

eSDK Huawei Storage Kubernetes CSI Plugins

用户指南（ Unionpay ）

文档版本 01
发布日期 2022-11-11



版权所有 © 华为技术有限公司 2022。保留一切权利。

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

商标声明



HUAWEI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://e.huawei.com>

前言






读者对象

本文档主要适用于以下读者对象：

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

符号约定

在本文中可能出现下列标志，它们所代表的含义如下。

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

修改记录

文档版本	发布日期	修改说明
01	2022-11-11	第一次正式发布。

目录

前言.....	ii
1 概述.....	1
2 兼容性和特性.....	3
2.1 Kubernetes 及操作系统兼容性.....	3
2.2 华为企业存储兼容性.....	3
3 安装前准备.....	5
3.1 前提条件.....	5
3.2 下载华为 CSI 软件包.....	5
3.3 上传华为 CSI 镜像.....	6
3.3.1 上传镜像到镜像仓库.....	6
3.3.2 导入镜像到所有节点.....	7
3.4 检查 CSI 依赖的镜像.....	7
3.5 检查华为存储上的账号配置.....	9
3.6 检查主机依赖软件状态.....	9
4 安装华为 CSI.....	10
4.1 创建华为 CSI 运行时需要的 ConfigMap.....	10
4.1.1 通过 NFS 对接企业存储 NAS.....	10
4.1.2 通过租户/账户对接华为存储后端配置.....	13
4.1.3 通过 CSI 对接多个后端存储的配置.....	13
4.2 启动 huawei-csi 服务.....	14
5 卸载华为 CSI.....	17
5.1 卸载 huawei-csi-node 服务.....	17
5.2 卸载 huawei-csi-controller 服务.....	18
5.3 删除 huawei-csi-configmap 对象.....	18
5.4 删除 huawei-csi-secret 对象.....	18
5.5 删除 RBAC 权限.....	19
5.6 删除老版本镜像.....	19
6 使用华为 CSI.....	21
6.1 持久卷管理.....	21
6.1.1 创建 PVC.....	21
6.1.1.1 动态卷供应.....	21

6.1.1.1.1 配置存储类.....	22
6.1.1.1.2 配置 PVC.....	26
6.1.1.2 静态卷供应.....	28
6.1.1.2.1 配置 PV.....	28
6.1.1.2.2 配置 PVC.....	30
7 高级特性.....	33
7.1 配置存储拓扑感知.....	33
7.1.1 配置步骤.....	34
8 常用操作.....	37
8.1 更新 CSI 上配置的存储用户名或密码.....	37
8.2 更新 huawei-csi 的 configmap 对象.....	38
8.3 为 huawei-csi 新增后端.....	40
8.4 更新 huawei-csi-controller 服务.....	40
8.5 更新 huawei-csi-node 服务.....	41
8.6 修改日志输出模式.....	41
8.6.1 修改 huawei-csi-controller 服务的日志输出模式.....	41
8.6.2 修改 huawei-csi-node 服务的日志输出模式.....	42
8.7 配置非 root 用户访问 Kubernetes 集群.....	44
9 FAQ.....	45
9.1 如何查看华为 CSI 日志.....	45
9.2 启动 huawei-csi 服务时，服务启动异常， 状态显示 InvalidImageName.....	46
9.3 创建 PVC 时， PVC 的状态为 Pending.....	47
9.4 删除 PVC 前， PVC 的状态为 Pending.....	49
9.5 创建 Pod 时， Pod 的状态长时间处于 ContainerCreating 状态.....	50
9.6 如何下载容器镜像到本地.....	50
9.7 如何获取 CSI 版本信息.....	51
10 附录.....	52
10.1 制作 CSI 镜像.....	52

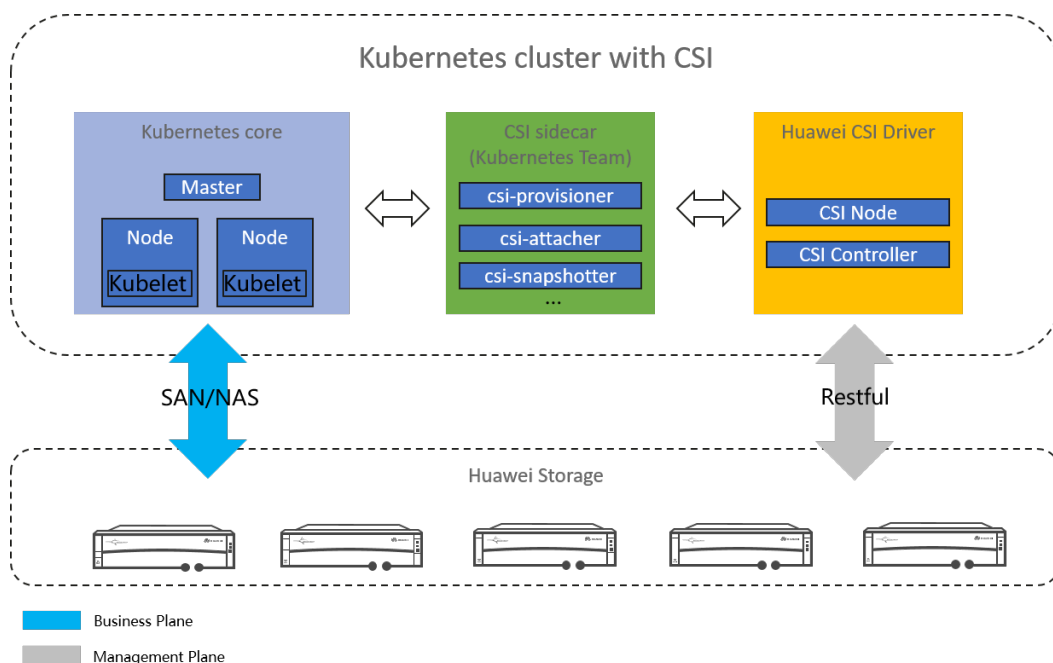
1 概述

Kubernetes，简称 k8s或者“kube”，是一个可移植、可扩展的开源平台，用于管理容器化的工作负载和服务。

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

Kubernetes与华为CSI以及华为存储的整体结构如下：

图 1-1 CSI 整体架构



- Kubernetes通过其官方维护的一系列sidecar组件负责注册监听Kubernetes对象资源，并在需要的时候发起对CSI Driver调用。
- 华为CSI Driver将sidecar发起的调用在华为存储上实施，如创建一个持久化卷的操作被实施为在华为存储上创建一个文件系统。

- CSI Controller: 以Deployment方式单独运行的一个Pod, 用于和存储交互, 在存储侧对资源进行创建、删除等操作, 如创建文件系统。
- CSI Node: 以DaemonSet方式运行在Kubernetes工作节点上的Pod, 用于在工作节点上对华为存储提供的文件系统资源进行挂载、卸载、格式化文件系统等操作。
- 华为存储为K8s的工作节点通过多种协议提供NAS存储资源服务, 华为存储和华为CSI驱动之间则使用Restful进行通信。

本文档主要介绍华为CSI插件安装部署和使用。

2 兼容性和特性

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

2.1 Kubernetes及操作系统兼容性

2.2 华为企业存储兼容性

2.1 Kubernetes 及操作系统兼容性

华为CSI插件支持如下容器管理平台

表 2-1 支持的容器管理平台

容器管理平台	版本
Kubernetes	1.12

华为CSI插件支持的操作系统信息如下：

表 2-2 支持的操作系统

操作系统名称	操作系统版本
CentOS x86_64	7.5,7.6, 7.7, 7.9

2.2 华为企业存储兼容性

华为CSI插件支持的存储版本如下表所示：

表 2-3 支持的华为企业存储

存储产品	版本
OceanStor Dorado V6	6.0.1, 6.1.0, 6.1.2, 6.1.3, 6.1.5
OceanStor V6	6.1.3, 6.1.5

华为CSI插件针对华为企业存储支持如下特性。

表 2-4 华为企业存储支持的特性及约束

特性	OceanStor V6	OceanStor Dorado V6
Static Provisioning	NAS: NFS3/4.0/4.1 ¹	NAS: NFS3/4.0/4.1 ¹
Dynamic Provisioning		
Topology	支持	支持
Access Mode	RWO/ROX/RWX	
应用类型	支持	支持
存储多租户	仅支持NAS类型的卷 ²	仅支持NAS类型的卷 ²

