

华为认证存储系列教程

HCIP-Storage

SmartMulti-Tenant特性配置

实验指导手册

版本：6.0



华为技术有限公司

版权所有 ©【2026】【华为技术有限公司】。保留一切权利。

未经华为事先书面许可，任何人不得对本资料进行复制、修改、整合、改编、翻译，也不得将本资料或其任何部分或基于本资料的衍生作品提供、出售、转让、再许可给任何他人用于华为许可以外的任何其他用途。

Copyright ©【2026】【Huawei Technologies Co., Ltd.】. All Rights Reserved.

Without prior written permission of Huawei, no one is allowed to reproduce, modify, integrate, adapt, translate this Document, or provide, sell, transfer, sublicense this Document or any part of this Document or its derivative to any third party for any other purpose otherwise permitted by Huawei.

商标声明



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>

华为认证体系介绍

华为认证是华为公司基于“平台+生态”战略，围绕“云-管-端”协同的新ICT技术架构，华为公司打造了覆盖ICT领域的认证体系，包含ICT基础设施、应用与开发两类技术认证及项目管理认证。

根据ICT从业者的学习和进阶需求，华为认证分为工程师级别、高级工程师级别和专家级别三个认证等级。

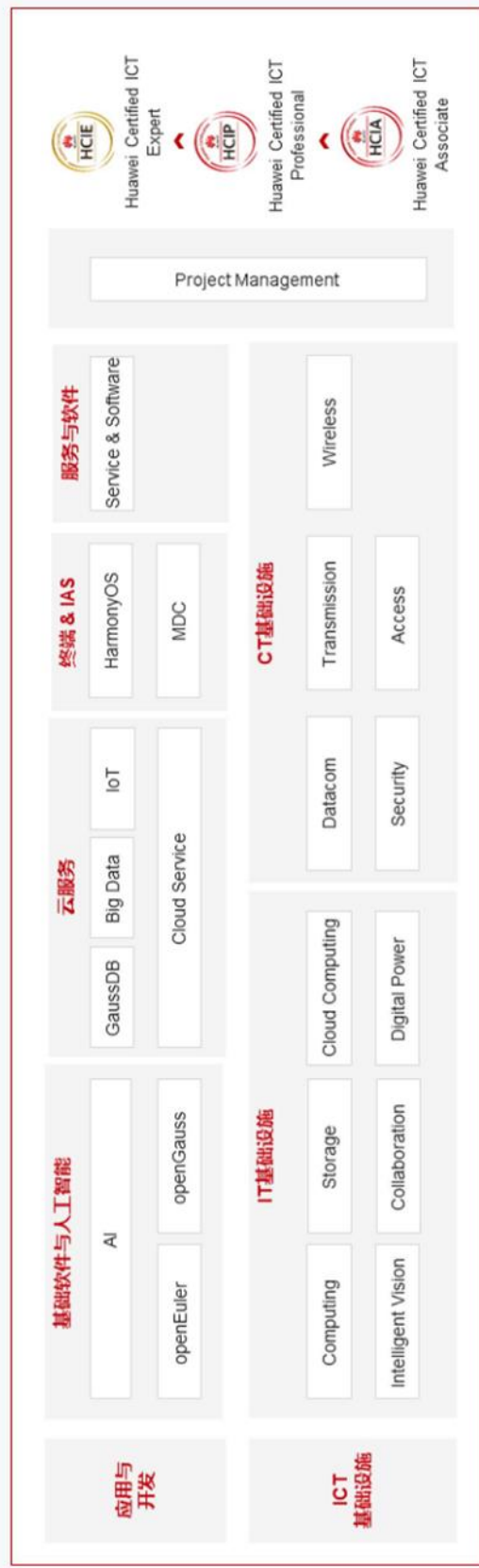
华为认证覆盖ICT全领域，符合ICT融合的技术趋势，致力于提供领先的人才培养体系和认证标准，培养数字化时代新型ICT人才，构建良性ICT人才生态。

HCIP-Storage（Huawei Certified ICT Professional-Storage，华为认证存储高级工程师）主要面向华为公司渠道工程师，高校学生，以及其他ICT从业人员等。HCIP-Storage认证在内容上涵盖闪存存储技术、数据保护技术、分布式存储技术、虚拟化技术、以及存储系统运维管理等内容。

华为认证协助您打开行业之窗，开启改变之门，屹立在存储行业的潮头浪尖！

华为职业认证全景图

华为职业认证覆盖ICT全领域，致力于提供领先的人才培养体系和认证标准，培养数字化时代的新型ICT人才，构建良性的ICT人才生态。



课程介绍

本课程将结合之前学习的内容，进行操作演练，帮助学员巩固和复习之前学习的内容，重点介绍华为闪存存储的 SmartMulti-Tenant 特性的操作演练。通过该课程的学习，能够熟悉华为闪存存储的 SmartMulti-Tenant 特性的配置操作。

内容描述

本实验指导书共包含 1 个实验，模拟同一台物理存储设备提供给多个租户使用的场景：
实验一：SmartMulti-Tenant 特性配置实验。

读者知识背景

本课程为华为认证高级课程，为了更好地掌握本书内容，阅读本书的读者应首先具备以下基本条件：

- 了解基本的存储基础知识，建议完成 HCIA-Storage 的学习，并通过了 HCIA-Storage 认证考试。
 - 熟悉 Linux 操作系统操作，了解网络基础知识。
-

目录

前 言	3
课程介绍	3
内容描述	3
读者知识背景	3
1 参考资料及工具	5
1.1 参考资料	5
1.2 软件工具	5
2 SmartMulti-Tenant 特性配置实验	6
2.1 实验介绍	6
2.1.1 关于本实验	6
2.1.2 实验目的	6
2.1.3 实验组网拓扑	6
2.2 系统用户操作	8
2.2.1 检查 License 文件	8
2.2.2 创建存储池	9
2.2.3 创建租户	10
2.2.4 创建租户用户	18
2.3 租户用户操作（块业务）	22
2.3.1 创建 LUN	22
2.3.2 创建主机	24
2.3.3 建立映射关系	24
2.3.4 验证主机连通性	25
2.3.5 验证业务隔离性	29
2.4 租户用户操作（文件业务）	30
2.4.1 创建文件系统	30
2.4.2 共享文件系统	32
2.4.3 验证业务隔离性	35
2.5 思考题	36

1 参考资料及工具

1.1 参考资料

文档中所列出的命令、文档及文档路径仅供参考，请根据实际环境中的不同产品版本使用对应的命令、文档及文档路径。

产品文档



说明：

您可以直接登录华为技术有限公司技术支持网（<http://support.huawei.com/enterprise/>），在搜索栏中，输入“文档或工具名称”，检索、浏览和下载对应文档或工具。

1.2 软件工具

1. PuTTY



说明：

使用开源软件 PuTTY 进行终端登录操作，您可以通过 PuTTY 常用域名（putty.org）浏览和下载对应文档或工具。

2. WinSCP



说明：

使用 WinSCP 在 Windows 系统和 Linux 系统间传输文件，您可以通过 WinSCP 常用域名（winscp.net）浏览和下载对应文档或工具。

3. 华为 OceanStor UltraPath



说明：

您可以登录华为技术有限公司技术支持网站（<http://support.huawei.com/enterprise/>），在搜索栏中，输入“UltraPath”，检索、浏览和下载对应文档或工具。

2 SmartMulti-Tenant 特性配置实验

2.1 实验介绍

2.1.1 关于本实验

本章节实验模拟某公司使用华为闪存存储，向公司的两个部门提供存储服务，包括块存储服务和 NAS 存储服务。

该公司使用 SmartMulti-Tenant 特性，将两个部门分别配置为租户 Tenant-A 和 Tenant-B，从而使两个部门可以相互独立的使用存储服务。

2.1.2 实验目的

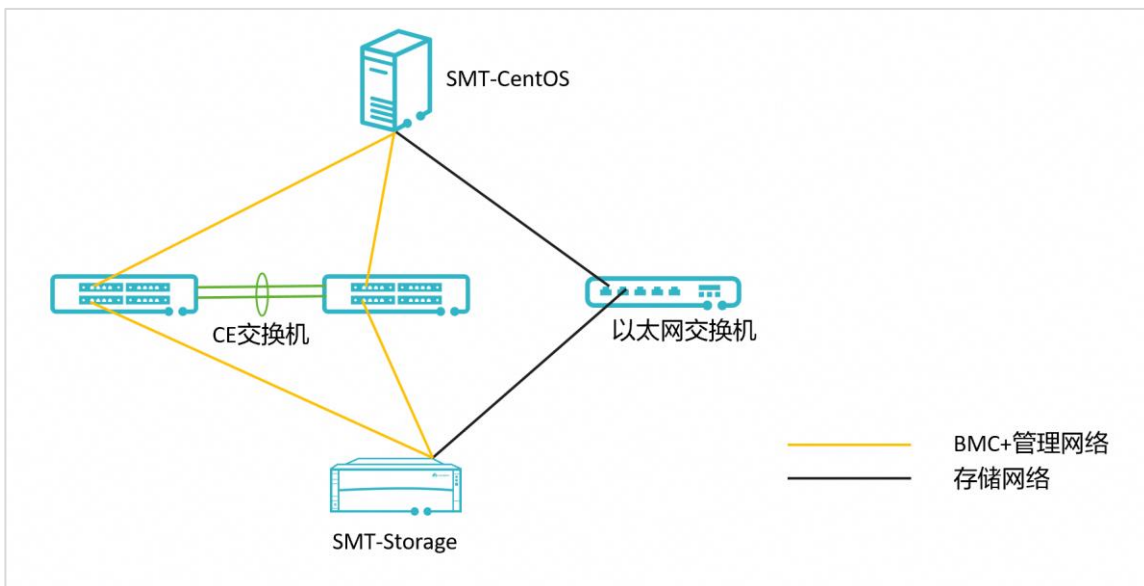
- 理解 SmartMulti-Tenant 特性的与原理和用途；
- 掌握块业务 SmartMulti-Tenant 特性的配置操作；
- 掌握文件业务 SmartMulti-Tenant 特性的配置操作。

2.1.3 实验组网拓扑

- 实验设备

设备类型	数量	软件版本
Linux主机	1	CentOS 7.6
华为闪存存储	1	V700R001, UltraPath_31.2.0
万兆光纤交换机	2	\
千兆BMC交换机	1	\

- 组网拓扑
-



实验网络信息

本实验的网络规划如下表所示：



说明：

以上信息仅供参考，实验操作前请联系管理员，根据实际环境进行访问或配置。

设备信息	管理IP	存储IP	说明
SMT-CentOS	192.168.5.151	192.168.6.151	Linux主机
SMT-Storage	192.168.5.200	192.168.6.201	块业务A
		192.168.6.202	块业务B
		192.168.6.203	文件业务A
		192.168.6.204	文件业务B

资源规划

租户	资源名称	资源容量
Tenant-A	LUN-A01	3GB
Tenant-B	FS-B01	2GB

说明： 本实验所使用的 IP 地址仅为示例，实际 IP 地址以实验环境的规划为准。

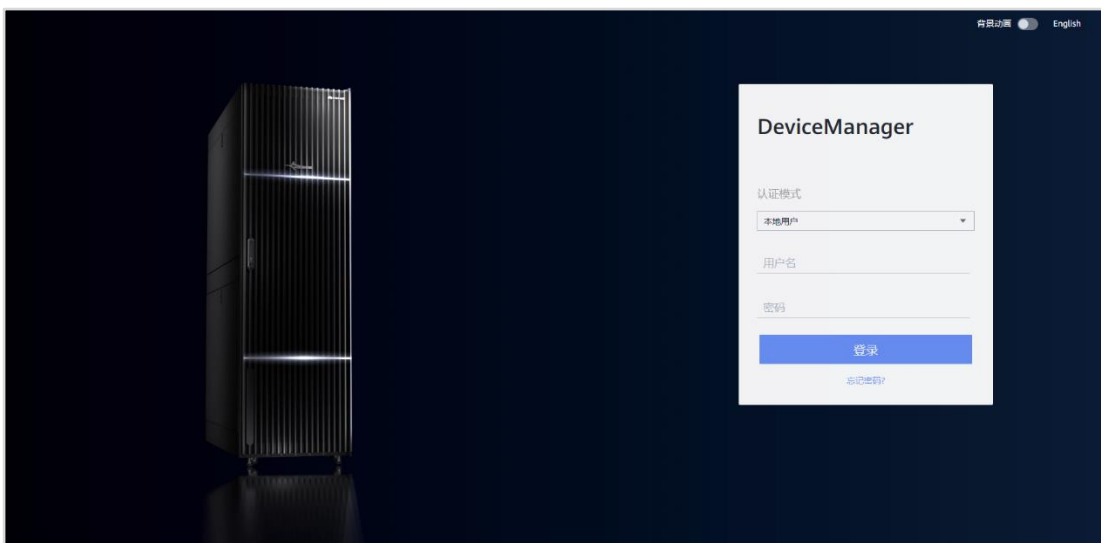
2.2 系统用户操作

2.2.1 检查 License 文件

步骤 1 登录 DeviceManager

在地址栏中输入控制框管理网口的 IP 地址 “https://XXX.XXX.XXX.XXX:8088”，并按 “Enter”。

系统进入 DeviceManager 登录界面。



在“用户名”和“密码”中输入系统用户的用户名和密码，单击“登录”，进入 DeviceManager 首页。



步骤 2 检查 License 文件

选择“设置 > License 管理”。

在中间信息展示区，确认“特性名称”列中已包含“多租户（SmartMulti-Tenant）”。

License管理

您可以查看当前设备的License信息，备份已激活的License文件，或导入并激活新的License文件。

License文件序号
License文件格式版本
创建时间

更新License
备份License

特性名称	总容量	运行截止时间
SAN双活 (HyperMetro)	无限容量	
HyperCDP	无限容量	
OceanStor OS系统软件	无限容量	
重删压缩软件基础版	999999 TB	
多租户 (SmartMulti-Tenant)	无限容量	
智能配额 (SmartQuota)	无限容量	
NAS基础特性 (NAS Foundation)	无限容量	
NAS双活 (HyperMetro)	无限容量	
WORM (HyperLock)	无限容量	

2.2.2 创建存储池

选择“系统 > 存储池”，单击“创建”。系统右侧弹出“创建存储池”对话框。配置如下参数：

名称：StoragePool001；勾选控制框 CTE0；需要磁盘数量：8；

其他参数使用默认值。点击“确定”。

创建存储池
高级

* 名称
StoragePool001

冗余策略
盘级冗余

* 控制框
☒ CTE0

存储池容量
3.461 TB

请至少选择6块盘。

单盘容量	类型	可用硬盘总数	每个控制框可选硬盘数量	需要硬盘数量
894.069 GB	SSD	8	8	8 x 1 = 8 选择

返回执行结果，点击“关闭”

执行结果 ?

创建存储池

进度

成功

任务已完成，数据更新可能存在延迟，请关闭执行结果页面后手动刷新列表。

名称 ▾	执行次数	对象	状态 ▾	持续时间	⌵
创建存储池	1/1	StoragePool001	成功	4秒	
检查硬盘域状态	1/1	StoragePool001	成功	1秒	
创建硬盘域	1/1	StoragePool001	成功	< 1秒	
任务初始化	1/1	--	成功	< 1秒	

总数: 4

<

1/1

>

关闭

说明：若存储池已创建则直接进入下个环节。

2.2.3 创建租户

创建两个租户：Tenant-A 和 Tenant-B。

Tenant-A 中创建两个逻辑端口 A1 和 A2：A1 用于块业务，A2 用于文件业务；

Tenant-B 中创建两个逻辑端口 B1 和 B2：B1 用于块业务，B2 用于文件业务。

步骤 1 创建第一个租户

选择“服务 > 租户服务 > 多租户”。单击“创建”。系统右侧弹出“创建租户”对话框。

输入名称 Tenant-A，在“逻辑端口”区域点击“增加”。

创建租户 ?

高级

基本信息

* 名称

Tenant-A

NAS容量配额 ?

GB

SAN容量配额 ?

GB

NAS容量配额告警 ?

☐

启用

SAN容量配额告警 ?

☐

启用

描述

长度范围是0到255位。

关联存储池 ?

☒

不限制

☐

自定义

关联FC端口 ?

☒

不限制

☐

自定义

网络设置

关联Netspace ?

System-defined

创建

逻辑端口

增加

移除

名称	数据协议	IP地址	主端口/当前端口	操作
Q 47				

进入“创建逻辑端口”界面，填入如下参数。

名称：A1

角色：管理+数据

数据协议：iSCSI

IP：192.168.6.201

子网掩码：255.255.255.0

网关：192.168.6.1

端口类型：以太网端口

主端口：CTE0.A.P2

点击“确定”。

创建逻辑端口 ?

* 名称

A1

* 角色 ?

管理+数据

* 数据协议 ?

☐ NFS

☐ CIFS

☐ NFS + CIFS

☒ iSCSI

☐ NVMe over RoCE

☐ NVMe over TCP

☐ NFS over RDMA

☐ OBJECT

所属租户

Tenant-A

* IP地址类型

☒ IPv4

☐ IPv6

* IP

192.168.6.201

* 子网掩码

255.255.255.0

网关

192.168.6.1

* 端口类型

以太网端口

* 主端口

CTE0.A.P2 (已连接)

确定

取消

返回“创建租户”界面，点击“增加”创建第二个逻辑端口。

进入“创建逻辑端口”界面，填入如下参数。

名称：A2

角色：数据

数据协议：NFS

IP：192.168.6.202

子网掩码：255.255.255.0

网关：192.168.6.1

端口类型：以太网端口

主端口：CTE0.A.P3

激活状态：激活

点击“确定”。

创建逻辑端口 ?

☐ 高级

* 名称

A2

* 角色 ?

数据

* 数据协议 ?

☒ NFS ☐ CIFS ☐ NFS + CIFS ☐ iSCSI ☐ NVMe over RoCE
☐ NVMe over TCP ☐ NFS over RDMA ☐ OBJECT

所属租户

Tenant-A

* IP地址类型

☒ IPv4 ☐ IPv6

* IP

192.168.6.202

* 子网掩码

255.255.255.0

网关

192.168.6.1

* 端口类型

以太网端口

* 主端口

CTE0.A.P3 (已连接)

* 激活状态

☒ 激活

确定

取消

返回“创建租户”界面，点击“确定”。

返回执行结果，点击“关闭”

创建租户 ?

执行结果

总数 3 成功 3 失败 0

操作 Q	状态 ▼	详情
创建租户 Tenant-A	成功	--
创建逻辑端口 A1	成功	--
创建逻辑端口 A2	成功	--

总数: 3

< 1/1 >

快捷操作

配置保护

创建远程复制租户Pair

创建双活租户Pair

配置域

关闭

步骤 2 创建第二个租户

选择“服务 > 租户服务 > 多租户”。单击“创建”。系统右侧弹出“创建租户”对话框。输入名称 Tenant-B，在“逻辑端口”区域点击“增加”。

创建租户

高级

基本信息

名称

Tenant-B

NAS容量配额

GB

SAN容量配额

GB

NAS容量配额告警

☐

启用

SAN容量配额告警

☐

启用

描述

长度范围是0到255位。

关联存储池

☒ 不限制

☐ 自定义

关联FC端口

☒ 不限制

☐ 自定义

网络设置

关联Netspace

System-defined

创建

逻辑端口

增加

移除

名称	数据协议	IP地址	主端口/当前端口	操作

进入“创建逻辑端口”界面，填入如下参数。

名称：B1

角色：管理+数据

数据协议：iSCSI

IP：192.168.6.203

子网掩码：255.255.255.0

网关：192.168.6.1

端口类型：以太网端口

主端口：CTE0.B.P2

点击“确定”。

创建逻辑端口 ?

* 名称

B1

* 角色 ?

管理+数据

* 数据协议 ?

☐ NFS
☐ CIFS
☐ NFS + CIFS
☒ iSCSI
☐ NVMe over RoCE
☐ NVMe over TCP
☐ NFS over RDMA
☐ OBJECT

所属租户

Tenant-B

* IP地址类型

☒ IPv4
☐ IPv6

* IP

192.168.6.203

* 子网掩码

255.255.255.0

网关

192.168.6.1

* 端口类型

以太网端口

* 主端口

CTE0.B.P2 (已连接)

返回“创建租户”界面，点击“增加”创建第二个逻辑端口。

进入“创建逻辑端口”界面，填入如下参数。

名称：B2

角色：数据

数据协议：NFS

IP：192.168.6.204

子网掩码：255.255.255.0

网关：192.168.6.1

端口类型：以太网端口

主端口：CTE0.B.P3

激活状态：激活

点击“确定”。

创建逻辑端口 ?

高级

* 名称

B2

* 角色 ?

数据

* 数据协议 ?

☒ NFS
☐ CIFS
☐ NFS + CIFS
☐ iSCSI
☐ NVMe over RoCE
☐ NVMe over TCP
☐ NFS over RDMA
☐ OBJECT

所属租户

Tenant-B

* IP地址类型

☒ IPv4
☐ IPv6

* IP

192.168.6.204

* 子网掩码

255.255.255.0

网关

192.168.6.1

* 端口类型

以太网端口

* 主端口

CTE0.B.P3 (已连接)

* 激活状态

☒ 激活

返回“创建租户”界面，点击“确定”。

创建租户 ?

高级

NAS容量配额 ?

GB

SAN容量配额 ?

GB

NAS容量配额告警 ?

☐ 启用

SAN容量配额告警 ?

☐ 启用

描述

长度范围是0到255位。

关联存储池 ?

☒ 不限制
☐ 自定义

关联FC端口 ?

☒ 不限制
☐ 自定义

网络设置

关联Netspace ?

System-defined

创建

逻辑端口

增加

移除

☐	名称 ?	数据协议	IP地址	主端口/当前端口	操作
☐	B1	iSCSI	192.168.6.203	CTE0.B.P2 (...)	移除
☐	B2	NFS	192.168.6.204	CTE0.B.P3 (...)	移除

确定

取消

返回执行结果，点击“关闭”

创建租户

执行结果

总数 3 成功 3 失败 0

操作	状态	详情
创建租户 Tenant-B	成功	--
创建逻辑端口 B1	成功	--
创建逻辑端口 B2	成功	--

总数: 3

1/1

快捷操作

配置保护

创建远程复制租户Pair

创建双活租户Pair

配置域

关闭

2.2.4 创建租户用户

步骤 1 为第一个租户创建用户

选择“服务 > 租户服务 > 多租户”。单击第一个租户的名称 Tenant-A，系统右侧弹出详情页面，在“用户管理”页签中单击“创建”。



系统右侧弹出“创建用户”页面。按如下信息配置：

类型：本地用户

用户名：user_a1

密码：自定义

角色：租户管理员

其他保持默认，点击“确定”。

创建用户

* 类型

本地用户

* 用户名

user_a1

* 密码

* 确认密码

* 角色

租户管理员

密码永不过期

描述

长度范围是0到255位。

* 登录方式

☒ RESTful
 ☒ DeviceManager

* 登录认证

☒ 登录密码
 ☐ 登录密码+邮件动态口令

确定

取消

弹出执行结果，点击“关闭”。



步骤 2 为第二个租户创建用户

选择“服务 > 租户服务 > 多租户”。单击第二个租户的名称 Tenant-B，系统右侧弹出详情页面，在“用户管理”页签中单击“创建”。



系统右侧弹出“创建用户”页面。按如下信息配置：

类型：本地用户

用户名：user_b1

密码：自定义

角色：租户管理员

其他保持默认，点击“确定”。

创建用户 ?

* 类型

本地用户 ▼

* 用户名

user_b1

* 密码

.....

* 确认密码

.....

* 角色

租户管理员 ▼

密码永不过期

☒

描述

长度范围是0到255位。

* 登录方式

☒ RESTful ☒ DeviceManager

* 登录认证

☒ 登录密码
☐ 登录密码+邮件动态口令

确定

取消

弹出执行结果，点击“关闭”。



2.3 租户用户操作（块业务）

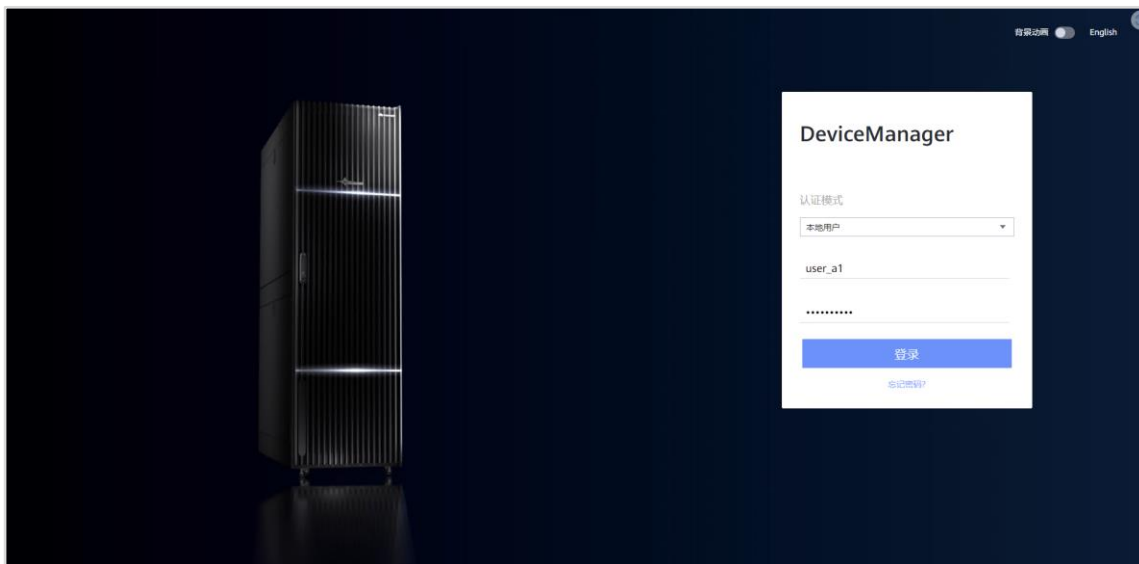
2.3.1 创建 LUN

步骤 1 登录 DeviceManager

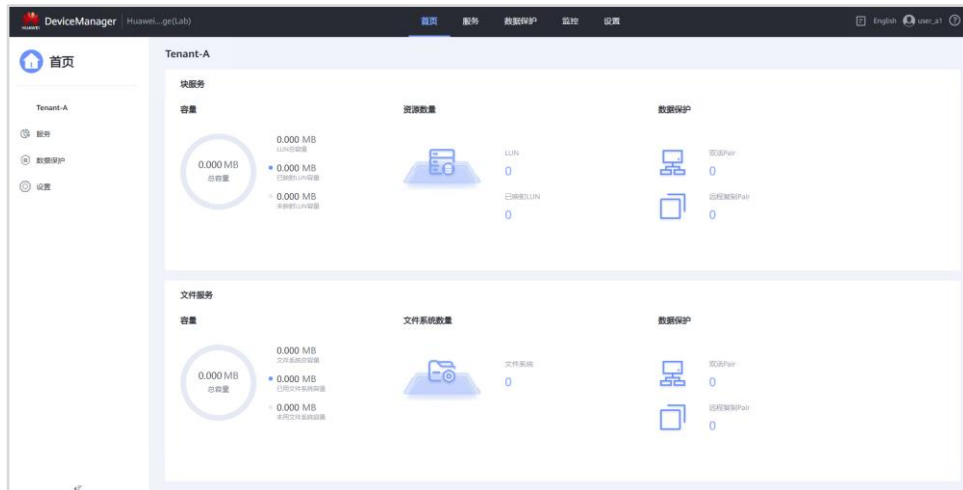
以租户 Tenant-A 的用户 user_a1 登录 DeviceManager。

在浏览器输入租户 Tenant-A 的管理逻辑端口 IP 和端口号 8088，如

“<https://192.168.6.201:8088>”，DeviceManager 的登录界面，输入租户 Tenant-A 的用户 user_a1 的用户名和密码，点击“登录”。



初始化密码后进入到首页，如下图。



步骤 2 创建 LUN

选择“服务 > 块服务 > LUN 组 > LUN”，单击“创建”。

配置如下参数：

名称：LUN-A01

容量：3GB

数量：1

其他保持默认，点击“确定”。

创建LUN

高级

名称

LUN-A01

所属租户

Tenant-A

所属存储池

StoragePool001

容量

3

GB

数量

1

(1 ~ 500)

应用类型

Default

类型：内置

应用请求大小：8 KB

增加到LUN组

请选择

映射到主机

请选择

创建

确定

取消

2.3.2 创建主机

选择“服务 > 块服务 > 主机组 > 主机”，在“主机”页签，选择“创建 > 创建主机”。



右侧弹出“创建主机”对话框，输入如下参数：

名称：Host-A01

操作系统：Linux

点击“确定”。

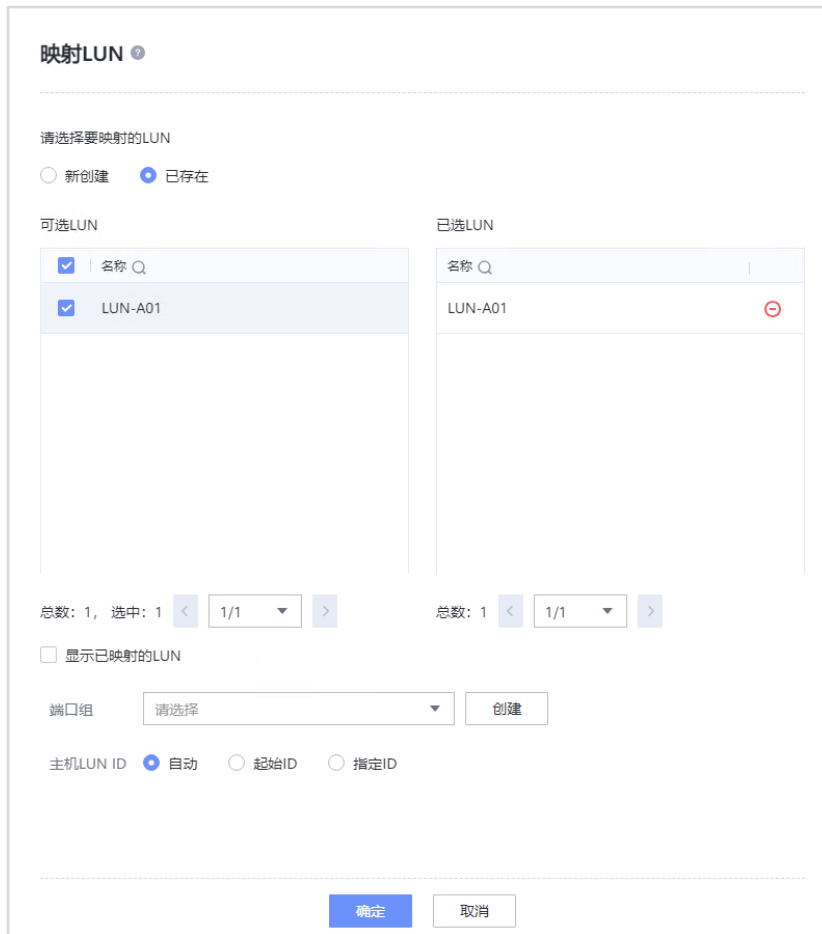


2.3.3 建立映射关系

选择“服务 > 块服务 > 主机组 > 主机”，勾选主机，单击“映射”，选择“映射到 LUN”。



弹出“映射 LUN”对话框，选择“已存在”，勾选 LUN，点击“确定”。



2.3.4 验证主机连通性

步骤 1 主机建立 iSCSI 连接

以 root 用户登录 Linux 业务主机，检查主机 IP 配置。主机需要有两个租户管理逻辑端口和数据逻辑端口同网段的 IP 地址，即需要有“192.168.6.X”网段的 IP 地址。

```
[root@smt-centos ~]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.5.151 netmask 255.255.255.0 broadcast 192.168.5.255
    inet6 fe80::2a6e:d4ff:fe88:ec12 prefixlen 64 scopeid 0x20<link>
```

```
ether 28:6e:d4:88:ec:12 txqueuelen 1000 (Ethernet)
RX packets 597405 bytes 36051370 (34.3 MiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 8822 bytes 1233112 (1.1 MiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.6.151 netmask 255.255.255.0 broadcast 192.168.6.255
inet6 fe80::2a6e:d4ff:fe88:e746 prefixlen 64 scopeid 0x20<link>
ether 28:6e:d4:88:e7:46 txqueuelen 1000 (Ethernet)
RX packets 8088206 bytes 412542204 (393.4 MiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 1246710 bytes 249536356 (237.9 MiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

检查 iSCSI 软件包：

```
[root@smt-centos ~]# rpm -qa | grep iscsi
iscsi-initiator-utils-iscsiuio-6.2.0.874-22.el7_9.x86_64
iscsi-initiator-utils-6.2.0.874-22.el7_9.x86_64
```



说明：

如果操作系统上没有安装 iSCSI 软件，或者是其版本过低，则需安装 iSCSI 软件。

为使启动器名称便于识别，推荐修改启动器名称，保证启动器名称不与其他的启动器名称重名，建议后缀使用主机名替换。

```
[root@smt-centos ~]# vi /etc/iscsi/initiatorname.iscsi
InitiatorName=iqn.1994-05.com.redhat:smt-centos
```

修改启动器名称后重启 iscsid。

```
[root@smt-centos ~]# systemctl restart iscsid
```

按照阵列端配置的租户 Tenant-A 的逻辑端口 A1 的 IP 地址，查找目标器：

```
[root@smt-centos ~]# iscsiadm -m discovery -t st -p 192.168.6.201
192.168.6.201:3260,8195 iqn.2006-08.com.huawei:oceanstor:2100fc73fb33909f::22002:192.168.6.201
```

登录 Tenant-A 的目标器：

```
[root@smt-centos ~]# iscsiadm -m node -p 192.168.6.201 -l
Logging in to [iface: default, target: iqn.2006-08.com.huawei:oceanstor:2100fc73fb33909f::22002:192.168.6.201, portal: 192.168.6.201,3260] (multiple)
Login to [iface: default, target: iqn.2006-08.com.huawei:oceanstor:2100fc73fb33909f::22002:192.168.6.201, portal: 192.168.6.201,3260] successful.
```

主机系统上修改 iscsi 服务为主机启动后自动开启：

```
systemctl enable iscsi.service
```

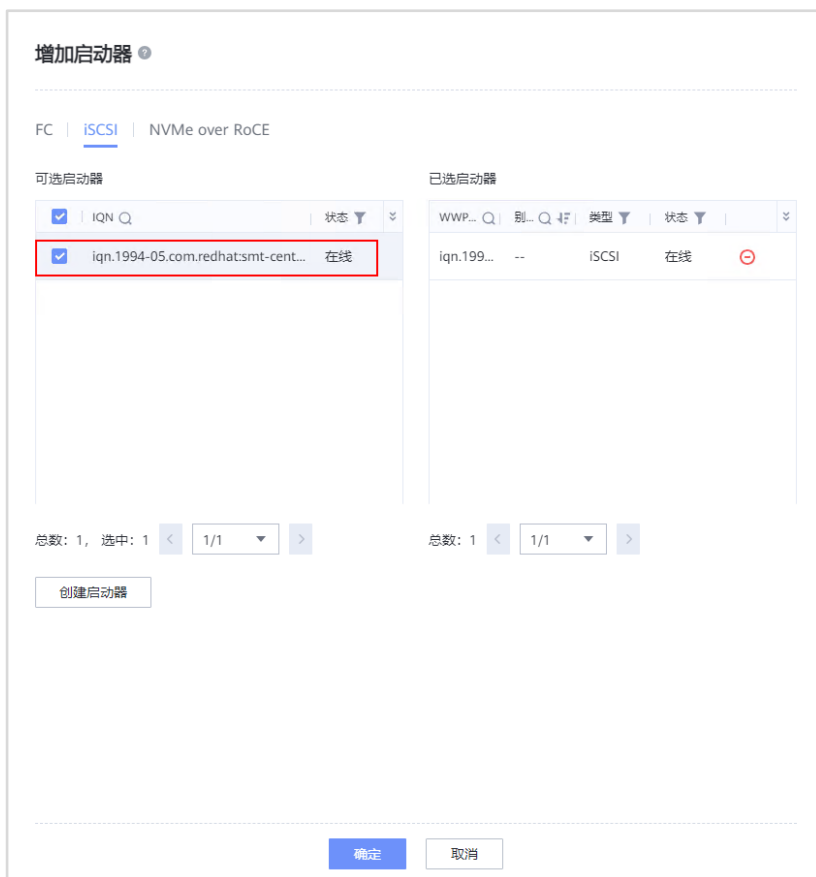
步骤 2 在存储上为主机添加启动器

以租户 Tenant-A 的用户 user_a1 登录 DeviceManager。

选择“服务 > 主机组 > 主机”，选择要添加启动器的主机，点击右侧的“更多”，并选择“增加启动器”。



选择 iSCSI 标签页，勾选对应启动器，单击“确定”。



确认警告，点击“确定”。

增加启动器



危险

您即将将启动器添加到主机 (Host-A01)。

如果逻辑主机或业务主机未进行正确配置, 可能导致业务异常。

建议:

1. 请确认您选择的启动器是否属于该主机。
2. 请确认逻辑主机的操作系统类型与业务主机操作系统类型是否一致。
3. 逻辑主机访问模式与业务主机的多路径访问模式一致。
4. 请确认双活LUN对应的逻辑主机与双活两端阵列上的访问模式配置保持一致。
5. 如果使用操作系统自带多路径, 详细信息请参考对应产品或版本的《主机连通性指南》。

类型	WWPN/iQN/NQN	别名
iSCSI	iqn.1994-05.com.redhat:smt-centos	--

总数: 1

<

1

>

☒ 我已阅读上述信息, 了解执行此操作带来的后果。

确定

取消

步骤 3 主机扫描 LUN

在主机上查看磁盘设备。

```
[root@smt-centos ~]# fdisk -l

Disk /dev/vda: 42.9 GB, 42949672960 bytes, 83886080 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x000a7317

   Device Boot      Start         End      Blocks    Id  System
/dev/vda1   *        2048     83886079     41942016   83   Linux
```

主机扫描 LUN。

```
[root@smt-centos ~]# iscsiadm -m session --rescan
Rescanning session [sid: 1, target: iqn.2006-08.com.huawei:oceastor:2100fc73fb33909f::22002:192.168.6.201, portal: 192.168.6.201,3260]
```

再次查看磁盘设备。发现大小为 3GB 的磁盘/dev/sda。

```
[root@smt-centos ~]# fdisk -l

Disk /dev/vda: 42.9 GB, 42949672960 bytes, 83886080 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
```

```
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x000a7317

    Device Boot      Start         End      Blocks   Id  System
/dev/vda1   *        2048     83886079     41942016   83   Linux

Disk /dev/sda: 3221 MB, 3221225472 bytes, 6291456 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

