会同步将Cache的数据持久化到存储介质中,而是通过Cache异步刷盘的方式将数据写入存储介质,Cache的后台刷盘将在写业务完成后根据刷盘周期定时刷盘。在多客户端场景下,对同一文件进行并行操作,文件Size的更新会受刷盘周期的影响,即当刷盘动作完成后才会更新文件的Size,更新通常会在数秒内完成。同步I/O不受刷盘周期影响。 |  |
| cflush       | 同步刷盘模式,即关闭命名空间<br>下的文件时立即刷盘。   | 默认使用同步刷盘模式。  |

步骤3 执行以下命令,基于该yaml文件创建新的PVC。

# kubectl create -f StorageClass.yaml

----结束

# **8** 卸载 CSI

当您需要卸载华为CSI时执行本节操作。

- 8.1 使用Helm卸载huawei-csi
- 8.2 手动卸载huawei-csi
- 8.3 (可选)卸载Snapshot依赖组件服务

# 8.1 使用 Helm 卸载 huawei-csi

使用Helm安装的华为CSI插件可以参考本章节卸载华为CSI,手动部署的华为CSI请参考8.2 手动卸载huawei-csi章节进行卸载操作。

#### 操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。
- 步骤2 执行helm uninstall helm-release-name -n huawei-csi命令卸载华为CSI,该命令将删除huawei-csi-controller、huawei-csi-node、huawei-csi-configmap和RBAC配置等,该命令不会删除huawei-csi-secret和huawei-csi容器镜像。

# helm uninstall helm-huawei-csi -n huawei-csi release "helm-huawei-csi" uninstalled

**步骤3** 执行**helm list -n** *huawei-csi*命令检查服务是否卸载成功。如果回显为空,则表示服务卸载成功。

其中, huawei-csi为chart所在的命名空间。

# helm list -n huawei-csi

NAME NAMESPACE REVISION UPDATED STATUS CHART APP VERSION

步骤4 (可选)卸载快照依赖组件服务,请参考8.3 (可选)卸载Snapshot依赖组件服务进行操作。

----结束

# 8.2 手动卸载 huawei-csi

本章节介绍如何手动卸载华为CSI,快照依赖组件服务卸载请参考**8.3(可选)卸载** Snapshot依赖组件服务。

## 8.2.1 卸载 huawei-csi-node 服务

#### 操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。
- **步骤2** 执行**kubectl delete daemonset huawei-csi-node -n** *huawei-csi*命令卸载huawei-csi-node服务,huawei-csi替换为实际的命名空间。

# kubectl delete daemonset huawei-csi-node -n huawei-csi

步骤3 执行以下命令检查服务是否已成功卸载(如果提示NotFound错误,表示已成功卸载)。

# kubectl get daemonset huawei-csi-node -n huawei-csi

-----结束

## 8.2.2 卸载 huawei-csi-controller 服务

#### 操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。
- 步骤2 执行kubectl delete deployment huawei-csi-controller -n huawei-csi命令卸载 huawei-csi-controller服务,huawei-csi替换为实际的命名空间。

# kubectl delete deployment huawei-csi-controller -n huawei-csi

步骤3 执行以下命令检查服务是否已成功卸载(如果提示NotFound错误,表示已成功卸载)。

# kubectl get deployment huawei-csi-controller -n huawei-csi

----结束

## 8.2.3 删除 huawei-csi-configmap 对象

#### 操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。
- 步骤2 执行kubectl delete configmap huawei-csi-configmap -n huawei-csi命令删除 configmap对象,其中huawei-csi-configmap是configmap的名称,huawei-csi是对象所在的命名空间。

# kubectl delete configmap huawei-csi-configmap -n huawei-csi

**步骤3** 执行以下命令检查对象是否已成功删除(如果提示NotFound错误,表示已成功删除)。

# kubectl get configmap huawei-csi-configmap -n huawei-csi

----结束

## 8.2.4 删除 huawei-csi-secret 对象

#### 操作步骤

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。

步骤2 执行kubectl delete secret huawei-csi-secret -n huawei-csi命令删除secret对象,其中huawei-csi-secret是secret对象的名称,huawei-csi是secret对象所在的命名空间。

# kubectl delete secret huawei-csi-secret -n huawei-csi

**步骤3** 执行以下命令检查secret对象是否删除成功,如果命令回显提示"NotFound"表示huawei-csi-secret对象已成功删除。

# kubectl get secret huawei-csi-secret -n huawei-csi Error from server (NotFound): secrets "huawei-csi-secret" not found

----结束

## 8.2.5 删除 RBAC 权限

#### 操作步骤

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意master节点。

#### 步骤2 删除RBAC权限。

● 当huawei-csi版本号高于2.2.15,执行以下命令进行删除,-n参数为命名空间,请按实际修改。

# kubectl -n huawei-csi -l provisioner=csi.huawei.com delete ServiceAccount,role,rolebinding,ClusterRole,ClusterRoleBinding

- 当huawei-csi版本号低于或等于2.2.15,按照以下步骤进行删除。
  - a. 执行以下命令创建delete-huawei-csi-rbac.sh文件,-n参数为命名空间,请按 实际修改。

# cat <<EOF > delete-huawei-csi-rbac.sh

kubectl delete ServiceAccount huawei-csi-controller -n huawei-csi

kubectl delete ServiceAccount huawei-csi-node -n huawei-csi

kubectl delete ClusterRole huawei-csi-attacher-runner -n huawei-csi

kubectl delete ClusterRole huawei-csi-driver-registrar-runner -n huawei-csi

kubectl delete ClusterRole huawei-csi-provisioner-runner -n huawei-csi

kubectl delete ClusterRole huawei-csi-resizer-runner -n huawei-csi

kubectl delete ClusterRole huawei-csi-snapshotter-runner -n huawei-csi

kubectl delete ClusterRole snapshot-controller-runner -n huawei-csi

kubectl delete ClusterRoleBinding huawei-csi-attacher-role -n huawei-csi

kubectl delete ClusterRoleBinding huawei-csi-driver-registrar-role -n huawei-csi

kubectl delete ClusterRoleBinding huawei-csi-provisioner-role -n huawei-csi

kubectl delete ClusterRoleBinding huawei-csi-resizer-role -n huawei-csi

kubectl delete ClusterRoleBinding huawei-csi-snapshotter-role -n huawei-csi

kubectl delete ClusterRoleBinding snapshot-controller-role -n huawei-csi

kubectl delete Role huawei-csi-resizer-cfg -n huawei-csi

kubectl delete Role huawei-csi-snapshotter-leaderelection -n huawei-csi

kubectl delete Role snapshot-controller-leaderelection -n huawei-csi

kubectl delete RoleBinding huawei-csi-resizer-role-cfg -n huawei-csi

kubectl delete RoleBinding huawei-csi-snapshotter-leaderelection -n huawei-csi

kubectl delete RoleBinding snapshot-controller-leaderelection -n huawei-csi

EOF

b. 执行以下命令删除RBAC权限。如果报NotFound错误,请忽略。 # sh delete-huawei-csi-rbac.sh

#### 步骤3 按照以下步骤检查RBAC权限是否已成功删除。

当huawei-csi版本号高于2.2.15,执行以下命令,-n参数为命名空间,请按实际修改。如果提示No resources found,则表示已成功删除。

# kubectl -n huawei-csi -l provisioner=csi.huawei.com get ServiceAccount,role,rolebinding,ClusterRole,ClusterRoleBinding

- 当huawei-csi版本号低于或等于2.2.15,按照以下步骤检查RBAC权限是否已成功 删除。
  - a. 执行以下命令创建check-huawei-csi-rbac.sh文件,-n参数为命名空间,请按 实际修改 。

# cat <<EOF > check-huawei-csi-rbac.sh

kubectl get ServiceAccount -n huawei-csi | grep huawei-csi

kubectl get ClusterRole -n huawei-csi | grep huawei-csi

kubectl get ClusterRoleBinding -n huawei-csi | grep huawei-csi

kubectl get Role -n huawei-csi | grep huawei-csi

kubectl get RoleBinding -n huawei-csi | grep huawei-csi

kubectl get ClusterRole snapshot-controller-runner -n huawei-csi --ignore-not-found=true kubectl get ClusterRoleBinding snapshot-controller-role -n huawei-csi --ignore-not-found=true kubectl get Role snapshot-controller-leaderelection -n huawei-csi --ignore-not-found=true kubectl get RoleBinding snapshot-controller-leaderelection -n huawei-csi --ignore-not-found=true

found=true

FOF

b. 执行以下命令,如果没有回显,表示RBAC权限已成功删除。 # sh check-huawei-csi-rbac.sh

#### ----结束

## 8.2.6 删除老版本镜像

如果需要在集群中删除huawei-csi镜像,需要在所有的worker节点都执行一遍删除操作。

以删除单个节点的镜像为例,操作步骤如下。

#### 前提条件

依赖该镜像的容器服务已经被停止,否则无法删除。

#### 操作步骤

步骤1 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录该worker节点。

步骤2 执行以下命令,查看现存的所有版本。

• 如果使用的docker, 执行docker image ls | grep huawei-csi命令。

# docker image ls | grep huawei-csi

REPOSITORY TAG IMAGE ID CREATED SIZE huawei-csi 2.2.15 b30b3a8b5959 2 weeks ago 79.7MB huawei-csi 3.0.0 14b854dba227 2 weeks ago 79.6MB