- 注释1 仅Dorado V6 6.1.0及以后版本支持NFS。仅Dorado V6 6.1.3及以后版本NFS4.1。
- 注释2 仅Dorado V6 6.1.3及以后版本支持多租户。

3 安装前准备

在安装华为CSI插件到您的容器管理平台前，需要CSI镜像制作、上传/导入镜像。本章节将对安装前的准备工作进行详细说明。

3.1 前提条件

3.2 下载华为CSI软件包

3.3 上传华为CSI镜像

3.4 检查CSI依赖的镜像

3.5 检查华为存储上的账号配置

3.6 检查主机依赖软件状态

3.1 前提条件

在进行本章节所说明的操作前，请确保如下条件已经具备：

- 容器管理平台已部署完成并正常运行，且满足[2.1 Kubernetes及操作系统兼容性](#)章节的要求。
- 已完成对接华为企业存储初始化配置，包括存储池划分、端口配置等。且存储产品的版本满足[2.2 华为企业存储兼容性](#)章节的要求。
- 完成华为存储和容器平台主机连通性配置，根据您的规划，请确保容器集群上工作节点已经安装对应协议所需要的软件客户端，如NFS客户端。
- 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[支持的操作系统及多路径信息](#)。
- Kubernetes的所有worker节点与待接入的存储设备业务IP通信正常（华为CSI使用ping命令判断）。
- 一台已安装Docker的Linux主机，且该主机支持访问华为Kubernetes CSI仓库。

3.2 下载华为 CSI 软件包

本章节详细说明了下载方法以及软件包组件结构。

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

步骤2 根据存储类型和CPU架构，下载对应的unionpay版本软件包。

📖 说明

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

步骤3 将下载的软件包解压。软件包组件结构如表3-1所示。

表 3-1 软件包组件描述

组件	组件描述
image/huawei-csi-unionpay- <i>arch</i> .tar	huawei-csi镜像，"arch"为x86或arm。
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。
deploy	CSI部署过程中，yaml示例文件。
examples	CSI使用过程中，yaml示例文件。
tools	在环境中无镜像仓库时，用于上传镜像的脚本。

----结束

3.3 上传华为 CSI 镜像

当前华为已经提供huawei-csi镜像供用户使用，镜像文件获取参见表3-1。如果用户需要重新制作镜像，请参见10.1 制作CSI镜像。

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

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

3.3.1 上传镜像到镜像仓库

前提条件

- 一台已安装Docker的Linux主机，且该主机支持访问镜像仓库。

操作步骤

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

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

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

```
# docker tag huawei-csi:unionpay repo.huawei.com/huawei-csi:unionpay
```

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

```
# docker push repo.huawei.com/huawei-csi:unionpay
```

----结束

3.3.2 导入镜像到所有节点

前提条件

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

操作步骤

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

```
# vi worker-list.txt
```

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

```
# ip
192.168.128.16
192.168.128.17
```

步骤3 上传并导入镜像。

- 如果使用的是`containrd`作为容器运行时，执行`./containerd-upload.sh worker-list.txt huawei-csi.tar`命令，按提示输入用户名和密码，上传并导入镜像。

```
# ./containerd-upload.sh worker-list.txt huawei-csi.tar
```
- 如果使用的是`Docker`作为容器运行时，执行`./docker-upload.sh worker-list.txt huawei-csi.tar`命令，按提示输入用户名和密码，上传并导入镜像。

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

步骤4 检查导入镜像是否成功。

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

```
All images are uploaded successfully
```
2. 如果脚本最后回显是如下的列表，则表示列表中的节点镜像导入是失败的，这时需要检查日志检查原因。

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

----结束

3.4 检查 CSI 依赖的镜像

Huawei CSI安装以及后续运行时，需要依赖下表中的组件镜像。请将这些容器镜像上传到您的镜像仓库中或者导入到所有节点中。

说明

在 **huawei-csi-controller** 服务依赖的镜像有：liveness-probe、csi-provisioner、csi-attacher和 huawei-csi-driver。

在 **huawei-csi-node** 服务依赖的镜像有：liveness-probe、driver-registrar和huawei-csi-driver。

关于每个镜像的功能和详情，请参考表3-2。

如需下载容器镜像到本地，请参考9.6 如何下载容器镜像到本地。

表 3-2 Huawei CSI 依赖的镜像

容器名称	容器镜像	功能描述	备注
liveness-probe	quay.io/k8scsi/ livenessprobe:v0.4. 1	用于监控CSI的健康状态，并上报给Kubernetes，使Kubernetes能够自动检测CSI程序的问题并重启Pod尝试修改该问题。	查看详情
csi-provisioner	quay.io/k8scsi/csi- provisioner:v0.4.2	<ul style="list-style-type: none">在创建PVC时，调用CSI Controller服务在存储上创建L文件系统作为PV，并将PV绑定至PVC。在删除PVC时，调用CSI Controller服务解除PV至PVC的绑定，然后在存储上删除该PV对应的文件系统。	查看详情
csi-attacher	quay.io/k8scsi/csi- attacher:v0.4.2	在创建/删除Pod时，调用CSI Controller服务执行Publish/Unpublish Volume操作。	查看详情
driver-registrar	quay.io/k8scsi/ driver- registrar:v0.4.2	用于获取CSI信息，并通过kubelet的插件注册机制将节点注册到kubelet中，从而Kubernetes能够感知该节点与华为存储的对接。	查看详情

3.5 检查华为存储上的账号配置

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

表 3-3 存储对接 CSI 时使用的账号要求

存储类型	用户类型	角色	级别	类型
OceanStor Dorado 6.1.3/6.1.5	系统用户	管理员	N/A	本地用户
	租户用户	租户管理员	N/A	本地用户
OceanStor 6.1	系统用户	管理员	N/A	本地用户

3.6 检查主机依赖软件状态

本章节介绍如何检查集群中工作节点上主机依赖软件状态是否正常。本例中主机操作系统为CentOS 7.5 x86_64。

- 检查NFS客户端状态。
systemctl status rpcbind

4 安装华为 CSI

本章节介绍如何安装部署华为CSI

华为CSI的安装支持root用户和非root用户。使用非root用户安装华为CSI时，需要保证当前用户能够访问Kubernetes集群的API Server，配置非root用户访问Kubernetes集群请参考[8.7 配置非root用户访问Kubernetes集群](#)。

华为CSI必须在root用户权限下运行。

[4.1 创建华为CSI运行时需要的ConfigMap](#)

[4.2 启动huawei-csi服务](#)

4.1 创建华为 CSI 运行时需要的 ConfigMap

ConfigMap 是一种 API 对象，用来将非机密性的数据保存到键值对中。华为CSI启动运行时，需要使用在本章节创建的ConfigMap中包含的信息作为重要的运行参数。

4.1.1 通过 NFS 对接企业存储 NAS

当您需要通过NFS对接企业存储NAS配置时，执行此操作。

前提条件

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

操作步骤

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

步骤2 执行`vi huawei-csi-configmap.yaml`命令，创建huawei-csi-configmap.yaml文件。

```
# vi huawei-csi-configmap.yaml
```

步骤3 配置huawei-csi-configmap.yaml文件， huawei-csi-configmap.yaml文件的模板如回显所示（您也可以参考软件包中deploy/huawei-csi-configmap/huawei-csi-configmap-oceanstor-nfs.yaml示例文件）。请根据实际情况设置相关参数，并保存为yaml格式的文件，详细请参考[表4-1](#)。

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

表 4-1 配置项描述

配置项	参数格式	描述	备注
data."csi.json".backends	列表	必填，待接入后端存储设备列表。	配置的后端存储设备数量没有限制。 单个后端存储设备支持配置的字段，请参考 表4-2 。

表 4-2 后端存储设备配置项描述

配置项	参数格式	描述	备注
storage	字符串	必填，待接入存储设备的类型。	对接企业存储NAS场景固定填写"oceanstor-nas"。
name	字符串	存储后端名。	自定义字符串，支持大小写字母、数字、中划线组合。 说明 如果需要配置多个存储后端，请保证存储后端名唯一。

