用户指南 V2.2.14

Kubernetes CSI

文档版本 01

发布日期 2021-09-29





版权所有 © 华为技术有限公司 2021。 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: https://www.huawei.com

客户服务邮箱: support@huawei.com

客户服务电话: 4008302118

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	用于警示紧急的危险情形,若不避免,将会导致人员死亡或严重 的人身伤害。
▲警告	用于警示潜在的危险情形,若不避免,可能会导致人员死亡或严 重的人身伤害。
▲注意	用于警示潜在的危险情形,若不避免,可能会导致中度或轻微的 人身伤害。
注意	用于传递设备或环境安全警示信息,若不避免,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。 "注意"不涉及人身伤害。
□ 说明	用于突出重要/关键信息、最佳实践和小窍门等。 "说明"不是安全警示信息,不涉及人身、设备及环境伤害。

目录

則言	ii
1 概述	1
2 环境要求	2
3 限制说明	4
4 安装部署	7
- スペロー 4.1 软件包获取	
4.2 软件包组件	
4.3 制作华为 CSI 镜像	
4.4 配置主机多路径	
4.4.1 安装多路径工具包	
4.4.2 配置多路径服务	
4.5.2 通过 FC 对接企业存储 SAN 配置	
4.5.3 通过 NFS 对接企业存储 NAS 配置	
4.5.4 通过 NVMe over RoCE 对接企业存储 SAN 配置	17
4.6 对接分布式存储配置	18
4.6.1 通过 SCSI 对接分布式存储 SAN 配置	18
4.6.2 通过 iSCSI 对接分布式存储 SAN 配置	20
4.6.3 通过 NFS 对接分布式存储 NAS 配置	21
4.7 启动 huawei-csi 服务	23
5 升级操作	26
5.1 卸载原 CSI	
5.2 安装新 CSI	
6 使用说明	
6.1 管理 StorageClass	
6.1.1 创建 StorageClass	
6.1.1.1 创建 LUN StorageClass	
6.1.1.2 创建文件系统 StorageClass	
6.1.2 删除 StorageClass	
6.2 管理 PVC	
H	

Nubernetes esi	
6.2.1 创建 PVC	31
6.2.2(可选)扩容 PVC	
6.2.2.1 安装扩容依赖组件服务	33
6.2.2.2 扩容 PVC	
6.2.3(可选)克隆 PVC	
6.2.4 (可选)从快照创建 PVC	
6.2.5 删除 PVC	37
6.3 管理 Pod	37
6.3.1 创建 Pod	37
6.3.2 删除 Pod	38
6.4 (可选)管理快照	39
6.4.1 安装 Snapshot 依赖组件服务	39
6.4.2 管理 VolumeSnapshotClass	39
6.4.2.1 创建 VolumeSnapshotClass	39
6.4.2.2 删除 VolumeSnapshotClass	40
6.4.3 管理 VolumeSnapshot	41
6.4.3.1 创建 VolumeSnapshot	41
6.4.3.2 删除 VolumeSnapshot	42
7 高级特性	43
7.1 配置多个后端	43
7.2 指定后端创建 PVC	44
7.3 指定存储池创建 PVC	44
7.4 配置 ALUA 特性	45
7.4.1 配置 OceanStor V3/V5 系列和 OceanStor Dorado V3 系列 ALUA	45
7.4.2 配置 OceanStor Dorado V6 ALUA	48
7.4.3 配置分布式存储 ALUA	50
7.5 配置存储拓扑感知	51
7.6 企业存储高级特性	55
7.6.1 配置 QoS	55
7.6.2 配置租户	56
7.6.3 配置 SAN 远程复制	57
7.6.4 配置 SAN 双活	58
7.6.5 配置 NAS 远程复制	60
7.6.6 配置 NAS 双活	61
7.6.7 配置应用类型	63
7.7 分布式存储高级特性	64
7.7.1 配置 QoS	64
8 常用操作	65
8.1 卸载 CSI	
8.2 更新 CSI 上配置的存储用户名或密码	
8.3 更新 huawei-csi 的 configmap 对象	
8.4 为 huawei-csi 新增后端	

Kubernetes CSI	日 录
8.5 更新 huawei-csi-controller 服务	69
8.6 更新 huawei-csi-node 服务	
8.7 修改日志输出模式	71
8.7.1 修改 huawei-csi-controller 服务的日志输出模式	71
8.7.2 修改 huawei-csi-node 服务的日志输出模式	72
9 FAQ	75
9.1 查看日志信息	75
9.2 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	76
9.3 启动 huawei-csi-node 失败,提示错误为: "/var/lib/iscsi is not a directory"	77
9.4 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	
9.5 启动 huawei-csi 服务时,服务启动异常, 状态显示 InvalidImageName	80
9.6 创建 PVC 时, PVC 的状态为 Pending	81
9.7 删除 PVC 前,PVC 的状态为 Pending	83
9.8 创建 Pod 时,Pod 的状态为 ContainerCreating	84
10 附录	86
10.1 OceanStor V3/V5 系列和 OceanStor Dorado V3 系列 ALUA 特性配置策略样例	86
10.2 OceanStor Dorado V6 ALUA 特性配置策略样例	
10.3 分布式存储 ALUA 特性配置策略样例	

■ 概述

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

2 环境要求

- Kubernetes环境部署完成并正常运行。
- 华为存储正常运行。
- 需要提前在Kubernetes的所有worker主机上安装扫盘和文件挂载必须的工具(如果由于系统工具欠缺导致容器和服务无法正常运行,可先参考9.1 查看日志信息进行日志查看,并自行安装主机侧工具)

表 2-1 Kubernetes 版本和企业存储产品版本要求

Kubernetes版本	企业存储产品版本
1.13/1.14/1.15/1.16/1.17/1.18/ 1.19/1.20/1.21	OceanStor Dorado V6 6.0.0/6.0.1/6.1.0/6.1.2
	OceanStor Dorado V3 V300R002
	OceanStor F V5/V5 V500R007/V500R007 Kunpeng
	OceanStor F V3/V3 V300R006

表 2-2 Kubernetes 版本和分布式存储产品版本要求

Kubernetes版本	分布式存储产品版本
1.13/1.14/1.15/1.16/1.17/1.18/ 1.19/1.20/1.21	FusionStorage V100R006C30 FusionStorage Block 8.0.0/8.0.1 OceanStor Pacific系列 8.1.0

表 2-3 Huawei CSI 特性支持说明 (√: 支持, x: 不支持)

特性	1.13	1.14	1.15	1.16	1.17	1.18	1.19	1.20	1.21
Create PVC	V	V	V	V	V	V	V	V	V

特性	1.13	1.14	1.15	1.16	1.17	1.18	1.19	1.20	1.21
Delete PVC	V	V	V	V	V	V	V	V	V
Create POD	$\sqrt{}$	V	$\sqrt{}$	V	V	V	V	V	\checkmark
Delete POD	V	V	V	V	V	V	V	V	\checkmark
Offline Resize	х	х	х	V	V	V	V	V	V
Online Resize	х	х	х	V	V	V	V	V	\checkmark
Create Snapsh ot	х	х	х	х	V	V	V	V	V
Delete Snapsh ot	х	х	х	х	V	V	V	V	V
Restore	х	х	х	х	V	V	$\sqrt{}$	$\sqrt{}$	$\sqrt{}$
Clone	х	х	х	х	V	V	V	V	V

3 限制说明

本章节用于说明CSI对接存储时的限制。

表 3-1 限制说明

场景描述	限制描述	支持存储	备注
PersistentVolume Claim (PVC) 的访 问模式	ReadWriteOnce: SAN/NAS ReadWriteMany: NAS ReadWriteOnly: SAN/NAS 如果SAN需要使用 ReadWriteMany模式,需要由Pod业 务保证数据一致 性。	SAN: OceanStor V3/V5, OceanStor Dorado V3/Dorado V6, FusionStorage 8.0 NAS: OceanStor V3/V5, OceanStor Dorado V6, OceanStor Pacific系列	
PVC	每批次最多可批量 创建/删除100个。	SAN: OceanStor V3/V5, OceanStor Dorado V3/Dorado V6, FusionStorage 8.0 NAS: OceanStor V3/V5, OceanStor Dorado V6, OceanStor Pacific系列	存储限制RESTful 请求并发量为100

场景描述	限制描述	支持存储	备注
Pod	每批次最多可批量 创建/删除100个。	SAN: OceanStor V3/V5, OceanStor Dorado V3/Dorado V6, FusionStorage 8.0 NAS: OceanStor	
		V3/V5, OceanStor Dorado V6, OceanStor Pacific系列	
快照	不涉及	SAN: OceanStor V3/V5, OceanStor Dorado V3/Dorado V6, FusionStorage 8.0 NAS: OceanStor V3/V5, OceanStor Dorado V6	
从快照创建PVC	快照恢复,从快照 创建一个新的 PVC,而不是快照 回滚到原PVC。	SAN: OceanStor V3/V5, OceanStor Dorado V3/Dorado V6, FusionStorage 8.0 NAS: OceanStor V3/V5	
扩容PVC	 存储资源的扩容,不支持缩容。 RWO模式的PVC,不支持扩容。 	SAN: OceanStor V3/V5, OceanStor Dorado V3/Dorado V6, FusionStorage 8.0 NAS: OceanStor V3/V5, OceanStor Dorado V6	
克隆PVC	源PVC和目标PVC 的StorageClass需 要相同。	SAN: OceanStor V3/V5, OceanStor Dorado V3/Dorado V6, FusionStorage 8.0 NAS: OceanStor V3/V5	

场景描述	限制描述	支持存储	备注
双活	● 只能在常用Pod, 均正常和Pod, 如果PVC和Pod, 如果单个能够是 如果,只是的能够。 是一个,是一个。 。 一个,是一个。 。 一个。 一个	SAN: OceanStor V3/V5, OceanStor Dorado V3/Dorado V6 NAS: OceanStor V3/V5	
远程复制	远程复制与双活不 能在一个 StorageClass同时 配置。	SAN: OceanStor V3/V5, OceanStor Dorado V3/Dorado V6 NAS: OceanStor V3/V5, OceanStor Dorado V6	
盘符残留	由于节点故障,容器应用漂移到其他节点,节点上则复后,节点上回存。需要自己的情况,可以是一个时间,不是一个时间,可以可以可以可以可以可以可以可以可以可以可以可以可以可以可以可以可以可以可以	SAN: OceanStor V3/V5, OceanStor Dorado V3/Dorado V6, FusionStorage 8.0	条件: iSCSI/FC + Multipath

4 安装部署

- 4.1 软件包获取
- 4.2 软件包组件
- 4.3 制作华为CSI镜像
- 4.4 配置主机多路径
- 4.5 对接企业存储配置
- 4.6 对接分布式存储配置
- 4.7 启动huawei-csi服务

4.1 软件包获取

通过华为自有Kubernetes CSI仓库获取对应的插件包。

步骤1 打开浏览器,访问仓库地址: https://github.com/Huawei/eSDK_K8S_Plugin/releases。

步骤2 选择对应的版本包,下载*eSDK_Cloud_Storage_Plugin_*.*.****zip,**.*.****表示发布版本号。

步骤3 解压该压缩包。

步骤4 在解压出的目录中找到对应的包和资料。

----结束

4.2 软件包组件

通过解压*eSDK_Cloud_Storage_Plugin_*.*.***.zip,获取到CSI安装和使用需要的软件包和示例文件。软件包组件结构如**表4-1**所示。

表 4-1 软件包组件描述

组件	组件描述
bin/huawei-csi	实现CSI规范接口的服务组件。
bin/secretGenerate	明文密码加密工具,用于生产secret对象。
bin/secretUpdate	明文密码加密工具,用于更新secret对象。
yamls	后续部署过程中用到的yaml示例文件集合。

4.3 制作华为 CSI 镜像

Huawei CSI在运行时,是以容器的形态存在。目前Huawei CSI只提供二进制包(bin/huawei-csi),无法直接使用,因此我们需要根据二进制文件制作CSI镜像,用于启动Huawei CSI服务。

前提条件

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

操作步骤

步骤1 登录该Linux主机。

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

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

步骤4 拷贝huawei-csi组件到"image"目录下。

步骤5 执行**vi Dockerfile**命令,创建Dockerfile文件。 # vi Dockerfile

步骤6 按l或Insert进入编辑状态,在Dockerfile文件中输入以下内容。修改完成后,按Esc,并输入:wq!,保存修改。

FROM busybox:stable-glibc

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

ENTRYPOINT ["/huawei-csi"]

须知

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

步骤7 执行docker build -f Dockerfile -t huawei-csi:2.2.13 .命令制作镜像。

docker build -f Dockerfile -t huawei-csi:2.2.13.