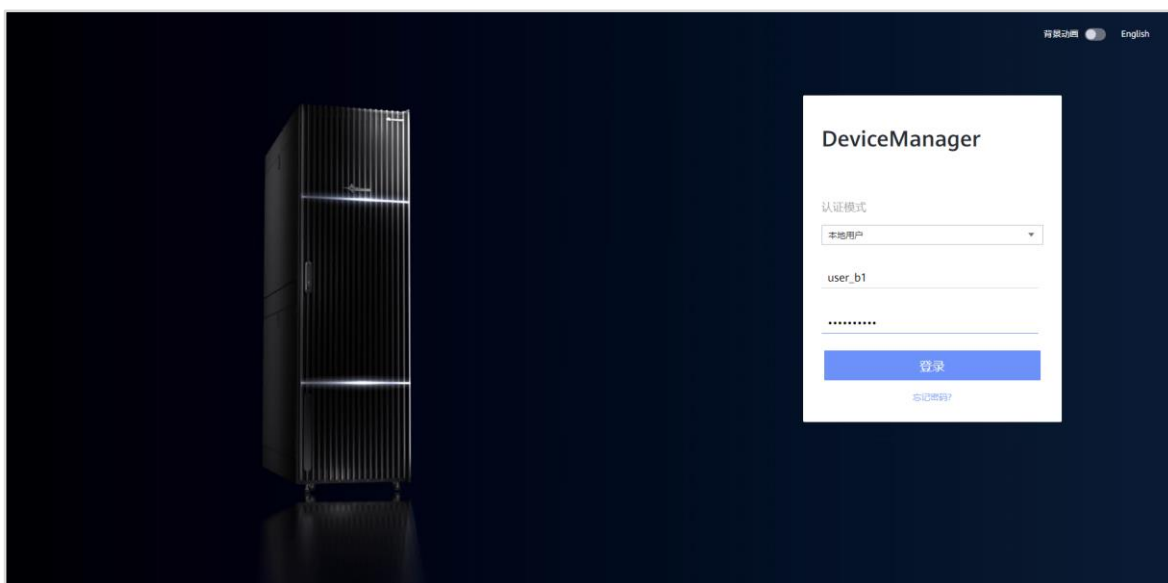
2.3.5 验证业务隔离性

步骤 1 登录 DeviceManager

以租户 Tenant-B 的用户 user_b1 登录 DeviceManager。

在浏览器输入租户 Tenant-B 的管理逻辑端口 IP 和端口号 8088，如

“<https://192.168.6.203:8088>”，DeviceManager 的登录界面，输入租户 Tenant-B 用户 user_b1 的用户名和密码，点击“登录”。



步骤 2 验证 LUN 隔离性

选择“服务 > 块服务 > LUN 组 > LUN”，租户 Tenant-A 的用户 user_a1 所创建的 3GB 的 LUN 不可见。



步骤 3 验证主机隔离性

选择“服务 > 块服务 > 主机组 > 主机”，租户 Tenant-A 的用户 user_a1 所创建的主机不可见。



2.4 租户用户操作（文件业务）

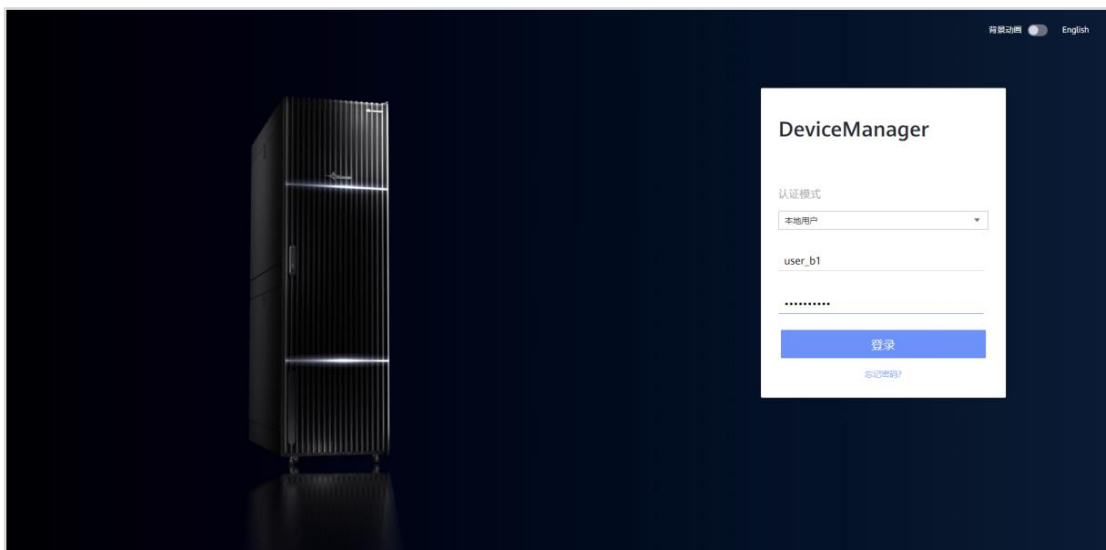
2.4.1 创建文件系统

步骤 1 登录 DeviceManager

以租户 Tenant-B 的用户 user_b1 登录 DeviceManager。

在浏览器输入租户 Tenant-B 的管理逻辑端口 IP 和端口号 8088，如

“<https://192.168.6.203:8088>”，DeviceManager 的登录界面，输入租户 Tenant-B 用户 user_b 的用户名和密码，点击“登录”。



步骤 2 创建文件系统

选择“服务 > 文件服务 > 文件系统”，单击“创建”。弹出“建文件系统”，配置如下参数：

名称：FS-B01

容量：2GB

NFS：关闭

CIFS：关闭

加入定时 HyperCDP 计划：关闭

其他参数使用默认值。点击“确定”。

创建文件系统

高级

基本信息

名称

FS-B01

所属租户

Tenant-B

所属存储池

StoragePool001

安全模式

UNIX

VAAI

启用

容量与调优

容量

2

GB

快照预留空间比例(%)

20

(0 ~ 50)

应用类型

NAS_Default

类型: 内置
 应用请求大小: 16 KB
 文件系统语义层分布算法: 目录均衡模式

共享

NFS

确定

取消

确认警告，点击“确定”。

创建文件系统

×

警告

您即将执行创建文件系统的操作。当文件系统容量不大于3 GB时，该文件系统对应的增值业务和主机业务可能超时，以及切换文件系统的工作控制器也可能超时。

建议：文件系统容量最好大于3 GB。

我已阅读上述信息，了解执行此操作带来的后果。

确定

取消

2.4.2 共享文件系统

步骤 1 配置 NFS 共享

选择“服务 > 文件服务 > 共享 > NFS 共享”，单击“创建”，系统右侧弹出“创建 NFS 共享”页面。

选择文件系统 FS-B01，点击“确定”。

创建NFS共享 ?

高级

基本信息

所属租户

Tenant-B

* 文件系统

FS-B01

创建

Dtree

请选择

共享名称

目录 ?

共享路径 ?

/FS-B01/

权限

只有加入权限列表的客户端才允许访问NFS共享。

增加

移除

刷新

步骤 2 增加 NFS 共享客户端

选择“服务 > 文件服务 > 共享 > NFS 共享”。单击 NFS 共享所在行右侧的“更多”，选择“增加客户端”。

共享 ?

NFS共享

CIFS共享

创建

模板创建

删除

共享名称 ?

共享路径 ?

操作

✓ /FS-B01	/FS-B01/	更多
--	----------------------------------	---------------

模板创建

增加客户端

增加文件扩展名过滤规则

修改

删除

系统右侧弹出“增加客户端”页面。配置如下参数：

类型：主机

客户端：192.168.6.151

UNIX 权限：读写

root 权限限制：no_root_squash

其他参数保持默认值，点击“确定”。

增加客户端
高级

* 类型
主机

如果增加主机名称，请至少配置LDAP域、NIS域和DNS中的一个。如果增加客户端IP，则不需要配置。

* 客户端
192.168.6.151 X

* UNIX权限
读写

* Kerberos5权限
无

* Kerberos5i权限
无

* Kerberos5p权限
无

* root权限限制
root_squash
no_root_squash

弹出执行结果，点击“关闭”。

执行结果

增加NFS共享权限

进度
成功

任务已完成，数据更新可能存在延迟，请关闭执行结果页面后手动刷新列表。

名称	执行次数	对象	状态	持续时间
增加NFS共享权限	1/1	--	成功	1秒

总数: 1
1/1

步骤 3 访问 NFS 共享

使用 root 用户登录到客户端。

运行 showmount -e ipaddress 命令，查看存储系统当前所有的 NFS 共享。其中 ipaddress 是 Tenant-B 的逻辑端口 B2 的 IP 地址。

```
[root@smt-centos ~]# showmount -e 192.168.6.204
Export list for 192.168.6.204:
/Fs-B01 192.168.6.151
```

客户端可以通过 Tenant-B 的逻辑端口 B2 访问到 NFS 共享。

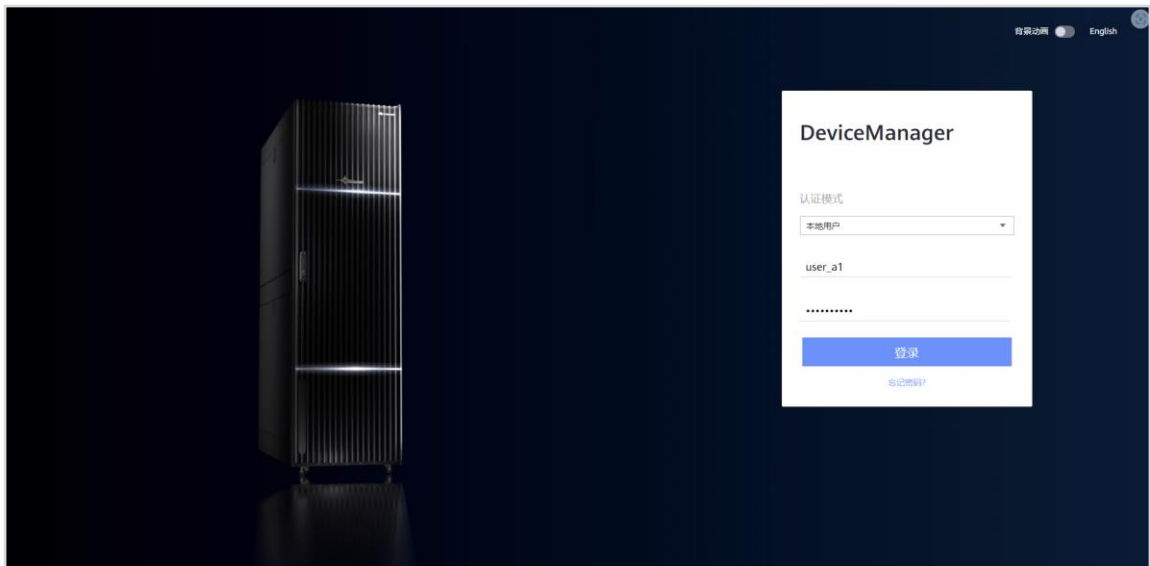
2.4.3 验证业务隔离性

步骤 1 登录 DeviceManager

以租户 Tenant-A 的用户 user_a1 登录 DeviceManager。

在浏览器输入租户 Tenant-A 的管理逻辑端口 IP 和端口号 8088，如

“<https://192.168.6.201:8088>”，DeviceManager 的登录界面，输入租户 Tenant-A 的用户 user_a1 的用户名和密码，点击“登录”。



步骤 2 验证文件系统隔离性

选择“服务 > 文件服务 > 文件系统”，租户 Tenant-B 的用户 user_b1 创建的文件系统 FS-B01 不可见。



步骤 3 验证 NFS 共享隔离性

使用 root 用户登录到客户端。

运行 showmount -e ipaddress 命令，查看存储系统当前所有的 NFS 共享。其中 ipaddress 是 Tenant-A 的逻辑端口 A2 的 IP 地址。

```
[root@smt-centos ~]# showmount -e 192.168.6.202
Export list for 192.168.6.202:
```

Tenant-B 的 NFS 共享/FS-B01 不可见。

2.5 思考题

问题：

在文件业务场景中，多租户的业务隔离是如何实现的？

参考答案：

多租户的业务隔离，主要体现为用户的业务数据、业务访问和业务配置（NAS 协议配置等）隔离。

业务数据隔离

每个租户有自己独立的文件系统，以此达到租户文件系统的隔离。同理基于文件系统的配额等业务也是隔离的。

业务访问隔离

每个租户都具有独立的 NAS 协议实例，包括 NFS 服务、NDMP 服务。

业务配置隔离

每个租户可以有自己独立的用户、用户组、用户映射规则、安全策略、NFS 共享、AD 域、DNS 服务、LDAP 服务、NIS 服务等。

华为认证存储系列教程

HCIP-Storage

远程复制特性配置

实验指导手册

版本：6.0



华为技术有限公司

版权所有 ©【2026】【华为技术有限公司】。保留一切权利。

未经华为事先书面许可，任何人不得对本资料进行复制、修改、整合、改编、翻译，也不得将本资料或其任何部分或基于本资料的衍生作品提供、出售、转让、再许可给任何他人用于华为许可以外的任何其他用途。

Copyright ©【2026】【Huawei Technologies Co., Ltd.】. All Rights Reserved.

Without prior written permission of Huawei, no one is allowed to reproduce, modify, integrate, adapt, translate this Document, or provide, sell, transfer, sublicense this Document or any part of this Document or its derivative to any third party for any other purpose otherwise permitted by Huawei.

商标声明



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>

华为认证体系介绍

华为认证是华为公司基于“平台+生态”战略，围绕“云-管-端”协同的新ICT技术架构，华为公司打造了覆盖ICT领域的认证体系，包含ICT基础设施、应用与开发两类技术认证及项目管理认证。

根据ICT从业者的学习和进阶需求，华为认证分为工程师级别、高级工程师级别和专家级别三个认证等级。

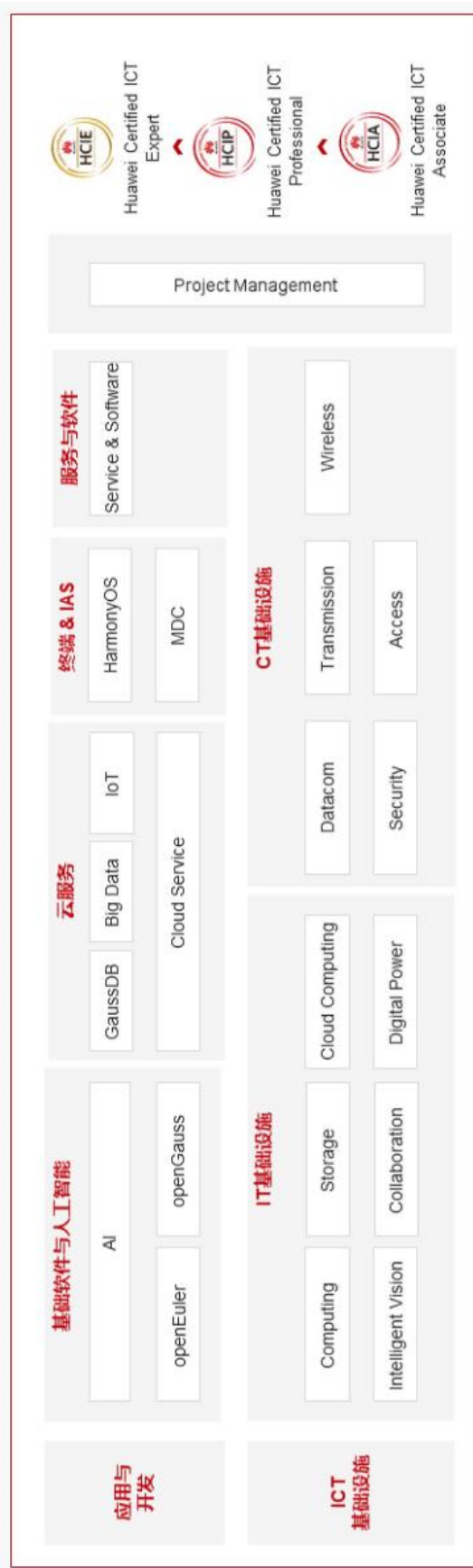
华为认证覆盖ICT全领域，符合ICT融合的技术趋势，致力于提供领先的人才培养体系和认证标准，培养数字化时代新型ICT人才，构建良性ICT人才生态。

HCIP-Storage（Huawei Certified ICT Professional-Storage，华为认证存储高级工程师）主要面向华为公司渠道工程师，高校学生，以及其他ICT从业人员等。HCIP-Storage认证在内容上涵盖闪存存储技术、数据保护技术、分布式存储技术、虚拟化技术、以及存储系统运维管理等内容。

华为认证协助您打开行业之窗，开启改变之门，屹立在存储行业的潮头浪尖！

华为职业认证全景图

华为职业认证覆盖ICT全领域，致力于提供领先的人才培养体系和认证标准，培养数字化时代的新型ICT人才，构建良性的ICT人才生态。



前言

简介

本课程将结合之前学习的内容，进行存储远程复制特性配置演练，帮助学员巩固和复习之前学习的内容。通过该课程的学习，能够熟悉存储文件系统远程复制特性配置相关流程和操作。

内容描述

本实验指导书共包含 1 个实验，模拟存储交付项目中的文件系统远程复制配置。

- 实验一：通过 HyperReplication 特性实现文件系统的远程复制。

读者知识背景

本课程为华为认证基础课程，为了更好地掌握本书内容，阅读本书的读者应首先具备以下基本条件：

- 了解基本的存储基础知识，建议完成 HCIA-Storage 的学习，并通过了 HCIA-Storage 认证考试。
 - 熟悉 Linux 操作系统操作，了解网络基础知识。
-

目录

前 言	4
简介	4
内容描述	4
读者知识背景	4
1 参考资料及工具	6
1.1 参考资料	6
1.2 软件工具	6
2 远程复制特性配置	7
2.1 实验介绍	7
2.1.1 关于本实验	7
2.1.2 实验目的	7
2.1.3 实验组网拓扑	8
2.2 基础业务配置	9
2.2.1 配置文件系统	9
2.2.2 配置 NFS 共享	15
2.3 配置远程复制	20
2.3.1 配置远程复制	20
2.3.2 配置验证	30
2.4 思考题	38

1 参考资料及工具

1.1 参考资料

文档中所列出的命令、文档及文档路径仅供参考，请根据实际环境中的不同产品版本使用对应的命令、文档及文档路径。

产品文档



说明：

您可以直接登录华为技术有限公司技术支持网（<http://support.huawei.com/enterprise/>），在搜索栏中，输入“文档或工具名称”，检索、浏览和下载对应文档或工具。

1.2 软件工具

1. PuTTY



说明：

使用开源软件 PuTTY 进行终端登录操作，您可以通过 PuTTY 常用域名（putty.org）浏览和下载对应文档或工具。

2. WinSCP



说明：

用于在 Windows 系统和 Linux 系统间传输文件。推荐使用“WinSCP”，可以根据需求选用其他类似工具。

2 远程复制特性配置

2.1 实验介绍

2.1.1 关于本实验

随着各行业数字化进程的推进，数据逐渐成为企业的运营核心，用户对承载数据的存储系统的稳定性要求也越来越高，同时支持文件系统的存储系统也在各行业中被广泛的应用。虽然企业目前拥有稳定性极高的存储设备，但还是无法防止各种自然灾害对存储系统造成不可恢复的毁坏。为了保证数据存取的持续性、可恢复性和高可用性，远程容灾解决方案应运而生，而文件系统远程复制技术则是远程容灾解决方案中的关键技术之一。

本章节以文件服务场景为例进行远程复制特性配置操作。创建租户，创建存储池，LUN 和文件系统等基础配置操作请参考《SmartMulti-Tenant 特性配置实验指导手册》。

2.1.2 实验目的

- 理解远程复制的技术原理。
 - 掌握存储远程复制组网架构。
 - 掌握存储远程复制特性的配置步骤和思路。
-

2.1.3 实验组网拓扑

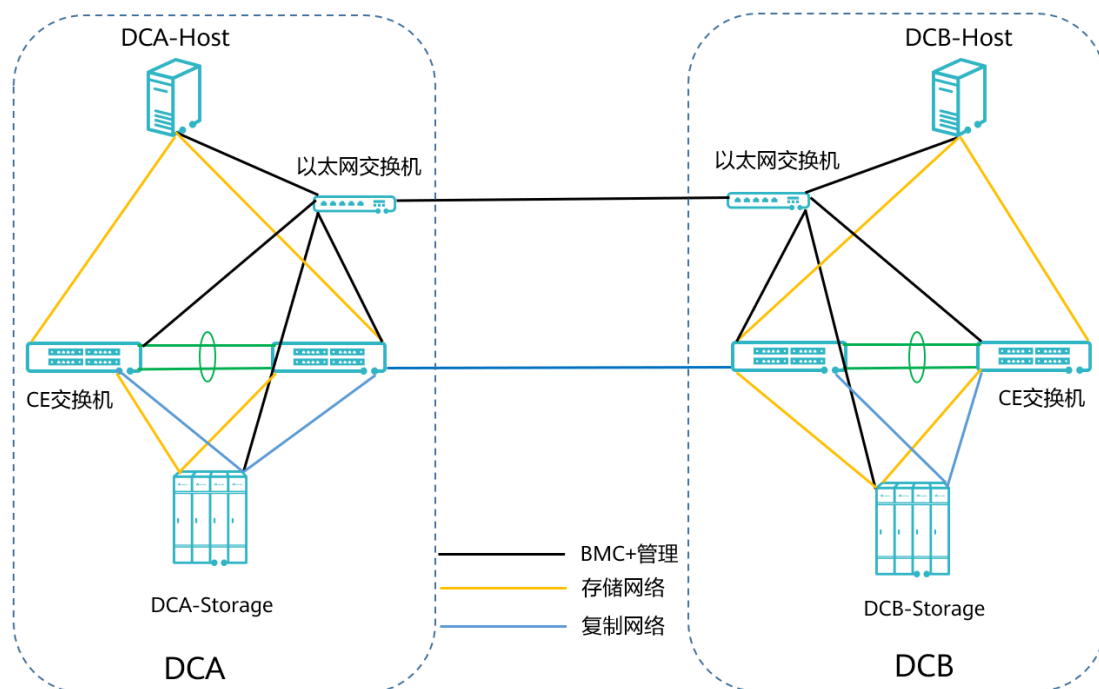


图2-1 远程复制配置实验拓扑图

上图为闪存存储主备容灾实验环境组网，其中 DCA 作为生产中心，DCB 作为灾备中心。在生产中心部署主机 DCA-Host 和存储 DCA-Storage；在灾备中心部署主机 DCB-Host 和存储 DCB-Storage。

设备配置

为了满足 HCIP-Storage 实验需要，建议每套实验环境采用以下配置：

设备类型	设备型号	数量	软件最低版本
闪存存储	OceanStor Dorado V700R001 或 OceanStor V700R001	2台	V700R001
应用服务器	X86服务器	2台	CentOS 7.6
管理交换机	以太网交换机	2台	Version 5.160
业务交换机	CE交换机	4台	Version 8.210

网络规划

本实验的网络规划如下表所示：

 说明：

以上信息仅供参考，实验操作前请联系管理员，根据实际环境进行访问或配置。

站点	设备信息	管理IP	存储IP	说明
DCA	DCA-Host	192.168.5.151	192.168.6.151	业务
	DCA-Storage	192.168.5.200	192.168.6.201	块业务A
			192.168.6.204	文件业务B
			192.168.6.205	复制1
			192.168.6.206	复制2
DCB	DCB-Host	192.168.5.152	192.168.6.152	业务
	DCB-Storage	192.168.5.204	192.168.6.209	块业务A
			192.168.6.212	文件业务B
			192.168.6.213	复制1
			192.168.6.214	复制2

2.2 基础业务配置

2.2.1 配置文件系统

步骤 1 创建存储池

为保障应用服务器能够使用存储系统的存储空间，登录 DCA 和 DCB 的存储设备，并创建存储池 StoragePool001。

选择“系统 > 存储池”，单击“创建”，设置存储池的参数。

创建存储池 ?

高级

* 名称

StoragePool001

冗余策略

盘级冗余

* 控制框

☒ CTE0

存储池容量

3.461 TB

请至少选择6块盘。

单盘容量	类型	可用硬盘总数	每个控制框可选硬盘数量 ?	需要硬盘数量 ?
894.069 GB	SSD	8	8	8 x 1 = 8 选择

单击“确定”。

步骤 2 创建文件系统

先在 DCA 的存储设备上创建一个容量为 10GB 的文件系统，名称为“FileSystem001”

选择“服务 > 文件服务 > 文件系统”。

在左上方的“租户”下拉框中选择需要创建文件系统的租户“Tenant-B”。

单击“创建”。

设置文件系统的名称为“FileSystem001”，所属存储池为“StoragePool001”。

设置文件系统的容量为“10GB”，应用类型为“NAS_Default”，暂时关闭“共享”与“保护”。

说明：如果租户“Tenant-B”已经删除，请参照《01 SmartMulti-Tenant 特性配置实验指导手册》进行创建。

创建文件系统 ?

高级

基本信息

* 名称

FileSystem001

所属租户

Tenant-B

* 所属存储池

StoragePool001

* 安全模式 ?

UNIX

当安全模式设置为UNIX、Native或Mixed时，如果用CIFS协议读写数据，建议开启用户映射开关，并配置映射关系。若未配置用户映射，Administrators组的用户数据会默认统计到root用户配额，非Administrators组的用户数据会默认统计到nobody用户配额。

VAAI ?

☐ 启用

容量与调优

* 容量 ?

10

GB

* 快照预留空间比例(%) ?

20

(0 ~ 90)

* 应用类型

NAS_Default

类型: 内置
 应用请求大小: 16 KB
 数据压缩: 已禁用
 重复数据删除: 已禁用
 文件系统语义层分布算法: 目录均衡模式

共享

确定

取消

单击“确定”。

文件系统 ?

租户 Tenant-B

创建

复制创建

保护

删除

名称	容量	健康状态	运行状态	创建时间	WORM	数据保护	共享	配额	Over个数	操作
FileSystem001	10000 GB	0.00% 正常	在线	20	15355	--			0	更多

总数: 1, 选中: 0

1

2

步骤 3 创建 Dtree

为了方便管理目录内所有文件的使用空间和该目录的访问权限，需要创建 Dtree。

选择“服务 > 文件服务 > Dtree”。

在左上方的“租户”下拉框中选择需要创建 Dtree 的文件系统所属的租户“Tenant-B”。

单击“创建”。设置 Dtree 的名称为“Dtree001”。

创建Dtree ?

☐ 高级

基本信息

* 所属文件系统

FileSystem001

所属租户

Tenant-B

* 名称

Dtree001 X

配额开关

☐ 启用

* 安全模式 ?

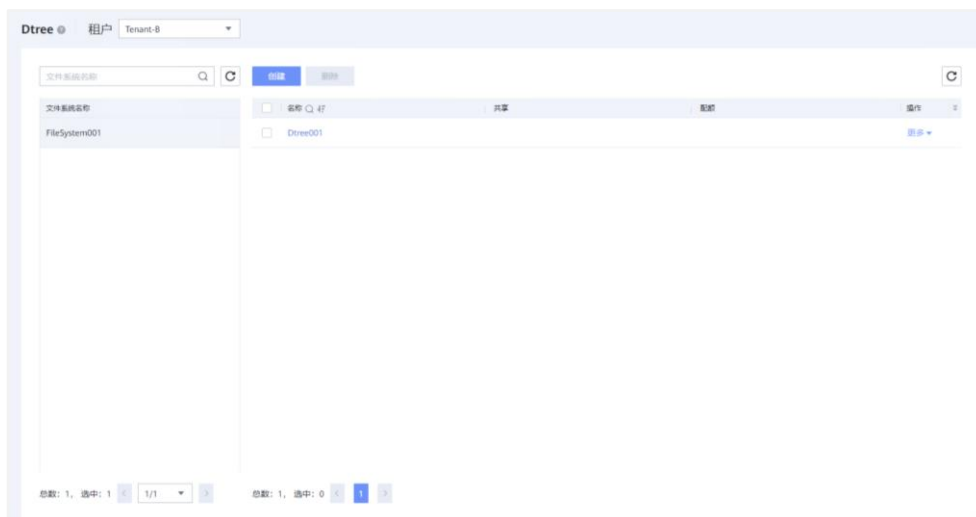
UNIX

当安全模式设置为UNIX、Native或Mixed时，如果用CIFS协议读写数据，建议开启用户映射开关，并配置映射关系。若未配置用户映射，Administrators组的用户数据会默认统计到root用户配额，非Administrators组的用户数据会默认统计到nobody用户配额。

确定

取消

单击“确定”。



步骤 4 创建配额

为了限制 Dtree 下的使用空间和文件数量，工程师需要创建配额。

选择“服务 > 文件服务 > 配额 > 自定义配额”。

在左上方的“租户”下拉框中选择需要创建配额的文件系统所属的租户“Tenant-B”。

单击“创建”，选择创建配额的文件系统“FileSystem001”和 Dtree“Dtree001”，选择创建的配额类型为“目录配额”。

设置空间配额，硬配额为 10GB，软配额为 8GB。

设置文件数配额，硬配额为 30 千，软配额为 20 千。

创建配额 ?

配额对象

所属租户

Tenant-B

* 文件系统

FileSystem001

创建

Dtree

Dtree001

配额类型

* 配额类型

☒ 目录配额

限制所选择Dtree的使用空间或文件数量。

☐ 用户配额

限制单个用户的使用空间或文件数量。

☐ 用户组配额

限制单个用户组的使用空间或文件数量。

空间配额

硬配额 ?

10

GB

(1 KB ~ 256 PB)

软配额 ?

8

GB

(1 KB ~ 256 PB)

文件数配额

硬配额 ?

30

千

(1个~20亿个文件)

软配额 ?

20

千

(1个~20亿个文件)

确定

取消

单击“确定”。

配额 ? 租户 Tenant-B

配额报表 自定义配额

文件系统名称

Q

创建

删除

文件系统名称	☐ Dtree Q	配额类型	用户	用户组	用户/用户组类型	空间配额(GB/MB)	文件数配额(GB/MB)	操作
FileSystem001	☐ Dtree001	目录配额	--	--	--	8,000 GB/10,000 GB	20000/30000	更多

总数: 1, 选中: 1

1/1

总数: 1, 选中: 0

1

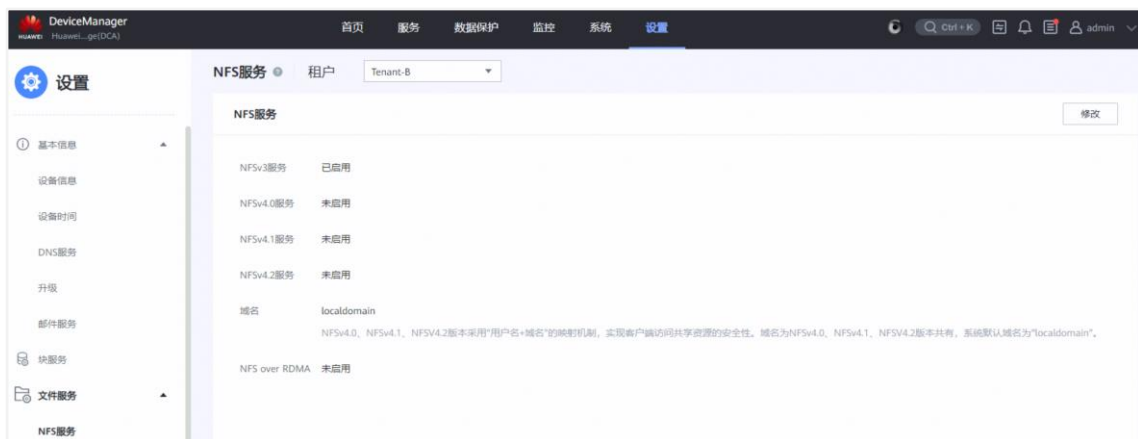
2.2.2 配置 NFS 共享

步骤 1 开启 NFSv4 服务

选择“设置 > 文件服务 > NFS 服务”。

在左上方的“租户”下拉框中选择需要开启 NFSv4 服务的租户“Tenant-B”。

单击右上角的“修改”。

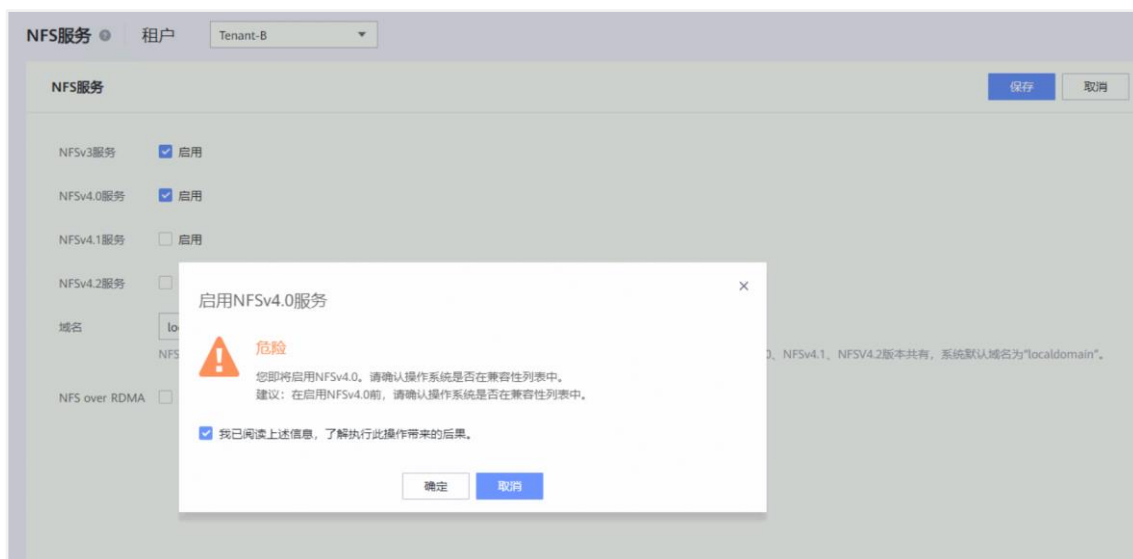


根据业务需要勾选“NFSv4.0 服务”。

单击“保存”。



仔细阅读对话框中的内容，确认无误后选择“我已阅读上述信息，了解执行此操作带来的后果”。然后单击“确定”。



步骤 2 配置网络，创建逻辑端口

选择“服务 > 网络 > 逻辑端口”。

单击“创建”。系统右侧弹出“创建逻辑端口”页面。

配置逻辑端口参数：

名称：B2

角色：数据

数据协议：NFS

IP：192.168.6.204

网关：192.168.6.1

子网掩码：255.255.255.0

端口类型：以太网端口

主端口：CTE0.B.P3

激活状态：激活

点击“确定”。

创建逻辑端口 ?

高级

* 名称

B2

* 角色 ?

数据

* 数据协议 ?

☒ NFS
☐ CIFS
☐ NFS + CIFS
☐ iSCSI
☐ NVMe over RoCE
☐ NVMe over TCP
☐ NFS over RDMA
☐ OBJECT

所属租户

Tenant-B

* IP地址类型

☒ IPv4
☐ IPv6

* IP

192.168.6.204

* 子网掩码

255.255.255.0

网关

192.168.6.1

* 端口类型

以太网端口

* 主端口

CTE0.B.P3 (已连接)

* 激活状态

☒ 激活

单击“确定”。

步骤 3 创建 NFS 共享

选择“服务 > 文件服务 > 共享 > NFS 共享”。

在左上方的“租户”下拉框中，选择需要创建共享的文件系统所属的租户“Tenant-B”。

单击“创建”。

设置 NFS 共享的基本参数，文件系统选择“FileSystem001”，Dtree 选择“Dtree001”。

创建NFS共享 ?

高级

基本信息

所属租户

Tenant-B

* 文件系统

FileSystem001

创建

Dtree

Dtree001

共享名称

目录 ?

共享路径 ?