配置项	参数格式	描述	备注
urls	列表	必填，待接入的存储设备的管理URL。	支持同一存储设备的一个或多个管理URL，用逗号分割，当前仅支持IPv4。例如：https://192.168.125.20:8088 说明 1个存储设备有多个控制器，每个控制器有一个管理URL，所以1个存储设备存在多个管理URL。
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。
parameters	字典	必填，NFS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。 逻辑端口IP地址可通过登录DeviceManager获取。以OceanStor Dorado 6.x系列为例：在DeviceManager管理界面，选择“服务 > 网络 > 逻辑端口”，获取数据协议类型为NFS的IP（其它系列请参照相应的操作说明进行获取）。

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

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

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

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

----结束

4.1.2 通过租户/账户对接华为存储后端配置

如果对接的华为存储时需要使用租户/账户来对接存储做资源隔离时，需要为每个后端配置租户/账户信息。如果未配置租户/账户信息时，CSI系统将使用存储系统默认的租户/账户创建资源。

说明

如果租户使用的后端是OceanStor 6.1或OceanStor Dorado 6.x存储，需要为该后端配置对应租户的指定租户的逻辑管理端口URL。请注意“urls”为租户的逻辑管理端口，“pools”和“portals”需要确保是当前租户可用的存储池和数据逻辑端口。

4.1.3 通过 CSI 对接多个后端存储的配置

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[4.1.1 通过NFS对接企业存储NAS](#)章节的说明。

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: huawei-csi-configmap
  namespace: huawei-csi
data:
  csi.json: |
    {
      "backends": [
        {
          "storage": "****",
          "name": "backend1",
          ...
        },
        {
          "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 huawei-csi | grep huawei-csi-configmap**命令，查看是否创建成功。显示如下回显时，则表示创建成功。

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

----结束

4.2 启动 huawei-csi 服务

本章节介绍如何启动huawei-csi服务。

前提条件

- 华为CSI镜像已制作完成，并且按照[3.3 上传华为CSI镜像](#)章节说明，上传到镜像仓库或者导入到所有节点。
- 华为CSI安装运行所依赖的组件镜像都已经上传到镜像仓库或者导入到所有节点。具体信息请参考[3.4 检查CSI依赖的镜像](#)章节说明。
- Kubernetes的所有节点与待接入的存储设备管理IP通信正常，且允许使用ping命令进行连通性校验。
- Kubernetes的所有worker节点与待接入的存储设备业务IP通信正常，且允许使用ping命令进行连通性校验。。
- Kubernetes的所有worker节点已安装对应协议所需要的软件客户端，如NFS客户端等。
- 已经在需要对接的华为存储上创建了对接华为CSI所需要的账号，具体信息请参考[3.5 检查华为存储上的账号配置](#)章节说明。

操作步骤

- 步骤1** 使用远程访问工具（以PuTTY为例），通过管理IP地址，登录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[表3-1](#)。
- 步骤3** 使用加密工具填写存储设备的用户名和密码。

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

Number	Configured	BackendName	Urls
1	false	strage-backend	[https://192.168.125.25:8088]
2	false	strage-backend-02	[https://192.168.125.26:8088]
3	false	strage-backend-03	[https://192.168.125.27:8088]
4	false	strage-backend-04	[https://192.168.125.28:8088]
5	false	strage-backend-05	[https://192.168.125.29:28443]
6	false	strage-backend-06	[https://192.168.125.30:28443]

Please enter the backend number to configure (Enter 'exit' to exit):3
3. 根据界面提示输入用户名和密码来创建secret对象。
*****Current backend information*****
Name:strage-backend-03
Urls:[https://192.168.125.27:8088]
Please enter this backend user name:admin

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

- 配置完成后，输入exit退出并保存配置。
Please enter the backend number to configure (Enter 'exit' to exit):*exit*
Saving configuration. Please wait.....
The configuration is saved successfully.
- 使用**kubectl get secret -n huawei-csi | grep huawei-csi-secret**命令检查secret对象是否已创建成功。
kubectl -n huawei-csi get secret huawei-csi-secret

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

步骤4 执行以下命令，创建RBAC权限。

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

步骤5 启动controller服务。

- 执行**vi huawei-csi-controller.yaml**命令，修改yaml文件。按I或Insert进入编辑状态，修改以下相关参数。修改完成后，按Esc，并输入:wq!，保存修改。

说明

- （必选）示例yaml文件中huawei-csi-driver的参数image配置项，修改huawei-csi:.*为[3.3 上传华为CSI镜像](#)章节中上传的华为CSI镜像<名称>:<版本号>。

containers:

```
...
- name: huawei-csi-driver
  image: repo.huawei.com/huawei-csi:unionpay
```

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

metadata:

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

- 执行以下命令，启动controller服务。

```
# kubectl apply -f huawei-csi-controller.yaml
```

步骤6 启动node服务。

- 执行**vi huawei-csi-node.yaml**命令，修改yaml文件。按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。

说明

- （必选）示例yaml文件中huawei-csi-driver的参数image配置项，修改huawei-csi:.*为[3.3 上传华为CSI镜像](#)章节中上传的华为CSI镜像<名称>:<版本号>。

containers:

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

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

metadata:

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

- 执行以下命令，启动node服务。

```
# kubectl apply -f huawei-csi-node.yaml
```

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

```
# kubectl get pod --all-namespaces | grep huawei
huawei-csi huawei-csi-controller-695b84b4d8-tg64l 7/7 Running 0 14s
huawei-csi huawei-csi-node-g6f7z 3/3 Running 0 14s
```

----结束

5 卸载华为 CSI

本章节介绍如何卸载华为CSI。

⚠ 注意

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

[5.1 卸载huawei-csi-node服务](#)

[5.2 卸载huawei-csi-controller服务](#)

[5.3 删除huawei-csi-configmap对象](#)

[5.4 删除huawei-csi-secret对象](#)

[5.5 删除RBAC权限](#)

[5.6 删除老版本镜像](#)

5.1 卸载 huawei-csi-node 服务

操作步骤

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

步骤2 执行**kubectl delete daemonset huawei-csi-node -n huawei-csi**命令卸载huawei-csi-node服务，huawei-csi替换为华为CSI所在的命名空间。

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

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

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

----结束

5.2 卸载 huawei-csi-controller 服务

操作步骤

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

步骤2 执行**kubectl delete deployment huawei-csi-controller -n huawei-csi**命令卸载huawei-csi-controller服务，huawei-csi替换为华为CSI所在的命名空间。

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

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

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

----结束

5.3 删除 huawei-csi-configmap 对象

操作步骤

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

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

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

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

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

----结束

5.4 删除 huawei-csi-secret 对象

操作步骤

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

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

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

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

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

----结束

5.5 删除 RBAC 权限

操作步骤

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

步骤2 删除RBAC权限。

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

步骤3 按照以下步骤检查RBAC权限是否已成功删除。

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

----结束

5.6 删除老版本镜像

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

以删除单个节点的镜像为例，操作步骤如下。

前提条件

依赖该镜像的容器服务已经被停止，否则无法删除。

操作步骤

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

步骤2 执行以下命令，查看现存的所有版本。

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

```
# docker image ls | grep huawei-csi  
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE  
huawei-csi     2.2.15    b30b3a8b5959   2 weeks ago    79.7MB  
huawei-csi     unionpay  14b854dba227   2 weeks ago    79.6MB
```

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

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

步骤3 执行以下命令，删除老版本镜像：

- 如果使用的docker，执行**docker rmi <REPOSITORY>:<TAG>**命令。

```
# docker rmi huawei-csi:2.2.15
```

- 如果使用的containerd，执行**crictl rmi <REPOSITORY>:<TAG>**命令。

```
# crictl rmi huawei-csi:2.2.15
```

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

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


```
# docker image ls | grep huawei-csi
huawei-csi                unionpay      14b854dba227  10 minutes ago  80MB
```

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

```
# crictl image ls | grep huawei-csi
docker.io/library/huawei-csi  unionpay      14b854dba2273  93.1MB
```