□ 说明

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

步骤8 执行docker image ls | grep huawei-csi命令,检查镜像是否制作完成。显示如下回显,则表示制作完成。

docker image ls | grep huawei-csi huawei-csi 2.2.13

c8b5726118ac

About a minute ago 39 MB

步骤9 执行docker save huawei-csi:2.2.13 -o huawei-csi.tar命令,导出镜像。

docker save huawei-csi:2.2.13 -o huawei-csi.tar

□ 说明

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

步骤10 执行**scp huawei-csi.tar** *<user>@<ip>:*/*<path>*命令将huawei-csi.tar镜像文件拷贝到 Kubernetes集群所有worker节点上,根据界面提示输入密码。

scp huawei-csi.tar <user>@<ip>:/<path>

- <user>: 登录Kubernetes集群worker节点的用户名。
- <ip>: Kubernetes集群worker节点的登录IP地址。
- <path>: 需要拷贝到Kubernetes集群worker节点的文件夹名称。
- **步骤11** 登录到Kubernetes集群的worker节点,执行**docker load -i huawei-csi.tar**命令,导入镜像。

docker load -i huawei-csi.tar

步骤12 导入完成后,可执行docker image ls | grep huawei-csi命令,检查导入是否成功,如下回显所示则表示导入成功。

docker image ls | grep huawei-csi

huawei-csi

2.2.13

c8b5726118ac 10 minutes ago 39MB

步骤13 重复执行步骤10到步骤12,将镜像导入到Kubernetes集群的所有worker节点。

----结束

4.4 配置主机多路径

4.4.1 安装多路径工具包

本章节介绍如何安装多路径工具包。

前提条件

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

注意事项

只有通过iSCSI和FC组网对接存储设备时需要配置主机多路径服务。

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worker节点。

步骤2 根据不同的操作系统安装多路径工具包。

CentOS:

yum install -y device-mapper-multipath

Ubuntu:

apt-get install -y multipath-tools apt-get install -y multipath-tools-boot

SUSE:

zypper install -y multipath-tools

步骤3 开启主机多路径服务。

CentOS:

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

Ubuntu:

systemctl restart multipath-tools.service

SUSE:

systemctl restart multipath-tools.service chkconfig multipathd on

步骤4 重复执行步骤1到步骤3,将多路径工具安装到所有worker节点。

----结束

4.4.2 配置多路径服务

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

注意事项

通过iSCSI和FC对接存储设备时,才需要配置多路径服务。

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worker节点。

步骤2 执行**vi** /etc/multipath.conf命令,修改"multipath.conf"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:**wg!**,保存修改。

须知

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

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

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

```
devices {
  device {
                              "HUAWEI"
          vendor
          product
                               "XSG1"
          path_grouping_policy
                                 multibus
          path_checker
                                tur
          prio
                             const
          path_selector
                                "service-time 0"
          failback
                              immediate
          no_path_retry
```

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

```
devices {
  device {
         vendor
                           "HUAWEI"
                           "XSG1"
         product
         path_grouping_policy group_by_prio
         path_checker
                             tur
         prio
                         alua
         path_selector
                            "round-robin 0"
         failback
                           immediate
         no_path_retry
                             15
```

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

```
devices {
  device {
                                "Huawei"
          vendor
                                "VBS fileIO"
          product
          path_grouping_policy
                                    multibus
          path_checker
          prio
                              const
          path_selector
                                 "service-time 0"
          failback
                               immediate
                                  "10"
          no_path_retry
      }
}
```

步骤3 配置完成后,执行以下命令,重启multipathd服务。

systemctl restart multipathd.service

步骤4 重复执行步骤1到步骤3,为所有worker节点配置多路径服务。

----结束

4.5 对接企业存储配置

本章节描述如何配置huawei-csi插件对接华为企业存储。

4.5.1 通过 iSCSI 对接企业存储 SAN 配置

当您需要通过iSCSI对接企业存储SAN配置时,执行此操作。

前提条件

- Kubernetes的所有worker节点已安装iSCSI客户端。
- Kubernetes的所有worker节点与待接入的存储设备管理IP通信正常。
- Kubernetes的所有worker节点与待接入的存储设备业务IP通信正常。
- 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4.4 配置主机多路径。
- 已向管理员获取Kubernetes集群中任意master节点的IP地址、登录帐号和密码。

操作步骤

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

表 4-2 配置项描述

配置项	参数格式	描述	备注
name	字符串	必填,对存储设备 自定义的名称。	自定义字符串,支持大小写字母、数 字、中划线组合。
storage	字符串	必填,待接入存储 设备的类型。	对接企业存储SAN场景,固定填写 "oceanstor-san"。
pools	列表	必填,被使用的待 接入存储设备上的 存储池名称。	支持同一存储设备上的一个或多个存储池, 用逗号分割。 可通过登录DeviceManager获取支持块存储服务的存储池。
urls	列表	必填,待接入的存 储设备的管理 URL。	支持同一存储设备的一个或多个管理 URL,用逗号分割,当前仅支持IPv4。 例如: https://192.168.125.20:8088 说明 1个存储设备有多个控制器,每个控制器有 一个管理URL,所以1个存储设备存在多个 管理URL。
paramete	字典	必填,iSCSI场景的可变参数。	iSCSI场景protocol参数固定填写 "iscsi"。 portals参数填写待接入存储设备的 iSCSI业务IP地址,多个业务IP地址请用 逗号分割。 iSCSI业务IP地址可通过登录 DeviceManager获取。 OceanStor V3/V5系列:指定存储 设备上的以太网端口IP地址,在 DeviceManager管理界面,选择 "资源分配 > 端口 > 以太网端口"。 OceanStor Dorado V3系列:指定存储设备上的以太网端口IP地址,在DeviceManager管理界面,选择 "资源分配 > 端口 > 以太网端口"。 OceanStor Dorado V6系列:指定存储设备上的逻辑端口IP地址,在DeviceManager管理界面,选择 "资源分配 > 端口 > 以太网端口"。 OceanStor Dorado V6系列:指定存储设备上的逻辑端口IP地址,在DeviceManager管理界面,选择 "服务 > 网络 > 逻辑端口",获取数据协议类型为iSCSI的IP。

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

kubectl create -f huawei-csi-configmap.yaml

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

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

----结束

4.5.2 通过 FC 对接企业存储 SAN 配置

当您需要通过FC对接企业存储SAN配置时,请执行此操作。

限制条件

当您需要通过FC对接企业存储SAN配置时,需要保证主机侧没有盘符残留,如果有盘符残留,请参考9.4 集群中worker节点宕机并恢复后,Pod完成failover,但是Pod所在源主机出现盘符残留进行盘符清理。

前提条件

- Kubernetes的所有worker节点与待接入的存储设备管理IP通信正常。
- Kubernetes的所有worker节点能通过FC与待接入的存储设备通信。
- 如果是多路径组网,请确保Kubernetes的所有worker节点上已安装多路径软件, 详细请参见**4.4 配置主机多路径**。
- 已向管理员获取Kubernetes集群中任意master节点的IP地址、登录帐号和密码。

操作步骤

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

表 4-3 配置项描述

配置项	参数格式	描述	备注
name	字符串	必填,对存储设备 自定义的名称。	自定义字符串,支持大小写字母、数 字、中划线组合。
storage	字符串	必填,待接入存储 设备的类型。	对接企业存储SAN场景,固定填写 "oceanstor-san"。
pools	列表	必填,被使用的待接入存储设备上的存储池名称。	支持同一存储设备上的一个或多个存储池, 用逗号分割。 可通过登录DeviceManager获取支持块存储服务的存储池。
urls	列表	必填,待接入存储 设备的管理URL。	支持同一存储设备的一个或多个管理URL,用逗号分割,当前仅支持IPv4。例如:https://192.168.125.20:8088
paramete rs	字典	必填,FC场景的可 变参数。	FC场景protocol参数固定填写"fc"。

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

kubectl create -f huawei-csi-configmap.yaml

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

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

----结束

4.5.3 通过 NFS 对接企业存储 NAS 配置

当您需要通过NFS对接企业存储NAS配置时,执行此操作。

前提条件

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

操作步骤

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

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

表 4-4 配置项描述

配置项	参数格式	描述	备注	
name	字符串	必填,对存储设备 自定义的名称。	自定义字符串,支持大小写字母、数 字、中划线组合。	
storage	字符串	必填,待接入存储 设备的类型。	对接企业存储NAS场景固定填写 "oceanstor-nas"。	
pools	列表	必填,被使用的待接入存储设备上的存储池名称。	支持同一存储设备上的一个或多个存储池, 用逗号分割。 可通过登录DeviceManager获取支持文件存储服务的存储池。	
urls	列表	必填,待接入的存储设备的管理URL。	支持同一存储设备的一个或多个管理 URL,用逗号分割,当前仅支持IPv4。 例如: https://192.168.125.20:8088 说明 1个存储设备有多个控制器,每个控制器有 一个管理URL,所以1个存储设备存在多个 管理URL。	
paramete rs	字典	必填,NAS场景的 可变参数。	protocol参数固定填写"nfs"。 portals:指定存储的逻辑端口IP或者 DNS Zone,只支持配置一个。	

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

kubectl create -f huawei-csi-configmap.yaml

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

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

----结束

4.5.4 通过 NVMe over RoCE 对接企业存储 SAN 配置

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

前提条件

- Kubernetes的所有worker节点与待接入的存储设备管理IP通信正常。
- Kubernetes的所有worker节点与待接入的存储设备业务IP通信正常。
- Kubernetes的所有worker节点已安装NVMe客户端。
- 已向管理员获取Kubernetes集群中任意master节点的IP地址、登录帐号和密码。
- 和NVMe相关的存储,主机,多路径等兼容性,请参考https://supportopen.huawei.com/zh/pages/user/compatibility/support-matrix.jsf。

操作步骤

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

表 4-5 配置项描述

配置项	参数格 式	描述	备注
name	字符串	必填,对存储设备 自定义的名称。	自定义字符串,支持大小写字母、数 字、中划线组合。
storage	字符串	必填,待接入存储 设备的类型。	对接企业存储SAN场景固定填写 "oceanstor-san"。

配置项	参数格 式	描述	备注
pools	列表	必填,被使用的待接入存储设备上的存储池名称。	支持同一存储设备上的一个或多个存储池, 用逗号分割。 可通过登录DeviceManager获取存储池。
urls	列表	必填,待接入的存储设备的管理URL。	支持同一存储设备的一个或多个管理 URL,用逗号分割,当前仅支持IPv4。 例如: https://192.168.125.20:8088 说明 1个存储设备有多个控制器,每个控制器有 一个管理URL,所以1个存储设备存在多个 管理URL。
paramete rs	字典	必填,RoCE场景的 可变参数。	RoCE场景protocol参数固定填写 "roce"。 portals参数填写存储的数据协议类型 为NVMe over RoCE的逻辑端口IP,用 逗号分割。

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

kubectl create -f huawei-csi-configmap.yaml

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

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

----结束

4.6 对接分布式存储配置

本章节描述如何配置huawei-csi插件对接华为分布式存储。

4.6.1 通过 SCSI 对接分布式存储 SAN 配置

当您需要通过SCSI对接分布式存储SAN配置,执行此操作。

前提条件

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

操作步骤

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

步骤2 配置huawei-csi-configmap.yaml文件, huawei-csi-configmap.yaml文件的模板如回显所示(您也可以参考软件包中yamls/huawei-csi-configmap-fusionstorage-scsi.yaml示例文件)。请根据实际情况设置相关参数,并保存为yaml格式的文件,详细请参考表4-6。

表 4-6 配置项描述

配置项	参数格式	描述	备注
name	字符串	必填,对存储设备 自定义的名称。	自定义字符串,支持大小写字母、数 字、中划线组合。
storage	字符串	必填,待接入存储 设备的类型。	对接分布式存储SAN场景固定填写 "fusionstorage-san"。
pools	列表	必填,被使用的待接入存储设备上的存储池名称。	支持同一存储设备上的一个或多个存储池, 用逗号分割。 可通过登录DeviceManager获取存储池。
urls	列表	必填,待接入存储 设备的管理URL。	支持同一存储设备的一个或多个管理 URL,用逗号分割,当前仅支持IPv4。 例如: https://192.168.125.20:28443 说明 1个存储设备有多个控制器,每个控制器有 一个管理URL,所以1个存储设备存在多个 管理URL。
paramete rs	字典	必填,可变参数。	protocol参数固定填写"scsi"。 portals参数填写主机和VBS节点IP Pair 列表,参数格式为 [{"hostname":"**.****"}],其中 hostname为对应的worker节点主机 名,"*.**.*"为分布式存储块客户端对 应的管理IP地址。 如果存在多个worker节点,则相应地 以字典格式配置多个,以逗号隔开。

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

kubectl create -f huawei-csi-configmap.yaml

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

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

----结束

4.6.2 通过 iSCSI 对接分布式存储 SAN 配置

当您需要通过iSCSI对接分布式存储SAN配置时,请执行此操作。

前提条件

- Kubernetes的所有worker节点已安装iSCSI客户端。
- Kubernetes的所有worker节点与待接入的存储设备管理IP通信正常。
- Kubernetes的所有worker节点与待接入的存储设备业务IP通信正常。
- 已向管理员获取Kubernetes集群中任意master节点的IP地址、登录帐号和密码。
- 如果是多路径组网,请确保所有worker节点上安装多路径软件。

注意事项

- Kubernetes的worker节点主机名称由数字、字母、"_"、"-"、"-"、"."和":"组成,首字符只能是数字、字母或"_",名称长度不超过31个字符。
- 仅FusionStorage 8.0及其以后版本支持iSCSI组网配置。

操作步骤

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

表 4-7 配置项描述

配置项	参数格式	描述	备注
name	字符串	必填,对存储设备 自定义的名称。	自定义字符串,支持大小写字母、数 字、中划线组合。
storage	字符串	必填,待接入存储 设备的类型。	对接分布式存储SAN场景固定填写 "fusionstorage-san"。
pools	列表	必填,被使用的待接入存储设备上的存储池名称。	支持同一存储设备上的一个或多个存储 池, 用逗号分割。 可通过登录DeviceManager获取存储 池。
urls	列表	必填,待接入存储 设备的管理URL。	支持同一存储设备的一个或多个管理 URL,用逗号分割,当前仅支持IPv4。 例如: https://192.168.125.20:28443 说明 1个存储设备有多个控制器,每个控制器有 一个管理URL,所以1个存储设备存在多个 管理URL。
paramete rs	字典	必填,可变参数	iSCSI场景protocol参数固定填写 "iscsi"。 portals参数填写指定存储设备的iSCSI 业务IP地址,用逗号分割,可通过登录 DeviceManager获取。