/FileSystem001/Dtree001

权限

只有加入权限列表的客户端才允许访问NFS共享。

增加

移除

刷新

客户端

类型

UNIX权限

操作

暂无数据

确定

取消

单击“确定”。

服务

租户服务

块服务

文件服务

文件系统

Dtree

共享

共享

租户

Tenant-B

NFS共享

CIFS共享

创建

高级创建

删除

共享名称

共享路径

操作

/FileSystem001/Dtree001

/FileSystem001/Dtree001

更多

步骤 4 增加 NFS 共享客户端

选择“服务 > 文件服务 > 共享 > NFS 共享”。

在左上方的“租户”下拉框中选择需要增加客户端的 NFS 共享所属的租户“Tenant-B”。

单击 NFS 共享所在行右侧的“更多”，选择“增加客户端”。



添加客户端信息，输入 DCA-Host 的 IP 地址，例如“192.168.6.151”，UNIX 权限设为“读写”，root 权限限制设为“no_root_squash”，其它选项保持默认。



单击“确定”。

步骤 5 访问 NFS 共享

使用 root 用户登录到 DCA-Host 的客户端。

运行 showmount -e 192.168.6.204 命令，查看存储系统当前所有的 NFS 共享。

```
[root@dca-host ~]# showmount -e 192.168.6.204
Export list for 192.168.6.204:
/FileSystem001/Dtree001 192.168.6.151
```

运行“mount -t nfs 192.168.6.204:/FileSystem001/Dtree001 /mnt”命令，挂载 NFS 共享。

运行“df -h”命令，确认已挂载成功。

```
[root@dca-host ~]# mount -t nfs 192.168.6.204:/FileSystem001/Dtree001 /mnt
[root@dca-host ~]# df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
devtmpfs	1.8G	0	1.8G	0%	/dev
tmpfs	1.8G	0	1.8G	0%	/dev/shm
tmpfs	1.8G	8.6M	1.8G	1%	/run

tmpfs	1.8G	0	1.8G	0%	/sys/fs/cgroup
/dev/sda1	40G	2.7G	35G	8%	/
tmpfs	367M	0	367M	0%	/run/user/0
192.168.6.204:/FileSystem001/Dtree001	8G	0	8G	0%	/mnt

步骤 6 创建测试文件

进入/mnt 目录，并创建 a.txt 文件。

```
cd /mnt
touch a.txt
ls
```

回显信息：

```
a.txt
```

在新创建的“a.txt”文件中写入“Hello World!”。

```
cat <<EOF > a.txt
Hello World!
EOF
```

2.3 配置远程复制

2.3.1 配置远程复制

步骤 1 检查 License

选择“设置 > License 管理”。

在中间信息展示区，确保存储系统已激活的 License 包含“远程复制（HyperReplication）”和“NAS 基础特性（NAS Foundation）”。

License管理

您可以查看当前设备的License信息，备份已激活的License文件，或导入并激活新的License文件。

License文件序列号
License文件格式版本
AdaptiveLMV100R009C00SPC300

更新License
备份License

特性名称	总容量
管理软件 (Management Console)	无限容量
智能精简配置 (SmartThin)	无限容量
智能数据迁移 (SmartMigration)	无限容量
智能服务质量控制 (SmartQoS)	无限容量
快照 (HyperSnap)	无限容量
远程复制 (HyperReplication)	无限容量
克隆 (HyperClone)	无限容量
异构虚拟化 (SmartVirtualization)	无限容量
SAN双活 (HyperMetro)	无限容量
HyperCDP	无限容量
OceanStor OS系统软件	无限容量
重删压缩软件基础版	999999 TB

步骤 2 创建复制逻辑端口

分别登录本端和远端设备的 DeviceManager。

选择“服务 > 网络 > 逻辑端口”。

在左上方的“租户”下拉框中选择“所有租户”。

单击“创建”。

系统右侧弹出“创建逻辑端口”页面。

配置逻辑端口参数：

名称：REP-1；

角色：复制；

IP 地址类型：IPv4；

IP：192.168.6.205（本端 DCA）/192.168.6.213（远端 DCB）；

子网掩码：255.255.255.0；

网关：192.168.6.1；

端口类型：以太网端口；

主端口：CTE0.A.P4。

创建逻辑端口 ?

名称

REP-1

角色 ?

复制

IP地址类型

☒ IPv4
 ☐ IPv6

IP

192.168.6.205

子网掩码

255.255.255.0

网关

192.168.6.1

端口类型

以太网端口

主端口

CTE0.A.P4 (已连接)

单击“确定”。完成逻辑端口创建。

再次“创建逻辑端口”，对话框中输入相关参数：

名称：REP-2；

角色：复制；

IP 地址类型：IPv4；

IP：192.168.6.206（本端 DCA）/192.168.6.214（远端 DCB）；

子网掩码：255.255.255.0；

网关：192.168.6.1；

端口类型：以太网端口；

主端口：CTE0.B.P4。

DCA-Storage 显示结果：

逻辑端口 ? 租户 所有租户							
创建 管理路由 回切 管理IPsec策略 删除 🔗 🔄							
☐	名称 ? 47	所属租户	数据协议 ?	IP ?	子网掩码/前缀 ?	网关 ?	主端口/当前端口 ? 47 操作 ?
☐	REP-2	--	无	192.168.6.206	255.255.255.0	192.168.6.1	CTE0.B.P4 更多
☐	REP-1	--	无	192.168.6.205	255.255.255.0	192.168.6.1	CTE0.A.P4 更多

DCB-Storage 显示结果：

逻辑端口		租户					
创建		管理路由		回切		管理IPsec策略	
删除							
名称	所属租户	数据协议	IP	子网掩码/前缀	网关	主端口/当前端口	操作
REP-2	--	无	192.168.6.214	255.255.255.0	192.168.6.1	CTE0.B.P4	更多
REP-1	--	无	192.168.6.213	255.255.255.0	192.168.6.1	CTE0.A.P4	更多

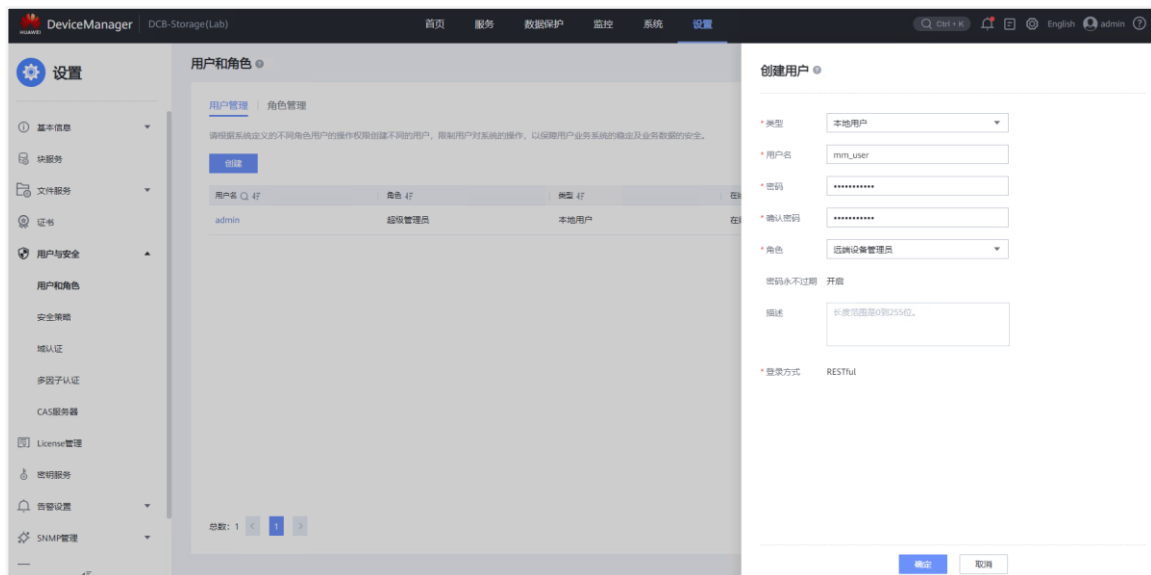
步骤 3 创建远端设备鉴权用户

登录远端设备的 DeviceManager。

选择“设置 > 用户与安全 > 用户和角色 > 用户管理”。

单击“创建”。系统右侧弹出“创建用户”页面。

设置用户信息：在“类型”中选择“本地用户”；用户名：mm_user；角色选择“远端设备管理员”。

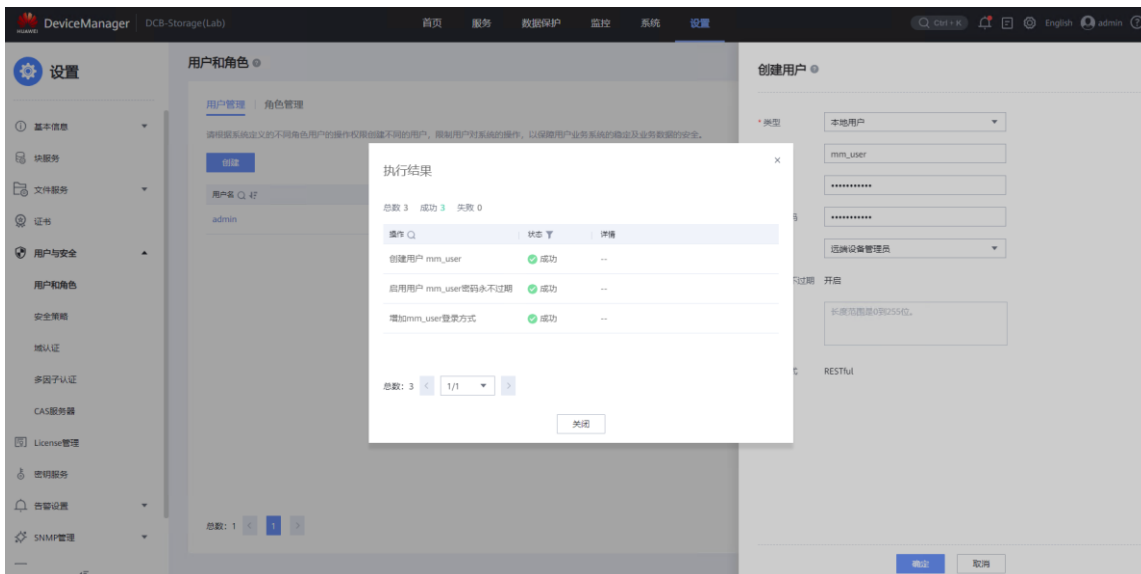


创建用户

- * 类型: 本地用户
- * 用户名: mm_user
- * 密码: [masked]
- * 确认密码: [masked]
- * 角色: 远端设备管理员
- 密码永不过期: 开启
- 描述: 长度范围是0到255位。
- * 登录方式: RESTful

确定 取消

单击“确定”。



点击“关闭”。



步骤 4 增加远端设备

登录本端设备的 DeviceManager。

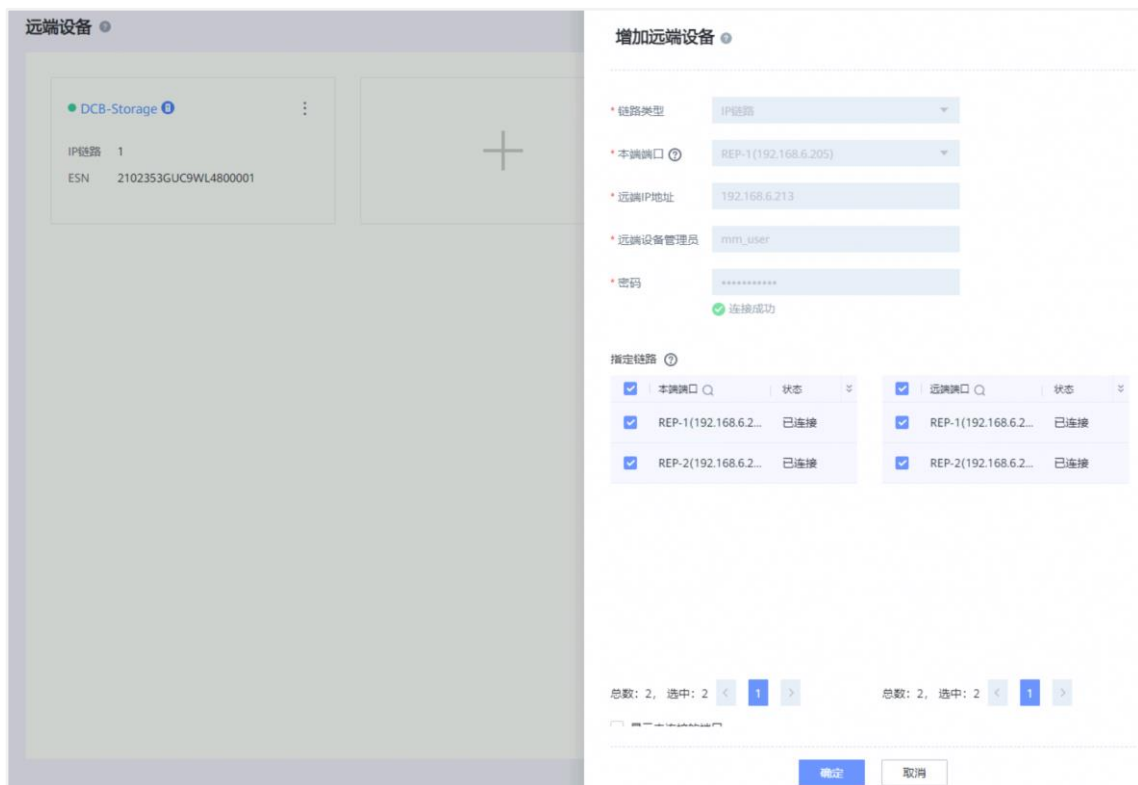
选择“数据保护 > 配置 > 远端设备”。

单击  图标。系统右侧弹出“增加远端设备”页面。

设置增加远端设备的信息：链路类型为 IP 链路；选择 IP 链路的本端端口：REP-1(192.168.6.205)；设置远端 IP 地址：192.168.6.213；设置远端设备管理员：mm_user。



单击“连接”，勾选冗余链路。



连接成功后单击“确定”。



步骤 5 创建远程复制租户 Pair

选择“数据保护 > 对象 > 多租户 > 远程复制租户 Pair”。

单击“创建”。

系统右侧弹出“创建远程复制租户 Pair”页面。

选择需要创建远程复制租户 Pair 的本端租户。

在“远端设备”的下拉列表中选择用于创建远程复制租户 Pair 的远端设备。

设置远程复制租户 Pair 的创建模式为“自动”。

设置远程复制同步的配置信息。

勾选“同步共享和鉴权”和“同步网络配置”。

创建远程复制租户Pair ?

Pair

* 本端租户 Tenant-B

本端设备

DCA-Storage

远端设备

* DCB-Storage(2102353GUC9WL4800001) 增加



* Pair创建模式 ☒ 自动 ☐ 手动

* 远端租户 Tenant-B

* 远端Netspace System-defined

同步共享和鉴权 ☒

同步网络配置 ☒

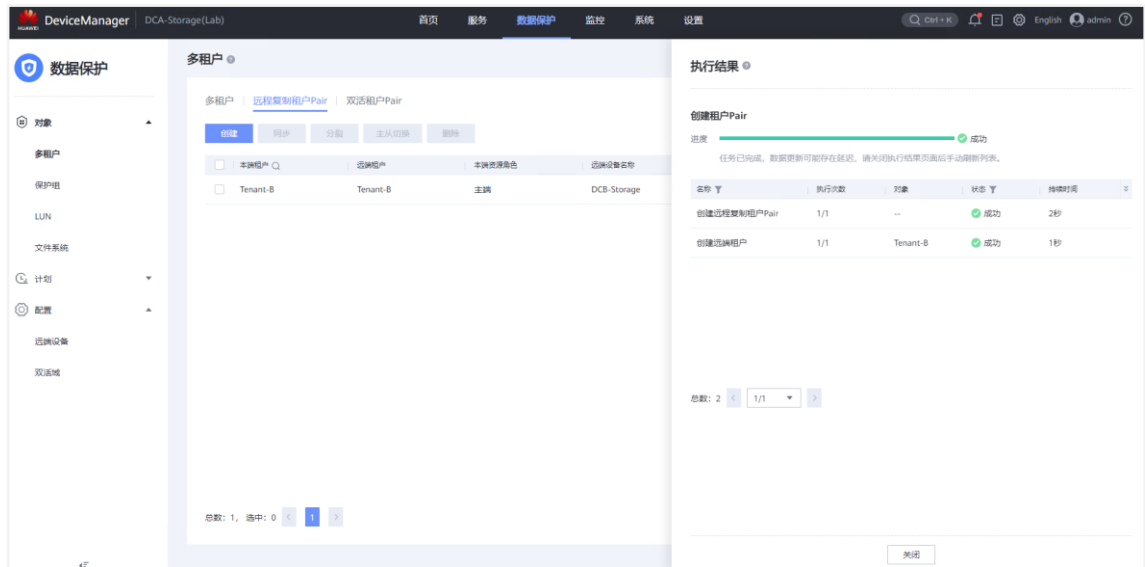
逻辑端口

增加 移除

	名称 Q 47	数据协议	IP地址	本端端口	操作
					

确定 取消

单击“确定”。



单击“关闭”。

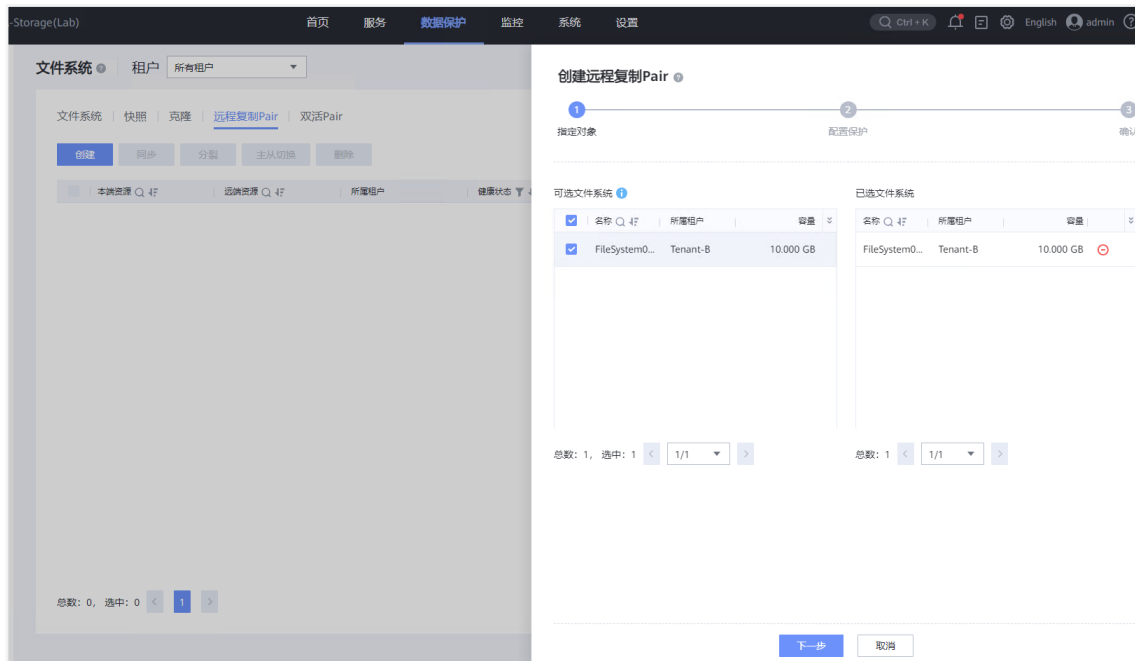


步骤 6 创建远程复制 Pair

选择“数据保护 > 对象 > 文件系统 > 远程复制 Pair”。

单击“创建”。系统右侧弹出“创建远程复制 Pair”页面。

勾选要创建远程复制的文件系统，增加文件系统到“已选文件系统”区域。



单击“下一步”，系统进入配置保护页面。

设置信息：

同步配置：是；

Pair 创建模式：自动；

同步类型：手动；

其他保持默认。



* 创建方式 ?

☒ 基于文件系统
 ☐ 基于一致性快照

保留一致性快照 ?

☐ 启用

* Pair创建模式 ?

☒ 自动
 ☐ 手动

* 远端存储池

StoragePool001

远端资源名称 ?

与本端一致

* 同步类型 ?

手动

* 速率 ?

中

* 从副本快照保留策略 ?

不保留

* 低空间占用同步策略 ?

☒ 开启
 ☐ 关闭

如果文件系统已配置SmartMobility或远端设备不支持低空间占用同步策略，该策略开启后将不会生效。

* 用户快照同步策略 ?

不同步

单击“下一步”，系统进入信息汇总页面。

DeviceManager

DCA-Storage(Lab)

[首页](#)
[服务](#)
[数据保护](#)
[监控](#)
[系统](#)
[设置](#)

English
admin

数据保护

对象

多租户

保护组

LUN

文件系统

计划

配置

远端设备

双活域

文件系统

租户

所有租户

文件系统

快照

克隆

远程复制Pair

双活Pair

创建

同步

分拆

主从切换

删除

本端资源	远端资源	所属租户	健康状态

总数: 0, 选中: 0

创建远程复制Pair

1 指定对象

2 配置保护

3 确认

远程复制租户Pair

Tenant-0 - Tenant-0

Pair创建模式

自动

远端存储池

StoragePool001

远端资源名称

与本端一致

速率

中

同步策略

自动

链路压缩

不压缩

同步类型

手动

初始同步

本端、远端数据不一致，创建完成后自动同步

用户快照同步策略

不同步

从副本快照保留策略

不保留

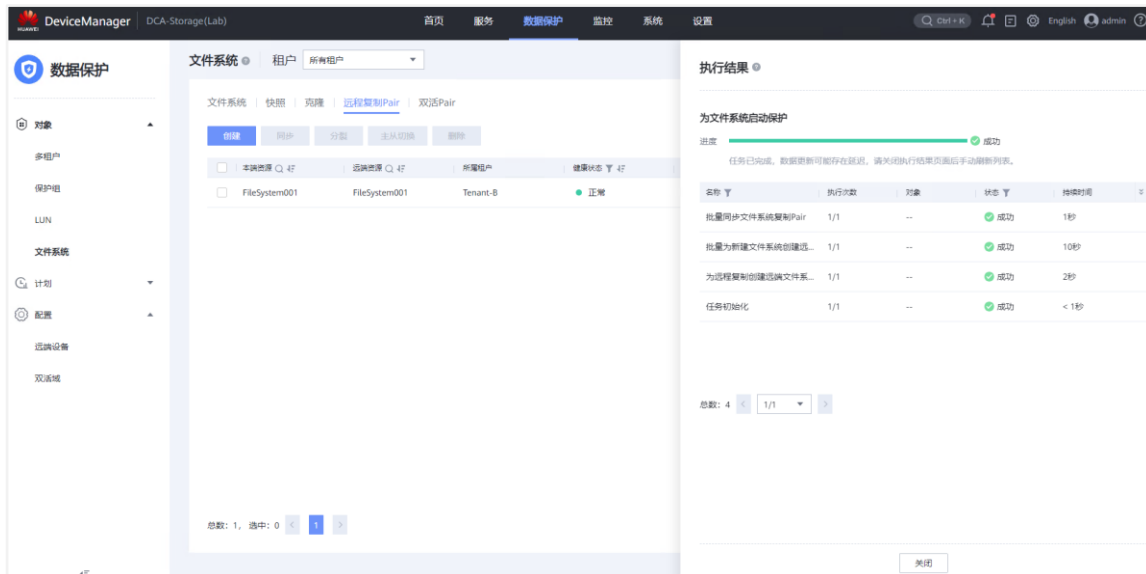
本端资源	本端资源容量
FileSystem001	10,000 GB

上一步

确定

取消

确认创建的远程复制 Pair 信息，单击“确定”。



单击“关闭”。



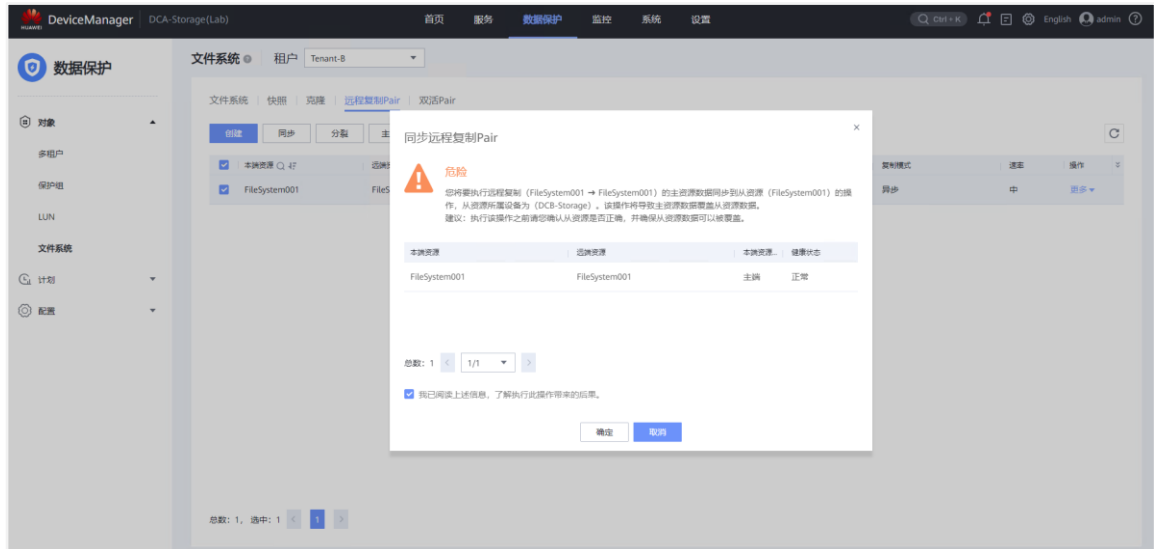
2.3.2 配置验证

步骤 1 手动同步

登录 DCA-Storage 的 DeviceManager，选择“数据保护 > 对象 > 文件系统 > 远程复制 Pair”，在左上方的“租户”下拉框中选择所属的租户“Tenant-B”，选择本端资源“FileSystem001”，单击“同步”进行手动同步。



在弹出的“同步远程复制 Pair”对话框中勾选“我已阅读上述信息，了解执行此操作带来的后果。”后单击“确定”。

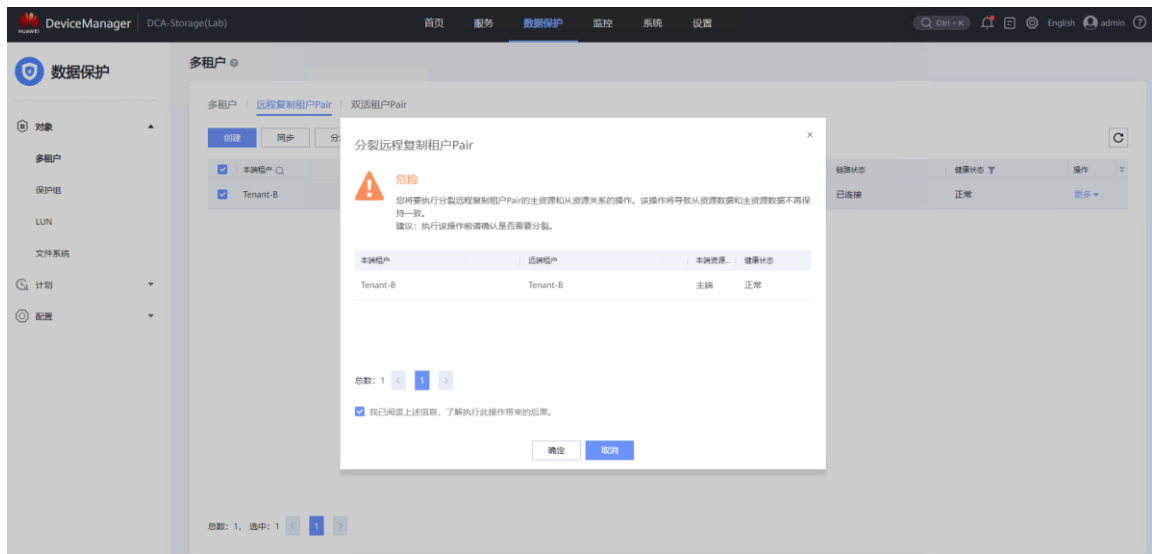


步骤 2 分裂远程复制租户 Pair

手动同步后，选择“数据保护 > 对象 > 多租户 > 远程复制租户 Pair”，单击具体的远程复制租户 Pair “Tenant-B”，单击“分裂”。



在弹出的“分裂远程复制租户 Pair”对话框中勾选“我已阅读上述信息，了解执行此操作带来的后果。”后单击“确定”。



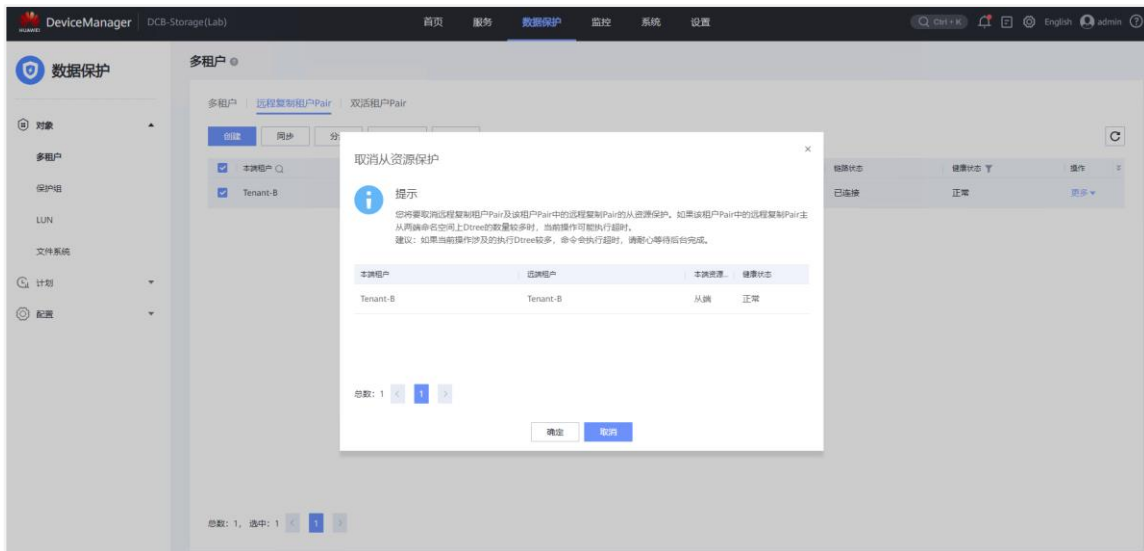
步骤 3 取消从资源保护

登录 DCB-Storage 的 DeviceManager，选择“数据保护 > 对象 > 多租户 > 远程复制租户 Pair”。

选择“Tenant-B”。单击“更多 > 取消从资源保护”。



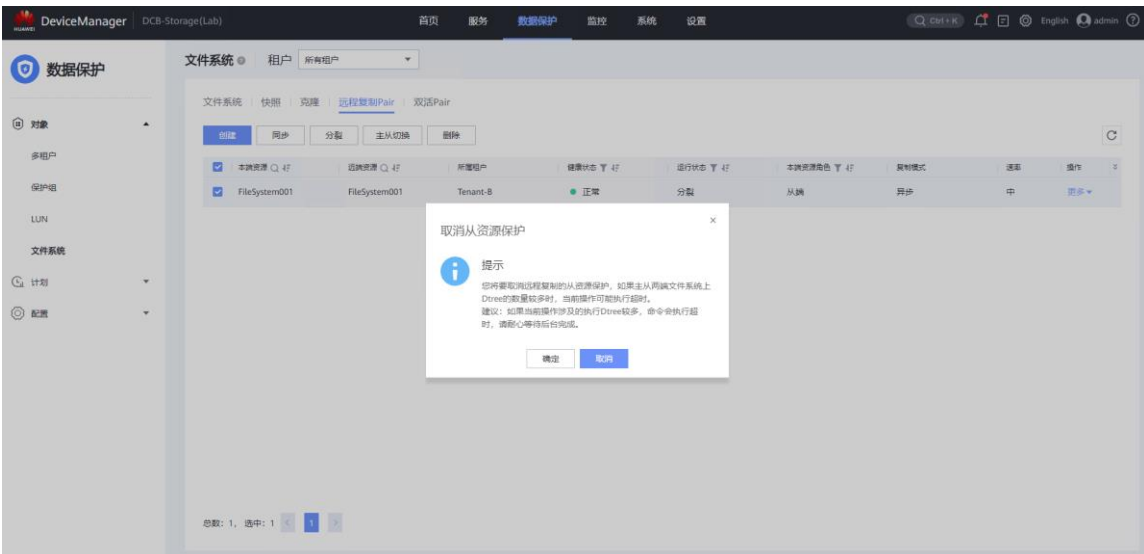
单击“确定”。



选择“数据保护 > 对象 > 文件系统 > 远程复制 Pair”，选择“FileSystem001”。单击“更多 > 取消从资源保护”。



单击“确定”。

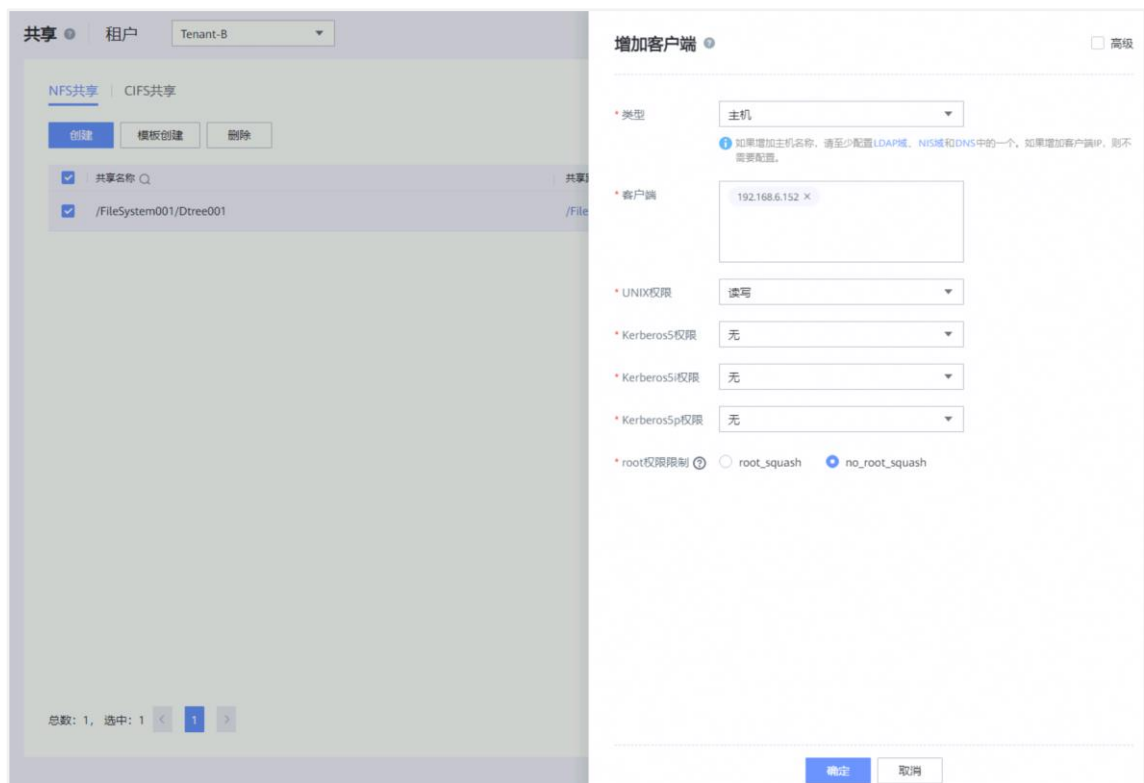


步骤 4 增加 NFS 共享客户端

选择“服务 > 文件服务 > 共享 > NFS 共享”，选择租户“Tenant-B”，在共享路径“/FileSystem001/Dtree001”右侧单击“更多 > 增加客户端”。



添加客户端信息，输入 DCB-Host 的业务 IP 地址“192.168.6.152”，UNIX 权限设为“读写”，root 权限限制设为“no_root_squash”，其它选项保持默认，单击“确定”。



步骤 5 增加逻辑端口

选择“服务 > 网络 > 逻辑端口”。

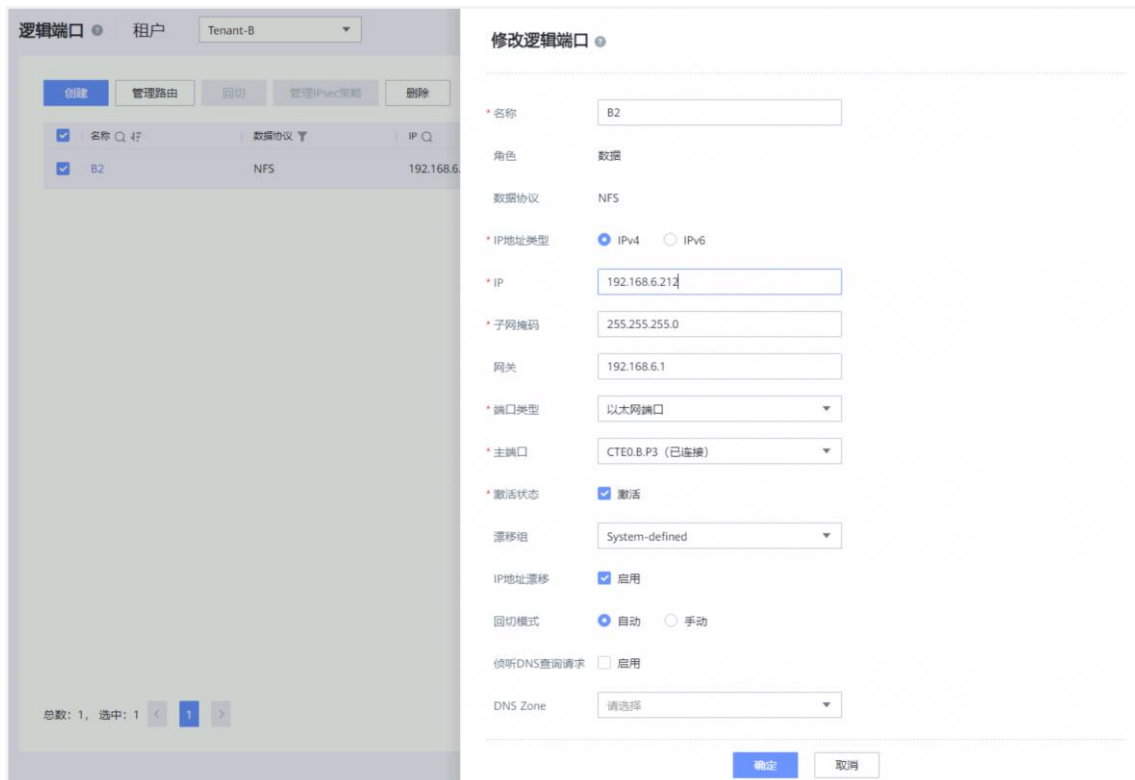
可以看到 DCA-Storage 的逻辑端口配置同步到 DCB-Storage。



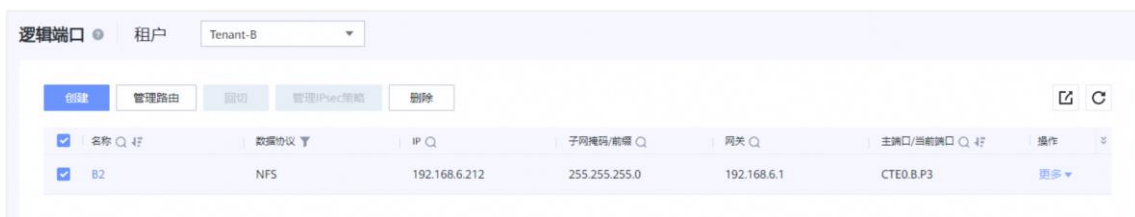
勾选逻辑端口“B2”。右侧单击“更多”选择“修改”。

修改逻辑端口参数：

IP 地址：192.168.6.212；



单击“确定”。

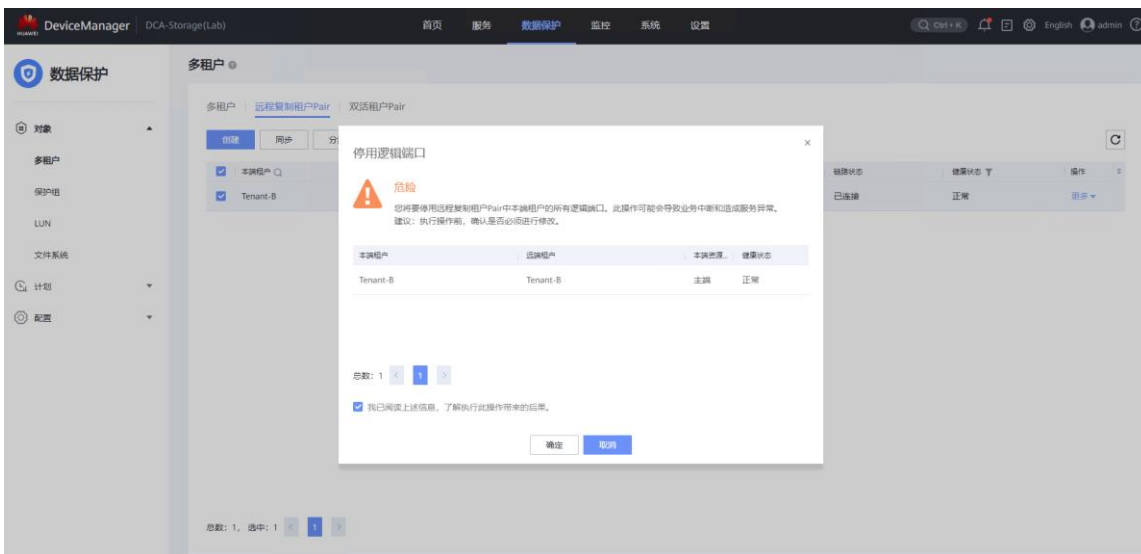


步骤 6 停用主端存储逻辑端口

登录主站点 DCA-Storage 的 DeviceManager，选择“数据保护 > 对象 > 多租户 > 远程复制租户 Pair”。在“Tenant-B”右侧单击“更多 > 停用逻辑端口”。



在弹出的“停用逻辑端口”对话框中勾选“我已阅读上述信息，了解执行此操作带来的后果。”后单击“确定”。



步骤 7 启用从端存储逻辑端口

登录从站点 DCB-Storage 的 DeviceManager，选择“数据保护 > 对象 > 多租户 > 远程复制租户 Pair”。在“Tenant-B”右侧单击“更多 > 启用逻辑端口”。



在弹出的“启用逻辑端口”对话框中勾选“我已阅读上述信息，了解执行此操作带来的后果。”后单击“确定”。



步骤 8 访问 NFS 共享

使用 root 用户登录到 DCB-Host 的客户端。

运行 `showmount -e 192.168.6.212` 命令，查看存储系统当前所有的 NFS 共享。

```
[root@dcb-host ~]# showmount -e 192.168.6.212
Export list for 192.168.6.212:
/FileSystem001/Dtree001 192.168.6.152,192.168.6.151
```

运行 “`mount -t nfs 192.168.6.212:/FileSystem001/Dtree001 /mnt`” 命令，挂载 NFS 共享。

运行 “`df -h`” 命令，确认已挂载成功。

```
[root@dcb-host ~]# mount -t nfs 192.168.6.212:/FileSystem001/Dtree001 /mnt
[root@dcb-host ~]# df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
devtmpfs	1.8G	0	1.8G	0%	/dev
tmpfs	1.8G	0	1.8G	0%	/dev/shm
tmpfs	1.8G	8.7M	1.8G	1%	/run
tmpfs	1.8G	0	1.8G	0%	/sys/fs/cgroup
/dev/sda1	40G	2.7G	35G	8%	/
tmpfs	367M	0	367M	0%	/run/user/0
192.168.6.212:/FileSystem001/Dtree001	8G	0	8G	0%	/mnt

步骤 9 验证测试文件

进入/mnt 目录，查看文件。

```
cd /mnt
cat a.txt
```

回显信息：

```
Hello World!
```

2.4 思考题

问题：

用户创建远程复制租户 Pair 和远程复制 Pair 后，是否可以对远程复制 Pair 执行操作？

参考答案：

当远程复制租户 Pair 和远程复制 Pair 的从资源保护状态一致且远程复制方向一致时，不允许对远程复制 Pair 执行主从切换、取消从资源保护、启用从资源保护操作；以上操作均对远程复制租户 Pair 执行。

当远程复制租户 Pair 和远程复制 Pair 的从资源保护状态不一致或远程复制方向不一致时，可以对远程复制 Pair 执行相关操作，目的是使远程复制租户 Pair 和远程复制 Pair 的状态一致。

对远程复制租户 Pair 执行同步、分裂、主从切换、取消从资源保护、启用从资源保护操作后，DeviceManager 上可以查看远程复制租户 Pair 的执行结果；远程复制 Pair 会在后台运行，可执行以下命令查看远程复制 Pair 的运行状态，确保远程复制 Pair 的操作已执行完成。

```
show remote_replication unified
```

华为认证存储系列教程

HCIP-Storage

双活特性配置

实验指导手册

版本：6.0



华为技术有限公司

版权所有 ©【2026】【华为技术有限公司】。保留一切权利。

未经华为事先书面许可，任何人不得对本资料进行复制、修改、整合、改编、翻译，也不得将本资料或其任何部分或基于本资料的衍生作品提供、出售、转让、再许可给任何他人用于华为许可以外的任何其他用途。

Copyright ©【2026】【Huawei Technologies Co., Ltd.】. All Rights Reserved.