• 如果使用的containerd,执行crictl image ls | grep huawei-csi命令。

# crictl image ls | grep huawei-csi

REPOSITORY TAG IMAGE ID CREATED SIZE docker.io/library/huawei-csi 2.2.15 b30b3a8b5959 2 weeks ago 79.7MB docker.io/library/huawei-csi 3.0.0 14b854dba227 2 weeks ago 79.6MB

#### 步骤3 执行以下命令,删除老版本镜像:

如果使用的docker, 执行docker rmi <REPOSITORY>:<TAG>命令。。
 # docker rmi huawei-csi:2.2.15

• 如果使用的containerd,执行**crictl rmi** *<REPOSITORY>:<TAG>*命令。 # crictl rmi huawei-csi:2.2.15

**步骤4** 再次执行以下命令,检查镜像是否已成功删除。如果已没有要删除版本的记录,表示已成功删除该版本。

• 如果使用的docker,执行**docker image ls | grep huawei-csi**命令。

# docker image ls | grep huawei-csi

● 如果使用的containerd,执行**crictl image ls | grep huawei-csi**命令。 # crictl image ls | grep huawei-csi docker.io/library/huawei-csi 3.0.0 14b854dba2273 93.1MB

----结束

# 8.3( 可选 )卸载 Snapshot 依赖组件服务

#### 须知

- 请勿在存在快照时卸载Snapshot依赖组件服务,否则Kubernetes会自动删除所有的用户快照且无法恢复,请谨慎操作。详细说明请参见删除
   CustomResourceDefinition。
- 请勿在CSI升级时卸载Snapshot依赖组件服务。

#### 场景说明

- 1. 当前Kubernetes集群仅存在华为CSI,且不再使用华为CSI。
- 2. 在卸载前请确保使用华为CSI清理Kubernetes集群中的VolumeSnapshot,清理操作请参考**6.5.3.2 删除VolumeSnapshot**。

#### 操作步骤

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意master节点。

步骤2 执行以下命令,卸载Snapshot依赖组件服务。

# kubectl delete crd volumesnapshotclasses.snapshot.storage.k8s.io volumesnapshotcontents.snapshot.storage.k8s.io volumesnapshots.snapshot.storage.k8s.io

步骤3 执行以下命令,检查服务是否已成功卸载。

如果结果为空,表示已成功卸载。 # kubectl get crd | grep snapshot.storage.k8s.io

----结束

# 9 常用操作

- 9.1 更新CSI上配置的存储用户名或密码
- 9.2 更新huawei-csi的configmap对象
- 9.3 为huawei-csi新增后端
- 9.4 更新huawei-csi-controller服务
- 9.5 更新huawei-csi-node服务
- 9.6 修改日志输出模式
- 9.7 开启ReadWriteOncePod功能门
- 9.8 配置非root用户访问Kubernetes集群

## 9.1 更新 CSI 上配置的存储用户名或密码

当存储设备的用户名或密码发生变更时,需要更新CSI上的配置信息,否则huawei-csi服务无法正常工作。

#### 操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意master节点。
- 步骤2 执行chmod +x secretUpdate命令,给secretUpdate添加可执行权限。

# chmod +x secretUpdate

步骤3 执行./secretUpdate namespace命令,运行secretUpdate工具,并根据界面提示输入需要配置backend的编号。Configured为false时表示这个backend尚未配置,为true时表示这个backend已配置。

6 true strage-backend-06 [https://192.168.125.30:28443]

Please enter the backend number to configure (Enter 'exit' to exit): 3

#### 步骤4 根据界面提示输入用户名和密码来更新secret对象。

Name:strage-backend-03

Urls:[https://192.168.125.27:8088]

Please enter this backend user name: admin

Please enter this backend password:

Verifying user name and password. Please wait.....

The acount information of the backend strage-backend-03 has been configured successfully.

#### 步骤5 配置完成后,输入exit退出并保存配置。

Please enter the backend number to configure (Enter 'exit' to exit): *exit* Saving configuration. Please wait............
The configuration is saved successfully.

#### 步骤6 执行以下命令,重启huawei-csi-controller服务。

# kubectl get deployment huawei-csi-controller -o yaml -n=huawei-csi | kubectl replace --force -f -

#### 步骤7 执行以下命令,重启huawei-csi-node服务。

# kubectl get daemonset huawei-csi-node -o yaml -n=huawei-csi | kubectl replace --force -f -

#### 步骤8 执行kubectl get pod -A | grep huawei命令检查服务是否重启成功。

# kubectl get pod -A | grep huawei huawei-csi huawei-csi-controller-695b84b4d8-tg64l 7/7 **Running** 0 14s huawei-csi huawei-csi-node-g6f7z 3/3 **Running** 0 14s

----结束

# 9.2 更新 huawei-csi 的 configmap 对象

当您需要在已有的后端上增加存储池,或者更改已有的业务IP地址时,执行此操作。

## 9.2.1 使用 Helm 更新 configmap 对象

#### 操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意master节点。
- 步骤2 进入Helm工程的目录下,如果无法找到之前的Helm工程,则将组件包中的helm目录 拷贝到master节点的任意目录下,组件包路径请参考3.3 上传软件包组件。
- **步骤3** 备份安装CSI时使用的values.yaml文件,如果上次安装时使用的values.yaml文件无法 找到,可通过**helm get values** *helm-huawei-csi* -**n** *huawei-csi* -**a**命令查询。 # cp values.yaml values.yaml.bak
- **步骤4** 执行**vi** *values.yaml*命令打开文件,根据升级需要更新参数值,修改完成后,按**Esc**,并输入:**wq!**,保存修改。您可以更改**表4-1**中提供的所有参数,后端参数请谨慎修改。 # vi values.yaml
- 步骤5 执行helm upgrade helm-huawei-csi./ -n huawei-csi-f values.yaml命令升级Helm chart。升级命令将更新Deployment、daemonset和RBAC资源,但是secret资源不会 更新,如果后端信息发生更改,则必须手动更新secret,更新操作请参考9.1 更新CSI 上配置的存储用户名或密码。

其中,helm-huawei-csi为自定义的chart名称,huawei-csi为自定义的命名空间。

```
# helm upgrade helm-huawei-csi / -n huawei-csi
Release "helm-huawei-csi" has been upgraded. Happy Helming!
NAME: helm-huawei-csi
LAST DEPLOYED: Thu Jun 9 07:58:15 2022
NAMESPACE: huawei-csi
STATUS: deployed
REVISION: 2
TEST SUITE: None
```

----结束

# 9.2.2 手动更新 configmap 对象

#### 操作步骤

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。

步骤2 执行kubectl edit configmap huawei-csi-configmap -n huawei-csi命令,按l或 Insert进入编辑状态,修改相关参数。以iSCSI组网为例,详细参数说明请参见<mark>表</mark>9-1。 修改完成后,按Esc,并输入:wq!,保存修改。

#### 表 9-1 配置项描述

| 配置项                          | 参数格式 | 描述                      | 备注   |
|------------------------------|------|-------------------------|--|
| data."csi.json".backe<br>nds | 列表   | 必填,待接<br>入后端存储<br>设备列表。 | 配置的后端存储设备数量没有限制。<br>单个后端存储设备支持配置的字段,请参考表9-2。 |

#### 表 9-2 后端存储设备配置项描述

| 配置项     | 参数格式 | 描述                 | 备注                                   |
|---------|------|--------------------|--------------------------------------|
| storage | 字符串  | 必填,待接入存<br>储设备的类型。 | 对接分布式存储SAN场景固定填写"fusionstorage-san"。 |

| 配置项        | 参数格式 | 描述                                 | 备注   |
|------------|------|------------------------------------|--|
| name       | 字符串  | 存储后端名。                             | <ul><li>自定义字符串,支持大小写字母、数字、中划线组合。</li></ul>   |
|            |      |                                    | ● 该参数不支持修改。  |
| urls       | 列表   | 必填,待接入的<br>存储设备的管理<br>URL。         | 支持同一存储设备的一个或多个<br>管理URL,用逗号分割,当前仅<br>支持IPv4。例如:https://<br>192.168.125.20:8088   |
|            |      |                                    | <b>说明</b><br>1个存储设备有多个控制器,每个<br>控制器有1个管理URL,所以1个存<br>储设备可能存在多个管理URL。   |
| pools      | 列表   | 必填,被使用的<br>待接入存储设备<br>上的存储池名<br>称。 | <ul><li>支持同一存储设备上的一个或多个存储池,用逗号分割。</li><li>当前只支持新增存储池。</li></ul>  |
|            |      |                                    | <ul><li>可通过登录DeviceManager<br/>获取支持块存储服务的存储<br/>池。</li></ul>   |
| parameters | 字典   | iSCSI场景的可变<br>参数。                  | iSCSI场景protocol参数固定填写"iscsi"。  |
|            |      |                                    | portals参数填写待接入存储设<br>备的iSCSI业务IP地址,多个业<br>务IP地址请用逗号分割。   |
|            |      |                                    | iSCSI业务IP地址可通过登录<br>DeviceManager获取。以<br>OceanStor Dorado 6.x系列为<br>例:在DeviceManager管理界<br>面,选择"服务 > 网络 > 逻辑<br>端口",获取数据协议类型为<br>iSCSI的IP(其它系列请参照相<br>应的操作说明进行获取)。 |