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

kubectl create -f huawei-csi-configmap.yaml

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

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

----结束

4.6.3 通过 NFS 对接分布式存储 NAS 配置

当您需要通过NFS对接分布式存储NAS配置时,请执行此步骤。

前提条件

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

操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意master节点。
- **步骤2** 配置huawei-csi-configmap.yaml文件, huawei-csi-configmap.yaml文件的模板如回显所示(您也可以参考软件包中yamls/huawei-csi-configmap-fusionstorage-nfs.yaml示例文件)。请根据实际情况设置相关参数,并保存为yaml格式的文件,详细请参考表4-8。

表 4-8 配置项描述

配置项	参数格式	描述	备注
name	字符串	必填,对存储 设备自定义的 名称。	自定义字符串,支持大小写字母、数 字、中划线组合。
storage	字符串	必填,待接入 存储设备的类 型。	对接分布式存储NAS场景固定填写 "fusionstorage-nas"。
pools	列表	必填,被使用 的待接入存储 设备上的存储 池名称。	支持同一存储设备上的一个或多个存储池, 用逗号分割。 可通过登录DeviceManager获取存储池。
urls	列表	必填,待接入 存储设备的管 理URL。	支持同一存储设备的一个或多个管理 URL,用逗号分割,当前仅支持IPv4。 例如: https://192.168.125.20:28443 说明 1个存储设备有多个控制器,每个控制器有 一个管理URL,所以1个存储设备存在多个 管理URL。
paramete rs	字典	必填,NAS场景 的可变参数。	portals: 指定存储的逻辑端口IP,可通 过登录DeviceManager获取,当前只 支持配置一个。

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

kubectl create -f huawei-csi-configmap.yaml

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

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

----结束

4.7 启动 huawei-csi 服务

本章节介绍如何启动huawei-csi服务。

注意事项

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

前提条件

- 已向管理员获取存储的用户名和密码。
- 至少需要角色为管理员且级别为管理员的存储本地用户。

操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。
- **步骤2** 将Kubernetes CSI组件包中的secretGenerate工具拷贝到master节点的任意目录下,工具路径参见**4.2 软件包组件**。
- 步骤3 使用加密工具填写存储设备的用户名和密码。
 - 分别执行chmod +x secretGenerate和./secretGenerate命令,根据界面提示输入相关信息。

chmod +x secretGenerate
./secretGenerate
********The 1 Racke

Current backend name is: <backend-1-name>
Current backend url is: [<backend-1-url>]

Enter backend <backend-1-name>'s user: #请填写存储1的用户名
Enter backend <backend-1-name>'s password: #请填写存储1的密码
Please Enter the password again: #再次填写存储1的密码

2. 配置完成后,使用kubectl get secret -n kube-system | grep huawei-csi-secret命令检查。

kubectl get secret -n kube-system | grep huawei-csi-secret huawei-csi-secret Opaque 1

步骤4 执行**vi** *huawei-csi-rbac.yaml*命令,修改yaml文件。按**l**或**Insert**进入编辑状态,修改相关参数。修改完成后,按**Esc**,并输入**:wq!**,保存修改。(可参考软件包中yamls/huawei-csi-rbac.yaml示例文件)。

8m

□ 说明

• csi-resizer服务从Kubernetes v1.16开始支持
huawei-csi-rbac.yaml文件请参考软件包中yamls/huawei-csi-resize-rbac.yaml示例文件

csi-snapshotter服务从Kubernetes v1.17开始支持
 huawei-csi-rbac.yaml文件请参考软件包中yamls/huawei-csi-resize-snapshot-rbac.yaml示例文件

步骤5 执行以下命令,创建RBAC权限。

kubectl create -f huawei-csi-rbac.yaml

步骤6 执行**vi** *huawei-csi-controller.yaml*命令,修改yaml文件。按**I**或**Insert**进入编辑状态, 修改相关参数。修改完成后,按**Esc**,并输入:**wq!**,保存修改。(可参考软件包中 yamls/huawei-csi-controller.yaml示例文件)。

□ 说明

- 示例yaml文件中huawei-csi:*.**配置项, *.*.*应替换为前面制作的华为CSI镜像版本号。
- 示例yaml文件中liveness-probe配置参数args中: --health-port表示监听端口号,如需修改,请参考如下。
 - 1. 在liveness-probe配置参数args中,修改health-port字段。
 - --csi-address=/var/lib/csi/sockets/pluginproxy/csi.sock
 - --health-port=9808
 - 2. 在huawei-csi-driver配置参数ports中,修改containerPort字段。ports:
 - containerPort: 9808 name: healthz protocol: TCP
- csi-resizer服务从Kubernetes v1.16开始支持

huawei-csi-controller.yaml文件请参考软件包中yamls/huawei-csi-resize-controller.yaml示例文件

◆ csi-snapshotter服务从Kubernetes v1.17开始支持

huawei-csi-controller.yaml文件请参考软件包中yamls/huawei-csi-resize-snapshot-controller.yaml示例文件

步骤7 执行以下命令,启动controller服务。

kubectl create -f huawei-csi-controller.yaml

步骤8 执行**vi** *huawei-csi-node.yaml*命令,修改yaml文件。按**I**或**Insert**进入编辑状态,修改相关参数。修改完成后,按**Esc**,并输入:**wq!**,保存修改。编写huawei-csi-node.yaml文件(可参考软件包中yamls/huawei-csi-node.yaml示例文件)。

□ 说明

- 示例yaml文件中huawei-csi:*.*:配置项, *.*.*应替换为前面制作的华为CSI镜像版本号。
- 示例yaml文件中liveness-probe配置参数args中: --health-port表示监听端口号,如需修改,请参考如下。
 - 1. 在liveness-probe配置参数args,修改health-port字段。

aras:

- --csi-address=/var/lib/csi/sockets/pluginproxy/csi.sock
- --health-port=9800
- 2. 在huawei-csi-driver配置参数ports,修改containerPort字段。ports:
 - containerPort: 9800 name: healthz protocol: TCP
- 示例yaml文件中huawei-csi-driver配置参数args: --volume-use-multipath 表示默认开启多路径,如需修改,请参考如下。

args:

- "--endpoint=/csi/csi.sock"
- "--containerized"
- "--driver-name=csi.huawei.com"
- "--volume-use-multipath=false"
- 步骤9 执行以下命令,启动node服务。

kubectl create -f huawei-csi-node.yaml

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

```
# kubectl get pod -A | grep huawei
kube-system huawei-csi-controller-695b84b4d8-tg64l 3/3 Running 0 14s
kube-system huawei-csi-node-g6f7z 2/2 Running 0 14s
```

----结束

5 升级操作

须知

- CSI升级过程中对已经下发的PVC/Snapshot/Pod等资源没有影响。
- 在升级过程中,不能使用CSI下发新的资源。
- 5.1 卸载原CSI
- 5.2 安装新CSI

5.1 卸载原 CSI

当您需要卸载CSI时,执行此操作。

前提条件

在卸载CSI前,请先执行**kubectl get configmap huawei-csi-configmap -n kube-system -o yaml >> huawei-csi-configmap.yaml.bak**命令,备份huawei-csi-configmap的内容。

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,删除huawei-csi-node服务(huawei-csi-node.yaml为**4.7 启动** huawei-csi服务中步骤8的配置信息)。

kubectl delete -f huawei-csi-node.yaml

- 步骤3 执行以下命令,查看huawei-csi-node服务,如果没有回显,表示删除完成。 # kubectl get pod -A | grep huawei-csi-node
- 步骤4 执行以下命令,删除huawei-csi-controller服务(huawei-csi-controller.yaml为4.7 启动huawei-csi服务中步骤6的配置信息)。

kubectl delete -f huawei-csi-controller.yaml

步骤5 执行以下命令,查看huawei-csi-controller服务,如果没有回显,表示删除完成。 # kubectl get pod -A | grep huawei-csi-controller **步骤6** 执行以下命令,删除RBAC权限(huawei-csi-rbac.yaml为**4.7 启动huawei-csi服务**中 **步骤4**的配置信息)。

kubectl delete -f huawei-csi-rbac.yaml

步骤7 执行以下命令,删除huawei-csi-configmap对象。当前huawei-csi-configmap对象默认"<namespace>"参数为"kube-system",请根据实际名称进行替换。

kubectl delete configmap huawei-csi-configmap -n <namespace>

步骤8 执行以下命令,查看huawei-csi的configmap信息,如果没有回显,表示删除完成。 # kubectl get configmap -A | grep huawei-csi-configmap

步骤9 执行以下命令,删除huawei-csi-secret对象。当前huawei-csi-secret对象默认 "<namespace>"参数为"kube-system",请根据实际名称进行替换。

kubectl delete secret huawei-csi-secret -n <namespace>

步骤10 执行以下命令,查看huawei-csi的secret信息,如果没有回显,表示删除完成。

kubectl get secret -A | grep huawei-csi-secret

----结束

5.2 安装新 CSI

卸载完成后需要重新安装CSI。

前提条件

已备份huawei-csi-configmap.yaml的内容。

注意事项

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

- "storage","name","pools"参数与5.1 <mark>卸载原CSI</mark>的**前提条件**中已备份的huawei-csi-configmap.yaml.bak文件保持一致。
- "urls"和"parameters"参数请参考5.1 卸载原CSI的前提条件中已备份的 huawei-csi-configmap.yaml.bak文件中的信息,按照当前版本的huawei-csiconfigmap.yaml模板填写,模板详细请参考4.5 对接企业存储配置和4.6 对接分布 式存储配置。以下回显仅为示例。

```
"backends": [

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

操作步骤

步骤1 安装CSI, 详细操作可参见4 安装部署。

----结束

6 使用说明

本章节主要介绍Kubernetes对接华为存储后,如何管理StorageClass、PVC、Pod以及快照。

- 6.1 管理StorageClass
- 6.2 管理PVC
- 6.3 管理Pod
- 6.4 (可选)管理快照

6.1 管理 StorageClass

6.1.1 创建 StorageClass

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

6.1.1.1 创建 LUN StorageClass

本章节介绍如何创建LUN StorageClass。

操作步骤

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

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

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

表 6-1 参数说明

参数	说明	备注
name	自定义的 StorageClass对 象名称	
provisioner	provisioner标识	固定填写"csi.huawei.com"
volumeType	待创建卷类型	固定填写"lun"
allocType	待创建卷的分配 类型	可选,支持thin/thick,默认为thin
cloneSpeed	指定克隆速度	可选,默认值3,支持1~4,4速度最快。 配置克隆PVC或从快照创建PVC时生效, 参考6.2.3 (可选)克隆PVC或6.2.4 (可选)从快照创建PVC
fsType	指定文件系统类 型	可选,支持ext2/ext3/ext4,默认为ext4

步骤4 执行以下命令,基于该yaml文件创建StorageClass。

kubectl create -f StorageClass.yaml

步骤5 执行以下命令,查看当前已经创建的StorageClass信息。

kubectl get sc NAME PROVISIONER AGE mysc csi.huawei.com 1h

----结束

6.1.1.2 创建文件系统 StorageClass

本章节介绍如何创建文件系统StorageClass。

操作步骤

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

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

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

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

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

表 6-2 参数说明

参数	说明	备注
name	自定义的StorageClass对象 名称	
provisioner	provisioner标识	固定填写"csi.huawei.com"
volumeTyp e	待创建卷类型	固定填写"fs"
authClient	指定可访问该卷的客户端	必选,支持输入客户端主机名称、客户端IP地址、客户端IP地址段或使用"*"表示全部客户端IP地址 支持指定多个客户端,以";"分号分隔
allocType	待创建卷的分配类型	可选,支持thin/thick,默认为thin
cloneSpee d	指定克隆速度	可选,默认值3,支持1~4,4速度最快。 配置克隆PVC或从快照创建PVC时生效, 参考6.2.3 (可选)克隆PVC或6.2.4 (可选)从快照创建PVC

步骤4 执行以下命令,基于该yaml文件创建StorageClass。

kubectl create -f StorageClass.yaml

步骤5 执行以下命令,查看当前已经创建的StorageClass信息。

kubectl get sc NAME PROVISIONER AGE mysc csi.huawei.com 1h

----结束

6.1.2 删除 StorageClass

本章节介绍如何删除StorageClass。

操作步骤

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

步骤2 执行以下命令, 查询集群中的StorageClass。

kubectl get sc NAME PROVISIONER AGE huawei-nas csi.huawei.com 3s mysc csi.huawei.com 16s 步骤3 执行以下命令, 删除StorageClass。例如删除名称为mysc的StorageClass。

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

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

kubectl get sc NAME PROVISIONER AGE huawei-nas csi.huawei.com 11s

----结束

6.2 管理 PVC

6.2.1 创建 PVC

当您需要创建PVC时,执行此操作。

操作步骤

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

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

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

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

kind: PersistentVolumeClaim apiVersion: v1 metadata: name: "mypvc" spec: accessModes: - ReadWriteMany storageClassName: "mysc" resources: requests: storage: 100Gi

表 6-3 参数说明

参数	说明	备注
name	自定义的PVC对象名称	
storageClas sName	StorageClass对象名称	填写 6.1 管理StorageClass 创建的 StorageClass对象名称。

参数	说明	备注
storage	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和主机规格限制,以OceanStor Dorado V6 6.1.2/OceanStor Pacific系列 8.1.0对接CentOS 7为例,示例表6-4。 其余存储和主机请根据StorageClass中"VolumeType"类型进行规格查看。 如果"volumeType"类型为"lun",需要参考存储自身规格,详情请参考https://info.support.huawei.com/storage/spec/#/home。同时还需要主机连通性指南,详情请参考https://support.huawei.com/enterprise/zh/doc/EDOC1100112792/87e2c8fd。 如果"volumeType"类型为"fs",参考存储自身支持的规格,详情请参考https://info.support.huawei.com/storage/spec/#/home。 如果"volumeType"类型为"fs",参考存储自身支持的规格,详情请参考https://info.support.huawei.com/storage/spec/#/home。
accessMode s	指定卷访问模式	LUN卷支持ReadWriteOnce, ReadOnlyMany和 ReadWriteMany,如果使用 ReadWriteMany模式且有多个Pod同 时访问该卷,需要由Pod业务保证数 据一致性 文件系统卷支持ReadWriteOnce, ReadOnlyMany和 ReadWriteMany。

表 6-4 PVC 容量的规格

volumeT ype类型	存储类型	存储规格限制	ext4规格限 制	CSI规格限制
lun	OceanStor Dorado V6 6.1.2	512Ki~256Ti	50Ti	512Ki~50Ti

volumeT ype类型	存储类型	存储规格限制	ext4规格限 制	CSI规格限制
	OceanStor Pacific系 列 8.1.0	64Mi~512Ti	50Ti	64Mi~50Ti
fs	OceanStor Dorado V6 6.1.2	1Gi~32Pi	N/A	1Gi~32Pi
	OceanStor Pacific系 列 8.1.0	1Ki~256Pi	N/A	1Ki~256Pi

步骤4 执行以下命令,基于该yaml文件创建PVC。

kubectl create -f PersistentVolumeClaim.yaml

步骤5 等待一段时间后,执行以下命令,查看已经创建的PVC信息。

kubectl get pvc

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

□说明

完成创建PVC操作后,如果PVC的状态是Pending,请参考**9.6 创建PVC时, PVC的状态为 Pending** 。

----结束

6.2.2 (可选)扩容 PVC

6.2.2.1 安装扩容依赖组件服务

按照**4.7 启动huawei-csi服务**进行配置。使用huawei-csi-resize-snapshot-controller.yaml部署huawei-csi服务,使用以下命令查看huawei-csi服务是否启动:

kubectl get pod -A | grep huawei

kube-system **huawei-csi-controller**-fd5f97768-qlldc **6/6** Running 0 16s kube-system huawei-csi-node-25txd 2/2 Running 0 15s

6.2.2.2 扩容 PVC

本章节介绍如何扩容PVC。

前提条件

已创建PVC,详细请参见6.2.1 创建PVC。

操作步骤

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

步骤2 执行vi StorageClass.yaml命令,编辑**新的**StorageClass.yaml文件。在文件中添加 allowVolumeExpansion配置项,示例如下:

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

步骤3 创建StorageClass,详细请参见**6.1 管理StorageClass**。如果已有同名StorageClass存在,请重新指定StorageClass.yaml的name。

步骤4 创建PVC,详细请参见**6.2.1 创建PVC**。如果已有同名PVC存在,请重新指定 PersistentVolumeClaim.yaml的name。

步骤5 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 V6 6.1.2/ OceanStor Pacific系列 8.1.0对接CentOS 7为例,示例表6-5。
- 其余存储和主机请根据StorageClass中"VolumeType"类型进行规格查看。
 - 如果 "volumeType" 类型为 "lun", 需要参考存储自身规格,详情请参考https://info.support.huawei.com/storage/spec/#/home。同时还需要主机连通性指南,详情请参考https://support.huawei.com/enterprise/zh/doc/EDOC1100112792/87e2c8fd。
 - 如果 "volumeType" 类型为 "fs",参考存储自身支持的规格,详情请参考https://info.support.huawei.com/storage/spec/#/home。
- 如果PVC容量不在规格范围内,可能会由于存储规格限制或主机文件系统规格限制导致创建 PVC或Pod失败。

表 6-5 PVC 容量的规格

volumeT ype类型	存储类型	存储规格限制	ext4规格限 制	CSI规格限制
lun	OceanStor Dorado V6 6.1.2	512Ki~256Ti	50Ti	512Ki~50Ti
	OceanStor Pacific系 列 8.1.0	64Mi~512Ti	50Ti	64Mi~50Ti
fs	OceanStor Dorado V6 6.1.2	1Gi~32Pi	N/A	1Gi~32Pi
	OceanStor Pacific系 列 8.1.0	1Ki~256Pi	N/A	1Ki~256Pi

步骤6 执行以下命令,查看容量是否发生改变。

kubectl get pvc

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

----结束

6.2.3 (可选)克隆 PVC

当您需要克隆Kubernetes上已有的PVC时,执行此操作。

操作步骤

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

步骤2 配置PersistentVolumeClaim.yaml文件。PersistentVolumeClaim.yaml文件的模板如回显所示。请根据实际情况设置相关参数,并保存为yaml格式的文件,详细请参考表6-6。

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
name: ***
spec:
storageClassName: ***
dataSource:
name: ***
kind: PersistentVolumeClaim
accessModes:
- ReadWriteMany
resources:
requests:
storage: ***Gi
```