----结束

6 使用华为 CSI

本章节主要介绍如何使用华为CSI对持久化卷、快照的生命周期进行管理。

6.1 持久卷管理

6.1 持久卷管理

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

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

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

本章将介绍如何使用华为CSI对持久卷/持久卷申领进行创建。

6.1.1 创建 PVC

华为CSI支持在华为存储上创建存储资源（文件系统），并根据用户的设置供给容器使用。具体支持的特性请参考[表2-4](#)。

根据持久卷创建的方式分为动态卷供应和静态卷供应。

动态卷供应不需要事先创建持久卷，在使用根据存储类自动的在存储设备上创建持久卷所需要的资源。并且可以在创建持久卷申领时同时创建持久卷。

静态卷供应需要管理员事先在存储设备上创建好所需要的资源，通过创建持久卷的方式使用已存在的资源。并且可以在创建持久卷申领指定关联的持久卷。

6.1.1.1 动态卷供应

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

为了完成动态卷供应，需要完成如下两步：

- 配置存储类
- 配置PVC

6.1.1.1.1 配置存储类

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

存储类支持配置如下参数信息。

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

表 6-1 存储类配置参数说明

参数	说明	备注
metadata.name	自定义的StorageClass对象名称。必选	以Kubernetes v1.12.3为例，支持数字、小写字母、中划线 (-) 和点 (.) 的组合，并且必须以字母数字字符开头和结尾。
provisioner	制备器名称，必选。	该字段需要指定为安装华为CSI时设置的驱动名。 默认的驱动名为“csi.huawei.com”。
reclaimPolicy	回收策略。可选。支持如下类型： Delete：自动回收资源。 Retain：手动回收资源。	自动回收资源：删除PV/PVC时会关联删除存储上的资源。 手动回收资源：删除PV/PVC时不会删除存储上的资源。
parameters.backend	待创建资源所在的后端名称。可选。	如果不设置，华为CSI随机选择一个满足容量要求的后端创建资源。 建议指定后端，防止创建的资源不在预期的后端上。
parameters.pool	待创建资源所在的存储资源池名称。可选设置。如果设置，则必须设置parameters.backend。	如果不设置，华为CSI会在所选后端上随机选择一个满足容量要求的存储池创建资源。建议指定存储池，防止创建的资源不在预期的存储池上。
parameters.volumeType	待创建卷类型。支持如下类型： fs：存储侧发放的资源是NAS。	--

参数	说明	备注
parameters.alloc Type	待创建卷的分配类型。 可选。支持如下类型： thin：创建时不会分配所有需要的空间，而是根据使用情况动态分配。 thick：创建时分配所有需要的空间。	如果不设置，默认为thin。
parameters.auth Client	指定可访问该卷的NFS客户端IP地址信息，在指定volumeType为“fs”时必选。 支持输入客户端主机名称（建议使用全称域名）、客户端IP地址、客户端IP地址段。	可以使用“*”表示任意客户端。当您不确定访问客户端IP信息时，建议使用“*”防止客户端访问被存储拒绝。 当使用客户端主机名称时建议使用全称域名。 IP地址支持IPv4、IPv6地址或两者的混合IP地址。 可以同时输入多个主机名称、IP地址或IP地址段，以英文分号，空格或按回车键隔开。如示例： "192.168.0.10;192.168.0.0/24;myserver1.test"
parameters.applic ationType	后端为Dorado V6存储时，指定创建NAS时的应用类型名称。	“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。
parameters.qos	指定PV在存储侧的NAS的QoS设置。可选。 配置项值是字典格式的json字符串（字符串两边由单引号修饰，字典key由双引号修饰）。 如：'{"maxMBPS": 999, "maxIOPS": 999}'	支持的QoS配置请参考 表6-2 说明。
parameters.fsPer mission	可选。指定挂载到容器内的目录权限。	配置格式参考linux权限设置，如“777”、“755”等。 NAS存储仅支持OceanStor Dorado V6

参数	说明	备注
parameters.rootSquash	可选。用于设置是否允许客户端的root权限。 可选值： - root_squash：表示不允许客户端以root用户访问，客户端使用root用户访问时映射为匿名用户。 - no_root_squash：表示允许客户端以root用户访问，保留root用户的权限。	仅支持NAS存储。
parameters.allSquash	可选。用于设置是否保留共享目录的UID和GID。 可选值： - all_squash：表示共享目录的UID和GID映射为匿名用户。 - no_all_squash：表示保留共享目录的UID和GID。	仅支持NAS存储。
parameters.snapshotDirectoryVisibility	可选。用于设置快照目录是否可见。 可选值： - visible：表示快照目录可见。 - invisible：表示快照目录不可见。	仅支持NAS存储。

表 6-2 支持的 QoS 配置

存储类型	参数名	参数描述	备注
Dorado V6	IOTYPE	控制读写类型	有效值如下： • 2：读写IO
	MAXBANDWIDTH	最大带宽策略	单位MB/s，类型为整数，范围1~999999999
	MINBANDWIDTH	最小带宽策略	单位MB/s，类型为整数，范围1~999999999

存储类型	参数名	参数描述	备注
	MAXIOPS	最大IOPS策略	类型为整数，范围100~999999999
	MINIOPS	最小IOPS策略	类型为整数，范围100~999999999
	LATENCY	最大时延策略	单位ms，仅支持配置0.5或1.5

使用NAS典型场景的存储类配置请参考如下示例：

- [存储类中设置后端和存储池](#)
- [存储类中设置QoS](#)

存储类中设置后端和存储池

如果在一个Kubernetes集群中配置了多个华为后端，或者一个华为后端提供多个存储池，建议在存储类中配置指定的后端和存储池信息，避免华为CSI随机选择后端和存储池，导致卷所在的存储不符合规划。设置后端和存储池可以参考如下配置示例。

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: mysc
provisioner: csi.huawei.com
parameters:
  backend: nfs_dorado_181
  pool: pool001
volumeType: fs
allocType: thin
authClient: ""
```

存储类中设置 QoS

容器使用企业存储作为存储资源时，可以为某些重要的容器使用的存储资源设置QoS，从而保证这些容器对存储读写满足一定的服务等级。

不同型号或版本的存储支持的QoS设置不同，请参考[表6-2](#)找到对应存储的配置项。本示例中的后端是Dorado V6存储，其他存储可以参考本例设置。

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: mysc
provisioner: csi.huawei.com
parameters:
  backend: nfs_dorado_181
  pool: pool001
volumeType: fs
allocType: thin
authClient: ""
qos: '{"IOTYPE": 2, "MINIOPS": 1000}'
```

完成存储类配置后，进行如下步骤创建存储类。

步骤1 执行以下命令，基于该yaml文件创建StorageClass。

```
# kubectl create -f mysc.yaml
```

步骤2 执行以下命令，查看当前已经创建的StorageClass信息。

```
# kubectl get sc
NAME PROVISIONER RECLAIMPOLICY VOLUMEBINDINGMODE ALLOWVOLUMEEXPANSION AGE
mysc  csi.huawei.com Delete Immediate false 34s
```

创建存储类后，就可以使用该存储类进行创建PV或者PVC。

----结束

注意

在存储类的使用中请注意如下事项：

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

6.1.1.1.2 配置 PVC

在完成配置存储类以后，就可以存储类来配置PVC。PVC的配置模板请参考华为CSI软件包中的examples目录下的pvc-xxxx.yaml文件示例。

表 6-3 pvc-xxxx.yaml 文件示例参数说明

参数	说明	备注
metadata.name	自定义的PVC对象名称	以Kubernetes v1.12.3为例，支持数字、小写字母、中划线（-）和点（.）的组合，并且必须以字母数字字符开头和结尾。
spec.storageClassName	存储类对象名称	业务需要的存储类对象名称。
spec.resources.requests.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 6.1.2对接CentOS 7为例，当使用的是EXT4文件系统时，容量限制见 表6-4 ；当使用的是XFS文件系统时，容量限制见 表6-5 。如果使用的是NFS，容量需满足使用的华为存储设备型号和版本所要求的规格约束。 如果PVC容量不在规格范围内，可能会由于存储规格限制或主机文件系统规格限制导致创建PVC或Pod失败。