步骤3 若后端参数storage、name或urls有修改,需要更新存储的用户名或密码。详细操作请参见9.1 更新CSI上配置的存储用户名或密码。

步骤4 执行以下命令,重启huawei-csi-controller服务。

# kubectl get deployment huawei-csi-controller -o yaml -n=huawei-csi | kubectl replace --force -f -

**步骤5** 执行以下命令,重启huawei-csi-node服务。

# kubectl get daemonset huawei-csi-node -o yaml -n=huawei-csi | kubectl replace --force -f -

步骤6 执行kubectl get pod -A | grep huawei命令,检查服务是否重启成功。

# kubectl get pod -A | grep huawei

huawei-csi huawei-csi-controller-695b84b4d8-tg64l 7/7 **Running** 0 14s huawei-csi huawei-csi-node-g6f7z 3/3 **Running** 0 14s

----结束

# 9.3 为 huawei-csi 新增后端

当您需要新增存储设备或存储池当做独立的后端时,执行此操作。

## 9.3.1 使用 Helm 新增后端

#### 操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。
- 步骤2 进入Helm工程的目录下,如果无法找到之前的Helm工程,则将组件包中的helm目录 拷贝到master节点的任意目录下,组件包路径请参考3.3 上传软件包组件。
- **步骤3** 备份安装CSI时使用的values.yaml文件,如果上次安装时使用的values.yaml文件无法 找到,可通过**helm get values** *helm-huawei-csi* -**n** *huawei-csi* -**a**命令查询。 # cp values.yaml values.yaml.bak
- **步骤4** 执行**vi** *values.yaml*命令打开配置文件,在backends下新增后端配置,示例如下所示, 修改完成后,按**Esc**,并输入:**wq!**,保存修改。

```
backends:
 - storage: "oceanstor-san"
  name: "***"
  urls:
    - "https://*.*.*:8088"
  pools:
  parameters:
    protocol: "iscsi"
    portals:
 - storage: "oceanstor-nas"
  name: "***"
   - "https://*.*.*:8088"
  pools:
  parameters:
    protocol: "nfs"
    portals:
     _ "* * * *
```

步骤5 执行helm upgrade helm-huawei-csi./ -n huawei-csi-f values.yaml命令升级Helm chart。升级命令将更新Deployment、daemonset和RBAC资源,但是secret资源不会更新,如果后端信息发生更改,则必须手动更新secret,更新操作请参考9.1 更新CSI上配置的存储用户名或密码。

```
其中,helm-huawei-csi为自定义的chart名称,huawei-csi为自定义的命名空间。
# helm upgrade helm-huawei-csi / -n huawei-csi
Release "helm-huawei-csi" has been upgraded. Happy Helming!
NAME: helm-huawei-csi
LAST DEPLOYED: Thu Jun 9 07:58:15 2022
NAMESPACE: huawei-csi
STATUS: deployed
REVISION: 2
TEST SUITE: None
```

#### ----结束

## 9.3.2 手动新增后端

#### 操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。
- 步骤2 配置多个后端,详细操作请参见7.1 配置多个后端。
- 步骤3 为新增的后端配置账号,详细操作请参见9.1 更新CSI上配置的存储用户名或密码。
- 步骤4 执行以下命令,重启huawei-csi-controller服务。
  # kubectl get deployment huawei-csi-controller -o yaml -n=huawei-csi | kubectl replace --force -f -
- 步骤5 执行以下命令,重启huawei-csi-node服务。
  # kubectl get daemonset huawei-csi-node -o yaml -n=huawei-csi | kubectl replace --force -f -
- 步骤6 执行kubectl get pod -A | grep huawei命令,检查服务是否重启成功。

```
# kubectl get pod -A | grep huawei
huawei-csi huawei-csi-controller-695b84b4d8-tg64l 7/7 Running 0 14s
huawei-csi huawei-csi-node-g6f7z 3/3 Running 0 14s
```

----结束

## 9.4 更新 huawei-csi-controller 服务

当您需要更新huawei-csi-controller服务时, 例如添加快照功能或者扩容功能,执行此操作。

## 9.4.1 使用 Helm 更新 controller 服务

#### 操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意master节点。
- 步骤2 进入Helm工程的目录下,如果无法找到之前的Helm工程,则将组件包中的helm目录 拷贝到master节点的任意目录下,组件包路径请参考3.3 上传软件包组件。
- **步骤3** 备份安装CSI时使用的values.yaml文件,如果上次安装时使用的values.yaml文件无法找到,可通过**helm get values** *helm-huawei-csi* -**n** *huawei-csi* -**a**命令查询。

# cp values.yaml values.yaml.bak

**步骤4** 执行**vi** *values.yaml*命令打开文件,修改controller相关参数,示例如下所示。修改完成后,按**Esc**,并输入:**wq!**,保存修改。

```
kubernetes:
namespace: huawei-csi

images:
# The image name and tag for the attacher, provisioner and registrar sidecars. These must match the appropriate Kubernetes version.
sidecar:
csiAttacher: k8s.gcr.io/sig-storage/csi-attacher:v3.3.0
csiProvisioner: k8s.gcr.io/sig-storage/csi-provisioner:v3.0.0
csiResizer: k8s.gcr.io/sig-storage/csi-resizer:v1.3.0
livenessProbe: k8s.gcr.io/sig-storage/livenessprobe:v2.5.0
csiSnapshotter: k8s.gcr.io/sig-storage/csi-snapshotter:v4.2.1
snapshotController: k8s.gcr.io/sig-storage/snapshot-controller:v4.2.1
```

```
# The image name and tag for the Huawei CSI Service container
 # Replace the appropriate tag name
 huaweiCSIService: huawei-csi:3.0.0
# The CSI driver parameter configuration
csi_driver:
 driverName: csi.huawei.com
 endpoint: /csi/csi.sock
 backendUpdateInterval: 60
 controllerLogging:
  module: file
  level: info
  fileDir: /var/log/huawei
  fileSize: 20M
  maxBackups: 9
huaweiCsiController:
 replicas: 1 # Default number of controller replicas
# Default image pull policy for sidecar container images
sidecarImagePullPolicy: "IfNotPresent"
# Default image pull policy for Huawei plugin container images
huaweiImagePullPolicy: "IfNotPresent"
# Flag to enable or disable snapshot (Optional)
snapshot:
 enable: true
# Flag to enable or disable resize (Optional)
resizer:
 enable: true
```

步骤5 执行helm upgrade helm-huawei-csi./ -n huawei-csi命令升级Helm chart。

其中,helm-huawei-csi为需要升级的chart名称,./为使用当前目录下Helm工程,huawei-csi为chart所在的命名空间。

# helm upgrade helm-huawei-csi ./ -n huawei-csi Release "helm-huawei-csi" has been upgraded. Happy Helming! NAME: helm-huawei-csi LAST DEPLOYED: Thu Jun 9 07:58:15 2022 NAMESPACE: huawei-csi STATUS: deployed

STATUS: deployed REVISION: 2 TEST SUITE: None

#### ----结束

## 9.4.2 手动更新 controller 服务

#### 操作步骤

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。

步骤2 卸载huawei-csi-controller服务,请参见8.2.2 卸载huawei-csi-controller服务。

步骤3 删除RBAC权限,请参见8.2.5 删除RBAC权限。

步骤4 创建RBAC权限,请参见步骤4。

步骤5 启动controller服务,请参见步骤5。

步骤6 完成huawei-csi服务部署后,可执行kubectl get pod -A | grep huawei-csi-controller命令检查服务是否启动。

# kubectl get pod -A | grep huawei-csi-controller huawei-csi huawei-csi-controller-695b84b4d8-tq64l 7/7 Running 0 14s

----结束

# 9.5 更新 huawei-csi-node 服务

当您需要更新huawei-csi-node服务时,执行此操作。

## 9.5.1 使用 Helm 更新 node 服务

#### 操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意master节点。
- 步骤2 进入Helm工程的目录下,如果无法找到之前的Helm工程,则将组件包中的helm目录 拷贝到master节点的任意目录下,组件包路径请参考3.3 上传软件包组件。
- 步骤3 备份安装CSI时使用的values.yaml文件,如果上次安装时使用的values.yaml文件无法找到,可通过**helm get values** *helm-huawei-csi* **-n** *huawei-csi* **-a**命令查询。