表 6-6 参数说明

参数	说明	备注
metadata/ name	自定义的新建PVC对象名称	
spec/ storageClas sName	StorageClass对象名称	填写 6.1 管理StorageClass 创建的 StorageClass对象名称,与 dataSource中的对象的StorageClass 需一致
spec/ dataSource/ name	源PVC对象名称	
spec/ resources/ requests/ storage	指定待创建新卷大小	不小于源PVC,格式为***Gi,单位为 GiB

步骤3 执行**vi** *PersistentVolumeClaim.yaml*命令,修改yaml文件。按**l**或**Insert**进入编辑状态,修改相关参数。修改完成后,按**Esc**,并输入:**wq!**,保存修改。

步骤4 执行以下命令,基于该yaml文件创建PVC。

kubectl create -f PersistentVolumeClaim.yaml

----结束

6.2.4 (可选)从快照创建 PVC

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

前提条件

已经创建Snapshot,详细请参考6.4 (可选)管理快照。

操作步骤

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

步骤2 配置PersistentVolumeClaim.yaml文件。PersistentVolumeClaim.yaml文件的模板如回显所示。请根据实际情况设置相关参数,并保存为yaml格式的文件,详细请参考表6-7。

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

表 6-7 参数说明

参数	说明	备注
metadata/ name	自定义的新建PVC对象名称	
spec/ storageClas sName	StorageClass对象名称	填写 6.1 管理StorageClass 创建的 StorageClass对象名称,与 dataSource中快照的原PVC的 StorageClass需一致
spec/ dataSource/ name	源VolumeSnapshot对象名称	
spec/ resources/ requests/ storage	指定待创建新卷大小	不小于源VolumeSnapshot,格式为 ***Gi,单位为GiB

步骤3 执行vi PersistentVolumeClaim.yaml命令,修改yaml文件。按l或Insert进入编辑状态,修改相关参数。修改完成后,按Esc,并输入:wq!,保存修改。

步骤4 执行以下命令,基于该yaml文件创建PVC

kubectl create -f PersistentVolumeClaim.yaml

-----结束

6.2.5 删除 PVC

本章节介绍如何删除PVC。

操作步骤

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

步骤2 执行以下命令,查询集群中的PVC。

kubectl get pvc

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

山 说明

删除PVC前,如果PVC的状态是Pending,不建议直接删除,如果需要删除,请参考**9.7 删除PVC**前,PVC的状态为Pending。

步骤3 执行以下命令,删除PVC。例如删除名称为mypvc的PVC。

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

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

kubectl get pvc

No resources found in default namespace.

----结束

6.3 管理 Pod

6.3.1 创建 Pod

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

操作步骤

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

步骤2 执行vi Pod.yaml命令,创建Pod.yaml文件

vi Pod.yaml

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

Pod.yaml文件的模板如回显所示(您也可以参考软件包中yamls/pod-example.yaml示例文件)。请根据实际情况设置相关参数,并保存为yaml格式的文件,详细参考<mark>表6-8</mark> kind: Pod apiVersion: v1 metadata:

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

表 6-8 参数说明

参数	说明	备注
metadata:name	自定义的Pod对象名称	
spec:containers:name	自定义的容器名称	
spec:containers:image	指定容器镜像	镜像根据实际情况填 写
spec:containers:image:volumeMo unts:mountPath	容器内卷的挂载路径	
spec:volumes:persistentVolumeCl aim:claimName	PVC对象名称	填写 6.2.1 创建PVC 创建的PVC对象名称

步骤4 执行以下命令,基于该yaml文件创建Pod.

kubectl create -f Pod.yaml

步骤5 执行以下命令,查看已经创建的Pod.

kubectl get pod

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

----结束

6.3.2 删除 Pod

本章节介绍如何删除Pod。

操作步骤

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

步骤2 执行以下命令,查询集群中的Pod。

kubectl get pod

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

步骤3 执行以下命令,删除Pod。例如删除名称为mypod的Pod。

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

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

kubectl get pod No resources found in default namespace.

----结束

6.4 (可选)管理快照

从Kubernetes 的v1.17 版本开始,CSI支持快照的v1beta1版本。详情请查看: https://kubernetes-csi.github.io/docs/external-snapshotter.html

6.4.1 安装 Snapshot 依赖组件服务

以下步骤仅需要在任意master节点上执行。

前提条件

安装前需要确认是否已经安装snapshot相关资源服务,在master节点上使用**kubectl** api-resources | grep snapshot | awk '{print \$1}'命令,如果回显如下,则不需要再次安装。

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

如果结果不包含上述服务,请按照以下指导进行安装

步骤1 如果缺少volumesnapshotclasses资源,编写volumesnapshotclasses.yaml文件(可参考软件包中yamls/snapshot.storage.k8s.io_volumesnapshotclasses.yaml示例文件)。

kubectl create -f volumesnapshotclasses.yaml

步骤2 如果缺少volumesnapshotcontents资源,编写volumesnapshotcontents.yaml文件(可参考软件包中yamls/snapshot.storage.k8s.io_volumesnapshotcontents.yaml示例文件)。

kubectl create -f volumesnapshotcontents.yaml

步骤3 如果缺少volumesnapshots资源,编写volumesnapshots.yaml文件(可参考软件包中 yamls/snapshot.storage.k8s.io_volumesnapshots.yaml示例文件)。
kubectl create -f volumesnapshots.yaml

步骤4 如果已经启动huawei-csi服务,请先按照8.1 卸载CSI进行配置。

步骤5 按照4.7 启动huawei-csi服务进行配置。使用huawei-csi-resize-snapshot-controller.yaml部署huawei-csi服务,使用以下命令查看huawei-csi服务是否启动:

kubectl get pod -A | grep huawei kube-system huawei-csi-controller-fd5f97768-qlldc 6/6 Running 0 16s kube-system huawei-csi-node-25txd 2/2 Running 0 15s

----结束

6.4.2 管理 VolumeSnapshotClass

6.4.2.1 创建 VolumeSnapshotClass

本章节介绍如何创建VolumeSnapshotClass。

操作步骤

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

步骤2 执行vi VolumeSnapshotClass.yaml命令,创建VolumeSnapshotClass.yaml文件 # vi VolumeSnapshotClass.yaml

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

VolumeSnapshotClass.yaml文件的模板如回显所示(您也可以参考软件包中yamls/snapshotclass.yaml示例文件)。请根据实际情况设置相关参数,并保存为yaml格式的文件,详细参考表6-9

apiVersion: snapshot.storage.k8s.io/v1beta1

kind: VolumeSnapshotClass

metadata:

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

表 6-9 参数说明

参数	说明	备注
name	自定义的 VolumeSnapshotClass对 象名称	
driver	driver标识	固定填写"csi.huawei.com"
deletionPo licy	VolumeSnapshot删除时, 处理 VolumeSnapshotContent 策略	必填, 可选值为Delete或Retain

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

kubectl create -f VolumeSnapshotClass.yaml

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

kubectl get volumesnapshotclass

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

----结束

6.4.2.2 删除 VolumeSnapshotClass

本章节介绍如何删除VolumeSnapshotClass。

操作步骤

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

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

kubectl get volumesnapshotclass

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

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

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

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

kubectl get volumesnapshotclass No resources found in default namespace.

----结束

6.4.3 管理 VolumeSnapshot

6.4.3.1 创建 VolumeSnapshot

本章节介绍如何创建VolumeSnapshot。

操作步骤

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

步骤2 执行vi VolumeSnapshot.yaml命令, 创建VolumeSnapshot.yaml文件 # vi VolumeSnapshot.yaml

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

VolumeSnapshot.yaml文件的模板如回显所示(您也可以参考软件包中yamls/snapshot.yaml示例文件)。请根据实际情况设置相关参数,并保存为yaml格式的文件,详细参考表6-10

apiVersion: snapshot.storage.k8s.io/v1beta1 kind: VolumeSnapshot

metadata:

name: mysnapshot

spec:

volumeSnapshotClassName: mysnapclass

source:

persistentVolumeClaimName: mypvc

表 6-10 参数说明

参数	说明	备注
name	自定义的VolumeSnapshot对 象名称	
volumeSna pshotClass Name	VolumeSnapshotClass对象名 称	填写 6.4.2.1 创建 VolumeSnapshotClass 创建的 VolumeSnapshotClass对象名称

参数	说明	备注
persistentV olumeClaim Name	源PVC对象名称	填写 6.2.1 创建PVC 创建的PVC对象 名称

步骤4 执行以下命令,基于该yaml文件创建VolumeSnapshot。

kubectl create -f VolumeSnapshot.yaml

步骤5 执行以下命令,查看已创建的VolumeSnapshot信息。

kubectl get volumesnapshot

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

----结束

6.4.3.2 删除 VolumeSnapshot

本章节介绍如何删除VolumeSnapshot。

操作步骤

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

步骤2 执行以下命令,查询集群中的VolumeSnapshot。

kubectl get volumesnapshot

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

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

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

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

kubectl get volumesnapshot

No resources found in default namespace.