参数	说明	备注
spec.accessModes	指定卷访问模式。 RWO (ReadWriteOnce) : 卷可以被一个节点以读写方式挂载。该模式也允许运行在同一节点上的多个 Pod 访问卷。 ROX (ReadOnlyMany) : 卷可以被多个节点以只读方式挂载。 RWX (ReadWriteMany) : 卷可以被多个节点以读写方式挂载。	RWO/ROX/RWX: 所有类型卷均支持。

表 6-4 EXT4 容量的规格

存储类型	存储规格限制	ext4规格限制	CSI规格限制
OceanStor Dorado 6.1.2	512Ki~256Ti	50Ti	512Ki~50Ti

表 6-5 XFS 容量的规格

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

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

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

步骤2 执行命令，使用配置文件创建PVC。

```
# kubectl create -f mypvc.yaml
```

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


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

⚠ 注意

- 完成创建PVC操作后，如果长时间后（如一分钟后）PVC的状态是Pending，请参考[9.3 创建PVC时，PVC的状态为Pending](#)。
- 建议每批次最多批量创建/删除100个PVC。

----结束

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

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  selector:
    matchLabels:
      app: nginx
  replicas: 2
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - image: nginx:alpine
          name: container-0
          volumeMounts:
            - mountPath: /tmp
              name: pvc-mypvc
          restartPolicy: Always
      volumes:
        - name: pvc-mypvc
          persistentVolumeClaim:
            claimName: mypvc # name of PVC
```

6.1.1.2 静态卷供应

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

为了完成静态卷供应，需要完成如下两步：

- 配置PV
- 配置PVC

6.1.1.2.1 配置 PV

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

```
kind: PersistentVolume
apiVersion: v1
metadata:
  name: mypv
```

```
spec:
  storageClassName: ""
  accessModes:
    - ReadWriteOnce
  csi:
    driver: csi.huawei.com
    volumeHandle: nfs_dorado_181.fs0001
  capacity:
    storage: 100Gi
```

如上例所示，静态卷供应的配置文件中，storageClassName参数必须配置为‘ “ ” ’，如果不配置，否则K8s会使用系统默认的存储类。其他参数及说明见下表。

表 6-6 静态卷供应参数

参数	说明	备注
metadata.name	自定义的PVC对象名称	以Kubernetes v1.12.3为例，支持数字、小写字母、中划线（-）和点（.）的组合，并且必须以字母数字字符开头和结尾。
spec.storageClassName	StorageClass对象名称，必选。	此处须设置为空字符串（即输入""）。
spec.accessModes	指定卷访问模式。 RWO （ReadWriteOnce）：卷可以被一个节点以读写方式挂载。该模式也允许运行在同一节点上的多个Pod访问卷。 ROX（ReadOnlyMany）：卷可以被多个节点以只读方式挂载。 RWX （ReadWriteMany）：卷可以被多个节点以读写方式挂载。	RWO/ROX/RWX：所有类型卷均支持。
spec.csi.driver	CSI驱动名称	固定填写csi.huawei.com。
spec.csi.volumeHandle	存储资源的唯一标志，必选。 格式为 <backendName>.<volume-name>	该参数值由以下两部分构成： <ul style="list-style-type: none">• <backendName>：该卷所在的后端名称，可使用如下命令获取配置的后端信息： kubectl get configmap huawei-csi-configmap -n huawei-csi -o yaml• <volume-name>：存储上资源（文件系统）的名称，可通过DeviceManager查看。

参数	说明	备注
spec.capacity.storage	指定卷大小	请确保与存储上对应资源的容量保持一致。Kubernetes并不会调用CSI检查此字段值的正确性，所以在PV容量与存储上对应资源的容量不一致也能被成功创建。

前提条件

存储侧已经存在待创建PV所需要的存储资源。如文件系统。如果存储资源时文件系统时，还需要创建文件系统的共享和客户端信息。

已参考表6-6，完成PV的配置文件。

操作步骤

步骤1 执行以下命令，基于准备好的yaml文件创建PV。

```
# kubectl create -f mypv.yaml
```

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

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

----结束

6.1.1.2.2 配置 PVC

当持久卷以静态卷供应的方式创建完成后，可以基于该持久卷创建持久卷申领，从而供容器使用。如下示例是一个使用静态卷供应的PVC配置文件。

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

如上例所示，通过设置PVC配置文件中“**volumeName**”参数为之前通过静态卷供应创建的PV。具体参数见下表。

表 6-7 PVC 参数说明

参数	说明	备注
metadata.name	自定义的PVC对象名称	以Kubernetes v1.12.3为例，支持数字、小写字母、中划线（-）和点（.）的组合，并且必须以字母数字字符开头和结尾。
spec.accessModes	指定卷访问模式。 RWO （ReadWriteOnce）：卷可以被一个节点以读写方式挂载。该模式也允许运行在同一节点上的多个 Pod 访问卷。 ROX（ReadOnlyMany）：卷可以被多个节点以只读方式挂载。 RWX （ReadWriteMany）：卷可以被多个节点以读写方式挂载。	RWO/ROX/RWX：所有类型卷均支持。
spec.resources.requests.storage	指定待创建卷大小	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 6.1.2/OceanStor Pacific系列 8.1.0对接CentOS 7为例，当使用的是EXT4文件系统时，容量限制见表6-4；当使用的是XFS文件系统时，容量限制见表6-5。如果使用的是NFS或者裸设备，容量需满足使用的华为存储设备型号和版本所要求的规格约束。 如果PVC容量不在规格范围内，可能会由于存储规格限制或主机文件系统规格限制导致创建PVC或Pod失败。 在通过静态PV创建PVC时，若PVC容量小于绑定PV容量，最终PVC容量大小为绑定PV容量，若PVC容量大于绑定PV容量，PVC将无法被创建。
spec.volumeName	PV对象名称	静态创建PVC时必选。

操作步骤

步骤1 执行以下命令，基于已配置的yaml文件创建PVC。

```
# kubectl create -f mypvc.yaml
```

步骤2 等待一段时间后，执行以下命令，查看已经创建的PVC信息。

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

说明

- 完成创建PVC操作后，如果长时间后（如一分钟后）PVC的状态是Pending，请参考[9.3 创建PVC时，PVC的状态为Pending](#)。
- 建议每批次最多批量创建/删除100个PVC。

----结束

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

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  selector:
    matchLabels:
      app: nginx
  replicas: 2
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - image: nginx:alpine
          name: container-0
          volumeMounts:
            - mountPath: /tmp
              name: pvc-mypvc
          restartPolicy: Always
      volumes:
        - name: pvc-mypvc
          persistentVolumeClaim:
            claimName: mypvc # name of PVC
```

7 高级特性

7.1 配置存储拓扑感知

7.1 配置存储拓扑感知

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

前提条件

需要在集群中的worker节点完成拓扑的标签配置，标签配置方法如下：

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

```
# kubectl get node
NAME      STATUS    ROLES          AGE  VERSION
node01    Ready     controlplane,etcd,worker  42d  v1.22.3
node02    Ready     worker          42d  v1.22.3
node03    Ready     worker          42d  v1.22.3
```
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7-1。

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

表 7-1 参数说明

参数名	参数描述	备注
<key>	拓扑标签的唯一标识。	可支持配置：zone, region, protocol.<protocol> 其中<protocol>可支持配置nfs。

参数名	参数描述	备注
<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.<protocol>，“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.nfs=csi.huawei.com
 - 同一节点上拓扑标签中同一个key只能支持一个value值。
 - 如果同一节点上拓扑标签中同时配置多个protocol，选择后端时，只需要满足其中一个protocol即可。
 - 如果同一节点上拓扑标签中同时配置region和zone，选择后端时，需要满足全部。
4. 执行**kubectl get nodes -o=jsonpath='{range .items[*]}{.metadata.name},{.metadata.labels}{"\n"}{end}' | grep --color "topology.kubernetes.io"**命令，查看当前集群中所有worker节点的标签信息。
- ```
kubectl get nodes -o=jsonpath='{range .items[*]}{.metadata.name},{.metadata.labels}{"\n"}{end}' | grep --color "topology.kubernetes.io"
[node01, {"beta.kubernetes.io/arch":"amd64","beta.kubernetes.io/os":"linux","kubernetes.io/arch":"amd64","kubernetes.io/hostname":"node01","kubernetes.io/os":"linux","node-role.kubernetes.io/controlplane":"true","node-role.kubernetes.io/etcd":"true","node-role.kubernetes.io/worker":"true","topology.kubernetes.io/zone":"Cheng
```