Without prior written permission of Huawei, no one is allowed to reproduce, modify, integrate, adapt, translate this Document, or provide, sell, transfer, sublicense this Document or any part of this Document or its derivative to any third party for any other purpose otherwise permitted by Huawei.

商标声明



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>

华为认证体系介绍

华为认证是华为公司基于“平台+生态”战略，围绕“云-管-端”协同的新ICT技术架构，华为公司打造了覆盖ICT领域的认证体系，包含ICT基础设施、应用与开发两类技术认证及项目管理认证。

根据ICT从业者的学习和进阶需求，华为认证分为工程师级别、高级工程师级别和专家级别三个认证等级。

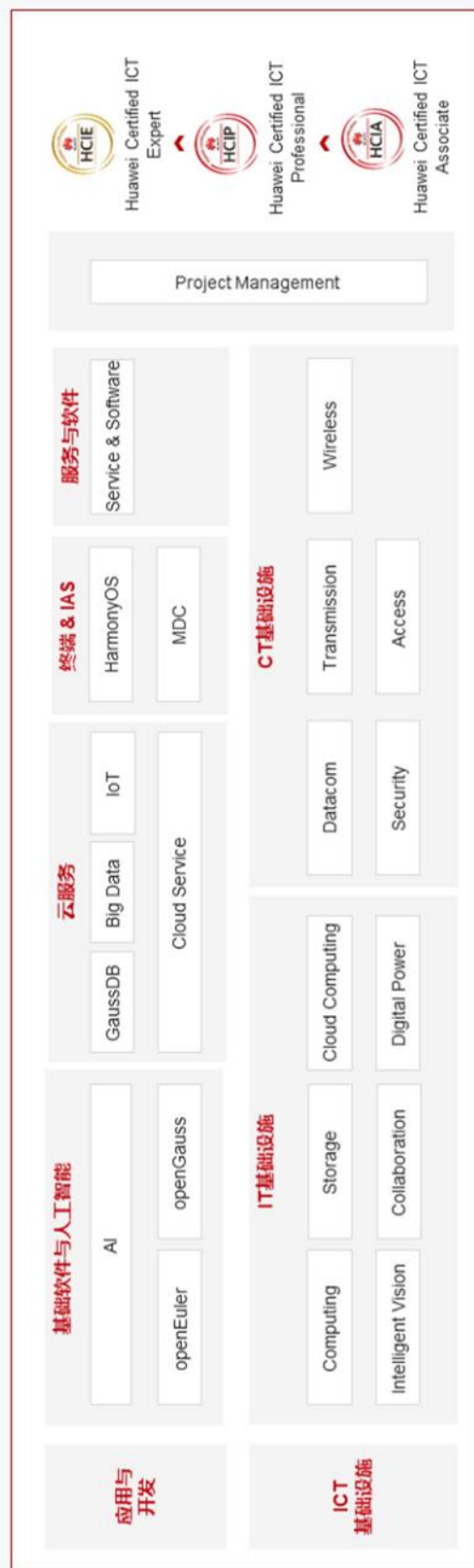
华为认证覆盖ICT全领域，符合ICT融合的技术趋势，致力于提供领先的人才培养体系和认证标准，培养数字化时代新型ICT人才，构建良性ICT人才生态。

HCIP-Storage（Huawei Certified ICT Professional-Storage，华为认证存储高级工程师）主要面向华为公司渠道工程师，高校学生，以及其他ICT从业人员等。HCIP-Storage认证在内容上涵盖闪存存储技术、数据保护技术、分布式存储技术、虚拟化技术、以及存储系统运维管理等内容。

华为认证协助您打开行业之窗，开启改变之门，屹立在存储行业的潮头浪尖！

华为职业认证全景图

华为职业认证覆盖ICT全领域，致力于提供领先的人才培养体系和认证标准，培养数字化时代的新型ICT人才，构建良性的ICT人才生态。



前言

简介

本课程将结合之前学习的内容，进行存储双活特性配置演练，帮助学员巩固和复习之前学习的内容。通过该课程的学习，能够熟悉存储文件系统双活特性配置相关流程和操作。

内容描述

本实验指导书共包含 1 个实验，模拟华为闪存存储的文件系统双活配置。

- 实验一：通过 HyperMetro 特性实现文件系统的双活。

读者知识背景

本课程为华为认证高级课程，为了更好地掌握本书内容，阅读本书的读者应首先具备以下基本条件：

- 了解基本的存储基础知识，建议完成 HCIA-Storage 的学习，并通过了 HCIA-Storage 认证考试。
 - 熟悉 Linux 操作系统操作，了解网络基础知识。
-

目录

前 言	4
简介	4
内容描述	4
读者知识背景	4
1 参考资料及工具	6
1.1 参考资料	6
1.2 软件工具	6
2 存储双活特性配置	7
2.1 实验介绍	7
2.1.1 关于本实验	7
2.1.2 实验目的	7
2.1.3 实验组网拓扑	8
2.2 基础业务配置	9
2.2.1 配置文件系统	9
2.2.2 配置 NFS 共享	15
2.3 基于双活模式的操作	20
2.3.1 配置仲裁服务器	20
2.3.2 配置存储双活	25
2.4 双活测试	37
2.5 思考题	40

1 参考资料及工具

1.1 参考资料

文档中所列出的命令、文档及文档路径仅供参考，请根据实际环境中的不同产品版本使用对应的命令、文档及文档路径。

产品文档



说明：

您可以直接登录华为技术有限公司技术支持网（<http://support.huawei.com/enterprise/>），在搜索栏中，输入“文档或工具名称”，检索、浏览和下载对应文档或工具。

1.2 软件工具

1. PuTTY



说明：

使用开源软件 PuTTY 进行终端登录操作，您可以通过 PuTTY 常用域名（putty.org）浏览和下载对应文档或工具。

2. WinSCP



说明：

用于在 Windows 系统和 Linux 系统间传输文件。推荐使用“WinSCP”，可以根据需求选用其他类似工具。

3. 仲裁软件包



说明：

软件名示例为 OceanStor_Dorado_版本号_QuorumServer_架构类型.zip；

登录 <http://support.huawei.com/enterprise/>，选择“技术支持> 软件下载 > 数据存储 > 闪存存储”，单击对应产品型号，在“选择版本过滤”中设置对应的产品版本，在版本及补丁区域单击对应的版本，进入软件下载界面，下载对应的软件。

2 存储双活特性配置

2.1 实验介绍

2.1.1 关于本实验

本实验为数据中心的本地高可用（存储双活）容灾实验。

本章节以文件服务场景为例进行远程复制特性配置操作。

基于闪存存储的双活容灾实验，将两套 OceanStor V700R001 配置文件系统，并将文件系统挂载给本地主机，通过存储双活功能，实现本地高可用。

创建租户，创建存储池，LUN 和文件系统等基础配置操作请参考《SmartMulti-Tenant 特性配置实验指导手册》。

2.1.2 实验目的

- 理解闪存存储双活容灾的基本原理。
 - 掌握闪存存储双活容灾组网架构。
 - 掌握闪存存储双活容灾的配置步骤和思路。
 - 掌握闪存存储双活容灾的操作配置方法。
-

2.1.3 实验组网拓扑

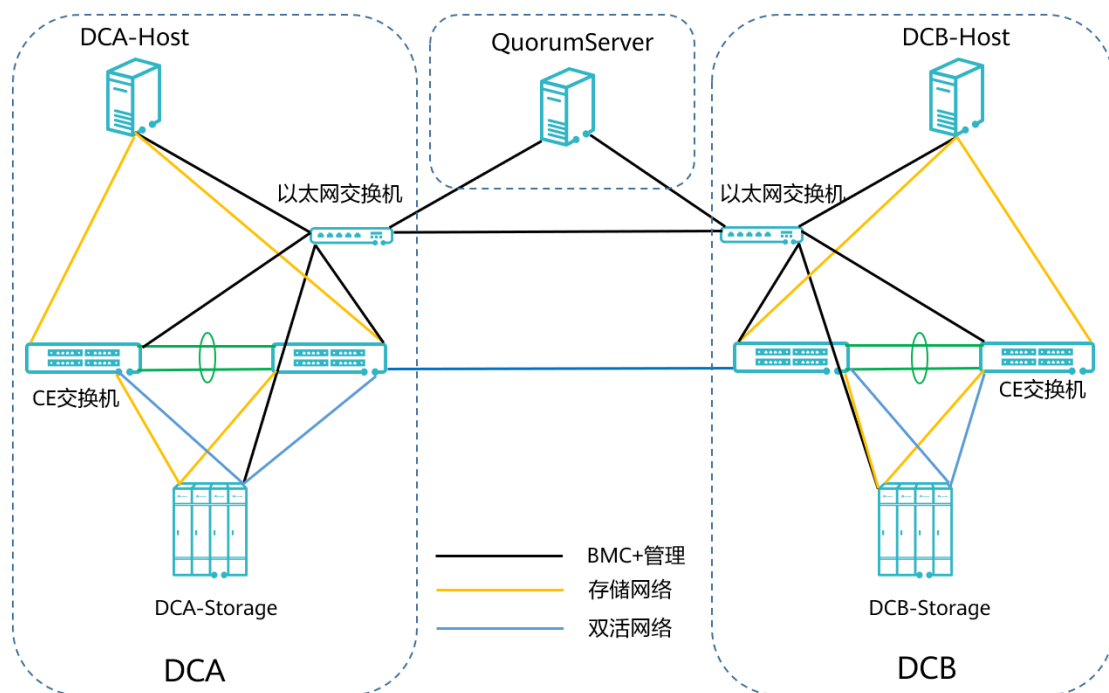


图2-1 闪存存储双活特性实验拓扑图

上图为闪存存储双活特性实验环境组网，其中 DCA 数据中心部署主机 DCA-Host 和存储 DCA-Storage；在 DCB 数据中心部署主机 DCB-Host 和存储 DCB-Storage。DCA 数据中心与 DCB 数据中心互为备份，且在两数据中心之间部署仲裁服务器 QuorumServer。

设备配置

为了满足 HCIP-Storage 实验需要，建议每套实验环境采用以下配置：

设备类型	设备型号	数量	最低软件版本
闪存存储	OceanStor Dorado V700R001 或 OceanStor V700R001	2台	V700R001
应用服务器	X86服务器	2台	CentOS 7.6
管理交换机	以太网交换机	2台	Version 5.160
业务交换机	CE交换机	4台	Version 8.210

网络规划

本实验的网络规划如下表所示：

站点	设备信息	管理IP	存储IP	说明
----	------	------	------	----

DCA	DCA-Host	192.168.5.151	192.168.6.151	业务
	DCA-Storage	192.168.5.200	192.168.6.201	块业务A
			192.168.6.202	文件业务B
			192.168.6.205	复制1
			192.168.6.206	复制2
DCB	DCB-Host	192.168.5.152	192.168.6.152	业务
	DCB-Storage	192.168.5.204	192.168.6.209	块业务A
			192.168.6.212	文件业务B
			192.168.6.213	复制1
			192.168.6.214	复制2
仲裁	QuorumServer	192.168.5.153	192.168.6.153	仲裁



说明：

以上信息仅供参考，实验操作前请联系管理员，根据实际环境进行访问或配置。

2.2 基础业务配置

如已经完成了《远程复制特性配置实验》相关内容，且环境未清理，可复用已有实验环境，仅需要重新激活逻辑端口，无需重复进行其他基础业务配置。

2.2.1 配置文件系统

步骤 1 创建存储池

为保障应用服务器能够使用存储系统的存储空间，先登录 DCA 和 DCB 的存储设备，并创建存储池 StoragePool001。

选择“系统 > 存储池”；

单击“创建”；

设置存储池的参数。

创建存储池 ?

高级

* 名称

StoragePool001

冗余策略

盘级冗余

* 控制框

☒ CTE0

存储池容量

3.461 TB

?

请至少选择6块盘。

单盘容量	类型	可用硬盘总数	每个控制框可选硬盘数量 ?	需要硬盘数量 ?
894.069 GB	SSD	8	8	8 x 1 = 8 选择

单击“确定”。

步骤 2 创建文件系统

在 DCA 存储设备中，创建一个容量为 10GB 的文件系统，名称为“FileSystem001”

选择“服务 > 文件服务 > 文件系统”。

根据 SmartMulti-Tenant 特性配置实验章节创建租户“Tenant-B”。

在左上方的“租户”下拉框中选择需要创建文件系统的租户“Tenant-B”。

单击“创建”。

设置文件系统的名称为“FileSystem001”，所属存储池为“StoragePool001”。

设置文件系统的容量为“10GB”，应用类型为“NAS_Default”，暂时关闭“共享”与“保护”。

创建文件系统 ?

☐ 高级

基本信息

* 名称

FileSystem001

所属租户

Tenant-B

* 所属存储池

StoragePool001

* 安全模式 ?

UNIX

当安全模式设置为UNIX、Native或Mixed时，如果用CIFS协议读写数据，建议开启用户映射开关，并配置映射关系。若未配置用户映射，Administrators组的用户数据会默认统计到root用户配额，非Administrators组的用户数据会默认统计到nobody用户配额。

VAAI ?

☐ 启用

容量与调优

* 容量 ?

10

GB

* 快照预留空间比例(%) ?

20

(0~90)

* 应用类型

NAS_Default

类型：内置
 应用请求大小：16 KB
 数据压缩：已禁用
 重复数据删除：已禁用
 文件系统语义层分布算法：目录均衡模式

共享

确定

取消

单击“确定”。

文件系统 ?

租户 Tenant-B

创建

模板创建

保护

删除

C

☐	名称 Q 4?	容量 4?	健康状态 Y 4?	运行状态 Y 4?	创建时间 Y 4?	WORM 4?	数据保护	共享	配额	Direct个数	操作
☐	FileSystem001	10.000 GB	0.00% 正常	在线	2023-05-31 15:53:55	--				0	更多

总数：1，选中：0

<

1

>

步骤 3 创建 Dtree

为了方便管理目录内所有文件的使用空间和该目录的访问权限，需要创建 Dtree。

选择“服务 > 文件服务 > Dtree”。

在左上方的“租户”下拉框中选择需要创建 Dtree 的文件系统所属的租户“Tenant-B”。

单击“创建”。

设置 Dtree 的名称为“Dtree001”。

创建Dtree ?

☐ 高级

基本信息

* 所属文件系统

FileSystem001

所属租户

Tenant-B

* 名称

Dtree001 X

配额开关

☐ 启用

* 安全模式 ?

UNIX

当安全模式设置为UNIX、Native或Mixed时，如果用CIFS协议读写数据，建议开启用户映射开关，并配置映射关系。若未配置用户映射，Administrators组的用户数据会默认统计到root用户配额，非Administrators组的用户数据会默认统计到nobody用户配额。

确定

取消

单击“确定”。



步骤 4 创建配额

为了限制 Dtree 下的使用空间和文件数量，工程师需要创建配额。

选择“服务 > 文件服务 > 配额 > 自定义配额”。

在左上方的“租户”下拉框中选择需要创建配额的文件系统所属的租户“Tenant-B”。

单击“创建”。

选择创建配额的文件系统“FileSystem001”和 Dtree “Dtree001”。

选择创建的配额类型为“目录配额”。

设置空间配额，硬配额为 10GB，软配额为 8GB。

设置文件数配额，硬配额为 30 千，软配额为 20 千。

创建配额

配额对象

所属租户

Tenant-B

文件系统

FileSystem001

创建

Dtree

Dtree001

配额类型

配额类型

目录配额

限制所选择Dtree的使用空间或文件数量。

用户配额

限制单个用户的使用空间或文件数量。

用户组配额

限制单个用户组的使用空间或文件数量。

空间配额

硬配额

10

GB

(1 KB ~ 256 PB)

软配额

8

GB

(1 KB ~ 256 PB)

文件数配额

硬配额

30

千

(1个 ~ 20亿个文件)

软配额

20

千

(1个 ~ 20亿个文件)

确定

取消

单击“确定”。

配额

租户

Tenant-B

配额列表

自定义配额

文件系统名称

Q

创建

删除

文件系统名称	Dtree Q	配额类型	用户	用户组	用户/用户组类型	空间配额(软/硬)	文件数配额(软/硬)	操作
FileSystem001	Dtree001	目录配额	--	--	--	8,000 GB/10,000 GB	20000/30000	更多

总数: 1, 选中: 1

1/1

总数: 1, 选中: 0

1

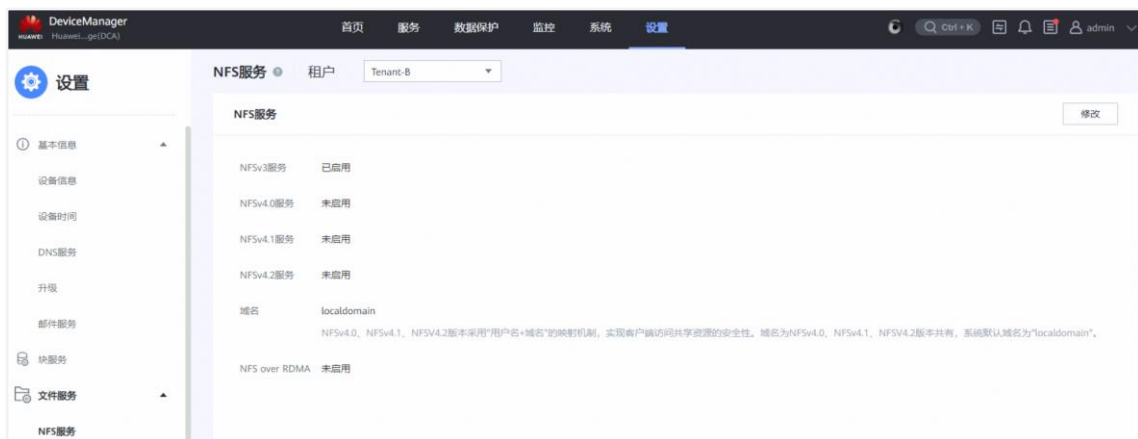
2.2.2 配置 NFS 共享

步骤 1 开启 NFSv4 服务

选择“设置 > 文件服务 > NFS 服务”。

在左上方的“租户”下拉框中选择需要开启 NFSv4 服务的租户“Tenant-B”。

单击右上角的“修改”。

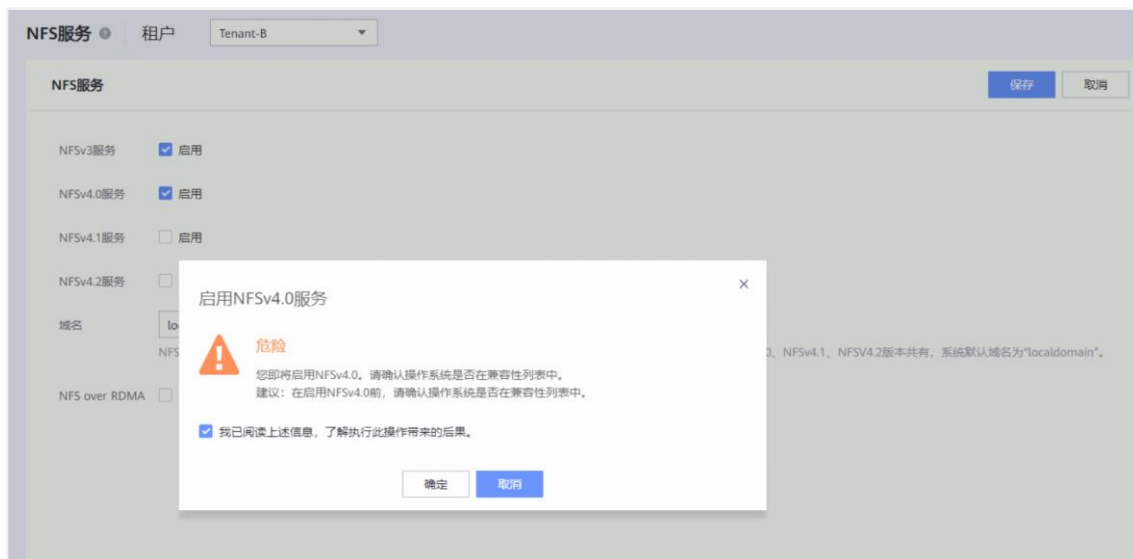


根据业务需要勾选“NFSv4.0 服务”。

单击“保存”。



仔细阅读对话框中的内容，确认无误后选择“我已阅读上述信息，了解执行此操作带来的后果”。



单击“确定”。

步骤 2 配置网络，创建逻辑端口

选择“服务 > 网络 > 逻辑端口”。

单击“创建”。系统右侧弹出“创建逻辑端口”页面。

配置逻辑端口参数：

名称：B2

角色：数据

数据协议：NFS

IP：192.168.6.204

网关：192.168.6.1

子网掩码：255.255.255.0

端口类型：以太网端口

主端口：CTE0.B.P3

激活状态：激活

点击“确定”。

创建逻辑端口 ?

高级

* 名称

B2

* 角色 ?

数据

* 数据协议 ?

☒ NFS
☐ CIFS
☐ NFS + CIFS
☐ iSCSI
☐ NVMe over RoCE
☐ NVMe over TCP
☐ NFS over RDMA
☐ OBJECT

所属租户

Tenant-B

* IP地址类型

☒ IPv4
☐ IPv6

* IP

192.168.6.204

* 子网掩码

255.255.255.0

网关

192.168.6.1

* 端口类型

以太网端口

* 主端口

CTE0.B.P3 (已连接)

* 激活状态

☒ 激活

单击“确定”。

步骤 3 创建 NFS 共享

选择“服务 > 文件服务 > 共享 > NFS 共享”。

在左上方的“租户”下拉框中选择需要创建共享的文件系统所属的租户“Tennant-B”。

单击“创建”。

设置 NFS 共享的基本参数，文件系统选择“FileSystem001”，Dtree 选择“Dtree001”。

创建NFS共享 ?

高级

基本信息

所属租户

Tenant-B

* 文件系统

FileSystem001

创建

Dtree

Dtree001

共享名称

目录 ?

共享路径 ?

/FileSystem001/Dtree001

权限

只有加入权限列表的客户端才允许访问NFS共享。

增加

移除

客户端

类型

UNIX权限

操作

暂无数据

确定

取消

单击“确定”。

服务

租户

Tenant-B

NFS共享

CIFS共享

创建

删除

共享名称

共享路径

操作

/FileSystem001/Dtree001

/FileSystem001/Dtree001

更多

步骤 4 增加 NFS 共享客户端

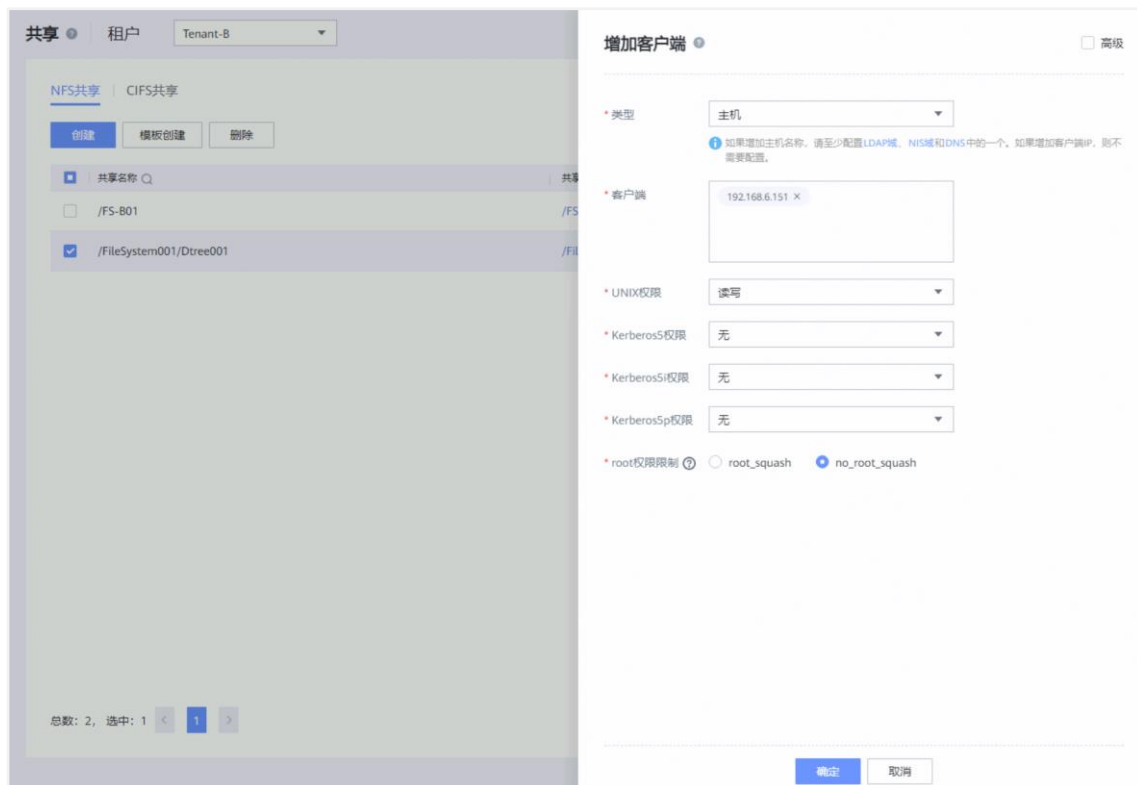
选择“服务 > 文件服务 > 共享 > NFS 共享”。

在左上方的“租户”下拉框中选择需要增加客户端的 NFS 共享所属的租户“Tenant-B”。

单击 NFS 共享所在行右侧的“更多”，选择“增加客户端”。



添加客户端信息，输入 DCA-Host 的 IP 地址“192.168.6.151”，UNIX 权限设为“读写”，root 权限限制设为“no_root_squash”，其它选项保持默认。



单击“确定”。

步骤 5 访问 NFS 共享

使用 root 用户登录到 DCA-Host 的客户端。

运行 showmount -e 192.168.6.204 命令，查看存储系统当前所有的 NFS 共享。

```
[root@dca-host ~]# showmount -e 192.168.6.204
Export list for 192.168.6.204:
/FileSystem001/Dtree001 192.168.6.151
```

运行“mount -t nfs 192.168.6.204:/FileSystem001/Dtree001 /mnt”命令，挂载 NFS 共享。

运行“df -h”命令，确认已挂载成功。

```
[root@dca-host ~]# mount -t nfs 192.168.6.204:/FileSystem001/Dtree001 /mnt
[root@dca-host ~]# df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
devtmpfs	1.8G	0	1.8G	0%	/dev
tmpfs	1.8G	0	1.8G	0%	/dev/shm
tmpfs	1.8G	8.6M	1.8G	1%	/run
tmpfs	1.8G	0	1.8G	0%	/sys/fs/cgroup
/dev/sda1	40G	2.7G	35G	8%	/
tmpfs	367M	0	367M	0%	/run/user/0
192.168.6.204:/FileSystem001/Dtree001	8G	0	8G	0%	/mnt

步骤 6 创建测试文件

进入/mnt 目录，并创建 a.txt 文件。

```
[root@dca-host ~]# cd /mnt
[root@dca-host ~]# touch a.txt
[root@dca-host ~]# ls
```

回显信息：

```
a.txt
```

在新创建的“a.txt”文件中写入“Hello World!”。

```
[root@dca-host ~]# cat <<EOF > a.txt
> Hello World!
> EOF
```

2.3 基于双活模式的操作

2.3.1 配置仲裁服务器

前提：已完成仲裁站点和两个数据中心间的连通，并已获取所需的仲裁软件包。

步骤 1 登录仲裁服务器

登录到系统 CLI，切换到 root 用户模式

步骤 2 配置网络

使用 vim 工具编辑网络配置文件，更改网络信息。

```
# vim /etc/sysconfig/network-scripts/ifcfg-eth0
```

修改 eth0 网口配置，参数“BOOTPROTO”设置为“static”，参数“ONBOOT”设置为“yes”。

```
BOOTPROTO=static
ONBOOT=yes
```

```
IPADDR=192.168.5.153
PREFIX=24
GATEWAY=192.168.5.1
```

修改 eth1 网口配置

```
BOOTPROTO=static
ONBOOT=yes
IPADDR=192.168.6.153
PREFIX=24
GATEWAY=192.168.6.1
```

重启网络服务

```
# systemctl restart network
```

步骤 3 配置防火墙

开启防火墙，并配置的防火墙允许通行端口号为 30002。

```
# systemctl start firewalld
# firewall-cmd --permanent --add-port=30002/tcp
```

回显信息

```
success
```

重启防火墙服务，查询防火墙配置是否生效。

```
# systemctl restart firewalld
# firewall-cmd --query-port=30002/tcp
```

回显信息

```
yes
```

步骤 4 安装仲裁软件

使用 SSH 工具上传与存储版本适配的仲裁软件包

“OceanStor_Dorado_XXX_QuorumServer_X86_64.zip” 到 /root 目录下。

解压并安装软件。

```
# unzip OceanStor_Dorado_XXX_QuorumServer_X86_64.zip
# cd package
# sh ./quorum_server.sh -install
Verify the QuorumServer existence.
The QuorumServer is not installed.
The current user is the root user. A quorum server administrator account needs to be provided. Continue
to install?
<Y|N>:Y
```

输入 Y 继续安装。

```
Enter an administrator account for the quorum server:[default: quorumsvr]:
```

按 “Enter” 使用默认配置继续安装。

```
Created new account: quorumsvr.
```

```
usermod: no changes
Changing password for user quorumsvr.
New password:
Retype new password:
```

输入自定义密码继续安装。

```
passwd: all authentication tokens updated successfully.
Installing the quorum server.
Preparing... ##### [100%]
Updating / installing...
  1:QuorumServer-Dorado_V6-linux ##### [100%]
[Notice] No old configuration need to resume.
Created symlink from /etc/systemd/system/multi-user.target.wants/quorum_server.service to
/etc/systemd/system/quorum_server.service.
QuorumServer install success completed.
```

验证是否安装成功。

```
# qsadmin
start main!
Waiting for connecting to server...
admin:/>
```

步骤 5 配置仲裁服务器

在仲裁软件的 CLI 页面下执行 add server_ip 命令，将仲裁服务器用于提供仲裁服务的所有 IP 地址和软件监测的端口号加入到仲裁软件里面进行管理。

```
admin:/>add server_ip ip=192.168.5.153 port=30002
Command executed successfully.
admin:/>add server_ip ip=192.168.6.153 port=30002
Command executed successfully.
```

配置完成后，执行 show server_ip 命令，当回显中显示刚添加的 IP 和端口号，表示配置成功。

```
admin:/>show server_ip
```

Index	Server IP	Server Port
1	192.168.5.153	30002
2	192.168.6.153	30002

Index	Local IP	Local Port	Remote IP	Remote Port	State
-------	----------	------------	-----------	-------------	-------

步骤 6 导入双活仲裁证书

以使用仲裁服务器内置 CA 给自身和双活本端、远端存储设备签发、导入证书为例。

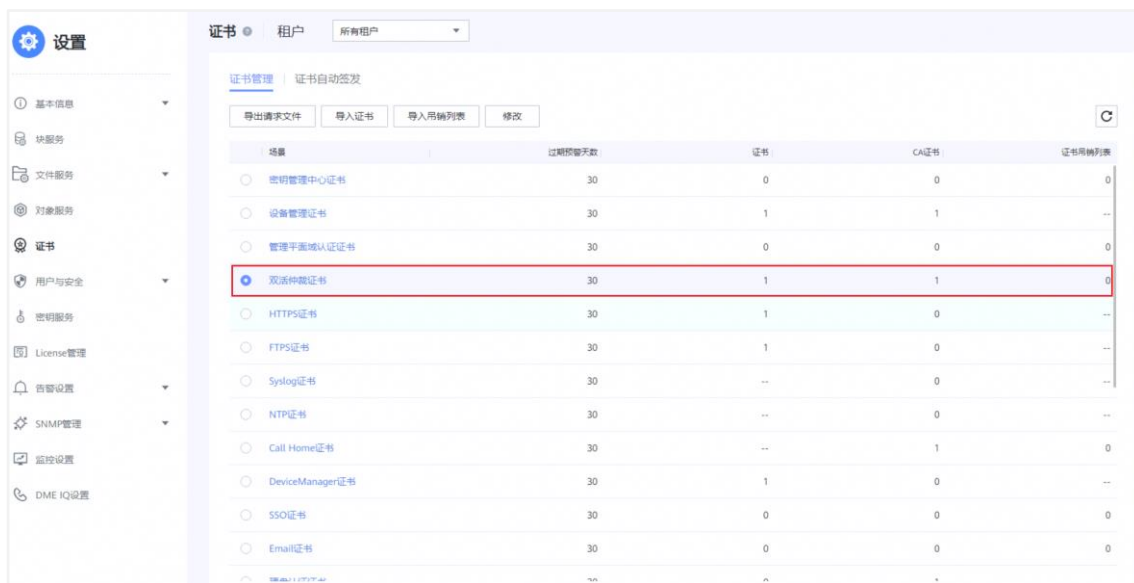
获取存储设备 SN。

分别在双活本端、远端存储系统的 CLI 下执行 show system general 命令，查询存储系统的 SN 号。

```
admin:/>show system general

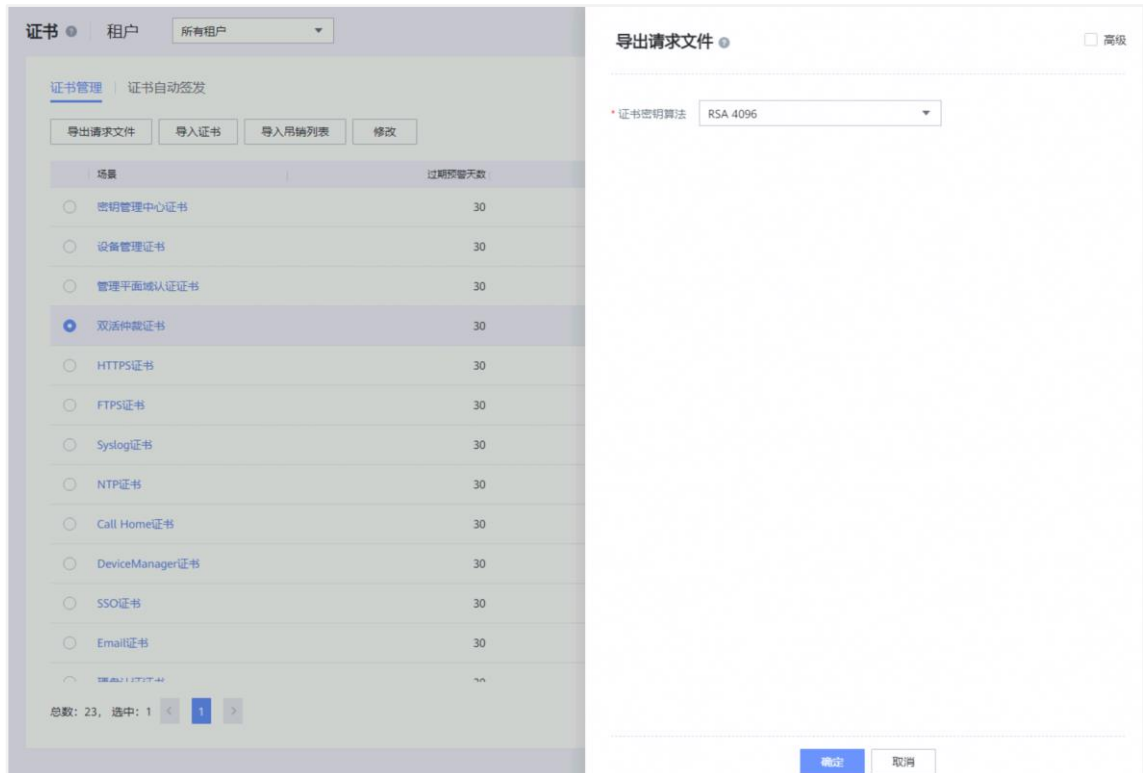
System Name : XXXXXX
Health Status : Normal
Running Status : Normal
Total Capacity : X.XXXTB
SN : XXXXXXXXXXXXXXXXXXXX
Location :
Product Model : XXXXX
Product Version : VX00R00XC00
High Water Level(%) : XX
Low Water Level(%) : XX
WWN : XXXXXXXXXXXXXXXX
Time : XXXX-XX-XX/15:11:15 UTC+08:00
```

分别登录双活本端、远端存储设备 DeviceManager，选择“设置 > 证书 > 证书管理”。选择“双活仲裁证书”，然后单击“导出请求文件”。



在“导出请求文件”页面选择证书密钥算法。

密钥算法包括 RSA 2048、RSA 4096 和 ECC 256，推荐使用 RSA 4096 算法对导出的请求文件进行加密。



单击“确定”。

在仲裁服务器上，分别签发双活本端、远端存储设备的证书请求文件。

将证书请求文件上传到仲裁服务器的“/opt/quorum_server/export_import”目录，将证书请求文件的归属用户与用户组修改为仲裁软件的安装用户与用户组，同时确保证书请求文件有读权限。

```
# cd /opt/quorum_server/export_import
# chown quorumsvr:quorumsvr *
```

在仲裁侧的 CLI 执行命令 `generate tls_cert csr=? [days=?] [cert_name=?] [sign_algorithm=?]` 签发证书。

命令会生成一个与证书请求文件同名但后缀为.crt 的证书文件和一个 cps_ca.crt 的 CA 文件。