----结束

了 高级特性

本章节介绍如何配置华为存储的高级特性。

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

7.1 配置多个后端

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

操作步骤

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

步骤2 配置huawei-csi-configmap.yaml文件, huawei-csi-configmap.yaml文件的模板如回显所示。请根据实际情况设置相关参数,并保存为yaml格式的文件。

多个后端间使用","进行分割,每个后端内容可以参考**4.5 对接企业存储配置**或**4.6 对接分布式存储配置**。

```
...
},
{
    "storage": "***",
    "name": "backend2",
    ...
}
]
```

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

kubectl create -f huawei-csi-configmap.yaml

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

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

步骤5 启动huawei-csi服务,详细请参见4.7 启动huawei-csi服务。

----结束

7.2 指定后端创建 PVC

配置多后端场景后,可以通过以下操作为指定后端创建PVC。

操作步骤

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

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

- "parameters"配置项下添加backend配置项。
- backend参数值是huawei-csi-configmap.yaml后端配置中的某个后端名称。

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

步骤3 执行以下命令,基于该yaml文件创建StorageClass。

kubectl create -f StorageClass.yaml

----结束

7.3 指定存储池创建 PVC

配置多存储池场景后,可以通过以下操作为指定存储池创建PVC。

操作步骤

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

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

- "parameters"配置项下添加backend配置项。
- "pool"参数值是huawei-csi-configmap.yaml后端配置中的某个存储池名称。

□ 说明

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

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

步骤3 执行以下命令,基于该yaml文件创建StorageClass。

kubectl create -f StorageClass.yaml

----结束

7.4 配置 ALUA 特性

7.4.1 配置 OceanStor V3/V5 系列和 OceanStor Dorado V3 系列 ALUA

在对接块存储时,如果有使用多路径,可以通过本章节配置ALUA。

操作步骤

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

"parameters"配置项下增加ALUA配置,参数配置请参见表7-1。

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

表 7-1 ALUA 参数说明

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

□ 说明

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

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

配置策略的规则:

- 优先级: 主机名一般配置 > 主机名通用配置(详见10.1 OceanStor V3/V5**系列和** OceanStor Dorado V3**系列ALUA特性配置策略样例**例1)
- 在一般配置中,采用满足配置策略的第一条ALUA配置段(详见10.1 OceanStor V3/V5 系列和OceanStor Dorado V3系列ALUA特性配置策略样例例2)
- 在一般配置中,如果需要精确匹配某个主机,参考10.1 OceanStor V3/V5**系列和** OceanStor Dorado V3**系列ALUA特性配置策略样例**例3
- OceanStor V3/V5系列和OceanStor Dorado V3系列使用此种配置方式,详细参数请参见表 7-2

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

使用场景	主机类型	是否为优 选路径存 储	ALUA配置推荐值
双活存储 场景	CentOS/RHEL系列的主机	优选路径 存储	ALUA="1" FAILOVERMODE="3" SPECIALMODETYPE="0" PATHTYPE="0"
		非优选路 径存储	ALUA="1" FAILOVERMODE="3" SPECIALMODETYPE="0" PATHTYPE="1"
	Ubuntu/SUSE/ Debian系列的主 机	优选路径 存储	ALUA="1" FAILOVERMODE="1" PATHTYPE="0"
7		非优选路 径存储	ALUA="1" FAILOVERMODE="1" PATHTYPE="1"
非双活存 储场景	CentOS/RHEL系 列的主机	N/A	ALUA="1" FAILOVERMODE="3" SPECIALMODETYPE="0" PATHTYPE="0"
	Ubuntu/SUSE/ Debian系列的主 机	N/A	ALUA="1" FAILOVERMODE="1" PATHTYPE="0"

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

kubectl create -f huawei-csi-configmap.yaml

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

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

步骤5 启动huawei-csi服务,详细请参见4.7 启动huawei-csi服务。

----结束

7.4.2 配置 OceanStor Dorado V6 ALUA

在对接块存储时,如果有使用多路径,可以通过本章节配置ALUA。

操作步骤

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

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

"parameters"配置项下增加ALUA配置,参数配置请参见表7-3。

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

表 7-3 ALUA 参数说明

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

□ 说明

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

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

配置策略的规则:

- 优先级: 主机名一般配置 > 主机名通用配置 (详见10.2 OceanStor Dorado V6 ALUA特性配置策略样例例1)
- 在一般配置中,采用满足配置策略的第一条ALUA配置段(详见**10.2 OceanStor Dorado V6 ALUA特性配置策略样例**例2)
- 在一般配置中,如果需要精确匹配某个主机,参考10.2 OceanStor Dorado V6 ALUA 特性配置策略样例例3
- 当主机仅使用OceanStor Dorado V6固态存储,详细参数请参见表7-4。
- 当同一主机先使用OceanStor融合存储,再新增OceanStor Dorado V6固态存储时,详细参数请参见表7-5。

表 7-4 OceanStor Dorado V6 固态存储推荐 ALUA 参数配置

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

表 7-5 OceanStor V3/V5、OceanStor Dorado V3、OceanStor Dorado V6 存储共存场景推荐 ALUA 参数配置

使用场景	主机类型	主机访问模式	ALUA配置推荐值
双活存 储场景	CentOS/RHEL/Ubuntu/ SUSE/Debian系列的主 机	负载均衡模式	ACCESSMODE="1" HYPERMETROPATHOPTIMIZ ED="1"

使用场景	主机类型	主机访问模式	ALUA配置推荐值
		非对称模式 +优选路径存 储	ACCESSMODE="1" HYPERMETROPATHOPTIMIZ ED="1"
		非对称模式 +非优选路径 存储	ACCESSMODE="1" HYPERMETROPATHOPTIMIZ ED="0"
非双活 存储场 景	CentOS/RHEL/Ubuntu/ SUSE/Debian系列的主 机	N/A	ACCESSMODE="1" HYPERMETROPATHOPTIMIZ ED="1"

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

kubectl create -f huawei-csi-configmap.yaml

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

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

步骤5 启动huawei-csi服务,详细请参见4.7 启动huawei-csi服务。

----结束

7.4.3 配置分布式存储 ALUA

在对接块存储时,如果有使用多路径,可以通过本章节配置ALUA。

操作步骤

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

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

"parameters"配置项下增加ALUA配置,参数配置请参见表7-6。

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

表 7-6 ALUA 参数说明

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

□ 说明

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

配置策略的规则:

- 优先级: 主机名一般配置 > 主机名通用配置 (详见10.3 分布式存储ALUA特性配置策略样例例1)
- 在一般配置中,采用满足配置策略的第一条ALUA配置段(详见**10.3 分布式存储ALUA** 特性配置策略样例例2)
- 在一般配置中,如果需要精确匹配某个主机,参考10.3 分布式存储ALUA特性配置策略 样例例3
- **步骤3** 执行**kubectl create -f** *huawei-csi-configmap.yaml*命令,创建huawei-csi-configmap。

kubectl create -f huawei-csi-configmap.yaml

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

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

步骤5 启动huawei-csi服务,详细请参见4.7 启动huawei-csi服务。

----结束

7.5 配置存储拓扑感知

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

前提条件

- Kubernetes 的v1.17 版本开始, 支持拓扑感知特性。
- 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:
 - 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 v1.19.3 42d v1.19.3 node02 Ready worker node03 Ready worker 42d v1.19.3

执行kubectl label node <nodename> topology.kubernetes.io/ <key>=<value>命令,给worker节点配置拓扑标签。其中<nodename>为 worker节点名称, key和value参数说明请参考表7-7。 # kubectl label node node01 topology.kubernetes.io/zone=ChengDu

node/node01 labeled

表 7-7 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><pre>protocol></pre></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。

d. 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/hostname":"node01","kubernetes.io/os":"linux","node-role.kubernetes.io/controlplane":"true","node-role.kubernetes.io/controlplane":"true","topology.kubernetes.io/zone":"ChengDu"}]

操作步骤

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

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

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

□ 说明

- "supportedTopologies"是一个列表,列表中的每个元素是一个字典。
- 列表中的每个元素只支持配置"topology.kubernetes.io/region"或 "topology.kubernetes.io/zone",参数值需要保持和**前提条件中设置的拓扑标签**一致。 (不支持配置"topology.kubernetes.io/protocol.<protocol>")
- **步骤3** 执行**kubectl create -f** *huawei-csi-configmap.yaml*命令,创建huawei-csi-configmap。

kubectl create -f huawei-csi-configmap.yaml

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

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

步骤5 启动huawei-csi服务,详细请参见4.7 启动huawei-csi服务。

步骤6 执行**vi** *StorageClass.yaml*命令,修改yaml文件。按**I**或**Insert**进入编辑状态,在yaml 文件下增加相关参数,详细参数说明请参见表7-8。修改完成后,按**Esc**,并输入:**wq!**,保存修改。

在StorageClass.yaml文件中添加以下配置项。

 示例1: 在StorageClass中配置zone和region信息 kind: StorageClass

```
apiVersion: storage.k8s.io/v1
metadata:
name: example-storageclass
```

provisioner: csi.huawei.com

parameters: volumeType: lun

allocType: thin
volumeBindingMode: WaitForFirstConsumer allowedTopologies:

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

示例2: 在StorageClass中配置协议信息

kind: StorageClass

apiVersion: storage.k8s.io/v1

metadata:

name: protocol-example-storageclass

provisioner: csi.huawei.com

parameters: volumeType: lun allocType: thin

volumeBindingMode: WaitForFirstConsumer allowedTopologies:

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

表 7-8 参数说明

参数名	参数描述	备注
volume Binding Mode	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。
allowed Topologi es/ matchLa belExpre ssions	pologi CSI后端和Kubernetes节 / 点。如果匹配失败,会导 atchLa 致PVC或Pod无法创建。 elExpre 配置时需要同时按照固定	"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。
	"value".	"value": "key"如果是"topology.kubernetes.io/ zone","topology.kubernetes.io/ region","value"值需要和 前提条件中设 置的拓扑标签保持一致。 "key"如果是topology.kubernetes.io/ protocol. region" "value"值固定为 "csi.huawei.com"

步骤7 执行以下命令,基于该yaml文件创建StorageClass。

kubectl create -f StorgeClass.yaml

步骤8 使用该StorageClass创建具有拓扑能力的PVC,详细操作请参考6.2.1 创建PVC。

步骤9 使用该PVC创建Pod,详细操作请参考6.3.1 创建Pod。

----结束

7.6 企业存储高级特性

7.6.1 配置 QoS

本章节介绍如何创建支持QoS的LUN/文件系统卷。

注意事项

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

操作步骤

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

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

- "parameters"配置项下添加qos配置项。
- qos配置项值是字典格式的json字符串(字符串两边由单引号修饰,字典key由双引号修饰),详细参数说明请参见表7-9。

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

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

表 7-9 qos 支持的参数说明

参数名	参数描述	备注
IOTYPE	控制读写类型	可选(未明确指定将使用后端存储 默认值,具体参考相关存储资料) 有效值如下: • 0: 读IO • 1: 写IO • 2: 读写IO
MAXBANDWIDTH	最大带宽限制策略	单位MB/s,有效值为>0的整数

参数名	参数描述	备注
MINBANDWIDTH	最小带宽保护策略	单位MB/s,有效值为>0的整数
MAXIOPS	最大IOPS限制策略	有效值为>0的整数
MINIOPS	最小IOPS保护策略	有效值为>0的整数
LATENCY	最大时延保护策略	单位ms,有效值为>0的整数

□ 说明

- 同属保护策略或者同属限制策略的多个参数可以同时指定,但是保护策略和限制策略参数不能同时指定。
- OceanStor Dorado存储仅支持IOTYPE设置为2(即读写IO),仅支持限制策略参数。
- 租户用户不支持配置QoS策略。
- 配置QoS后只能在新建的PVC上生效;对于同名StorageClass已经发放的PVC,不能自动添加 OoS

步骤3 执行以下命令,基于该yaml文件创建StorageClass。

kubectl create -f StorgeClass.yaml

步骤4 使用该StorageClass创建具有QoS能力的PVC,详细操作请参考6.2.1 创建PVC。

----结束

7.6.2 配置租户

操作步骤

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

在某个后端配置中增加vstoreName参数,vstoreName为存储设备上的租户名称。

```
{
    "backends": [
    {
        ...
        "vstoreName": "***"
    }
    ]
}
```

□ 说明

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

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

kubectl create -f huawei-csi-configmap.yaml

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

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

步骤5 启动huawei-csi服务,详细请参见4.7 启动huawei-csi服务。

□ 说明

启动huawei-csi服务时,**4.7 启动huawei-csi服务**中的**步骤3**请填写存储设备租户的用户名和密码。

----结束

7.6.3 配置 SAN 远程复制

当您需要配置SAN远程复制时,执行此操作。

注意事项

使用SAN远程复制,需要事先在两台存储之间完成远程复制关系配置。具体请参考华 为存储相关配置指导。

操作步骤

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

在huawei-csi-configmap.yaml文件的backends配置段中,添加构成复制关系的两个后端,并且为每个后端增加replicaBackend配置项。

```
{
  "backends":[
  {
    ...
    "name": "replica1",
    "replicaBackend": "replica2"
  },
  {
    ...
    "name": "replica2",
    "replicaBackend": "replica1"
  }
}
```

山 说明

replicaBackend是远程复制对端的后端名,两个复制后端相互构成复制关系。如上所示,后端replica1的远程复制对端是replica2,相应的,replica2的远程复制对端是replica1。

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

kubectl create -f huawei-csi-configmap.yaml

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

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

步骤5 启动huawei-csi服务,详细请参见4.7 启动huawei-csi服务。

步骤6 执行**vi** *StorageClass.yaml*命令,修改yaml文件。按**I**或**Insert**进入编辑状态,在yaml 文件的"parameters"配置项下增加相关参数,详细参数说明请参见表7-10。修改完成后,按**Esc**,并输入:**wq!**,保存修改。

kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
name: "***"
provisioner: "csi.huawei.com"
parameters:
...
volumeType: lun
replication: "true"
replicationSyncPeriod: "3600"
backend: "***"