# cp values.yaml values.yaml.bak

**步骤4** 执行**vi** *values.yaml*命令打开文件,修改node相关参数,示例如下所示。修改完成后,按**Esc**,并输入:**wq!**,保存修改。

```
kubernetes:
 namespace: huawei-csi
images:
 # The image name and tag for the attacher, provisioner and registrar sidecars. These must match the
appropriate Kubernetes version.
 sidecar:
  livenessProbe: k8s.gcr.io/sig-storage/livenessprobe:v2.5.0
  registrar: k8s.gcr.io/sig-storage/csi-node-driver-registrar:v2.3.0
 # The image name and tag for the Huawei CSI Service container
 # Replace the appropriate tag name
 huaweiCSIService: huawei-csi:3.0.0
# The CSI driver parameter configuration
csi_driver:
 driverName: csi.huawei.com
 endpoint: /csi/csi.sock
 connectorThreads: 4
 volumeUseMultipath: true # Flag to enable or disable volume multipath access
 scsiMultipathType: DM-multipath #Required, if volume-use-multipath is set to TRUE
 nvmeMultipathType: HW-UltraPath-NVMe #Required, if volume-use-multipath is set to TRUE
 scanVolumeTimeout: 3
 backendUpdateInterval: 60
 nodeLogging:
  module: file
  level: info
  fileDir: /var/log/huawei
  fileSize: 20M
  maxBackups: 9
# Default image pull policy for sidecar container images
sidecarImagePullPolicy: "IfNotPresent"
```

# Default image pull policy for Huawei plugin container images

huaweiImagePullPolicy: "IfNotPresent"

步骤5 执行helm upgrade helm-huawei-csi./ -n huawei-csi命令升级Helm chart。

其中,helm-huawei-csi为需要升级的chart名称,./为使用当前目录下Helm工程,huawei-csi为chart所在的命名空间。

# helm upgrade helm-huawei-csi ./ -n huawei-csi Release "helm-huawei-csi" has been upgraded. Happy Helming! NAME: helm-huawei-csi LAST DEPLOYED: Thu Jun 9 07:58:15 2022 NAMESPACE: huawei-csi STATUS: deployed REVISION: 2 TEST SUITE: None

#### ----结束

## 9.5.2 手动更新 node 服务

#### 操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。
- 步骤2 卸载huawei-csi-node服务,请参见8.2.1 卸载huawei-csi-node服务。
- 步骤3 安装新的huawei-csi-node服务,请参见4.2.3 启动huawei-csi服务。
- 步骤4 完成huawei-csi服务部署后,可执行kubectl get pod -A | grep huawei-csi-node命令检查服务是否启动。

# kubectl get pod -A | grep huawei-csi-node huawei-csi huawei-csi-node-g6f7z 3/3 Running 0 14s

----结束

## 9.6 修改日志输出模式

huawei-csi支持两种日志输出模式,分别是file和console。file指的是输出到固定的日志目录(例如:/var/log/huawei);console指的是输出到容器标准目录。用户可以根据自身需求自行设置日志输出模式,默认为file.

## 9.6.1 使用 Helm 修改 controller 或 node 服务的日志输出模式

#### 操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。
- 步骤2 进入Helm工程的目录下,如果无法找到之前的Helm工程,则将组件包中的helm目录 拷贝到master节点的任意目录下,组件包路径请参考3.3 上传软件包组件。
- **步骤3** 备份安装CSI时使用的values.yaml文件,如果上次安装时使用的values.yaml文件无法 找到,可通过**helm get values** *helm-huawei-csi* **-n** *huawei-csi* **-a**命令查询。 # cp values.yaml values.yaml.bak
- **步骤4** 执行**vi** *values.yaml*命令打开文件,修改controller或node的日志相关参数,示例如下所示。修改完成后,按**Esc**,并输入:**wq!**,保存修改。

# The CSI driver parameter configuration csi driver:

```
controllerLogging:
module: file
level: info
fileDir: /var/log/huawei
fileSize: 20M
maxBackups: 9
nodeLogging:
module: file
level: info
fileDir: /var/log/huawei
fileSize: 20M
maxBackups: 9
```

步骤5 执行helm upgrade helm-huawei-csi./ -n huawei-csi命令升级Helm chart。

其中,helm-huawei-csi为需要升级的chart名称,./为使用当前目录下Helm工程,huawei-csi为chart所在的命名空间。

```
# helm upgrade helm-huawei-csi // -n huawei-csi
Release "helm-huawei-csi" has been upgraded. Happy Helming!
NAME: helm-huawei-csi
LAST DEPLOYED: Thu Jun 9 07:58:15 2022
NAMESPACE: huawei-csi
STATUS: deployed
REVISION: 2
TEST SUITE: None
```

#### ----结束

## 9.6.2 手动修改 huawei-csi-controller 服务的日志输出模式

当您需要为huawei-csi-controller服务设置日志输出模式,执行此操作。

#### 操作步骤

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。

步骤2 卸载huawei-csi-controller服务,请参见8.2.2 卸载huawei-csi-controller服务。

步骤3 进入"deploy"目录下,组件包路径参考3.3 上传软件包组件。

**步骤4** 执行**vi** *huawei-csi-controller.yaml*命令,修改yaml文件。按**l**或**Insert**进入编辑状态, 修改以下相关参数,详细请参考<mark>表</mark>9-3。修改完成后,按**Esc**,并输入:**wq!**,保存修 改。

# args: - "--endpoint=\$(CSI\_ENDPOINT)" - "--controller" - "--containerized" - "--driver-name=csi.huawei.com" - "--logqingModule=file"

- "--logLevel=info" - "--logFileDir=/var/log/huawei"

- "--logFileSize=20M"

- "--maxBackups=9"

#### 表 9-3 日志输出参数说明

| 配置项           | 描述                    | 备注                                    |
|---------------|-----------------------|---------------------------------------|
| loggingModule | huawei-csi日志输出<br>模式。 | 当前可以支持"file"和<br>"console",默认为"file"。 |

| 配置项         | 描述                                      | 备注   |
|-------------|---|--|
| logLevel    | huawei-csi日志输出<br>级别。                   | 支持的级别有:debug,info,<br>warning,error,fatal。默认的级<br>别为info。      |
| logFileDir  | huawei-csi日志在file<br>输出模式下的日志目<br>录。    | 仅当"loggingModule"为"file"<br>时生效,默认的日志目录为<br>"/var/log/huawei"。 |
| logFileSize | huawei-csi日志在file<br>输出模式下单个日志<br>文件大小。 | 仅当"loggingModule"为"file"<br>时生效,默认的日志文件大小为<br>20MiB。           |
| maxBackups  | huawei-csi日志在file<br>输出模式下日志文件<br>备份上限。 | 仅当"loggingModule"为"file"<br>时生效,默认的日志文件备份数量<br>为9个。            |

步骤5 执行以下命令,启动controller服务。

# kubectl create -f huawei-csi-controller-snapshot-v1.yaml

步骤6 完成huawei-csi服务部署后,可执行kubectl get pod -A -o wide | grep huawei命令检查服务是否启动。

# kubectl get pod -A -o wide | grep huawei huawei-csi huawei-csi-controller-b59577886-qqzm8 7/7 Running 0 18h 10.244.1.67 node <none> <none>

#### 步骤7 查看huawei-csi-controller服务的日志。

- 如果设置"loggingModule"为"file",请登录到具体的node节点,然后进入到 logFileDir设置的日志目录,执行以下命令,查看huawei-csi-controller日志。
  # tail -f huawei-csi-controller
- 如果设置"loggingModule"为"console",请执行以下命令,查看huawei-csi-controller日志

# kubectl logs *huawei-csi-controller* -c huawei-csi-driver -n huawei-csi

----结束

## 9.6.3 手动修改 huawei-csi-node 服务的日志输出模式

当您需要为huawei-csi-node服务设置日志输出模式,执行此操作。

#### 操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意master节点。
- 步骤2 卸载huawei-csi-node服务,请参见8.2.1 卸载huawei-csi-node服务。
- 步骤3 执行vi huawei-csi-node.yaml命令,修改yaml文件。按l或Insert进入编辑状态,修改相关参数。修改完成后,按Esc,并输入:wq!,保存修改。编写huawei-csi-node.yaml文件(可参考软件包中deploy/huawei-csi-node.yaml示例文件)。参数详情可参考表9-4。

args

- "--endpoint=/csi/csi.sock"
- "--containerized"

- "--driver-name=csi.huawei.com"
- "--volume-use-multipath=false"
- "--loggingModule=file"
- "--logLevel=info"
- "--logFileDir=/var/log/huawei"
- "--logFileSize=20M"
- "--maxBackups=9"

