# eSDK Huawei Storage Kubernetes CSI Plugins V4.2.0

# 用户指南

**文档版本** 03

发布日期 2024-03-13





#### 版权所有 © 华为技术有限公司 2024。 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>

# 安全声明

#### 产品生命周期政策

华为公司对产品生命周期的规定以"产品生命周期终止政策"为准,该政策的详细内容请参见如下网址: https://support.huawei.com/ecolumnsweb/zh/warranty-policy

#### 漏洞处理流程

华为公司对产品漏洞管理的规定以"漏洞处理流程"为准,该流程的详细内容请参见如下网址:

https://www.huawei.com/cn/psirt/vul-response-process

如企业客户须获取漏洞信息,请参见如下网址:

https://securitybulletin.huawei.com/enterprise/cn/security-advisory

#### 华为预置证书权责说明

华为公司对随设备出厂的预置数字证书,发布了"华为设备预置数字证书权责说明",该说明的详细内容请参见如下网址:

https://support.huawei.com/enterprise/zh/bulletins-service/ENEWS2000015766

#### 华为企业业务最终用户许可协议(EULA)

本最终用户许可协议是最终用户(个人、公司或其他任何实体)与华为公司就华为软件的使用所缔结的协议。最终用户对华为软件的使用受本协议约束,该协议的详细内容请参见如下网址: https://e.huawei.com/cn/about/eula

#### 产品资料生命周期策略

华为公司针对随产品版本发布的售后客户资料(产品资料),发布了"产品资料生命周期策略",该策略的详细内容请参见如下网址:

https://support.huawei.com/enterprise/zh/bulletins-website/ENEWS2000017760

# 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 损坏、数据丢失、设备性能降低或其它不可预知的结果。 "须知"不涉及人身伤害。
□ 说明	对正文中重点信息的补充说明。 "说明"不是安全警示信息,不涉及人身、设备及环境伤害信 息。

# 修改记录

文档版本	发布日期	修改说明
03	2024-03-13	第三次正式发布。

文档版本	发布日期	修改说明	
02	2024-01-15	第二次正式发布。	
01	2023-09-30	第一次正式发布。	

# 目录

<u> </u>	iii
1 概述	1
	2
<b>2.1 Kubernetes 及操作系统兼容性</b>	
2.2 Kubernetes 特性矩阵	
2.4 华为分布式存储兼容性	8
3 安装前准备	10
3.1 前提条件	
3.2 下载华为 CSI 软件包	11
3.3 上传华为 CSI 镜像	11
3.3.1 上传镜像到镜像仓库	12
3.3.2 导入镜像到所有节点	12
3.4 检查 CSI 依赖的镜像	13
3.5 检查卷快照依赖组件	14
3.6 检查主机多路径配置	16
3.7 检查华为存储上的账号配置	17
3.8 检查主机依赖软件状态	17
3.9 通信矩阵	17
4 安装部署	18
4.1 使用 Helm 安装华为 CSI	18
4.1.1 准备 values.yaml 文件	18
- 4.1.1.1 images 参数配置说明	18
4.1.1.2 controller 参数配置说明	19
4.1.1.3 node 参数配置说明	21
4.1.1.4 csiDriver 参数配置说明	21
4.1.1.5 其他参数配置说明	25
4.1.2 安装华为 CSI	28
4.1.2.1 Kubernetes、OpenShift、Tanzu 安装华为 CSI	29
4.1.2.2 CCE / CCE Agile 安装华为 CSI	32
- 4.1.2.2.1 制作 Helm 安装包	32
4.1.2.2.2 安装华为 CSI	33

4.2 手动安装华为 CSI	34
5 卸载华为 CSI	37
5.1 Helm 卸载华为 CSI	37
5.1.1 Kubernetes、OpenShift、Tanzu 卸载华为 CSI	37
5.1.2 CCE / CCE Agile 卸载华为 CSI	38
5.1.3 删除 CSI 依赖组件服务	38
5.1.3.1 删除 huawei-csi-host-info 对象	38
5.1.3.2 删除 Webhook 资源	39
5.1.3.3 卸载 Snapshot 依赖组件服务	39
5.2 手动卸载华为 CSI	40
5.2.1 卸载 huawei-csi-node 服务	40
5.2.2 卸载 huawei-csi-controller 服务	40
5.2.3 卸载 csidriver 对象	40
5.2.4 删除 RBAC 权限	41
5.2.5 其它资源卸载	41
6 升级/回退操作	42
6.1 使用 Helm 升级/回退华为 CSI	42
6.1.1 升级华为 CSI	42
6.1.1.1 Kubernetes、OpenShift、Tanzu 升级华为 CSI	43
6.1.2 回退华为 CSI	
6.1.2.1 Kubernetes、OpenShift、Tanzu 回退华为 CSI	44
6.2 手动升级/回退	44
6.2.1 升级华为 CSI	44
6.2.2 回退华为 CSI	45
7 存储后端管理	46
7.1 新增存储后端	46
7.1.1 准备存储后端配置文件	47
7.1.2 创建存储后端	53
7.2 查询存储后端	55
7.3 更新存储后端	55
7.4 删除存储后端	56
7.5 新增证书到存储后端(可选)	56
7.5.1 创建证书到存储后端	57
7.6 查询存储后端证书(可选)	58
7.7 更新存储后端证书(可选)	58
7.8 删除存储后端证书(可选)	59
8 使用华为 CSI	60
8.1 PV/PVC 管理	
8.1.1 创建 PVC	60
8.1.1.1 动态卷供应	61
8.1.1.1.1 配置 StorageClass	61

8.1.1.1.2 配置 PVC	73
8.1.1.2 纳管卷供应	77
8.1.1.2.1 配置 StorageClass	77
8.1.1.2.2 配置 PVC	89
8.1.1.3 静态卷供应	95
8.1.1.3.1 配置 PV	95
8.1.1.3.2 配置 PVC	99
8.1.2 扩容 PVC	102
8.1.3 克隆 PVC	103
8.1.4 从快照创建 PVC	104
8.2 创建 VolumeSnapshot	104
8.2.1 检查卷快照依赖组件信息	105
8.2.2 配置 VolumeSnapshotClass	105
8.2.3 配置 VolumeSnapshot	106
9 高级特性	108
9.1 配置 ALUA 特性	108
9.1.1 通过 Helm 配置 ALUA 特性	108
9.1.1.1 配置华为企业存储后端的 ALUA 参数	108
9.1.1.2 配置分布式存储后端的 ALUA 参数	111
9.2 配置存储拓扑感知	113
9.2.1 通过 Helm 配置存储拓扑感知	114
10 常用操作	117
10.1 日志收集	117
10.2 更新 huawei-csi-controller 服务	118
10.3 更新 huawei-csi-node 服务	121
10.4 修改日志输出模式	123
10.5 开启 ReadWriteOncePod 功能门	124
10.6 配置非 root 用户访问 Kubernetes 集群	126
11 FAQ	127
11.1 如何查看华为 CSI 日志	
11.2 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	129
11.3 启动 huawei-csi-node 失败,提示错误为:"/var/lib/iscsi is not a directory"	129
11.4 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	130
11.5 启动 huawei-csi 服务时,服务启动异常, 状态显示 InvalidImageName	132
11.6 创建 PVC 时, PVC 的状态为 Pending	133
11.7 删除 PVC 前,PVC 的状态为 Pending	
11.8 创建 Pod 时,Pod 的状态为 ContainerCreating	136
11.9 创建 Pod 时,Pod 的状态长时间处于 ContainerCreating 状态	137
	137
11.11 创建 Pod 失败,日志显示执行 mount 命令失败	138
11.12 创建 Pod 失败或重启 kubelet 后,日志显示挂载点已存在	139

707 7410	
11.13 创建 Pod 失败,Events 日志显示"publishInfo doesn't exist"	139
11.14 如何下载容器镜像到本地	140
11.15 如何获取 CSI 版本信息	140
11.16 对接 Tanzu Kubernetes 集群常见问题及解决方法	
11.16.1 未创建 PSP 权限导致 Pod 无法创建	
11.16.2 修改主机挂载点	
11.16.3 修改 livenessprobe 容器的默认端口	143
11.17 使用 Tanzu Kubernetes 集群时常见问题及解决方法	143
11.17.1 创建临时卷失败	143
11.17.1 创建临时卷失败 11.18 通用临时卷扩容失败	144
11.19 PVC 扩容的目标容量超过存储池容量导致扩容失败	144
11.20 使用 oceanctl 工具管理后端时调用 webhook 失败	145
11.21 存储侧更新密码后账户被锁定	145
12 附录	147
12.1 OceanStor V3/V5 系列和 OceanStor Dorado V3 系列 ALUA 特性配置策略样例	147
12.2 OceanStor Dorado 6.x ALUA 特性配置策略样例	148
12.3 分布式存储 ALUA 特性配置策略样例	149
12.4 安装 Helm 3	150
12.5 配置自定义权限	150

■ 概述

容器存储接口(Container Storage Interface),简称 CSI,是一种行业标准,用于将 块和文件存储系统暴露给 Kubernetes 等容器编排系统 (CO) 上的容器工作负载。华为 CSI插件用于和华为企业存储和分布式存储产品进行通信,为Kubernetes的容器工作负载提供存储服务。是华为企业存储和分布式存储在Kubernetes环境中使用的必须插件。

Kubernetes通过其官方维护的一系列sidecar组件负责注册监听Kubernetes对象资源,并在需要的时候发起对CSI Driver调用,华为CSI Driver将sidecar发起的调用在华为存储上实施,如创建一个**持久卷(Persistent Volume,PV)**的操作被实施为在华为存储上创建一个LUN/文件系统。Kubernetes、华为CSI以及华为存储的整体结构如下图所示:

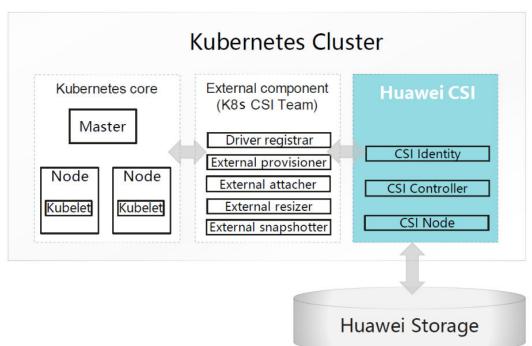


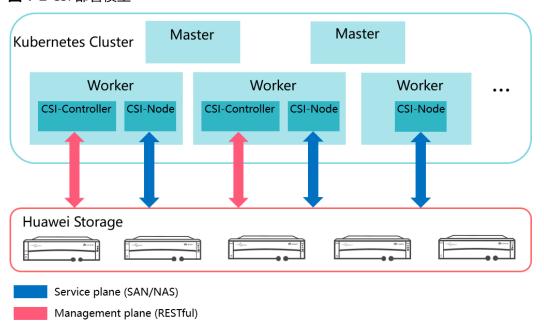
图 1-1 CSI 整体架构

华为CSI主要有两大组件,分别为CSI Controller和CSI Node:

- CSI Controller: 以Deployment方式运行的一个或多个Pod,主要负责与华为存储交互,使用RESTful方式进行通信,因此运行CSI Controller组件的节点需要连通存储的管理面网络。
- CSI Node: 以DaemonSet方式运行在Kubernetes工作节点上的Pod,用于在工作 节点上对华为存储提供的LUN/文件系统资源进行挂载和卸载等操作,因此运行 CSI Node组件的节点需要连通存储的业务面网络。

华为CSI的部署模型如下所示:

图 1-2 CSI 部署模型



本文档主要介绍华为CSI V4.2.0插件的安装部署和使用。

本章节会详细说明华为CSI插件支持的容器管理平台、操作系统、多路径软件以及CSI 插件配合华为存储所提供的特性和功能。

- 2.1 Kubernetes及操作系统兼容性
- 2.2 Kubernetes特性矩阵
- 2.3 华为企业存储兼容性
- 2.4 华为分布式存储兼容性

# 2.1 Kubernetes 及操作系统兼容性

华为CSI插件支持如下容器管理平台:

表 2-1 支持的容器管理平台

容器管理平台	版本
Kubernetes	1.16, 1.18~1.27
Red Hat OpenShift Container Platform	4.6 EUS, 4.7, 4.8, 4.9, 4.10, 4.11, 4.12, 4.13
Tanzu Kubernetes	TKGI 1.14.1, TKGI 1.15, TKGI 1.16, TKGI 1.17
CCE Agile	22.3.2
CCE	22.9.5

#### 须知

- 华为CSI对接Red Hat OpenShift的相关操作请参见《eSDK Enterprise Storage Plugins 用户指南(Kubernetes CSI for Red Hat OpenShift)》。
- 华为CSI对接Tanzu Kubernetes仅支持NAS场景,相关FAQ请参见11.16 对接Tanzu Kubernetes集群常见问题及解决方法。

华为CSI插件支持的操作系统以及多路径信息如下表所示。

表 2-2 支持的主机操作系统及多路径软件版本

操作系统名称	操作系统版本	原生DM- Multipath版本	华为UltraPath版本
CentOS x86_64	7.6, 7.7, 7.9	随OS自带,支 持FC/iSCSI	UltraPath 31.1.0,支持FC/iSCSI
CentOS x86_64	8.2, 8.4	随OS自带,支 持FC/iSCSI	UltraPath 31.1.0,支持FC/iSCSI UltraPath-NVMe 31.1.RC8,支 持NVMe over RoCE/NVMe over FC
CentOS ARM	7.6	随OS自带,支 持FC/iSCSI	不支持
Rocky Linux x86_64	8.6	随OS自带,支 持FC/iSCSI	不支持
SUSE 15 x86_64	SP2, SP3	随OS自带,支持FC/iSCSI	UltraPath 31.1.0,支持FC/iSCSI UltraPath-NVMe 31.1.RC8,支 持NVMe over RoCE/NVMe over FC
Red Hat CoreOS x86_64	4.6, 4.7, 4.8, 4.9, 4.10, 4.11, 4.12, 4.13	随OS自带,支 持FC/iSCSI	不支持
Ubuntu x86_64	18.04, 20.04, 22.04	随OS自带,支 持FC/iSCSI	不支持
Ubuntu ARM	22.04	随OS自带,支 持FC/iSCSI	不支持
Kylin x86_64	V10 SP1, V10 SP2, 7.6	随OS自带,支 持FC/iSCSI	不支持
Kylin ARM	V10 SP1, V10 SP2	随OS自带,支 持FC/iSCSI	不支持
Debian x86_64	9, 11	随OS自带,支 持FC/iSCSI	不支持
EulerOS x86_64	V2R9, V2R10, V2R11	随OS自带,支 持FC/iSCSI	不支持
EulerOS ARM	V2R10	随OS自带,支 持FC/iSCSI	不支持

#### 山 说明

因DM-Multipath在0.7版本存在执行multipathd show maps时可能无法回显所有的虚拟设备,因此建议使用0.8及以上版本。

DM-Multipath版本可以通过以下途径查询:

● 如果使用的是rpm包,执行: rpm -qa | grep device-mapper

• 如果使用的是deb包,执行: dpkg -l | grep multipath

# 2.2 Kubernetes 特性矩阵

本章节说明华为CSI在不同Kubernetes版本下支持的特性。

表 2-3 Kubernetes 版本与支持的特性

特性	V1.16	V1.18	V1.19	V1.20	V1.21+
Static Provisioning	√	√	√	√	√
Dynamic Provisioning	√	√	√	√	√
Manage Provisioning <sup>1</sup>	√	√	√	√	√
Expand Persistent Volume	√	√	√	√	√
Create VolumeSnapshot	х	√	√	√	√
Restore VolumeSnapshot	х	√	√	√	√
Delete VolumeSnapshot	x	√	√	√	√
Clone Persistent Volume	x	√	√	√	<b>√</b>
Raw Block Volume	√	√	√	√	<b>→</b>
Topology	√	√	√	√	√
Generic Ephemeral Volumes	×	×	×	×	√

• 注释1 Manage Provisioning是华为CSI自定义的纳管卷特性,该特性支持将已有存储资源纳管至Kubernetes。不允许将一个存储资源纳管多次。

# 2.3 华为企业存储兼容性

华为CSI插件兼容华为OceanStor系列的全闪存存储和混合闪存存储,具体支持的存储版本如下表所示:

表 2-4 支持的华为企业存储

存储产品	版本		
OceanStor V3	V300R006		
OceanStor V5	V500R007, V500R007 Kunpeng		
OceanStor Dorado V3	V300R002		
OceanStor V6	6.1.3, 6.1.5, 6.1.6		
OceanStor Dorado V6	6.1.0, 6.1.2, 6.1.3, 6.1.5, 6.1.6		

华为CSI插件针对华为企业存储支持如下特性。

表 2-5 华为企业存储支持的特性及约束

特性	OceanSt or V3	OceanStor V5	OceanStor Dorado V3	OceanStor V6	OceanStor Dorado V6
Static Provisionin g	SAN: FC/ iSCSI <sup>2</sup> NAS:	SAN: FC/ iSCSI <sup>2</sup> NAS: NFS	SAN: FC/ iSCSI <sup>2</sup>	SAN: FC/ iSCSI/NVMe over RoCE/	SAN: FC/ iSCSI/NVMe over RoCE/
Dynamic Provisionin	NFS 3	3		NVMe over FC <sup>3</sup>	NVMe over FC <sup>3</sup>
g				NAS: NFS 3/4.0/4.1	NAS: NFS 3/4.0/4.1 <sup>4</sup>
Manage Provisionin g <sup>1</sup>				3, 115, 111	3/4.0/4.1
Expand Persistent Volume	支持使用Dynamic Provisioning,Manage Provisioning方式创建的卷				
Create VolumeSna pshot	支持使用Dynamic Provisioning,Manage Provisioning方式创建的非双活卷				
Delete VolumeSna pshot	支持	支持	支持	支持	支持

特性	OceanSt or V3	OceanStor V5	OceanStor Dorado V3	OceanStor V6	OceanStor Dorado V6
Restore VolumeSna pshot	支持	支持	支持	SAN: 支持 NAS: 仅 6.1.5及其之 后版本支持	SAN: 支持 NAS: 仅 6.1.5及其之 后版本支持
Clone Persistent Volume		namic Provisioning, ovisioning方式创建的非双		SAN: 支持使 Provisioning, Provisioning, 双活卷 NAS: 仅6.1.5 支持使用Dyna Provisioning, Provisioning, 双活卷	Manage 古式创建的非 及其之后版本 amic Manage
Raw Block Volume	仅支持 SAN类型 的卷	仅支持SAN 类型的卷	仅支持SAN 类型的卷	仅支持SAN 类型的卷	仅支持SAN 类型的卷
Topology	支持	支持	支持	支持	支持
Generic Ephemeral Volumes	支持	支持	支持	支持	支持
Access Mode	RWO/ROX/RWOP: 所有类型卷均支持,RWOP需Kubernetes 1.22版本以上支持。 RWX: 仅Raw Block卷和NFS类型的卷支持。				ernetes 1.22版
QoS	支持 <sup>5</sup>	支持 <sup>5</sup>	支持	支持	支持
应用类型	不涉及	不涉及	不涉及	支持	支持
卷双活	不支持	不支持	不涉及	仅支持NAS类	型的卷
存储多租户	仅支持NAS	类型的卷	不涉及	仅支持NAS类	型的卷6

- 注释1 Manage Provisioning是华为CSI自定义的纳管卷特性,该特性支持将已有存储资源纳管至Kubernetes。不允许将一个存储资源纳管多次。
- 注释2 若用户的容器平台部署在虚拟化环境中,则仅支持iSCSI组网。
- 注释3 使用NVMe over RoCE或NVMe over FC时,worker节点nvme-cli工具版本不低于1.9,查询命令为:nvme version。
- 注释4 仅OceanStor Dorado V6 6.1.0及以后版本支持NFS。仅OceanStor Dorado V6 6.1.3及以后版本NFS 4.1。
- 注释5 仅系统用户支持配置QoS。
- 注释6 仅OceanStor Dorado V6 6.1.3及以后版本支持多租户。

华为CSI插件针对华为企业存储Dtree特性支持如下表所示。

表 2-6 Dtree 支持的特性

特性	支持情况
Static Provisioning	√
Dynamic Provisioning	√
Expand Persistent Volume	√
Access Mode	√(RWX/RWO/ROX/RWOP: RWOP需 Kubernetes 1.22版本以上支持。)
多租户	√
Create VolumeSnapshot	X
Delete VolumeSnapshot	X
Restore VolumeSnapshot	X
Clone Persistent Volume	X
QoS	Х
卷双活	X
应用类型	Х

#### 表 2-7 Dtree 支持的华为存储版本

存储产品	版本		
OceanStor Dorado V6	6.1.0, 6.1.2, 6.1.3, 6.1.5, 6.1.6		
OceanStor Dorado V3	不支持		

# 2.4 华为分布式存储兼容性

华为CSI插件兼容华为OceanStor系列的分布式存储系统,具体支持的存储版本如下表所示:

表 2-8 支持的华为分布式存储

存储产品	版本		
FusionStorage Block	8.0.0, 8.0.1		
OceanStor Pacific系列	8.1.0, 8.1.1, 8.1.2, 8.1.3, 8.1.5		

华为CSI插件针对华为分布式存储支持如下特性。

表 2-9 华为分布式存储支持的特性及约束

特性	FusionStorage	FusionStorage Block	OceanStor Pacific 系列	
Static Provisioning	SAN: iSCSI/	SAN: iSCSI/SCSI	SAN: iSCSI/SCSI	
Dynamic Provisioning	SCSI		NAS: DPC <sup>2</sup> /NFS 3/4.1 <sup>3</sup>	
Manage Provisioning <sup>1</sup>				
Expand Persistent Volume	支持使用Dynamic 创建的卷	Provisioning, Man	age Provisioning方式	
Create VolumeSnapshot	支持使用Dynamic 创建的SAN类型卷		age Provisioning方式	
Delete VolumeSnapshot	支持	支持	仅支持SAN类型的卷 快照	
Restore VolumeSnapshot	支持	支持	仅支持SAN类型的卷 快照	
Clone Persistent Volume	支持使用Dynamic Provisioning,Manage Provisioning方式 创建的SAN类型卷			
Raw Block Volume	仅支持SAN类型 的卷	仅支持SAN类型的 卷	仅支持SAN类型的卷	
Topology	支持	支持	支持	
Generic Ephemeral Volumes	支持	支持	支持	
Access Mode	RWO/ROX/RWOP: 所有类型卷均支持,RWOP在Kubernetes 1.22及以上版本支持。			
	RWX: 仅Raw Blo	ck卷和NFS类型的卷列	支持。 	
QoS	支持	支持	支持	
软硬配额	不支持	不支持		
存储多租户	不支持	不支持	仅支持NAS类型的卷	

- 注释1 Manage Provisioning是华为CSI自定义的纳管卷特性,该特性支持将已有存储资源纳管至Kubernetes。不允许将一个存储资源纳管多次。
- 注释2 仅OceanStor Pacific系列 8.1.2及以后版本支持DPC。华为CSI支持的操作系统对DPC的支持请参考对应产品版本兼容性文档。
- 注释3 仅OceanStor Pacific系列 8.1.2及以后版本支持NFS 4.1。

# **3** 安装前准备

本章节将对安装前的准备工作进行详细说明。

- 3.1 前提条件
- 3.2 下载华为CSI软件包
- 3.3 上传华为CSI镜像
- 3.4 检查CSI依赖的镜像
- 3.5 检查卷快照依赖组件
- 3.6 检查主机多路径配置
- 3.7 检查华为存储上的账号配置
- 3.8 检查主机依赖软件状态
- 3.9 通信矩阵

# 3.1 前提条件

在进行本章节所说明的操作前,请确保如下条件已经具备:

- 容器管理平台已部署完成并正常运行,且兼容性满足2.1 Kubernetes及操作系统 兼容性章节的要求。
- (企业存储必选)已完成对接华为企业存储初始化配置,包括存储池划分、端口配置等。且存储产品的版本满足2.3 华为企业存储兼容性章节的要求。
- (分布式存储必选)已完成对接华为分布式存储初始化配置,包括存储池划分、端口配置等。且存储产品的版本满足**2.4 华为分布式存储兼容性**章节的要求。
- 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,详细操作请参考表2-2。
- 完成华为存储和容器平台主机连通性配置,例如运行huawei-csi-controller的 worker节点与待接入的存储设备的管理IP地址通信正常,运行huawei-csi-node的 worker节点与待接入的存储设备的业务IP地址通信正常,iSCSI场景下允许使用 ping命令进行连通性校验。
- 容器集群的所有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。

请确保操作系统的语言是英文。

# 3.2 下载华为 CSI 软件包

本章节详细说明了下载方法以及软件包组件结构。

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 根据存储类型和CPU架构,下载对应的4.2.0版本软件包。

#### □□ 说明

软件包命名规范:存储类型+插件名称(Kubernetes\_CSI\_Plugin)+版本号+CPU架构

例如:使用分布式存储对接x86架构的主机时,需要下载的软件包名为eSDK\_Huawei\_Storage\_Kubernetes\_CSI\_Plugin\_V4.2.0\_X86\_64.zip

步骤3 将下载的软件包解压。软件包组件结构如下表所示。

表 3-1 软件包组件描述

组件	组件描述		
image/huawei-csi-v4.2.0- arch.tar	huawei-csi-driver镜像,"arch"为x86或arm。		
image/storage-backend- controller-v4.2.0- <i>arch</i> .tar	后端管理控制器镜像,"arch"为x86或arm。		
image/storage-backend- sidecar-v4.2.0- <i>arch</i> .tar	后端管理sidecar镜像,"arch"为x86或arm。		
bin/	华为提供的镜像使用的二进制文件。		
bin/oceanctl	华为提供的命令行工具,可用于管理存储后端。		
helm/	Helm工程,用于部署华为CSI。		
manual/	用于手动安装部署华为CSI。		
examples/	CSI使用过程中的yaml示例文件。		
examples/backend	创建存储后端的yaml示例文件。		
tools/	无镜像仓库时,用于上传镜像的脚本。		

#### ----结束

# 3.3 上传华为 CSI 镜像

华为提供huawei-csi镜像供用户使用,镜像文件获取参见3.2 下载华为CSI软件包。

为了后续在容器管理平台中可以使用CSI镜像,需要按照以下方式中的一种提前将CSI 镜像导入到集群中:

- 使用Docker工具,将CSI镜像上传至镜像仓库(推荐)
- 使用镜像上传脚本,将CSI镜像导入到所有需要部署华为CSI的节点。

## 3.3.1 上传镜像到镜像仓库

安装华为CSI依赖如下三个华为提供的镜像文件,请按照说明依次导入并上传下列镜像文件,镜像文件获取请参考3.2 下载华为CSI软件包。

- huawei-csi-v4.2.0-arch.tar
- storage-backend-controller-v4.2.0-arch.tar
- storage-backend-sidecar-v4.2.0-*arch*.tar

#### 前提条件

已准备一台已安装Docker的Linux主机,且该主机支持访问镜像仓库。

#### 操作步骤

步骤1 执行docker load -i huawei-csi.tar命令,将CSI镜像导入当前节点。

# docker load -i huawei-csi.tar Loaded image: huawei-csi:4.2.0

**步骤2** 执行docker tag huawei-csi:4.2.0 repo.huawei.com/huawei-csi:4.2.0命令,添加镜像仓库地址到镜像标签。其中repo.huawei.com表示镜像仓库的地址。

# docker tag huawei-csi:4.2.0 repo.huawei.com/huawei-csi:4.2.0

**步骤3** 执行docker push repo.huawei.com/huawei-csi:4.2.0命令,将CSI镜像上传到镜像仓库。其中repo.huawei.com表示镜像仓库的地址。

# docker push repo.huawei.com/huawei-csi:4.2.0

#### ----结束

#### 须知

- 也可以使用containerd来进行镜像的导入和上传。
- CCE / CCE Agile平台请参考该平台用户手册完成镜像导入和上传。

# 3.3.2 导入镜像到所有节点

若镜像已上传至镜像仓库,则跳过本章节。

#### 前提条件

- 镜像所在的主机与需要导入镜像的所有主机之间能进行SSH通信。
- 镜像所在的主机上已经安装expect、sshpass和scp软件包。

#### 操作步骤

步骤1 执行vi worker-list.txt命令,创建并打开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。 # ./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。
  # ./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是失败的。 List of nodes to which the image fails to be imported: 192.168.128.16 192.168.128.17