```
admin:/>generate tls_cert csr=certificate2102353XXXXXXX.csr cert_name=dcastorage.crt
Command executed successfully.
```

修改证书权限为 777 后进行下载。

```
# chmod 777 *.crt
# ls
cps_ca.crt
dcastorage.crt
```

建议下载 DCA-Storage 证书之后，再继续生成 DCB-Storage 的证书并下载。

注意：通过仲裁服务器生成证书时，会自动生成 cps_ca.crt 文件，后续再次执行 generate 命令生成证书时，会覆盖之前生成的内容，因此每次生成完成后，请及时下载。

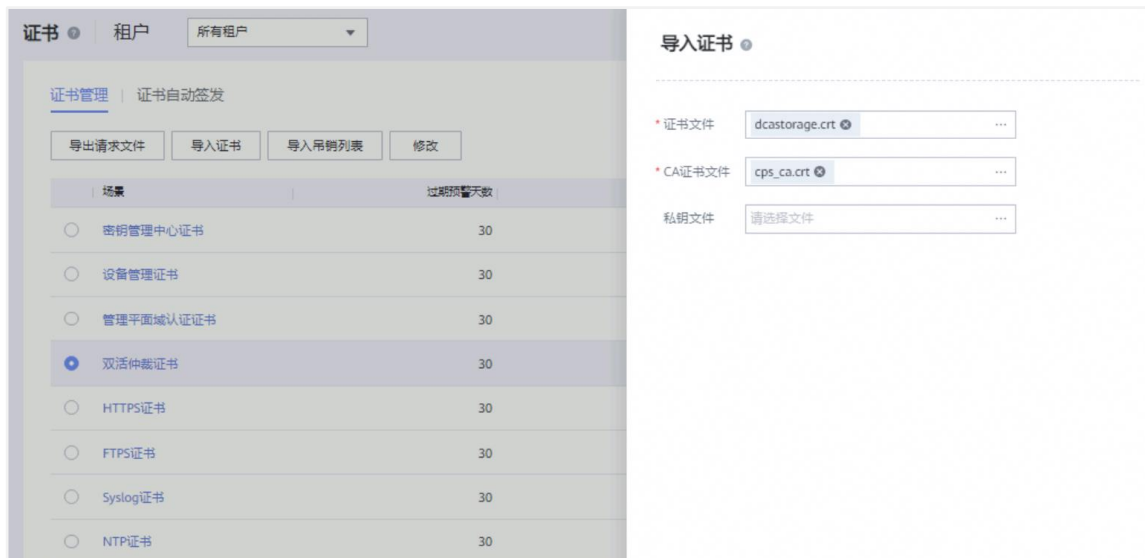
```
admin:/>generate tls_cert csr=certificate2102353XXXXXXX.csr cert_name=dcbstorage.crt
```

Command executed successfully.

分别登录本端和远端存储设备 DeviceManager，选择“设置 > 证书 > 证书管理”。

选择“双活仲裁证书”，单击“导入证书”。

在“导入证书”页面选择已签发的证书和 CA 证书，然后单击“确定”。



步骤 7 添加仲裁白名单

登录仲裁服务器，执行 qsadmin 命令进入仲裁软件的 CLI 页面，在 CLI 页面下执行 add white_list sn=命令添加两台存储的 SN 到白名单。

```
admin:/> add white_list sn=2102XXXXXXXXXX
Command executed successfully.
```

查看白名单。

```
admin:/>show white_list list_all=yes

White List Enabled: YES
Cert id:           0
Cert name:         hm_third_cert
Cert sn:           2102353XXXXXXXXX
Cert sn:           2102353 XXXXXXXXXX
```

2.3.2 配置存储双活

步骤 1 检查 License

分别登录双活本端设备与远端设备的 DeviceManager。双活模式需具备 NAS 基础特性与 NAS HyperMetro 特性。在 DeviceManager 管理界面，NAS 基础特性的“特性名称”显示为“NAS 基础特性（NAS Foundation）”，HyperMetro 特性的“特性名称”显示为“NAS 双活（HyperMetro）”。

选择“设置 > License 管理”。

在中间信息展示区，浏览存储设备已激活的 License 文件信息。

License管理

您可以查看当前设备的License信息，备份已激活的License文件，或导入并激活新的License文件。

License文件序列号
License文件格式版本
AdaptiveLMV100R009C00SPC300

更新License
备份License

特性名称	总容量
克隆 (HyperClone)	无限容量
异构虚拟化 (SmartVirtualization)	无限容量
SAN双活 (HyperMetro)	无限容量
HyperCDP	无限容量
OceanStor OS系统软件	无限容量
重删压缩软件基础版	999999 TB
多租户 (SmartMulti-Tenant)	无限容量
智能配额 (SmartQuota)	无限容量
NAS基础特性 (NAS Foundation)	无限容量
NAS双活 (HyperMetro)	无限容量
WORM (HyperLock)	无限容量
智能文件分级 (SmartMobility-NAS)	无限容量

步骤 2 创建复制逻辑端口

分别登录本端和远端设备的 DeviceManager。

选择“服务 > 网络 > 逻辑端口”。

在左上方的“租户”下拉框中选择“所有租户”。

单击“创建”。

系统右侧弹出“创建逻辑端口”页面。

配置逻辑端口参数：

名称：REP-1；

角色：复制；

IP 地址类型：IPv4；

IP：192.168.6.205（本端 DCA）/192.168.6.213（远端 DCB）；

子网掩码：255.255.255.0；

端口类型：以太网端口；

主端口：CTE0.A.P4。

创建逻辑端口 ?

* 名称

REP-1

* 角色 ?

复制

* IP地址类型

☒ IPv4
 ☐ IPv6

* IP

192.168.6.205

* 子网掩码

255.255.255.0

网关

192.168.6.1

* 端口类型

以太网端口

* 主端口

CTE0.A.P4 (已连接)

单击“确定”。完成逻辑端口创建。

再次“创建逻辑端口”，对话框中输入相关参数：

名称：REP-2；

角色：复制；

IP 地址类型：IPv4；

IP：192.168.6.206（本端 DCA）/192.168.6.214（远端 DCB）；

子网掩码：255.255.255.0；

端口类型：以太网端口；

主端口：CTE0.B.P4。

DCA-Storage 显示结果：

逻辑端口 ? 租户 所有租户							
创建 管理路由 回切 管理IPsec策略 删除 🔗 🔄							
☐	名称 ?	所属租户	数据协议 ?	IP ?	子网掩码/前缀 ?	网关 ?	主端口/当前端口 ? 47 操作 ?
☐	REP-2	--	无	192.168.6.206	255.255.255.0	192.168.6.1	CTE0.B.P4 更多
☐	REP-1	--	无	192.168.6.205	255.255.255.0	192.168.6.1	CTE0.A.P4 更多

DCB-Storage 显示结果：

逻辑端口		租户					
创建		管理路由		回切		管理IPsec策略	
删除							
名称	所属租户	数据协议	IP	子网掩码/前缀	网关	主接口/当前接口	操作
REP-2	--	无	192.168.6.214	255.255.255.0	192.168.6.1	CTE0.B.P4	更多
REP-1	--	无	192.168.6.213	255.255.255.0	192.168.6.1	CTE0.A.P4	更多

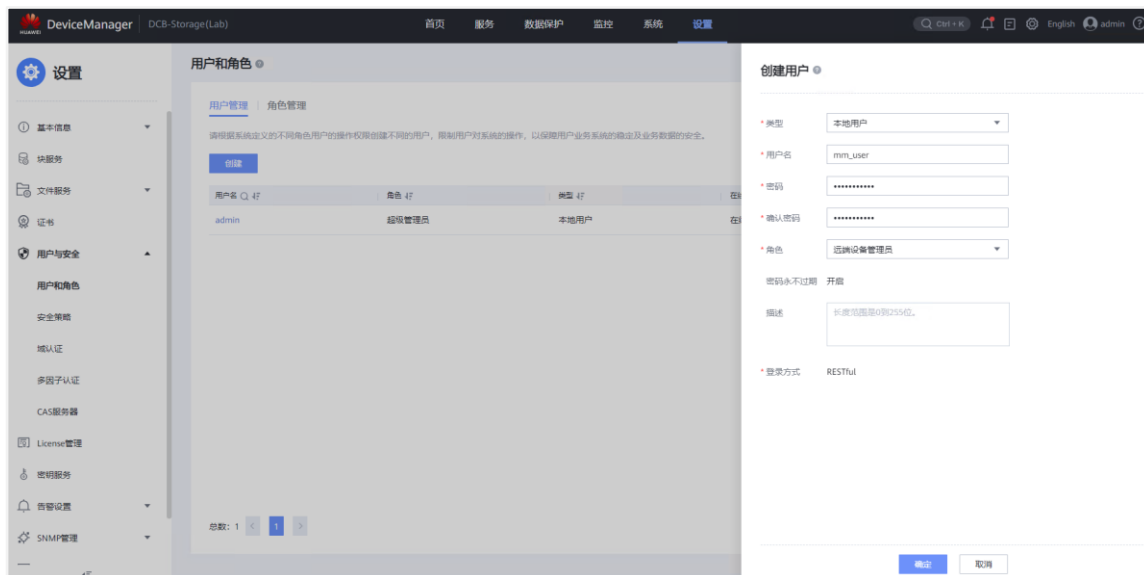
步骤 3 创建远端设备鉴权用户

登录远端设备的 DeviceManager。

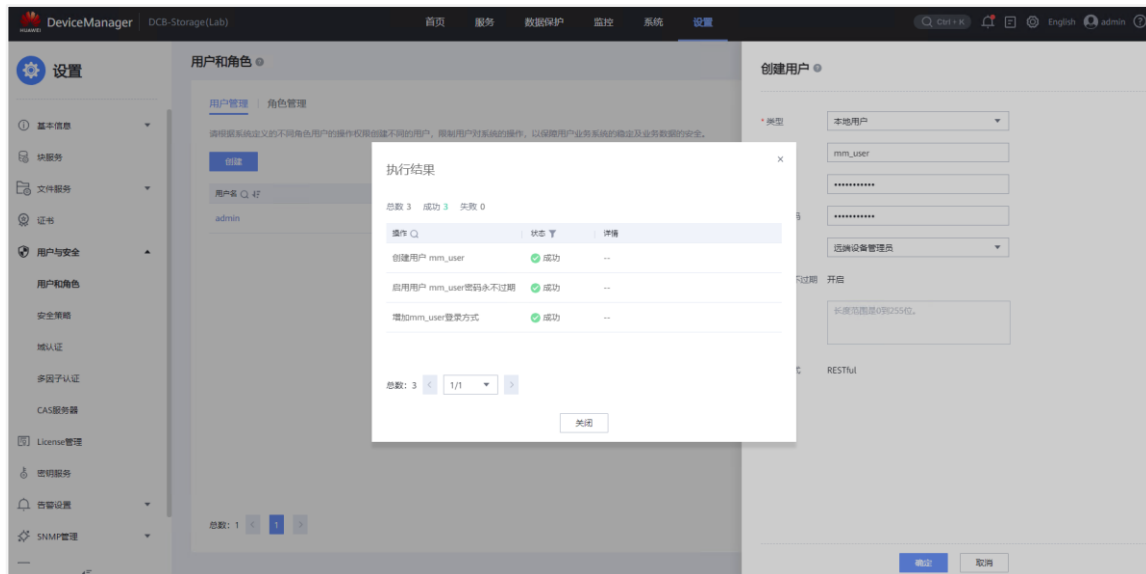
选择“设置 > 用户与安全 > 用户和角色 > 用户管理”。

单击“创建”。系统右侧弹出“创建用户”页面。

设置用户信息：在“类型”中选择“本地用户”；用户名：mm_user；角色选择“远端设备管理员”。



单击“确定”。



点击“关闭”。



步骤 4 增加远端设备

登录本端设备的 DeviceManager。

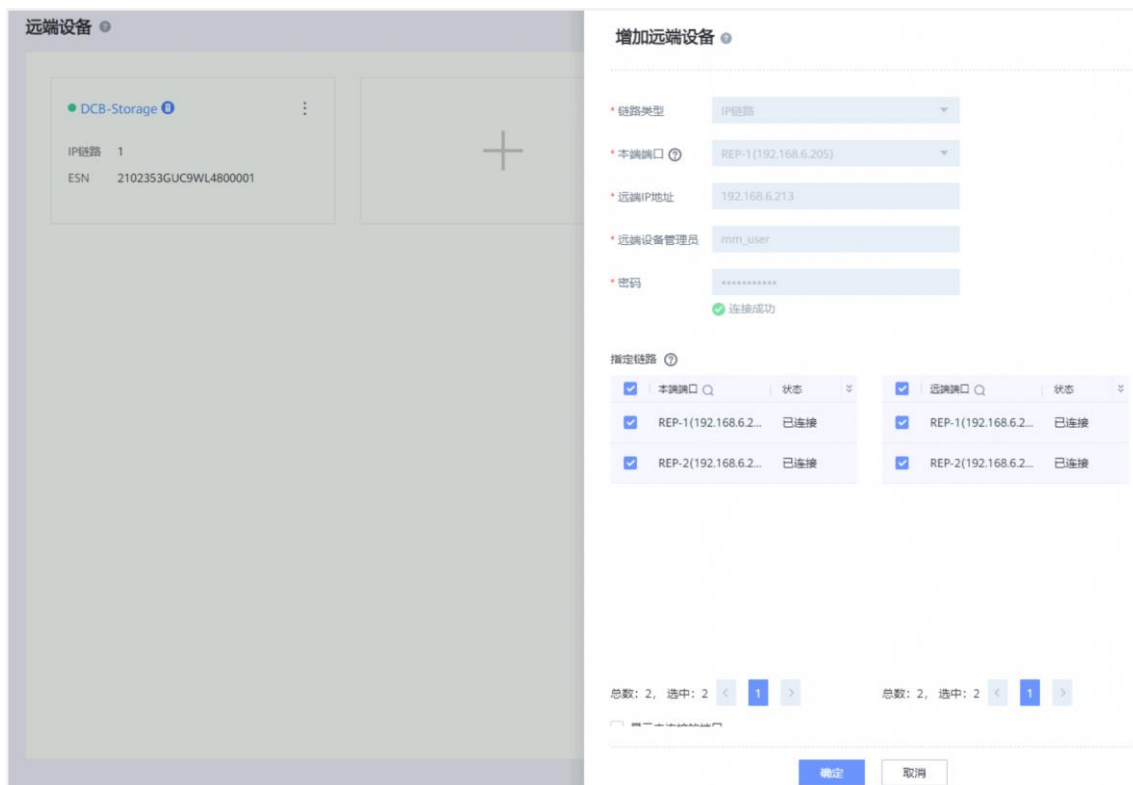
选择“数据保护 > 配置 > 远端设备”。

单击 。系统右侧弹出“增加远端设备”页面。

设置增加远端设备的信息：链路类型为 IP 链路；选择 IP 链路的本端端口：REP-1(192.168.6.205)；设置远端 IP 地址：192.168.6.213；设置远端设备管理员：mm_user。



单击“连接”，勾选冗余链路。



连接成功后单击“确定”。



步骤 5 创建文件双活域

登录双活本端设备的 DeviceManager。

选择“数据保护 > 配置 > 双活域”。

单击文件系统双活域区域的 $\oplus$ 。系统右侧弹出“创建双活域”页面。

设置双活域的“FileHyperMetroDomain_AA”。

双活域

块双活域
应用于块服务。

+

文件系统双活域
应用于文件服务。

+

仲裁服务器

创建双活域

名称FileHyperMetroDomain_AA

描述长度范围是0到127位。

业务类型文件

工作模式

双活AA模式

双活AP模式

同步模式

数据完整后回切业务

仲裁服务器

本端设备
DCA-Storage

远端设备
DCB-Storage(2102353G...

+

+

仲裁服务器

请选择

+

确定

取消

选择双活域的工作模式为“双活 AA 模式”。

选择开启仲裁服务器。

- “远端设备”区域，选择需要加入到双活域的远端设备。
- 在“仲裁服务器”区域，选择需要加入到双活域的仲裁服务器。
- 单击+，在弹出的页面中可以创建仲裁服务器。

双活域

块双活域
应用于块服务。

+

文件系统双活域
应用于文件服务。

+

仲裁服务器

创建仲裁服务器

描述
长度范围是0到127位。

本端仲裁链路

仲裁服务器IP
192.168.5.153

端口
30002

+-

端口类型
管理端口

端口
CTE0.A.MGMTV4...

仲裁服务器IP
192.168.5.153

+-

远端仲裁链路

仲裁服务器IP
192.168.5.153

端口
30002

+-

端口类型
管理端口

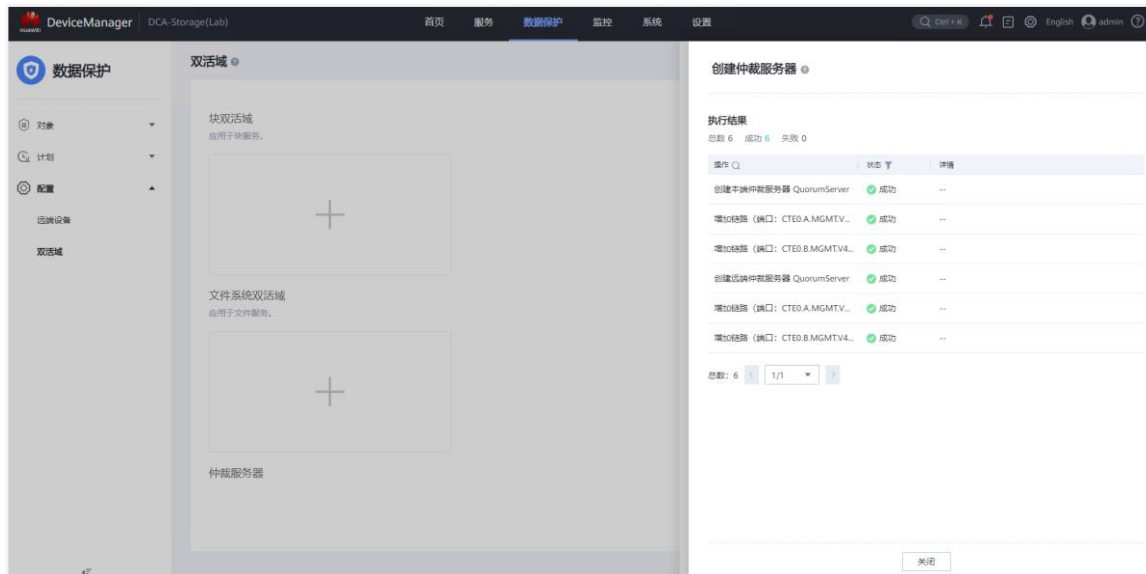
端口
CTE0.A.MGMTV4...

仲裁服务器IP
192.168.5.153

+-

确定 取消

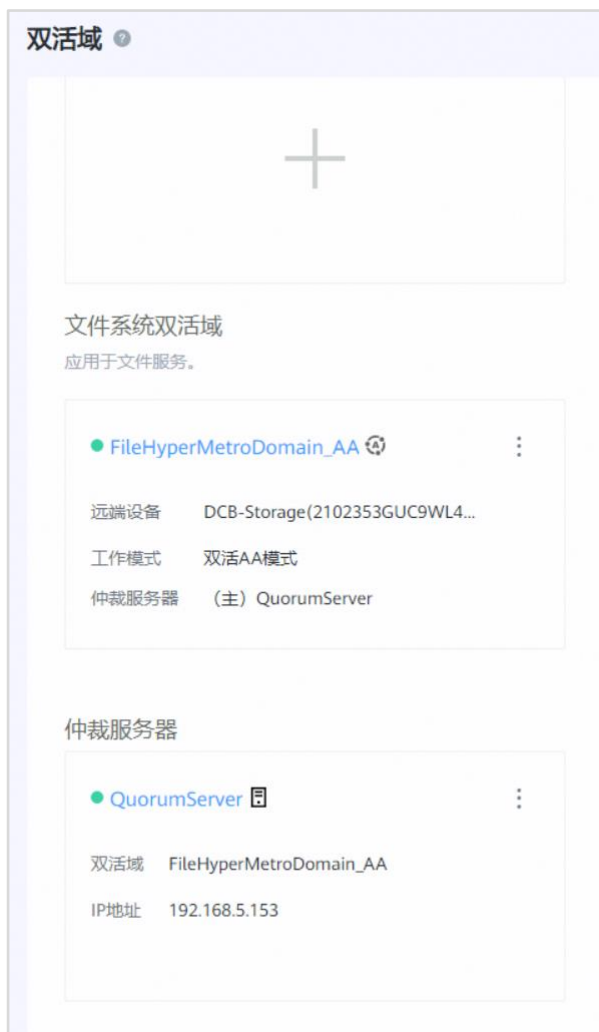
完成配置后，点击“确定”开始创建仲裁服务器。



点击“关闭”返回到创建双活域页面，选择新增的仲裁服务器，点击“确定”。



按照提示完成配置，此时在文件系统双活域列表中，可以看到新创建的双活域信息。



步骤 6 创建双活租户 Pair

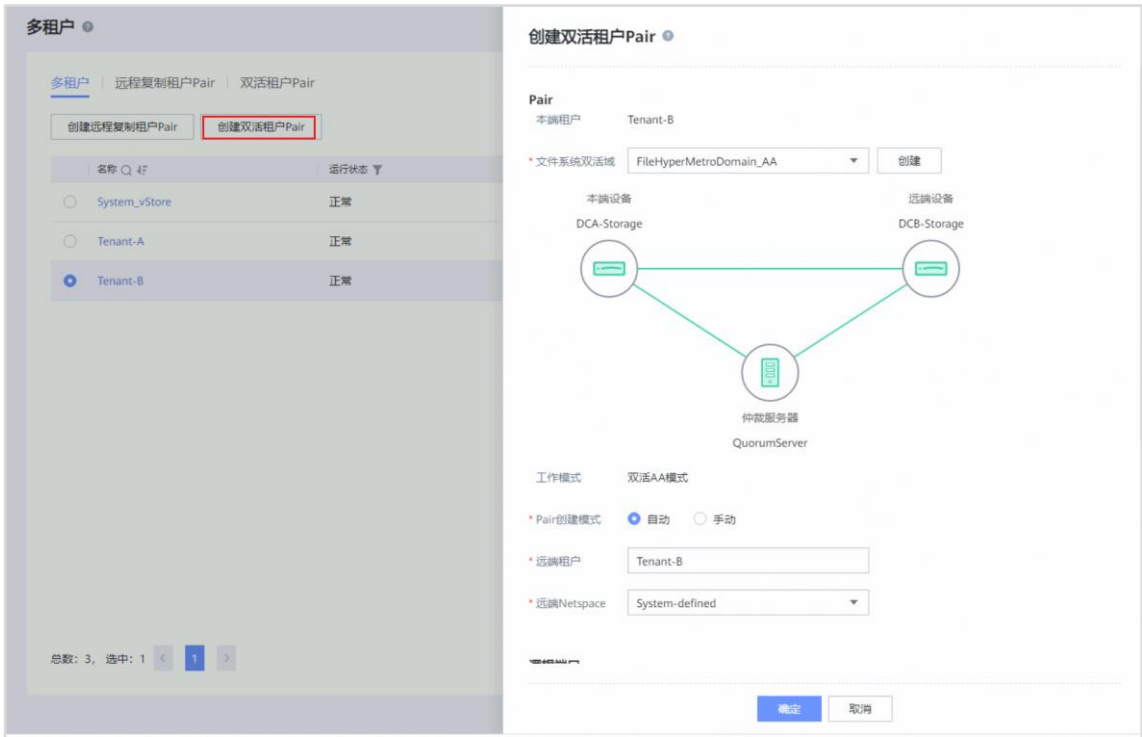
登录双活本端设备的 DeviceManager。

选择“数据保护 > 对象 > 多租户 > 多租户”。

选择要创建双活租户 Pair 的租户，单击“创建双活租户 Pair”。系统右侧弹出“创建双活租户 Pair”页面。

选择创建双活的“文件系统双活域”为“FileHyperMetroDomain_AA”。

设置双活租户 Pair 的创建模式为“自动”，设置远端租户名称为“Tenant-B”。



单击“确定”。按照界面提示完成确认操作。



步骤 7 创建双活 Pair

选择“数据保护 > 对象 > 文件系统 > 双活 Pair”。

在左上方的“租户”下拉框中选择需要创建双活 Pair 的文件系统所属的租户。

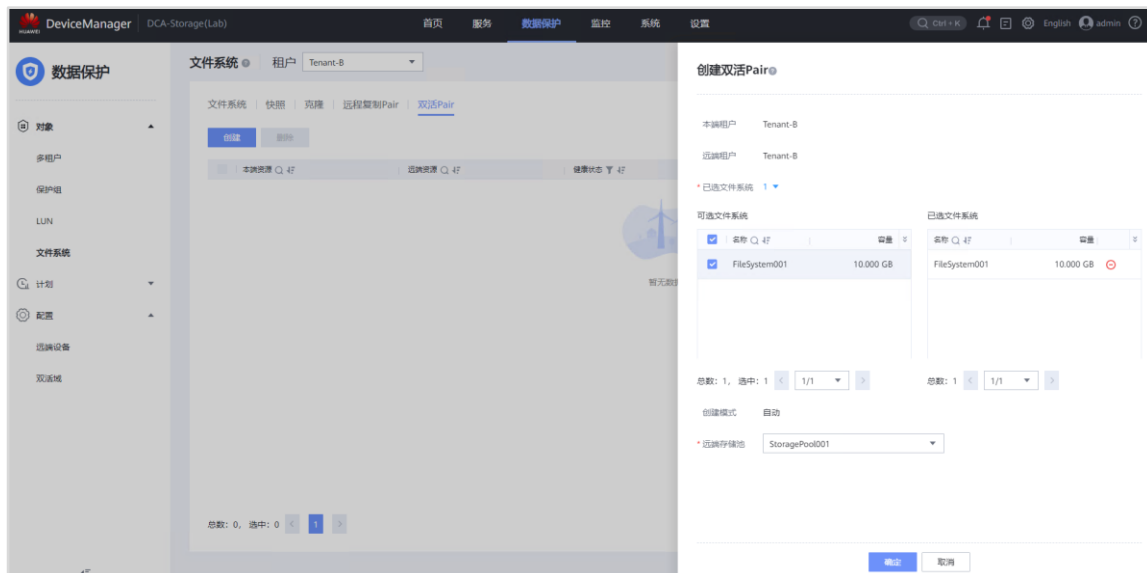
单击“创建”。

系统右侧弹出“创建双活 Pair”对话框。

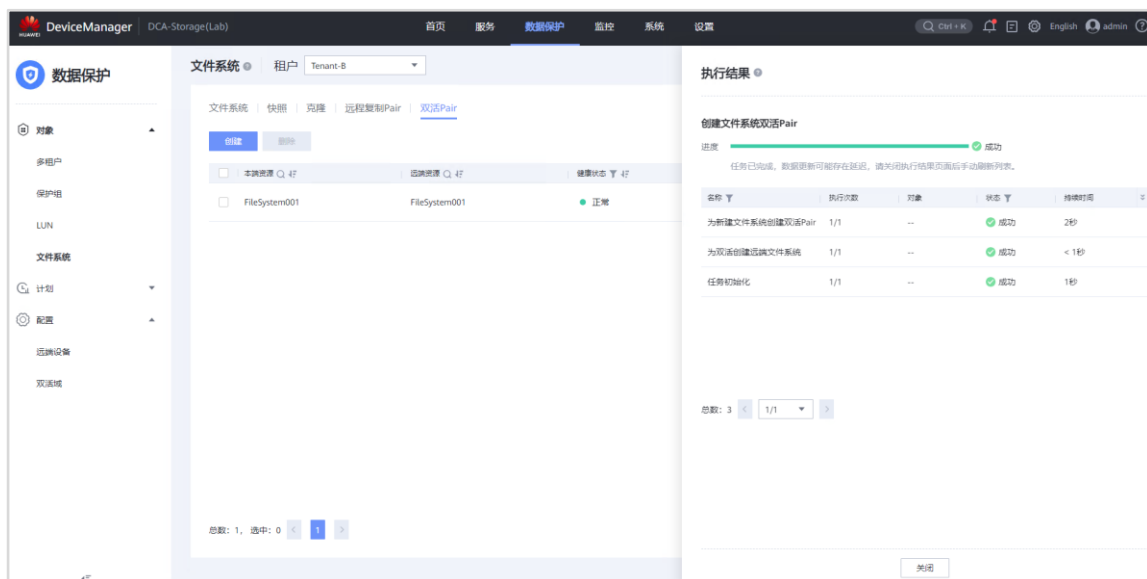
在“本端租户”下拉框中选择需要创建双活 Pair 的文件系统所属的租户。

根据业务需求，在“可选文件系统”区域，勾选一个或多个文件系统。

选择创建远端文件系统的“远端存储池”。系统将在双活租户 Pair 的远端设备上创建远端文件系统，并为本端文件系统和远端文件系统创建双活 Pair。



单击“确定”。



点击“关闭”完成操作并查看创建的双活 Pair。

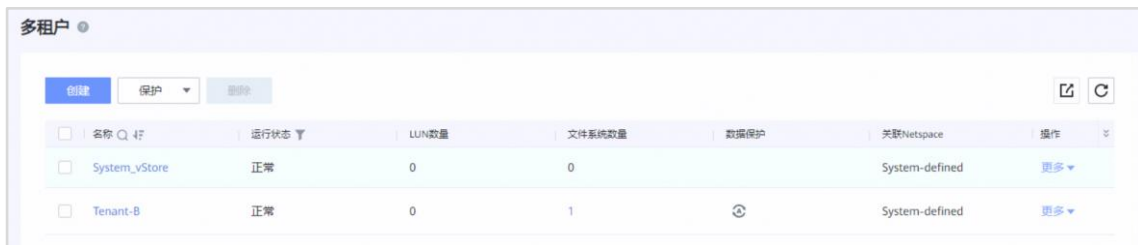


2.4 双活测试

步骤 1 查看远端存储设备配置

登录 DCB 存储设备的 DeviceManager。

选择“服务 > 租户服务 > 多租户”，进行查看。

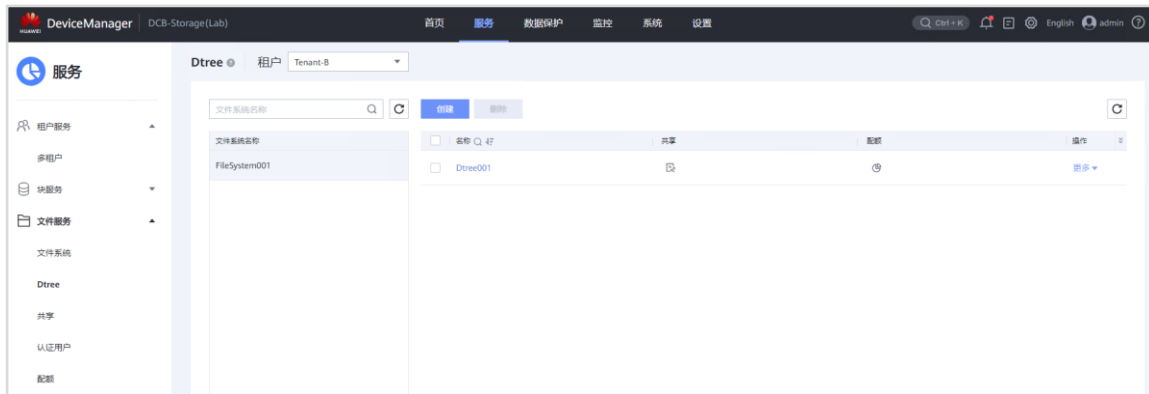


选择“服务 > 文件服务 > 文件系统”。

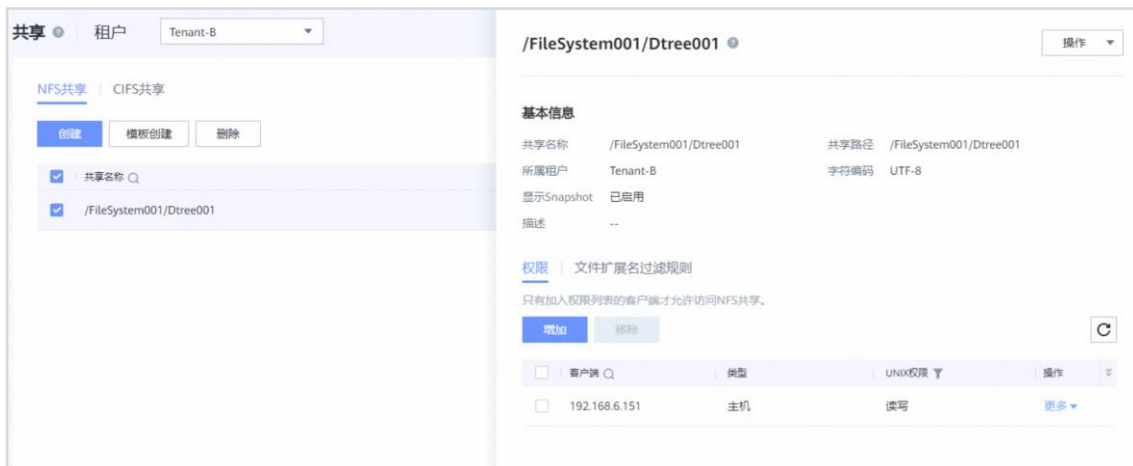
在左上方的“租户”下拉框中选择“Tenant-B”，进行查看。



选择“服务 > 文件服务 > Dtree”，进行查看。



选择“服务 > 文件服务 > 共享”，并单击已呈现的共享路径进行查看。



选择“服务 > 文件服务 > 配额”，进行查看。



经查看，DCB-Storage 与 DCA-Storage 的配置相同，说明双活特性配置已生效。

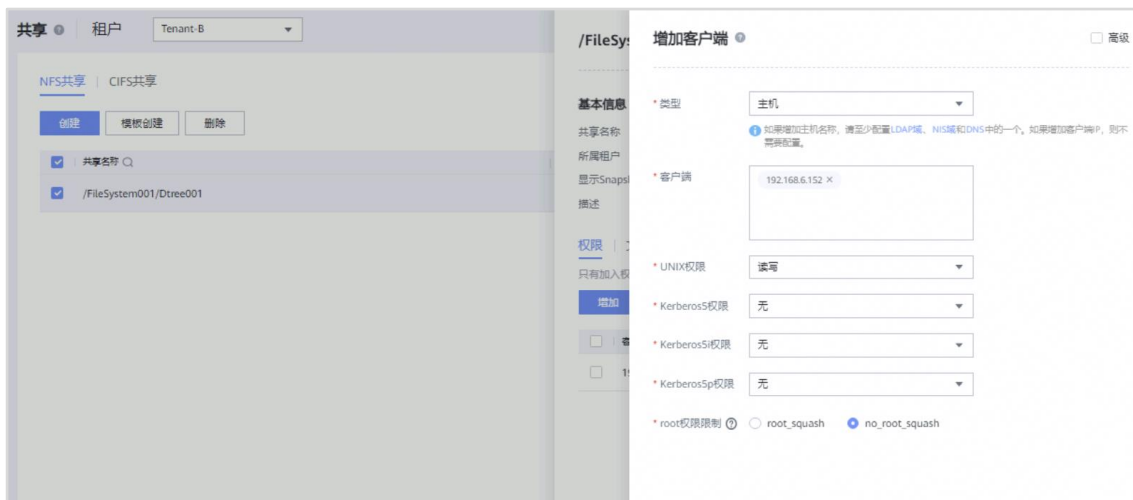
步骤 2 添加共享客户端

登录 DCA 存储设备的 DeviceManager。

选择“服务 > 文件服务 > 共享 > NFS 共享”。

在左上方的“租户”下拉框中选择需要增加客户端的 NFS 共享所属的租户“Tenant-B”。

单击 NFS 共享路径“/Filesystem001/Dtree001”，在“权限”页面点击“增加”，填写 DCB-Host 的主机信息：客户端：192.168.6.152；UNIX 权限设为“读写”，root 权限限制设为“no_root_squash”，其它选项保持默认。



点击“确定”。



登录 DCB 存储设备的 DeviceManager。

选择“服务 > 文件服务 > 共享 > NFS 共享”，单击 NFS 共享路径“/FileSystem001/Dtree001”。



可以发现已自动添加 DCB-Host 的信息。

步骤 3 查看测试文件

使用 root 用户登录到 DCB-Host 的客户端。

运行 showmount -e 192.168.6.204 命令，查看存储系统当前所有的 NFS 共享。

```
[root@dcb-host ~]# showmount -e 192.168.6.204
Export list for 192.168.6.204:
/FileSystem001/Dtree001 192.168.6.151,192.168.6.152
```

运行“mount -t nfs 192.168.6.204:/FileSystem001/Dtree001 /mnt”命令，挂载 NFS 共享。

运行“df -h”命令，确认已挂载成功。

```
[root@dcb-host ~]# mount -t nfs 192.168.6.204:/FileSystem001/Dtree001 /mnt
[root@dcb-host ~]# df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
devtmpfs	1.8G	0	1.8G	0%	/dev
tmpfs	1.8G	0	1.8G	0%	/dev/shm

tmpfs	1.8G	8.6M	1.8G	1%	/run
tmpfs	1.8G	0	1.8G	0%	/sys/fs/cgroup
/dev/sda1	40G	2.7G	35G	8%	/
tmpfs	367M	0	367M	0%	/run/user/0
192.168.6.204:/FileSystem001/Dtree001	10G	0	10G	0%	/mnt

进入/mnt 目录，查看文件。

```
# cd /mnt
# cat a.txt
```

回显信息：

```
Hello World!
```

2.5 思考题

问题：

文件业务双活模式的双活方案部署场景有哪些？

参考答案：

包括本地数据中心部署和跨数据中心部署两种：

1、本地数据中心部署

通常部署在同园区的不同机房中。

主机与存储间通常通过交换机方式通信，对于 NAS 文件系统，需要采用 IP 交换机。同时，存储两端通过双活复制网络连接，保证双活业务的运行。

2、跨数据中心部署

通常部署在同城或相近城市存在的两个数据中心，其物理距离在 300km 以内，两个数据中心均处于运行状态，可同时承担相同业务，提高数据中心的整体服务能力和系统资源利用率，当单数据中心故障时，业务自动切换到另一个数据中心。

对于跨数据中心场景，在长距离传输时，方案需要配套波分设备来保证网络传输时延。同时存储侧两端通过双活复制网络连接，保证双活业务的运行。

华为认证 Storage 系列教程

HCIP-Storage

Ceph安装与配置

实验指导手册

版本：6.0



华为技术有限公司

版权所有 ©【2026】【华为技术有限公司】。保留一切权利。

未经华为事先书面许可，任何人不得对本资料进行复制、修改、整合、改编、翻译，也不得将本资料或其任何部分或基于本资料的衍生作品提供、出售、转让、再许可给任何他人用于华为许可以外的任何其他用途。

Copyright ©【2026】【Huawei Technologies Co., Ltd.】. All Rights Reserved.

Without prior written permission of Huawei, no one is allowed to reproduce, modify, integrate, adapt, translate this Document, or provide, sell, transfer, sublicense this Document or any part of this Document or its derivative to any third party for any other purpose otherwise permitted by Huawei.