表 7-10 参数说明

参数名	参数描述	备注
replication	指定是否创建远程 复制卷	配置为true,表示需要创建远程复制卷; 不配置或配置为false,表示不创建远程复 制卷。
replicationSync Period	指定远程复制同步 时间间隔	可选,单位为秒,默认为3600s。
backend	指定创建远程复制 卷的主端名	必须与huawei-csi-configmap.yaml中配置 的某个远程复制后端名相同。

步骤7 执行以下命令,基于该yaml文件创建StorageClass。

kubectl create -f StorgeClass.yaml

步骤8 使用该StorageClass创建具有SAN远程复制能力的PVC,详细操作请参考6.2.1 创建PVC。

----结束

7.6.4 配置 SAN 双活

当您需要配置SAN双活时,执行此操作。

注意事项

使用SAN双活,需要事先在两台存储之间完成双活关系配置。具体请参考华为存储相 关配置指导。

操作步骤

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

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

在huawei-csi-configmap.yaml文件的backends配置段中,添加构成双活关系的两个后端,并且为每个后端增加hyperMetroDomain配置项。

```
{
  "backends":[
  {
    ...
    "name": "hyperMetro1",
    "hyperMetroDomain": "***",
    "metroBackend": "hyperMetro2"
  },
  {
    ...
    "name": "hyperMetro2",
    "hyperMetroDomain": "***",
    "metroBackend": "hyperMetro1"
  }
}
```

□ 说明

- hyperMetroDomain是华为存储之间配置的双活域名。
- metroBackend是双活对端的后端名,两个双活后端相互构成双活关系。如上所示,后端 hyperMetro1的双活对端是hyperMetro2,相应的,hyperMetro2的双活对端是 hyperMetro1。

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

kubectl create -f huawei-csi-configmap.yaml

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

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

步骤5 启动huawei-csi服务,详细请参见4.7 启动huawei-csi服务。

步骤6 执行**vi** *StorageClass.yaml*命令,修改yaml文件。按**l**或**Insert**进入编辑状态,在yaml 文件的"parameters"配置项下增加相关参数,详细参数说明请参见表7-11。修改完成后,按**Esc**,并输入:**wq!**,保存修改。

在StorageClass.yaml文件的parameters配置项下,添加以下配置项。

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

表 7-11 参数说明

参数名	参数描述	备注
hyperMetro	指定是否创建双活 卷	配置为true,表示需要创建双活卷;不配置 或配置为false,表示不创建双活卷。

步骤7 执行以下命令,基于该yaml文件创建StorageClass。

kubectl create -f StorgeClass.yaml

步骤8 使用该StorageClass创建具有SAN双活能力的PVC,详细操作请参考6.2.1 创建PVC。

----结束

7.6.5 配置 NAS 远程复制

当您需要配置NAS远程复制时,执行此操作。

注意事项

使用NAS远程复制,需要事先在两台存储之间完成远程复制关系配置。具体请参考华 为存储相关配置指导。

操作步骤

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

在huawei-csi-configmap.yaml文件的backends配置段中,添加构成复制关系的两个后端,并且为每个后端增加replicaBackend配置项。

```
{
  "backends":[
  {
    ...
    "name": "replica1",
    "replicaBackend": "replica2"
  },
  {
    ...
    "name": "replica2",
    "replicaBackend": "replica1"
  }
  ]
}
```

□ 说明

replicaBackend是远程复制对端的后端名,两个复制后端相互构成复制关系。如上所示,后端replica1的远程复制对端是replica2,相应的,replica2的远程复制对端是replica1。

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

kubectl create -f huawei-csi-configmap.yaml

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

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

步骤5 启动huawei-csi服务,详细请参见4.7 启动huawei-csi服务。

步骤6 执行**vi** *StorageClass.yaml*命令,修改yaml文件。按**l**或**Insert**进入编辑状态,在yaml 文件的"parameters"配置项下增加相关参数,详细参数说明请参见表7-12。修改完成后,按**Esc**,并输入:**wq!**,保存修改。

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

```
name: "***"
provisioner: "csi.huawei.com"
parameters:
...
volumeType: fs
replication: "true"
replicationSyncPeriod: "3600"
backend: "***"
```

表 7-12 参数说明

参数名	参数描述	备注
replication	指定是否创建远程 复制卷	配置为true,表示需要创建远程复制卷; 不配置或配置为false,表示不创建远程复 制卷。
replicationSync Period	指定远程复制同步 时间间隔	可选,单位为秒,默认为3600s。
backend	指定创建远程复制 卷的主端名	条件必选。在使用租户的场景下,如果租户属于远程复制租户pair,远程复制卷只能在租户pair主端创建,所以需要通过backend参数指定创建复制卷的主端名,且必须与huawei-csi-configmap.yaml中配置的某个远程复制后端名相同。

步骤7 执行以下命令,基于该yaml文件创建StorageClass。

kubectl create -f StorgeClass.yaml

步骤8 使用该StorageClass创建具有NAS远程复制能力的PVC,详细操作请参考6.2.1 创建PVC。

----结束

7.6.6 配置 NAS 双活

当您需要配置NAS双活时,执行此操作。

注意事项

使用NAS双活,需要事先在两台存储之间完成双活关系配置,并创建双活租户pair。具体请参考华为存储相关配置指导。

操作步骤

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

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

在huawei-csi-configmap.yaml文件的backends配置段中,添加构成双活关系的两个后端,并且为每个后端增加metrovStorePairID配置项。

```
{
  "backends":[
```

```
"name": "hyperMetro1",
  "vstoreName": "***",
  "metrovStorePairID": "***",
  "metroBackend": "hyperMetro2"
},
{
    ...
    "name": "hyperMetro2",
    "vstoreName": "***",
    "metrovStorePairID": "***",
    "metrovStorePairID": "***",
    "metroBackend": "hyperMetro1"
}
```

□ 说明

- 只有配置了租户的条件下才支持NAS双活。配置租户可参考7.6.2 配置租户。
- metrovStorePairID是租户归属的双活租户pair ID。
- metroBackend是双活对端的后端名,两个双活后端相互构成双活关系。如上所示,后端 hyperMetro1的双活对端是hyperMetro2,相应的,hyperMetro2的双活对端是 hyperMetro1。
- **步骤3** 执行**kubectl create -f** *huawei-csi-configmap.yaml*命令,创建huawei-csi-configmap。

kubectl create -f huawei-csi-configmap.yaml

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

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

步骤5 启动huawei-csi服务,详细请参见4.7 启动huawei-csi服务。

步骤6 执行**vi** *StorageClass.yaml*命令,修改yaml文件。按**J**或**Insert**进入编辑状态,在yaml文件的"parameters"配置项下增加"hyperMetro"参数,详细参数说明请参见表7-13。修改完成后,按**Esc**,并输入:**wq!**,保存修改。

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

表 7-13 参数说明

参数名	参数描述	备注
hyperMetro	指定是否创建双活 卷	配置为true,表示需要创建双活卷;不配 置或配置为false,表示不创建双活卷。

步骤7 执行以下命令,基于该yaml文件创建StorageClass。

kubectl create -f StorgeClass.yaml

步骤8 使用该StorageClass创建具有NAS双活能力的PVC,详细操作请参考6.2.1 创建PVC。

----结束

7.6.7 配置应用类型

本章节介绍如何创建支持不同应用类型的LUN/文件系统卷。

注意事项

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

操作步骤

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

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

"parameters"配置项下添加applicationType配置项,applicationType配置项值是字符串,详细参数说明请参见表7-14。

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

表 7-14 applicationType 参数说明

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

□ 说明

该特性仅支持OceanStor Dorado V6系列存储。

步骤3 执行以下命令,基于该yaml文件创建StorageClass。

kubectl create -f StorgeClass.yaml

步骤4 使用该StorageClass创建具有应用类型能力的PVC,详细操作请参考6.2.1 创建PVC。

----结束

7.7 分布式存储高级特性

7.7.1 配置 QoS

本章节介绍如何创建支持QoS的LUN卷。

注意事项

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

操作步骤

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

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

- "parameters"配置项下添加qos配置项。
- qos配置项值是字典格式的json字符串(字符串两边由单引号修饰,字典key由双引号修饰),详细参数说明请参见表7-15。

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

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

表 7-15 qos 支持的参数说明

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

步骤3 执行以下命令,基于该yaml文件创建新的PVC。

kubectl create -f StorageClass.yaml

----结束

8 常用操作

- 8.1 卸载CSI
- 8.2 更新CSI上配置的存储用户名或密码
- 8.3 更新huawei-csi的configmap对象
- 8.4 为huawei-csi新增后端
- 8.5 更新huawei-csi-controller服务
- 8.6 更新huawei-csi-node服务
- 8.7 修改日志输出模式

8.1 卸载 CSI

当您需要卸载CSI时执行此操作。

操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。
- **步骤2** 执行以下命令,删除huawei-csi-node服务(huawei-csi-node.yaml为**4.7 启动** huawei-csi服务中步骤8的配置信息)。

kubectl delete -f huawei-csi-node.yaml

- **步骤3** 执行以下命令,查看huawei-csi-node服务,如果没有回显,表示删除完成。 # kubectl get pod -A | grep huawei-csi-node
- **步骤4** 执行以下命令,删除huawei-csi-controller服务(huawei-csi-controller.yaml为**4.7 启** 动huawei-csi服务中步骤**6**的配置信息)。

kubectl delete -f huawei-csi-controller.yaml

- **步骤5** 执行以下命令,查看huawei-csi-controller服务,如果没有回显,表示删除完成。 # kubectl get pod -A | grep huawei-csi-controller
- 步骤6 执行以下命令,删除RBAC权限(huawei-csi-rbac.yaml为4.7 启动huawei-csi服务中步骤4的配置信息)。

kubectl delete -f huawei-csi-rbac.yaml

- **步骤7** 执行以下命令,删除huawei-csi-configmap对象。当前huawei-csi-configmap对象默认"<namespace>"参数为"kube-system",请根据实际名称进行替换。
 - # kubectl delete configmap huawei-csi-configmap -n <namespace>
- **步骤8** 执行以下命令,查看huawei-csi的configmap信息,如果没有回显,表示删除完成。 # kubectl get configmap -A | grep huawei-csi-configmap
- **步骤9** 执行以下命令,删除huawei-csi-secret对象。当前huawei-csi-secret对象默认 "<namespace>"参数为"kube-system",请根据实际名称进行替换。

kubectl delete secret huawei-csi-secret -n <namespace>

步骤10 执行以下命令,查看huawei-csi的secret信息,如果没有回显,表示删除完成。

kubectl get secret -A | grep huawei-csi-secret

----结束

8.2 更新 CSI 上配置的存储用户名或密码

当存储设备的用户名或密码发生变更时,需要更新CSI上的配置信息,否则huawei-csi服务无法正常工作。

操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意master节点。
- 步骤2 执行chmod +x secretUpdate命令,给secretUpdate添加可执行权限。
- 步骤3 执行./secretUpdate命令,运行secretUpdate工具,按照提示信息输入相关参数。

./secretUpdate **********All Secret Info***************** <current secret info> ************************ The 1 backend name is: <backend-1-name> backend url is: [<backend-1-url>] Do you want to update it? Y/N #请填写是否更新 Enter backend <backend-1-name>'s user: #请填写存储1的用户名 Enter backend <backend-1-name>'s password: #请填写存储1的密码 #再次填写存储1的密码 Please Enter the password again: The 2 backend name is: <backend-2-name> backend url is: [<backend-2-url>] Do you want to update it? Y/N #请填写是否更新 Enter backend <backend-2-name>'s user: #请填写存储2的用户名 Enter backend <backend-2-name>'s password: #请填写存储2的密码 #再次填写存储2的密码 Please Enter the password again:

步骤4 执行以下命令,重启huawei-csi-controller服务。

kubectl get deployment huawei-csi-controller -o yaml --namespace=kube-system | kubectl replace --force -f -

步骤5 执行以下命令,重启huawei-csi-node服务。

kubectl get daemonset huawei-csi-node -o yaml --namespace=kube-system | kubectl replace --force -f

步骤6 执行kubectl get pod -A | grep huawei命令检查服务是否重启成功。

kubectl get pod -A | grep huawei kube-system huawei-csi-controller-695b84b4d8-tg64l 3/3 **Running** 0 14s kube-system huawei-csi-node-g6f7z 2/2 **Running** 0 14s

----结束

8.3 更新 huawei-csi 的 configmap 对象

当您需要在已有的后端上增加存储池,或者更改已有的业务IP时,执行此操作。

操作步骤

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

表 8-1 配置项描述

配置项	参数格式	描述	备注
name	字符串	对存储设备自定义 的名称。	自定义字符串,支持大小写字母、数字、中划线组合。该参数不支持修改。
storage	字符串	待接入存储设备的 类型。	对接企业存储SAN场景,固定填写 "oceanstor-san"。该参数不支持修改。
pools	列表	被使用的待接入存储设备上的存储池名称。	 支持同一存储设备上的一个或多个存储池,用逗号分割。 当前只支持新增存储池。 可通过登录DeviceManager获取支持块存储服务的存储池。

配置项	参数格 式	描述	备注
urls	列表	待接入的存储设备 的管理URL。	支持同一存储设备的一个或多个管理 URL,用逗号分割,当前仅支持IPv4。 例如: https://192.168.125.20:8088 说明 1个存储设备有多个控制器,每个控制器有 一个管理URL,所以1个存储设备存在多个 管理URL。
paramete	字典	iSCSI场景的可变参数。	iSCSI场景protocol参数固定填写 "iscsi"。 portals参数填写待接入存储设备的 iSCSI业务IP地址,多个业务IP地址请用 逗号分割。 iSCSI业务IP地址可通过登录 DeviceManager获取。 OceanStor V3/V5系列:指定存储设备上的以太网端口IP地址,在 DeviceManager管理界面,选择 "资源分配 > 端口 > 以太网端口"。 OceanStor Dorado V3系列:指定存储设备上的以太网端口IP地址,在DeviceManager管理界面,选择 "资源分配 > 端口 > 以太网端口"。 OceanStor Dorado V6系列:指定存储设备上的逻辑端口IP地址,在DeviceManager管理界面,选择 "资源分配 > 端口 > 以太网端口"。 OceanStor Dorado V6系列:指定存储设备上的逻辑端口IP地址,在DeviceManager管理界面,选择 "服务 > 网络 > 逻辑端口",获取数据协议类型为iSCSI的IP。