#### ----结束

# 3.4 检查 CSI 依赖的镜像

华为CSI安装过程中需要依赖下表中的镜像,若集群中的所有worker节点已连接互联网 且能够在线拉取镜像,则可跳过本章节。若集群中的节点无法连接互联网,则请根据 使用的Kubernetes版本,下载对应的镜像文件并上传到镜像仓库中或者导入 Kubernetes集群的所有worker节点中。

huawei-csi-controller服务依赖的sidecar镜像: livenessprobe、csi-provisioner、csi-attacher、csi-resizer、csi-snapshotter、snapshot-controller。

huawei-csi-node服务依赖的sidecar镜像: livenessprobe、csi-node-driver-registrar和 huawei-csi-driver。

关于每个镜像的功能和详情,请参考下表。

表 3-2 Huawei CSI 依赖的镜像

容器名称	容器镜像	K8s 版本 要求	功能描述
livenesspr obe	k8s.gcr.io/sig-storage/ livenessprobe:v2.5.0	v1.13 +	用于监控CSI的健康状态,并上报给 Kubernetes,使Kubernetes能够自 动检测CSI程序的问题并重启Pod尝 试修改该问题。
csi-resizer	k8s.gcr.io/sig-storage/ csi-resizer:v1.4.0	v1.13 +	在扩容PVC时,调用CSI给PVC提供 更多的存储容量空间。

容器名称	容器镜像	K8s 版本 要求	功能描述
csi-node- driver- registrar	k8s.gcr.io/sig-storage/ csi-node-driver- registrar:v2.3.0	v1.13 +	用于获取CSI信息,并通过kubelet的 插件注册机制将节点注册到kubelet 中,从而Kubernetes能够感知该节 点与华为存储的对接。
csi- snapshott er	k8s.gcr.io/sig-storage/ csi-snapshotter:v4.2.1	v1.17 +	在创建/删除VolumeSnapshot时, 调用CSI在存储侧完成快照的创建和 删除。
snapshot- controller	k8s.gcr.io/sig-storage/ snapshot- controller:v4.2.1	v1.17 +	在创建/删除VolumeSnapshot时, 监听Kubernetes API中关于 VolumeSnapshot和 VolumeSnapshotContent的对象, 并触发csi-snapshotter在存储上完成 快照的创建。
csi- provisione	k8s.gcr.io/sig-storage/ csi-provisioner:v3.0.0	v1.17 +	● 在创建PVC时,调用CSI Controller服务在存储上创建
r	quay.io/k8scsi/csi- provisioner:v1.4.0	v1.16 .x	LUN/文件系统作为PV,并将PV 绑定至PVC。 • 在删除PVC时,调用CSI Controller服务解除PV至PVC的绑 定,然后在存储上删除该PV对应 的LUN/文件系统
csi- attacher	k8s.gcr.io/sig-storage/ csi-attacher:v3.4.0	v1.17 +	在创建/删除Pod时,调用CSI Controller服务执行Publish/
	quay.io/k8scsi/csi- attacher:v1.2.1	v.1.16 .x	Unpublish Volume操作。

#### 山 说明

如何下载容器镜像到本地,请参考11.14 如何下载容器镜像到本地。

# 3.5 检查卷快照依赖组件

本章节介绍如何检查集群中卷快照依赖组件情况。

## <u>/</u>注意

由于Kubernetes v1.17.0之前不支持快照功能,如果部署快照CRD可能导致集群出现问题,因此在低于Kubernetes v1.17.0版本上部署华为CSI,请务必按照**Kubernetes低于v1.17.0**章节检查。

#### Kubernetes 低于 v1.17.0

Kubernetes低于v1.17.0时,部署快照时将会导致集群出现问题,请按照以下步骤删除快照CRD安装文件。

步骤1 执行以下命令查看Kubernetes版本。如下示例中,Kubernetes版本为v1.16.0。

# kubectl get node
NAME STATUS ROLES AGE VERSION
test-master Ready master 311d v1.16.0
test-node Ready <none> 311d v1.16.0

步骤2 进入/helm/esdk/crds/snapshot-crds目录,执行以下命令,删除快照CRD安装文件。 组件包路径请参考<mark>表3-1</mark>。

# rm -rf ./huawei-csi-snapshot-crd-v1.yaml

----结束

#### Kubernetes 高于或等于 v1.17.0

如果您需要在容器环境中使用卷快照以及卷快照关联的特性,请通过如下步骤检查您的环境是否部署了卷快照依赖组件以及卷快照api-versions信息。

步骤1 执行命令,查看snapshot相关资源服务安装详情。

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

- 如果回显如上所示,则说明快照依赖组件服务已安装,请进入**步骤2**,检查apiversions信息。
- 如果回显缺少如上所示的任意一项服务,请进入步骤3,安装Snapshot依赖组件服务。

#### 步骤2 执行命令,查看卷快照api-versions信息。

# kubectl api-versions | grep "snapshot.storage.k8s.io" snapshot.storage.k8s.io/v1 snapshot.storage.k8s.io/v1beta1

- 如果回显如上所示,则说明快照依赖组件服务支持v1和v1beta1版本,跳过本章 节。
- 如果回显缺少如上所示的任意一项服务,请继续执行后续步骤进行安装。

**步骤3** 进入/helm/esdk/crds/snapshot-crds目录,执行以下命令,安装Snapshot依赖组件服务。组件包路径请参考**表3-1**。

# kubectl apply -f huawei-csi-snapshot-crd-v1.yaml --validate=false

Warning: resource volumesnapshots/mysnapshot is missing the kubectl.kubernetes.io/last-applied-configuration annotation which is required by kubectl apply. kubectl apply should only be used on resources created declaratively by either kubectl create --save-config or kubectl apply. The missing annotation will be patched automatically.

volumesnapshot.snapshot.storage.k8s.io/mysnapshot configured

#### ----结束

完成安装后,可以使用步骤1中所示的命令检查snapshot相关资源服务安装详情。

# 3.6 检查主机多路径配置

当您计划在容器环境中使用FC/iSCSI/NVMe over RoCE/NVMe over FC协议对华为存储进行访问时,推荐您使用主机多路径软件增强主机和存储的链路冗余和性能。如果您不准备使用多路径软件,请跳过本章节。

华为CSI软件支持对接的操作系统和多路径软件请参考表2-2。

#### □ 说明

- 如果您准备使用FC/iSCSI协议对接华为存储时,推荐使用操作系统自带的原生DM-Multipath。
- 如果您准备使用NVMe over RoCE/NVMe over FC协议对接华为存储时,推荐使用华为自研的UltraPath-NVMe。
- 如果您使用SCSI协议对接华为存储时,请关闭操作系统自带的DM-Multipath。

#### 前提条件

主机多路径软件已经被正确的安装在主机上。

- 如果您使用的是操作系统自带的原生DM-Multipath,请咨询您的主机或操作系统 提供商获取安装所需的资料和软件包。
- 如果您使用的是华为自研的UltraPath或者UltraPath-NVMe,请联系华为工程师获取UltraPath或者UltraPath-NVMe的资料和软件包。软件包版本请参考表2-2。

#### 检查步骤

- 步骤1 如果您使用iSCSI/FC协议对接华为企业存储,请参考OceanStor Dorado 6.x & OceanStor 6.x在Red Hat下的主机连通性指南文档的"配置多路径"->"配置多路径(非双活场景)"章节,对主机多路径进行配置和检查。
- 步骤2 如果您使用NVMe over RoCE/NVMe over FC协议对接华为企业存储,请参考 OceanStor Dorado 6.x & OceanStor 6.x在Red Hat下的主机连通性指南文档的"配置多路径"->"配置多路径(非双活场景)"->"UltraPath多路径"章节,对主机 多路径进行配置和检查。
- 步骤3 如果您使用iSCSI协议对接华为分布式存储,请参考《FusionStorage 8.0.1 块存储基础业务配置指南》中的"应用服务器配置多路径"章节,对主机多路径进行配置和检查。
- **步骤4** 如果您使用了操作系统原生多路径时,需要检查/etc/multipath.conf文件,检查文件是否存在如下配置:

```
defaults {
    user_friendly_names yes
    find_multipaths no
}
```

如果配置不存在,请在/etc/multipath.conf文件开始处增加该配置项。

#### □ 说明

user\_friendly\_names 和find\_multipaths 的参数作用请参考: dm\_multipath/config file defaults

----结束

# 3.7 检查华为存储上的账号配置

当华为存储接入容器平台后,华为CSI需要在华为存储上根据业务要求,管理存储资源,如创建卷、映射卷等操作。此时,华为CSI需要使用华为存储上已经创建的账号和华为存储进行通信。针对不同存储设备所需要的账号信息如下表所示。

表 3-	3 存储对接	CSLE	计使用的	小叶石面外
4X J-		COLL	114.4	ソバレーマハ

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 Dorado V3	系统用户	管理员	管理员	本地用户
OceanStor 6.1.3, 6.1.5, 6.1.6	系统用户	管理员/自定 义角色 <sup>1</sup>	N/A	本地用户
OceanStor Dorado 6.1.0, 6.1.2, 6.1.3,	系统用户	管理员/自定 义角色 <sup>1</sup>	N/A	本地用户
6.1.5, 6.1.6	租户用户	租户管理员	N/A	本地用户
OceanStor Pacific系列	系统用户	管理员	N/A	本地用户

注释1 使用自定义角色,需要给角色配置权限,最小权限请参考12.5 配置自定义权限章节配置。

# 3.8 检查主机依赖软件状态

本章节介绍如何检查集群中工作节点上主机依赖软件状态是否正常。本例中主机操作系统为CentOS 7.9 x86\_64。

- 检查iSCSI客户端状态。# systemctl status iscsi iscsid
- 检查NFS客户端状态。# systemctl status rpcbind
- 检查DM-Multipath多路径软件状态。 # systemctl status multipathd.socket multipathd
- 检查UltraPath多路径软件状态。# systemctl status nxup
- 检查UltraPath-NVMe多路径软件状态。
   # systemctl status upudev upService\_plus

# 3.9 通信矩阵

请联系华为工程师,在华为support网站中获取,企业存储请参考eSDK Enterprise Storage Plugins 2.5.0,分布式存储请参考eSDK Cloud Storage Plugins 2.5.RC4。

**4** 安装部署

本章节介绍如何在Kubernetes、OpenShift、Tanzu、CCE、CCE Agile平台安装华为CSI。

- 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

# 4.1.1 准备 values.yaml 文件

在使用Helm安装CSI时,需要您根据部署时需要使用的特性准备Helm工程的values.yaml文件。华为CSI已经在软件包的helm/esdk目录下提供了values.yaml模板文件。

本章节将详细说明values.yaml中的配置项以及典型场景下的后端配置示例。

# 4.1.1.1 images 参数配置说明

values.yaml中的images配置项主要配置华为CSI运行时依赖的组件镜像信息。需要配置的参数如下:

表 4-1 images 配置项说明

参数	描述	必选参数	默认值
images.huaweiCSI Service	huawei-csi镜像。	是	huawei-csi:4.2.0
images.storageBa ckendSidecar	华为后端管理sidecar 镜像。	是	storage-backend-sidecar:4.2.0
images.storageBa ckendController	华为后端管理控制器镜 像。	是	storage-backend- controller:4.2.0

参数	描述	必选参数	默认值
images.sidecar.live	<b>livenessprobe</b> sidecar	是	k8s.gcr.io/sig-storage/
nessProbe	镜像。		livenessprobe:v2.5.0
images.sidecar.pro	<b>csi-provisioner</b>	畏	k8s.gcr.io/sig-storage/csi-
visioner	sidecar镜像。		provisioner:v3.0.0
images.sidecar.att	<b>csi-attacher</b> sidecar	是	k8s.gcr.io/sig-storage/csi-
acher	镜像。		attacher:v3.4.0
images.sidecar.resi	<b>csi-resizer</b> sidecar镜	是	k8s.gcr.io/sig-storage/csi-
zer	像。		resizer:v1.4.0
images.sidecar.sna	<b>csi-snapshotter</b>	是	k8s.gcr.io/sig-storage/csi-
pshotter	sidecar镜像。		snapshotter:v4.2.1
images.sidecar.sna pshotController	<b>snapshot-controller</b> sidecar镜像。	是	k8s.gcr.io/sig-storage/ snapshot-controller:v4.2.1
images.sidecar.reg istrar	<b>csi-node-driver-</b> <b>registrar</b> sidecar镜 像。	是	k8s.gcr.io/sig-storage/csi- node-driver-registrar:v2.3.0

#### 须知

- huaweiCSIService、storageBackendSidecar、storageBackendController参数的值,请参考3.3 上传华为CSI镜像章节的说明,使用您的最终生成镜像的名称和版本。
- 其他sidecar镜像参数,请参考**3.4 检查CSI依赖的镜像**章节的说明,使用您的最终上 传的镜像的名称和版本。

#### 4.1.1.2 controller 参数配置说明

controller配置项用于配置huawei-csi-controller组件的相关配置。

#### 表 4-2 controller 配置项说明

参数	描述	必选 参数	默认值	备注
controller.controll erCount	huawei-csi- controller组件的副 本数	是	1	-

参数	描述	必选 参数	默 认 值	备注
controller.volume NamePrefix	PV名称的前缀,默 认值为pvc,即创 建的PV名称为: pvc- <uuid>。前缀 必须满足DNS 子域 名的命名规则,且 PV名称总长度不 超过253个字符。</uuid>	否	pvc	对应的provisioner参数名称为:volume-name-prefix 详细配置请参考配置PV名称前缀。  • 对接后端是OceanStor V3/V5 SAN时,建议
controller.webhoo kPort	webhook服务使用 的端口。	是	443 3	如果存在端口冲突可修改 为其他不冲突的端口。
controller.snapsho t.enabled	是否开启快照特 性。	是	tru e	如果要使用快照相关功能,请开启该特性。 要求Kubernetes版本高于v1.17。
controller.resizer.e nabled	是否开启扩容特 性。	是	tru e	要求Kubernetes版本高于 v1.16。
controller.nodeSel ector	huawei-csi- controller的节点选 择器。配置后 huawei-csi- controller仅会调度 到存在该标签的节 点上。	否	-	节点选择器的详细说明请参考: 将 Pod 分配给节点
controller.toleratio ns	huawei-csi- controller的污点容 忍。配置后 huawei-csi- controller能够容忍 节点上存在该污 点。	否	-	污点和容忍度的详细说明 请参考: <b>污点和容忍度</b>

#### 山 说明

当controller.snapshot.enabled参数配置为true时,执行**helm install**命令将自动读取"helm/crd"目录下的卷快照CRD,并将安装快照CRD资源。

#### 4.1.1.3 node 参数配置说明

node配置项用于配置huawei-csi-node组件的相关配置。

表 4-3 node 配置项说明

参数	描述	必选 参数	默认值	备注
node.maxVolum esPerNode	节点可使用的华 为CSI发放卷的最 大数量。不定义 或者配置为0时则 认为不限制。 如果创建Pod 时,指定 nodeName,则 会忽略该配置。	否	100	详细说明请参 考: Volume Limits
node.nodeSelect or	huawei-csi-node 的节点选择器。 配置后huawei- csi-node仅会调 度到存在该标签 的节点上。	否	-	节点选择器的详细说明请参考: 将 Pod 分配给节点
node.tolerations	huawei-csi-node 的污点容忍。配 置后huawei-csi- node能够容忍节 点上存在该污 点。	否	- key: "node.kubernetes.io/ memory-pressure" operator: "Exists" effect: "NoExecute" - key: "node.kubernetes.io/ disk-pressure" operator: "Exists" effect: "NoExecute" - key: "node.kubernetes.io/ network- unavailable" operator: "Exists" effect: "NoExecute"	污点和容忍度的 详细说明请参 考: 污点和容忍 度

#### 4.1.1.4 csiDriver 参数配置说明

csiDriver配置项包括了华为CSI运行时的基本配置,如华为驱动名称、多路径类型等配置信息。

表 4-4 csiDriver 配置项说明

参数	描述	必选参数	默认值	备注
csiDriver.driv erName	注册的驱动名称。	是	csi.huawei.c om	<ul> <li>直接使用默认值。</li> <li>对于CCE Agile平台,需要修改该字段,例如:csi.oceanstor.com。</li> </ul>
csiDriver.end point	通信端点。	是	/csi/csi.sock	直接使用默认值。
csiDriver.con nectorThread s	最大并发扫盘/ 卸盘数。参数 格式为整型, 支持范围为 1~10。	是	4	该值设置越大,同一时间单个节点中的针对多路径的扫盘、卸盘并发操作就越多。在使用DM-Multipath时,并发数过大可能会导致未知问题,影响整体时间。
csiDriver.volu meUseMultip ath	是否使用多路 径软件。参数 格式为布尔 值。	是	true	强烈建议开启多路径 软件,以增强存储链 路的冗余度和性能。
csiDriver.scsi MultipathTyp e	存储协议为fc/iscsi时,使用的多路径软件。支持配置如下参数: DM-multipath HW-UltraPath HW-UltraPathNVMe	当 volumeUs eMultipat h为true时 必填。	DM- multipath	建议使用DM- multipath取值。
csiDriver.nvm eMultipathTy pe	存储协议为 roce/fc-nvme 时,使用的多 路径软件。仅 支持配置HW- UltraPath- NVMe。	当 volumeUs eMultipat h为true时 必填。	HW- UltraPath- NVMe	-

参数	描述	必选参数	默认值	备注
csiDriver.scan VolumeTime out	在主机上使用 DM-Multipath 多路径时,等 待多路径聚合 的超时时间, 支持范围为 1~600,单位 秒。	是	3	-
csiDriver.exec CommandTi meout	在主机上执行 命令的超时时 间	是	30	CSI插件在挂载,扩容 盘符等场景下,需要 运行一些主机命令, 例如使用mount命令 挂载文件系统。该配 置用于控制执行单条 命令的超时时间。
csi_driver.allP athOnline	是Multipath经的特数。 否dultipath经面面的, 在dultipath经面面的, 在dultipath经面面的, 在dultipath经面面的, 在dultipath经面面的, 在dultipath经面面的, 在dultipath经面面的。 在dultipath经面面的。 在dultipath经面面的。 在dultipath经面面的。 在dultipath经面面的。 是dultipath经面面的。 是dultipath经面面的。 是dultipath经面面的。 是dultipath经面面的。 是dultipath经面面的。 是dultipath经面面的。 是dultipath经面面的。 是dultipath经面面的。 是dultipath经面面的。 是dultipath经面面的。 是dultipath经面面的。 是dultipath经面面的。 是dultipath经面面的。 是dultipath经面面的。 是dultipath经面面的。 是dultipath经面面的。 是dultipath经面面的。 是dultipath经面面的。 是dultipath是面面的。 是dultipath是面面的。 是dultipath是面面的。 是dultipath是面面的。 是dultipath是面面的。 是dultipath是面面的。 是dultipath是面面的。 是dultipath。 是dultipath自然是 是dultipa	当 csi_driver.s csiMultipa thType为 DM- multipath 时必填。	false	
csiDriver.ena bleLabel	是否支持将 Kubernetes资 源对象标签下 发至存储。	是	false	只有集中式存储6.1.7 及以上版本支持。 分布式存储、dtree、 静态PV均不支持。

参数	描述	必选参数	默认值	备注
csiDriver.back endUpdateIn terval	后端能力的更 新时间间隔, 支持范围 60~600,单位 秒。	是	60	-
csiDriver.cont rollerLogging .module	controller日志 记录类型。支 持配置如下参 数: • file • console	是	file	使用file选项时,日志将被保留在节点指定的目录下,当CSI所在的Pod被销毁时,日志任然被保留。使用console选项时,日志将被保留在CSI所在Pod的临时空间中,当CSI所在的Pod被销毁时,日志也随之被销毁。
csiDriver.cont rollerLogging .level	controller日志 输出级别。支 持配置如下参 数: • debug • info • warning • error • fatal	是	info	
csiDriver.cont rollerLogging .fileDir	controller日志 在file输出模式 下的日志目 录。	是	/var/log/ huawei	请确保该目录下有足够的空间保留日志。 空间大小建议不小于 200 MB。
csiDriver.cont rollerLogging .fileSize	controller日志 在file输出模式 下单个日志文 件大小。	是	20M	-
csiDriver.cont rollerLogging .maxBackups	controller日志 在file输出模式 下日志文件备 份上限。	是	9	-

参数	描述	必选参数	默认值	备注
csiDriver.nod eLogging.mo dule	node日志记录 类型。支持配 置如下参数: • file • console	是	file	使用file选项时,日志将被保留在节点指定的目录下,当CSI所在的Pod被销毁时,日志任然被保留。使用console选项时,日志将被保留在CSI所在Pod的临时空间中,当CSI所在的Pod被销毁时,日志也随之被销毁。
csiDriver.nod eLogging.lev el	node日志输出 级别。支持配 置如下参数: • debug • info • warning • error • fatal	是	info	-
csiDriver.nod eLogging.file Dir	node日志在file 输出模式下的 日志目录。	是	/var/log/ huawei	请确保该目录下有足够的空间保留日志。 空间大小建议不小于 200 MB。
csiDriver.nod eLogging.file Size	node日志在file 输出模式下单 个日志文件大 小。	是	20M	-
csiDriver.nod eLogging.ma xBackups	node日志在file 输出模式下日 志文件备份上 限。	是	9	-

#### <u>/</u>注意

如果您的容器环境已经部署了华为CSI,请确保csiDriver.driverName的设置和之前部署时的配置保持一致。否则会导致系统中已存在的有华为CSI发放的卷/快照无法被新部署的华为CSI管理。

#### 4.1.1.5 其他参数配置说明

其他配置项包括了CSI插件某些特性的开关或者镜像获取策略。

表 4-5 其他配置项说明

参数	描述	必选参数	默认值	备注
kubernetes.nam espace	华为CSI运行时所在 Kubernetes命名空间, 支持用户自定义。名称 必须由小写字母、数字 和"-"组成,例如: my-name、123-abc。	否	huawei-csi	-
kubeletConfigDir	kubelet工作目录。	是	/var/lib/ kubelet	<ul> <li>直接使用默认值。</li> <li>对于Tanzu平台,需要修改该字段为/var/vcap/data/kubelet。</li> <li>对于CCEAgile平台,需要修改该字段为/mnt/paas/kubernetes/kubelet。</li> </ul>
sidecarImagePull Policy	sidecar镜像的拉取策 略。	是	IfNotPresent	-
huaweiImagePul lPolicy	huawei-csi镜像的拉取 策略。	是	IfNotPresent	-
CSIDriverObject.i sCreate	是否创建CSIDriver对象	是	false	CSIDriver特性在 Kubernetes v1.18成为GA版 本,因此要求 Kubernetes版本 高于v1.18,当 Kubernetes版本 低于 v1.18时, 请设置该参数为 false。

参数	描述	必选参数	默认值	备注
CSIDriverObject. attachRequired	CSI插件是否跳过attach操作。支持配置如下参数: • true:需要attach操作。 • false:跳过attach操作。	是	true	参数 attachRequired 在Kubernetes v1.18支持配置。 如果 CSIDriverObject. isCreate为true并且 attachRequired 参数设置为false 时,huawei-csi 插件将不会部署 csi-attacher这个 sidecar。  使用NAS存储 时支持配置为 false。 使用SAN存储 时,请配置为 true。

参数	描述	必选参数	默认值	备注
CSIDriverObject. fsGroupPolicy	基础卷是否支持在装载之前更改卷的所置如下参数:  ReadWriteOnceWithFSType: 仅当定的 accessModes包包 ReadWriteOnce时,才支持卷的名 ReadWriteOnce时,才支持卷的权和权限更不够的以匹明fsGroup有权的以匹配户向方域的以匹配户向方域的以匹配户向方域的以匹配户向方域的对外,不够不够的方域。  None: 将在下装载。  null: 将不设置 fsGroupPolicy参数	否	null	参数 fsGroupPolicy在 Kubernetes v1.20支持配置, 并且当 CSIDriverObject. isCreate为true时 该参数生效。 该特性在 Kubernetes v1.20中为Beta版 本,在 Kubernetes v1.23成为GA版 本,因此要求 Kubernetes版本 高于v1.20。
leaderElection.le aseDuration	领导者持续时间。	否	8s	仅多controller场 景生效。
leaderElection.re newDeadline	领导者重新选举时间。	否	6s	仅多controller场 景生效。
leaderElection.re tryPeriod	领导者选举重试时间。	否	2s	仅多controller场 景生效。

#### 须知

请确保此kubernetes.namespace填入的命名空间在Kubernetes上已经存在,如果不存在请使用如下命令创建对应的命名空间。本例中,华为CSI运行的命名空间为

"huawei-csi" 。 # kubectl create namespace *huawei-csi* 

# 4.1.2 安装华为 CSI

本章节介绍如何使用Helm 3安装部署华为CSI。

#### 须知

- 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10.6 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。

Helm是Kubernetes生态系统中的一个软件包管理工具,类似Ubuntu的APT、CentOS的YUM、或Python的pip一样,专门负责管理Kubernetes的应用资源。

使用Helm可以对Kubernetes应用进行统一打包、分发、安装、升级以及回退等操作。

- Helm的获取、安装请参考: https://helm.sh/docs/intro/install/
- Helm与Kubernetes版本对应关系请参考: https://helm.sh/docs/topics/version\_skew/

Helm在安装huawei-csi-controller时,将在指定命名空间的Deployment类型的工作负载中部署以下组件:

- huawei-csi-driver: 华为CSI驱动。
- storage-backend-controller: 华为后端管理控制器,管理storageBackendClaim资源。
- storage-backend-sidecar: 用于管理storageBackendContent资源。
- Kubernetes External Provisioner: 用于提供卷。
- Kubernetes External Attacher: 用干挂载券。
- (可选)Kubernetes External Snapshotter:提供快照支持(作为CRD安装)。
- (可选) Kubernetes External Snapshot Controller: 用于卷快照控制。
- Kubernetes External Resizer: 用于扩容卷。
- Kubernetes External liveness-probe: 用来判断Pod健康状态。

Helm在安装huawei-csi-node时,将在指定命名空间的DaemonSet类型的工作负载中部署以下组件:

- huawei-csi-driver: 华为CSI驱动。
- Kubernetes Node Registrar: 处理驱动程序注册。
- liveness-probe: 用来判断Pod健康状态。

# 4.1.2.1 Kubernetes、OpenShift、Tanzu 安装华为 CSI

本章节介绍如何在Kubernetes、OpenShift、Tanzu平台安装华为CSI。

#### 前提条件

- 华为CSI镜像已制作完成,并且按照3.3 上传华为CSI镜像章节说明,上传到镜像仓库或者导入到所有节点。
- 华为CSI安装运行所依赖的组件镜像都已经上传到镜像仓库或者导入到所有节点。
   具体信息请参考3.4 检查CSI依赖的镜像章节说明。
- 华为CSI运行所依赖的卷快照组件CRD已经安装。具体信息请参考3.5 检查卷快照 依赖组件章节说明。