## 7.1.1 配置步骤

### 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。

多个后端间使用“,”进行分割，每个后端内容可以参考[4.1 创建华为CSI运行时需要的ConfigMap](#)。

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-nas",
 ...
 "parameters": {"protocol": "nfs", "portals": ["192.168.125.22", "192.168.125.23"]},
 "supportedTopologies": [
 {"topology.kubernetes.io/region": "China-west", "topology.kubernetes.io/zone": "ChengDu"},
 {"topology.kubernetes.io/region": "China-south", "topology.kubernetes.io/zone": "ShenZhen"}
]
 }
]
}
```

```
]
}
```

### 说明

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

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

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

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

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

**步骤5** 启动 huawei-csi 服务，详细请参见[4.2 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7-2](#)。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: fs
 allocType: thin
 authClient: "*"
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: fs
 allocType: thin
 authClient: "*"
volumeBindingMode: WaitForFirstConsumer
allowedTopologies:
- matchLabelExpressions:
 - key: topology.kubernetes.io/protocol.nfs
 values:
 - csi.huawei.com
```



表 7-2 参数说明

| 参数名                                  | 参数描述                                                                                             | 备注                                                                                                                                                                                                                                                                                                                                                                       |
|--------------------------------------|--------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>volumeBindingMode</b>             | 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”<br>“WaitForFirstConsumer”：表示延迟 PersistentVolume 的绑定和调配，直到创建使用 PVC 的 Pod。<br>“Immediate”：表示创建 PVC 后，立即发生 PersistentVolume 绑定和调配。                                                                                                                                                                                                 |
| <b>allowedTopologies.matchLabels</b> | 拓扑信息标签，用于过滤 CSI 后端和 Kubernetes 节点。如果匹配失败，会导致 PVC 或 Pod 无法创建。<br>配置时需要同时按照固定格式配置 “key” 和 “value”。 | “key”：可支持配置<br>“topology.kubernetes.io/zone”，<br>“topology.kubernetes.io/region”，<br>topology.kubernetes.io/<br>protocol.<protocol>，其中<protocol>为协议类型，例如：nfs。<br>“value”：<br>“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。<br>“key” 如果是 topology.kubernetes.io/protocol.<protocol>，“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 StorageClass.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[6.1.1 创建PVC](#)

----结束

# 8 常用操作

8.1 更新CSI上配置的存储用户名或密码

8.2 更新huawei-csi的configmap对象

8.3 为huawei-csi新增后端

8.4 更新huawei-csi-controller服务

8.5 更新huawei-csi-node服务

8.6 修改日志输出模式

8.7 配置非root用户访问Kubernetes集群

## 8.1 更新 CSI 上配置的存储用户名或密码

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

### 操作步骤

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

**步骤2** 执行**chmod +x secretUpdate**命令，给secretUpdate添加可执行权限。

```
chmod +x secretUpdate
```

**步骤3** 执行**./secretUpdate namespace**命令，运行secretUpdate工具，并根据界面提示输入需要配置backend的编号。Configured为false时表示这个backend尚未配置，为true时表示这个backend已配置。

```
./secretUpdate huawei-csi
Getting backend configuration information.....

Number Configured BackendName Urls
1 true strage-backend [https://192.168.125.25:8088]
2 true strage-backend-02 [https://192.168.125.26:8088]
3 true strage-backend-03 [https://192.168.125.27:8088]
4 true strage-backend-04 [https://192.168.125.28:8088]
5 true strage-backend-05 [https://192.168.125.29:28443]
6 true strage-backend-06 [https://192.168.125.30:28443]

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

**步骤4** 根据界面提示输入用户名和密码来更新secret对象。

```
The backend strage-backend-05 has been configured. Update? [Y/N]:y
*****Current backend information*****
Name:strage-backend-03
Urls:[https://192.168.125.27:8088]
Please enter this backend user name:admin
Please enter this backend password:
Verifying user name and password. Please wait.....
The account information of the backend strage-backend-03 has been configured successfully.
```

**步骤5** 配置完成后，输入exit退出并保存配置。

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

**步骤6** 执行以下命令，重启huawei-csi-controller服务。

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

**步骤7** 执行以下命令，重启huawei-csi-node服务。

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

**步骤8** 执行**kubectl get pod --all-namespaces | grep huawei**命令检查服务是否重启成功。

```
kubectl get pod --all-namespaces | grep huawei
huawei-csi huawei-csi-controller-695b84b4d8-tg64l 7/7 Running 0 14s
huawei-csi huawei-csi-node-g6f7z 3/3 Running 0 14s
```

----结束

## 8.2 更新 huawei-csi 的 configmap 对象

当您需要在已有的后端上增加存储池，或者更改已有的业务IP地址时，执行此操作。

### 操作步骤

**步骤1** 使用远程访问工具（以PuTTY为例），通过管理IP地址，登录Kubernetes集群的任意master节点。**步骤2** 执行**kubectl edit configmap huawei-csi-configmap -n huawei-csi**命令，按**I**或**Insert**进入编辑状态，修改相关参数。详细参数说明请参见[表8-1](#)。修改完成后，按**Esc**，并输入**:wq!**，保存修改。

```
kind: ConfigMap
apiVersion: v1
metadata:
 name: huawei-csi-configmap
 namespace: huawei-csi
data:
 csi.json: |
 {
 "backends": [
 {
 "storage": "oceanstor-nas",
 "name": "storage",
 "urls": ["https://192.168.125.20:8088"],
 "pools": ["storagepool01"],
 "parameters": {"protocol": "nfs", "portals": ["192.168.125.22"]}
 }
]
 }
}
```

表 8-1 配置项描述

| 配置项                      | 参数格式 | 描述              | 备注                                           |
|--------------------------|------|-----------------|----------------------------------------------|
| data."csi.json".backends | 列表   | 必填，待接入后端存储设备列表。 | 配置的后端存储设备数量没有限制。<br>单个后端存储设备支持配置的字段，请参考表8-2。 |

表 8-2 后端存储设备配置项描述

| 配置项        | 参数格式 | 描述                     | 备注                                                                                                                                                                                                  |
|------------|------|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| storage    | 字符串  | 必填，待接入存储设备的类型。         | 固定填写"oceanstor-nas"。                                                                                                                                                                                |
| name       | 字符串  | 存储后端名。                 | <ul style="list-style-type: none"><li>自定义字符串，支持大小写字母、数字、中划线组合。</li><li>该参数不支持修改。</li></ul>                                                                                                          |
| 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<br><b>说明</b><br>1个存储设备有多个控制器，每个控制器有1个管理URL，所以1个存储设备可能存在多个管理URL。                                                                  |
| pools      | 列表   | 必填，被使用的待接入存储设备上的存储池名称。 | <ul style="list-style-type: none"><li>支持同一存储设备上的一个或多个存储池，用逗号分割。</li><li>当前只支持新增存储池。</li><li>可通过登录DeviceManager获取支持块存储服务的存储池。</li></ul>                                                              |
| parameters | 字典   | NFS场景的可变参数。            | protocol参数固定填写“nfs”。<br>portals：指定存储的逻辑端口IP或者DNS Zone，只支持配置一个。<br>逻辑端口IP地址可通过登录DeviceManager获取。以OceanStor Dorado 6.x系列为例：在DeviceManager管理界面，选择“服务 > 网络 > 逻辑端口”，获取数据协议类型为NFS的IP（其它系列请参照相应的操作说明进行获取）。 |

**步骤3** 若后端参数storage、name或urls有修改，需要更新存储的用户名或密码。详细操作请参见[8.1 更新CSI上配置的存储用户名或密码](#)。

**步骤4** 执行以下命令，重启huawei-csi-controller服务。

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

**步骤5** 执行以下命令，重启huawei-csi-node服务。

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

**步骤6** 执行**kubectl get pod --all-namespaces | grep huawei**命令，检查服务是否重启成功。