#### 表 9-4 日志输出参数说明

| 配置项           | 描述                                     | 备注   |
|---------------|--|--|
| loggingModule | huawei-csi日志输出模<br>式                   | 当前可以支持"file"和<br>"console",默认为"file"。                        |
| logLevel      | huawei-csi日志输出级<br>别                   | 支持的级别有:debug,info,<br>warning,error,fatal。默认的级<br>别为info。    |
| logFileDir    | huawei-csi日志在file输<br>出模式下的日志目录        | 仅当"loggingModule"为<br>"file"时生效,默认的日志目录<br>为/var/log/huawei。 |
| logFileSize   | huawei-csi日志在file输<br>出模式下单个日志文件<br>大小 | 仅当"loggingModule"为<br>"file"时生效,默认的日志文件<br>大小为20MiB。         |
| maxBackups    | huawei-csi日志在file输<br>出模式下日志文件备份<br>上限 | 仅当"loggingModule"为<br>"file"时生效,默认的日志文件<br>备份数量为9个。          |

#### 步骤4 执行以下命令,启动node服务。

# kubectl create -f huawei-csi-node.yaml

# 步骤5 完成huawei-csi服务部署后,可执行kubectl get pod -A -o wide | grep huawei-csi-node命令检查服务是否启动。

# kubectl get pod -A | grep huawei-csi-node huawei-csi huawei-csi-node-4sfwr 3/3 Running 0 18h 10.244.1.68 node <none> <none>

#### 步骤6 查看huawei-csi-node服务的日志。

- 如果设置"loggingModule"为"file",请登录到具体的node节点,然后进入到logFileDir设置的日志目录,执行以下命令,查看huawei-csi-node日志。
  # tail -f huawei-csi-node
- 如果设置"loggingModule"为"console",请执行以下命令,查看huawei-csi-node日志

# kubectl logs *huawei-csi-node* -c huawei-csi-driver -n huawei-csi

#### ----结束

# 9.7 开启 ReadWriteOncePod 功能门

ReadWriteOnce访问模式是Kubernetes v1.22版本为PV和PVC引入的第四种访问模式。如果您使用ReadWriteOncePod访问模式的PVC创建一个Pod,Kubernetes会确保该Pod是整个集群中唯一可以读取或写入该PVC的Pod。

由于ReadWriteOncePod访问模式在当前已发布的Kubernetes v1.22/1.23/1.24版本中是alpha特性,需要先在kube-apiserver、kube-scheduler和kubelet的feature-gates中开启ReadWriteOncePod特性才能使用。

#### 操作步骤

步骤1 为kube-apiserver启用ReadWriteOncePod功能门。

- 1. 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意master节点。
- 2. 执行**vi /etc/kubernetes/manifests/kube-apiserver.yaml**命令,按**l**或**Insert**进入编辑状态,为 kube-apiserver容器添加参数--feature-gates=ReadWriteOncePod=true。修改完成后,按**Esc**,并输入**:wq!**,保存修改。

spec: containers: - command: - kube-apiserver

- --feature-gates=ReadWriteOncePod=true

□ 说明

在编辑完成后,Kubernetes会自动应用更新,不需要手动更新。

步骤2 为kube-scheduler启用ReadWriteOncePod功能门。

- 1. 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意master节点。
- 执行vi /etc/kubernetes/manifests/kube-scheduler.yaml命令,按I或Insert进入编辑状态,为kube-scheduler容器添加参数--feature-gates=ReadWriteOncePod=true。修改完成后,按Esc,并输入:wq!,保存修改。

spec:
 containers:
 - command:
 - kube-scheduler
 - --feature-gates=ReadWriteOncePod=true

□ 说明

在编辑完成后,Kubernetes会自动应用更新,不需要手动更新。

步骤3 为kubelet启用ReadWriteOncePod功能门。

#### 须知

由于动态Kubelet配置功能在v1.22中已弃用,并且在v1.24中删除,因此集群中每个worker节点上的kubelet都需要执行以下操作。

- 1. 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意worker节点。
- 2. 执行**vi /var/lib/kubelet/config.yaml**命令,按**I**或**Insert**进入编辑状态,为 KubeletConfiguration对象的feature-gates字段添加ReadWriteOncePod: true,

如果没有feature-gates字段请一并添加。修改完成后,按**Esc**,并输入:wq!,保存

apiVersion: kubelet.config.k8s.io/v1beta1

featureGates:

ReadWriteOncePod: true

#### □ 说明

kubelet配置文件的默认路径为/var/lib/kubelet/config.yaml,请根据实际情况填写。

在配置完成后,执行systemctl restart kubelet命令重启kubelet。

----结束

## 9.8 配置非 root 用户访问 Kubernetes 集群

#### 操作步骤

步骤1 拷贝Kubernetes集群的认证文件,/etc/kubernetes/admin.conf修改为实际使用的认证

\$ mkdir -p \$HOME/.kube

\$ sudo cp -i /etc/kubernetes/admin.conf \$HOME/.kube/config

步骤2 修改认证文件的用户与用户组。

\$ sudo chown \$(id -u):\$(id -g) \$HOME/.kube/config

步骤3 配置当前用户的KUBECONFIG环境变量,以Ubuntu 20.04举例如下。

\$ echo "export KUBECONFIG=\$HOME/.kube/config" >> ~/.bashrc

\$ source ~/.bashrc

----结束

# **10** FAQ

- 10.1 查看日志信息
- 10.2 Kubernetes平台第一次搭建时, iscsi\_tcp服务没有正常启动,导致创建Pod失败
- 10.3 启动huawei-csi-node失败,提示错误为: "/var/lib/iscsi is not a directory"
- 10.4 集群中worker节点宕机并恢复后,Pod完成failover,但是Pod所在源主机出现盘符残留
- 10.5 启动huawei-csi服务时,服务启动异常, 状态显示InvalidImageName
- 10.6 创建PVC时, PVC的状态为Pending
- 10.7 删除PVC前,PVC的状态为Pending
- 10.8 创建Pod时,Pod的状态为ContainerCreating
- 10.9 创建Pod时,Pod的状态长时间处于ContainerCreating状态

## 10.1 查看日志信息

#### 查看配置 secret 时的日志

步骤1 执行cd /var/log/huawei命令, 进入日志目录

# cd /var/log/huawei

步骤2 执行以下命令,查看huawei-csi-install日志

# vi huawei-csi-install

----结束

#### 查看 huawei-csi-controller 服务的日志

步骤1 执行以下命令,获取huawei-csi-controller所在的节点

# kubectl get pod -A -o wide | grep huawei huawei-csi huawei-csi-controller-695b84b4d8-tg64l 7/7 **Running** 0 14s <host1-ip> <host1-name> <none>

步骤2 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的huawei-csi-controller节点

#### 步骤3 执行cd /var/log/huawei命令,进入日志目录

# cd /var/log/huawei

#### 步骤4 执行以下命令,查看容器自定义输出日志

# vi huawei-csi-controller

#### 步骤5 执行cd /var/log/containers命令, 进入到容器目录

# cd /var/log/containers

#### 步骤6 执行以下命令, 查看容器标准输出日志

# vi huawei-csi-controller-<name>\_huawei-csi\_huawei-csi-driver-<contrainer-id>.log

----结束

#### 查看 huawei-csi-node 服务的日志

**步骤1** 执行以下命令,获取huawei-csi-node所在的节点

# kubectl get pod -A -o wide | grep huawei

huawei-csi huawei-csi-node-q6f7z 3/3 Runnina 0 14s <host2-ip> <host2-

name> <none>

步骤2 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的 huawei-csi-node节点

#### 步骤3 执行cd /var/log/huawei命令,进入日志目录

# cd /var/log/huawei

#### **步骤4** 执行以下命令,查看容器自定义输出日志

# vi huawei-csi-node

#### 步骤5 执行cd /var/log/containers命令, 进入到容器目录

# cd /var/log/containers

#### 步骤6 执行以下命令, 查看容器标准输出日志

# vi huawei-csi-node-<name>\_huawei-csi\_huawei-csi-driver-<contrainer-id>.log

----结束

## 10.2 Kubernetes 平台第一次搭建时, iscsi\_tcp 服务没有正 常启动,导致创建 Pod 失败

#### 现象描述

创建Pod时报错,在/var/log/huawei-csi-node日志中报错 " Cannot connect ISCSI portal \*.\*.\*.\*: libkmod: kmod module insert module: could not find module by name='iscsi tcp'。

#### 根因分析

搭建Kubernete和安装iSCSI服务后, iscsi\_tcp服务可能会被停掉,可通过执行**lsmod |** grep iscsi | grep iscsi\_tcp查看服务是否被停掉。

# lsmod | grep iscsi | grep iscsi\_tcp iscsi\_tcp 18333 6 25146 1 iscsi\_tcp libiscsi\_tcp

57233 2 libiscsi\_tcp,iscsi\_tcp lihiscsi scsi\_transport\_iscsi 99909 3 iscsi\_tcp,libiscsi

#### 解决措施或规避方法

执行以下命令,手动加载iscsi\_tcp服务。

# modprobe iscsi\_tcp
# lsmod | grep iscsi | grep iscsi\_tcp
iscsi\_tcp 18333 6
libiscsi\_tcp 25146 1 iscsi\_tcp

## 10.3 启动 huawei-csi-node 失败,提示错误为: "/var/lib/iscsi is not a directory"

#### 现象描述

启动huawei-csi-node时,无法启动huawei-csi-node服务, 使用**kubectl describe daemonset huawei-csi-node -n huawei-csi**命令查看,提示错误为: "/var/lib/iscsi is not a directory"