- 如果您准备使用多路径联通华为存储,请确保所有计算节点上已安装多路径软件,具体信息请参考3.6 检查主机多路径配置章节说明。
- 容器管理平台已经安装部署了Helm 3。
- 已经准备安装CSI所需的values.yaml文件。具体信息请参考4.1.1 准备 values.yaml文件章节说明。
- 集群的所有worker节点与待接入的存储设备的业务IP地址通信正常,iSCSI场景下 允许使用ping命令进行连通性校验。
- 集群的所有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。
- 已经在需要对接的华为存储上创建了对接华为CSI所需要的账号,具体信息请参考 3.7 检查华为存储上的账号配置章节说明。

# 安装准备

OpenShift平台请根据以下命令创建SecurityContextConstraints资源

执行vi helm\_scc.yaml命令,创建SecurityContextConstraints文件。其中,下列回显中huawei-csi是指创建的命名空间,请根据客户实际情况填写。

# vi helm\_scc.yaml apiVersion: security.openshift.io/v1 kind: SecurityContextConstraints metadata: name: helm-scc allowHostDirVolumePlugin: true allowHostIPC: true allowHostPID: true allowHostPID: true allowHostPorts: true allowPrivilegeEscalation: true allowPrivilegedContainer: true

defaultAddCapabilities:
- SYS\_ADMIN
runAsUser:
type: RunAsAny
seLinuxContext:

type: RunAsAny fsGroup: type: RunAsAny users:

- system:serviceaccount:huawei-csi:huawei-csi-controller
- system:serviceaccount:huawei-csi:huawei-csi-node
- 执行oc create -f helm\_scc.yaml命令,创建SecurityContextConstraints。
   # oc create -f helm\_scc.yaml

Tanzu平台请执行以下命令配置kubelet安装目录

进入到安装包的helm/esdk目录下,执行vi values.yaml命令打开配置文件,修改 后保存。安装包目录请参见请参见表3-1。

# vi values.yaml

# Specify kubelet config dir path.

# kubernetes and openshift is usually /var/lib/kubelet

# Tanzu is usually /var/vcap/data/kubelet

# CCE is usually /mnt/paas/kubernetes/kubelet
kubeletConfigDir: /var/vcap/data/kubelet

Tanzu平台TKGI 1.16版本及以下请执行以下命令配置RBAC权限

执行vi rbac.yaml命令, 创建rbac.yaml文件。

# vi rbac.yaml apiVersion: rbac.authorization.k8s.io/v1 kind: ClusterRole metadata: name: huawei-csi-psp-role apiGroups: ['policy'] resources: ['podsecuritypolicies'] verbs: ['use'] apiVersion: rbac.authorization.k8s.io/v1 kind: ClusterRoleBinding metadata: name: huawei-csi-psp-role-cfg roleRef: kind: ClusterRole name: huawei-csi-psp-role apiGroup: rbac.authorization.k8s.io subjects: - kind: Group apiGroup: rbac.authorization.k8s.io name: system:serviceaccounts:huawei-csi - kind: Group apiGroup: rbac.authorization.k8s.io name: system:serviceaccounts:default

• 执行**kubectl create -f** *rbac.yaml*命令,创建RBAC权限。

# kubectl create -f rbac.yaml

## 部署步骤

**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工具路径请参见表3-1。

步骤3 进入到helm/esdk的工作目录下。

# cd helm/esdk

**步骤4** 执行kubectl apply -f crds/backend/命令更新CRD。

# kubectl apply -f crds/backend/

customresourcedefinition.apiextensions.k8s.io/storagebackendclaims.xuanwu.huawei.io configured customresourcedefinition.apiextensions.k8s.io/storagebackendcontents.xuanwu.huawei.io configured

步骤5 请务必按照3.5 检查卷快照依赖组件章节检查快照依赖组件,确认无误后执行helm install helm-huawei-csi./ -n huawei-csi --create-namespace命令安装华为CSI。

其中,helm-huawei-csi为自定义的Helm Chart名称,./表示使用当前目录下的Helm 工程,huawei-csi为自定义的Helm Chart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6 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,OpenShift平台使用oc get pod -n huawei-csi命令检查。

# kubectl get pod -n huawei-csi

NAME READY STATUS RESTARTS AGE huawei-csi-controller-6dfcc4b79f-9vjtq 9/9 Running 0 24m huawei-csi-controller-6dfcc4b79f-csphc 9/9 Running 0 24m

huawei-csi-node-g6f4k	3/3	Running	0	20m
huawei-csi-node-tas87	3/3	Running	Λ	20m

----结束

# 4.1.2.2 CCE / CCE Agile 安装华为 CSI

本章节介绍如何在CCE / CCE Agile平台安装华为CSI。

#### 前提条件

- 华为CSI镜像已制作完成,并且按照**3.3 上传华为CSI镜像**章节说明,上传到镜像仓库或者导入到所有节点。
- 华为CSI安装运行所依赖的组件镜像都已经上传到镜像仓库或者导入到所有节点。
   具体信息请参考3.4 检查CSI依赖的镜像章节说明。
- 华为CSI运行所依赖的卷快照组件CRD已经安装。具体信息请参考3.5 检查卷快照依赖组件章节说明。
- 如果您准备使用多路径联通华为存储,请确保所有计算节点上已安装多路径软件,具体信息请参考3.6 检查主机多路径配置章节说明。
- 容器管理平台已经安装部署了Helm 3。
- 已经准备安装CSI所需的values.yaml文件。具体信息请参考4.1.1 准备 values.yaml文件章节说明。
- 集群的所有worker节点与待接入的存储设备的业务IP地址通信正常,iSCSI场景下 允许使用ping命令进行连通性校验。
- 集群的所有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。
- 已经在需要对接的华为存储上创建了对接华为CSI所需要的账号,具体信息请参考 3.7 检查华为存储上的账号配置章节说明。

#### 4.1.2.2.1 制作 Helm 安装包

CCE / CCE Agile平台安装华为CSI需要制作Helm安装包,本章节介绍如何制作Helm安装包。

#### 前提条件

- 节点服务器已经安装部署了Helm 3。
- 已经准备安装CSI所需的values.yaml文件。具体信息请参考4.1.1 准备 values.yaml文件章节说明。

## 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,登录已部署Helm的任意节点。

步骤2 将华为CSI组件包中的"helm"目录拷贝到节点的任意目录下。Helm工具路径请参见表3-1。

步骤3 进入到helm的工作目录下。

# cd helm/

步骤4 修改helm/esdk/values.yaml文件中kubeletConfigDir和csiDriver.driverName参数。

- # vi ./esdk/values.yaml
- # Specify kubelet config dir path.
- # kubernetes and openshift is usually /var/lib/kubelet
- # Tanzu is usually /var/vcap/data/kubelet
- # CCE is usually /mnt/paas/kubernetes/kubelet kubeletConfigDir: /mnt/paas/kubernetes/kubelet
- # The CSI driver parameter configuration csiDriver:
- # Driver name, it is strongly recommended not to modify this parameter
- # The CCE platform needs to modify this parameter, e.g. csi.oceanstor.com

**步骤5** 执行helm package ./esdk/ -d ./命令制作Helm安装包,该命令会将安装包生成到当前路径下。

# helm package ./esdk/ -d ./ Successfully packaged chart and saved it to: esdk-1.0.0.tgz

----结束

#### 4.1.2.2.2 安装华为 CSI

本章节介绍如何在CCE / CCE Agile平台安装华为CSI,以CCE Agile v22.3.2为例。

#### 前提条件

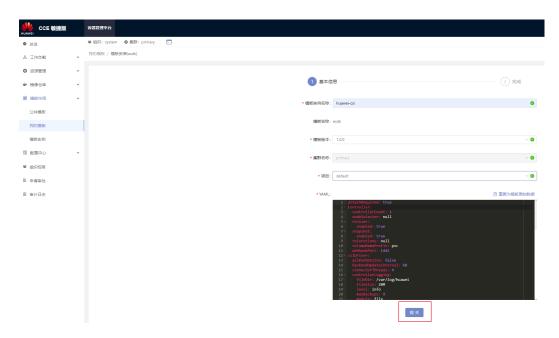
- 已按照前提条件检查安装准备事项。
- 华为CSI Helm安装包已制作完成,具体信息请参考4.1.2.2.1 制作Helm安装包。

## 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,登录已部署CCE Agile平台 master的任意节点。
- **步骤2** 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。
  - # kubectl create namespace huawei-csi
- 步骤3 导出Helm安装包,具体请参考4.1.2.2.1 制作Helm安装包。
- 步骤4 在主页单击"模板市场> 我的模板>上传模板",进入上传模板对话框。将导出的 Helm安装包导入CCE Agile平台。



**步骤5** 安装包上传完毕,在主页单击"模板市场>我的模板",进入我的模板页面,单击"安装>提交"。其中模板实例名称可自定义填写。



**步骤6** 在主页单击"模板市场>模板实例",选择安装时指定的项目(例如样例中的项目是"default")。安装成功后执行状态将回显为"安装成功"。



----结束

# 4.2 手动安装华为 CSI

本章节介绍如何手动安装华为CSI。

#### 山 说明

手动安装华为CSI当前仅支持Kubernetes平台。

#### 前提条件

- 华为CSI镜像已制作完成,并且按照3.3 上传华为CSI镜像章节说明,上传到镜像仓库或者导入到所有节点。
- 华为CSI安装运行所依赖的组件镜像都已经上传到镜像仓库或者导入到所有节点。
   具体信息请参考3.4 检查CSI依赖的镜像章节说明。
- 华为CSI运行所依赖的卷快照组件CRD已经安装。具体信息请参考**3.5 检查卷快照** 依赖组件章节说明。

- 如果您准备使用多路径联通华为存储,请确保所有计算节点上已安装多路径软件,具体信息请参考3.6 检查主机多路径配置章节说明。
- 集群的所有worker节点与待接入的存储设备的业务IP地址通信正常,iSCSI场景下允许使用ping命令进行连通性校验。
- 集群的所有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。
- 已经在需要对接的华为存储上创建了对接华为CSI所需要的账号,具体信息请参考 3.7 检查华为存储上的账号配置章节说明。

#### 安装步骤

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

步骤2 将Kubernetes CSI组件包中的"manual"目录拷贝到master节点的任意目录下。

步骤3 进入到manual/esdk的工作目录下。具体路径请参见表3-1。

# cd manual/esdk

#### **步骤4** 执行kubectl apply -f crds/backend/命令安装CRD。

# kubectl apply -f crds/backend/

customresourcedefinition.apiextensions.k8s.io/resourcetopologies.xuanwu.huawei.io configured customresourcedefinition.apiextensions.k8s.io/storagebackendclaims.xuanwu.huawei.io configured customresourcedefinition.apiextensions.k8s.io/storagebackendcontents.xuanwu.huawei.io configured

#### 步骤5 执行kubectl create -f ./deploy命令安装部署CSI。

# kubectl create -f ./deploy

csidriver.storage.k8s.io/csi.huawei.com created

clusterrole.rbac.authorization.k8s.io/storage-backend-controller-role created clusterrole.rbac.authorization.k8s.io/storage-backend-sidecar-role created clusterrolebinding.rbac.authorization.k8s.io/storage-backend-controller-binding created

clusterrolebinding.rbac.authorization.k8s.io/storage-backend-controller-binding created clusterrolebinding.rbac.authorization.k8s.io/storage-backend-sidecar-binding created

serviceaccount/huawei-csi-controller created

rolebinding.rbac.authorization.k8s.io/huawei-csi-attacher-role-cfg created

role.rbac.authorization.k8s.io/huawei-csi-attacher-cfg created

 $role binding. rbac. authorization. k8s. io/huawei-csi-provisioner-role-cfg\ created$ 

role.rbac.authorization.k8s.io/huawei-csi-provisioner-cfg created

 $cluster role binding. rbac. authorization. k8s. io/huawei-csi-provisioner-role\ created$ 

clusterrole.rbac.authorization.k8s.io/huawei-csi-provisioner-runner created clusterrolebinding.rbac.authorization.k8s.io/huawei-csi-attacher-role created

clusterrole.rbac.authorization.k8s.io/huawei-csi-attacher-runner created

rolebinding.rbac.authorization.k8s.io/huawei-csi-resizer-role-cfg created

rolebinding.rbac.authorization.k8s.io/huawei-csi-csi-resizer-role-ctg role.rbac.authorization.k8s.io/huawei-csi-resizer-cfg created

clusterrolebinding.rbac.authorization.k8s.io/huawei-csi-csi-resizer-role created

clusterrole.rbac.authorization.k8s.io/huawei-csi-resizer-runner created

clusterrole.rbac.authorization.k8s.io/huawei-csi-controller-runner created

 $cluster role binding.rbac. authorization. k8s. io/huawei-csi-controller-role\ created$ 

deployment.apps/huawei-csi-controller created

service/huawei-csi-controller created

serviceaccount/huawei-csi-node created

 $cluster role binding.rbac. authorization. k8s. io/huawei-csi-driver-registrar-role\ created$ 

 $cluster role.rbac. authorization. k8s. io/huawei-csi-driver-registrar-runner\ created$ 

clusterrole.rbac.authorization.k8s.io/huawei-csi-node-runner created

clusterrolebinding.rbac.authorization.k8s.io/huawei-csi-node-role created

daemonset.apps/huawei-csi-node created

# 步骤6 完成步骤5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-6dfcc4b79f-9vjtq 9/9 Running 0 24m huawei-csi-controller-6dfcc4b79f-csphc 9/9 Running 0 24m 中指南 4 安装部署

huawei-csi-node-g6f4k	3/3	Running	0	20m
huawei-csi-node-tqs87	3/3	Running	0	20m

----结束

# 5 卸载华为 CSI

- 5.1 Helm卸载华为CSI
- 5.2 手动卸载华为CSI

# 5.1 Helm 卸载华为 CSI

本章节介绍如何卸载华为CSI。根据您安装时的方式,请使用不同的方式进行卸载。

## **注意**

如果您不是出于升级的目的卸载华为CSI,请确保卸载华为CSI前已经在您的容器平台中将华为CSI发放的资源(PV、PVC、快照、存储后端等)全部清理。否则一旦您卸载华为CSI后,这些资源将无法被自动调度、管理或者清理。

# 5.1.1 Kubernetes、OpenShift、Tanzu 卸载华为 CSI

本章节介绍如何在Kubernetes、OpenShift、Tanzu平台卸载华为CSI。

## 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-huawei-csi* -n *huawei-csi*命令卸载华为CSI,*helm-huawei-csi是*自定义的Helm Chart名称,*huawei-csi是*该Helm Chart所在的命名空间。该卸载命令将会删除华为CSI的huawei-csi-controller、huawei-csi-node和RBAC资源。

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

删除命令执行后,还需要检查卸载是否成功。

# helm list -n huawei-csi

NAME NAMESPACE REVISION UPDATED STATUS CHART APP VERSION

其中,huawei-csi为chart所在的命名空间。

如果回显为空,则表示服务删除成功。

步骤3 删除huawei-csi-host-info对象,请参考5.1.3.1 删除huawei-csi-host-info对象进行操作。

步骤4 删除webhook资源,请参考5.1.3.2 删除Webhook资源进行操作。

步骤5 (可选)卸载快照依赖组件服务,请参考5.1.3.3 卸载Snapshot依赖组件服务进行操作。

----结束

# 5.1.2 CCE / CCE Agile 卸载华为 CSI

本章节介绍如何在CCE / CCE Agile平台卸载华为CSI,以CCE Agile v22.3.2为例。

## 操作步骤

步骤1 登录CCE Agile平台。

步骤2 在主页单击"模板市场>模板实例",进入模板实例页面。

步骤3 选择华为CSI模板实例,单击"卸载",在弹出的提示框中单击"确定"。



**步骤4** 删除huawei-csi-host-info对象,请参考**5.1.3.1 删除huawei-csi-host-info对象**进行操作。

步骤5 删除webhook资源,请参考5.1.3.2 删除Webhook资源进行操作。

步骤6 (可选)卸载快照依赖组件服务,请参考5.1.3.3 卸载Snapshot依赖组件服务进行操作。

----结束

# 5.1.3 删除 CSI 依赖组件服务

本章节介绍如何删除CSI依赖组件服务。

# 5.1.3.1 删除 huawei-csi-host-info 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-host-info -n huawei-csi命令删除secret对象,其中huawei-csi-host-info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-host-info -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-host-info对象已成功删除。

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

#### ----结束

#### 5.1.3.2 删除 Webhook 资源

**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webhook依赖组件服务。

# kubectl get validatingwebhookconfigurations.admissionregistration.k8s.io NAME WEBHOOKS AGE storage-backend-controller.xuanwu.huawei.io 1 12d

步骤3 执行以下命令, 卸载webhook依赖组件服务。

# kubectl delete validatingwebhookconfigurations.admissionregistration.k8s.io storage-backend-controller.xuanwu.huawei.io

步骤4 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 validatingwebhookconfigurations.admissionregistration.k8s.io

----结束

## 5.1.3.3 卸载 Snapshot 依赖组件服务

#### **注意**

- 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

- 当前华为CSI使用了快照特性。
- 当前Kubernetes集群仅存在华为CSI,且不再使用华为CSI。
- 在卸载前请确保在Kubernetes集群中已经没有华为CSI管理的VolumeSnapshot资源。

## 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

# 5.2 手动卸载华为 CSI

本章节介绍如何手动卸载华为CSI。

## <u>注意</u>

如果您不是出于升级的目的卸载华为CSI,请确保卸载华为CSI前已经在您的容器平台中将华为CSI发放的资源(PV、PVC、快照、存储后端等)全部清理。否则一旦您卸载华为CSI后,这些资源将无法被自动调度、管理或者清理。

# 5.2.1 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 替换为华为CSI所在的命名空间。

# 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

# 5.2.2 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 替换为华为CSI所在的命名空间。

# 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

# 5.2.3 卸载 csidriver 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 csidriver csi.huawei.com命令卸载csidriver对象。

# kubectl delete csidriver csi.huawei.com

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 csidriver csi.huawei.com

----结束

# 5.2.4 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。

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

----结束

# 5.2.5 其它资源卸载

## 操作步骤

步骤1 删除huawei-csi-host-info对象,请参考5.1.3.1 删除huawei-csi-host-info对象进行操作。

步骤2 删除webhook资源,请参考5.1.3.2 删除Webhook资源进行操作。

步骤3 (可选)卸载快照依赖组件服务,请参考5.1.3.3 卸载Snapshot依赖组件服务进行操作。

----结束

# 6 升级/回退操作

本章节介绍如何升级/回退华为CSI。

- 6.1 使用Helm升级/回退华为CSI
- 6.2 手动升级/回退

# 6.1 使用 Helm 升级/回退华为 CSI

# 6.1.1 升级华为 CSI

本章节介绍如何升级华为CSI。

升级/回退过程中,已经存在的PVC/快照/Pod等资源会正常运行,不会影响您的业务访问。

#### **注意**

- 部分2.x版本CSI已经下架,若升级失败,可能无法回退到已下架版本的CSI。
- 在升级/回退过程中,不能使用华为CSI创建新的资源,或者对已有的PVC做挂载/卸载操作。
- 在升级/回退过程中,请勿卸载Snapshot依赖组件服务。

# 6.1.1.1 Kubernetes、OpenShift、Tanzu 升级华为 CSI

#### 2.x 和 3.x 版本的 CSI 升级至 4.2.0 版本

#### □ 说明

在CSI 2.x或3.x 版本中,使用块存储时,与存储建立映射的操作是在huawei-csi-node服务进行的,所以huawei-csi-node服务需要和存储管理网络通信,又由于huawei-csi-node服务是以DaemonSet部署的,在集群中每个节点都会部署一个huawei-csi-node服务,这样部署模型导致了在大规模集群下,每个huawei-csi-node服务都会向存储发起请求,可能导致存储连接数被占满,使得huawei-csi-node不能提供正常服务。

在CSI 4.x版本优化了该部署模型,将于存储建立映射操作迁移至huawei-csi-controller服务,huawei-csi-node服务不再需要和存储管理网络通信,降低了huawei-csi依赖的组网复杂度,同时huawei-csi-controller服务以Deployment形式部署,副本数根据客户可靠性要求设置,一般情况下,副本数为1~3。所以极大的减少了huawei-csi与存储的连接数量,使得huawei-csi服务能够接入大规模集群。

该架构变化可能会导致一个问题出现,即使用2.x或3.x发放的工作负载,升级CSI至4.x版本之后,如果产生了一次新的挂载流程,并且CO(Container Orchestration system)未调用huawei-csi提供的huawei-csi-controller服务,会导致挂载失败,解决该问题请参考11.13 创建Pod失败,Events日志显示"publishInfo doesn't exist"

如果您从2.x和3.x版本的CSI升级至4.2.0版本,请按照以下操作步骤升级:

**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 get cm huawei-csi-configmap -n** *huawei-csi* **-o json > configmap.json**命令备份后端信息到configmap.json文件中。OpenShift平台使用**oc**替 换**kubectl**命令。

# kubectl get cm huawei-csi-configmap -n huawei-csi -o json > configmap.json

步骤3 参考5.1 Helm卸载华为CSI卸载CSI。

步骤4 参考4.1.2 安装华为CSI安装当前版本的CSI。

步骤5 将步骤2中备份的后端信息,按照**7.1 新增存储后端**章节的说明安装。

----结束

## 从 4.x 版本的 CSI 升级至 4.2.0 版本。

如果您从4.x版本的CSI升级至4.2.0版本,请按照以下操作步骤升级:

**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 参考5.1 Helm卸载华为CSI卸载CSI。

步骤3 参考4.1.2 安装华为CSI安装当前版本的CSI。

----结束

# 6.1.2 回退华为 CSI

如果您从2.x和3.x版本的CSI升级至4.2.0版本失败,需要回退时,请然后参考**5.1 Helm 卸载华为CSI**卸载CSI,然后下载安装升级之前版本的CSI。

## **注意**

- 在升级/回退过程中,已经存在的PVC/快照/Pod等资源会正常运行,不会影响您的业务访问。
- 在升级/回退过程中,不能使用华为CSI创建新的资源,或者对已有的PVC做挂载/卸载操作。
- 在升级/回退过程中,请勿卸载Snapshot依赖组件服务。

# 6.1.2.1 Kubernetes、OpenShift、Tanzu 回退华为 CSI

#### 前提条件

已下载原版本CSI的软件包。

## 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集群的任意 master节点。

步骤2 参考5.1 Helm卸载华为CSI卸载CSI。

步骤3 参考原版本CSI安装包中的用户指南,重新安装原版本的CSI。

----结束

# 6.2 手动升级/回退

# 6.2.1 升级华为 CSI

本章节介绍如何手动升级华为CSI。

升级/回退过程中,已经存在的PVC/快照/Pod等资源会正常运行,不会影响您的业务访问。

## <u> 注意</u>

- 部分2.x版本CSI已经下架,若升级失败,可能无法回退到已下架版本的CSI。
- 在升级/回退过程中,不能使用华为CSI创建新的资源,或者对已有的PVC做挂载/卸载操作。
- 在升级/回退过程中,请勿卸载Snapshot依赖组件服务。

# 2.x 和 3.x 版本的 CSI 升级至 4.2.0 版本

如果您从2.x和3.x版本的CSI升级至4.2.0版本,请按照以下操作步骤升级:

**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 get cm huawei-csi-configmap -n** *huawei-csi* **-o json > configmap.json**命令备份后端信息到configmap.json文件中。OpenShift平台使用**oc**替换**kubectl**命令。

# kubectl get cm huawei-csi-configmap -n huawei-csi -o json > configmap.json

步骤3 参考5.2 手动卸载华为CSI卸载CSI。

步骤4 参考4.2 手动安装华为CSI安装当前版本的CSI。

步骤5 将步骤2中备份的后端信息,按照**7.1 新增存储后端**章节的说明安装。

----结束

## 从 4.x 版本的 CSI 升级至 4.2.0 版本。

如果您从4.x版本的CSI升级至4.2.0版本,请按照以下操作步骤升级:

**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 参考5.2 手动卸载华为CSI卸载CSI。

步骤3 参考4.2 手动安装华为CSI安装当前版本的CSI。

----结束

# 6.2.2 回退华为 CSI

请参考5.2 手动卸载华为CSI卸载CSI,然后下载安装升级之前版本的CSI。

#### **注意**

- 在升级/回退过程中,已经存在的PVC/快照/Pod等资源会正常运行,不会影响您的业务访问。
- 在升级/回退过程中,不能使用华为CSI创建新的资源,或者对已有的PVC做挂载/卸载操作。
- 在升级/回退过程中,请勿卸载Snapshot依赖组件服务。

#### 前提条件

已下载原版本CSI的软件包。

#### 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集群的任意 master节点。

步骤2 参考5.2 手动卸载华为CSI卸载CSI。

步骤3 参考4.2 手动安装华为CSI重新安装原版本的CSI。

----结束

# 了 存储后端管理

后端是华为存储资源的抽象概念,每台华为存储设备可以通过租户/存储池/协议等特性抽象出多个后端资源,每个后端独立存在,其中定义了为Kubernetes集群供应持久卷时所需要的华为存储信息。

本章节用于描述使用oceanctl工具管理存储后端,包括后端的创建/查询/更新/删除操作,oceanctl工具的帮助说明请执行命令"oceanctl --help"获取。

# 前提条件

- 获取oceanctl工具,将oceanctl工具拷贝到环境目录下,例如(/usr/local/bin), 且赋予可执行权限,oceanctl工具位于软件包/bin/oceanctl。
- oceanctl工具需要在master节点执行。
- oceanctl创建后端的命名空间默认为huawei-csi。
- 7.1 新增存储后端
- 7.2 查询存储后端
- 7.3 更新存储后端
- 7.4 删除存储后端
- 7.5 新增证书到存储后端(可选)
- 7.6 查询存储后端证书(可选)
- 7.7 更新存储后端证书(可选)
- 7.8 删除存储后端证书(可选)

# 7.1 新增存储后端

本章节介绍如何创建存储后端,当前支持根据配置的后端yaml文件和导出的 configmap.json文件两种方式创建后端。

如果通过新增后端yaml文件创建后端,请参考**7.1.1 准备存储后端配置文件**章节配置后端文件。

如果已存在导出的configmap.json文件,请参考**7.1.2 创建存储后端**章节创建存储后端。

# 7.1.1 准备存储后端配置文件

#### 山 说明

- 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。
- 对接NAS双活后端的账号必须为存储租户的租户管理员账号。

## 后端配置文件说明

后端配置文件样例模板为/examples/backend/backend.yaml,该文件为一个示例文件,具体配置项如下表所示:

表 7-1 backend 配置项说明

参数	描述	必选参数	默认值	备注
storage	存储服务类型。  • 企业存储提供SAN存储时填写oceanstor-san。  • 企业存储提供NAS存储时填写oceanstor-nas。  • 企业存储提供Dtree类型的NAS存储时填写oceanstor-dtree。  • 分布式存储提供SAN存储时填写fusionstoragesan。  • 分布式存储提供NAS存储时填写fusionstoragenas。	足	oce ans tor- nas	一个后端只允许提供一种 存储服务。如果单套华为 存储系统可以同时提供 SAN和NAS的存储服务 时,可以配置创建多个后 端,每个后端使用不同的 存储服务类型。
name	存储后端名称。支持小写字 母、数字和特殊字符"-",且 需要以字母或数字开头,最 多63个字符。	是	-	请保证存储后端名称唯 一。
namespac e	命名空间。	否	-	存储后端必须与华为CSI在相同的命名空间中。
vstoreNa me	存储侧的租户名称。当对接 后端是OceanStor V3/V5存 储,需要在指定租户下发放 资源时,需要指定该参数。	条件必选	-	仅对接后端是OceanStor V3/V5且需要支持租户 时,需要指定该参数。