商标声明



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>

华为认证体系介绍

华为认证是华为公司基于“平台+生态”战略，围绕“云-管-端”协同的新ICT技术架构，华为公司打造了覆盖ICT领域的认证体系，包含ICT基础设施、应用与开发两类技术认证及项目管理认证。

根据ICT从业者的学习和进阶需求，华为认证分为工程师级别、高级工程师级别和专家级别三个认证等级。

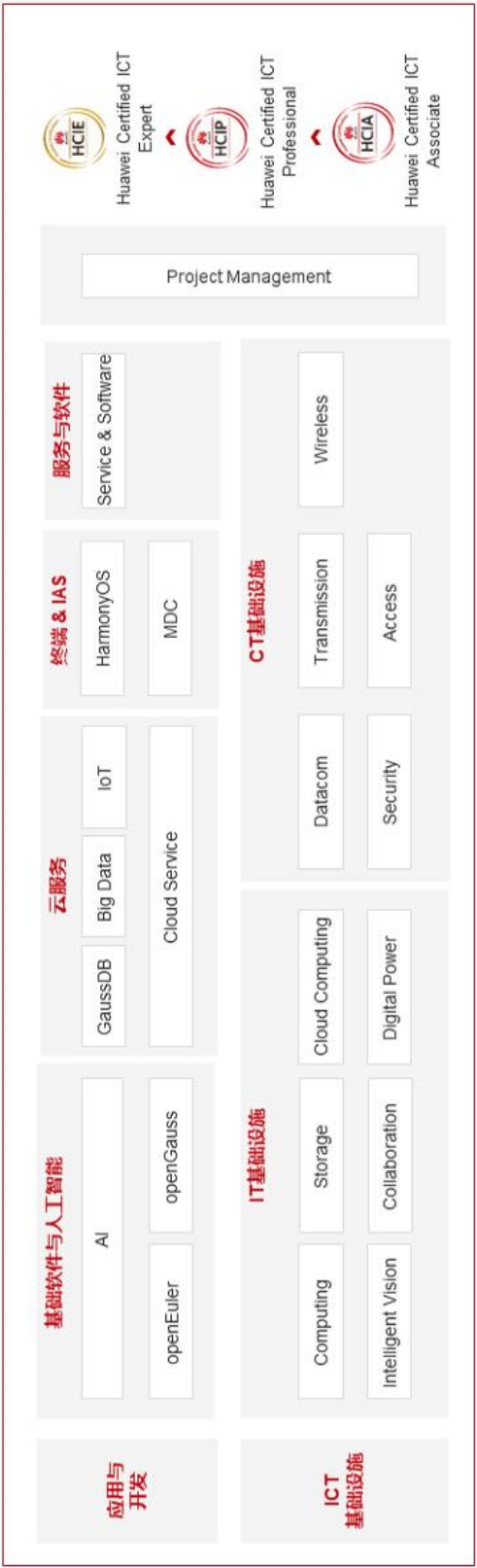
华为认证覆盖ICT全领域，符合ICT融合的技术趋势，致力于提供领先的人才培养体系和认证标准，培养数字化时代新型ICT人才，构建良性ICT人才生态。

HCIP-Storage（Huawei Certified ICT Professional-Storage，华为认证存储高级工程师）主要面向华为公司渠道工程师，高校学生，以及其他ICT从业人员等。HCIP-Storage认证在内容上涵盖闪存存储技术、数据保护技术、分布式存储技术、虚拟化技术、以及存储系统运维管理等内容。

华为认证协助您打开行业之窗，开启改变之门，屹立在存储行业的潮头浪尖！

华为职业认证全景图

华为职业认证覆盖ICT全领域，致力于提供领先的人才培养体系和认证标准，培养数字化时代的新型ICT人才，构建良性的ICT人才生态。



前言

简介

本课程将结合之前学习的理论内容，进行 Ceph 安装配置演练，帮助学员巩固和复习之前学习的内容。通过该课程的学习，能够熟悉 Ceph 系统安装配置相关流程和操作。

内容描述

本实验指导书共包含 6 个实验，逐一介绍了基于 Ceph 的安装、Ceph 资源池的管理、Ceph 集群配置、Ceph 对象存储网关的配置与使用、Ceph 块存储与文件存储的配置和使用。

- 实验一为 Ceph 的安装实验，本实验通过三台 ECS 完成 Ceph 集群的安装及初始化配置，同时进行常见的服务管理操作，如添加、删除等。
- 实验二为 Ceph 的资源池管理的基本操作，包括创建资源池、对资源池进行副本及规格的设置，创建快照等操作。
- 实验三 Ceph 的用户管理，包括用户的创建、用户权限的授权以及授权文件的导入导出等。
- 实验四为 Ceph 对象存储的相关实践，包括对象存储网关的创建、对象桶的配置限额及多区域容灾的配置等。
- 实验五为 Ceph 块存储的相关实践，包括块存储的创建、客户端的映射挂载、RBD 快照的创建删除恢复等操作。
- 实验六为 Ceph 文件存储的相关实践，包括文件存储资源池的创建、客户端挂载等操作。

读者知识背景

本课程为华为认证高级课程，为了更好地掌握本书内容，阅读本书的读者应首先具备以下基本条件：

- 了解基本的存储基础知识，建议完成 HCIA-Storage 的学习，并通过了 HCIA-Storage 认证考试。
 - 熟悉 Linux 操作系统操作，了解网络基础知识。
-

目录

前 言	4
简介	4
内容描述	4
读者知识背景	4
1 Ceph 安装	7
1.1 实验介绍	7
1.1.1 关于本实验	7
1.1.2 实验目的	7
1.1.3 实验环境说明	7
1.1.4 准备实验环境	8
1.2 前期准备	8
1.3 安装 Ceph	10
1.4 创建集群	12
2 资源池管理	15
2.1 实验介绍	15
2.1.1 关于本实验	15
2.1.2 实验目的	15
2.2 副本池创建及配置	15
2.3 纠删码池创建	19
3 Ceph 用户管理实践	21
3.1 实验介绍	21
3.1.1 关于本实验	21
3.1.2 实验目的	21
3.2 用户创建及赋权	21
4 Ceph 对象存储实践	24
4.1 实验介绍	24
4.1.1 关于本实验	24
4.1.2 实验目的	24
4.2 创建对象存储网关	24
4.3 客户端操作	30

4.4 桶配额限制.....	32
5 Ceph 块存储 RBD 实践.....	36
5.1 实验介绍	36
5.1.1 关于本实验	36
5.1.2 实验目的.....	36
5.2 创建块设备并挂载给客户机.....	36
5.3 RBD 设备扩容	39
5.4 RBD 快照管理	40
6 Ceph 文件存储 cephfs 实践.....	43
6.1 实验介绍	43
6.1.1 关于本实验	43
6.1.2 实验目的.....	43
6.2 创建 cephfs 所需的资源.....	43
6.3 客户端挂载.....	44

1 Ceph 安装

1.1 实验介绍

1.1.1 关于本实验

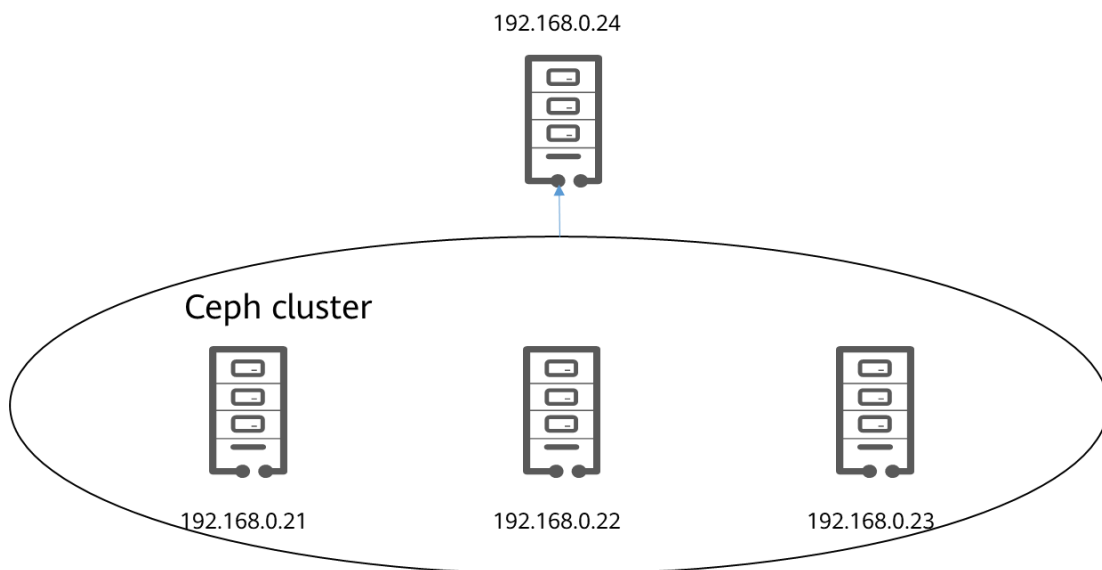
本实验将在前期申请好的 ECS 中完成基于 openEuler 24.03 LTS 的 Ceph 安装，并将所有节点的硬盘配置为 OSD，供后续实验使用。本实验使用 Ceph-Reef 的 18.2.2 版本。

1.1.2 实验目的

- 理解 Ceph 的安装流程；
- 掌握 Ceph 对象业务的配置操作；
- 掌握 Ceph 块业务的配置操作；
- 掌握 Ceph 文件业务的配置操作。

1.1.3 实验环境说明

本实验共使用四台 ECS，分别是 ceph01、ceph02、ceph03 和 Client，对应地址为 192.168.0.21、192.168.0.22、192.168.0.23 和 192.168.0.24，其中 ceph0x 作为 Ceph 节点，都分别挂载 3 块 10G 的数据盘，规格采用 c7.xlarge.2（4U8G），Client 作为客户端节点，规格采用 c7.large.2（2U4G）。组网如下图所示：



请参考下表，设置 ECS 的入方向安全组规则，出方向规则保持默认：

优先级	协议端口	源地址
1	全部	本安全组
1	22	个人实验 PC 公网地址
1	8443	个人实验 PC 公网地址

1.1.4 准备实验环境

检查设备

实验开始之前请按照下表在华为云申请对应的 ECS：

ECS名称	镜像	规格	弹性IP	备注
ceph01	openEuler 24.03 64bit	4vCPUs 8GiB c7.xlarge.2	50Mbit/s, 按流量计费	系统盘40G, 新增三块10G的数据盘
ceph02	openEuler 24.03 64bit	4vCPUs 8GiB c7.xlarge.2	50Mbit/s, 按流量计费	系统盘40G, 新增三块10G的数据盘
ceph03	openEuler 24.03 64bit	4vCPUs 8GiB c7.xlarge.2	50Mbit/s, 按流量计费	系统盘40G, 新增三块10G的数据盘
Client	openEuler 24.03 64bit	2vCPUs 4GiB c7.large.2	50Mbit/s, 按流量计费	系统盘40G

1.2 前期准备

在正式安装前，需完成以下准备过程：

- 1 配置主机名解析；
- 2 关闭防火墙和 selinux；
- 3 配置时间同步；
- 4 配置主机免密登录；

步骤 1 配置主机名及主机名解析

将三台主机主机名设置为 ceph01、ceph02、ceph03

```
[root@ceph01 ~]# hostnamectl set-hostname ceph01
```

登录所有主机，备份 hosts 文件

```
[root@ceph01 ~]# cp /etc/hosts /etc/hosts.bak.$(date +%Y%m%d)
```

向 hosts 文件添加以下内容：

```
[root@ceph01 ~]# tee -a /etc/hosts > /dev/null << EOF
192.168.0.21 ceph01
192.168.0.22 ceph02
192.168.0.23 ceph03
EOF
```

添加完成后，在 ceph01 上 ping 各个主机名，保证网络通信正常。

```
[root@ceph01 ~]# ping ceph01
PING ceph01.novalocal (192.168.0.21) 56(84) bytes of data.
64 bytes from ceph01.novalocal (192.168.0.21): icmp_seq=1 ttl=64 time=0.038 ms

--- ceph01.novalocal ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.038/0.038/0.038/0.000 ms
[root@ceph01 ~]# ping ceph02
PING ceph02.novalocal (192.168.0.22) 56(84) bytes of data.
64 bytes from ceph02.novalocal (192.168.0.22): icmp_seq=2 ttl=64 time=0.196 ms
64 bytes from ceph02.novalocal (192.168.0.22): icmp_seq=3 ttl=64 time=0.141 ms

--- ceph02.novalocal ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2032ms
rtt min/avg/max/mdev = 0.141/0.225/0.338/0.083 ms
[root@ceph01 ~]# ping ceph03
PING ceph03.novalocal (192.168.0.23) 56(84) bytes of data.
64 bytes from ceph03.novalocal (192.168.0.23): icmp_seq=1 ttl=64 time=0.304 ms
64 bytes from ceph03.novalocal (192.168.0.23): icmp_seq=2 ttl=64 time=0.172 ms

--- ceph03.novalocal ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1061ms
rtt min/avg/max/mdev = 0.172/0.238/0.304/0.066 ms
```

步骤 2 关闭防火墙和 SELinux

登录所有主机，使用命令“systemctl disable firewalld”使防火墙停止开机自启动。

并将 selinux 配置为 disabled。

```
# 临时关闭 SELinux
[root@ceph01 ~]# setenforce 0
# 清空 iptables 规则
[root@ceph01 ~]# iptables -F
[root@ceph01 ~]# iptables -t nat -F
# 停止防火墙服务
[root@ceph01 ~]# systemctl stop firewalld
# 禁用防火墙开机自启
[root@ceph01 ~]# systemctl disable firewalld.service
# 永久禁用 SELinux
[root@ceph01 ~]# sudo sed -i 's/^SELINUX=.*$/SELINUX=disabled/' /etc/selinux/config
```

步骤 3 配置时间同步

在所有主机中使用以下命令安装 chrony，并将其设置为开机自启：

```
[root@ceph01 ~]# dnf install -y chrony
[root@ceph01 ~]# systemctl enable chronyd
```

```
[root@ceph01 ~]# systemctl status chronyd
```

查看当前 ECS 的时钟源是否一致。

```
[root@ceph01 ~]# chronyc sources
```

保证一致后再进行后续操作，如下图所示：

```
[root@ceph01 ~]# chronyc sources
MS Name/IP address         Stratum Poll Reach LastRx Last sample
=====
^* 100.125.0.251           4      6   377   55    +11us[+4160ns] +/-   75ms
[root@ceph01 ~]#
```

修改系统信息，防止兼容性问题。

```
[root@ceph01 ~]# cat >/etc/os-release <<'EOF'
NAME="Rocky Linux"
ID=rocky
VERSION="9.3 (Blue Onyx)"
VERSION_ID="9.3"
PLATFORM_ID="platform:el9"
PRETTYNAME="Rocky Linux 9.3 (Blue Onyx)"
ANSI_COLOR="0;32"
CPENAME="cpe:/o:rockyLinux:rocky:9::baseos"
HOME_URL="https://rockyLinux.org/"
BUG_REPORT_URL="https://bugs.rockylinux.org/"
EOF
```

执行后重启 ECS。

```
[root@ceph01 ~]# reboot
```

步骤 4 配置主机免密登录（可选）

在 ceph01 节点生成 SSH 密钥并分发至其他节点，方便集群管理。

```
# 生成 RSA 密钥对，无需密码
[root@ceph01 ~]# ssh-keygen -t rsa -b 4096 -N "" -q
# 将公钥拷贝到 ceph2 节点
[root@ceph01 ~]# ssh-copy-id -i ~/.ssh/id_rsa.pub root@ceph02
# 将公钥拷贝到 ceph3 节点
[root@ceph01 ~]# ssh-copy-id -i ~/.ssh/id_rsa.pub root@ceph03
```

1.3 安装 Ceph

待 ECS 重启完成后，开始安装部署 Ceph 集群。

步骤 1 安装 cephadm

本步骤仅需在一台主机上完成即可，本实验中是在 ceph01 中进行的。

使用以下命令下载 openEuler 可用的 cephadm。

```
[root@ceph01 ~]# dnf install cephadm -y
```

完成后，通过查看 cephadm 帮助信息确定是否安装正确，安装正确的返回信息如下：

```
[root@ceph01 ~]# cephadm --help
usage: cephadm [-h] [--image IMAGE] [--docker] [--data-dir DATA_DIR] [--log-dir LOG_DIR] [--logrotate-dir LOGROTATE_DIR]
               [--sysctl-dir SYSCTL_DIR] [--unit-dir UNIT_DIR] [--verbose] [--timeout TIMEOUT] [--retry RETRY] [--env ENV]
               [--no-container-init] [--no-cgroups-split]
               {version,pull,inspect-image,ls,list-networks,adopt,rm-daemon,rm-cluster,run,shell,enter,ceph-volume,zap-osds,unit,logs,bootstrap,deploy,_orch,check-host,prepare-host,add-repo,rm-repo,install,registry-login,gather-facts,host-maintenance,agent,disk-rescan}
               ...

Bootstrap Ceph daemons with systemd and containers.

.....
```

安装 Ceph 客户端命令工具。

```
[root@ceph01 ~]# yum install ceph-common --nobest -y
```

步骤 2 在 ceph01 上初始化 Ceph 集群

使用以下命令初始化 Ceph 集群：

```
[root@ceph01 ~]# cephadm --image swr.cn-east-3.myhuaweicloud.com/storage/ceph:v18.2.2 bootstrap
--mon-ip 192.168.0.21 --allow-fqdn-hostname --initial-dashboard-user admin --initial-dashboard-password Huawei@123 --dashboard-password-noupdate --skip-monitoring-stack
```

初始化完成后，系统会给出集群的图形化登录方式，如下图所示：

```
Enabling firewalld port 8443/tcp in current zone...
Ceph Dashboard is now available at:

  URL: https://ceph01.novalocal:8443/
  User: admin
  Password: Huawei@123

Enabling client.admin keyring and conf on hosts with "admin" label
You can access the Ceph CLI with:

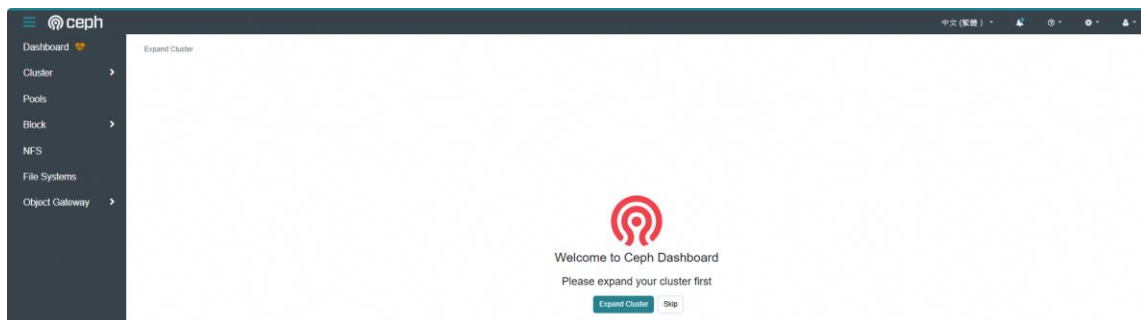
  sudo /usr/sbin/cephadm shell --fsid f0daaa14-55c6-11ee-8452-fa163edb3348 -c /etc/ceph/ceph.conf -k /etc/ceph/ceph.client.admin.keyring

Please consider enabling telemetry to help improve Ceph:

  ceph telemetry on

For more information see:
  https://docs.ceph.com/docs/pacific/mgr/telemetry/
```

使用 ceph01 的弹性 IP 登录 Ceph 的 web 界面，例如 <https://ceph01.弹性IP地址:8443>，账号 admin，密码 Huawei@123，登录后界面如下图所示：



1.4 创建集群

步骤 1 安装官方推荐容器引擎

在所有主机上使用以下命令安装容器引擎和 lvm 组件

```
[root@ceph01 ~]# dnf install podman lvm2 -y
```

*注：在安装过程中，如果有其他互相冲突的软件包已安装在对应的节点上，请先卸载对应的软件包，再继续安装。安装完成后将对应的容器引擎设置为开机启动。

完成后，重启所有 ECS 主机。

步骤 2 为集群添加 node 节点

在 ceph01 节点，使用以下命令生成集群公钥，并将其拷贝到剩余主机：

```
[root@ceph01 ~]# ceph cephadm get-pub-key > ~/ceph.pub
[root@ceph01 ~]# ssh-copy-id -f -i ~/ceph.pub root@ceph02
[root@ceph01 ~]# ssh-copy-id -f -i ~/ceph.pub root@ceph03
```

在添加密钥的过程中，按照提示输入主机的 root 密码，如下图所示：

```
[ceph: root@ceph01 /]# ssh-copy-id -f -i ~/ceph.pub root@ceph03.novalocal
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/ceph.pub"
The authenticity of host 'ceph03.novalocal (192.168.0.23)' can't be established.
ECDSA key fingerprint is SHA256:WA1yHugKw0z1+YrE1jbVsFEbU6k7HwLB/3HbTskpfY.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Authorized users only. All activities may be monitored and reported.
root@ceph03.novalocal's password:
Number of key(s) added: 1
Now try logging into the machine, with: "ssh 'root@ceph03.novalocal'"
and check to make sure that only the key(s) you wanted were added.
```

使用以下命令将全部主机添加到集群内：

```
[root@ceph01 ~]# ceph orch host add ceph02
[root@ceph01 ~]# ceph orch host add ceph03
```

添加完成后，使用以下命令查看当前集群中主机状态：

```
[root@ceph01 ~]# ceph orch host ls
```

```
[root@ceph01 ~]# ceph orch host ls
HOST                ADDR                LABELS             STATUS
ceph01.novalocal    192.168.0.21        _admin
ceph02.novalocal    192.168.0.22
ceph03.novalocal    192.168.0.23
3 hosts in cluster
```

等待系统自动将服务部署到所有节点中（时间较长，请耐心等待）

```
[root@ceph01 ~]# ceph orch ls
```

部署完成后状态如下：

```
[root@ceph01 ~]# ceph orch ls
NAME    PORTS    RUNNING    REFRESHED    AGE    PLACEMENT
crash                3/3    66s ago    86s    *
mgr                2/2    66s ago    86s    count:2
mon                3/5    66s ago    87s    count:5
[root@ceph01 ~]#
```

步骤 3 Ceph 集群初始化

使用以下命令取消 mon 服务的自动扩展功能:

```
[root@ceph01 ~]# ceph orch apply mon --unmanaged=true
```

取消以后, mon 对应的“PLACEMENT”值变为“unmanaged”, 如下图所示:

```
[root@ceph01 ~]# ceph orch ls
NAME    PORTS    RUNNING    REFRESHED    AGE    PLACEMENT
crash                3/3    2m ago    2m    *
mgr                2/2    2m ago    2m    count:2
mon                3/5    2m ago    4s    <unmanaged>
[root@ceph01 ~]#
```

使用以下命令为 ceph02 和 ceph03 添加标签 “_admin”:

```
[root@ceph01 ~]# ceph orch host label add ceph02 _admin
Added label _admin to host ceph02
[root@ceph01 ~]# ceph orch host label add ceph03 _admin
Added label _admin to host ceph03
```

使用以下命令将 mon 和 mgr 组件部署到指定节点中:

```
[root@ceph01 ~]# ceph orch apply mon --placement="label:_admin"
Scheduled mon update...
[root@ceph01 ~]# ceph orch apply mgr --placement="label:_admin"
Scheduled mgr update...
```

```
[root@ceph01 ~]# ceph orch host ls
HOST    ADDR          LABELS    STATUS
ceph01  192.168.0.21  _admin
ceph02  192.168.0.22  _admin
ceph03  192.168.0.23  _admin
3 hosts in cluster
[root@ceph01 ~]# ceph orch ls
NAME    PORTS    RUNNING    REFRESHED    AGE    PLACEMENT
crash                3/3    3m ago    3m    *
mgr                3/3    3m ago    17s    label:_admin
mon                3/3    3m ago    21s    label:_admin
[root@ceph01 ~]#
```

步骤 4 添加 OSD

使用以下命令，将所有主机上的硬盘添加为 OSD：

列出所有可用设备

```
[root@ceph01 ~]# ceph orch device ls
```

手动添加各节点的磁盘作为 OSD。

添加 ceph01 节点的三块数据盘

```
[root@ceph01 ~]# ceph orch daemon add osd ceph01:/dev/sdb
```

```
[root@ceph01 ~]# ceph orch daemon add osd ceph01:/dev/sdc
```

```
[root@ceph01 ~]# ceph orch daemon add osd ceph01:/dev/sdd
```

添加 ceph02 节点的三块数据盘

```
[root@ceph01 ~]# ceph orch daemon add osd ceph02:/dev/sdb
```

```
[root@ceph01 ~]# ceph orch daemon add osd ceph02:/dev/sdc
```

```
[root@ceph01 ~]# ceph orch daemon add osd ceph02:/dev/sdd
```

添加 ceph03 节点的三块数据盘

```
[root@ceph01 ~]# ceph orch daemon add osd ceph03:/dev/sdb
```

```
[root@ceph01 ~]# ceph orch daemon add osd ceph03:/dev/sdc
```

```
[root@ceph01 ~]# ceph orch daemon add osd ceph03:/dev/sdd
```

注意：也可使用自动添加命令（不推荐生产使用）。

```
[root@ceph01 ~]# ceph orch apply osd --all-available-devices
```

验证集群状态。

```
[root@ceph01 ~]# ceph -s
```

查看所有主机上的组件。

```
[root@ceph01 ~]# ceph orch ps
```

```
[root@ceph01 ~]# ceph orch ps
NAME                                HOST                                PORTS                                STATUS                                REFRESHED                                AGE                                MEM USE                                MEM LIM                                VERSION                                IMAGE ID                                CONTAINER ID
alertmanager.ceph01                ceph01.novalocal                  *:9093,9094                        running (102s)                        42s ago                                4h                                15.3M                                -                                0.23.0                                ba2b418f427c                                31bd1fe1167a
crash.ceph01                        ceph01.novalocal                  *:9093,9094                        running (3h)                          42s ago                                4h                                13.2M                                -                                16.2.13                                e08a45948779                                909db008b2aa
crash.ceph02                        ceph02.novalocal                  *:9093,9094                        running (3h)                          42s ago                                3h                                6643k                                -                                16.2.13                                e08a45948779                                6c0b125e4873
crash.ceph03                        ceph03.novalocal                  *:9093,9094                        running (3h)                          42s ago                                3h                                6651k                                -                                16.2.13                                e08a45948779                                1323edd41ee7
grafana.ceph01                     ceph01.novalocal                  *:3000                             running (77m)                        42s ago                                78m                                57.2M                                -                                8.3.5                                dad864ee21e9                                bdc9d3d5827a
mgr.ceph01.novalocal.pmrwv         ceph01.novalocal                  *:9283                             running (3h)                          42s ago                                4h                                522M                                -                                16.2.13                                e08a45948779                                878cda687e90
mgr.ceph02.bmmilba                 ceph02.novalocal                  *:8443,9283                        running (109s)                       42s ago                                109s                                390M                                -                                16.2.13                                e08a45948779                                4cf9c13dc38b
mgr.ceph03.zoarwo                   ceph03.novalocal                  *:8443                             running (78m)                        42s ago                                78m                                393M                                -                                16.2.13                                e08a45948779                                6260ed083108
mon.ceph01.novalocal                ceph01.novalocal                  *:6666                             running (3h)                          42s ago                                4h                                603M                                2048M                                16.2.13                                e08a45948779                                f8ebff2838ff
mon.ceph02                          ceph02.novalocal                  *:6666                             running (78m)                        42s ago                                78m                                185M                                2048M                                16.2.13                                e08a45948779                                f3e4ba55e83
mon.ceph03                          ceph03.novalocal                  *:6666                             running (78m)                        42s ago                                78m                                184M                                2048M                                16.2.13                                e08a45948779                                4a242b7b08f7
node-exporter.ceph01                ceph01.novalocal                  *:9100                             running (78m)                        42s ago                                78m                                13.1M                                -                                1.3.1                                1db0e0931976                                4abee19ee088
node-exporter.ceph02                ceph02.novalocal                  *:9100                             running (78m)                        42s ago                                78m                                13.1M                                -                                1.3.1                                1db0e0931976                                5f9b332d1355
node-exporter.ceph03                ceph03.novalocal                  *:9100                             running (78m)                        42s ago                                78m                                13.1M                                -                                1.3.1                                1db0e0931976                                6dd0d01ce149
osd.0                              ceph02.novalocal                  *:8080                             running (54s)                        42s ago                                54s                                23.7M                                4096M                                16.2.13                                e08a45948779                                febb2d85a36a
osd.1                              ceph03.novalocal                  *:8080                             running (54s)                        42s ago                                54s                                23.6M                                4096M                                16.2.13                                e08a45948779                                0798455f9547
osd.2                              ceph01.novalocal                  *:8080                             running (53s)                        42s ago                                53s                                31.5M                                4096M                                16.2.13                                e08a45948779                                596c177a609
osd.3                              ceph02.novalocal                  *:8080                             running (51s)                        42s ago                                51s                                26.3M                                4096M                                16.2.13                                e08a45948779                                e8f335ee6ffe
osd.4                              ceph03.novalocal                  *:8080                             running (51s)                        42s ago                                51s                                25.3M                                4096M                                16.2.13                                e08a45948779                                df7c5c058c64
osd.5                              ceph01.novalocal                  *:8080                             running (50s)                        42s ago                                50s                                22.7M                                4096M                                16.2.13                                e08a45948779                                af7b24d9801c
osd.6                              ceph02.novalocal                  *:8080                             running (48s)                        42s ago                                48s                                19.2M                                4096M                                16.2.13                                e08a45948779                                7dc281c3943f
osd.7                              ceph03.novalocal                  *:8080                             running (48s)                        42s ago                                48s                                21.8M                                4096M                                16.2.13                                e08a45948779                                2a26a7dc30e9
osd.8                              ceph01.novalocal                  *:8080                             running (47s)                        42s ago                                47s                                18.5M                                4096M                                16.2.13                                e08a45948779                                41f93021e10c
prometheus.ceph02                   ceph02.novalocal                  *:9095                             running (101s)                       42s ago                                77m                                37.3M                                -                                2.33.4                                514e6a882f6e                                64981a041e95
```

2 资源池管理

2.1 实验介绍

2.1.1 关于本实验

本实验将基于上个实验中创建的 Ceph 集群完成资源池的初步操作，如资源池的创建、删除和配置等，并通过测试还原的方式体验快照的功能。

2.1.2 实验目的

本实验通过实践的方式，完成副本池和纠删码池的创建、查看及配置，并完成资源池快照的创建及还原，以加强对存储技术的理解

2.2 副本池创建及配置

本实验将创建一个资源池，并对其进行重命名、配置修改、对象上传下载等操作，然后进行快照创建、查看，通过以上一系列的操作，可掌握基本的资源池管理。

步骤 1 创建副本池 test

使用以下命令创建副本池 test，全部配置使用默认，具体为：

```
[root@ceph01 ~]# ceph osd pool create test  
pool 'test' created
```

创建完成后，使用以下命令查看资源池的详细信息：

```
[root@ceph01 ~]# ceph osd pool ls detail | grep test  
pool 2 'test' replicated size 3 min_size 2 crush_rule 0 object_hash rjenkins pg_num 32 pgp_num 32  
autoscale_mode on last_change 74 flags hashpspool stripe_width 0
```

或者使用以下命令，查看到更详细的资源池信息：

```
[root@ceph01 ~]# ceph osd pool get test all  
size: 3  
min_size: 2  
pg_num: 32  
pgp_num: 32  
crush_rule: replicated_rule  
hashpspool: true  
nodelete: false
```

```
nopgchange: false
nosizechange: false
write_fadvise_dontneed: false
noscrub: false
nodeep-scrub: false
use_gmt_hitset: 1
fast_read: 0
pg_autoscale_mode: on
bulk: false
```

步骤 2 修改资源池

根据官方推荐的范围，PG 数量建议为 256 个，因此使用以下命令将 PG 数调整为 256：

```
[root@ceph01 ~]# ceph osd pool set test pg_num 256
set pool 2 pg_num to 256
```

多次使用命令“ceph osd pool get test all”可以查看到 pg_num 及 pgp_num 的调整过程：

```
[root@ceph01 ~]# ceph osd pool get test all
size: 3
min_size: 2
pg_num: 99
pgp_num: 93
crush_rule: replicated_rule
hashpspool: true
nodelete: false
nopgchange: false
nosizechange: false
write_fadvise_dontneed: false
noscrub: false
nodeep-scrub: false
use_gmt_hitset: 1
fast_read: 0
pg_autoscale_mode: on
bulk: false
```

等待一段时间后，pg_num 数量将调整完成：

```
[root@ceph01 ~]# ceph osd pool get test all
size: 3
min_size: 2
pg_num: 256
pgp_num: 256
crush_rule: replicated_rule
hashpspool: true
nodelete: false
nopgchange: false
nosizechange: false
write_fadvise_dontneed: false
noscrub: false
```

```
nodeep-scrub: false
use_gmt_hitset: 1
fast_read: 0
pg_autoscale_mode: on
bulk: false
```

思考：在进行 pg_num 调整的过程中，是先调整 pgp_num 还是 pg_num？为什么？

回答：先调整 pgp_num。因为 pgp_num 是 OSD 的排列组合，只有完成 pgp_num 的调整后，pg 才能找到对应的 OSD 完成数据存储。

思考：pg_num 在修改为 256 后，会一直稳定在该值不变么？

回答：由于开启了“pg_autoscale_mode”，因此 Ceph 集群还会根据资源池中实际存储的数据量，进行自动选择合适的 pg_num 值。

步骤 3 修改 min_size

默认的 min_size 为 2，表示最少有两个副本可用时数据才能读写，这样保证了数据的一致性，但是牺牲了数据的可用性，使用以下命令将其修改为 1：

```
[root@ceph01 ~]# ceph osd pool set test min_size 1
set pool 2 min_size to 1
```

步骤 4 重命名资源池

将资源池 test 重命名为 test_rbd，具体命令为：

```
[root@ceph01 ~]# ceph osd pool rename test test_rbd
pool 'test' renamed to 'test_rbd'
```

步骤 5 使用 rados 命令进行简单对象操作

将/etc/hosts 文件上传到 test_rbd 中，具体命令为：

```
[root@ceph01 ~]# rados -p test_rbd put hosts /etc/hosts
```

上传完成后，查看资源池中已经存在的对象，具体命令为：

```
[root@ceph01 ~]# rados -p test_rbd ls
hosts
```

步骤 6（可选）查看上传对象所在硬盘

使用以下命令可查看到对象 hosts 所映射的 osd：

```
[root@ceph01 ~]# ceph osd map test_rbd hosts
osdmap e3478 pool 'test_rbd' (2) object 'hosts' -> pg 2.ea1b298e (2.e) -> up ([4,2,3], p4) acting ([4,2,3], p4)
```

输出中[4,2,3]为具体的 OSD id，根据此 id，可查看到对应的硬盘：

```
[root@ceph01 ~]# ceph device ls
```

步骤 7 为资源池创建快照

使用以下命令为资源池 test_rbd 创建快照 snapshot1：

```
[root@ceph01 ~]# ceph osd pool mksnap test_rbd snapshot1
created pool test_rbd snap snapshot1
```

步骤 8 删除对象

使用以下命令删除对象：

```
[root@ceph01 ~]# rados -p test_rbd rm hosts
[root@ceph01 ~]# rados -p test_rbd ls
hosts
[root@ceph01 ~]# rados -p test_rbd rm hosts
error removing test_rbd>hosts: (2) No such file or directory
```

思考：为什么删除对象 hosts 后，在 ls 时还可以看到？

回答：删除对象 hosts 后，Ceph 并不会立刻将其删除，仅是将其标注为“已删除”，虽然 ls 时还可以看到，但是无法对其读写。

步骤 9 使用快照还原对象

使用以下命令确定快照中仍然存在对应对象：

```
[root@ceph01 ~]# rados -p test_rbd -s snapshot1 ls
selected snap 1 'snapshot1'
hosts
```

使用以下命令将快照中对象下载到本地，命名为 hosts-obj：

```
[root@ceph01 ~]# rados -p test_rbd -s snapshot1 get hosts hosts-obj
selected snap 3 'snapshot1'
```

查看对象

```
[root@ceph01 ~]# ls
ceph.pub  hosts-obj
```

使用 rados 的 put 命令将对象上传到资源池中，并同样命名为 hosts：

```
root@ceph01 ~]# rados -p test_rbd put hosts ~/hosts-obj
```

使用 rados 的 get 命令下载对象 hosts，命名为 hosts_test，确保对象恢复正常：

```
[root@ceph01 ~]# rados -p test_rbd get hosts hosts_test
[root@ceph01 ~]# rados -p test_rbd ls
hosts
```

步骤 10 资源池配额设置

使用以下命令设置 test_rbd 最多能上传 5 个对象：

```
[root@ceph01 ~]# ceph osd pool set-quota test_rbd max_objects 5
set-quota max_objects = 5 for pool test_rbd
```

尝试再上传三个对象到资源池 test_rbd 中：

```
[root@ceph01 ~]# rados -p test_rbd put hosts1 /etc/hosts
[root@ceph01 ~]# rados -p test_rbd put hosts2 /etc/hosts
```

```
[root@ceph01 ~]# rados -p test_rbd put hosts3 /etc/hosts
```

在上传第 6 个对象时，系统会一直处于上传状态，同时系统报错如下内容：

```
[root@ceph01 ~]# ceph health detail
HEALTH_WARN 3 failed cephadm daemon(s); 1 pool(s) do not have an application enabled; 1 pool(s) full
[WRN] CEPHADM_FAILED_DAEMON: 3 failed cephadm daemon(s)
    daemon mgr.ceph02.rzahcz on ceph02 is in error state
    daemon ceph-exporter.ceph02 on ceph02 is in error state
    daemon ceph-exporter.ceph03 on ceph03 is in error state
[WRN] POOL_APP_NOT_ENABLED: 1 pool(s) do not have an application enabled
    application not enabled on pool 'test_rbd'
    use 'ceph osd pool application enable <pool-name> <app-name>', where <app-name> is 'cephfs',
    'rbd', 'rgw', or freeform for custom applications.
[WRN] POOL_FULL: 1 pool(s) full
    pool 'test_rbd' is full (running out of quota)
```

步骤 11 删除资源池

开启资源池允许删除选项，具体命令为：

```
[root@ceph01 ~]# ceph config set mon mon_allow_pool_delete true
```

使用以下命令删除资源池 test_rbd：

```
[root@ceph01 ~]# ceph osd pool rm test_rbd test_rbd --yes-i-really-really-mean-it
pool 'test_rbd' removed
[root@ceph01 ~]# ceph osd pool ls
.mgr
```

2.3 纠删码池创建

步骤 1 创建纠删码规则

使用以下命令创建纠删码规则 test，并将数据块数量设置为 4，校验块数量设置为 2，为了避免报错，在本实验环境中，建议将故障域设置为 osd，具体命令如下：

```
[root@ceph01 ~]# ceph osd erasure-code-profile set test k=4 m=2 crush-failure-domain=osd
```

步骤 2 创建纠删码池

利用创建好的纠删码规则创建纠删码池，具体命令如下：

```
[root@ceph01 ~]# ceph osd pool create object_test erasure test
pool 'object_test' created
[root@ceph01 ~]# ceph osd pool get object_test all
size: 6
min_size: 5
pg_num: 1
pgp_num: 1
```

```
crush_rule: object_test
hashpspool: true
allow_ec_overwrites: false
nodelete: false
npgchange: false
nosizechange: false
write_fadvise_dontneed: false
noscrub: false
nodeep-scrub: false
use_gmt_hitset: 1
erasure_code_profile: test
fast_read: 0
pg_autoscale_mode: on
eio: false
bulk: false
```