#### 根因分析

huawei-csi-node中容器内部无/var/lib/iscsi目录

#### 解决措施或规避方法

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。
- **步骤2** 执行以下命令,删除huawei-csi-node服务(huawei-csi-node.yaml为**4.2.3 启动** huawei-csi服务中步骤6的配置文件)。

# kubectl delete -f huawei-csi-node.yaml

- **步骤3** 执行以下命令,查看huawei-csi-node服务,如果没有回显,表示删除完成。 # kubectl get pod -A | grep huawei-csi-node
- 步骤4 执行vi huawei-csi-node.yaml命令,修改yaml文件。按l或Insert进入编辑状态,将huawei-csi-node.yaml -> volumes -> iscsi-dir -> hostPath中"path"设置为"/var/lib/iscsi",删除"type"行。修改完成后,按Esc,并输入:wq!,保存修改。编写huawei-csi-node.yaml文件(可参考软件包中deploy/huawei-csi-node.yaml示例文件)。
- 步骤5 执行以下命令,启动node服务。

# kubectl create -f huawei-csi-node.yaml

步骤6 完成huawei-csi服务部署后,可执行kubectl get pod -A | grep huawei-csi-node命令检查服务是否启动。

# kubectl get pod -A | grep huawei-csi-node huawei-csi huawei-csi-node-q6f7z 3/3 Running 0 14s

#### ----结束

## 10.4 集群中 worker 节点宕机并恢复后,Pod 完成failover,但是 Pod 所在源主机出现盘符残留

#### 现象描述

worker节点 A上运行Pod, 并通过CSI挂载外置块设备到该Pod; 异常掉电节点worker节点A; Kubernetes平台会在感知到节点故障后,将Pod切换至worker节点B; 恢复worker节点A, 节点A上的盘符会从正常变为故障。

#### 环境配置

Kubernetes版本: 1.18及以上

存储类型: 块存储

#### 根因分析

worker节点A恢复后,Kubernetes会向存储发起解除映射操作,但是不会发起主机侧的移除盘符操作。在Kubernetes解除映射后,worker节点A上就会出现盘符残留。

#### 解决措施或规避方法

目前的解决方法只能人工介入,手动清理掉主机的残留盘符(或者再次重启主机,利用主机重启过程中扫盘机制,清理掉残留盘符)。具体方法如下:

#### 步骤1 排查主机的残留盘符。

1. 执行multipath -ll命令,判断是否存在多路径状态异常的DM多路径设备:

如下图:路径状态为failed faulty running表示异常,对应的DM多路径设备为dm-12,关联的SCSI磁盘为sdi和sdj,在配置多条路径时,会有多个SCSI磁盘。记录这些SCSI磁盘。

# multipath -ll mpathb (3618cf24100f8f457014a764c000001f6) dm-12 HUAWEI ,XSG1 size=100G features='0' hwhandler='0' wp=rw `-+- policy='service-time 0' prio=-1 status=active |- 39:0:0:1 sdi 8:48 failed faulty running `- 38:0:0:1 sdj 8:64 failed faulty running

- 是 => 步骤1.2。
- **否** => 不涉及。
- 2. 判断残留的DM多路径设备是否可读。

执行**dd** if=/dev/dm-xx of=/dev/null count=1 bs=1M iflag=direct命令 (dm-xx为**步骤1.1**查到的设备号)

如果返回结果为: Input/output error,且读取数据为"0 bytes (0 B) copied",表示该设备不可读。

#dd if=/dev/dm-12 of=/dev/null count=1 bs=1M iflag=direct dd: error reading '/dev/dm-12': Input/output error 0+0 records in 0+0 records out 0 bytes (0 B) copied, 0.0236862 s, 0.0 kB/s

- 是 => 记录残留的dm-xx设备以及关联磁盘号(见步骤1.1 ),执行清理步骤。

- 命令卡死 => 步骤1.3
- **其他** => 联系技术支持。
- 3. 在另一窗口再次登录该节点。
  - a. 执行以下命令,查看卡死的进程。

# ps -ef | grep dm-12 | grep -w dd root 21725 9748 0 10:33 pts/10 00:00:00 dd if=/dev/dm-12 of=/dev/null count=1 bs=10M iflag=direct

b. 将该pid杀死。 # kill -9 *pid* 

c. 记录残留的dm-xx设备以及关联磁盘号(见步骤1.1),执行清理步骤。

#### 步骤2 清理主机的残留盘符。

根据步骤1获取的DM多路径设备,执行multipath -f /dev/dm-\*命令,清理残留的多路径聚合设备信息。

# multipath -f /dev/dm-12

如果执行报错,请联系技术支持。

2. 清理残留的SCSI磁盘,根据"排查方法"获取的残留磁盘的盘符,依次执行命令:

echo 1 > /sys/block/xxxx/device/delete

配置多条多路径时,依次根据盘符清除,本次残留路径为sdi/sdj:

# echo 1 > /sys/block/sdi/device/delete # echo 1 > /sys/block/sdj/device/delete

如果执行报错,请联系技术支持。

3. 确认DM多路径设备和SCSI磁盘信息是否已经清理干净。

依次执行"multipath -ll"、"ls -l /sys/block/"、"ls -l /dev/disk/by-id/"命令,查询的多路径和磁盘信息显示,残留的dm-12和SCSI磁盘sdi/sdj均已消失,则证明清理完成。

```
# multipath -ll
mpathb (3618cf24100f8f457014a764c000001f6) dm-3 HUAWEI ,XSG1
size=100G features='0' hwhandler='0' wp=rw
-+- policy='service-time 0' prio=-1 status=active
|- 39:0:0:1
                sdd 8:48 active ready running
  - 38:0:0:1
                sde 8:64 active ready running
mpathn (3618cf24100f8f457315a764c000001f6) dm-5 HUAWEI ,XSG1
size=100G features='0' hwhandler='0' wp=rw
`-+- policy='service-time 0' prio=-1 status=active
|- 39:0:0:2

`- 38:0:0:2
                sdc 8:32 active ready running
                sdb 8:16 active ready running
# ls -l /sys/block/
total 0
lrwxrwxrwx 1 root root 0 Aug 11 19:56 dm-0 -> ../devices/virtual/block/dm-0
lrwxrwxrwx 1 root root 0 Aug 11 19:56 dm-1 -> ../devices/virtual/block/dm-1
lrwxrwxrwx 1 root root 0 Aug 11 19:56 dm-2 -> ../devices/virtual/block/dm-2
lrwxrwxrwx 1 root root 0 Aug 11 19:56 dm-3 -> ../devices/virtual/block/dm-3
lrwxrwxrwx 1 root root 0 Aug 11 19:56 sdb -> ../devices/platform/host35/session2/
target35:0:0/35:0:0:1/block/sdb
lrwxrwxrwx 1 root root 0 Aug 11 19:56 sdc -> ../devices/platform/host34/
target34:65535:5692/34:65535:5692:0/block/sdc
lrwxrwxrwx 1 root root 0 Aug 11 19:56 sdd -> ../devices/platform/host39/session6/
target39:0:0/39:0:0:1/block/sdd
lrwxrwxrwx 1 root root 0 Aug 11 19:56 sde -> ../devices/platform/host38/session5/
target38:0:0/38:0:0:1/block/sde
lrwxrwxrwx 1 root root 0 Aug 11 19:56 sdh -> ../devices/platform/host39/session6/
target39:0:0/39:0:0:3/block/sdh
lrwxrwxrwx 1 root root 0 Aug 11 19:56 sdi -> ../devices/platform/host38/session5/target38:0:0/38:0:0:3/
```

ls -l /dev/disk/by-id/

```
lrwxrwxrwx 1 root root 10 Aug 11 19:57 dm-name-mpathb -> ../../dm-3
lrwxrwxrwx 1 root root 10 Aug 11 19:58 dm-name-mpathn -> ../../dm-5
lrwxrwxrwx 1 root root 10 Aug 11 19:57 dm-uuid-mpath-3618cf24100f8f457014a764c000001f6 -> ../../dm-3
lrwxrwxrwx 1 root root 10 Aug 11 19:58 dm-uuid-mpath-3618cf24100f8f457315a764c000001f6 -> ../../dm-5
lrwxrwxrwx 1 root root 9 Aug 11 19:57 scsi-3618cf24100f8f457014a764c000001f6 -> ../../sdd
lrwxrwxrwx 1 root root 9 Aug 11 19:57 scsi-3618cf24100f8f45712345678000103e8 -> ../../sdi
lrwxrwxrwx 1 root root 9 Aug 3 15:17 scsi-3648435a10058805278654321ffffffff -> ../../sdc
lrwxrwxrwx 1 root root 9 Aug 2 14:49 scsi-36886030000020aff44cc0d060c987f1 -> ../../sdd
lrwxrwxrwx 1 root root 9 Aug 3 15:17 wwn-0x618cf24100f8f45712345678000103e8 -> ../../sdd
lrwxrwxrwx 1 root root 9 Aug 3 15:17 wwn-0x618cf24100f8f45712345678000103e8 -> ../../sdb
lrwxrwxrwx 1 root root 9 Aug 3 15:17 wwn-0x648435a10058805278654321ffffffff -> ../../sdb
lrwxrwxrwx 1 root root 9 Aug 3 15:17 wwn-0x648435a10058805278654321ffffffff -> ../../sdb
lrwxrwxrwx 1 root root 9 Aug 2 14:49 wwn-0x68886030000020aff44cc0d060c987f1 -> ../../sdc
```