参数	描述	必选参数	默 认 值	备注
accountN ame	存储侧的账户名称。当对接 资源是OceanStor Pacific NAS存储,需要在指定账户 下发放NAS资源时,需要指 定该参数。	条件必选	-	仅对接后端是OceanStor Pacific NAS存储且需要支 持账号时,需要指定该参 数。
urls	存储设备的管理URL。参数格式为列表。支持按照域名或者IP+端口的方式进行配置。仅支持IPv4。	是	-	当对接后端是OceanStor V6或OceanStor Dorado V6存储,需要在指定租户 下发放资源时,该参数配 置为指定租户的逻辑管理 端口URL。
pools	存储设备的存储池。参数格 式为列表。	条件必选	-	storage为oceanstor- dtree时, 可以不填。
parameter s.protocol	存储协议。参数格式为字符串。  • iscsi • fc • roce • fc-nvme • nfs • dpc • scsi	是	-	<ul> <li>使用iscsi时,请确保对接的计算节点已安装iSCSI客户端。</li> <li>使用nfs时,请确保对接的计算节点已安装NFS客户端工具。</li> <li>使用fc-nvme/roce时,请可安装nvme-cli工具,且版本不低于1.9。</li> <li>使用dpc时,请确保对接的计算节点已客户端,并自己的方式。</li> <li>使用scsi时,请确保对方点。</li> <li>使用scsi时,请确保对方点已有好点。</li> </ul>
parameter s.portals	业务访问端口。节点会使用该端口对存储资源进行读写访问。参数格式为一个列表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,仅支持IPv4。	条件必选	-	<ul> <li>使用租户/账户对接后端时,此时portals必须配置为租户/账户所拥有的逻辑端口信息。</li> <li>如果使用nfs协议,支持填写为域名地址。</li> </ul>

参数	描述	必选参数	默认值	备注
parameter s.ALUA	存储后端ALUA参数配置。当工作节点使用操作系统原生多路径,且启用了ALUA时,需要进行配置。	条件必选	-	如果主机多路径配置启用了ALUA,请确保后端ALUA配置和主机的ALUA配置一致。ALUA详细配置请参考9.1.1 通过Helm配置ALUA特性。
parameter s.parentn ame	当前存储上的某一个文件系统名称,在此文件系统下创建Dtree。 storage为oceanstor-dtree时必选。	条件必选	-	请到DeviceManager文件 系统界面查询。
metrovSto rePairID	双活租户Pair ID。 当需要创建PV在存储侧支持 NAS双活特性时,该字段必 填。此时需要填入待创建的 PV所归属的存储侧双活租户 Pair ID。	条件必选	-	双活租户Pair ID请到 DeviceManager界面查 询。
metroBac kend	双活对端的后端名称。参数格式为字符串。 当需要创建PV在存储侧支持NAS双活特性时,该字段必填。此时需要填入准备和当前后端组成双活的另一个后端名称。	条件必选	-	组对的两个后端都必须将 对方名称填入。这两个后 端组成双活关系后,不允 许再和其他后端组成双活 关系。
supported Topologie s	存储拓扑感知配置。参数格 式为列表类型的JSON。	条件必选	-	如果启用存储拓扑感知, 需要配置该参数。具体请 参考9.2.1 通过Helm配置 存储拓扑感知。
maxClient Threads	同时连接到存储后端的最大 连接数。	否	30	如果不配置该参数,则默 认最大连接数为30。

#### 典型场景的backend配置请参考如下示例

- 配置iSCSI协议类型的存储后端
- 配置FC协议类型的存储后端
- 配置NVMe over RoCE协议类型的存储后端
- 配置NVMe over FC协议类型的存储后端
- 配置NFS协议类型的存储后端
- 配置SCSI协议类型的存储后端

- 配置DPC协议类型的存储后端
- 配置Dtree类型的存储后端
- 配置双活类型的存储后端

# 配置 iSCSI 协议类型的存储后端

企业存储配置iSCSI协议类型的后端配置文件示例如下:

#### 分布式存储配置iSCSI协议类型的后端配置文件示例如下:

```
storage: "fusionstorage-san"
name: "pacific-iscsi-125"
namespace: "huawei-csi"
urls:
- "https://192.168.129.125:8088"
- "https://192.168.129.126:8088"
pools:
- "StoragePool001"
parameters:
protocol: "iscsi"
portals:
- "192.168.128.122"
- "192.168.128.123"
maxClientThreads: "30"
```

# 配置 FC 协议类型的存储后端

企业存储配置FC协议类型的后端配置文件示例如下:

```
storage: "oceanstor-san"
name: "fc-155"
namespace: "huawei-csi"
urls:
- "https://192.168.129.155:8088"
- "https://192.168.129.156:8088"
pools:
- "StoragePool001"
parameters:
protocol: "fc"
maxClientThreads: "30"
```

#### 配置 NVMe over RoCE 协议类型的存储后端

企业存储配置NVMe over RoCE协议类型的后端配置文件示例如下:

```
storage: "oceanstor-san"
name: "roce-155"
namespace: "huawei-csi"
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"

maxClientThreads: "30"
```

## 配置 NVMe over FC 协议类型的存储后端

企业存储配置NVMe over FC协议类型的后端配置文件示例如下:

```
storage: "oceanstor-san"
name: "fc-nvme-155"
namespace: "huawei-csi"
urls:
    - "https://192.168.129.155:8088"
    - "https://192.168.129.156:8088"
pools:
    - "StoragePool001"
parameters:
    protocol: "fc-nvme"
maxClientThreads: "30"
```

#### 配置 NFS 协议类型的存储后端

企业存储配置NFS协议类型的后端配置文件示例如下:

```
storage: "oceanstor-nas"
name: "nfs-155"
namespace: "huawei-csi"
urls:
    - "https://192.168.129.155:8088"
    - "https://192.168.129.156:8088"
pools:
    - "StoragePool001"
parameters:
protocol: "nfs"
portals:
    - "192.168.128.155"
maxClientThreads: "30"
```

#### 分布式存储配置NFS协议类型的后端配置文件示例如下:

```
storage: "fusionstorage-nas"
name: "nfs-126"
namespace: "huawei-csi"
urls:
- "https://192.168.129.125:8088"
- "https://192.168.129.126:8088"
pools:
- "StoragePool001"
parameters:
protocol: "nfs"
portals:
- "192.168.128.123"
maxClientThreads: "30"
```

## 配置 SCSI 协议类型的存储后端

分布式存储配置SCSI协议类型的后端配置文件示例如下:

```
storage: "fusionstorage-san"
name: "scsi-155"
```

```
namespace: "huawei-csi"
urls:
- "https://192.168.129.155:8088"
pools:
- "StoragePool001"
parameters:
protocol: "scsi"
portals:
- {"hostname01": "192.168.125.21", "hostname02": "192.168.125.22"}
maxClientThreads: "30"
```

## 配置 DPC 协议类型的存储后端

分布式存储配置DPC协议类型的后端配置文件示例如下:

```
storage: "fusionstorage-nas"
name: "dpc-155"
namespace: "huawei-csi"
urls:
- "https://192.168.129.155:8088"
- "https://192.168.129.156:8088"
pools:
- "StoragePool001"
parameters:
protocol: "dpc"
maxClientThreads: "30"
```

## 配置 Dtree 类型的存储后端

企业存储配置Dtree类型后端配置文件示例如下:

```
storage: "oceanstor-dtree"
name: "nfs-dtree"
namespace: "huawei-csi"
urls:
- "https://192.168.129.155:8088"
parameters:
protocol: "nfs"
parentname: "parent-filesystem"
portals:
- "192.168.128.155"
maxClientThreads: "30"
```

#### 配置双活类型的存储后端

CSI支持在对接OceanStor V6或OceanStor Dorado V6时,在存储侧发放NFS类型的双活卷。配置互为双活的存储后端时,需要分别创建两个配置文件,逐一创建后端。

本示例展示了如何为华为OceanStor V6或OceanStor Dorado V6存储配置双活类型的后端。首先创建本端的存储后端配置文件nfs-hypermetro-155.yaml:

```
storage: "oceanstor-nas"
name: "nfs-hypermetro-155"
namespace: "huawei-csi"
urls:
- "https://192.168.129.155:8088"
- "https://192.168.129.156:8088"
pools:
- "StoragePool001"
metrovStorePairID: "f09838237b93c000"
metroBackend: "nfs-hypermetro-157"
parameters:
protocol: "nfs"
portals:
- "192.168.129.155"
maxClientThreads: "30"
```

#### 创建本端后端完成后,创建远端的存储后端配置文件nfs-hypermetro-157.yaml:

```
storage: "oceanstor-nas"
name: "nfs-hypermetro-157"
namespace: "huawei-csi"
urls:
- "https://192.168.129.157:8088"
- "https://192.168.129.158:8088"
pools:
- "StoragePool001"
metrovStorePairID: "f09838237b93c000"
metroBackend: "nfs-hypermetro-155"
parameters:
protocol: "nfs"
portals:
- "192.168.129.157"
maxClientThreads: "30"
```

# 7.1.2 创建存储后端

#### □ 说明

- 使用oceanctl创建存储后端时,输入的账号和秘钥信息保存在Secret对象中,建议客户容器平台根据供应商或者K8s社区的建议自行对Secret进行加密。K8s社区对Secret加密可参考启用静态加密。
- 2. 通过json文件创建后端时,旧版本的backend名称中可能存在大写字母或"\_"字符。如果出现这种情况,旧的名称将会被重映射为一个新的名称,映射过程自动发生,不会影响原有功能。例如"ABC\_123"将会被映射为"abc-123-fd68e",具体映射规则如下:
  - 大写字母转换成小写字母。
  - "\_"字符转换成 "-"字符。
  - 末尾追加5位Hash码。

#### 命令说明

- 执行以下命令获取创建后端帮助。
  - oceanctl create backend -h
- 执行以下命令根据指定的yaml文件创建存储后端。 oceanctl create backend -f /path/to/backend.yaml -i yaml
- 执行以下命令根据指定的json文件创建存储后端,json文件仅支持通过json格式导出huawei-csi-configmap文件。
   oceanctl create backend -f /path/to/configmap.json -i json
- oceancii create backenu -i /patri/to/confighap.json -i json
- 执行以下命令在指定命名空间创建一个存储后端。
   oceanctl create backend -f /path/to/backend.yaml -i yaml -n <namespace>
- 执行以下命令创建存储后端,并忽略存储后端名称校验,例如大写和字符"-"。 oceanctl create backend -f /path/to/backend.yaml -i yaml --not-validate-name
- 执行以下命令创建存储后端,并指定provisioner,其中"csi.oceanstor.com"是 安装时指定的驱动名称,详情可以参考步骤4。

#### □ 说明

仅在CCE / CCE Agile平台创建后端时使用该命令。

oceanctl create backend -f /path/to/backend.yaml -i yaml --provisioner=csi.oceanstor.com

## 创建后端示例

**步骤1** 参考**7.1.1 准备存储后端配置文件**章节准备后端配置文件,如backend.yaml,若需创建 多个后端,请使用'''---''分隔。

```
storage: "oceanstor-san"
name: "backend-1"
namespace: "huawei-csi"
- "https://192.168.129.157:8088"
- "StoragePool001"
parameters:
 protocol: "roce"
 portals:
 - "10.10.30.20"
- "10.10.30.21"
maxClientThreads: "30"
storage: "oceanstor-san"
name: "backend-2"
namespace: "huawei-csi"
- "https://192.168.129.158:8088"
 - "StoragePool001"
parameters:
 protocol: "roce"
 portals:
  - "10.10.30.20"
  - "10.10.30.21"
maxClientThreads: "30"
```

#### 步骤2 执行以下命令创建存储后端。

#### 步骤3 输入待创建后端序号,并输入账号密码。

#### 步骤4 检查存储后端创建结果。

----结束

# 7.2 查询存储后端

# 命令说明

- 执行以下命令获取查询后端帮助。 oceanctl get backend -h
- 执行以下命令查询默认命名空间下单个存储后端。 oceanctl get backend <backend-name>
- 执行以下命令查询指定命名空间下所有存储后端。 oceanctl get backend -n <namespace>
- 执行以下命令格式化输出,当前支持json,yaml和wide。oceanctl get backend <backend-name> -o json

# 7.3 更新存储后端

#### 须知

- 更新存储后端当前只支持更新密码。
- 若在存储侧更新了后端的账号密码,CSI插件会因登录失败而重试,可能会导致账号被锁定。如果账号被锁定,请参考11.21 存储侧更新密码后账户被锁定章节修改。

# 命令说明

- 执行以下命令获取更新后端帮助。 oceanctl update backend -h
- 执行以下命令更新指定命名空间存储后端信息。 oceanctl update backend <backend-name> -n <namespace> --password

# 更新后端示例

#### 步骤1 执行以下命令获取更新存储后端帮助。

# oceanctl update backend -h Update a backend for Ocean Storage in Kubernetes

#### Usage:

oceanctl update backend <name> [flags]

#### Examples

# Update backend account information in default(huawei-csi) namespace oceanctl update backend <name> --password

# Update backend account information in specified namespace oceanctl update backend <name> -n namespace --password

Flags

```
-h, --help help for backend
-n, --namespace string namespace of resources
--password Update account password
```

#### 步骤2 执行以下命令更新存储后端信息。

# oceanctl update backend backend-1 --password Please enter this backend user name:admin Please enter this backend password:

backend/backend-1 updated

----结束

# 7.4 删除存储后端

#### 命令说明

- 执行以下命令获取删除后端帮助。
  - oceanctl delete backend -h
- 执行以下命令删除默认命名空间下所有存储后端。 oceanctl delete backend --all
- 执行以下命令删除指定命名空间下存储后端。
   oceanctl delete backend <backend-name...> -n <namespace>

#### 删除后端示例

#### 步骤1 执行以下命令获取存储后端。

#### 步骤2 执行以下命令删除指定存储后端。

# oceanctl delete backend backend-1 backend/backend-1 deleted

#### 步骤3 检查删除结果,如果回显为"not found"则删除成功。

# oceanctl get backend backend-1 Error from server (NotFound): backend "backend-1" not found

----结束

# 7.5 新增证书到存储后端(可选)

本章节介绍如何创建证书到存储后端,如果有对服务登录存储添加证书校验的需要,可以参考本章节新增证书,当前支持根据crt文件或者pem文件创建证书到存储后端。

须知

创建证书到存储后端前,须提前将准备好的证书导入存储阵列。

# 7.5.1 创建证书到存储后端

## 前提条件

完成证书制作。以OceanStor Dorado V6为例,证书制作过程请参考:点此前往。

# 命令说明

- 执行以下命令获取查询证书帮助。 oceanctl create cert -h
- 执行以下命令根据指定的crt证书文件为默认命名空间单个存储后端创建证书。oceanctl create cert <name> -f /path/to/cert.crt -b <base>backend-name>
- 执行以下命令根据指定的crt证书文件为指定命名空间单个存储后端创建证书。oceanctl create cert <name> -f /path/to/cert.crt -b <backend-name> -n <namespace>
- 执行以下命令根据指定的pem证书文件为指定命名空间单个存储后端创建证书。oceanctl create cert <name> -f /path/to/cert.pem -b <backend-name> -n <namespace>

# 创建证书示例

步骤1 提前准备好证书文件,如cert.crt。

步骤2 执行以下命令获取存储后端。

#### 步骤3 执行以下命令为指定存储后端创建证书。

# oceanctl create cert cert-1 -b backend-1 -f /path/to/cert.crt cert/cert-1 created

#### 步骤4 检查证书创建结果。

```
# oceanctl get cert -b backend-1
+------+
| NAMESPACE | NAME | BOUNDBACKEND |
+------+
| huawei-csi | cert-1 | backend-1 |
+-------+
```

#### ----结束

# 7.6 查询存储后端证书(可选)

## 命令说明

- 执行以下命令获取查询证书帮助。 oceanctl get cert -h
- 执行以下命令查询默认命名空间指定存储后端的证书。 oceanctl get cert -b <br/>backend-name>
- 执行以下命令查询指定命名空间下指定存储后端的证书。oceanctl get cert -b <base> oceanctl get cert -b <base>

# 7.7 更新存储后端证书(可选)

更新证书前请准备好新的证书文件,并参考本章节更新存储后端证书。如果不再使用证书,请参考7.8 删除存储后端证书(可选)章节移除存储后端上的证书。

## 命令说明

- 执行以下命令获取更新证书帮助。
   oceanctl update cert -h
- 执行以下命令根据指定的crt证书文件为默认命名空间指定存储后端更新证书。oceanctl update cert -b <base> -f /path/to/cert.crt
- 执行以下命令根据指定的crt证书文件为指定命名空间指定存储后端更新证书。oceanctl update cert -b <base> -n <namespace> -f /path/to/cert.crt
- 执行以下命令根据指定的pem证书文件为指定命名空间指定存储后端更新证书。oceanctl update cert -b <base> -n <namespace> -f /path/to/cert.pem

# 更新证书示例

步骤1 执行以下命令获取存储后端。

步骤2 执行以下命令查看指定存储后端是否存在证书。

```
# oceanctl get cert -b backend-1
+------+
| NAMESPACE | NAME | BOUNDBACKEND |
+------+
| huawei-csi | cert-1 | backend-1 |
+------+
```

步骤3 执行以下命令更新指定存储后端的证书。

# oceanctl update cert -b backend-1 -f /path/to/cert.crt cert/cert-1 updated

----结束

# 7.8 删除存储后端证书(可选)

#### 命令说明

- 执行以下命令获取删除证书帮助。 oceanctl delete cert -h
- 执行以下命令删除默认命名空间指定存储后端的证书。 oceanctl delete cert -b <backend-name>
- 执行以下命令删除指定命名空间指定存储后端的证书。
   oceanctl delete cert -b <br/> backend-name> -n <namespace>

#### 删除证书示例

#### 步骤1 执行以下命令获取存储后端。

#### 步骤2 执行以下命令获取指定存储后端的证书。

```
# oceanctl get cert -b backend-1
+-----+
| NAMESPACE | NAME | BOUNDBACKEND |
+------+
| huawei-csi | cert-1 | backend-1 |
+---------+
```

#### 步骤3 执行以下命令删除指定存储后端的证书。

# oceanctl delete cert -b backend-1 cert/cert-1 deleted

#### 步骤4 检查删除结果,如果回显为"no cert found"则删除成功。

# oceanctl get cert -b backend-1 Error from server (NotFound): no cert found on backend backend-1 in huawei-csi namespace

#### ----结束

# **8** 使用华为 CSI

本章节主要介绍如何使用华为CSI对PV、快照的生命周期进行管理。

#### 须知

在映射block卷时,华为CSI会自动创建创建主机、主机组、LUN组等这些卷映射需要的关联对象,以及映射视图。如果手动在存储上创建了这些对象,会影响华为CSI的映射逻辑,请确保在使用华为CSI映射卷前删除这些对象。

8.1 PV/PVC管理

8.2 创建VolumeSnapshot

# 8.1 PV/PVC 管理

根据业务的需求,容器中的文件需要在磁盘上进行持久化。当容器被重建或者重新分配至新的节点时,可以继续使用这些持久化数据。

为了可以将数据持久化到存储设备上,您需要在发放容器时使用**持久卷** (PersistentVolume, PV)以及**持久卷申领**(PersistentVolumeClaim, PVC)。

- PV:是Kubernetes集群中的一块存储,可以由管理员事先制备,或者使用存储类 (StorageClass)来动态制备。
- PVC: 是用户对存储的请求。PVC会耗用 PV 资源。PVC可以请求特定的大小和访问模式 (例如,可以要求 PV能够以 ReadWriteOnce、ReadOnlyMany 或 ReadWriteMany 模式之一来挂载,参见访问模式)。

本章将介绍如何使用华为CSI对PV/PVC进行创建、扩容、克隆以及从快照创建PVC。

# 8.1.1 创建 PVC

华为CSI支持在华为存储上创建存储资源(LUN/文件系统),并根据用户的设置供给容器使用。具体支持的特性请参考表2-5或者表2-9。

创建PVC的方式分为动态卷供应和静态卷供应。

动态卷供应不需要事先创建PV,华为CSI会根据StorageClass自动在存储设备上创建PV所需要的资源。并且可以在创建PVC时同时创建PV。

● 静态卷供应需要管理员事先在存储设备上创建好所需要的资源,通过创建PV的方式使用已存在的资源。并且可以在创建PVC时指定关联的PV。

#### 8.1.1.1 动态卷供应

动态卷供应(Dynamic Volume Provisioning)允许按需创建存储卷。动态卷供应依赖 StorageClass对象。 集群管理员可以根据需要定义多个StorageClass对象,在声明PV 或者PVC时,指定满足业务要求的StorageClass。华为CSI在从华为存储设备上申请资 源时,会根据StorageClass的预置定义,创建满足业务要求的存储资源。

为了完成动态卷供应,需要完成如下两步:

- 配置StorageClass
- 配置PVC

#### 8.1.1.1.1 配置 StorageClass

**存储类(StorageClass**)为管理员提供了描述存储 "类" 的方法。 不同的类型可能会映射到一组不同的能力定义。Kubernetes集群用户可基于StorageClass进行动态卷制备。

StorageClass支持配置如下参数信息。

使用SAN存储时可参考示例文件/examples/sc-lun.yaml,使用NAS存储时可参考示例文件/examples/sc-fs.yaml。

表 8-1 StorageClass 配置参数说明

参数	说明	必选参数	默认 值	备注
metadata.nam e	自定义的 StorageClass对象 名称。	是	-	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	制备器名称。	是	csi.h uawe i.com	该字段需要指定为安装华为CSI 时设置的驱动名称。 取值和values.yaml文件中 driverName一致。
reclaimPolicy	回收策略。支持如下类型:  Delete:自动回收资源。  Retain:手动回收资源。	否	Delet e	<ul> <li>Delete: 删除PV/PVC时会关 联删除存储上的资源。</li> <li>Retain: 删除PV/PVC时不会 删除存储上的资源。</li> </ul>

参数	说明	必选参数	默认值	备注
allowVolumeE xpansion	是否允许卷扩展。 参数设置为true 时,使用该 StorageClass的PV 可以进行扩容操 作。	否	false	此功能仅可用于扩容PV,不能 用于缩容PV。 扩容PV功能在Kubernetes 1.14 (alpha)后才支持。
parameters.ba ckend	待创建资源所在的 后端名称。	否	-	如果不设置,华为CSI随机选择 一个满足容量要求的后端创建 资源。 建议指定后端,确保创建的资 源在预期的后端上。
parameters.po ol	待创建资源所在的存储资源池名称。如果设置,则必须设置 parameters.backend。	否	-	如果不设置,华为CSI会在所选 后端上随机选择一个满足容量 要求的存储池创建资源。建议 指定存储池,确保创建的资源 在预期的存储池上。
parameters.vol umeType	待创建卷类型。支持如下类型:  • lun:存储侧发放的资源是LUN。  • fs:存储侧发放的资源是 放的资源是 放 的资源是 放 的资源是 放 的资源是 放 的资源是 文件系统。  • dtree:存储侧发放离处件系统。  • dtree:存储侧发放的资源是 及tree类型的卷	是	-	<ul> <li>使用NAS存储时,必须配置为fs。</li> <li>使用SAN存储时,必须配置为lun。</li> <li>使用Dtree类型的NAS存储时,必须配置为dtree</li> </ul>
parameters.all ocType	待创建卷的分配类型。支持如下类型:  thin:创建时不会分配所有需要的空间,而是根据使用情况动态分配。  thick:创建时分配所有需要的空间。	否	-	传空相当于传thin,创建时不会分配所有需要的空间,而是根据使用情况动态分配。 OceanStor Dorado V3/V6 不支持thick

参数	说明	必选参数	默认值	备注
parameters.fsT ype	主机文件系统类型。支持类型为:     ext2     ext3     ext4     xfs	否	ext4	仅当StorageClass的 volumeType设置为"lun",且 PVC的volumeMode配置为 "Filesystem"时生效。
parameters.au thClient	可访问该卷的NFS 客户端IP地址信息,在指定 volumeType为 "fs"时必选。 支持输入客户端别名称(建议包含。 在和这个人。 支持输入客户端别名称(建议)。 在一个人。 证明地址以图,	条件必选	-	可以使用"*"表示任意客户端。当您不确定访问客户端IP信息时,建议使用"*"防止客户端访问被存储拒绝。 当使用客户端主机名称时建议使用全称域名。 IP地址支持IPv4、IPv6地址或两者的混合IP地址。 可以同时输入多个主机名称、IP地址或IP地址段,以英文分号,空格或按回车键隔开。如示例: "192.168.0.10;192.168.0.0/24; myserver1.test"
parameters.clo neSpeed	克隆速度。支持配 置为1~4。	否	3	4速度最快。配置克隆PVC或从 快照创建PVC时生效,参考 8.1.3 克隆PVC或8.1.4 从快照 创建PVC。
parameters.ap plicationType	后端为OceanStor Dorado V6存储 时,指定创建 LUN/NAS时的应用 类型名称。	否	-	<ul> <li>"volumeType"为"lun"时,在DeviceManager管理界面,选择"服务&gt;块服务&gt;比队组&gt;LUN&gt;创建&gt;应用类型",获取应用类型名称。</li> <li>"volumeType"为"fs"时,在DeviceManager管理界面,选择"服务&gt;文件服务&gt;文件服务&gt;文件服务&gt;文件系统&gt;创建&gt;应用类型",获取应用类型名称。</li> </ul>

参数	说明	必选参数	默认值	备注
parameters.qo s	PV在存储侧的 LUN/NAS的QoS设置。 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)。如:'{"maxMBPS":999,"maxIOPS":999}'	否	-	支持的QoS配置请参考 <b>表8-2</b> 说明。
parameters.sto rageQuota	PV在存储侧配额设置。仅在对接 OceanStor Pacific 系列存储使用NAS 时生效。 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)。如: '{"spaceQuota": "softQuota", "gracePeriod": 100}'	否	-	支持的配额配置请参考表8-3说明。
parameters.hy perMetro	是否创建双活卷。 当使用的后端是双 活类型的后端需要 配置。 • "true": 创建的 卷为双活卷。 • "false": 创建的 卷为普通卷。	条件必选	false	当使用的后端是双活类型的后端,且需要发放双活卷时,设置该参数。
parameters.fsP ermission	挂载到容器内的目 录权限。	否	-	配置格式参考Linux权限设置,如"777"、"755"等。 支持所有的SAN存储,NAS存储 仅支持OceanStor Dorado V6、OceanStor V6、 OceanStor Pacific 8.1.2及之后 版本的存储设备。

参数	说明	必选参数	默认值	备注
parameters.ro otSquash	用于设置是否允许客户端的root权限。可选值:  root_squash:表示公的,实现的一个方面,不可以不可以不可以不可以不可以不可以不可以不可以不可以不可以不可以不可以不可以不	否	-	仅支持NAS存储。
parameters.all Squash	用于设置是否保留 共享目录的UID和 GID。 可选值:  all_squash:表示共享目录的 UID和GID映射为匿名用户。  no_all_squash:表示保留共享目录的UID和 GID。	否	-	仅支持NAS存储。
parameters.sn apshotDirector yVisibility	用于设置快照目录是否可见。可选值: visible:表示快照目录可见。 invisible:表示快照目录不可见。	否	1	仅支持NAS存储。
parameters.res ervedSnapshot SpaceRatio	用于配置快照预留空间。 参数类型:字符串取值范围:"0"~"50"	否	-	支持OceanStor Dorado 6.1.5+、OceanStor 6.1.5+的 NAS存储。