步骤3 执行以下命令,重启huawei-csi-controller服务。

kubectl get deployment huawei-csi-controller -o yaml --namespace=kube-system | kubectl replace --force -f -

步骤4 执行以下命令, 重启huawei-csi-node服务。

kubectl get daemonset huawei-csi-node -o yaml --namespace=kube-system | kubectl replace --force -f -

步骤5 执行kubectl get pod -A | grep huawei命令检查服务是否重启成功。

kubectl get pod -A | grep huawei kube-system huawei-csi-controller-695b84b4d8-tg64l 3/3 **Running** 0 14s kube-system huawei-csi-node-g6f7z 2/2 **Running** 0 14s

----结束

8.4 为 huawei-csi 新增后端

当您需要新增存储设备或存储池当做独立的后端时,执行此操作。

操作步骤

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

步骤2 配置多个后端,详细操作请参见7.1 配置多个后端。

步骤3 执行chmod +x secretUpdate命令,给secretUpdate添加可执行权限。

步骤4 执行./secretUpdate命令,运行secretUpdate工具,按照提示信息输入相关参数。

```
# ./secretUpdate
  <current secret info>
The 1 backend name is: <backend-1-name> backend url is: [<backend-1-url>]
Do you want to update it? Y/N
                         #请填写是否更新
Enter backend <backend-1-name>'s user: #请填写存储1的用户名
Enter backend <backend-1-name>'s password: #请填写存储1的密码
                         #再次填写存储1的密码
Please Enter the password again:
The 2 backend name is: <backend-2-name> backend url is: [<backend-2-url>]
                       #请填写是否更新
Do you want to update it? Y/N
Enter backend <backend-2-name>'s user:
                             #请填写存储2的用户名
Enter backend <backend-2-name>'s password: #请填写存储2的密码
Please Enter the password again:
                          #再次填写存储2的密码
```

步骤5 执行以下命令,重启huawei-csi-controller服务。

kubectl get deployment huawei-csi-controller -o yaml --namespace=kube-system | kubectl replace --force -f -

步骤6 执行以下命令,重启huawei-csi-node服务。

kubectl get daemonset huawei-csi-node -o yaml --namespace=kube-system | kubectl replace --force -f -

步骤7 执行kubectl get pod -A | grep huawei命令检查服务是否重启成功。

```
# kubectl get pod -A | grep huawei
kube-system huawei-csi-controller-695b84b4d8-tg64l 3/3 Running 0 14s
kube-system huawei-csi-node-g6f7z 2/2 Running 0 14s
```

----结束

8.5 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** 执行以下命令,删除huawei-csi-controller服务(huawei-csi-controller.yaml为**4.7 启 动huawei-csi服务中步骤6**的配置信息)。

kubectl delete -f huawei-csi-controller.yaml

步骤3 执行以下命令,查看huawei-csi-controller服务,如果没有回显,表示删除完成。 # kubectl get pod -A | grep huawei-csi-controller 步骤4 执行以下命令,删除RBAC权限(huawei-csi-rbac.yaml为4.7 启动huawei-csi服务中步骤4的配置信息)。

kubectl delete -f huawei-csi-rbac.yaml

步骤5 执行vi huawei-csi-rbac.yaml命令,修改yaml文件。按l或lnsert进入编辑状态,修改相关参数。修改完成后,按Esc,并输入:wq!,保存修改。(可参考软件包中yamls/huawei-csi-rbac.yaml示例文件)。

□ 说明

- csi-resizer服务从Kubernetes v1.16开始支持
 huawei-csi-rbac.yaml文件请参考软件包中yamls/huawei-csi-resize-rbac.yaml示例文件
- csi-snapshotter服务从Kubernetes v1.17开始支持
 huawei-csi-rbac.yaml文件请参考软件包中yamls/huawei-csi-resize-snapshot-rbac.yaml示例文件
- 步骤6 执行以下命令,创建RBAC权限。

kubectl create -f huawei-csi-rbac.yaml

步骤7 执行vi huawei-csi-controller.yaml命令,修改yaml文件。按**I**或**Insert**进入编辑状态, 修改相关参数。修改完成后,按Esc,并输入:wq!,保存修改。(可参考软件包中 yamls/huawei-csi-controller.yaml示例文件)。

□ 说明

- 示例yaml文件中huawei-csi:*.**配置项,*.*.*应替换为前面制作的华为CSI镜像版本号。
- csi-resizer服务从Kubernetes v1.16开始支持
 huawei-csi-controller.yaml文件请参考软件包中yamls/huawei-csi-resize-controller.yaml示例文件
- csi-snapshotter服务从Kubernetes v1.17开始支持
 huawei-csi-controller.yaml文件请参考软件包中yamls/huawei-csi-resize-snapshot-controller.yaml示例文件
- 步骤8 执行以下命令,启动controller服务。

kubectl create -f huawei-csi-controller.yaml

步骤9 完成huawei-csi服务部署后,可执行kubectl get pod -A | grep huawei-csi-controller 命令检查服务是否启动。

kubectl get pod -A | grep huawei-csi-controller kube-system huawei-csi-controller-695b84b4d8-tg64l 3/3 Running 0 14s

----结束

8.6 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 执行以下命令,删除huawei-csi-node服务(huawei-csi-node.yaml为**4.7 启动** huawei-csi服务中步骤8的配置信息)。

kubectl delete -f huawei-csi-node.yaml

步骤3 执行以下命令,查看huawei-csi-node服务,如果没有回显,表示删除完成。 # kubectl get pod -A | grep huawei-csi-node

步骤4 执行vi huawei-csi-node.yaml命令,修改yaml文件。按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。编写huawei-csi-node.yaml文件(可参考软件包中yamls/huawei-csi-node.yaml示例文件)。

- 示例yaml文件中huawei-csi:*.*.*配置项,*.*.*应替换为前面制作的华为CSI镜像版本号。
- 示例yaml文件中huawei-csi-driver配置参数args中: --volume-use-multipath 表示默认开启多路径,如需修改,请参考如下。

args:

- "--endpoint=/csi/csi.sock"
- "--containerized"
- "--driver-name=csi.huawei.com"
- "--volume-use-multipath=false"

步骤5 执行以下命令,启动node服务。

kubectl create -f huawei-csi-node.yaml

步骤6 完成huawei-csi服务部署后,可执行kubectl get pod -A | grep huawei-csi-node命令检查服务是否启动。

kubectl get pod -A | grep huawei-csi-node kube-system huawei-csi-node-g6f7z 2/2 Running 0 14s

----结束

8.7 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.

8.7.1 修改 huawei-csi-controller 服务的日志输出模式

当您需要为huawei-csi-controller服务设置日志输出模式,执行此操作。

操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意master节点。
- **步骤2** 执行以下命令,删除huawei-csi-controller服务(huawei-csi-controller.yaml为**4.7 启动huawei-csi服务中步骤6**的配置信息)。

kubectl delete -f huawei-csi-controller.yaml

步骤3 执行以下命令,查看huawei-csi-controller服务,如果没有回显,表示删除完成。 # kubectl get pod -A | grep huawei-csi-controller

步骤4 执行vi huawei-csi-controller.yaml命令,修改yaml文件。按l或**Insert**进入编辑状态,修改相关参数。修改完成后,按Esc,并输入:wq!,保存修改。(可参考软件包中yamls/huawei-csi-controller.yaml示例文件)。参数详情可参考<mark>表8-2</mark>。

args

- "--endpoint=\$(CSI_ENDPOINT)"
- "--controller"
- "--containerized"
- "--driver-name=csi.huawei.com"

- "--loggingModule=file"
- "--logLevel=info"
- "--logFileDir=/var/log/huawei"
- "--logFileSize=20M"
- "--maxBackups=9"

表 8-2 日志输出参数说明

配置项	描述	备注
logging Module	huawei-csi日志输出 模式。	当前可以支持"file"和"console",默认为 "file"。
logLevel	huawei-csi日志输出 级别。	支持的级别有:debug,info,warning, error,fatal。默认的级别为info。
logFileD ir	huawei-csi日志在file 输出模式下的日志目 录。	仅当"loggingModule"为"file"时生效,默 认的日志目录为/var/log/huawei。
logFileSi ze	huawei-csi日志在file 输出模式下单个日志 文件大小。	仅当"loggingModule"为"file"时生效,默 认的日志文件大小为20MiB。
maxBac kups	huawei-csi日志在file 输出模式下日志文件 备份上限。	仅当"loggingModule"为"file"时生效,默 认的日志文件备份数量为9个。

步骤5 执行以下命令,启动controller服务。

kubectl create -f huawei-csi-controller.yaml

步骤6 完成huawei-csi服务部署后,可执行kubectl get pod -A | grep huawei-csi-controller 命令检查服务是否启动。

kubectl get pod -A -o wide | grep huawei kube-system huawei-csi-controller-b59577886-qqzm8 3/3 Running 0 18h 10.244.1.67 node <none> <none>

步骤7 查看huawei-csi-controller服务的日志。

- 如果设置"loggingModule"为"file",请登录到具体的node节点,然后进入到logFileDir设置的日志目录,执行以下命令,查看huawei-csi-controller日志。 # tail -f huawei-csi-controller
- 如果设置"loggingModule"为"console",请执行以下命令,查看huawei-csi-controller日志

kubectl logs huawei-csi-controller -c huawei-csi-driver -n kube-system

----结束

8.7.2 修改 huawei-csi-node 服务的日志输出模式

当您需要为huawei-csi-node服务设置日志输出模式,执行此操作。

操作步骤

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

步骤2 执行以下命令,删除huawei-csi-node服务(huawei-csi-node.yaml为**4.7 启动** huawei-csi服务中步骤8的配置信息)。

kubectl delete -f huawei-csi-node.yaml

步骤3 执行以下命令,查看huawei-csi-node服务,如果没有回显,表示删除完成。 # kubectl get pod -A | grep huawei-csi-node

步骤4 执行vi huawei-csi-node.yaml命令,修改yaml文件。按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。编写huawei-csi-node.yaml文件(可参考软件包中yamls/huawei-csi-node.yaml示例文件)。参数详情可参考表8-3。

args:

- "--endpoint=/csi/csi.sock"
- "--containerized"
- "--driver-name=csi.huawei.com"
- "--volume-use-multipath=false"
- "--loggingModule=file"
- "--logLevel=info"
- "--logFileDir=/var/log/huawei"
- "--logFileSize=20M"
- "--maxBackups=9"

表 8-3 日志输出参数说明

配置项	描述	备注
logging Module	huawei-csi日志输出 模式	当前可以支持"file"和"console",默认为 "file"。
logLevel	huawei-csi日志输出 级别	支持的级别有:debug,info,warning, error,fatal。默认的级别为info。
logFileD ir	huawei-csi日志在file 输出模式下的日志目 录	仅当"loggingModule"为"file"时生效,默 认的日志目录为/var/log/huawei。
logFileSi ze	huawei-csi日志在file 输出模式下单个日志 文件大小	仅当"loggingModule"为"file"时生效,默 认的日志文件大小为20MiB。
maxBac kups	huawei-csi日志在file 输出模式下日志文件 备份上限	仅当"loggingModule"为"file"时生效,默 认的日志文件备份数量为9个。

步骤5 执行以下命令,启动node服务。

kubectl create -f huawei-csi-node.yaml

步骤6 完成huawei-csi服务部署后,可执行kubectl get pod -A | grep huawei-csi-node命令检查服务是否启动。

步骤7 查看huawei-csi-node服务的日志。

如果设置"loggingModule"为"file",请登录到具体的node节点,然后进入到logFileDir设置的日志目录,执行以下命令,查看huawei-csi-node日志。
 # tail -f huawei-csi-node

● 如果设置"loggingModule"为"console",请执行以下命令,查看huawei-csi-node日志

kubectl logs *huawei-csi-node* -c huawei-csi-driver -n kube-system

----结束

9 FAQ

- 9.1 查看日志信息
- 9.2 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
- 9.3 启动huawei-csi-node失败,提示错误为: "/var/lib/iscsi is not a directory"
- 9.4 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
- 9.5 启动huawei-csi服务时,服务启动异常, 状态显示InvalidImageName
- 9.6 创建PVC时, PVC的状态为Pending
- 9.7 删除PVC前,PVC的状态为Pending
- 9.8 创建Pod时,Pod的状态为ContainerCreating

9.1 查看日志信息

查看配置 secret 时的日志

步骤1 执行cd /var/log/huawei命令,进入日志目录

cd /var/log/huawei

步骤2 执行以下命令,查看huawei-csi-install日志

vi huawei-csi-install

----结束

查看 huawei-csi-controller 服务的日志

步骤1 执行以下命令,获取huawei-csi-controller所在的节点

kubectl get pod -A -o wide | grep huawei kube-system huawei-csi-controller-695b84b4d8-tg64l 3/3 **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>_kube-system_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 kube-system huawei-csi-node-g6f7z name> <none> <none>

2/2 **Running** 0

0 14s <host2-ip> <host2-

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>_kube-system_huawei-csi-driver-<contrainer-id>.log

----结束

9.2 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'。

环境配置

Kubernetes版本: 1.13及以上

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。

解决措施或规避方法

执行以下命令"modprobe iscsi_tcp",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

9.3 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 kube-system命令查看,提示错误为: "/var/lib/iscsi is not a directory"