```
kubectl get pod --all-namespaces | grep huawei
huawei-csi huawei-csi-controller-695b84b4d8-tg64l 7/7 Running 0 14s
huawei-csi huawei-csi-node-g6f7z 3/3 Running 0 14s
```

----结束

## 8.3 为 huawei-csi 新增后端

当您需要新增存储设备或存储池当做独立的后端时，执行此操作。

### 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[4.1.3 通过CSI对接多个后端存储的配置](#)。

**步骤3** 为新增的后端配置账号，详细操作请参见[8.1 更新CSI上配置的存储用户名或密码](#)。

**步骤4** 执行以下命令，重启huawei-csi-controller服务。

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

**步骤5** 执行以下命令，重启huawei-csi-node服务。

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

**步骤6** 执行**kubectl get pod --all-namespaces | grep huawei**命令，检查服务是否重启成功。

```
kubectl get pod --all-namespaces | grep huawei
huawei-csi huawei-csi-controller-695b84b4d8-tg64l 7/7 Running 0 14s
huawei-csi huawei-csi-node-g6f7z 3/3 Running 0 14s
```

----结束

## 8.4 更新 huawei-csi-controller 服务

当您需要更新huawei-csi-controller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** 卸载huawei-csi-controller服务，请参见[5.2 卸载huawei-csi-controller服务](#)。

**步骤3** 删除RBAC权限，请参见[5.5 删除RBAC权限](#)。

**步骤4** 创建RBAC权限，请参见[创建RBAC权限](#)。

**步骤5** 启动controller服务，请参见[启动controller服务](#)。

**步骤6** 完成huawei-csi服务部署后，可执行**kubect**l get pod --all-namespaces | grep **huawei-csi-controller**命令检查服务是否启动。

```
kubectl get pod --all-namespaces | grep huawei-csi-controller
huawei-csi huawei-csi-controller-695b84b4d8-tg64l 7/7 Running 0 14s
```

----结束

## 8.5 更新 huawei-csi-node 服务

当您需要更新huawei-csi-node服务时，执行此操作。

### 操作步骤

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

**步骤2** 卸载huawei-csi-node服务，请参见[5.1 卸载huawei-csi-node服务](#)。

**步骤3** 安装新的huawei-csi-node服务，请参见[启动node服务](#)。

**步骤4** 完成huawei-csi服务部署后，可执行**kubect**l get pod --all-namespaces | grep **huawei-csi-node**命令检查服务是否启动。

```
kubectl get pod --all-namespaces | grep huawei-csi-node
huawei-csi huawei-csi-node-g6f7z 3/3 Running 0 14s
```

----结束

## 8.6 修改日志输出模式

huawei-csi支持两种日志输出模式，分别是file和console。file指的是输出到固定的日志目录（例如：/var/log/huawei）；console指的是输出到容器标准目录。用户可以根据自身需求自行设置日志输出模式，默认为file。

### 8.6.1 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** 卸载huawei-csi-controller服务，请参见[5.2 卸载huawei-csi-controller服务](#)。

**步骤3** 进入“deploy”目录下，组件包路径参考[表3-1](#)。

**步骤4** 执行**vi huawei-csi-controller.yaml**命令，修改yaml文件。按I或Insert进入编辑状态，修改以下相关参数，详细请参考[表8-3](#)。修改完成后，按Esc，并输入:wq!，保存修改。

```
args:
- "--endpoint=$(CSI_ENDPOINT)"
- "--controller"
```

```
- "--containerized"
- "--driver-name=csi.huawei.com"
- "--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                                                     |
|---------------|---------------------------------|--------------------------------------------------------|
| loggingModule | 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。      |
| logFileDir    | 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”。 |
| logFileSize   | 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。           |
| maxBackups    | 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-snapshot-v1.yaml
```

**步骤6** 完成huawei-csi服务部署后，可执行**kubectl get pod --all-namespaces -o wide | grep huawei**命令检查服务是否启动。

```
kubectl get pod --all-namespaces -o wide | grep huawei
huawei-csi huawei-csi-controller-b59577886-qzrm8 7/7 Running 0 18h 10.244.1.67
node <none> <none>
```

**步骤7** 查看huawei-csi-controller服务的日志。

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

```
tail -f huawei-csi-controller
```

- 如果设置“loggingModule”为“console”，请执行以下命令，查看huawei-csi-controller日志

```
kubectl logs huawei-csi-controller -c huawei-csi-driver -n huawei-csi
```

----结束

## 8.6.2 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** 卸载huawei-csi-node服务，请参见[5.1 卸载huawei-csi-node服务](#)。

**步骤3** 执行**vi huawei-csi-node.yaml**命令，修改yaml文件。按**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deploy/huawei-csi-node.yaml示例文件）。参数详情可参考[表 8-4](#)。

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

表 8-4 日志输出参数说明

| 配置项           | 描述                             | 备注                                                   |
|---------------|--------------------------------|------------------------------------------------------|
| loggingModule | 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。    |
| logFileDir    | 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。 |
| logFileSize   | 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。         |
| maxBackups    | 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4** 执行以下命令，启动node服务。

```
kubectl create -f huawei-csi-node.yaml
```

**步骤5** 完成huawei-csi服务部署后，可执行**kubectl get pod --all-namespaces -o wide | grep huawei-csi-node**命令检查服务是否启动。

```
kubectl get pod --all-namespaces | grep huawei-csi-node
huawei-csi huawei-csi-node-4sfwr 3/3 Running 0 18h 10.244.1.68 node
<none> <none>
```

**步骤6** 查看huawei-csi-node服务的日志。

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



```
tail -f huawei-csi-node
```

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

```
kubectl logs huawei-csi-node -c huawei-csi-driver -n huawei-csi
```

----结束

## 8.7 配置非 root 用户访问 Kubernetes 集群

### 操作步骤

**步骤1** 拷贝Kubernetes集群的认证文件，/etc/kubernetes/admin.conf修改为实际使用的认证文件。

```
$ mkdir -p $HOME/.kube
$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
```

**步骤2** 修改认证文件的用户与用户组。

```
$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

**步骤3** 配置当前用户的KUBECONFIG环境变量，以Ubuntu 20.04举例如下。

```
$ echo "export KUBECONFIG=$HOME/.kube/config" >> ~/.bashrc
$ source ~/.bashrc
```

----结束

# 9 FAQ

- [9.1 如何查看华为CSI日志](#)
- [9.2 启动huawei-csi服务时，服务启动异常，状态显示InvalidImageName](#)
- [9.3 创建PVC时，PVC的状态为Pending](#)
- [9.4 删除PVC前，PVC的状态为Pending](#)
- [9.5 创建Pod时，Pod的状态长时间处于ContainerCreating状态](#)
- [9.6 如何下载容器镜像到本地](#)
- [9.7 如何获取CSI版本信息](#)

## 9.1 如何查看华为 CSI 日志

### 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 --all-namespaces -o wide | grep huawei
huawei-csi huawei-csi-controller-695b84b4d8-tg64l 7/7 Running 0 14s <host1-ip>
<host1-name> <none> <none>
```

**步骤2** 使用远程访问工具（以PuTTY为例），通过管理IP地址，登录Kubernetes集群的huawei-csi-controller节点

**步骤3** 执行`cd /var/log/huawei`命令，进入日志目录

```
cd /var/log/huawei
```

**步骤4** 执行以下命令，查看容器自定义输出日志

```
vi huawei-csi-controller
```

**步骤5** 执行`cd /var/log/containers`命令，进入到容器目录

```
cd /var/log/containers
```

**步骤6** 执行以下命令，查看容器标准输出日志

```
vi huawei-csi-controller- <name>_huawei-csi_huawei-csi-driver-<container-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 --all-namespaces -o wide | grep huawei
huawei-csi huawei-csi-node-g6f7z 3/3 Running 0 14s <host2-ip> <host2-
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-node节点