参数	说明	必选参数	默认 值	备注
parameters.de scription	用于配置创建的文件系统/LUN的描述信息。 参数类型:字符串长度限制:0-255	否	-	仅支持企业存储文件系统及 LUN。
mountOptions .nfsvers	主机侧NFS挂载选项。支持如下挂载选项: nfsvers: 挂载NFS时的协议版本。支持配置的参数值为"3","4","4.0"和"4.1"。	否	-	在主机执行mount命令时-o参数后的可选选项。列表格式。 指定NFS版本挂载时,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。
mountOptions .acl	DPC命名空间支持 ACL功能。DPC客 户端支持POSIX ACL、NFSv4 ACL、NT ACL的鉴 权行为。	否	-	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。
mountOptions .aclonlyposix	DPC命名空间支持 POSIX ACL功能, DPC客户端支持 POSIX ACL的鉴权 行为。 支持POSIX ACL的 协议有: DPC、 NFSv3、HDFS。如 使用NFSv4 ACL或 NT ACL,会导致 DPC客户端无法识 别该类型的ACL, 从而导致该类型的 ACL不会生效。	否	-	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。

参数	说明	必选参数	默认 值	备注
mountOptions .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。	否	-	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。
mountOptions .cflush	同步刷盘模式,即 关闭命名空间下的 文件时立即刷盘。	否	-	默认使用同步刷盘模式。

## 表 8-2 支持的 QoS 配置

存储类型	参数名	参数描述	备注
OceanSt or V3/ OceanSt or V5	IOTYPE	控制读写类型。	可选参数(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 读I/O ① 1: 写I/O
			● 2: 读写I/O
	MAXBAN DWIDTH	最大带宽限制策 略。	单位MB/s,有效值为>0的整数。
	MINBAND WIDTH	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。

存储类型	参数名	参数描述	备注
OceanSt or Dorado	IOTYPE	控制读写类型。	有效值如下: ● 2: 读写I/O
V3	MAXBAN DWIDTH	最大带宽限制策 略。	单位MB/s,整数, 范围 1~999999999。
	MAXIOPS	最大IOPS限制策 略。	类型为整数, 范围 100~999999999。
OceanSt or Dorado	IOTYPE	控制读写类型。	有效值如下: ● 2: 读写I/O
V6/ OceanSt	MAXBAN DWIDTH	最大带宽限制策 略。	单位MB/s,类型为整数, 范围 1~999999999。
or V6 MINBAND 最小带宽 WIDTH 略。		最小带宽保护策 略。	单位MB/s,类型为整数, 范围 1~999999999。
	MAXIOPS	最大IOPS限制策 略。	类型为整数, 范围 100~999999999。
	MINIOPS	最小IOPS保护策 略。	类型为整数, 范围 100~999999999。
	LATENCY	最大时延保护策 略。	单位ms,仅支持配置0.5或1.5。
FusionSt orage/ OceanSt or	maxMBPS	最大带宽限制策 略。	必填。有效值为大于0的整数,单位 MB/s。最大值请参考存储设备实际 限制,如OceanStor Pacific NAS最 大值为1073741824。
Pacific系 列	maxIOPS	最大IOPS限制策 略。	必填。有效值为大于0的整数。最大值请参考存储设备实际限制,如OceanStor Pacific NAS最大值为1073741824000。

### □ 说明

- OceanStor V3/OceanStor V5 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 QoS

### 表 8-3 支持的配额配置

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"。

参数名	参数描述	备注
gracePeriod	配置软配额时,允许的超 限天数。	条件可选,当 "spaceQuota"配置为 "softQuota"时可选。 类型为整数,支持范围为0 ~4294967294。

### 典型场景下StorageClass配置请参考如下示例:

- StorageClass中设置后端和存储池
- StorageClass中设置NFS访问方式
- StorageClass中设置Dtree类型
- StorageClass中设置本地文件系统访问方式
- StorageClass中设置DPC访问方式
- StorageClass中设置应用类型
- StorageClass中设置软配额
- StorageClass中设置双活
- StorageClass中设置挂载目录权限
- StorageClass中设置QoS
- CCE / CCE Agile平台中配置StorageClass

## StorageClass 中设置后端和存储池

如果在一个Kubernetes集群中配置了多个华为后端,或者一个华为后端提供多个存储池,建议在StorageClass中配置指定的后端和存储池信息,避免华为CSI随机选择后端和存储池,导致卷所在的存储不符合规划。

SAN存储设置后端和存储池可以参考如下配置示例。

kind: StorageClass apiVersion: storage.k8s.io/v1 metadata: name: mysc provisioner: csi.huawei.com allowVolumeExpansion: true parameters: backend: "iscsi\_san\_181" pool: "pool001" volumeType: lun allocType: thin

NAS存储设置后端和存储池可以参考如下配置示例。

kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
name: mysc
provisioner: csi.huawei.com
allowVolumeExpansion: true
parameters:
backend: "iscsi\_nas\_181"
pool: "pool001"
volumeType: fs

allocType: thin authClient: "\*"

## StorageClass 中设置 NFS 访问方式

容器使用NFS文件系统作为存储资源时,可以参考如下配置示例。该示例中,NFS挂载时指定版本为4.1。

kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
name: mysc
provisioner: csi.huawei.com
parameters:
backend: nfs\_nas\_181
pool: pool001
volumeType: fs
allocType: thin
authClient: "192.168.0.10;192.168.0.0/24;myserver1.test" #use \* for all client
mountOptions:
- nfsvers=4.1

## StorageClass 中设置 Dtree 类型

容器使用Dtree作为存储资源时,可以参考如下配置示例。

kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
name: mysc
provisioner: csi.huawei.com
parameters:
backend: nfs\_dtree
volumeType: dtree
allocType: thin
authClient: "\*"
mountOptions:
- nfsvers=4.1

# StorageClass 中设置本地文件系统访问方式

容器使用企业存储或者分布式存储的LUN作为存储资源时,且需要格式化文件系统为本地文件系统时,可以参考如下示例。

kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
name: mysc
provisioner: csi.huawei.com
parameters:
backend: iscsi\_lun\_181
pool: pool001
volumeType: lun
allocType: thin
fsType: xfs

# StorageClass 中设置 DPC 访问方式

当容器使用OceanStor Pacific系列存储,且存储支持DPC协议访问时,可以在 StorageClass中配置DPC访问的挂载参数。本例中设置挂载时使用"acl"做鉴权参 数,使用"cnflush"为设置异步刷盘模式。

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

```
metadata:
name: mysc
provisioner: csi.huawei.com
parameters:
backend: nfs_dpc_101
pool: pool001
volumeType: fs
allocType: thin
authClient: "*" #use * for all client
mountOptions:
- acl
- cnflush
```

## StorageClass 中设置应用类型

当容器使用OceanStor Dorado V6存储的LUN作为存储时,如果使用存储默认的应用类型无法满足某些业务的I/O模型要求(如容器对外提供数据库OLAP服务),可以在StorageClass中配置应用类型,提升存储性能。具体需要使用的应用类型请参考对应存储产品的产品文档说明。

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
name: mysc
provisioner: csi.huawei.com
parameters:
backend: iscsi_lun_181
pool: pool001
volumeType: lun
allocType: thin
fsType: xfs
applicationType: Oracle_OLAP
```

## StorageClass 中设置软配额

当容器使用OceanStor Pacific系列存储的文件系统作为存储时,可以在StorageClass中配置软配额信息,可以参考如下配置示例。

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
name: mysc
provisioner: csi.huawei.com
parameters:
backend: nfs_pacific_101
pool: pool001
volumeType: fs
allocType: thin
authClient: "*"
storageQuota: '{"spaceQuota": "softQuota", "gracePeriod": 100}'
mountOptions:
- nfsvers=3
```

# StorageClass 中设置 QoS

容器使用企业存储或者分布式存储作为存储资源时,可以为容器使用的存储资源设置 QoS,从而保证这些容器对存储读写满足一定的服务等级。

不同型号或版本的存储支持的QoS设置不同,请参考表8-2找到对应存储的配置项。本示例中的后端是OceanStor Dorado V6存储,其他存储可以参考本例设置。

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

```
name: mysc
provisioner: csi.huawei.com
parameters:
backend: iscsi_qos_181
pool: pool001
volumeType: lun
allocType: thin
fsType: xfs
qos: '{"IOTYPE": 2, "MINIOPS": 1000}'
```

## StorageClass 中设置双活

容器使用NFS双活文件系统作为存储资源时,可以参考如下配置示例。该示例中,使用的后端是支持双活的后端,且配置"hyperMetro"参数为"true"。

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
name: mysc
provisioner: csi.huawei.com
parameters:
backend: nfs_hypermetro_dorado_181
pool: pool001
volumeType: fs
hyperMetro: "true"
allocType: thin
authClient: "*"
```

#### 须知

发放NAS双活卷前,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。

# StorageClass 中设置挂载目录权限

当需要修改容器内挂载目录的权限时,可以在StorageClass中配置目录权限信息,可以 参考如下配置示例。

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
name: mysc
provisioner: csi.huawei.com
allowVolumeExpansion: true
parameters:
volumeType: fs
allocType: thin
authClient: "*"
fsPermission: "777"
rootSquash: "no_root_squash" # 该参数仅支持NAS存储
allSquash: "no_all_squash" # 该参数仅支持NAS存储
```

完成StorageClass配置后,进行如下步骤创建StorageClass。

步骤1 执行以下命令,基于该yaml文件创建StorageClass。

```
# kubectl create -f mysc.yaml
storageclass.storage.k8s.io/mysc created
```

步骤2 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

创建StorageClass后,就可以使用该StorageClass进行创建PV或者PVC。

#### ----结束

## <u> 注意</u>

在StorageClass的使用中请注意如下事项:

- 请不要删除正在被PV使用的StorageClass。否则会导致PV的StorageClass信息缺失,导致在主机侧挂载时出现错误。
- 针对StorageClass进行的修改将不会在已经创建的PV上生效。您需要删除这些PV,并重新使用修改后的StorageClass创建才能应用修改的参数。

## CCE / CCE Agile 平台中配置 StorageClass

在CCE / CCE Agile平台中创建NAS类型StorageClass,可以参考如下配置示例。其中 provisioner保持和values.yaml文件中driverName一致。

kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
name: mysc
annotations:
storageclass.kubernetes.io/storageType: file
provisioner: csi.oceanstor.com
allowVolumeExpansion: true
parameters:
volumeType: fs
allocType: thin
authClient: "\*"

在CCE / CCE Agile平台中创建Block类型StorageClass,可以参考如下配置示例。其中provisioner保持和values.yaml文件中driverName一致。

kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
name: mysc
annotations:
storageclass.kubernetes.io/storageType: block
provisioner: csi.oceanstor.com
allowVolumeExpansion: true
parameters:
volumeType: lun
allocType: thin

#### 8.1.1.1.2 配置 PVC

在完成配置StorageClass以后,就可以用该StorageClass来配置PVC。PVC的配置模板 请参考华为CSI软件包中的examples目录下的pvc\*.yaml文件示例。

表 8-4 pvc\*.yaml 文件示例参数说明

参数	说明	必选参数	默认值	备注
metadata. name	自定义的PVC对象 名称。	是	-	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 meMode	卷模式。可选参数。当使用LUN类型的卷时,支持配置以下类型: • Filesystem:本地文件系统。 • Block:裸设备。	否	Files yste m	该参数在挂载PV时生效,默认为Filesystem。  Filesystem表示在容器通过一个本地文件系统访问PV,本地文件系统类型为指定StorageClass中的fsType字段指定, Dtree类型存储也使用此参数进行描述。  Block表示使用裸卷的方式访问访问PV。
spec.stora geClassNa me	StorageClass对象 名称。	是	-	业务需要的StorageClass对象名 称。
spec.resou rces.reque sts.storage	指定待创建卷大小,格式为***Gi,单位为GiB。需要满足大小为512字节的整数倍。	是	10Gi	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,当使用的是ext4文件系统时,容量限制见表8-5;当使用的是XFS文件系统时,容量限制见表8-6。如果使用的是NFS或者裸设备,容量需满足使用的华为存储设备型号和版本所要求的规格约束。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。

参数	说明	必选参数	默认 值	备注
spec.acces sModes	指 ■ RWO nce 一写。许节 Po	是	Read Writ eOn ce	<ul> <li>RWO/ROX/RWOP: 所有类型卷均支持, RWOP需Kubernetes 1.22版本以上支持。请参考10.5 开启ReadWriteOncePod功能门章节,检查您的Kubernetes集群是否开启该特性。</li> <li>RWX: volumeMode设置为Block或者NFS类型的卷支持。</li> </ul>

### 表 8-5 ext4 容量的规格

存储类型	存储规格限制	ext4规格限制	CSI规格限制
OceanStor Dorado 6.1.2	512Ki~256Ti	50Ti	512Ki~50Ti
OceanStor Pacific系列 8.1.0	64Mi~512Ti	50Ti	64Mi~50Ti

### 表 8-6 XFS 容量的规格

存储类型	存储规格限制	XFS规格限制	CSI规格限制
OceanStor Dorado 6.1.2	512Ki~256Ti	500Ti	512Ki~500Ti
OceanStor Pacific系列 8.1.0	64Mi~512Ti	500Ti	64Mi~500Ti

**步骤1** 根据业务需要,参考本节描述和PVC配置文件示例,修改具体参数,生成本次需要创建的PVC配置文件,如本例中mypvc.yaml文件。

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

### 步骤2 执行命令,使用配置文件创建PVC。

# kubectl create -f mypvc.yaml persistentvolumeclaim/mypvc created

**步骤3** 等待一段时间后,执行以下命令,查看已经创建的PVC信息。如果PVC的状态是 "Bound"时,则说明该PVC已经创建成功,后续可以被Pod使用。

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

## <u></u>注意

- 完成创建PVC操作后,如果长时间后(如一分钟后)PVC的状态是Pending,请参考 11.6 创建PVC时, PVC的状态为Pending。
- 建议每批次最多批量创建/删除100个PVC。

### ----结束

在完成PVC创建后,就可以使用PVC来创建Pod。如下示例是一个简单的使用PVC示例,在该示例中,创建的Pod使用了刚刚创建的*mypvc*。

```
apiVersion: apps/v1
kind: Deployment
metadata:
name: nginx-deployment
spec:
selector:
matchLabels:
app: nginx
replicas: 2
template:
metadata:
labels:
app: nginx
```

spec:

containers:

- image: nginx:alpine name: container-0 volumeMounts: - mountPath: /tmp name: pvc-mypvc restartPolicy: Always volumes: - name: pvc-mypvc

 name: pvc-mypvc persistentVolumeClaim: claimName: mypvc

# name of PVC

### 8.1.1.2 纳管卷供应

纳管卷供应(Manage Volume Provisioning)允许管理员使用已经在存储侧创建的资源做为PV,并能够支持动态卷的特性,例如:扩容,快照,克隆等,属于华为CSI自定义能力。使用该特性可满足如下场景:

- 容器化应用的改造场景,需要使用已有的存储卷。
- 重建Kubernetes集群。
- 容灾场景下,对存储数据进行迁移。

## 前提条件

- 已在CSI中注册需要纳管卷所在存储。
- 已登录存储设备获取需要纳管卷的名称和容量。

### 8.1.1.2.1 配置 StorageClass

**存储类(StorageClass**)为管理员提供了描述存储 "类" 的方法。 不同的类型可能会映射到一组不同的能力定义。Kubernetes集群用户可基于StorageClass进行动态卷制备。

StorageClass支持配置如下参数信息。

使用SAN存储时可参考示例文件/examples/sc-lun.yaml,使用NAS存储时可参考示例文件/examples/sc-fs.yaml。

表 8-7 StorageClass 配置参数说明

参数	说明	必选参数	默认 值	备注
metadata.nam e	自定义的 StorageClass对象 名称。	是	-	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	制备器名称。	是	csi.h uaw ei.co m	该字段需要指定为安装华为 CSI时设置的驱动名。 取值和values.yaml文件中 driverName一致。

参数	说明	必选参数	默认值	备注	
reclaimPolicy	回收策略。支持如下类型:  • Delete: 自动回收资源。  • Retain: 手动回收资源。	否	Delet e	<ul><li>Delete: 删除PV/PVC时会 关联删除存储上的资源。</li><li>Retain: 删除PV/PVC时不 会删除存储上的资源。</li></ul>	
allowVolumeE xpansion	是否允许卷扩展。 参数设置为true 时,使用该 StorageClass的PV 可以进行扩容操 作。	否	false	此功能仅可用于扩容PV,不能用于缩容PV。 扩容PV功能在Kubernetes 1.14 (alpha)后才支持。	
parameters.ba ckend	待创建资源所在的 后端名称。	否	-	如果不设置,华为CSI随机选 择一个满足容量要求的后端创 建资源。 建议指定后端,确保创建的资 源在预期的后端上。	
parameters.po ol	待创建资源所在的存储资源池名称。如果设置,则必须设置 parameters.backend。	否	-	如果不设置,华为CSI会在所 选后端上随机选择一个满足容 量要求的存储池创建资源。建 议指定存储池,确保创建的资 源在预期的存储池上。	
parameters.vol umeType	待创建卷类型。支持如下类型:  • lun: 存储侧发放的资源是LUN。  • fs: 存储侧发放的资源是文件系统。	是	-	<ul> <li>使用NAS存储时,必须配置为fs。</li> <li>使用SAN存储时,必须配置为lun。</li> </ul>	

参数	说明	必选参数	默认值	备注
parameters.all ocType	待创建卷的分配类型。支持如下类型:  thin: 创建时不会分配所有需要的空间,而是根据使用情况动态分配。 thick: 创建时分配所有需要的空间。	否	-	传空相当于传thin,创建时不会分配所有需要的空间,而是根据使用情况动态分配。 OceanStor Dorado V3/V6 不支持thick
parameters.fsT ype	主机文件系统类型。支持类型为:     ext2     ext3     ext4     xfs	否	ext4	仅当StorageClass的 volumeType设置为"lun", 且PVC的volumeMode配置为 "Filesystem"时生效。
parameters.au thClient	可访问该卷的NFS 客户端IP地址信 息,在指定 volumeType为 "fs"时必选。 支持输入客户端主 机名称(建议使用 全称域名)、客户 端IP地址段。	条件必选	*	可以使用"*"表示任意客户端。当您不确定访问客户端IP信息时,建议使用"*"防止客户端访问被存储拒绝。当使用客户端主机名称时建议使用全称域名。IP地址支持IPv4、IPv6地址或两者的混合IP地址。可以同时输入多个主机名称、IP地址或IP地址段,以英文分号,空格或按回车键隔开。如示例:"192.168.0.10;192.168.0.0/24;myserver1.test"
parameters.clo neSpeed	克隆速度。支持配 置为1~4。	否	-	如果不设置,默认值3。4速度最快。配置克隆PVC或从快照创建PVC时生效,参考8.1.3克隆PVC或8.1.4从快照创建PVC。

参数	说明	必选参数	默认值	备注
parameters.ap plicationType	后端为OceanStor Dorado V6存储 时,指定创建 LUN/NAS时的应用 类型名称。	否	-	<ul> <li>"volumeType"为"lun"时,在DeviceManager管理界面,选择"服务 &gt; 块服务 &gt; 比UN &gt; 创建 &gt; 应用类型",获取应用类型名称。</li> <li>"volumeType"为"fs"时,在DeviceManager管理界面,选择"服务 &gt; 文件服务 &gt; 文件服务 &gt; 文件服务 &gt; 放理 &gt; 应用类型",获取应用类型名称。</li> </ul>
parameters.qo s	PV在存储侧的 LUN/NAS的QoS设置。 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)。如:'{"maxMBPS":999, "maxIOPS":999}'	否	-	支持的QoS配置请参考 <b>表8-8</b> 说明。
parameters.sto rageQuota	PV在存储侧配额设置。仅在对接 OceanStor Pacific 系列存储使用NAS 时生效。 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)。 如: '{"spaceQuota": "softQuota", "gracePeriod": 100}'	否	-	支持的配额配置请参考表8-9 说明。

参数	说明	必选参数	默认 值	备注
parameters.hy perMetro	是否创建双活卷。 当使用的后端是双 活类型的后端需要 配置。 • "true": 创建的 卷为双活卷。 • "false": 创建的 卷为普通卷。	条件必选	false	当使用的后端是双活类型的后端,且需要发放双活卷时,设置该参数。
parameters.fsP ermission	挂载到容器内的目 录权限。	否	-	配置格式参考Linux权限设置,如"777"、"755"等。 支持所有的SAN存储,NAS存储仅支持OceanStor Dorado V6、OceanStor V6、 OceanStor Pacific 8.1.2及之 后版本的存储设备。
parameters.ro otSquash	用于设置是否允许客户端的root权限。可选值:  root_squash:表示公许图,这个时间,不可以不可以不可以不可以不可以不可以不可以不可以不可以不可以不可以不可以不可以不	否	-	仅支持NAS存储。

参数	说明	必选参数	默认值	备注
parameters.all Squash	m于设置是否保留 共享目录的UID和 GID。 可选值: • all_squash:表示共享目录的 UID和GID映射 为匿名用户。 • no_all_squash :表示保留共享 目录的UID和 GID。		-	仅支持NAS存储。
parameters.sn apshotDirector yVisibility	用于设置快照目录是否可见。可选值: visible:表示快照目录可见。 invisible:表示快照目录不可见。	否	-	仅支持NAS存储。
parameters.res ervedSnapshot SpaceRatio	用于配置快照预留空间。 参数类型:字符串取值范围: "0"~"50"	否	-	支持OceanStor Dorado 6.1.5+、OceanStor 6.1.5+的 NAS存储。
parameters.de scription	用于配置创建的文件系统/LUN的描述信息。 参数类型:字符串长度限制:0-255	否	-	仅支持企业存储文件系统及 LUN。
mountOptions .nfsvers	主机侧NFS挂载选项。支持如下挂载选项: 选项: nfsvers: 挂载NFS时的协议版本。支持配置的参数值为"3","4","4.0"和"4.1"。	否	-	在主机执行mount命令时-o参数后的可选选项。列表格式。 指定NFS版本挂载时,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。

参数	说明	必选参数	默认值	备注
mountOptions .acl	DPC命名空间支持 ACL功能。DPC客 户端支持POSIX ACL、NFSv4 ACL、NT ACL的鉴 权行为。	否	-	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。
mountOptions .aclonlyposix	DPC命名空间支持 POSIX ACL功能, DPC客户端支持 POSIX ACL的鉴权 行为。 支持POSIX ACL的 协议有: DPC、 NFSv3、HDFS。如 使用NFSv4 ACL或 NT ACL,会导致 DPC客户端无法识 别该类型的ACL, 从而导致该类型的 ACL不会生效。	否	-	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。
mountOptions .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。	否	-	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。
mountOptions .cflush	同步刷盘模式,即 关闭命名空间下的 文件时立即刷盘。	否	-	默认使用同步刷盘模式。

表 8-8 支持的 QoS 配置

存储类型	参数名	参数描述	备注	
OceanSt or V3/ OceanSt	IOTYPE	控制读写类型。	可选参数(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)。	
or V5			有效值如下:	
			● 0: 读I/O	
			● 1: 写I/O	
			● 2: 读写I/O	
	MAXBAN DWIDTH	最大带宽限制策略。	单位MB/s,有效值为>0的整数。	
	MINBAND WIDTH	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。	
OceanSt	IOTYPE	控制读写类型。	有效值如下:	
or Dorado			● 2: 读写I/O	
V3	MAXBAN DWIDTH	最大带宽限制策略。	单位MB/s,整数, 范围 1~999999999。	
	MAXIOPS	最大IOPS限制策略。	类型为整数, 范围 100~999999999。	
OceanSt	IOTYPE	控制读写类型。	有效值如下:	
or Dorado			● 2: 读写I/O	
V6/ OceanSt or V6	MAXBAN DWIDTH	最大带宽限制策略。	单位MB/s,类型为整数, 范围 1~999999999。	
	MINBAND WIDTH	最小带宽保护策略。	单位MB/s,类型为整数, 范围 1~999999999。	
	MAXIOPS	最大IOPS限制策略。	类型为整数, 范围 100~999999999。	
	MINIOPS	最小IOPS保护策略。	类型为整数, 范围 100~999999999。	
	LATENCY	最大时延保护策略。	单位ms,仅支持配置0.5或1.5。	

存储类型	参数名	参数描述	备注
FusionSt orage/	maxMBPS	最大带宽限制策略。	必填。有效值为>0的整数,单位 MB/s。
OceanSt or Pacific系 列	maxIOPS	最大IOPS限制策略。	必填。有效值为>0的整数。

#### □ 说明

- OceanStor V3/OceanStor V5 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 QoS

### 表 8-9 支持的配额配置

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"配置为 "softQuota"时可选。 类型为整数,支持范围为0 ~4294967294。

### 典型场景下StorageClass配置请参考如下示例:

- StorageClass中设置后端和存储池
- StorageClass中设置NFS访问方式
- StorageClass中设置本地文件系统访问方式
- StorageClass中设置DPC访问方式
- StorageClass中设置应用类型
- StorageClass中设置软配额
- StorageClass中设置双活
- StorageClass中设置挂载目录权限
- StorageClass中设置QoS

# StorageClass 中设置后端和存储池

如果在一个Kubernetes集群中配置了多个华为后端,或者一个华为后端提供多个存储 池,建议在StorageClass中配置指定的后端和存储池信息,避免华为CSI随机选择后端 和存储池,导致卷所在的存储不符合规划。

### SAN存储设置后端和存储池可以参考如下配置示例。

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
name: mysc
provisioner: csi.huawei.com
allowVolumeExpansion: true
parameters:
backend: "iscsi_san_181"
pool: "pool001"
volumeType: lun
allocType: thin
```

### NAS存储设置后端和存储池可以参考如下配置示例。

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
name: mysc
provisioner: csi.huawei.com
allowVolumeExpansion: true
parameters:
backend: "iscsi_nas_181"
pool: "pool001"
volumeType: fs
allocType: thin
authClient: "*"
```

## StorageClass 中设置 NFS 访问方式

容器使用NFS文件系统作为存储资源时,可以参考如下配置示例。该示例中,NFS挂载时指定版本为4.1。

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
name: mysc
provisioner: csi.huawei.com
parameters:
backend: nfs_nas_181
pool: pool001
volumeType: fs
allocType: thin
authClient: "192.168.0.10;192.168.0.0/24;myserver1.test" #use * for all client
mountOptions:
- nfsvers=4.1
```

# StorageClass 中设置本地文件系统访问方式

容器使用企业存储或者分布式存储的LUN作为存储资源时,且需要格式化文件系统为本地文件系统时,可以参考如下示例。

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
name: mysc
provisioner: csi.huawei.com
parameters:
backend: iscsi_lun_181
pool: pool001
volumeType: lun
allocType: thin
fsType: xfs
```

## StorageClass 中设置 DPC 访问方式

当容器使用OceanStor Pacific系列存储,且存储支持DPC协议访问时,可以在 StorageClass中配置DPC访问的挂载参数。本例中设置挂载时使用"acl"做鉴权参数,使用"cnflush"为设置异步刷盘模式。

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
name: mysc
provisioner: csi.huawei.com
parameters:
backend: nfs_dpc_101
pool: pool001
volumeType: fs
allocType: fs
allocType: thin
authClient: "*" #use * for all client
mountOptions:
- acl
- cnflush
```

## StorageClass 中设置应用类型

当容器使用OceanStor Dorado V6存储的LUN作为存储时,如果使用存储默认的应用类型无法满足某些业务的I/O模型要求(如容器对外提供数据库OLAP服务),可以在StorageClass中配置应用类型,提升存储性能。具体需要使用的应用类型请参考对应存储产品的产品文档说明。

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
name: mysc
provisioner: csi.huawei.com
parameters:
backend: iscsi_lun_181
pool: pool001
volumeType: lun
allocType: thin
fsType: xfs
applicationType: Oracle_OLAP
```