环境配置

Kubernetes版本: 1.13及以上

根因分析

huawei-csi-node中容器内部无/var/lib/iscsi目录

解决措施或规避方法

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

步骤2 执行以下命令,删除huawei-csi-node服务(huawei-csi-node.yaml为**4.7 启动** huawei-csi服务中步骤8的配置信息)。

kubectl delete -f huawei-csi-node.yaml

步骤3 执行以下命令,查看huawei-csi-node服务,如果没有回显,表示删除完成。 # kubectl get pod -A | grep huawei-csi-node

步骤4 执行vi huawei-csi-node.yaml命令,修改yaml文件。按l或Insert进入编辑状态,将huawei-csi-node.yaml -> volumes -> iscsi-dir -> hostPath中"path"设置为"/var/lib/iscsi",删除"type"行。修改完成后,按Esc,并输入:wq!,保存修改。编写huawei-csi-node.yaml文件(可参考软件包中yamls/huawei-csi-node.yaml示例文件)。

步骤5 执行以下命令,启动node服务。

kubectl create -f huawei-csi-node.yaml

步骤6 完成huawei-csi服务部署后,可执行kubectl get pod -A | grep huawei-csi-node命令检查服务是否启动。

kubectl get pod -A | grep huawei-csi-node kube-system huawei-csi-node-g6f7z 2/2 Running 0 14s

----结束

9.4 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: 1.13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:

排查方法:

步骤1 执行命令: multipath -ll, 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。

- 是 ⇒ 步骤3。
- 否 => 不涉及。

步骤2 判断残留的DM多路径设备是否可读。

执行命令: dd if=/dev/dm-xx of=/dev/null count=1 bs=1M iflag=direct

(dm-xx为步骤1查到的设备号)

如果返回结果为: Input/output error,且读取数据为"0 bytes (0 B) copied",表示该设备不可读。

#dd if=/dev/dm-12 of=/dev/null count=1 bs=1M iflag=direct dd: error reading '/dev/dm-12': Input/output error 0+0 records in 0+0 records out 0 bytes (0 B) copied, 0.0236862 s, 0.0 kB/s

- 是 => 记录残留的dm-xx设备以及关联磁盘号(见步骤1),执行清理步骤。
- 命令卡死 => 3
- **其他** => 联系技术支持。

步骤3 在另一窗口再次登录该节点。

1. 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

- 2. 将该pid杀死。
 - # kill -9 *pid*
- 3. 记录残留的dm-xx设备以及关联磁盘号(见<mark>步骤1</mark>),执行清理步骤 。

----结束

清理步骤:

步骤1 根据"排查方法"获取的DM多路径设备,执行命令:multipath -f /dev/*dm-xx*,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。

步骤2 清理残留的SCSI磁盘,根据"排查方法"获取的残留磁盘的盘符,依次执行命令:

echo 1 > /sys/block/xxxx/device/delete

配置多条多路径时,依次根据盘符清除,本次残留路径为sdi/sdj:

echo 1 > /sys/block/sdi/device/delete
echo 1 > /sys/block/sdj/device/delete

如果执行报错,请联系技术支持。

步骤3 确认DM多路径设备和SCSI磁盘信息是否已经清理干净。

依次执行命令"multipath -ll"、"ls -l /sys/block/"、"ls -l /dev/disk/by-id/",查询的多路径和磁盘信息显示,残留的dm-12和SCSI磁盘sdi/sdj均已消失,则证明清理完成。

multipath -ll

mpathb (3618cf24100f8f457014a764c000001f6) dm-3 HUAWEI ,XSG1

mpathn (3618cf24100f8f457315a764c000001f6) dm-5 HUAWEI ,XSG1

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

```
lrwxrwxrux 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
lrwxrwxrux 1 root root 0 Aug 11 19:56 sdd -> ../devices/platform/host39/session6/target39:0:0/39:0:0:1/
lrwxrwxrwx 1 root root 0 Aug 11 19:56 sde -> ../devices/platform/host38/session5/target38:0:0/38:0:0:1/
lrwxrwxrwx 1 root root 0 Aug 11 19:56 sdh -> ../devices/platform/host39/session6/target39:0:0/39:0:0:3/
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 -> ../../dm-3
lrwxrwxrwx 1 root root 10 Aug 11 19:58 dm-uuid-mpath-3618cf24100f8f457315a764c000001f6 -> ../../dm-5
lrwxrwxrwx 1 root root 9 Aug 11 19:57 scsi-3618cf24100f8f457014a764c000001f6 -> ../../sdd
lrwxrwxrwx 1 root root 9 Aug 11 19:57 scsi-3618cf24100f8f45712345678000103e8 -> ../../sdi
lrwxrwxrwx 1 root root 9 Aug 3 15:17 scsi-3648435a10058805278654321fffffffff -> ../../sdb
lrwxrwxrwx 1 root root 9 Aug 2 14:49 scsi-368886030000020aff44cc0d060c987f1 -> ../../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

9.5 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 kube-system huawei-csi-controller-fd5f97768-qlldc 5/6 InvalidImageName 0 16s kube-system huawei-csi-node-25txd 1/2 InvalidImageName 0 15s
```

环境配置

Kubernetes版本: 1.13及以上

根因分析

huawei-csi-controller.yaml和huawei-csi-node.yaml中,配置Huawei CSI镜像版本号错误。例如:

```
...
- name: huawei-csi-driver
image: huawei-csi:*.*.*
...
```

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l**nsert**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 --namespace=kube-system

□ 说明

示例yaml文件中huawei-csi:*.*.*配置项,*.*.*应替换为前面制作的华为CSI镜像版本号。

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 --namespace=kube-system

□ 说明

示例yaml文件中huawei-csi:*:**配置项, *:*:*应替换为前面制作的华为CSI镜像版本号。

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
kube-system huawei-csi-controller-58799449cf-zvhmv 6/6 Running 0 2m29s
kube-system huawei-csi-node-7fxh6 2/2 Running 0 12m
```

----结束

9.6 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。

环境配置

Kubernetes版本: 1.13及以上

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。

			J				
Events:							
Type	Reason	Age	From	Message			
Warnii	Warning ProvisioningFailed 0s (x15 over 3m24s) persistent/volume-controller						
storage	torageclass.storage.k8s.io " <i>mysc</i> " not found						

- a. 删除PVC,详细请参考**6.2.5 删除PVC**。
- b. 创建StorageClass, 详细请参考6.1.1 创建StorageClass。
- c. 创建新的PVC,详情请参考**6.2.1 创建PVC**。
- 如果由原因2导致PVC处于Pending状态,执行以下步骤。

7-1-1-1	/// · · · · · · · · · · · · · · · · · ·	,		
Events:				
Type	Reason	Age		
From			Message	
Normal	l Provisioning	63s (x	over 64s) csi.huawei.com_huawei-csi-	-controller-b59577886-
qqzm8_5	8533e4a-884c-	4c7f-92c3-6e	a7b327515 External provisioner is pro	ovisioning volume for
claim "de	efault/mypvc"			
Warnin	a Provisionina	Failed 63s	3 over 64s) csi huawei.com huawei-c	si-controller-b59577886-

Warning ProvisioningFailed 63s (x3 over 64s) csi.huawei.com_huawei-csi-controller-b59577886-qqzm8_58533e4a-884c-4c7f-92c3-6e8a7b327515 failed to provision volume with StorageClass "mysc": rpc error: code = Internal desc = **failed to select pool**, the capability filter failed, error: failed to select pool, the final filter field: **replication**, parameters map[allocType:thin replication:True size: 1099511627776 volumeType:lun]. please check your storage class

- a. 删除PVC, 详细请参考**6.2.5 删除PVC**。
- b. 删除StorageClass, 详细请参考6.1.2 删除StorageClass。
- c. 根据Events信息修改StorageClass.yaml文件。
- d. 创建StorageClass, 详细请参考6.1.1 创建StorageClass。
- e. 创建新的PVC, 详情请参考**6.2.1 创建PVC**。
- 如果由原因3导致PVC处于Pending状态、请联系华为工程师处理。

ントノト		TV: 4 CVT 1 :	Channey John	,内状心一刀工作为处理。
Events:				
Type	Reason	Age		
From				Message
Norm	al Provisio	ning 63s	(x4 over 68s) cs	si.huawei.com_huawei-csi-controller-b59577886-
qqzm8	_58533e4a-	884c-4c7f-92c3	-6e8a7b327515	External provisioner is provisioning volume for
claim "	default/myr	ovc"		
Warn	ing Provisio	ningFailed 62	2s (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: o	ode = Internal	desc = Create vo	olume map[ALLOCTYPE:1 CAPACITY:20
DESCR	IPTION:Crea	ted from Kuber	netes CSI NAME	E:pvc-63ebfda5-4cf0-458e-83bd-ecc PARENTID:0]
error: 5	0331651			

● 如果由原因4导致PVC处于Pending状态,执行以下步骤。

如未田	原囚4号致₽V	/CXLT Pending	仏心,がけ以下少塚。
Events:			
Type	Reason	Age	
From			Message
Normal	l Provisioning	63s (x3 over	52s) csi.huawei.com_huawei-csi-controller-b59577886-
qqzm8_5	8533e4a-884c-4	1c7f-92c3-6e8a7b3	27515 External provisioner is provisioning volume for
claim "de	efault/mypvc"		
Warnin	g ProvisioningF	ailed 63s (x3 ove	er 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

9.7 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。

环境配置

Kubernetes版本: 1.13及以上

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。

Age	From	Message				
Warning ProvisioningFailed 0s (x15 over 3m24s) persistentvolume-controller						
storageclass.storage.k8s.io " <i>mysc</i> " not found						
	ingFailed Os (x	ingFailed Os (x15 over 3m24s)	ingFailed 0s (x15 over 3m24s) persistentvolume-controller			

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

9.8 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**9.1 查看日志信息**),报错"Fibre Channel volume device not found"。

环境配置

Kubernetes版本: 1.13及以上

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, 详情请参考6.3.2 删除Pod。

步骤4 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的*node1* 节点。*node1*节点为**步骤**2中查询的节点。

步骤5 移除盘符残留,详情请参考解决措施或规避方法。

----结束

10 附录

- 10.1 OceanStor V3/V5系列和OceanStor Dorado V3系列ALUA特性配置策略样例
- 10.2 OceanStor Dorado V6 ALUA特性配置策略样例
- 10.3 分布式存储ALUA特性配置策略样例

10.1 OceanStor V3/V5 系列和 OceanStor Dorado V3 系列 ALUA 特性配置策略样例

例1.配置文件如下:

对于主机名为"node1",上述ALUA配置段都能用于配置启动器。根据**7.4.1 配置OceanStor V3/V5系列和OceanStor Dorado V3系列ALUA**中的配置策略规则,优先级顺序为第2条配置段(HostName为"node1")高于第1条配置段(HostName为"*")。

例2.配置文件如下:

对于主机名为"node6"的主机,上述ALUA配置段都能用于配置启动器。根据**7.4.1** 配置OceanStor V3/V5系列和OceanStor Dorado V3系列ALUA中的配置策略规则,选择第一条ALUA配置段来配置启动器。

例3.配置文件如下:

```
...
"parameters": {..., "ALUA": {
```

```
"^node1$": {"MULTIPATHTYPE": "1", "FAILOVERMODE": "3", "SPECIALMODETYPE": "0", "PATHTYPE": "0"}, "^node10$": {"MULTIPATHTYPE": "1", "FAILOVERMODE": "3", "SPECIALMODETYPE": "0", "PATHTYPE": "1"}}} ...
```

根据**7.4.1 配置OceanStor V3/V5系列和OceanStor Dorado V3系列ALUA**中的配置 策略规则,对于主机名为"node1"的主机,选择第一条ALUA配置段来配置启动器; 对于主机名为"node10"的主机,选择第二条ALUA配置段来配置启动器。^表示匹配字符串的开头,\$表示匹配字符串的结尾。

10.2 OceanStor Dorado V6 ALUA 特性配置策略样例

例1.配置文件如下:

对于主机名为 "node1",上述ALUA配置段都能用于配置启动器。根据**7.4.2 配置OceanStor Dorado V6 ALUA**中的配置策略规则,优先级顺序为第2条配置段(HostName为"node1")高于第1条配置段(HostName为"*")。

例2.配置文件如下:

对于主机名为"node6"的主机,上述ALUA配置段都能用于配置启动器。根据**7.4.2** 配置OceanStor Dorado V6 ALUA中的配置策略规则,选择第一条ALUA配置段来配置启动器。

例3.配置文件如下:

根据**7.4.2 配置OceanStor Dorado V6 ALUA**中的配置策略规则,对于主机名为"node1"的主机,选择第一条ALUA配置段来配置启动器;对于主机名为"node10"的主机,选择第二条ALUA配置段来配置启动器。^表示匹配字符串的开头,\$表示匹配字符串的结尾。

10.3 分布式存储 ALUA 特性配置策略样例

例1.配置文件如下:

对于主机名为"node1",上述ALUA配置段都能用于配置启动器。根据**7.4.3 配置分布式存储ALUA**中的配置策略规则,优先级顺序为第2条配置段(HostName为"node1")高于第1条配置段(HostName为"*")。

例2.配置文件如下:

对于主机名为"node6"的主机,上述ALUA配置段都能用于配置启动器。根据7.4.3配置分布式存储ALUA中的配置策略规则,选择第一条ALUA配置段来配置启动器。

例3.配置文件如下:

根据**7.4.3 配置分布式存储ALUA**中的配置策略规则,对于主机名为"node1"的主机,选择第一条ALUA配置段来配置启动器;对于主机名为"node10"的主机,选择第二条ALUA配置段来配置启动器。^表示匹配字符串的开头,\$表示匹配字符串的结尾。