步骤 3 设置资源池类型

将创建好的资源池类型设置为 rgw，具体命令如下：

```
[root@ceph01 ~]# ceph osd pool application enable object_test rgw
enabled application 'rgw' on pool 'object_test'
```

3 Ceph 用户管理实践

3.1 实验介绍

3.1.1 关于本实验

从本实验开始，所有对 Ceph 集群的操作将在 Client 上完成，为了 Client 有对 Ceph 的管理权限，本实验将创建用户 hci_storage，并完成相关操作，使 Client 具备相关操作。

3.1.2 实验目的

本实验通过用户创建、用户赋权及认证相关文件的操作，以加强对 Ceph 配置操作的理解。

3.2 用户创建及赋权

步骤 1 创建用户

使用以下命令创建用户 hci_storage：

```
[root@ceph01 ~]# ceph auth add client.hci_storage
added key for client.hci_storage
[root@ceph01 ~]# ceph auth get client.hci_storage
[client.hci_storage]
key = AQCzBBRqwmh8OhAAu6bcztF9Yv4caAh4hGJ13w==
```

步骤 2 为新创用户赋权

新创用户 hci_storage 需要对集群有管理权限，因此，需要对其赋予 mon 的读写权限，具体命令为：

```
[root@ceph01 ~]# ceph auth caps client.hci_storage mon 'allow rw'
updated caps for client.hci_storage
[root@ceph01 ~]# ceph auth get client.hci_storage
[client.hci_storage]
key = AQCzBBRqwmh8OhAAu6bcztF9Yv4caAh4hGJ13w==
caps mon = "allow rw"
```

步骤 3 安装客户端软件

使用以下命令在 Client 中安装客户端软件：

```
[root@client ~]# dnf install -y ceph-2:18.2.2-10.oe2403.x86_64
```

步骤 4 导出认证文件

将用户 hciip_storage 的 keyring 导出，并将其拷贝到 Client 主机中，具体命令如下：

```
[root@ceph01 ~]# ceph auth get client.hciip_storage -o client.hciip_storage.keyring
[root@ceph01 ~]# scp client.hciip_storage.keyring root@192.168.0.24:/etc/ceph/
The authenticity of host '192.168.0.24 (192.168.0.24)' can't be established.
ED25519 key fingerprint is SHA256:hLpwRBojc508x7ViDn57QZgO9sRGprtM9V7IIUS302U.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.0.24' (ED25519) to the list of known hosts.

Authorized users only. All activities may be monitored and reported.
root@192.168.0.24's password:
client.hciip_storage.keyring
```

再将 Ceph 集群的配置文件也拷贝到 Client 主机中，具体命令如下：

```
[root@ceph01 ~]# scp /etc/ceph/ceph.conf root@192.168.0.24:/etc/ceph/
```

步骤 5 测试

在客户端进行测试，是否能够查看到 Ceph 集群的信息，具体命令如下：

```
[root@client ~]# ceph -s --id hciip_storage --keyring /etc/ceph/client.hciip_storage.keyring
cluster:
  id:      37511d4e-57ee-11f1-b5ff-fa163e4da5ba
  health: HEALTH_WARN
          3 failed cephadm daemon(s)

services:
  mon: 3 daemons, quorum ceph01,ceph02,ceph03 (age 89m)
  mgr: ceph01.gzuzln(active, since 4h), standbys: ceph03.bfvrvx
  osd: 9 osds: 9 up (since 89m), 9 in (since 89m)

data:
  pools:   2 pools, 33 pgs
  objects: 2 objects, 577 KiB
  usage:   648 MiB used, 89 GiB / 90 GiB avail
  pgs:     33 active+clean
```

- 思考一：在 Client 中操作为什么需要指定用户名？
- 回答：如果不指定具体的用户名，系统会默认使用 admin 进行集群访问，而当前 Client 中不存在相关的认证文件，因此系统会报错，具体如下图所示：

```
[root@client ~]# ceph -s
2023-10-27T03:16:48.960+0000 7f01b14e2640 -1 auth: unable to find a keyring on /etc/ceph/ceph.client.admin.keyring,/etc/ceph/ceph.keyring,/
2023-10-27T03:16:48.960+0000 7f01b14e2640 -1 AuthRegistry(0x7f01ac05eaa0) no keyring found at /etc/ceph/ceph.client.admin.keyring,/etc/ceph/
2023-10-27T03:16:48.960+0000 7f01b14e2640 -1 auth: unable to find a keyring on /etc/ceph/ceph.client.admin.keyring,/etc/ceph/ceph.keyring,/
2023-10-27T03:16:48.960+0000 7f01b14e2640 -1 AuthRegistry(0x7f01b14e10d0) no keyring found at /etc/ceph/ceph.client.admin.keyring,/etc/ceph/
2023-10-27T03:16:48.961+0000 7f01ab7fe640 -1 monclient(hunting): handle_auth_bad_method server allowed_methods [2] but i only support [1]
2023-10-27T03:16:48.961+0000 7f01aa7fc640 -1 monclient(hunting): handle_auth_bad_method server allowed_methods [2] but i only support [1]
2023-10-27T03:16:48.961+0000 7f01aaffd640 -1 monclient(hunting): handle_auth_bad_method server allowed_methods [2] but i only support [1]
2023-10-27T03:16:48.961+0000 7f01b14e2640 -1 monclient: authenticate NOTE: no keyring found; disabled cephx authentication
[errno 13] RADOS permission denied (error connecting to the cluster)
[root@client ~]# ceph -s --id hciip_storage --keyring /etc/ceph/
```

- 思考二：以上的操作是否可以证明用户 `hcid_storage` 对 Ceph 集群具备了管理权限？
 - 回答：不能，以上的操作仅能说明该用户对 Ceph 集群有可读权限，可查看 Ceph 状态等信息，后续对 Ceph 进行配置管理后，才能证明该用户有管理权限。
-

4 Ceph 对象存储实践

4.1 实验介绍

4.1.1 关于本实验

本实验将使用 swift 和 ceph-rgw 对接，完成对象的上传下载等操作，并进行多站点的配置以实现对象存储容灾配置。

4.1.2 实验目的

本实验将通过对象存储网关的部署、对象的上传下载、对象网关的多站点部署等操作，以加强对 Ceph 内容的理解。

4.2 创建对象存储网关

创建对象存储网关，并把相关的 realm、zone 等相关资源全部定义为 “for_swift”。

步骤 1 创建 realm

使用以下命令创建 realm for_swift:

```
[root@ceph01 ~]# radosgw-admin realm create --rgw-realm=for_swift --default
{
  "id": "76fa9b3a-68fa-460c-baf5-7e9d4337b201",
  "name": "for_swift",
  "current_period": "dd346702-abe0-4044-815a-dbf7fd6e7f48",
  "epoch": 1
}
```

步骤 2 创建 zonegroup

使用以下命令创建 zonegroup for_swift，并将其设置为 master:

```
[root@ceph01 ~]# radosgw-admin zonegroup create --rgw-realm=for_swift --rgw-zonegroup=for_swift -
-master --default
{
  "id": "1448adbb-df90-4bde-846e-f580ecc979c8",
  "name": "for_swift",
  "api_name": "for_swift",
}
```

```
"is_master": true,
"endpoints": [],
"hostnames": [],
"hostnames_s3website": [],
"master_zone": "",
"zones": [],
"placement_targets": [
  {
    "name": "default-placement",
    "tags": [],
    "storage_classes": []
  }
],
"default_placement": "default-placement",
"realm_id": "76fa9b3a-68fa-460c-baf5-7e9d4337b201",
"sync_policy": {
  "groups": []
},
"enabled_features": [
  "resharding"
]
}
```

步骤 3 创建 zone

使用以下命令创建 zone for_swift:

```
[root@ceph01 ~]# radosgw-admin zone create --rgw-realm=for_swift --rgw-zonegroup=for_swift --rgw-
zone=for_swift --master --default
NOTICE: set zone's realm_id=76fa9b3a-68fa-460c-baf5-7e9d4337b201
{
  "id": "9c1affb6-8fbb-4f1a-987a-90db86cbfece",
  "name": "for_swift",
  "domain_root": "for_swift.rgw.meta:root",
  "control_pool": "for_swift.rgw.control",
  "gc_pool": "for_swift.rgw.log:gc",
  "lc_pool": "for_swift.rgw.log:lc",
  "log_pool": "for_swift.rgw.log",
  "intent_log_pool": "for_swift.rgw.log:intent",
  "usage_log_pool": "for_swift.rgw.log:usage",
  "roles_pool": "for_swift.rgw.meta:roles",
  "reshard_pool": "for_swift.rgw.log:reshard",
  "user_keys_pool": "for_swift.rgw.meta:users.keys",
  "user_email_pool": "for_swift.rgw.meta:users.email",
  "user_swift_pool": "for_swift.rgw.meta:users.swift",
  "user_uid_pool": "for_swift.rgw.meta:users.uid",
  "otp_pool": "for_swift.rgw.otp",
  "system_key": {
    "access_key": "",
```

```

    "secret_key": ""
  },
  "placement_pools": [
    {
      "key": "default-placement",
      "val": {
        "index_pool": "for_swift.rgw.buckets.index",
        "storage_classes": {
          "STANDARD": {
            "data_pool": "for_swift.rgw.buckets.data"
          }
        },
        "data_extra_pool": "for_swift.rgw.buckets.non-ec",
        "index_type": 0,
        "inline_data": true
      }
    }
  ],
  "realm_id": "76fa9b3a-68fa-460c-baf5-7e9d4337b201",
  "notif_pool": "for_swift.rgw.log:notif"
}

```

使用以下命令使配置生效，具体如下：

```

[root@ceph01 ~]# radosgw-admin period update --rgw-realm=for_swift --commit
20XX-05-25T16:33:36.730+0800 7f5b69b6bcc0 0 period (dd346702-abe0-4044-815a-dbf7fd6e7f48 does
not have zone 9c1affb6-8fbb-4f1a-987a-90db86cbfece configured
{
  "id": "d7c6cdb3-5389-4137-b104-9dea86630e4d",
  "epoch": 1,
  "predecessor_uuid": "dd346702-abe0-4044-815a-dbf7fd6e7f48",
  "sync_status": [],
  "period_map": {
    "id": "d7c6cdb3-5389-4137-b104-9dea86630e4d",
    "zonegroups": [
      {
        "id": "1448adbb-df90-4bde-846e-f580ecc979c8",
        "name": "for_swift",
        "api_name": "for_swift",
        "is_master": true,
        "endpoints": [],
        "hostnames": [],
        "hostnames_s3website": [],
        "master_zone": "9c1affb6-8fbb-4f1a-987a-90db86cbfece",
        "zones": [
          ...

```

步骤 4 创建 RGW

使用以下命令创建 rgw，并和已创建的 for_swift 进行关联，数量为 2：

```
[root@ceph01 ~]# ceph orch apply rgw for_swift --realm=for_swift --zone=for_swift --placement="2" --port=8080
```

步骤 5 创建用户

创建用户 swift，具体命令如下：

```
[root@ceph01 ~]# radosgw-admin user create --uid=swift --display-name=swift
{
  "user_id": "swift",
  "display_name": "swift",
  "email": "",
  "suspended": 0,
  "max_buckets": 1000,
  "subusers": [],
  "keys": [
    {
      "user": "swift",
      "access_key": "85W8400X40R0ZS6FHQZH",
      "secret_key": "R3GPps39Y1GyeX82iUbB0sKuAF9biak9b3CJieoO"
    }
  ],
  "swift_keys": [],
  "caps": [],
  "op_mask": "read, write, delete",
  "default_placement": "",
  "default_storage_class": "",
  "placement_tags": [],
  "bucket_quota": {
    "enabled": false,
    "check_on_raw": false,
    "max_size": -1,
    "max_size_kb": 0,
    "max_objects": -1
  },
  "user_quota": {
    "enabled": false,
    "check_on_raw": false,
    "max_size": -1,
    "max_size_kb": 0,
    "max_objects": -1
  },
  "temp_url_keys": [],
  "type": "rgw",
  "mfa_ids": []
}
```

创建子账号 swift_rgw，具体命令如下：

```
[root@ceph01 ~]# radosgw-admin subuser create --uid=swift --subuser=swift:swift_rgw --access=full
{
  "user_id": "swift",
  "display_name": "swift",
  "email": "",
  "suspended": 0,
  "max_buckets": 1000,
  "subusers": [
    {
      "id": "swift:swift_rgw",
      "permissions": "full-control"
    }
  ],
  "keys": [
    {
      "user": "swift",
      "access_key": "85W8400X40R0ZS6FHQZH",
      "secret_key": "R3GPps39Y1GyeX82iUbB0sKuAF9biak9b3CJieoO"
    }
  ],
  "swift_keys": [
    {
      "user": "swift:swift_rgw",
      "secret_key": "tRLwrCsCMbSjIGp8beUw6mTI2hlInXbtCJKFRy9v"
    }
  ],
  "caps": [],
  "op_mask": "read, write, delete",
  "default_placement": "",
  "default_storage_class": "",
  "placement_tags": [],
  "bucket_quota": {
    "enabled": false,
    "check_on_raw": false,
    "max_size": -1,
    "max_size_kb": 0,
    "max_objects": -1
  },
  "user_quota": {
    "enabled": false,
    "check_on_raw": false,
    "max_size": -1,
    "max_size_kb": 0,
    "max_objects": -1
  },
  "temp_url_keys": [],
```

```
}  
  "type": "rgw",  
  "mfa_ids": []  
}
```

为创建的子账号生成 swift 可用的 key，具体命令如下：

```
[root@ceph01 ~]# radosgw-admin key create --subuser=swift:swift_rgw --key-type=swift --gen-secret  
{  
  "user_id": "swift",  
  "display_name": "swift",  
  "email": "",  
  "suspended": 0,  
  "max_buckets": 1000,  
  "subusers": [  
    {  
      "id": "swift:swift_rgw",  
      "permissions": "full-control"  
    }  
  ],  
  "keys": [  
    {  
      "user": "swift",  
      "access_key": "85W8400X40R0ZS6FHQZH",  
      "secret_key": "R3GPps39Y1GyeX82iUbB0sKuAF9biak9b3CJieoO"  
    }  
  ],  
  "swift_keys": [  
    {  
      "user": "swift:swift_rgw",  
      "secret_key": "L9BTtzP0onjZOAchVOvCehcrl7O45l8DgryEkc7z"  
    }  
  ],  
  "caps": [],  
  "op_mask": "read, write, delete",  
  "default_placement": "",  
  "default_storage_class": "",  
  "placement_tags": [],  
  "bucket_quota": {  
    "enabled": false,  
    "check_on_raw": false,  
    "max_size": -1,  
    "max_size_kb": 0,  
    "max_objects": -1  
  },  
  "user_quota": {  
    "enabled": false,  
    "check_on_raw": false,  
    "max_size": -1,  
    "max_size_kb": 0,
```

```
"max_objects": -1
},
"temp_url_keys": [],
"type": "rgw",
"mfa_ids": []
}
```

配置完成后，记录下生成的 Key，后续会使用该值进行认证。

4.3 客户端操作

步骤 1 客户端 swift 安装

使用以下命令在 Client 中安装组件：

```
[root@client ~]# dnf install -y python-setuptools python-pip
```

使用 pip 安装 Swift 客户端。

```
[root@client ~]# pip install --upgrade python-swiftclient
DEPRECATION: Loading egg at /usr/local/lib/python3.11/site-packages/cloud_init-19.1-py3.11.egg is
deprecated. pip 24.3 will enforce this behaviour change. A possible replacement is to use pip for package
installation.. Discussion can be found at https://github.com/pypa/pip/issues/12330
Collecting python-swiftclient
  Downloading python_swiftclient-4.10.0-py3-none-any.whl.metadata (3.9 kB)
Requirement already satisfied: requests>=2.4.0 in /usr/local/lib/python3.11/site-packages (from python-
swiftclient) (2.32.5)
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib64/python3.11/site-packages
(from requests>=2.4.0->python-swiftclient) (3.4.4)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/site-packages (from
requests>=2.4.0->python-swiftclient) (3.11)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/site-packages (from
requests>=2.4.0->python-swiftclient) (2.5.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/site-packages (from
requests>=2.4.0->python-swiftclient) (2025.10.5)
Downloading python_swiftclient-4.10.0-py3-none-any.whl (88 kB)
----- 88.6/88.6 kB
69.6 kB/s eta 0:00:00
Installing collected packages: python-swiftclient
Successfully installed python-swiftclient-4.10.0
```

使用以下命令检查客户端是否安装成功：

```
[root@client ~]# swift --version
python-swiftclient 4.10.0
```

步骤 2 用户认证

使用 4.2 步骤 5 中创建的用户连接 Ceph，查看用户名和 Key 是否可用（下面的 Key 值要替换成创建用户时生成的值），具体命令如下：

```
[root@client ~]# swift -A http://192.168.0.21:8080/auth/1.0 -U "swift:swift_rgw" -K
L9BTtzP0onjZOAChVOvCehcrI7O45l8DgryEkc7z stat -v --insecure
StorageURL: http://192.168.0.21:8080/swift/v1
Auth Token:
AUTH_rgwtk0f00000073776966743a73776966745f72677797417bd55e9b6e36dc5c156a5f6fe112e8c35e1
2fc29ebc4b2a617aa2c85bd487e3c8317
Account: v1
Containers: 0
Objects: 0
Bytes: 0
Objects in policy "default-placement-bytes": 0
Bytes in policy "default-placement-bytes": 0
Containers in policy "default-placement": 0
Objects in policy "default-placement": 0
Bytes in policy "default-placement": 0
X-Timestamp: 1779698524.32120
X-Account-Bytes-Used-Actual: 0
X-Trans-Id: tx00000839f1c2cf76b7e8f-006a140b5c-39e4-for_swift
X-Openstack-Request-Id: tx00000839f1c2cf76b7e8f-006a140b5c-39e4-for_swift
Accept-Ranges: bytes
Content-Type: text/plain; charset=utf-8
Connection: Keep-Alive
```

步骤 3 使用 swift 进行基本的对象操作

使用 swift 创建容器（桶）test，具体命令如下：

```
[root@client ~]# swift -A http://192.168.0.21:8080/auth/1.0 -U "swift:swift_rgw" -K
L9BTtzP0onjZOAChVOvCehcrI7O45l8DgryEkc7z post test --insecure
```

创建完成后，使用以下命令查看创建的容器（桶）：

```
[root@client ~]# swift -A http://192.168.0.21:8080/auth/1.0 -U "swift:swift_rgw" -K
L9BTtzP0onjZOAChVOvCehcrI7O45l8DgryEkc7z list --insecure
test
```

使用以下命令将 Client 上的/etc/hosts 文件上传到容器（桶）test 中，命名为 hosts：

```
[root@client ~]# swift -A http://192.168.0.21:8080/auth/1.0 -U "swift:swift_rgw" -K
L9BTtzP0onjZOAChVOvCehcrI7O45l8DgryEkc7z upload --object-name hosts test /etc/hosts -v --insecure
hosts
```

上传完成后，使用以下命令确定是否上传完成：

```
[root@client ~]# swift -A http://192.168.0.21:8080/auth/1.0 -U "swift:swift_rgw" -K
L9BTtzP0onjZOAChVOvCehcrI7O45l8DgryEkc7z list test -v --insecure
hosts
```

4.4 桶配额限制

桶的配额可以针对用户，也可以针对指定的桶，本实验将分别进行实践操作。

步骤 1 针对用户进行配额限制

使用以下命令启用对用户 swift 的配额限制：

```
[root@ceph01 ~]# radosgw-admin quota enable --quota-scope=user --uid=swift
```

使用以下命令将用户 swift 可上传的对象数量设置为不超过 3：

```
[root@ceph01 ~]# radosgw-admin quota set --quota-scope=user --uid=swift --max-objects=3
```

设置完成后，使用以下命令查看配置是否生效：

```
[root@ceph01 ~]# radosgw-admin user info --uid=swift
{
  "user_id": "swift",
  "display_name": "swift",
  "email": "",
  "suspended": 0,
  "max_buckets": 1000,
  "subusers": [
    {
      "id": "swift:swift_rgw",
      "permissions": "full-control"
    }
  ],
  "keys": [
    {
      "user": "swift",
      "access_key": "85W8400X40R0ZS6FHQZH",
      "secret_key": "R3GPps39Y1GyeX82iUbB0sKuAF9biak9b3CJieoO"
    }
  ],
  "swift_keys": [
    {
      "user": "swift:swift_rgw",
      "secret_key": "L9BTtzP0onjZOAchVOvCehcrI7O45l8DgryEkc7z"
    }
  ],
  "caps": [],
  "op_mask": "read, write, delete",
  "default_placement": "",
  "default_storage_class": "",
  "placement_tags": [],
  "bucket_quota": {
    "enabled": false,
    "check_on_raw": false,
    "max_size": -1,
```

```

        "max_size_kb": 0,
        "max_objects": -1
    },
    "user_quota": {
        "enabled": true,
        "check_on_raw": false,
        "max_size": -1,
        "max_size_kb": 0,
        "max_objects": 3
    },
    "temp_url_keys": [],
    "type": "rgw",
    "mfa_ids": []
}

```

在客户端使用上传命令进行测试，当上传第三个时，客户端会收到报错，具体如下：

```

[root@client ~]# swift -A http://192.168.0.21:8080/auth/1.0 -U "swift:swift_rgw" -K
L9BTtzP0onjZOAChVOvCehcrI7O45l8DgryEk7z upload --object-name hosts test /etc/hosts -v --insecure
hosts
[root@client ~]# swift -A http://192.168.0.21:8080/auth/1.0 -U "swift:swift_rgw" -K
L9BTtzP0onjZOAChVOvCehcrI7O45l8DgryEk7z upload --object-name hosts1 test /etc/hosts -v --
insecure
hosts1
[root@client ~]# swift -A http://192.168.0.21:8080/auth/1.0 -U "swift:swift_rgw" -K
L9BTtzP0onjZOAChVOvCehcrI7O45l8DgryEk7z upload --object-name hosts2 test /etc/hosts -v --
insecure
hosts2
[root@client ~]# swift -A http://192.168.0.21:8080/auth/1.0 -U "swift:swift_rgw" -K
L9BTtzP0onjZOAChVOvCehcrI7O45l8DgryEk7z upload --object-name hosts3 test /etc/hosts -v --
insecure
Object PUT failed: http://192.168.0.21:8080/swift/v1/test/hosts3 413 Request Entity Too Large
b'QuotaExceeded' (txn: tx000003eb1f5392fee12ba-006a141217-39e4-for_swift)
Consider using the --segment-size option to chunk the object

```

将 “--max-objects” 设置为-1 可取消对象数量配额的限制，具体命令如下：

```

[root@ceph01 ~]# radosgw-admin quota set --quota-scope=user --uid=swift --max-objects=-1

```

修改完成后，在客户端重新上传对象，则会顺利上传完成，具体如下：

```

[root@client ~]# swift -A http://192.168.0.21:8080/auth/1.0 -U "swift:swift_rgw" -K
L9BTtzP0onjZOAChVOvCehcrI7O45l8DgryEk7z upload --object-name hosts3 test /etc/hosts --insecure
hosts3

```

- 思考：除了设置对象数量外，是否还有其他的配额限制方式？
- 回答：有，除了对象数量，还可以最大上传的容量进行限制，具体方式将在对桶进行限额限制中演示

步骤 2 针对桶进行配额限制

使用以下命令启用对桶 test 的配额限制：

```
[root@ceph01 ~]# radosgw-admin quota enable --quota-scope=bucket --bucket=test
```

使用以下命令将桶 test 的最大容量设置为 10M:

```
[root@ceph01 ~]# radosgw-admin quota set --quota-scope=bucket --bucket=test --max-size=10M
```

配置完成后, 查看配置是否生效:

```
[root@ceph01 ~]# radosgw-admin bucket stats
[
  {
    "bucket": "test",
    "num_shards": 11,
    "tenant": "",
    "zonegroup": "1448adbb-df90-4bde-846e-f580ecc979c8",
    "placement_rule": "default-placement",
    "explicit_placement": {
      "data_pool": "",
      "data_extra_pool": "",
      "index_pool": ""
    },
    "id": "9c1affb6-8fbb-4f1a-987a-90db86cbfece.14826.1",
    "marker": "9c1affb6-8fbb-4f1a-987a-90db86cbfece.14826.1",
    "index_type": "Normal",
    "versioned": false,
    "versioning_enabled": false,
    "object_lock_enabled": false,
    "mfa_enabled": false,
    "owner": "swift",
    "ver": "0#2,1#1,2#1,3#2,4#1,5#2,6#1,7#1,8#2,9#1,10#1",
    "master_ver": "0#0,1#0,2#0,3#0,4#0,5#0,6#0,7#0,8#0,9#0,10#0",
    "mtime": "2026-05-25T09:13:51.055320Z",
    "creation_time": "2026-05-25T09:01:33.989360Z",
    "max_marker": "0#,1#,2#,3#,4#,5#,6#,7#,8#,9#,10#",
    "usage": {
      "rgw.main": {
        "size": 1612,
        "size_actual": 16384,
        "size_utilized": 1612,
        "size_kb": 2,
        "size_kb_actual": 16,
        "size_kb_utilized": 2,
        "num_objects": 4
      }
    },
    "bucket_quota": {
      "enabled": true,
      "check_on_raw": true,
      "max_size": 10485760,
      "max_size_kb": 10240,
```

```
        "max_objects": -1
    }
}
]
```

在客户端创建一个 6M 的文件 file，进行测试，具体命令为：

```
[root@client ~]# dd if=/dev/zero of=file bs=1M count=6
```

将文件 file 上传到容器（桶）test 中，当上传第二个时，系统会返回报错，具体如下：

```
[root@client ~]# swift -A http://192.168.0.21:8080/auth/1.0 -U "swift:swift_rgw" -K
L9BTtzP0onjZOAChVOvCehcrI7O45l8DgryEk7z upload --object-name file1 test /root/file -v --insecure
file1
[root@client ~]# swift -A http://192.168.0.21:8080/auth/1.0 -U "swift:swift_rgw" -K
L9BTtzP0onjZOAChVOvCehcrI7O45l8DgryEk7z upload --object-name file2 test /root/file -v --insecure
Object PUT failed: http://192.168.0.21:8080/swift/v1/test/file2 413 Request Entity Too Large
b'QuotaExceeded' (txn: tx00000e2ada875636a4b77-006a141354-39e4-for_swift)
Consider using the --segment-size option to chunk the object
```

- 任务：自行设计实验取消对桶容量的配额限制。
- 参考：radosgw-admin quota set --quota-scope=bucket --bucket=test --max-size=-1

5 Ceph 块存储 RBD 实践

5.1 实验介绍

5.1.1 关于本实验

本实验将在 Ceph 中创建 RBD 类型的资源池，并以块设备的方式挂载到客户端节点，并在客户端完成格式化、数据读写等操作。

5.1.2 实验目的

本实验主要针对块设备的使用，包括创建、挂载等操作，以加强对 RBD 知识的理解。

5.2 创建块设备并挂载给客户机

本实验将创建一个 RBD 类型的资源池 block，并创建两个 rbd 设备，分别命名为 rbd1 和 rbd2，容量分别是 3G 和 5G，并将其挂载到客户机的/mnt/rbd1 和/mnt/rbd2 目录下。

步骤 1 创建 RBD 资源池

使用以下命令创建资源池 block，并将其设置为 rbd 类型：

```
[root@ceph01 ~]# ceph osd pool create block
[root@ceph01 ~]# ceph osd pool application enable block rbd
```

步骤 2 创建 rbd 设备

使用以下命令按照前期规划创建两个 rbd 设备：

```
[root@ceph01 ~]# rbd create --size 3G block/rbd1
[root@ceph01 ~]# rbd create --size 5G block/rbd2
[root@ceph01 ~]# rbd info block/rbd1
rbd image 'rbd1':
    size 3 GiB in 768 objects
    order 22 (4 MiB objects)
    snapshot_count: 0
    id: 3b28d2af918b
    block_name_prefix: rbd_data.3b28d2af918b
    format: 2
    features: layering, exclusive-lock, object-map, fast-diff, deep-flatten
```

```
op_features:
flags:
create_timestamp: Mon May 25 17:31:07 2026
access_timestamp: Mon May 25 17:31:07 2026
modify_timestamp: Mon May 25 17:31:07 2026
```

步骤 3 创建用户

创建用户 block，使其对所有的 rbd 设备有读写权限：

```
[root@ceph01 ~]# ceph auth add client.block mon 'allow r' osd 'allow rwx pool=block'
added key for client.block
```

将用户 block 的 keyring 文件导出为 ceph.client.block.keyring，具体命令为：

```
[root@ceph01 ~]# ceph auth get client.block -o ceph.client.block.keyring
[root@ceph01 ~]# cat ceph.client.block.keyring
[client.block]
key = AQD0FxRqlu9WDBAAIsKOY0OBu0m3Lz3RTh+5XA==
caps mon = "allow r"
caps osd = "allow rwx pool=block"
```

将该文件发送到客户端的/etc/ceph 目录中，具体操作如下：

```
[root@ceph01 ~]# scp ceph.client.block.keyring root@192.168.0.24:/etc/ceph/

Authorized users only. All activities may be monitored and reported.
root@192.168.0.24's password:
ceph.client.block.keyring
```

步骤 4 客户端挂载

在 Client 端，使用 rbd 命令将创建好的 rbd 设备映射到 Client，具体命令为：

```
[root@client ~]# rbd map block/rbd1 --name client.block
/dev/rbd0
[root@client ~]# rbd map block/rbd2 --name client.block
/dev/rbd1
```

映射完成后，查看是否可以查看到对应的设备：

```
[root@client ~]# lsblk
NAME        MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS
sda          8:0    0  40G  0 disk
├─sda1       8:1    0   1M  0 part
└─sda2       8:2    0  40G  0 part /
rbd0        253:0    0   3G  0 disk
rbd1        253:16   0   5G  0 disk
```

将映射后的设备格式化，并按照规划，挂载到指定目录下，具体操作如下：

```
[root@client ~]# mkdir -p /mnt/rbd{1,2}
[root@client ~]# mkfs.xfs /dev/rbd0
[root@client ~]# mkfs.xfs /dev/rbd1
```

```
[root@client ~]# mount /dev/rbd0 /mnt/rbd1
[root@client ~]# mount /dev/rbd1 /mnt/rbd2
```

挂载完成后，查看目录信息为：

```
[root@client ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda2       40G   4.5G   34G   12% /
devtmpfs        4.0M    0   4.0M    0% /dev
tmpfs           1.7G    0   1.7G    0% /dev/shm
tmpfs           4.0M    0   4.0M    0% /sys/fs/cgroup
tmpfs           657M   8.6M   649M    2% /run
tmpfs           1.7G   4.0K   1.7G    1% /tmp
/dev/rbd0       3.0G   91M   2.9G    4% /mnt/rbd1
/dev/rbd1       5.0G  130M   4.9G    3% /mnt/rbd2
```

步骤 5 将 rbd 设备设置为自动挂载

编辑/etc/fstab，添加以下内容：

```
[root@client ~]# vim /etc/fstab

/dev/rbd/block/rbd1      /mnt/rbd1      xfs      defaults,_netdev 0 0
/dev/rbd/block/rbd2      /mnt/rbd2      xfs      defaults,_netdev 0 0
```

```
UUID=ae2cd454-a1aa-4683-8c9d-711a1d610f40 / ext4 defaults 0 0
UUID=5930-B586 /boot vfat defaults 0 0
/dev/rbd/block/rbd1      /mnt/rbd1      xfs      defaults,_netdev 0 0
/dev/rbd/block/rbd2      /mnt/rbd2      xfs      defaults,_netdev 0 0
```

编辑/etc/ceph/rbdmap 文件，添加以下内容：

```
[root@client ~]# vim /etc/ceph/rbdmap

block/rbd1      id=block,keyring=/etc/ceph/ceph.client.block.keyring
block/rbd2      id=block,keyring=/etc/ceph/ceph.client.block.keyring
```

```
# RbdDevice      Parameters
#poolname/imagename id=client,keyring=/etc/ceph/ceph.client.keyring
block/rbd1      id=block,keyring=/etc/ceph/ceph.client.block.keyring
block/rbd2      id=block,keyring=/etc/ceph/ceph.client.block.keyring
```

最后将 rbdmap 服务设置为开机自启动：

```
[root@client ~]# systemctl enable rbdmap
Created symlink /etc/systemd/system/multi-user.target.wants/rbdmap.service →
/usr/lib/systemd/system/rbdmap.service.
```

请检查 rbd 设备是否有互斥锁，如果有，请根据实际回显删除再重新挂载，否则开机无法自动挂载。以下命令仅给出 rbd1 的操作方法，rbd2 请自行检查。

```
[root@ceph01 ~]# rbd lock list block/rbd1
There is 1 exclusive lock on this image.
Locker      ID      Address
client.34232 auto 18446462598732840961 192.168.0.24:0/3633730909
```

```
[root@ceph01 ~]# rbd lock rm block/rbd1 "auto 18446462598732840961" "client.34232"
```

自行重启测试 Client 是否可以开机自动挂载两个 rbd 设备。

5.3 RBD 设备扩容

本实验会将 5.2 中创建的 rbd1 扩容到 5G。

步骤 1 将 rbd1 扩容至 5G

使用以下命令将 rbd1 扩容至 5G：

```
[root@ceph01 ~]# rbd --pool block resize rbd1 --size 5G
Resizing image: 100% complete...done.
```

扩容完成后，检查是否生效，具体命令如下：

```
[root@ceph01 ~]# rbd --pool block du rbd1
NAME PROVISIONED USED
rbd1          5 GiB 36 MiB
```

步骤 2 客户端扩容

Rbd 设备扩容后，无法直接在客户端生效，需要在执行分区扩容才可以完全完成扩容操作，具体如下：

```
[root@client ~]# xfs_growfs /dev/rbd0
meta-data=/dev/rbd0          isize=512    agcount=8, agsize=98304 blks
        =                   sectsz=512   attr=2, projid32bit=1
        =                   crc=1        finobt=1, sparse=1, rmapbt=1
        =                   reflink=1    bigtime=1 inobtcount=1 nnext64=1
data      =                   bsize=4096   blocks=786432, imaxpct=25
        =                   sunit=16     swidth=16 blks
naming    =version 2          bsize=4096   ascii-ci=0, ftype=1
log        =internal log     bsize=4096   blocks=16384, version=2
        =                   sectsz=512   sunit=16 blks, lazy-count=1
realtime  =none              extsz=4096   blocks=0, rtextents=0
data blocks changed from 786432 to 1310720
[root@client ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda2       40G   4.5G   34G   12% /
devtmpfs        4.0M    0   4.0M    0% /dev
tmpfs           1.7G    0   1.7G    0% /dev/shm
tmpfs           4.0M    0   4.0M    0% /sys/fs/cgroup
tmpfs           657M   8.6M  649M    2% /run
tmpfs           1.7G   4.0K   1.7G    1% /tmp
/dev/rbd0       5.0G  131M   4.9G    3% /mnt/rbd1
/dev/rbd1       5.0G  130M   4.9G    3% /mnt/rbd2
```

扩容完成后，检查是否生效

5.4 RBD 快照管理

本实验将为 rbd2 创建快照 snap1，并基于该快照克隆一个新的 rbd 设备，将其命名为 rbd3。

步骤 1 准备工作

在 Client 的 /mnt/rbd2 中创建测试文件 test，并输入 “this is a test file”，具体操作如下：

```
[root@client ~]# echo "this is a test file" > /mnt/rbd2/test
[root@client ~]# cat /mnt/rbd2/test
this is a test file
```

步骤 2 创建快照

返回 ceph01，使用以下命令为 rbd2 创建快照 snap1：

```
[root@ceph01 ~]# rbd snap create block/rbd2@snap1
Creating snap: 100% complete...done.
[root@ceph01 ~]# rbd snap ls block/rbd2
SNAPID NAME SIZE PROTECTED TIMESTAMP
4 snap1 5 GiB Mon May 25 18:02:17 20XX
```

步骤 3 测试快照功能

返回 Client，将 /mnt/rbd2 中的 test 删除，然后回滚快照，检查是否可以恢复 test。

首先，在 Client 上删除 test（自行完成），取消 rbd 设备的映射和挂载，具体操作如下：

```
[root@client ~]# umount /mnt/rbd2
[root@client ~]# rbdmap unmap block/rbd2

[root@client ~]# umount /mnt/rbd2
[root@client ~]# rbdmap unmap block/rbd2
[root@client ~]# lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS
sda 8:0 0 40G 0 disk
├─sda1 8:1 0 2G 0 part /boot
└─sda2 8:2 0 38G 0 part /
```

完成后，在 Ceph 管理节点上，使用以下命令回滚快照：

```
[root@ceph01 ~]# rbd snap rollback block/rbd2@snap1
[ceph: root@ceph01 /]# rbd snap rollback block/rbd2@snap1
Rolling back to snapshot: 100% complete...done.
```

完成后在客户端重新映射和挂载 RBD 设备，查看 test 文件是否存在：

```
[root@ceph01 /]# rbdmap map block/rbd2 --name client.block
[root@ceph01 /]# lsblk
[root@ceph01 /]# mount -a
[root@ceph01 /]# ls /mnt/rbd2
```

```
[root@client ~]# rbdmap map block/rbd2 --name client.block
[root@client ~]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda          8:0    0   40G  0 disk
├─sda1       8:1    0    2G  0 part /boot
└─sda2       8:2    0   38G  0 part /
rbd0        253:0    0    5G  0 disk /mnt/rbd1
rbd1        253:16   0    5G  0 disk /mnt/rbd2
[root@client ~]# mount -a
[root@client ~]# ls /mnt/rbd2
test
[root@client ~]#
```

步骤 4 使用快照创建克隆卷并挂载

在 ceph01 上，使用以下命令将快照设为保护模式：

```
[root@ceph01 ~]# rbd snap protect block/rbd2@snap1
```

然后使用以下命令创建 snap1 的克隆卷 rbd3：

```
[root@ceph01 ~]# rbd clone block/rbd2@snap1 block/rbd3
```

将克隆卷转为独立 rbd 设备：

```
[root@ceph01 ~]# rbd flatten block/rbd3
```

```
[ceph: root@ceph01 /]# rbd snap protect block/rbd2@snap1
[ceph: root@ceph01 /]# rbd clone block/rbd2@snap1 block/rbd3
[ceph: root@ceph01 /]# rbd -p block ls
rbd1
rbd2
rbd3
[ceph: root@ceph01 /]# rbd flatten block/rbd3
Image flatten: 100% complete...done.
```

在客户端将生成的 rbd3 映射到 Client 中，具体操作如下：

```
[root@client ~]# rbd map block/rbd3 --name client.block
```

然后将该克隆卷的母卷卸载，最后再将克隆卷挂载到新创建的目录/mnt/rbd3 中，具体命令如下：

```
[root@client ~]# umount /mnt/rbd2
[root@client ~]# mkdir /mnt/rbd3
[root@client ~]# mount /dev/rbd2 /mnt/rbd3
```

挂载完成后，查看是否存在测试文件 test：

```
[root@client ~]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda          8:0    0   40G  0 disk
├─sda1       8:1    0    2G  0 part /boot
└─sda2       8:2    0   38G  0 part /
rbd0        253:0    0    5G  0 disk /mnt/rbd1
rbd1        253:16   0    5G  0 disk /mnt/rbd2
rbd2        253:32   0    5G  0 disk /mnt/rbd3
[root@client ~]# ls /mnt/rbd3
test
[root@client ~]#
```

- 思考：卸载母卷是否是必须的？为什么？

- 回答：在本实验中是必须的，因为母卷被格式化为 xfs 格式，其 uuid 和克隆卷是一致的，因此在挂载克隆卷时，系统会认为已经挂载而报错。如果是其他文件系统则可以不卸载。
-

6 Ceph 文件存储 cephfs 实践

6.1 实验介绍

6.1.1 关于本实验

本实验将在 Ceph 中创建两个副本池资源池，一个用于存放 cephfs 的元数据，一个用于存放 cephfs 的数据，并创建响应的用户，用于客户端挂载时的鉴权。

6.1.2 实验目的

本实验主要针对 cephfs 的使用，包括资源池的创建、挂载等操作，以加强对 cephfs 知识的理解

6.2 创建 cephfs 所需的资源

步骤 1 创建资源池

创建两个副本池，分别命名为 data_cephfs 和 metadata_cephfs，其中 data_cephfs 作为 cephFS 的数据池，metadata_cephfs 作为 cephFS 的元数据池，具体命令如下：

```
[root@ceph01 ~]# ceph osd pool create metadata_cephfs
pool 'metadata_cephfs' created
[root@ceph01 ~]# ceph osd pool create data_cephfs
pool 'data_cephfs' created
```

步骤 2 格式化资源池

将创建好的资源池进行格式化，具体命令如下：

```
[root@ceph01 ~]# ceph fs new fs01 metadata_cephfs data_cephfs
Pool 'data_cephfs' (id '12') has pg autoscale mode 'on' but is not marked as bulk.
Consider setting the flag by running
  # ceph osd pool set data_cephfs bulk true
new fs with metadata pool 11 and data pool 12
```

步骤 3 部署 MDS

在 Ceph 集群中部署两个 MDS，具体命令如下：

```
[root@ceph01 ~]# ceph orch apply mds fs01 --placement="2"
```

```
Scheduled mds.fs01 update...
```

部署完成后，查看 MDS 的状态，具体如下：

```
[root@ceph01 ~]# ceph orch ls
NAME          PORTS  RUNNING  REFRESHED  AGE  PLACEMENT
crash         3/3    8m ago   85m  *
mds.fs01     2/2    15s ago  20s  count:2
mgr           3/3    8m ago   81m  label:_admin
mon           3/3    8m ago   81m  label:_admin
osd           9      8m ago   -      <unmanaged>
rgw.for_swift 2/2    8m ago   49m  count:2
```

步骤 4 创建用户

创建用户 user01，使其可对 fs01 有读写权限，具体命令如下：

```
[root@ceph01 ~]# ceph fs authorize fs01 client.user01 / rwps -o ceph.client.user01.keyring
```

创建完成后检查是否配置正确，如下图所示：

```
[root@ceph01 ~]# ceph auth ls | grep -A 4 user
client.user01
  key: AQC4DhVqD32UDxAACczgG3nUhMjNWnVngatJXQ==
  caps: [mds] allow rwps fsname=fs01
  caps: [mon] allow r fsname=fs01
  caps: [osd] allow rw tag cephfs data=fs01
```

6.3 客户端挂载

将创建的 cephfs 挂载到客户端的/mnt/cephfs 目录中，并需实现开机自动挂载。

步骤 1 拷贝 user01 的 keyring 到客户端

在 Ceph01 服务器中，将 6.2 中生成的 user01 的 keyring 文件拷贝到客户端的/etc/ceph 目录中，具体如下：

```
[root@ceph01 ~]# scp ceph.client.user01.keyring root@192.168.0.24:/etc/ceph/

Authorized users only. All activities may be monitored and reported.
root@192.168.0.24's password:
ceph.client.user01.keyring
```

步骤 2 创建认证文件

创建认证文件 user.txt，将用户信息和密钥信息保存其中，具体如下：

```
[root@client ~]# echo "AQC4DhVqD32UDxAACczgG3nUhMjNWnVngatJXQ==" > user.txt
[root@client ~]# cat user.txt
AQC4DhVqD32UDxAACczgG3nUhMjNWnVngatJXQ==
```

步骤 3 修改/etc/fstab 文件

在/etc/fstab 文件中加入以下内容：

```
[root@client ~]# vim /etc/fstab

192.168.0.22:6789,192.168.0.23:6789:/mnt/cephfs ceph
noatime,_netdev,name=user01,secretfile=/root/user.txt 0 2
```

修改完成后，使用命令 `mount -a` 进行挂载，然后使用命令 `mount` 检查是否挂载成功，具体如下：

```
[root@client ~]# systemctl daemon-reload
[root@client ~]# mkdir -p /mnt/cephfs
[root@client ~]# mount -a
[root@client ~]# mount | grep ceph
192.168.0.22:6789,192.168.0.23:6789:/ on /mnt/cephfs type ceph
(rw,noatime,name=user01,secret=<hidden>,fsid=37511d4e-57ee-11f1-b5ff-fa163e4da5ba,acl,_netdev)
```

自行尝试/mnt/cephfs 中进行文件创建和删除，测试 user01 是否对该资源池有读写权限。

- 思考：/etc/fstab 文件中的 192.168.0.22 和 192.168.0.23 是什么地址？使用其他地址是否可以？
- 答：192.168.0.22 和 192.168.0.23 是 MDS 所在节点的地址，需要在部署为 MDS 后在 Ceph 管理节点上获取到该地址。在本实验中，这两个地址不可替换，如果在部署 MDS 指定了节点，或者某个 MDS 进程出现故障，系统重新将 MDS 部署到其他节点时，此处的地址需要进行更改，以保证客户端对 cephfs 的可用。

华为认证存储系列教程

HCIP-Storage

存储系统运维管理

实验指导手册

版本：6.0



华为技术有限公司

版权所有 ©【2026】【华为技术有限公司】。保留一切权利。

未经华为事先书面许可，任何人不得对本资料进行复制、修改、整合、改编、翻译，也不得将本资料或其任何部分或基于本资料的衍生作品提供、出售、转让、再许可给任何他人用于华为许可以外的任何其他用途。

Copyright ©【2026】【Huawei Technologies Co., Ltd.】. All Rights Reserved.