# StorageClass 中设置软配额

当容器使用OceanStor Pacific系列存储的文件系统作为存储时,可以在StorageClass中配置软配额信息,可以参考如下配置示例。

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
name: mysc
provisioner: csi.huawei.com
parameters:
backend: nfs_pacific_101
pool: pool001
volumeType: fs
allocType: thin
authClient: "*"
storageQuota: '{"spaceQuota": "softQuota", "gracePeriod": 100}'
mountOptions:
- nfsvers=3
```

## StorageClass 中设置 QoS

容器使用企业存储或者分布式存储作为存储资源时,可以为容器使用的存储资源设置 QoS,从而保证这些容器对存储读写满足一定的服务等级。

不同型号或版本的存储支持的QoS设置不同,请参考表8-8找到对应存储的配置项。本示例中的后端是OceanStor Dorado V6存储,其他存储可以参考本例设置。

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
name: mysc
provisioner: csi.huawei.com
parameters:
backend: iscsi_qos_181
pool: pool001
volumeType: lun
allocType: thin
fsType: xfs
qos: '{"IOTYPE": 2, "MINIOPS": 1000}'
```

## StorageClass 中设置双活

容器使用NFS双活文件系统作为存储资源时,可以参考如下配置示例。该示例中,使用的后端是支持双活的后端,且配置"hyperMetro"参数为"true"。

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
name: mysc
provisioner: csi.huawei.com
parameters:
backend: nfs_hypermetro_dorado_181
pool: pool001
volumeType: fs
hyperMetro: "true"
allocType: thin
authClient: "*"
```

#### 须知

发放NAS双活卷前,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。

# StorageClass 中设置挂载目录权限

当需要修改容器内挂载目录的权限时,可以在StorageClass中配置目录权限信息,可以 参考如下配置示例。

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
name: mysc
provisioner: csi.huawei.com
allowVolumeExpansion: true
parameters:
volumeType: fs
allocType: thin
authClient: "*"
fsPermission: "777"
```

rootSquash: "no\_root\_squash" # 该参数仅支持NAS存储 allSquash: "no\_all\_squash" # 该参数仅支持NAS存储

完成StorageClass配置后,进行如下步骤创建StorageClass。

步骤1 执行以下命令,基于该yaml文件创建StorageClass。

# kubectl create -f mysc.yaml storageclass.storage.k8s.io/mysc created

步骤2 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

创建StorageClass后,就可以使用该StorageClass进行创建PV或者PVC。

----结束

## <u></u>注意

在StorageClass的使用中请注意如下事项:

- 请不要删除正在被PV使用的StorageClass。否则会导致PV的StorageClass信息缺失,导致在主机侧挂载时出现错误。
- 针对StorageClass进行的修改将不会在已经创建的PV上生效。您需要删除这些PV,并重新使用修改后的StorageClass创建才能应用修改的参数。

### 8.1.1.2.2 配置 PVC

在完成配置StorageClass以后,就可以用该StorageClass来配置PVC。PVC的配置模板请参考华为CSI软件包中的examples目录下的pvc-manager.yaml文件示例。

表 8-10 pvc-manager.yaml 文件示例参数说明

参数	说明	必选参数	默认值	备注
metadat a.annota tions	PVC对象的注释。配置以下参数:  • 驱动名称/ manageVolume Name: 卷在存储 侧的名称。  • 驱动名称/ manageBackend Name: 卷所属后 端的名称。	是	csi.huawei.com/ manageVolumeNa me: * csi.huawei.com/ manageBackendNa me: *	<ul> <li>驱动名称获取请参考4.1.1.4 csiDriver参数配置说明。</li> <li>驱动名称/manageVolumeName:为存储上英有卷的名称,其他家字符不支持。</li> <li>驱动名称/manageBackendName:CSI中存储后端的名称。</li> <li>可执行oceanctl getbackend-nhuawei-csi命令获取后端名称。</li> </ul>
metadat a.labels	PVC对象的标签。	是	-	格式: provisioner: 安装时指定的驱动名 称。 例如 provisioner: csi.huawei.com。 该参数在创建PVC时 生效,用于监听PVC 资源,获取 metadata.annotatio ns信息。
metadat a.name	自定义的PVC对象名 称。	是	-	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。

参数	说明	必选参数	默认值	备注
spec.vol umeMo de	卷模用LUN类型:  • Filesystem: 文 是	否	Filesystem	该参数在挂载PV时生效。  Filesystem表示在容器通过一个问题,不可能够不是一个问题,不可能够不是一个可能的。 Filesystem表示在容器通过一个问题,不可能够不是一个问题,不可能够不是一个问题,不可能够不是一个问题,不可能够可能。  Filesystem表示在容器通过,不可能够可能的,可能够可能够可能够可能够可能够可能。  Filesystem表示在容器通过,可能够可能够可能够可能够可能。  Filesystem表示在容器通过,可能够可能够可能够可能够可能够可能够可能够可能够可能够可能够可能。  Filesystem表示在容器通过,可能够可能够可能够可能够可能够可能够可能够可能够可能够可能够可能够可能够可能够可
spec.stor ageClass Name	StorageClass对象名 称。	是	-	StorageClass的配置 需要与纳管卷的配置 保持一致。

参数	说明	必选参数	默认值	备注
spec.res ources.re quests.st orage	指定待创建卷大小,格式为***Gi,单位为GiB。需要满足大小为512字节的整数倍。		-	PVC容量的规格取决于机规制。OceanStor Dorado 6.1.2/OceanStor Pacific系列 8.1.0对接 CentOS 7为例件别的 8.1.0对连 CentOS 7为文制的 Py文制的 Py文制的 Py文制的 Py文制的 Py文制的 Py文制的 Py文制的 Py文制的 Py文制的 Py文本。是 Py文本的,格统规文本的,是 Py文本的,是 Py文本的,

参数	说明	必选参数	默认值	备注
spec.acc essMode s	指 ■ RWO ( ReadWriteOnc e ) 节挂允节点访 ■ ROX E Pod OnlyMan y 个式 WX E ROX E	是	ReadWriteOnce	<ul> <li>RWO/ROX/RWOP: 所有类型卷均支持,RWOP需Kubernetes         1.22版本多10.5开启ReadWriteOncePod功能门章节,检查您的Kubernetes集群是否开启该特性。</li> <li>RWX: volumeMode设置为Block或者NFS类型的卷支持。</li> </ul>

## 表 8-11 ext4 容量的规格

存储类型	存储规格限制	ext4规格限制	CSI规格限制
OceanStor Dorado 6.1.2	512Ki~256Ti	50Ti	512Ki~50Ti
OceanStor Pacific系列 8.1.0	64Mi~512Ti	50Ti	64Mi~50Ti

## 表 8-12 XFS 容量的规格

存储类型	存储规格限制	XFS规格限制	CSI规格限制
OceanStor Dorado 6.1.2	512Ki~256Ti	500Ti	512Ki~500Ti

存储类型	存储规格限制	XFS规格限制	CSI规格限制
OceanStor Pacific系列 8.1.0	64Mi~512Ti	500Ti	64Mi~500Ti

**步骤1** 根据业务需要,参考本节描述和PVC配置文件示例,修改具体参数,生成本次需要创建的PVC配置文件,如本例中mypvc.yaml文件。

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
 name: mypvc
 annotations:
  csi.huawei.com/manageVolumeName: *
  csi.huawei.com/manageBackendName: *
 labels:
  provisioner: csi.huawei.com
spec:
 accessModes:
  - ReadWriteOnce
volumeMode: Filesystem
storageClassName: mysc
resources:
 requests:
  storage: 100Gi
```

### 步骤2 执行命令,使用配置文件创建PVC。

# kubectl create -f mypvc.yaml persistentvolumeclaim/mypvc created

步骤3 等待一段时间后,执行以下命令,查看已经创建的PVC信息。如果PVC的状态是 "Bound"时,则说明该PVC已经创建成功,后续可以被Pod使用。

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

## <u> 注意</u>

- 完成创建PVC操作后,如果长时间后(如一分钟后)PVC的状态是Pending,请参考 11.6 创建PVC时, PVC的状态为Pending。
- 建议每批次最多批量创建/删除100个PVC。

#### ----结束

在完成PVC创建后,就可以使用PVC来创建Pod。如下示例是一个简单的使用PVC示例,在该示例中,创建的Pod使用了刚刚创建的*mypvc*。

```
apiVersion: apps/v1
kind: Deployment
metadata:
name: nginx-deployment
spec:
selector:
matchLabels:
app: nginx
replicas: 2
template:
metadata:
labels:
```

```
app: nginx
spec:
containers:
- image: nginx:alpine
name: container-0
volumeMounts:
- mountPath: /tmp
name: pvc-mypvc
restartPolicy: Always
volumes:
- name: pvc-mypvc
persistentVolumeClaim:
claimName: mypvc # name of PVC
```

### 8.1.1.3 静态卷供应

静态卷供应(Static Volume Provisioning)允许管理员使用已经在存储侧创建的资源做为PV,供集群中的容器使用。

为了完成静态卷供应,需要完成如下两步:

- 配置PV
- 配置PVC

### 8.1.1.3.1 配置 PV

使用静态卷供应时,不需要配置StorageClass,直接创建PV。创建PV前需要配置PV。如下示例是一个静态卷供应的配置文件。

```
kind: PersistentVolume
apiVersion: v1
metadata:
name: mypv
spec:
volumeMode: Filesystem
storageClassName: ""
accessModes:
- ReadWriteOnce
csi:
driver: csi.huawei.com
volumeHandle: iscsi-dorado-181.lun0001
fsType: xfs
capacity:
storage: 100Gi
```

如上例所示,静态卷供应的配置文件中,storageClassName参数必须配置为 """,如果不配置,Kubernetes会使用系统默认的StorageClass。其他参数及说 明见表8-13。

#### 表 8-13 静态卷供应参数

参数	说明	必选参数	默认 值	备注
metadata.n ame	自定义的PVC对象名 称。	是	-	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。

参数	说明	必选参数	默认值	备注
spec.volume Mode	卷模式。可选参数。 当使用LUN类型的卷时,支持配置以下类型:  Filesystem: 本地文件系统。 Block: 裸设备。	否	Filesy stem	该参数在挂载PV时生效,默认为Filesystem。  Filesystem表示在容器通过一个本地文件系统访问PV,本地文件系统类型为指定StorageClass中的fsType字段指定。  Block表示使用裸卷的方式访问访问PV。
spec.storage ClassName	StorageClass对象名 称。必选参数。	是	-	此处须设置为空字符串(即 输入"")。
spec.access Modes	指 ■ RWO (ReadWriteOnce ):点。 一节载运的卷 E ROX ● READWriteMany ) 节载 ● RWOP ● ReadWriteOnce Pod): 卷以读性 要以特 要以特 要以上版本 ● 1.22 以上版本	是	Read Write Once	<ul> <li>RWO/ROX/RWOP: 所有 类型卷均支持, RWOP需 Kubernetes 1.22版本以上 支持。请参考10.5 开启 ReadWriteOncePod功能 门章节,检查您的 Kubernetes集群是否开启 该特性。</li> <li>RWX: volumeMode设置 为Block或者NFS类型的卷 支持。</li> </ul>
spec.csi.driv er	CSI驱动名称。	是	csi.hu awei. com	该字段需要指定为安装华为 CSI时设置的驱动名称。

参数	说明	必选参数	默认 值	备注
spec.csi.volu meHandle	存储资源的唯一标志。必选参数。 格式为: <backendname>.<vo lume-name&gt;</vo </backendname>	是	-	该参数值由以下两部分构成:  ● <backendname>: 该卷所在的后端名称,可使用如下命令获取配置的后端信息:     oceanctl get backend  ● <volume-name>: 存储上资源(LUN/文件系统)的名称,可通过DeviceManager查看。</volume-name></backendname>
spec.csi.fsTy pe	指定主机文件系统类型。可选参数。支持类型为: • ext2 • ext3 • ext4 • xfs	否	-	如果不设置,默认为ext4。仅 当volumeMode配置为 "Filesystem"时生效。
spec.capacit y.storage	指定卷大小。	是	100Gi	请确保与存储上对应资源的容量保持一致。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。
spec.mount Options.nfsv ers	主机侧NFS挂载选 项。支持如下挂载选 项: nfsvers: 挂载NFS时 的协议版本。支持配 置的参数值为"3", "4","4.0"和 "4.1"。	否	-	在主机执行mount命令时-o 参数后的可选选项。列表格式。 指定NFS版本挂载时,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。

参数	说明	必选参数	默认 值	备注
spec.mount Options.acl	DPC命名空间支持ACL 功能。DPC客户端支 持POSIX ACL、NFSv4 ACL、NT ACL的鉴权 行为。	否	-	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。
spec.mount Options.acl onlyposix	DPC命名空间支持 POSIX ACL功能,DPC 客户端支持POSIX ACL的鉴权行为。 支持POSIX ACL的协议有: DPC、 NFSv3、HDFS。如使用NFSv4 ACL或NT ACL,会导致DPC客户端无法识别该类型的 ACL,从而导致该类型的ACL不会生效。	否	_	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。
spec.mount Options.cnfl ush	异步刷盘模式,即关 闭命名空间下的文件 时不会立即刷盘。	否	-	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。
spec.mount Options.cflu sh	同步刷盘模式,即关 闭命名空间下的文件 时立即刷盘。	否	-	默认使用同步刷盘模式。

## 前提条件

- 存储侧已经存在待创建PV所需要的存储资源,如LUN或者文件系统。如果存储资 源是文件系统,还需要创建文件系统的共享和客户端信息。
- 已参考表8-13,完成PV的配置文件。

## 操作步骤

步骤1 执行以下命令,基于准备好的yaml文件创建PV。

# kubectl create -f mypv.yaml persistentvolume/mypv created

**步骤2** 等待一段时间后,执行以下命令,查看已经创建的PV信息。当PV状态为"Available"时,表明PV创建成功。

# kubectl get pv
NAME CAPACITY ACCESS MODES RECLAIM POLICY STATUS CLAIM STORAGECLASS
REASON AGE
mypv 100Gi RWO Retain Available 4s

----结束

### 8.1.1.3.2 配置 PVC

当PV以静态卷供应的方式创建完成后,可以基于该PV创建PVC,从而供容器使用。如下示例是一个使用静态卷供应的PVC配置文件。

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

如上例所示,通过设置PVC配置文件中"volumeName"参数为之前通过静态卷供应创建的PV。具体参数见表8-14。

### 表 8-14 PVC 参数说明

参数	说明	必选参数	默认 值	备注
metadat a.name	自定义的PVC对象名 称。	是	-	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。

参数	说明	必选参数	默认 值	备注
spec.acc essMode s	指定卷访问模式。 RWO (ReadWriteOnce ):卷可以被一个节点以读写方式挂载。该模式也允许运行在同一节点上的多个Pod 访问卷。 ROX (ReadOnlyMany):卷可以被多个节点以只读方式挂载。 RWX (ReadWriteMany):卷可以被多个节点以读写方式挂载。 RWOP (ReadWriteOncePod):卷只能被单个Pod 以读写方式挂载。该特性需要Kubernetes 1.22 以		Rea dWri teO nce	<ul> <li>RWO/ROX/RWOP: 所有类型 卷均支持, RWOP需 Kubernetes 1.22版本以上支持。请参考10.5 开启 ReadWriteOncePod功能门章 节,检查您的Kubernetes集群是否开启该特性。</li> <li>RWX: volumeMode设置为 Block或者NFS类型的卷支持。</li> </ul>
spec.vol umeMo de	卷模式。	名	Files yste m	可选, 支持Filesystem或Block, 默认为Filesystem。该参数在创建 Pod时生效,其中Filesystem表示 在PVC上创建一个文件系统访问存 储, Block表示使用裸卷的方式访问存储。

参数	说明	必选参数	默认值	备注
spec.res ources.re quests.st orage	指定待创建卷大小。	是		指定待创建卷大小,格式为 ***Gi,单位为GiB。 PVC容量的规格取决于存储规格限制。以OceanStor Dorado 6.1.2/OceanStor Pacific 系列 8.1.0对接CentOS 7为例,当使用的是ext4文件系统时,容量限制见表8-5;当使用的是XFS文件系统时,容量限制见表8-6。对果使用的是NFS或者裸设备,容量需涉及各种,它容量不在规格范围内,文件系统规格限制导致的建PVC或中的类的,可能会统规格限制导致创建PVC或中的类的,是不是不是不是的。  在通过静态PV创建PVC对,是不是的类型,是不是的。  在通过静态PV创建PVC时,是不是的类型,是不是的类型,是不是的类型,是不是的类型,是不是的类型。  在通过静态PV创建PVC对量,是是一个一个方式。
spec.vol umeNa me	PV对象名称。	是	-	静态创建PVC时必选。

#### 操作步骤

步骤1 执行以下命令,基于已配置的yaml文件创建PVC。

# kubectl create -f mypvc.yaml persistentvolumeclaim/mypvc created

步骤2 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 RWO 12s

#### □ 说明

- 完成创建PVC操作后,如果长时间后(如一分钟后)PVC的状态是Pending,请参考**11.6 创 建PVC时,PVC的状态为Pending**。
- 建议每批次最多批量创建/删除100个PVC。

#### ----结束

在完成PVC创建后,就可以使用PVC来创建Pod。如下示例是一个简单的使用PVC示例,在该示例中,创建的Pod使用了刚刚创建的*mypvc*。

apiVersion: apps/v1 kind: Deployment

```
metadata:
name: nginx-deployment
 selector:
  matchLabels:
   app: nginx
 replicas: 2
 template:
  metadata:
   labels:
    app: nginx
  spec:
   containers:
    - image: nginx:alpine
    name: container-0
    volumeMounts:
     - mountPath: /tmp
      name: pvc-mypvc
   restartPolicy: Always
   volumes:
    - name: pvc-mypvc
    persistentVolumeClaim:
      claimName: mypvc
                                         # name of PVC
```

## 8.1.2 扩容 PVC

当容器使用的PVC容量不足时,需要对该PVC进行扩容操作。

#### 前提条件

- PVC已创建,所在的backend存在且支持扩容。
- 支持扩容的存储请参考表2-5和表2-9,支持扩容的Kubernetes版本请参考表2-3。
- huawei-csi-controller启用了csi-resizer服务。

```
# kubectl describe deploy huawei-csi-controller -n huawei-csi | grep csi-resizer csi-resizer:
Image: k8s.gcr.io/sig-storage/csi-resizer:v1.4.0
```

#### 操作步骤

**步骤1** 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 RW0
mysc 145m
```