#### ----结束

## 10.5 启动 huawei-csi 服务时,服务启动异常, 状态显示 InvalidImageName

#### 现象描述

启动huawei-csi时,无法启动huawei-csi服务(huawei-csi-controller服务或者 huawei-csi-node服务),使用kubectl get pod -A | grep huawei命令查看,显示状态 为InvalidImageName

```
# kubectl get pod -A | grep huawei
huawei-csi huawei-csi-controller-fd5f97768-qlldc 6/7 InvalidImageName 0 16s
huawei-csi huawei-csi-node-25txd 2/3 InvalidImageName 0 15s
```

#### 根因分析

controller和node的yaml配置文件中,配置Huawei CSI的镜像版本号错误。例如:

```
...
- name: huawei-csi-driver
image: huawei-csi:3.0.0
...
```

#### 解决措施或规避方法

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。

**步骤2** 执行以下命令,修改huawei-csi-node服务的配置文件。按l或lnsert进入编辑状态,修改相关参数。修改完成后,按**Esc**,并输入:**wq!**,保存修改。

# kubectl edit daemonset huawei-csi-node -o yaml -n=huawei-csi

#### □ 说明

● 示例yaml文件中huawei-csi-driver的参数image配置项,修改huawei-csi:\*.\*.\*应替换为前面 制作的华为CSI镜像<*名称*>:<*版本号*>。

```
containers:
...
- name: huawei-csi-driver
image: huawei-csi:3.0.0
```

**步骤3** 执行以下命令,修改huawei-csi-controller服务的配置文件。按l或lnsert进入编辑状态,修改相关参数。修改完成后,按**Esc**,并输入:**wq!**,保存修改。

# kubectl edit deployment huawei-csi-controller -o yaml -n=huawei-csi

#### □ 说明

● 示例yaml文件中huawei-csi-driver的参数image配置项,修改huawei-csi:\*.\*.\*应替换为前面 制作的华为CSI镜像<*名称*>:<*版本号*>。

containers:

 name: huawei-csi-driver image: huawei-csi:3.0.0

步骤4 等待huawei-csi-node和huawei-csi-controller服务启动。

步骤5 执行以下命令,查看huawei csi服务是否启动。

```
# kubectl get pod -A | grep huawei
huawei-csi huawei-csi-controller-58799449cf-zvhmv 7/7 Running 0 2m29s
huawei-csi huawei-csi-node-7fxh6 3/3 Running 0 12m
```

----结束

## 10.6 创建 PVC 时, PVC 的状态为 Pending

#### 现象描述

执行完成PVC的创建操作,一段时间后,PVC的状态仍然处于Pending。

#### 根因分析

原因1:由于没有提前创建指定名称的StorageClass,导致Kubernetes在创建PVC时无法找到指定StorageClass名称。

原因2:由于存储池能力和StorageClass能力不匹配,导致huawei-csi选择存储池失败。

原因3:由于存储RESTful接口执行返回具体错误码(例如:50331651),导致huawei-csi在执行创建PVC时失败。

原因4:由于存储在huawei-csi设定的超时时间内没有返回,huawei-csi向Kubernetes返回超时错误。

原因5: 其他原因。

#### 解决措施或规避方法

创建PVC时,如果PVC处于Pending状态,需要根据以下不同的原因采取不同的解决措施。

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。

步骤2 执行以下命令,查看PVC的详细信息。

# kubectl describe pvc mypvc

步骤3 根据PVC详细信息中Events信息,执行相应操作。

• 如果由原因1导致PVC处于Pending状态, ,执行以下步骤。

```
Events:

Type Reason Age From Message

Warning ProvisioningFailed 0s (x15 over 3m24s) persistentvolume-controller

storageclass.storage.k8s.io "mysc" not found
```

- a. 删除PVC,详细请参考6.3.5 删除PVC。
- b. 创建StorageClass,详细请参考6.1.1 创建StorageClass。
- c. 创建新的PVC,详情请参考**6.3.1 创建PVC**。
- 如果由原因2导致PVC处于Pending状态,执行以下步骤。

| Type<br>From | Age  | Message |
|--------------|------|---------|
|              | <br> | <br>_   |

Normal Provisioning 63s (x3 over 64s) csi.huawei.com\_huawei-csi-controller-b59577886-qqzm8\_58533e4a-884c-4c7f-92c3-6e8a7b327515 External provisioner is provisioning volume for claim "default/mypvc"

Warning ProvisioningFailed 63s (x3 over 64s) csi.huawei.com\_huawei-csi-controller-b59577886-qqzm8\_58533e4a-884c-4c7f-92c3-6e8a7b327515 failed to provision volume with StorageClass "mysc": rpc error: code = Internal desc = **failed to select pool**, the capability filter failed, error: failed to select pool, the final filter field: **replication**, parameters map[allocType:thin replication:True size: 1099511627776 volumeType:lun]. please check your storage class

- a. 删除PVC,详细请参考**6.3.5 删除PVC**。
- b. 删除StorageClass, 详细请参考6.1.2 删除StorageClass。
- c. 根据Events信息修改StorageClass.yaml文件。
- d. 创建StorageClass,详细请参考6.1.1 创建StorageClass。
- e. 创建新的PVC, 详情请参考**6.3.1 创建PVC**。
- 如果由原因3导致PVC处于Pending状态,请联系华为工程师处理。

| Events:<br>Type | Reason | Age |  |         |  |
|-----------------|--------|-----|--|---------|--|
| From            |        |     |  | Message |  |
|                 |        |     |  |         |  |
|                 |        |     |  |         |  |

Normal Provisioning 63s (x4 over 68s) csi.huawei.com\_huawei-csi-controller-b59577886-qqzm8\_58533e4a-884c-4c7f-92c3-6e8a7b327515 External provisioner is provisioning volume for claim "default/mypvc"

Warning ProvisioningFailed 62s (x4 over 68s) csi.huawei.com\_huawei-csi-controller-b59577886-qqzm8\_58533e4a-884c-4c7f-92c3-6e8a7b327515 failed to provision volume with StorageClass "mysc": rpc error: code = Internal desc = Create volume map[ALLOCTYPE:1 CAPACITY:20 DESCRIPTION:Created from Kubernetes CSI NAME:pvc-63ebfda5-4cf0-458e-83bd-ecc PARENTID:0] error: 50331651

● 如果由原因4导致PVC处于Pending状态,执行以下步骤。

| Events:<br>Type<br>From | Reason | Age | Message    |
|-------------------------|--------|-----|------------|
|                         |        |     | <br>essage |

Normal Provisioning 63s (x3 over 52s) csi.huawei.com\_huawei-csi-controller-b59577886-qqzm8\_58533e4a-884c-4c7f-92c3-6e8a7b327515 External provisioner is provisioning volume for claim "default/mypvc"

Warning ProvisioningFailed 63s (x3 over 52s) csi.huawei.com\_huawei-csi-controller-b59577886-qqzm8\_58533e4a-884c-4c7f-92c3-6e8a7b327515 failed to provision volume with StorageClass "mysc": rpc error: code = Internal desc = context deadline exceeded (Client.Timeout exceeded while awaiting headers)

- a. 请先等待10分钟,参考本章节再次检查PVC详细信息
- b. 如果还处于Pending状态,请联系华为工程师处理。
- 如果由原因5导致PVC处于Pending状态,请联系华为工程师处理。

#### ----结束

## 10.7 删除 PVC 前,PVC 的状态为 Pending

#### 现象描述

在执行删除PVC前,PVC的状态处于Pending。

#### 根因分析

原因1:由于没有提前创建指定名称的StorageClass,导致Kubernetes在创建PVC时无法找到指定StorageClass名称。

原因2:由于存储池能力和StorageClass能力不匹配,导致huawei-csi选择存储池失败。

原因3:由于存储RESTful接口执行返回具体错误码(例如:50331651),导致huawei-csi在执行创建PVC时失败。

原因4:由于存储在huawei-csi设定的超时时间内没有返回,huawei-csi向Kubernetes返回超时错误。

原因5: 其他原因。

#### 解决措施或规避方法

删除Pending状态下的PVC,需要根据以下不同的原因采取不同的解决措施。

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意master节点。

步骤2 执行以下命令,查看PVC的详细信息。

# kubectl describe pvc mypvc

步骤3 根据PVC详细信息中Events信息,执行相应操作。

如果由原因1导致PVC处于Pending状态,可以执行kubectl delete pvc mypvc命令,删除PVC。

| · / /  |        |     |      |         |  |  |
|--|--------|-----|------|---------|--|--|
| Events:  |        |     |      |         |  |  |
| Type   | Reason | Age | From | Message |  |  |
|  |        |     |      |         |  |  |
| Warning ProvisioningFailed 0s (x15 over 3m24s) persistentvolume-controller |        |     |      |         |  |  |
| storageclass.storage.k8s.io " <i>mvsc</i> " not found                      |        |     |      |         |  |  |