**步骤3** 执行`cd /var/log/huawei`命令，进入日志目录

```
cd /var/log/huawei
```

**步骤4** 执行以下命令，查看容器自定义输出日志

```
vi huawei-csi-node
```

**步骤5** 执行`cd /var/log/containers`命令，进入到容器目录

```
cd /var/log/containers
```

**步骤6** 执行以下命令，查看容器标准输出日志

```
vi huawei-csi-node- <name>_huawei-csi_huawei-csi-driver-<container-id>.log
```

----结束

## 9.2 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 --all-namespaces | grep huawei`命令查看，显示状态为InvalidImageName

```
kubectl get pod --all-namespaces | grep huawei
huawei-csi huawei-csi-controller-fd5f97768-qlldc 6/7 InvalidImageName 0 16s
huawei-csi huawei-csi-node-25txd 2/3 InvalidImageName 0 15s
```

### 根因分析

controller和node的yaml配置文件中，配置Huawei CSI的镜像版本号错误。例如：

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

## 解决措施或规避方法

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

**步骤2** 执行以下命令，修改huawei-csi-node服务的配置文件。按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 daemonset huawei-csi-node -o yaml -n=huawei-csi
```

### 说明

- 示例yaml文件中huawei-csi-driver的参数image配置项，修改huawei-csi:\*.应替换为前面制作的华为CSI镜像<名称>:<版本号>。

containers:

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

**步骤3** 执行以下命令，修改huawei-csi-controller服务的配置文件。按I或Insert进入编辑状态，修改相关参数。修改完成后，按Esc，并输入:wq!，保存修改。

```
kubectl edit deployment huawei-csi-controller -o yaml -n=huawei-csi
```

### 说明

- 示例yaml文件中huawei-csi-driver的参数image配置项，修改huawei-csi:\*.应替换为前面制作的华为CSI镜像<名称>:<版本号>。

containers:

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

**步骤4** 等待huawei-csi-node和huawei-csi-controller服务启动。

**步骤5** 执行以下命令，查看huawei csi服务是否启动。

```
kubectl get pod --all-namespaces | grep huawei
huawei-csi huawei-csi-controller-58799449cf-zvhmv 7/7 Running 0 2m29s
huawei-csi huawei-csi-node-7fxh6 3/3 Running 0 12m
```

----结束

## 9.3 创建 PVC 时，PVC 的状态为 Pending

### 现象描述

执行完成PVC的创建操作，一段时间后，PVC的状态仍然处于Pending。

### 根因分析

原因1：由于没有提前创建指定名称的StorageClass，导致Kubernetes在创建PVC时无法找到指定StorageClass名称。

原因2：由于存储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3：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4：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，请执行以下步骤。

```
Events:
 Type Reason Age From Message
 ---- -
Warning ProvisioningFailed 0s (x15 over 3m24s) persistentvolume-controller
storageclass.storage.k8s.io "mysc" not found
```

- 删除Pending状态的PVC。
- 创建缺少的StorageClass，参考[6.1.1.1 配置存储类](#)。
- 重新创建PVC。

- 如果由原因2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 (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3导致PVC处于Pending状态，执行以下步骤。

```
Events:
 Type Reason Age From Message
 ---- -
Normal Provisioning 63s (x3 over 52s) csi.huawei.com_huawei-csi-controller-b59577886-qqzm8_58533e4a-884c-4c7f-92c3-6e8a7b327515 External provisioner is provisioning volume for claim "default/mypvc"
Warning ProvisioningFailed 63s (x3 over 52s) csi.huawei.com_huawei-csi-controller-b59577886-qqzm8_58533e4a-884c-4c7f-92c3-6e8a7b327515 failed to provision volume with StorageClass "mysc": rpc error: code = Internal desc = context deadline exceeded (Client.Timeout exceeded while awaiting headers)
```

- 请先等待10分钟，参考本章节再次检查PVC详细信息
- 如果还处于Pending状态，请联系华为工程师处理。

- 如果由原因4导致PVC处于Pending状态，请联系华为工程师处理。

----结束

## 9.4 删除 PVC 前，PVC 的状态为 Pending

### 现象描述

在执行删除PVC前，PVC的状态处于Pending。

### 根因分析

原因1：由于没有提前创建指定名称的StorageClass，导致Kubernetes在创建PVC时无法找到指定StorageClass名称。

原因2：由于存储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3：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4：其他原因。

### 解决措施或规避方法

删除Pending状态下的PVC，需要根据以下不同的原因采取不同的解决措施。

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

**步骤2** 执行以下命令，查看PVC的详细信息。

```
kubectl describe pvc mypvc
```

**步骤3** 根据PVC详细信息中Events信息，执行相应操作。

- 如果由原因1导致PVC处于Pending状态，可以执行**kubectl delete pvc mypvc**命令，删除PVC。

```
Events:
 Type Reason Age From Message
 ---- -
Warning ProvisioningFailed 0s (x15 over 3m24s) persistentvolume-controller
storageclass.storage.k8s.io "mysc" not found
```

- 如果由原因2导致PVC处于Pending状态，可以执行**kubectl delete pvc mypvc**命令，删除PVC。

```
Events:
 Type Reason Age From Message
 ---- -
Normal Provisioning 63s (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3导致PVC处于Pending状态，请联系华为工程师处理。

```
Events:
 Type Reason Age From Message
 ---- -
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
```

```

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4导致PVC处于Pending状态，请联系华为工程师处理。

----结束

## 9.5 创建 Pod 时，Pod 的状态长时间处于 ContainerCreating 状态

### 现象描述

创建Pod时，Pod长时间处于ContainerCreating状态，此时查看huawei-csi-node的日志信息（详情请参考[9.1 如何查看华为CSI日志](#)），huawei-csi-node的日志中无创建Pod的日志记录，执行**kubectrl get volumeattachment**命令后，PV列无该Pod使用的PV名称。在等待较长时间后（超过十分钟），Pod正常创建，Pod状态变为Running状态。

### 根因分析

该问题是因为Kubernetes的kube-controller-manager组件服务异常导致。

### 解决措施或规避方法

请联系容器平台侧工程师解决。

## 9.6 如何下载容器镜像到本地

### 使用 containerd 下载容器镜像

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

```
ctr image pull image:tag
```

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

```
ctr image export image.tar image:tag
```

----结束

### 使用 Docker 下载容器镜像

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

```
docker pull image:tag
```

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

```
docker save image:tag -o image.tar
```

----结束

## 使用 Podman 下载容器镜像

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

```
podman pull image:tag
```

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

```
podman save image:tag -o image.tar
```

----结束

## 9.7 如何获取 CSI 版本信息

本章节指导用户如何查看CSI版本信息。

### 操作步骤

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

**步骤2** 执行以下命令，查看huawei-csi-node所在节点信息。

```
kubectl get pod --all-namespaces -owide | grep huawei-csi-node
```

| NAMESPACE  | NAME                  | READY     | STATUS  | RESTARTS | AGE   | IP          |
|------------|-----------------------|-----------|---------|----------|-------|-------------|
| NODE       | NOMINATED NODE        | READINESS | GATES   |          |       |             |
| huawei-csi | huawei-csi-node-87mss | 3/3       | Running | 0        | 6m41s | 8.44.128.33 |
| node-1     | <none>                | <none>    |         |          |       |             |
| huawei-csi | huawei-csi-node-xp8cc | 3/3       | Running | 0        | 6m41s | 8.44.128.32 |
| node-2     | <none>                | <none>    |         |          |       |             |

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

**步骤4** 执行以下命令，查看CSI版本信息。

```
cat /var/lib/kubelet/plugins/csi.huawei.com/version
unionpay
```

----结束



# 10 附录

## 10.1 制作CSI镜像

### 10.1 制作 CSI 镜像

本章节指导用户如何制作CSI镜像。

#### 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二进制文件到“image”目录下。

**步骤5** 执行以下命令，创建Dockerfile文件。

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

LABEL maintainers="The Huawei CSI Team"
LABEL description="Huawei Storage CSI Driver."

COPY huawei-csi /

ENTRYPOINT ["/huawei-csi"]

EOF
```

## 须知

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

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

```
docker build -f Dockerfile -t huawei-csi:unionpay .
```

### 说明

“unionpay”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。如果环境中已经存在相同的镜像，请使用docker image rm <image-id>。

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

```
docker image ls | grep huawei-csi
huawei-csi unionpay c8b5726118ac About a minute ago 39 MB
```

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

```
docker save huawei-csi:unionpay -o huawei-csi.tar
```

### 说明

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

----结束