步骤2 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

NAME PROVISIONER RECLAIMPOLICY VOLUMEBINDINGMODE

ALLOWVOLUMEEXPANSION AGE

mysc csi.huawei.com Delete Immediate true 172m
```

如果ALLOWVOLUMEEXPANSION的值为true,表示当前StorageClass已经支持扩容,请跳转至步骤<mark>步骤4</mark>。

**步骤3** 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4 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/ OceanStor Pacific系列 8.1.0对接CentOS 7为例,当使用的是ext4文件系统时,容量限制见表8-5;当使用的是XFS文件系统时,容量限制见表8-6。如果使用的是NFS或者裸设备,容量需满足使用的华为存储设备型号和版本所要求的规格约束。
- 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。
- 如果扩容的目标容量超过存储池容量导致扩容失败,请参考11.19 PVC扩容的目标容量超过 存储池容量导致扩容失败。

#### **步骤5** 执行以下命令,检查容量修改是否生效。

# kubectl get pvc

NAME STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS AGE mypvc Bound pvc-3383be36-537c-4cb1-8f32-a415fa6ba384 **120Gi** RWO mysc 24s

----结束

#### 8.1.3 克隆 PVC

本章节描述如何克隆PVC。

在克隆PVC时,需要指定数据源。如下示例是一个简单的克隆PVC示例,在该示例中,使用"mypvc"作为数据源,新创建了一个名叫"myclone"的PVC。

kind: PersistentVolumeClaim

apiVersion: v1

metadata:

name: myclone

spec:

storageClassName: mysc

dataSource:

name: mypvc

kind: PersistentVolumeClaim

volumeMode: Filesystem

accessModes:

- ReadWriteOnce

resources: requests:

storage: 2Gi

#### <u> 注意</u>

- 指定的storageClassName必须和dataSource中的源卷的StorageClass需一致。
- 克隆卷的容量必须不小于源卷容量,建议和源卷容量保持一致。

#### 前提条件

系统中已经存在源PVC,且源PVC所在的backend存在支持克隆。支持克隆的存储请参考表2-5和表2-9,支持克隆的Kubernetes版本请参考表2-3。

#### 操作步骤

步骤1 执行以下命令,基于克隆卷的配置文件创建PVC。

# kubectl create -f myclone.yaml persistentvolumeclaim/myclone created

----结束

## 8.1.4 从快照创建 PVC

本章节描述如何从快照创建PVC。

在创建这个PVC时,需要指定数据源。如下示例是一个简单的从快照创建PVC示例,在该示例中,使用快照"mysnapshot"作为数据源,新创建了一个名叫"myrestore"的PVC。

apiVersion: v1 kind: PersistentVolumeClaim metadata: name: myrestore spec:

storageClassName: mysc

dataSource:

name: mysnapshot kind: VolumeSnapshot

apiGroup: snapshot.storage.k8s.io

volumeMode: Filesystem

accessModes:
- ReadWriteOnce resources:

requests: storage: 100Gi

#### **注意**

- 指定的storageClassName必须和dataSource中的快照源卷的StorageClass需一致。
- 克隆卷的容量必须不小于快照容量,建议和快照容量保持一致。

#### 前提条件

系统中已经存在快照,且快照所在的backend存在支持克隆。支持快照创建PVC的存储请参考表2-5和表2-9,支持快照创建PVC的Kubernetes版本请参考表2-3。

#### 操作步骤

步骤1 执行以下命令,基于从快照创建卷的配置文件创建PVC。

# kubectl create -f myrestore.yaml persistentvolumeclaim/myrestore created

----结束

## 8.2 创建 VolumeSnapshot

在Kubernetes中,**卷快照(VolumeSnapshot**)是一个存储系统上卷的快照。 VolumeSnapshot能力为Kubernetes用户提供了一种标准的方式来在指定时间点复制卷 的内容,并且不需要创建全新的卷。 例如,这一功能使得数据库管理员能够在执行编辑或删除之类的修改之前对数据库执行备份。

本章将介绍如何使用华为CSI创建VolumeSnapshot。为了完成创建VolumeSnapshot,需要完成如下三步:

- 检查卷快照依赖组件信息
- 配置VolumeSnapshotClass
- 配置VolumeSnapshot

## 8.2.1 检查卷快照依赖组件信息

如果您需要在容器环境中使用卷快照以及卷快照关联的特性,请通过**3.5 检查卷快照依赖组件**检查您的环境是否部署了卷快照依赖组件以及卷快照api-versions信息。

## 8.2.2 配置 VolumeSnapshotClass

<mark>卷快照类(VolumeSnapshotClass</mark>)提供了一种在配置VolumeSnapshot时描述存储 "类"的方法。每个VolumeSnapshotClass都包含"driver"、"deletionPolicy"和 "parameters"字段, 在需要动态配置属于该类的VolumeSnapshot时使用。

VolumeSnapshotClass对象的名称很重要,是用户可以请求特定类的方式。 管理员在首次创建VolumeSnapshotClass对象时设置类的名称和其他参数, 对象一旦创建就无法更新。

华为CSI使用的VolumeSnapshotClass示例如下:

如果您的环境中api-versions支持v1,请使用以下示例:

apiVersion: snapshot.storage.k8s.io/v1

kind: VolumeSnapshotClass

metadata:

name: mysnapclass driver: csi.huawei.com deletionPolicy: Delete

如果您的环境中api-versions支持v1beta1,请使用以下示例:

apiVersion: snapshot.storage.k8s.io/v1beta1

kind: VolumeSnapshotClass

metadata:

name: mysnapclass driver: csi.huawei.com deletionPolicy: Delete

如果您的环境中api-versions同时支持v1和v1beta1,我们推荐您使用v1版本。

实际参数可以参考<mark>表8-15</mark>中的说明修改。由于当前华为CSI还不支持在 VolumeSnapshotClass中设置自定义参数(parameters),因此建议只创建一个 VolumeSnapshotClass,供所有快照使用。

表 8-15 VolumeSnapshotClass 参数说明

参数	说明	备注
metadata.n ame	自定义的 VolumeSnapshotCla ss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。
driver	driver标识。必填参 数。	该字段需要指定为安装华为CSI时设置的驱动名。默认的驱动名为"csi.huawei.com"。

参数	说明	备注
deletionPoli cy	快照删除策略。必填 参数。可选值为: • Delete • Retain	<ul> <li>如果删除策略是 Delete,那么存储设备上的快照会和VolumeSnapshotContent对象一起删除。</li> <li>如果删除策略是Retain,那么存储设备上的快照和VolumeSnapshotContent对象都会被保留。</li> </ul>

#### 前提条件

华为CSI支持快照且运行所依赖的卷快照组件CRD已经安装。具体CRD信息请参考3.5 检查卷快照依赖组件章节说明,支持创建VolumeSnapshot的Kubernetes版本请参考表2-3。

#### 操作步骤

**步骤1** 执行以下命令,使用已经创建的VolumeSnapshotClass配置文件创建 VolumeSnapshotClass。

# kubectl create -f mysnapclass.yaml volumesnapshotclass.snapshot.storage.k8s.io/mysnapclass created

步骤2 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

## 8.2.3 配置 VolumeSnapshot

VolumeSnapshot可以通过两种方式进行制备: 预制备或动态制备。华为CSI当前仅支持动态制备。本章节将说明如何使用华为CSI动态制备VolumeSnapshot。

VolumeSnapshot的配置文件示例如下:

如果您的环境中api-versions支持v1,请使用以下示例:

apiVersion: snapshot.storage.k8s.io/v1

kind: VolumeSnapshot

metadata:

name: mysnapshot

spec:

volumeSnapshotClassName: mysnapclass

source:

persistentVolumeClaimName: mypvc

如果您的环境中api-versions支持v1beta1,请使用以下示例:

apiVersion: snapshot.storage.k8s.io/v1beta1

kind: VolumeSnapshot

metadata:

name: mysnapshot

spec:

volumeSnapshotClassName: mysnapclass

source:

persistentVolumeClaimName: mypvc

VolumeSnapshot中api-versions信息,请和创建VolumeSnapshotClass使用的版本保持一致。

实际参数可以参考表8-16中的说明修改。

表 8-16 VolumeSnapshot 参数说明

参数	说明	备注
metadata.name	自定义的 VolumeSnapshot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。
spec.volumeSnapshotCl assName	VolumeSnapshotCl ass对象名称。	
spec.source.persistentVo lumeClaimName	源PVC对象名称。	快照源PVC对应的名称

#### 前提条件

- 源PVC存在,且PVC所在的backend存在支持创建VolumeSnapshot。支持创建VolumeSnapshot的存储请参考表2-5和表2-9,支持创建VolumeSnapshot的Kubernetes版本请参考表2-3。
- 华为CSI运行所依赖的卷快照组件CRD已经安装。具体信息请参考3.5 检查卷快照 依赖组件章节说明。
- 系统中已经存在使用华为CSI的VolumeSnapshotClass。

#### 操作步骤

步骤1 执行以下命令,使用已经创建的VolumeSnapshot配置文件创建VolumeSnapshot。

# kubectl create -f mysnapshot.yaml volumesnapshot.snapshot.storage.k8s.io/mysnapshot created

步骤2 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

# 9高级特性

- 9.1 配置ALUA特性
- 9.2 配置存储拓扑感知

## 9.1 配置 ALUA 特性

ALUA(Asymmetric Logical Unit Access,非对称逻辑单元访问),是一种多目标器端口访问模型。在多路径状态下,ALUA标准提供了一种将卷的Active/Passive模型呈现给主机的方式。同时还提供了端口的可访问状态切换接口,可用来实现卷工作控制器切换等。例如,卷在一个控制器故障时,可以将该控制器的端口置为Unavailable,支持ALUA的主机多路径软件收到该状态后,会将I/O切换到另一端控制器。

## 9.1.1 通过 Helm 配置 ALUA 特性

#### 9.1.1.1 配置华为企业存储后端的 ALUA 参数

华为企业存储针对ALUA的配置请参考产品对应的主机连通性指南文档说明。

针对不同的操作系统,ALUA配置可能有所不同。进入**华为技术支持**,在搜索输入框中输入"主机连通性指南",单击搜索。在搜索结果中,选择对应操作系统的主机连通性指南。结合实际需要根据指南的说明进行ALUA配置。华为CSI将在华为存储上对该主机的启动器应用您设置的配置项。

#### □ 说明

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。

## OceanStor V3/V5 系列和 OceanStor Dorado V3 系列存储后端的 ALUA 参数

华为CSI支持的OceanStor V3/V5系列和OceanStor Dorado V3系列存储的ALUA参数见表9-1。

表 9-1 华为 CSI 支持的 OceanStor V3/V5 系列和 OceanStor Dorado V3 系列存储的 ALUA 参数说明

参数名	参数描述	备注
HostName	主机名规则。必填,可 使用正则表达式。	主机名通常使用 cat /etc/hostname 可获取。支持正则表达式方式匹配,如当HostName= "*"时,该条配置对任意主机名的主机生效。可参考《正则表达式》。 当计算节点的主机名可已匹配多条ALUA配置选项,会根据匹配的精确度进行排序,使用第一条ALUA配置选项。排序规
MULTIPATHTY PE	多路径类型。必填,取值为:  ① 3 不使用第三方多路径	则见ALUA配置项匹配主机名的规则。 
	• 1:使用第三方多路 径	
FAILOVERMO DE	启动器的切换模式。条件必选,取值为:	当使用第三方多路径时该参数才需要指 定。请参考连通性指南的说明,配置启 动器的切换模式。
SPECIALMODE TYPE	启动器的特殊模式类型。条件必选,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"特殊模式 ALUA"时该参数才需要指定。请参考连 通性指南的说明,配置启动器的特殊模 式类型。
PATHTYPE	启动器的路径类型。条件必选,取值为:  ①:优选路径  1:非优选路径	当使用第三方多路径时该参数才需要指 定。请参考连通性指南的说明,配置启 动器的路径类型。

以OceanStor 18500 V5存储对接Red Hat操作系统为例,主机连通性指南见《华为SAN存储在Red Hat系统下的主机连通性指南》。

如下ALUA设置示例,是非双活存储场景下,OceanStor 18500 V5存储的Red Hat操作系统的连通性指南的推荐设置。本例中假设Kubernetes集群中计算节点"myhost01"的操作系统是RHEL 5.x,其他计算节点操作系统均为RHEL 7.x。根据推荐,RHEL 5.x的切换模式应该为"不使用ALUA",RHEL 7.x的切换模式应该为"通用ALUA"。

storage: oceanstor-san name: oceanstor-iscsi-155 - https://192.168.129.155:8088 - https://192.168.129.156:8088 - StoragePool001 parameters: protocol: iscsi portals: - 192.168.128.120 - 192.168.128.121 ALUA: ^myhost01\$: **MULTIPATHTYPE: 1 FAILOVERMODE: 2** PATHTYPE: 0 **MULTIPATHTYPE: 1 FAILOVERMODE: 1** PATHTYPE: 0

#### OceanStor V6 和 OceanStor Dorado V6 系列存储后端的 ALUA 参数

华为CSI支持的OceanStor V6和OceanStor Dorado V6系列存储的ALUA参数见表9-2。

#### 山 说明

OceanStor V6和OceanStor Dorado V6系列存储在默认情况下启动器主机访问模式即为"均衡模式",因此不建议对OceanStor V6和OceanStor Dorado V6系列存储配置ALUA参数。

表 9-2 OceanStor V6 和 OceanStor Dorado V6 系列存储的 ALUA 参数说明

参数名	参数描述	备注
HostName	主机名规则。必 填,可使用正则表 达式。	主机名通常使用 cat /etc/hostname 可获取。支持正则表达式方式匹配,如当HostName="*"时,该条配置对任意主机名的主机生效。可参考《正则表达式》。 当计算节点的主机名可已匹配多条ALUA配置选项,会根据匹配的精确度进行排序,使用第一条ALUA配置选项。排序规则见ALUA配置项匹配主机名的规则。
accessMode	主机访问模式。必 填,取值为: • 0: 均衡模式 • 1: 非对称模式	非双活场景下建议使用均衡模式。当前 华为CSI未支持SAN双活场景,请谨慎使 用非对称模式。
hyperMetroPathO ptimized	双活场景下,主机 在当前阵列的路径 是否优选。取值 为: • 1: 是 • 0: 否	当主机访问模式设置为非对称模式时, 才需要配置该参数。 当前华为CSI未支持SAN双活场景,请谨 慎使用非对称模式。

以OceanStor Dorado 18500 V6存储对接Red Hat操作系统为例,主机连通性指南见《OceanStor Dorado 6.x & OceanStor 6.x在Red Hat下的主机连通性指南》。

如下ALUA设置示例,是非双活存储场景下,OceanStor Dorado 18500 V6存储的Red Hat操作系统的连通性指南的推荐设置。

#### ALUA 配置项匹配主机名的规则

● 如果设置的主机名规则精确匹配的业务节点主机名,则使用该主机名规则对应的 ALUA配置项。

如配置项1中主机名规则为"\*",配置项2中的主机名规则为"^myhost01\$"。 当计算节点的主机名是"myhost01"时,精确匹配配置项2,华为CSI将使用配置 项2中的配置应用到存储侧。

● 如果设置的主机名规则无法精确匹配的业务节点主机名,则直接使用正则匹配到的第一条ALUA配置项。

如配置项1中主机名规则为"myhost0[0-9]",配置项2中的主机名规则为"myhost0[5-9]",配置项1的优先级高于配置项2。当计算节点的主机名是"myhost06"时,两个配置项均可以匹配,此时华为CSI将使用配置项1中的配置应用到存储侧。

#### 9.1.1.2 配置分布式存储后端的 ALUA 参数

华为分布式存储针对ALUA的配置请参考产品对应的主机连通性指南文档说明。

针对不同的操作系统,ALUA配置可能有所不同。进入**华为技术支持**,在搜索输入框中输入"主机连通性指南",单击搜索。在搜索结果中,选择对应操作系统的主机连通性指南。结合实际需要根据指南的说明进行ALUA配置。华为CSI将在华为存储上对该主机的启动器应用您设置的配置项。

#### □ 说明

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。

分布式存储非双活场景,存储系统自身为Active/Active模式,选择"启用ALUA"没有实际意义,建议选择存储默认的"禁用ALUA"。因此不建议对分布式存储配置ALUA参数。

华为CSI支持的分布式存储的ALUA参数见表9-3。

表 9-3 分布式存储 ALUA 参数说明

参数名	参数描述	备注
HostName	HostName的值为worker节 点的主机名,如 HostName1、 HostName2。	主机名通常使用 cat /etc/hostname 可获取。支持正则表达式方式匹配,如当HostName= "*"时,该条配置对任意主机名的主机生效。可参考《正则表达式》。 当计算节点的主机名可已匹配多条ALUA配置选项,会根据匹配的精确度进行排序,使用第一条ALUA配置选项。排序规则见ALUA配置项匹配主机名的规则。
switchoverMode	切换模式。必选,取值为: ● Disable_alua:禁用 ALUA ● Enable_alua:启用 ALUA	非双活场景,存储系统自身为Active/Active模式,选择"启用ALUA"没有实际意义,建议选择"禁用ALUA"。当前华为CSI未支持SAN双活场景,请谨慎启用ALUA。
pathType	路径类型。条件必选,取值为:  optimal_path: 优选路径  non_optimal_path: 非优选路径	切换模式为启动ALUA时需要 设置该选项。

#### ALUA 配置项匹配主机名的规则

如果设置的主机名规则精确匹配的业务节点主机名,则使用该主机名规则对应的 ALUA配置项。

如配置项1中主机名规则为 "\*",配置项2中的主机名规则为 "^myhost01\$"。 当计算节点的主机名是 "myhost01"时,精确匹配配置项2,华为CSI将使用配置 项2中的配置应用到存储侧。

● 如果设置的主机名规则无法精确匹配的业务节点主机名,则直接使用正则匹配到的第一条ALUA配置项。

如配置项1中主机名规则为"myhost0[0-9]",配置项2中的主机名规则为 "myhost0[5-9]",配置项1的优先级高于配置项2。当计算节点的主机名是 "myhost06"时,两个配置项均可以匹配,此时华为CSI将使用配置项1中的配置 应用到存储侧。

## 9.2 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
node02 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9-4。

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

#### 表 9-4 参数说明

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

## 9.2.1 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,组件包路径请参考表3-1。
- **步骤3** 进入后端服务配置目录/examples/backend/下,备份backend.yaml文件 # cp backend.yaml backend.yaml.bak
- **步骤4** 执行**vi** backend.yaml命令打开文件,按需求配置拓扑感知,示例如下所示。修改完成后,按**Esc**,并输入:**wq!**,保存修改。

步骤5 执行以下命令删除待修改存储后端,其中"dorado-iscsi-155"为存储后端名称。

# oceanctl delete backend dorado-iscsi-155 -n huawei-csi backend/dorado-iscsi-155 deleted

步骤6 执行以下命令创建存储后端

# oceanctl create backend -f ../examples/backend/backend.yaml
Please enter this backend user name:admin
Please enter this backend password:
backend/dorado-iscsi-155 created

**步骤7** 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9-5</mark>。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 9-5 参数说明

参数名	参数描述	备注
volumeBindin gMode	PersistentVolume 绑定方式,用于控 制何时进行 PersistentVolume 动态资源调配和绑 定。	可配置"WaitForFirstConsumer"或 "Immediate" "WaitForFirstConsumer":表示延迟 PersistentVolume的绑定和调配,直到创建 使用PVC的Pod。 "Immediate":表示创建PVC后,立即发 生PersistentVolume绑定和调配。
allowedTopol ogies.matchLa belExpression s	"key":可支持配置 "topology.kubernetes.io/zone", "topology.kubernetes.io/region", topology.kubernetes.io/ protocol. <i><protocol></protocol></i> , 其中 <i><protocol></protocol></i> 为协 议类型,例如: iscsi, fc, nfs等。	
	配置时需要同时按 照固定格式配置 "key"和 "value".	"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> 置的拓扑标签保持一致。 "key"如果是topology.kubernetes.io/ protocol. resi.huawei.com

步骤8 执行以下命令,基于该yaml文件创建StorageClass。

# kubectl create -f StorgeClass.yaml

步骤9 使用该StorageClass创建具有拓扑能力的PVC,详细操作请参考8.1.1.1.2 配置PVC。

-----结束

# 10 常用操作

- 10.1 日志收集
- 10.2 更新huawei-csi-controller服务
- 10.3 更新huawei-csi-node服务
- 10.4 修改日志输出模式
- 10.5 开启ReadWriteOncePod功能门
- 10.6 配置非root用户访问Kubernetes集群

## 10.1 日志收集

#### 前置检查

**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集群内有 oceanctl工具的节点。

步骤2 执行oceanctl version命令,显示版本号为v4.2.0。

```
$ oceanctl version
+------+
| OCEANCTL VERSION |
+------+
| v4.2.0 |
```

#### 步骤3 执行oceanctl --help命令,返回信息如下。

```
$ oceanctl --help
A CLI tool for Ocean Storage in Kubernetes

Usage:
    oceanctl [command]

Available Commands:
    collect collect messages in Kubernetes
    create Create a resource to Ocean Storage in Kubernetes
    delete Delete one or more resources from Ocean Storage in Kubernetes
    get Get one or more resources from Ocean Storage in Kubernetes
    help Help about any command
    update Update a resource for Ocean Storage in Kubernetes
    Print the version of oceanctl
```

Flags:

-h, --help help for oceanctl

Use "oceanctl [command] --help" for more information about a command.

步骤4 执行kubectl get deploy -n *\${NAMESPACE}*命令,检查Pod是否正常启动,其中,\$ {NAMESPACE}为CSI安装的命名空间,以huawei-csi为例。

```
# kubectl get deploy -n huawei-csi
NAME READY UP-TO-DATE AVAILABLE AGE
huawei-csi-controller 1/1 1 1 21h
```

----结束

#### 使用 oceanctl 收集 CSI 命名空间下所有日志

- 步骤1 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录<mark>前置检查</mark>章节中检查的节点。
- **步骤2** 执行**oceanctl collect logs -n <namespace> -a**命令,收集集群内所有CSI容器所在节点的CSI日志。

步骤3 检查/tmp目录下生成的日志压缩包,可以使用unzip \${zip\_name} -d collect\_logs解 压日志压缩包,其中\${zip\_name}为压缩包的名字。

# date Wed Sep 20 02:49:24 EDT 2023

# ls

huawei-csi-2023-09-20-02:48:22-all.zip

----结束

#### 使用 oceanctl 收集 CSI 单个节点日志

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录**前置检查**章节中检查的节点。
- 步骤2 执行oceanctl collect logs -n <namespace> -N <nodeName>命令,收集集群内所有CSI容器所在节点的CSI日志。

步骤3 检查/tmp目录下生成的日志压缩包,可以使用unzip \${zip\_name} -d collect\_logs解 压日志压缩包,其中\${zip\_name}为压缩包的名字。

# date

Thu Sep 21 04:08:47 EDT 2023

# ls

huawei-csi-2023-09-21-04:05:15-node-1.zip

----结束

## 10.2 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。

#### 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** 备份安装CSI时使用的values.yaml文件,可通过**helm get values** *helm-huawei-csi* **-n** *huawei-csi* **-a** > values.yaml.bak 命令备份,其中,helm-huawei-csi为安装时自定义的Helm Chart名称,huawei-csi为安装时自定义的Helm Chart命名空间。

# helm get values helm-huawei-csi -n huawei-csi -a > values.yaml.bak

步骤3 进入/helm/esdk 目录,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。其中组件包路径请参考表3-1。

```
kubernetes:
 namespace: huawei-csi
images:
 # Images provided by Huawei
 huaweiCSIService: huawei-csi:4.2.0
 storageBackendSidecar: storage-backend-sidecar:4.2.0
 storageBackendController: storage-backend-controller:4.2.0
 # CSI-related sidecar images provided by the Kubernetes community.
 # These must match the appropriate Kubernetes version.
 sidecar:
  attacher: k8s.gcr.io/sig-storage/csi-attacher:v3.4.0
  provisioner: k8s.gcr.io/sig-storage/csi-provisioner:v3.0.0
  resizer: k8s.gcr.io/sig-storage/csi-resizer:v1.4.0
  registrar: k8s.gcr.io/sig-storage/csi-node-driver-registrar:v2.3.0
  livenessProbe: k8s.gcr.io/sig-storage/livenessprobe:v2.5.0
  snapshotter: 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:4.2.0
# The CSI driver parameter configuration
csiDriver:
 # Driver name, it is strongly recommended not to modify this parameter
 # The CCE platform needs to modify this parameter, e.g. csi.oceanstor.com
 driverName: csi.huawei.com
 # Endpoint, it is strongly recommended not to modify this parameter
 endpoint: /csi/csi.sock
 # DR Endpoint, it is strongly recommended not to modify this parameter
 drEndpoint: /csi/dr-csi.sock
 # Maximum number of concurrent disk scans or detaches, support 1~10
 connectorThreads: 4
 # Flag to enable or disable volume multipath access, support [true, false]
 volumeUseMultipath: true
 # Multipath software used by fc/iscsi. support [DM-multipath, HW-UltraPath, HW-UltraPath-NVMe]
 scsiMultipathType: DM-multipath
 # Multipath software used by roce/fc-nvme. only support [HW-UltraPath-NVMe]
 nvmeMultipathType: HW-UltraPath-NVMe
 # Timeout interval for waiting for multipath aggregation when DM-multipath is used on the host. support
1~600
 scanVolumeTimeout: 3
 # Interval for updating backend capabilities, support 60~600
 backendUpdateInterval: 60
 # Huawei-csi-controller log configuration
 controllerLogging:
  # Log record type, support [file, console]
  module: file
  # Log Level, support [debug, info, warning, error, fatal]
  level: info
  # Directory for storing logs
```

```
fileDir: /var/log/huawei
  # Size of a single log file
  fileSize: 20M
  # Maximum number of log files that can be backed up.
  maxBackups: 9
 # Huawei-csi-node log configuration
 nodeLogging:
  # Log record type, support [file, console]
  module: file
  # Log Level, support [debug, info, warning, error, fatal]
  level: info
  # Directory for storing logs
  fileDir: /var/log/huawei
  # Size of a single log file
  fileSize: 20M
  # Maximum number of log files that can be backed up.
  maxBackups: 9
controller:
 # controllerCount: Define the number of huawei-csi controller
 # Allowed values: n, where n > 0
 # Default value: 1
 # Recommended value: 2
 controllerCount: 1
 # volumeNamePrefix: Define a prefix that is prepended to volumes.
 # THIS MUST BE ALL LOWER CASE.
 # Default value: pvc
 # Examples: "volumes", "vol"
 volumeNamePrefix: pvc
 # Port used by the webhook service. The default port is 4433.
 # You can change the port to another port that is not occupied.
 webhookPort: 4433
  # enabled: Enable/Disable volume snapshot feature
  # If the Kubernetes version is lower than 1.17, set this parameter to false.
  # true: enable volume snapshot feature(install snapshotter sidecar)
  # false: disable volume snapshot feature(do not install snapshotter sidecar)
  # Default value: None
  enabled: true
 resizer.
  # enabled: Enable/Disable volume expansion feature
  # Allowed values:
  # true: enable volume expansion feature(install resizer sidecar)
  # false: disable volume snapshot feature(do not install resizer sidecar)
  # Default value: None
  enabled: true
 # nodeSelector: Define node selection constraints for controller pods.
 # For the pod to be eligible to run on a node, the node must have each
 # of the indicated key-value pairs as labels.
 # Leave as blank to consider all nodes
 # Allowed values: map of key-value pairs
 # Default value: None
 nodeSelector:
 # Uncomment if nodes you wish to use have the node-role.kubernetes.io/master taint
 # node-role.kubernetes.io/master: "'
 # Uncomment if nodes you wish to use have the node-role.kubernetes.io/control-plane taint
 # node-role.kubernetes.io/control-plane: "
 # tolerations: Define tolerations that would be applied to controller deployment
 # Leave as blank to install controller on worker nodes
 # Allowed values: map of key-value pairs
 # Default value: None
 tolerations:
```

# Uncomment if nodes you wish to use have the node-role.kubernetes.io/master taint
# - key: "node-role.kubernetes.io/master"
# Uncomment if nodes you wish to use have the node-role.kubernetes.io/control-plane taint
# - key: "node-role.kubernetes.io/control-plane"
operator: "Exists"
# effect: "NoSchedule"

步骤4 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

----结束

kubernetes:

TEST SUITE: None

## 10.3 更新 huawei-csi-node 服务

当您需要更新huawei-csi-node服务时,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 备份安装CSI时使用的values.yaml文件,可通过**helm get values** *helm-huawei-csi* **-n** *huawei-csi* **-a** > values.yaml.bak 命令备份,其中,helm-huawei-csi为安装时自定义的Helm Chart名称,huawei-csi为安装时自定义的Helm Chart命名空间。

# helm get values helm-huawei-csi -n huawei-csi -a > values.yaml.bak

步骤3 进入/helm/esdk 目录,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。其中组件包路径请参考表3-1。

```
namespace: huawei-csi
images:
 # Images provided by Huawei
 huaweiCSIService: huawei-csi:4.2.0
 storageBackendSidecar: storage-backend-sidecar:4.2.0
 storageBackendController: storage-backend-controller:4.2.0
 # CSI-related sidecar images provided by the Kubernetes community.
 # These must match the appropriate Kubernetes version.
  attacher: k8s.gcr.io/sig-storage/csi-attacher:v3.4.0
  provisioner: k8s.gcr.io/sig-storage/csi-provisioner:v3.0.0
  resizer: k8s.gcr.io/sig-storage/csi-resizer:v1.4.0
  registrar: k8s.gcr.io/sig-storage/csi-node-driver-registrar:v2.3.0
  livenessProbe: k8s.gcr.io/sig-storage/livenessprobe:v2.5.0
  snapshotter: 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:4.2.0

```
# The CSI driver parameter configuration
 # Driver name, it is strongly recommended not to modify this parameter
 # The CCE platform needs to modify this parameter, e.g. csi.oceanstor.com
 driverName: csi.huawei.com
 # Endpoint, it is strongly recommended not to modify this parameter
 endpoint: /csi/csi.sock
 # DR Endpoint, it is strongly recommended not to modify this parameter
 drEndpoint: /csi/dr-csi.sock
 # Maximum number of concurrent disk scans or detaches, support 1~10
 connectorThreads: 4
 # Flag to enable or disable volume multipath access, support [true, false]
 volumeUseMultipath: true
 # Multipath software used by fc/iscsi. support [DM-multipath, HW-UltraPath, HW-UltraPath-NVMe]
 scsiMultipathType: DM-multipath
 # Multipath software used by roce/fc-nvme. only support [HW-UltraPath-NVMe]
 nvmeMultipathType: HW-UltraPath-NVMe
 # Timeout interval for waiting for multipath aggregation when DM-multipath is used on the host. support
1~600
 scanVolumeTimeout: 3
 # Interval for updating backend capabilities. support 60~600
 backendUpdateInterval: 60
 # Huawei-csi-controller log configuration
 controllerLogging:
  # Log record type, support [file, console]
  module: file
  # Log Level, support [debug, info, warning, error, fatal]
  level: info
  # Directory for storing logs
  fileDir: /var/log/huawei
  # Size of a single log file
  fileSize: 20M
  # Maximum number of log files that can be backed up.
  maxBackups: 9
 # Huawei-csi-node log configuration
 nodeLogging:
  # Log record type, support [file, console]
  module: file
  # Log Level, support [debug, info, warning, error, fatal]
  level: info
  # Directory for storing logs
  fileDir: /var/log/huawei
  # Size of a single log file
  fileSize: 20M
  # Maximum number of log files that can be backed up.
  maxBackups: 9
node:
 # maxVolumesPerNode: Defines the maximum number of volumes that can be used by a node.
 # Uncomment if you want to limit the number of volumes that can be used in a Node.
 # maxVolumesPerNode: 100
 # nodeSelector: Define node selection constraints for node pods.
 # For the pod to be eligible to run on a node, the node must have each
 # of the indicated key-value pairs as labels.
 # Leave as blank to consider all nodes
 # Allowed values: map of key-value pairs
 # Default value: None
 nodeSelector:
 # Uncomment if nodes you wish to use have the node-role.kubernetes.io/master taint
 # node-role.kubernetes.io/master:
 # Uncomment if nodes you wish to use have the node-role.kubernetes.io/control-plane taint
 # node-role.kubernetes.io/control-plane: ""
 # tolerations: Define tolerations that would be applied to node daemonset
 # Add/Remove tolerations as per requirement
 # Leave as blank if you wish to not apply any tolerations
```

```
# Allowed values: map of key-value pairs
# Default value: None
tolerations:
 - key: "node.kubernetes.io/memory-pressure"
  operator: "Exists"
  effect: "NoExecute"
 - key: "node.kubernetes.io/disk-pressure"
  operator: "Exists"
  effect: "NoExecute"
 - key: "node.kubernetes.io/network-unavailable"
  operator: "Exists'
  effect: "NoExecute"
 - key: "node-role.kubernetes.io/control-plane"
  operator: "Exists"
   effect: "NoSchedule"
  - key: "node-role.kubernetes.io/master"
   operator: "Exists'
   effect: "NoSchedule"
```

步骤4 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

## 10.4 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.

#### 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<mark>表3-1</mark>。
- 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
csiDriver:

# Driver name, it is strongly recommended not to modify this parameter

# The CCE platform needs to modify this parameter, e.g. csi.oceanstor.com
driverName: csi.huawei.com

# Endpoint, it is strongly recommended not to modify this parameter
endpoint: /csi/csi.sock
```

# DR Endpoint, it is strongly recommended not to modify this parameter

```
drEndpoint: /csi/dr-csi.sock
 # Maximum number of concurrent disk scans or detaches, support 1~10
 connectorThreads: 4
 # Flag to enable or disable volume multipath access, support [true, false]
 volumeUseMultipath: true
 # Multipath software used by fc/iscsi. support [DM-multipath, HW-UltraPath, HW-UltraPath-NVMe]
 scsiMultipathType: DM-multipath
 # Multipath software used by roce/fc-nvme. only support [HW-UltraPath-NVMe]
 nvmeMultipathType: HW-UltraPath-NVMe
 # Timeout interval for waiting for multipath aggregation when DM-multipath is used on the host. support
1~600
 scanVolumeTimeout: 3
 # Interval for updating backend capabilities. support 60~600
 backendUpdateInterval: 60
 # Huawei-csi-controller log configuration
 controllerLogging:
  # Log record type, support [file, console]
  module: file
  # Log Level, support [debug, info, warning, error, fatal]
  level: info
  # Directory for storing logs
  fileDir: /var/log/huawei
  # Size of a single log file
  fileSize: 20M
  # Maximum number of log files that can be backed up.
  maxBackups: 9
 # Huawei-csi-node log configuration
 nodeLogging:
  # Log record type, support [file, console]
  module: file
  # Log Level, support [debug, info, warning, error, fatal]
  level: info
  # Directory for storing logs
  fileDir: /var/log/huawei
  # Size of a single log file
  fileSize: 20M
  # Maximum number of log files that can be backed up.
  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

## 10.5 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。

#### □ 说明

CCE / CCE Agile平台暂时不支持开启ReadWriteOncePod功能门

#### 操作步骤

步骤1 为kube-apiserver启用ReadWriteOncePod功能门。

- 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-apiserver.yaml命令,按I或Insert进 入编辑状态,为 kube-apiserver容器添加参数--featuregates=ReadWriteOncePod=true。修改完成后,按Esc,并输入:wq!,保存修 改。

spec:

containers:

- command:
- kube-aniserver
- --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。

- 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--featuregates=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。

- 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 1. 任意worker节点。
- 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featureGates字段添加ReadWriteOncePod: true,如

果没有featureGates字段请一并添加。修改完成后,按**Esc**,并输入**:wq!**,保存修改。

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。

3. 在配置完成后,执行systemctl restart kubelet命令重启kubelet。

----结束

## 10.6 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文件。

\$ 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

# **11** FAQ

- 11.1 如何查看华为CSI日志
- 11.2 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
- 11.3 启动huawei-csi-node失败,提示错误为: "/var/lib/iscsi is not a directory"
- 11.4 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
- 11.5 启动huawei-csi服务时,服务启动异常, 状态显示InvalidImageName
- 11.6 创建PVC时, PVC的状态为Pending
- 11.7 删除PVC前,PVC的状态为Pending
- 11.8 创建Pod时,Pod的状态为ContainerCreating
- 11.9 创建Pod时,Pod的状态长时间处于ContainerCreating状态
- 11.10 创建Pod失败,日志显示执行mount命令超时
- 11.11 创建Pod失败,日志显示执行mount命令失败
- 11.12 创建Pod失败或重启kubelet后,日志显示挂载点已存在
- 11.13 创建Pod失败,Events日志显示"publishInfo doesn't exist"
- 11.14 如何下载容器镜像到本地
- 11.15 如何获取CSI版本信息
- 11.16 对接Tanzu Kubernetes集群常见问题及解决方法
- 11.17 使用Tanzu Kubernetes集群时常见问题及解决方法
- 11.18 通用临时卷扩容失败
- 11.19 PVC扩容的目标容量超过存储池容量导致扩容失败
- 11.20 使用oceanctl工具管理后端时调用webhook失败
- 11.21 存储侧更新密码后账户被锁定