如果由原因2导致PVC处于Pending状态,可以执行kubectl delete pvc mypvc命令,删除PVC。

| Events.  |                |               |                 |                            |                       |  |
|--|----------------|---------------|-----------------|----------------------------|-----------------------|--|
| Type   | Reason         | Age           |                 |                            |                       |  |
| From   |                |               |                 | Message                    |                       |  |
|  |                |               |                 | -                          |                       |  |
|  |                |               |                 |                            |                       |  |
| Norma  | l Provisioning | 63s (x3       | over 64s) csi.l | nuawei.com_huawei-csi-coi  | ntroller-b59577886-   |  |
| qqzm8_58533e4a-884c-4c7f-92c3-6e8a7b327515 External provisioner is provisioning volume for |                |               |                 |                            |                       |  |
| claim "default/mypvc"  |                |               |                 |                            |                       |  |
| Warnin   | g Provisioning | Failed 63s (  | x3 over 64s) cs | i.huawei.com_huawei-csi-c  | controller-b59577886- |  |
| aazm8 5  | 58533e4a-884c  | -4c7f-92c3-6e | 8a7b327515 fa   | iled to provision volume w | ith StorageClass      |  |

qqzm8\_58533e4a-884c-4c7f-92c3-6e8a7b327515 failed to provision volume with StorageClass "mysc": rpc error: code = Internal desc = **failed to select pool**, the capability filter failed, error: failed to select pool, the final filter field: *replication*, parameters map[allocType:thin replication:True size: 1099511627776 volumeType:lun]. please check your storage class

● 如果由原因3导致PVC处于Pending状态,可以执行**kubectl delete pvc** *mypvc*命令,删除PVC。

Events:
Type Reason Age
From Message

Normal Provisioning 63s (x4 over 68s) csi.huawei.com\_huawei-csi-controller-b59577886-qqzm8\_58533e4a-884c-4c7f-92c3-6e8a7b327515 External provisioner is provisioning volume for claim "default/mypvc"

Warning ProvisioningFailed 62s (x4 over 68s) csi.huawei.com\_huawei-csi-controller-b59577886-qqzm8\_58533e4a-884c-4c7f-92c3-6e8a7b327515 failed to provision volume with StorageClass "mysc": rpc error: code = Internal desc = Create volume map[ALLOCTYPE:1 CAPACITY:20 DESCRIPTION:Created from Kubernetes CSI NAME:pvc-63ebfda5-4cf0-458e-83bd-ecc PARENTID:0] error: 50331651

如果由原因4导致PVC处于Pending状态,请联系华为工程师处理。

Type Reason Age
From Message

Normal Provisioning 63s (x3 over 52s) csi.huawei.com\_huawei-csi-controller-b59577886-qqzm8\_58533e4a-884c-4c7f-92c3-6e8a7b327515 External provisioner is provisioning volume for claim "default/mypvc"

Warning ProvisioningFailed 63s (x3 over 52s) csi.huawei.com\_huawei-csi-controller-b59577886-qqzm8\_58533e4a-884c-4c7f-92c3-6e8a7b327515 failed to provision volume with StorageClass "mysc": rpc error: code = Internal desc = context deadline exceeded (Client.Timeout exceeded while awaiting headers)

● 如果由原因5导致PVC处于Pending状态,请联系华为工程师处理。

----结束

## 10.8 创建 Pod 时,Pod 的状态为 ContainerCreating

#### 现象描述

执行完成Pod的创建操作,一段时间后,Pod的状态仍然处于ContainerCreating,查看具体日志信息(详情请参考**10.1 查看日志信息**),报错"Fibre Channel volume device not found"。

### 根因分析

该问题是因为在主机节点有磁盘残留,导致下次创建Pod时,查找磁盘失败。

#### 解决措施或规避方法

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意master节点。

步骤2 执行以下命令,查看Pod所在节点信息。

# kubectl get pod -o wide

NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE

READINESS GATES

mypod 0/1 ContainerCreating 0 51s 10.244.1.224 node1 <none> <none>

步骤3 删除Pod, 详情请参考6.4.2 删除Pod。

**步骤4** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的*node1* 节点。*node1*节点为**步骤**2中查询的节点。

步骤5 移除盘符残留,详情请参考解决措施或规避方法。

----结束

## 10.9 创建 Pod 时,Pod 的状态长时间处于 ContainerCreating 状态

#### 现象描述

创建Pod时,Pod长时间处于ContainerCreating状态,此时查看huawei-csi-node的日志信息(详情请参考**10.1 查看日志信息**),huawei-csi-node的日志中无创建Pod的日志记录,执行**kubectl get volumeattachment**命令后,PV列无该Pod使用的PV名称。在等待较长时间后(超过十分钟),Pod正常创建,Pod状态变为Running状态。

#### 根因分析

该问题是因为Kubernetes的kube-controller-manager组件服务异常导致。

#### 解决措施或规避方法

请联系容器平台侧工程师解决。

# 11 1 附录

- 11.1 OceanStor V3/V5系列和OceanStor Dorado V3系列ALUA特性配置策略样例
- 11.2 OceanStor Dorado 6.x ALUA特性配置策略样例
- 11.3 分布式存储ALUA特性配置策略样例

## 11.1 OceanStor V3/V5 系列和 OceanStor Dorado V3 系列 ALUA 特性配置策略样例

#### 例1.配置文件如下:

对于主机名为"node1",上述ALUA配置段都能用于配置启动器。根据**7.4.2.1 配置OceanStor V3/V5系列和OceanStor Dorado V3系列ALUA**中的配置策略规则,优先级顺序为第2条配置段(HostName为"node1")高于第1条配置段(HostName为"\*")。

#### 例2.配置文件如下:

对于主机名为"node6"的主机,上述ALUA配置段都能用于配置启动器。根据**7.4.2.1 配置OceanStor V3/V5系列和OceanStor Dorado V3系列ALUA**中的配置策略规则,选择第一条ALUA配置段来配置启动器。

#### 例3.配置文件如下:

```
...
"parameters": {..., "ALUA": {
```

根据**7.4.2.1 配置OceanStor V3/V5系列和OceanStor Dorado V3系列ALUA**中的配置 策略规则,对于主机名为"node1"的主机,选择第一条ALUA配置段来配置启动器;对于主机名为"node10"的主机,选择第二条ALUA配置段来配置启动器。^表示匹配字符串的开头,\$表示匹配字符串的结尾。

## 11.2 OceanStor Dorado 6.x ALUA 特性配置策略样例

#### 例1.配置文件如下:

对于主机名为"node1",上述ALUA配置段都能用于配置启动器。根据**7.4.2.2 配置OceanStor Dorado 6.x ALUA**中的配置策略规则,优先级顺序为第2条配置段(HostName为"node1")高于第1条配置段(HostName为"\*")。

#### 例2.配置文件如下:

对于主机名为"node6"的主机,上述ALUA配置段都能用于配置启动器。根据**7.4.2.2 配置OceanStor Dorado 6.x ALUA**中的配置策略规则,选择第一条ALUA配置段来配置启动器。

#### 例3.配置文件如下:

根据**7.4.2.2 配置OceanStor Dorado 6.x ALUA**中的配置策略规则,对于主机名为"node1"的主机,选择第一条ALUA配置段来配置启动器;对于主机名为"node10"的主机,选择第二条ALUA配置段来配置启动器。^表示匹配字符串的开头,\$表示匹配字符串的结尾。

## 11.3 分布式存储 ALUA 特性配置策略样例

#### 例1.配置文件如下:

```
...
"parameters": {..., "ALUA": {
    "*": {"switchoverMode": "Enable_alua", "pathType": "optimal_path"},
    "node1": {"switchoverMode": "Enable_alua", "pathType": "non_optimal_path"}}}
...
```

对于主机名为"node1",上述ALUA配置段都能用于配置启动器。根据**7.4.2.3 配置分布式存储ALUA**中的配置策略规则,优先级顺序为第2条配置段(HostName为"node1")高于第1条配置段(HostName为"\*")。

#### 例2.配置文件如下:

对于主机名为"node6"的主机,上述ALUA配置段都能用于配置启动器。根据**7.4.2.3** 配置分布式存储ALUA中的配置策略规则,选择第一条ALUA配置段来配置启动器。

#### 例3.配置文件如下:

根据**7.4.2.3 配置分布式存储ALUA**中的配置策略规则,对于主机名为"node1"的主机,选择第一条ALUA配置段来配置启动器;对于主机名为"node10"的主机,选择第二条ALUA配置段来配置启动器。^表示匹配字符串的开头,\$表示匹配字符串的结尾。