## 11.1 如何查看华为 CSI 日志

#### 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 9/9 **Running** 0 14s <host1-ip> <host1-name> <none> <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-g6f7z 3/3 **Running** 0 14s <host2-ip> <host2name> <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

## 11.2 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。

# Ismod | grep iscsi | grep iscsi\_tcp
iscsi\_tcp 18333 6
libiscsi\_tcp 25146 1 iscsi\_tcp
libiscsi 57233 2 libiscsi\_tcp,iscsi\_tcp
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

## 11.3 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 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,组件包路径请参考表3-1。

步骤3 进入下一级目录templates,找到huawei-csi-node.yaml文件。

# cd /templates

**步骤4** 执行以下命令,将huawei-csi-node.yaml > volumes > iscsi-dir > hostPath中"path"设置为"/var/lib/iscsi",然后保存并退出文件。

# vi huawei-csi-node.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。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

## 11.4 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

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
- 命令卡死 => 步骤1.3
- **其他** => 联系技术支持。
- 在另一窗口再次登录该节点。
  - 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

将该pid杀死。 # kill -9 *pid* 

记录残留的dm-xx设备以及关联磁盘号(见<mark>步骤1.1</mark> ),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。

清理残留的SCSI磁盘,根据<mark>步骤1</mark>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。

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/sdi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 sdc 8:32 active ready running

`- 38:0:0:2 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/
block/sdi
ls -l /dev/disk/by-id/
total 0
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 -> ../../
lrwxrwxrwx 1 root root 10 Aug 11 19:58 dm-uuid-mpath-3618cf24100f8f457315a764c000001f6 -> ../../
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 -> ../../sdb
lrwxrwxrwx 1 root root 9 Aug 2 14:49 scsi-36888603000020aff44cc0d060c987f1 -> ../../sdc
lrwxrwxrwx 1 root root 9 Aug 11 19:57 wwn-0x618cf24100f8f457014a764c000001f6 -> ../../sdd
lrwxrwxrwx 1 root root 9 Aug 11 19:57 wwn-0x618cf24100f8f45712345678000103e8 -> ../../sdi
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

## 11.5 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 6/9 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:4.2.0
...
```

#### 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,修改华为CSI镜像huawei-csi:4.2.0。

containers:

- name: huawei-csi-driver image: huawei-csi:4.2.0

**步骤3** 执行以下命令,修改huawei-csi-controller服务的配置文件。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。

# 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,修改华为CSI镜像huawei-csi:4.2.0。

containers:

- name: huawei-csi-driver image: huawei-csi:4.2.0

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 9/9 Running 0 2m29s huawei-csi huawei-csi-node-7fxh6 3/3 Running 0 12m

----结束

## 11.6 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.jo " <i>mysc</i> " not found					

- a. 删除PVC。
- b. 创建StorageClass, 可参考8.1.1.1.1 配置StorageClass。
- c. 创建新的PVC,可参考**8.1.1.1.2 配置PVC**。
- 如果由原因2导致PVC处于Pending状态,执行以下步骤。

Events:			
Type	Reason	Age	
From		-	Message
Normal	l Provisioning	63s (x3 over 6	4s) csi.huawei.com_huawei-csi-controller-b59577886-
aazm8 5	85336/2-88/6-	1c7f_02c3_668a7h32	7515 External provisioner is provisioning volume for

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-b5957788

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。
- b. 删除StorageClass。
- c. 根据Events信息修改StorageClass.yaml文件。
- d. 创建StorageClass,详细请参考8.1.1.1.1 配置StorageClass。
- e. 创建新的PVC,详情请参考**8.1.1.1.2 配置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:						
Type	Reason	Age				
From				Message		
Norma	al Provisioning	63s (x4 over	58s) csi.hı	uawei.com	_huawei-csi-controll	er-b59577886-
qqzm8_	58533e4a-884c-4	c7f-92c3-6e8a7b3	27515 Ext	ternal prov	visioner is provisionin	ig volume for
claim "d	lefault/mypvc"			•	•	
Warnir	ng ProvisioningFa	ailed 62s (x4 ove	r 68s) csi	.huawei.co	om_huawei-csi-contro	oller-b59577886-
aazmo	E0E22012 0016 1	c7f 02c2 6c0a7h2	27515 fai	lad to pro	vicion volumo with C	toragoClass

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:
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)

- 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

## 11.7 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:	35,500			
Type	Reason	Age	From	Message

Warning ProvisioningFailed 0s (x15 over 3m24s) persistentvolume-controller storageclass.storage.k8s.io "mysc" 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
---- ---- ---- ----

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

● 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。

Events:
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

## 11.8 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**11.1 如何查看华为CSI日志**),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>

步骤3 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

## 11.9 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**11.1 如何查看华为CSI日志**),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。

## 11.10 创建 Pod 失败,日志显示执行 mount 命令超时

#### 现象描述

创建Pod时,Pod一直处于ContainerCreating状态,此时查看huawei-csi-node的日志信息(详情请参考**11.1 如何查看华为CSI日志**),日志显示执行mount命令超时。

## 根因分析

原因1:该问题可能由于配置的业务IP网络不通,导致mount命令执行超时失败。

原因2:对于部分操作系统,如Kylin V10 SP1和SP2,使用NFSv3从容器内执行mount 命令耗时较长,导致mount命令超时,该问题可能由于容器运行时containerd的 LimitNOFILE参数值过大(10亿+)。

原因3:可能由于网络问题导致挂载失败,CSI默认挂载超时时间为30秒,超过30秒仍挂载失败,日志会显示执行mount命令超时。

## 解决措施或规避方法

步骤1 执行ping命令判断业务IP网络是否连通,如果无法ping通,则为原因1,请配置可用的业务IP地址,如果可以ping通,则执行步骤2。

步骤2 进入任意可以执行mount命令的容器中,指定使用NFSv3执行mount命令。如果命令超时,则可能是原因2,继续执行systemctl status containerd.service命令查看配置文件路径,然后执行cat /xxx/containerd.service命令查看配置文件。文件中如果有LimitNOFILE=infinity或LimitNOFILE的值大小为10亿,请执行步骤3。否则请联系华为工程师处理。

#### 步骤3 原因2可参考以下方式处理:

- 尝试使用NFSv4.0或NFSv4.1协议。
- 参考社区修改方案,将LimitNOFILE参数值修改为合适的值。该方案将会重启容器 运行时,请评估对业务的影响。

**步骤4** 在挂载失败的宿主机手动挂载该文件系统,如果时间超过30秒,需要用户自行排查该宿主机到存储节点网络是否存在问题。mount命令示例如下

- 执行以下命令创建测试目录。# mkdir /tmp/test mount
- 执行mount命令,挂载文件系统,并观察耗时,其中ip:nfs\_share\_path可以从huawei-csi-node日志中获取,详情请参考11.1 如何查看华为CSI日志# time mount ip:nfs\_share\_path /tmp/test\_mount
- 测试结束,执行以下命令解挂载文件系统 # umount /tmp/test\_mount

----结束

## 11.11 创建 Pod 失败,日志显示执行 mount 命令失败

#### 现象描述

NAS场景下,创建Pod时,Pod一直处于ContainerCreating状态,此时查看huaweicsi-node的日志信息(详情请参考**11.1 如何查看华为CSI日志**),日志显示执行mount命令失败。

## 根因分析

该问题可能由于存储侧未开启NFS 4.0/4.1协议,主机在使用NFS v4协议挂载失败后,未进行协商使用NFS v3协议挂载。

## 解决措施或规避方法

- 开启存储侧的NFS 3/4.0/4.1协议, 重新尝试默认挂载。
- 直接指定可用的NFS协议进行挂载,参考8.1.1.1.1 配置StorageClass。

# 11.12 创建 Pod 失败或重启 kubelet 后,日志显示挂载点已存在

## 现象描述

创建Pod时,Pod一直处于ContainerCreating状态,或者重启kubelet后,日志中显示挂载点已存在。此时查看huawei-csi-node的日志信息(详情请参考**11.1 如何查看华为CSI日志**),日志提示错误为: The mount /var/lib/kubelet/pods/xxx/mount is already exist, but the source path is not /var/lib/kubelet/plugins/kubernetes.io/xxx/globalmount

## 根因分析

该问题的根因是Kubernetes进行重复挂载操作。

## 解决措施或规避方法

执行以下命令,将已存在的路径解除挂载,其中"/var/lib/kubelet/pods/xxx/mount"为日志中提示的已存在的挂载路径。
umount /var/lib/kubelet/pods/xxx/mount

## 11.13 创建 Pod 失败,Events 日志显示"publishInfo doesn't exist"

## 现象描述

创建Pod时,Pod一直处于ContainerCreating状态,查看Pod中有打印告警事件:rpcerror:code = Internal desc = publishInfo doesn't exist。

## 根因分析

按照CSI协议约定,工作负载要使用一个PV卷时,CO(Container Orchestration system,通过RPC请求与CSI插件通信)会调用CSI插件提供的CSI协议中的 "ControllerPublishVolume"接口(huawei-csi-controller服务提供)完成PV卷的映射,然后调用CSI插件提供的"NodeStageVolume"接口(huawei-csi-node服务提供)完成PV卷的挂载。导致出现"publishInfo doesn't exist"错误的原因是在一次完整的挂载时,仅huawei-csi-node服务收到了"NodeStageVolume"请求,而在此之前huawei-csi-controller服务未收到"ControllerPublishVolume"请求,导致huawei-csi-controller服务未完成PV卷的映射,没有把映射信息传递给huawei-csi-node服务。

## 解决措施

解决该问题,需要触发Kubernetes调用"ControllerPublishVolume"接口。

如果集群中所有旧版本创建的工作负载均触发了该操作,则后续将不会出现该问题。

## 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,获取工作负载所在节点信息。

# kubectl get pod error-pod -n error-pod-in-namespace -owide

NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS

**GATES** 

pod-nfs 0/1 ContainerCreating 0 3s <none> node-1 <none>

步骤3 将该工作负载漂移至其他节点。

**步骤4** 若在集群内无法完成漂移,可在原节点完成工作负载重建,即进行删除-新建操作。

**步骤5** 观察该工作负载是否成功拉起,如果拉起失败请联系华为工程师

----结束

## 11.14 如何下载容器镜像到本地

## 使用 containerd 下载容器镜像

**步骤1** 执行**ctr image pull** *image:tag*命令,下载镜像到本地。其中*image:tag*表示需要拉取的镜像及其标签。

# ctr image pull k8s.gcr.io/sig-storage/livenessprobe:v2.5.0

**步骤2** 执行ctr image export *image*.tar *image:tag* 命令,导出镜像到文件。其中*image:tag* 表示需要导出的镜像,*image*.tar表示镜像导出后的文件名称。

# ctr image export livenessprobe.tar k8s.gcr.io/sig-storage/livenessprobe:v2.5.0

----结束

## 使用 Docker 下载容器镜像

**步骤1** 执行**docker pull** *image:tag*命令,下载镜像到本地。其中*image:tag*表示需要拉去的镜像。

# docker pull k8s.gcr.io/sig-storage/livenessprobe:v2.5.0

**步骤2** 执行**docker save** *image:tag* **-o** *image.tar* 命令,导出镜像到文件。其中*image:tag*表示需要导出的镜像,*image.*tar表示镜像导出后的文件名称。

# docker save k8s.gcr.io/sig-storage/livenessprobe:v2.5.0 -o livenessprobe.tar

----结束

## 使用 Podman 下载容器镜像

**步骤1** 执行**podman pull** *image:tag*命令,下载镜像到本地。其中*image:tag*表示需要拉去的镜像。

# podman pull k8s.gcr.io/sig-storage/livenessprobe:v2.5.0

**步骤2** 执行**podman save** *image:tag* **-o** *image.tar* 命令,导出镜像到文件。其中*image:tag* 表示需要导出的镜像,*image.*tar表示镜像导出后的文件名称。

# podman save k8s.gcr.io/sig-storage/livenessprobe:v2.5.0 -o livenessprobe.tar

----结束

## 11.15 如何获取 CSI 版本信息

本章节指导用户如何查看CSI版本信息。

## 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,查看huawei-csi-node所在节点信息。

# kubectl get pod -A -owide | grep huawei-csi-node NAMESPACE NAME READY STATUS RESTARTS AGE IP NOMINATED NODE READINESS GATES NODE huawei-csi huawei-csi-node-87mss 3/3 6m41s Runnina 0 192.168.129.155 node-1 <none> <none> huawei-csi huawei-csi-node-xp8cc 3/3 Running 0 6m41s 192.168.129.156 <none node-2 <none>

**步骤3** 使用远程访问工具(以PuTTY为例),通过节点IP地址,登录任意huawei-csi-node所在节点。

步骤4 执行以下命令,查看CSI版本信息。

# cat /var/lib/kubelet/plugins/csi.huawei.com/version

----结束

## 11.16 对接 Tanzu Kubernetes 集群常见问题及解决方法

本章节用于说明对接Tanzu Kubernetes集群时常见问题及解决办法,目前对接Tanzu Kubernetes集群时主要有以下三个问题:

- 未创建PSP权限导致Pod无法创建
- 主机挂载点与原生Kubernetes不同导致挂载卷失败
- livenessprobe容器端口与Tanzu vSphere端口冲突导致容器不断重启

## 11.16.1 未创建 PSP 权限导致 Pod 无法创建

## 现象描述

创建huawei-csi-controller和huawei-csi-node时,仅Deployment和DaemonSet资源创建成功,controller和node的Pod未创建。

## 根因分析

创建资源使用的service account没有PSP策略的"use"权限。

## 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vi psp-use.yaml命令,创建psp-use.yaml文件。

# vi psp-use.yaml

#### 步骤3 配置psp-use.yaml文件。

apiVersion: rbac.authorization.k8s.io/v1

kind: ClusterRole metadata:

name: huawei-csi-psp-role

rules:

- apiGroups: ['policy']

resources: ['podsecuritypolicies']

verbs: ['use']

---

apiVersion: rbac.authorization.k8s.io/v1

kind: ClusterRoleBinding

metadata:

name: huawei-csi-psp-role-cfg

roleRef:

kind: ClusterRole

name: huawei-csi-psp-role

apiGroup: rbac.authorization.k8s.io

subjects:

- kind: Group

apiGroup: rbac.authorization.k8s.io

name: system:serviceaccounts:huawei-csi

- kind: Group

apiGroup: rbac.authorization.k8s.io name: system:serviceaccounts:default

#### **步骤4** 执行kubectl create -f psp-use.yaml命令,创建PSP权限。

# kubectl create -f psp-use.yaml

----结束

## 11.16.2 修改主机挂载点

#### 现象描述

创建Pod时失败,华为CSI日志中报错"mount point does not exist"。

## 根因分析

huawei-csi-node中的"pods-dir"目录原生Kubernetes集群与Tanzu Kubernetes集群不一致。

## 解决措施或规避方法

#### 步骤1 进入helm/esdk/目录,执行vi values.yaml命令打开配置文件。

# vi values.yaml

#### 步骤2 将kubeletConfigDir参数修改为kubelet实际的安装目录。

#

Specify kubelet config dir path.

# kubernetes and openshift is usually /var/lib/kubelet

# Tanzu is usually /var/vcap/data/kubelet

kubeletConfigDir: /var/vcap/data/kubelet

#### ----结束

## 11.16.3 修改 livenessprobe 容器的默认端口

## 现象描述

huawei-csi-controller组件中livenessprobe容器一直重启。

## 根因分析

huawei-csi-controller的livenessprobe容器的默认端口(9808)与已有的Tanzu的vSphere CSI端口冲突。

## 解决措施或规避方法

将livenessprobe容器的默认端口修改为未占用端口。

步骤1 进入"helm/esdk/templates"目录,执行vi huawei-csi-controller.yaml命令打开 controller配置文件。

# vi huawei-csi-controller.yaml

步骤2 将livenessprobe容器的默认端口9808修改为其他未占用端口。

----结束

## 11.17 使用 Tanzu Kubernetes 集群时常见问题及解决方法

本章节用于说明在使用Tanzu Kubernetes集群时常见问题及解决办法

## 11.17.1 创建临时卷失败

## 现象描述

创建**通用临时卷**失败,报错PodSecurityPolicy: unable to admit pod: [spec.volumes[0]: Invalid value: "ephemeral": ephemeral volumes are not allowed to be used spec.volumes[0]

## 根因分析

当前使用的PSP策略中没有使用 "ephemeral" 卷的权限。

## 解决措施或规避方法

在默认PSP "pks-privileged"和"pks-restricted"中增加使用"ephemeral"卷的权限,以修改"pks-privileged"举例:

**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 psp pks-privileged命令,修改pks-privileged的配置。

# kubectl edit psp pks-privileged

步骤3 在spec.volumes中增加 "ephemeral", 示例如下

# Please edit the object below. Lines beginning with a '#' will be ignored, # and an empty file will abort the edit. If an error occurs while saving this file will be

```
# reopened with the relevant failures.
apiVersion: policy/v1beta1
kind: PodSecurityPolicy
metadata:
 annotations:
  apparmor.security.beta.kubernetes.io/allowedProfileName: '*'
  seccomp.security.alpha.kubernetes.io/allowedProfileNames: '*'
 creationTimestamp: "2022-10-11T08:07:00Z"
 name: pks-privileged
 resourceVersion: "1227763"
 uid: 2f39c44a-2ce7-49fd-87ca-2c5dc3bfc0c6
 allowPrivilegeEscalation: true
 allowedCapabilities:
 supplementalGroups:
  rule: RunAsAny
 volumes:
 - glusterfs
 - hostPath
 - iscsi
 - nfs
 - persistentVolumeClaim
 - ephemeral
```

## 步骤4 执行kubectl get psp pks-privileged -o yaml命令,确认是否添加成功。

# kubectl get psp pks-privileged -o yaml

----结束

## 11.18 通用临时卷扩容失败

## 现象描述

在Kubernetes版本低于1.25环境中,对LUN类型的**通用临时卷**扩容失败,显示PV已经扩容,但PVC未成功更新容量。

## 根因分析

该问题是由Kubernetes的bug导致,Kubernetes在1.25版本中修复了该问题。

## 11.19 PVC 扩容的目标容量超过存储池容量导致扩容失败

## 现象描述

在低于1.23版本的Kubernetes环境中,对PVC进行扩容,当目标容量超过存储池容量时,扩容失败。

## 根因分析

Kubernetes社区已知问题,详情请参考处理扩充卷过程中的失败。

## 解决措施或规避方法

参考处理扩充卷过程中的失败。

## 11.20 使用 oceanctl 工具管理后端时调用 webhook 失败

## 现象描述

当webhook的配置发生改变后,例如修改webhookPort参数值后,此时使用oceanctl工具对后端进行管理时调用webhook报错,如下:

```
root@ubuntu-master:/opt/huawei-csi/backend# oceanctl delete backend onas

Error: secret "onas" deleted
configmap "onas" deleted
configmap "onas" deleted
rror from server (InternalError): Internal error occurred: failed calling webhook "storage-backend-controller.xuanwu.huawei.i
ler.huawei-csi.svc:443/storagebackendclaim?timeout=10s": no service port 443 found for service "huawei-csi-controller"
```

## 根因分析

当webhook的配置发生改变后,导致validatingwebhookconfiguration资源失效。

## 解决措施或规避方法

**步骤1** 执行以下命令,删除validatingwebhookconfiguration资源。

# kubectl delete validatingwebhookconfiguration storage-backend-controller.xuanwu.huawei.io validatingwebhookconfiguration.admissionregistration.k8s.io "storage-backend-controller.xuanwu.huawei.io" deleted

**步骤2** 执行以下命令,重启CSI Controller,请通过"**--replicas=\***"恢复CSI Controller的副本数,下例为恢复至1个,请根据实际情况修改。

# kubectl scale deployment huawei-csi-controller -n huawei-csi --replicas=0 deployment.apps/huawei-csi-controller scaled # kubectl scale deployment huawei-csi-controller -n huawei-csi --replicas=1 deployment.apps/huawei-csi-controller scaled

步骤3 执行以下命令,检查CSI Controller是否成功拉起。

```
# kubectl get pod -n huawei-csi
NAME READY STATUS RESTARTS AGE
huawei-csi-controller-58d5b6b978-s2dsq 9/9 Running 0 19s
huawei-csi-node-dt6nd 3/3 Running 0 77m
```

----结束

## 11.21 存储侧更新密码后账户被锁定

## 现象描述

用户在存储侧修改后端密码之后,该后端账号被锁定。

## 根因分析

CSI登录存储时使用存储后端配置的账户和密码,当存储侧修改了该账户密码之后, CSI登录失败后会重试。以Dorado V6存储为例,默认的登录策略是密码校验失败3次 后将会锁定账户,因此当CSI重试超过3次之后,该账户就会被锁定。

## 解决措施或规避方法

**步骤1** 如果后端配置的账户是admin,请执行以下命令,将huawei-csi-controller服务副本数置为0,如果使用的是非admin账户,忽略此步骤。

kubectl scale deployment huawei-csi-controller -n huawei-csi --replicas=0

- 步骤2 使用admin账户登录存储,修改登录策略。以Dorado V6存储为例,在 DeviceManager管理界面,选择"设置 > 用户与安全 > 安全策略 >登录策略 >修改> 密码锁定",取消密码锁定。
- 步骤3 如果如果后端配置的账户是admin,执行以下命令,通过"--replicas=\*"恢复CSI Controller的副本数,下例为恢复至1个,请根据实际情况修改。如果使用的是非 admin账户,忽略此步骤。

kubectl scale deployment huawei-csi-controller -n huawei-csi --replicas=1

- 步骤4 使用oceanctl工具修改存储后端密码,修改后端密码请参考7.3 更新存储后端章节。
- 步骤5 使用admin账户登录存储,修改登录策略,以Dorado V6存储为例,在 DeviceManager管理界面,选择"设置 > 用户与安全 > 安全策略 >登录策略 >修改> 密码锁定",恢复密码锁定。

----结束

# **12**<sub>附录</sub>

- 12.1 OceanStor V3/V5系列和OceanStor Dorado V3系列ALUA特性配置策略样例
- 12.2 OceanStor Dorado 6.x ALUA特性配置策略样例
- 12.3 分布式存储ALUA特性配置策略样例
- 12.4 安装Helm 3
- 12.5 配置自定义权限

## 12.1 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:

```
parameters:
ALUA:

"*":

MULTIPATHTYPE: 1

FAILOVERMODE: 3

SPECIALMODETYPE: 0

PATHTYPE: 0

node1:

MULTIPATHTYPE: 1

FAILOVERMODE: 3

SPECIALMODETYPE: 0

PATHTYPE: 1
```

对于主机名为"node1",上述ALUA配置段都能用于配置启动器。根据**9.1.1.1 配置华 为企业存储后端的ALUA参数**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:

```
parameters:
ALUA:
node[0-9]:
MULTIPATHTYPE: 1
FAILOVERMODE: 3
SPECIALMODETYPE: 0
PATHTYPE: 0
node[5-7]:
MULTIPATHTYPE: 1
```

```
FAILOVERMODE: 3
SPECIALMODETYPE: 0
PATHTYPE: 1
```

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**9.1.1.1 配置华为企业存储后端的ALUA参数**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:
node$:
MULTIPATHTYPE: 1
FAILOVERMODE: 3
SPECIALMODETYPE: 0
PATHTYPE: 0
node10$:
MULTIPATHTYPE: 1
FAILOVERMODE: 3
SPECIALMODETYPE: 0
PATHTYPE: 1
```

根据**9.1.1.1 配置华为企业存储后端的ALUA参数**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。

## 12.2 OceanStor Dorado 6.x ALUA 特性配置策略样例

#### 例1.配置文件如下:

```
parameters:
ALUA:

"*":

accessMode: 1
hyperMetroPathOptimized: 1
node1:
accessMode: 1
hyperMetroPathOptimized: 0
```

对于主机名为"node1",上述ALUA配置段都能用于配置启动器。根据**9.1.1.1 配置华** 为企业存储后端的ALUA参数中的配置策略规则,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:

```
parameters:
ALUA:
node[0-9]:
accessMode: 1
hyperMetroPathOptimized: 1
node[5-7]:
accessMode: 1
hyperMetroPathOptimized: 0
```

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**9.1.1.1 配置华为企业存储后端的ALUA参数**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:
node1$:
```

```
node[0-9]:
accessMode: 1
hyperMetroPathOptimized: 1
node10$:
accessMode: 1
hyperMetroPathOptimized: 0
```

根据**9.1.1.1 配置华为企业存储后端的ALUA参数**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。

## 12.3 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**9.1.1.2 配置分布式存储后端的ALUA参数**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:

```
parameters:

ALUA:

node[0-9]:

switchoverMode: Enable_alua

pathType: optimal_path

node[5-7]:

switchoverMode: Enable_alua

pathType: non_optimal_path
```

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**9.1.1.2 配置分布式存储后端的ALUA参数**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:
node1$:
switchoverMode: Enable_alua
pathType: optimal_path
node10$:
switchoverMode: Enable_alua
pathType: non_optimal_path
```

根据**9.1.1.2 配置分布式存储后端的ALUA参数**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。

## 12.4 安装 Helm 3

本章节指导用户如何安装Helm 3。

参考: https://helm.sh/docs/intro/install/

## 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

 $version. Build Info \{Version: "v3.9.0", GitCommit: "7ceeda6c585217a19a1131663d8cd1f7d641b2a7", GitTreeState: "clean", GoVersion: "go1.17.5"\}$ 

----结束

## 12.5 配置自定义权限

## 自定义角色配置

针对不同的存储资源,请参考以下配置:

- NAS相关资源请参考表12-1配置最小权限。
- SAN相关资源请参考表12-2配置最小权限。

#### □ 说明

可参考存储文档:《 OceanStor Dorado 6000, Dorado 18000**系列 产品文档**》配置自定义角色权限。

#### 表 12-1 NAS 相关资源最小权限说明

权限对象	父级对象	读写权限	功能说明
workload_type	file_storage_service	只读	查询应用类型
file_system	file_storage_service	读写	管理文件系统
fs_snapshot	file_storage_service	读写	管理文件系统快照

权限对象	父级对象	读写权限	功能说明
quota	file_storage_service	读写	管理文件系统配额
nfs_service	file_storage_service	只读	查询NFS服务
share	file_storage_service	读写	管理NFS共享
dtree	file_storage_service	读写	管理dtree
hyper_metro_p air	hyper_metro	读写	创建文件系统双活Pair
hyper_metro_d omain	hyper_metro	只读	查询文件系统双活域信 息
remote_device	local_data_protection	只读	查询远端设备信息
storage_pool	pool	只读	查询存储池信息
smart_qos	resource_performanc e_tuning	读写	管理SmartQoS策略
system	system	只读	查询存储设备信息(仅 所属组为系统组时需要 配置)
vstore	vstore	只读	查询租户信息

## 表 12-2 SAN 相关资源最小权限说明

权限对象	父级对象	读写权限	功能说明
remote_device	local_data_protection	只读	查询远端设备信息
hyper_clone	local_data_protection	读写	管理Clone Pair
lun_snapshot	local_data_protection	读写	管理LUN快照
workload_type	lun	只读	查询应用类型
lun	lun	读写	管理LUN
host	mapping_view	读写	管理主机
host_group	mapping_view	读写	管理主机组
initiator	mapping_view	读写	管理启动器
lun_group	mapping_view	读写	管理LUN组
mapping_view	mapping_view	读写	管理映射视图
target	mapping_view	只读	查询iSCSI启动器
port	network	只读	查询逻辑端口
storage_pool	pool	只读	查询存储池信息

权限对象	父级对象	读写权限	功能说明
smart_qos	resource_performanc e_tuning	读写	管理SmartQoS策略
system	system	只读	查询存储设备信息(仅 所属组为系统组时需要 配置)
vstore	vstore	只读	查询租户信息