Without prior written permission of Huawei, no one is allowed to reproduce, modify, integrate, adapt, translate this Document, or provide, sell, transfer, sublicense this Document or any part of this Document or its derivative to any third party for any other purpose otherwise permitted by Huawei.

商标声明



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>

华为认证体系介绍

华为认证是华为公司基于“平台+生态”战略，围绕“云-管-端”协同的新ICT技术架构，华为公司打造了覆盖ICT领域的认证体系，包含ICT基础设施、应用与开发两类技术认证及项目管理认证。

根据ICT从业者的学习和进阶需求，华为认证分为工程师级别、高级工程师级别和专家级别三个认证等级。

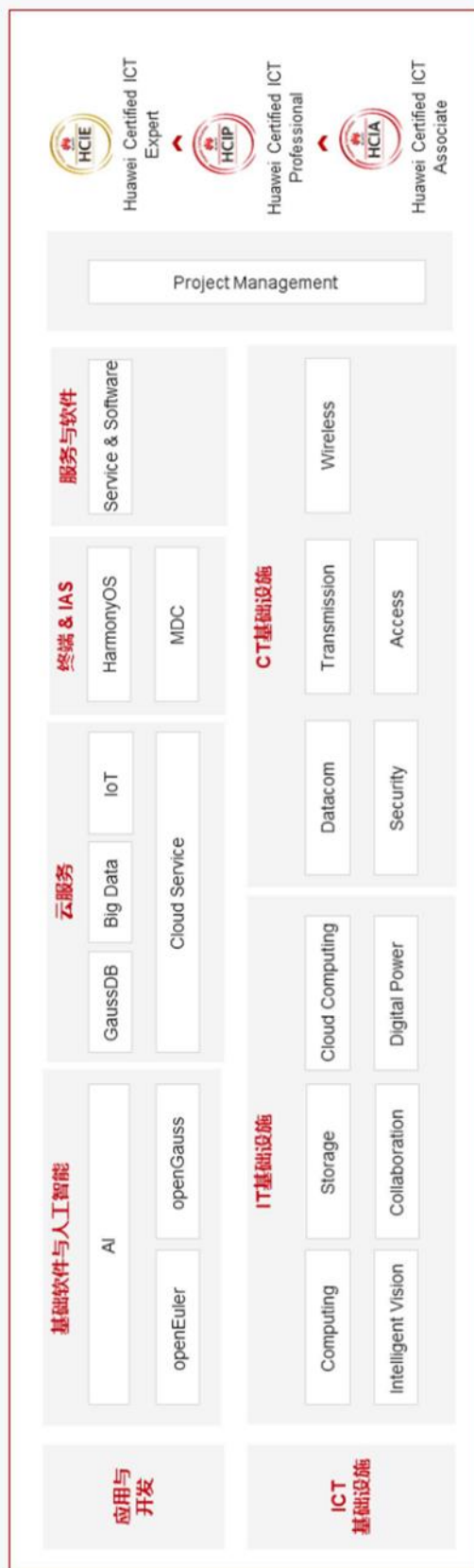
华为认证覆盖ICT全领域，符合ICT融合的技术趋势，致力于提供领先的人才培养体系和认证标准，培养数字化时代新型ICT人才，构建良性ICT人才生态。

HCIP-Storage（Huawei Certified ICT Professional-Storage，华为认证存储高级工程师）主要面向华为公司渠道工程师，高校学生，以及其他ICT从业人员等。HCIP-Storage认证在内容上涵盖闪存存储技术、数据保护技术、分布式存储技术、虚拟化技术、以及存储系统运维管理等内容。

华为认证协助您打开行业之窗，开启改变之门，屹立在存储行业的潮头浪尖！

华为职业认证全景图

华为职业认证覆盖ICT全领域，致力于提供领先的人才培养体系和认证标准，培养数字化时代的新型ICT人才，构建良性的ICT人才生态。



前言

课程介绍

本课程将结合之前学习的内容，进行操作演练，帮助学员巩固和复习之前学习的内容，重点介绍华为闪存存储的运维管理与故障处理的操作演练。

内容描述

本实验指导书共包含 2 个实验，介绍运维管理以及相关工具的操作，以及故障处理的操作：

实验一：运维管理实验。

实验二：SmartKit 巡检实验

读者知识背景

本课程为华为认证高级课程，为了更好地掌握本书内容，阅读本书的读者应首先具备以下基本条件：

- 了解基本的存储基础知识，建议完成 HCIA-Storage 的学习，并通过了 HCIA-Storage 认证考试。
 - 熟悉 Linux 操作系统操作，了解网络基础知识。
-

目录

前 言	4
课程介绍	4
内容描述	4
读者知识背景	4
1 参考资料及工具	6
1.1 参考资料	6
1.2 软件工具	6
2 存储系统运维管理	7
2.1 实验介绍	7
2.1.1 关于本实验	7
2.1.2 实验目标	7
2.1.3 实验组网拓扑	7
2.2 日常管理	8
2.2.1 管理 License 文件	8
2.2.2 创建本地用户	9
2.2.3 配置安全策略	11
2.2.4 查看并处理告警	13
2.2.5 利用 DeviceManager 收集日志	16
2.3 SmartKit	17
2.3.1 安装 SmartKit	17
2.3.2 运行 SmartKit	21
2.3.3 使用 SmartKit 进行巡检	25
2.4 思考题	32

1 参考资料及工具

1.1 参考资料

文档中所列出的命令、文档及文档路径仅供参考，请根据实际环境中的不同产品版本使用对应的命令、文档及文档路径。

产品文档



说明：

您可以直接登录华为技术有限公司技术支持网（<http://support.huawei.com/enterprise/>），在搜索栏中，输入“文档或工具名称”，检索、浏览和下载对应文档或工具。

1.2 软件工具

1. PuTTY



说明：

使用开源软件 PuTTY 进行终端登录操作，您可以通过 PuTTY 常用域名（putty.org）浏览和下载对应文档或工具。

2. WinSCP



说明：

使用 WinSCP 在 Windows 系统和 Linux 系统间传输文件，您可以通过 WinSCP 常用域名（winscp.net）浏览和下载对应文档或工具。

3. SmartKit



说明：

您可以登录华为技术有限公司技术支持网站（<http://support.huawei.com/enterprise/>），在搜索栏中，输入“SmartKit”，检索、浏览和下载对应文档或工具。

2 存储系统运维管理

2.1 实验介绍

2.1.1 关于本实验

某公司购买了一台华为全闪存存储设备，用来运行多个业务。计划在数据中心 DCA 中的存储 DCA-Storage 上创建 LUN 映射到 Linux 主机 DCA-Host。为了保障业务运行正常，需要运维人员在 DCA-Win 上安装 SmartKit 工具进行运维管理操作。

2.1.2 实验目标

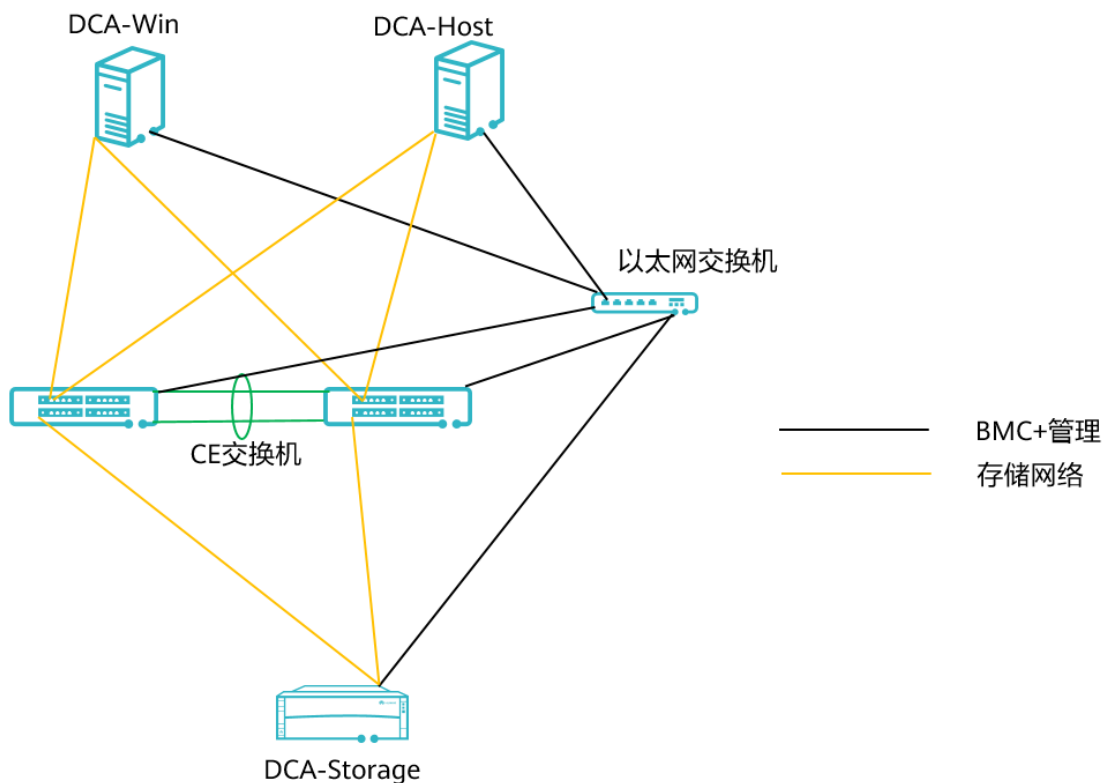
- 通过实验学习能够进行存储系统运维管理；
- 通过实验学习能够使用 SmartKit 工具进行存储巡检。

2.1.3 实验组网拓扑

- 实验设备

设备类型	数量	软件版本
Linux主机	1	CentOS 7.6
Windows主机	1	Windows Server 2016, SmartKit 25.0
华为闪存存储	1	V700R001, UltraPath_31.2.0

- 组网拓扑
-



• 实验网络信息

设备信息	管理IP	存储IP	说明
DCA-Host	192.168.5.151	192.168.6.151	Linux主机
DCA-Win	192.168.5.101	192.168.6.101	Windows主机
DCA-Storage	192.168.5.200	192.168.6.201	块业务A
		192.168.6.202	块业务B
		192.168.6.203	文件业务A
		192.168.6.204	文件业务B

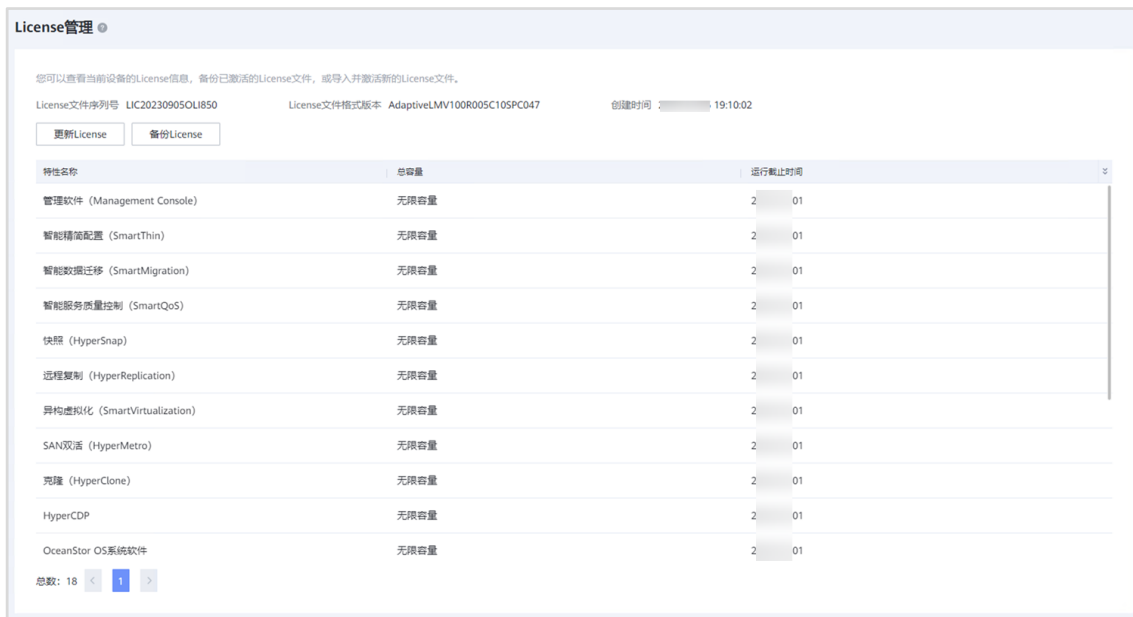
2.2 日常管理

2.2.1 管理 License 文件

步骤 1 浏览已激活 License 信息

登录 DeviceManager，选择“设置 > License 管理”。

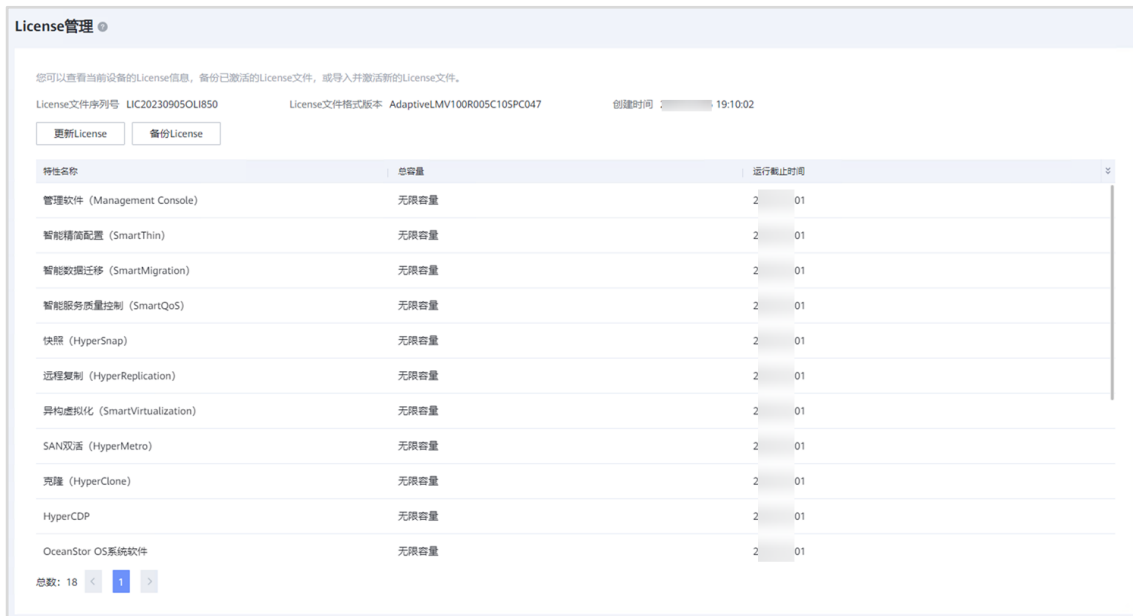
在中间信息展示区浏览存储设备已激活的 License 文件信息。



- 基于运行时间控制：显示该 License 的“运行截止时间”，“已使用/总容量”显示为“无限容量”或“--”。
- 基于容量控制：显示该 License 的“已使用/总容量”，“运行截止时间”显示为“永久”。

步骤 2 备份已激活的 License 文件

登录 DeviceManager，选择“设置 > License 管理”，单击“备份 License”。



License 文件会被下载和保存在浏览器设置的下载保存路径。

2.2.2 创建本地用户

登录 DeviceManager，选择“设置 > 用户与安全 > 用户和角色 > 用户管理”，系统进入“用户管理”界面，单击“创建”。

用户和角色

[用户管理](#) | [角色管理](#)

请根据系统定义的不同角色用户的操作权限创建不同的用户，限制用户对系统的操作，以保障用户业务系统的稳定及业务数据的安全。

创建

系统弹出“创建用户”对话框。设置用户信息。在“类型”中选择“本地用户”，设置本地用户信息。

类型：本地用户

用户名、密码：自定义

角色：管理员

其他参数默认，点击“确定”。

创建用户

* 类型

本地用户

* 用户名

test123

* 密码

.....

* 确认密码

.....

* 角色

管理员

密码永不过期

描述

长度范围是0到255位。

* 登录方式

☒ CLI

☒ RESTful

☒ DeviceManager

☒ SFTP

☒ 串口

用户为非超级管理员时，请至少勾选上述一种登录方式。

* 登录认证

☒ 登录密码

☐ 登录密码+邮件动态口令

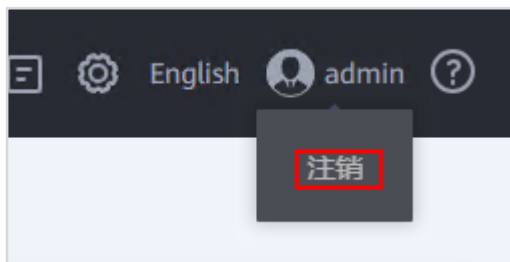
确定

取消

弹出执行结果，点击“关闭”。



完成后，可退出当前登录用户，尝试使用新建用户登录。

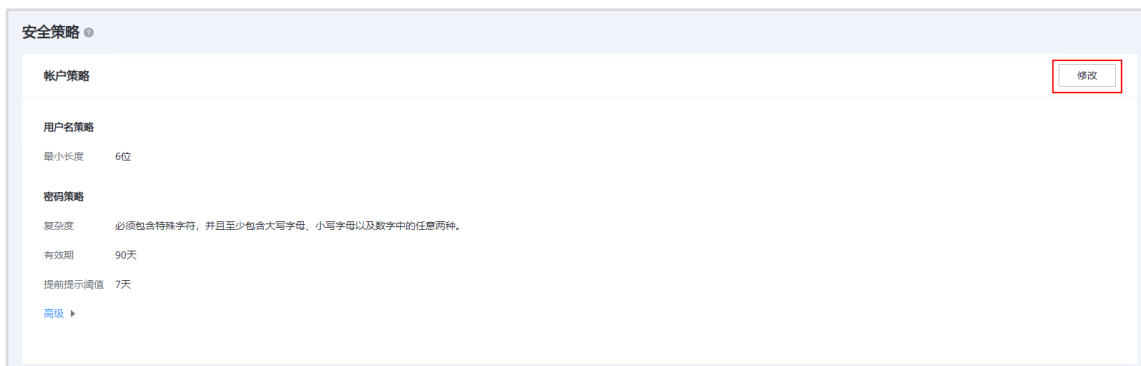


测试完新建用户后，退出新用户，重新使用 admin 用户登录。

2.2.3 配置安全策略

步骤 1 配置帐户策略

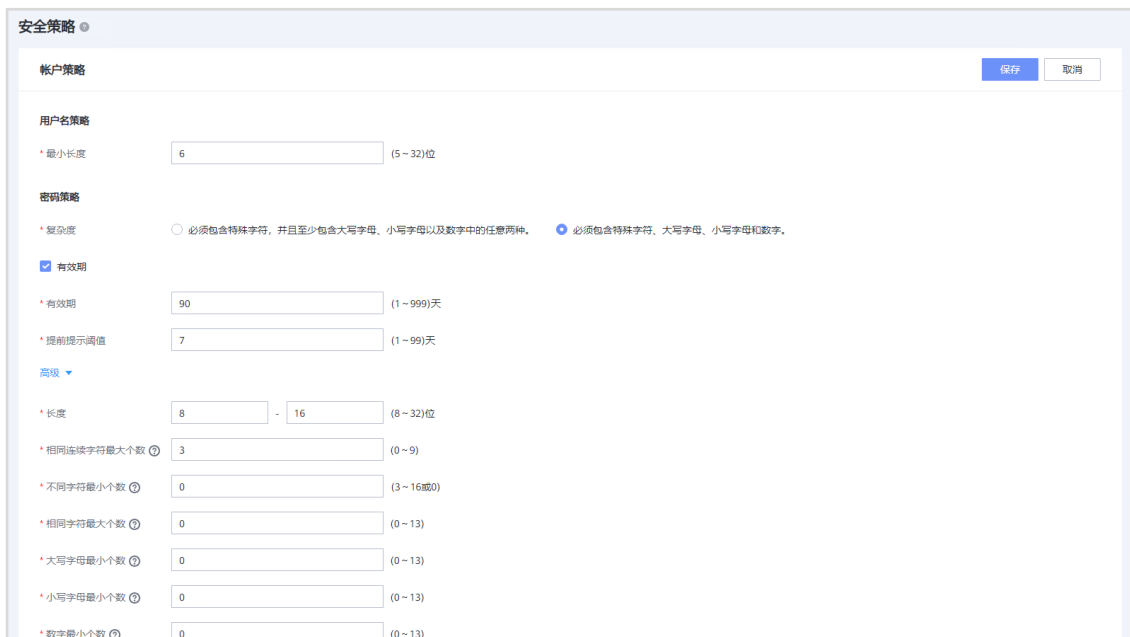
登录 DeviceManager，选择“设置 > 用户与安全 > 安全策略”，单击“帐户策略”右侧的“修改”。



单击“高级”可展开“帐户策略”的高级设置，如下参数配置帐户策略：

密码策略下的复杂度选择则“必须包含特殊字符、大写字母、小写字母和数字。”

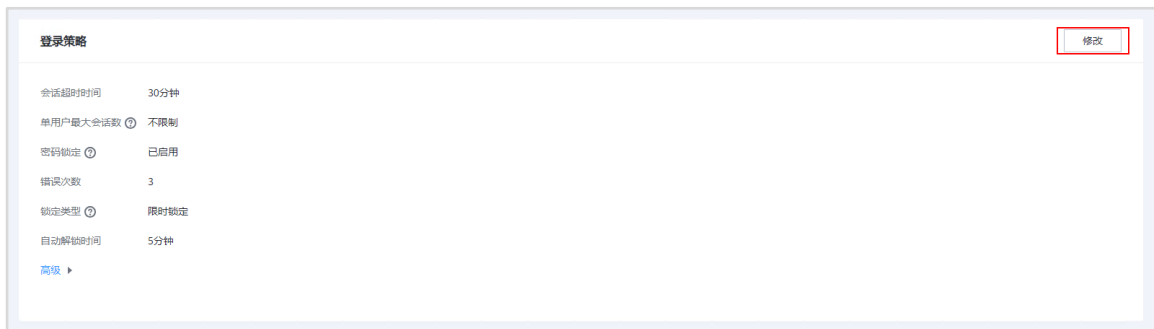
其他参数保持默认，点击“保存”。



完成后，参考“创建本地用户”的步骤，创建新用户，验证密码策略生效。

步骤 2 配置登录策略

选择“设置 > 用户与安全 > 安全策略”，单击“登录策略”右侧的“修改”。



单击“高级”可展开“登录策略”的高级设置，如下参数配置帐户策略：

自动解锁时间：3 分钟

其他参数保持默认，点击“保存”。

登录策略

保存 取消

* 会话超时时间

30

(1 ~ 100)分钟

* 单用户最大会话数

0

(0 ~ 32)

☒ 密码锁定

* 错误次数

3

(1 ~ 9)

* 锁定类型

限时锁定

* 自动解锁时间

3

(3 ~ 2000)分钟

高级

☐ 闲置锁定

☒ IP校验

☐ 登录安全提示

☐ 自定义提示

完成后，退出当前用户重新登录，故意输入错误密码 3 次，验证账户被锁定。等待 3 分钟后登录，验证自动解锁。

2.2.4 查看并处理告警

登录 DeviceManager。选择“监控 > 告警和事件 > 当前告警”。

告警和事件

当前告警 | 所有事件

您可以 [发送模拟告警](#)。未测试设备的告警功能是否正常。当产生重要及以上级别的告警时，设备前面板上的告警指示灯会亮起。如果不期望告警指示灯亮起，您可以单击 [设置告警级别](#) 降低告警的级别。

清除

级别

描述

告警对象

发生时间

☐	警告	(智能WORM) 特性的License即将过期。	license	
☐	警告	(智能精英配置) 特性的License即将过期。	license	
☐	警告	(克隆) 特性的License即将过期。	license	
☐	警告	(可得容量) 特性的License即将过期。	license	
☐	警告	(远程复制) 特性的License即将过期。	license	
☐	警告	(智能数据迁移) 特性的License即将过期。	license	
☐	警告	(智能克隆) 特性的License即将过期。	license	
☐	警告	(管理软件) 特性的License即将过期。	license	
☐	警告	(NAS基础特性) 特性的License即将过期。	license	

总数: 23, 选中: 0

1

2

可以单击右侧的 ，定制显示的列，可以根据需要，设置是否显示“ID”等选项。

告警对象

发生时间

级别

ID

告警对象

发生时间

流水号

license		☒ 级别
license		☐ ID
license		☒ 告警对象
license		☒ 发生时间
license		☐ 流水号

可以单击“级别”、“告警对象”或“发生时间”旁的，对告警进行筛选。

告警对象: 端口 X				
☐	级别 	描述	告警对象 	发生时间 
☐	 重要	控制框 (CTE0.A) 与控制框 (CTE0.B) 之间RDMA链路异常。	端口	
☐	 警告	当前集群内管理网络存在冗余情况。	端口	

或单击“ID”旁的，对告警进行搜索。

ID: 0xFOCF0006 X				
☐	级别 	ID 	描述	告警对象 
☐	 重要	0xFOCF0006	控制器 (控制框 CTE0, 控制器 B) 被拔出。	控制器
☐	 重要	0xFOCF0006	控制器 (控制框 CTE0, 控制器 A) 被拔出。	控制器

单击某条告警的“描述”，参考右侧显示的“修复建议”处理告警。

告警和事件

当前告警 所有事件

您可以 [发送模拟告警](#) 来测试设备的告警功能是否正常。当产生重要及以上级别的告警时，设备前面板上的告警指示灯会亮。

[清除](#)

☐	级别 	描述
☐	 警告	(智能WORM) 特性的License即将过期。
☐	 警告	(智能精简配置) 特性的License即将过期。
☐	 警告	(克隆) 特性的License即将过期。
☐	 警告	(可得容量) 特性的License即将过期。
☐	 警告	(远程复制) 特性的License即将过期。
☐	 警告	(智能数据迁移) 特性的License即将过期。
☒	 警告	(智能配额) 特性的License即将过期。
☐	 警告	(管理软件) 特性的License即将过期。
☐	 警告	(NAS基础特性) 特性的License即将过期。

总数: 23, 选中: 1

告警详情

基本信息

级别  警告

设备名称 Huawei.Storage

设备位置 Lab

ID 0xFOC90001

发生时间

恢复时间

类型 当前告警

状态 未恢复

流水号 113

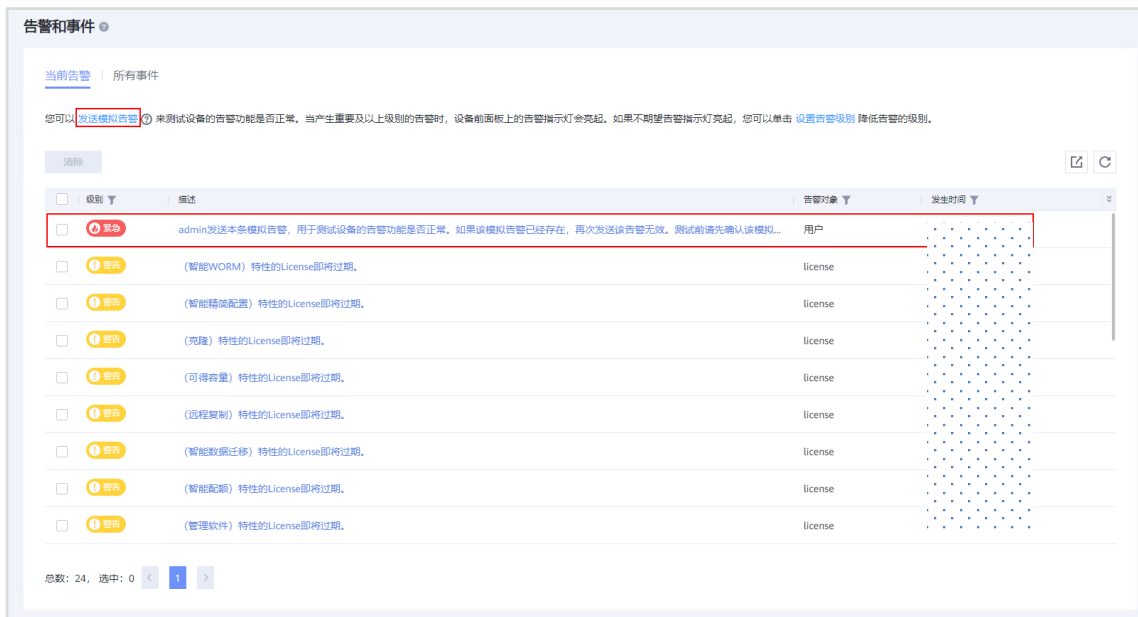
描述 (智能配额) 特性的License将在2023-12-03过期。License过期后会进入试用期，试用期结束后会失效。

修复建议

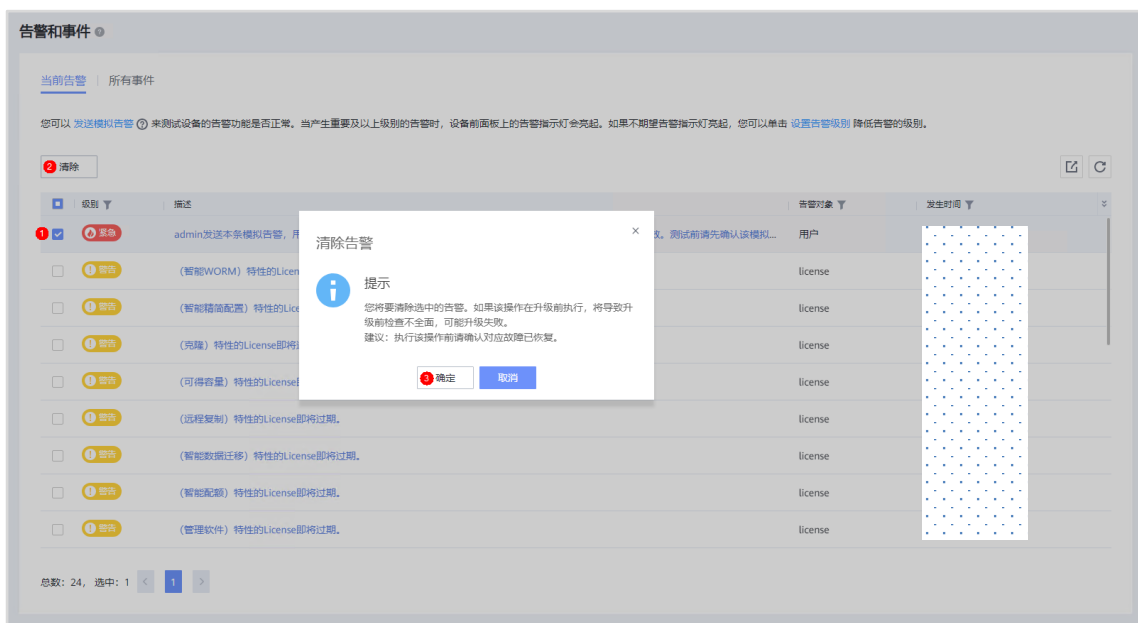
请购买并导入有效的License文件。

[关闭](#)

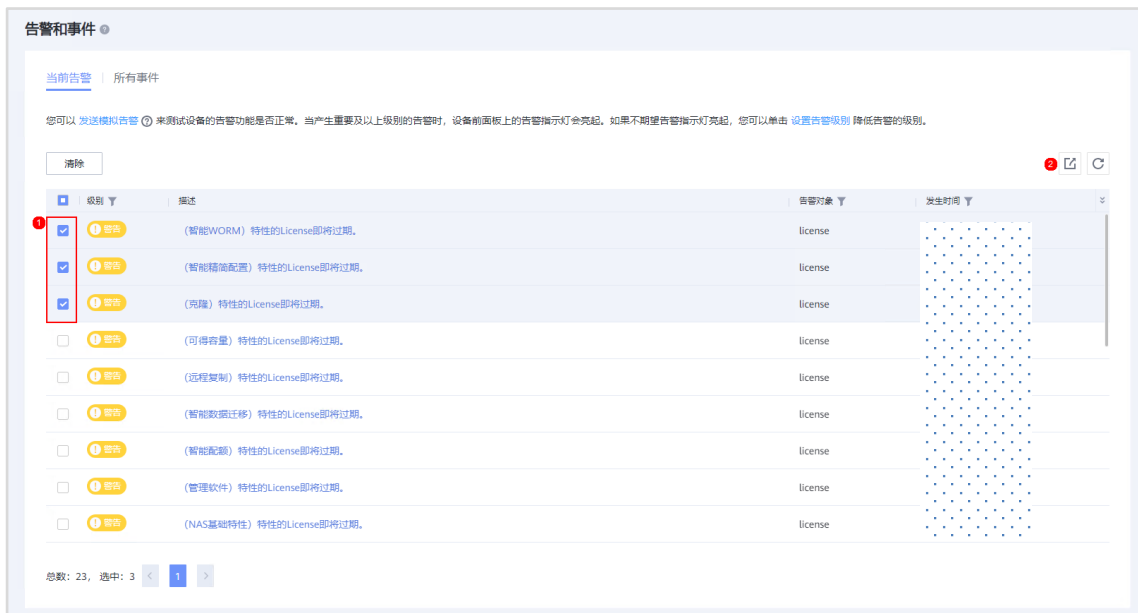
单击“发送模拟告警”，模拟上报故障告警。



在列表框中选择模拟告警，单击“清除”。在弹出的“提示”对话框中，单击“确定”。



勾选需要导出的告警，单击，导出所选高级为.xls 文件。



2.2.5 利用 DeviceManager 收集日志

登录 DeviceManager。选择“ > 导出数据”。



选择以下任务中的 2 项进行练习：

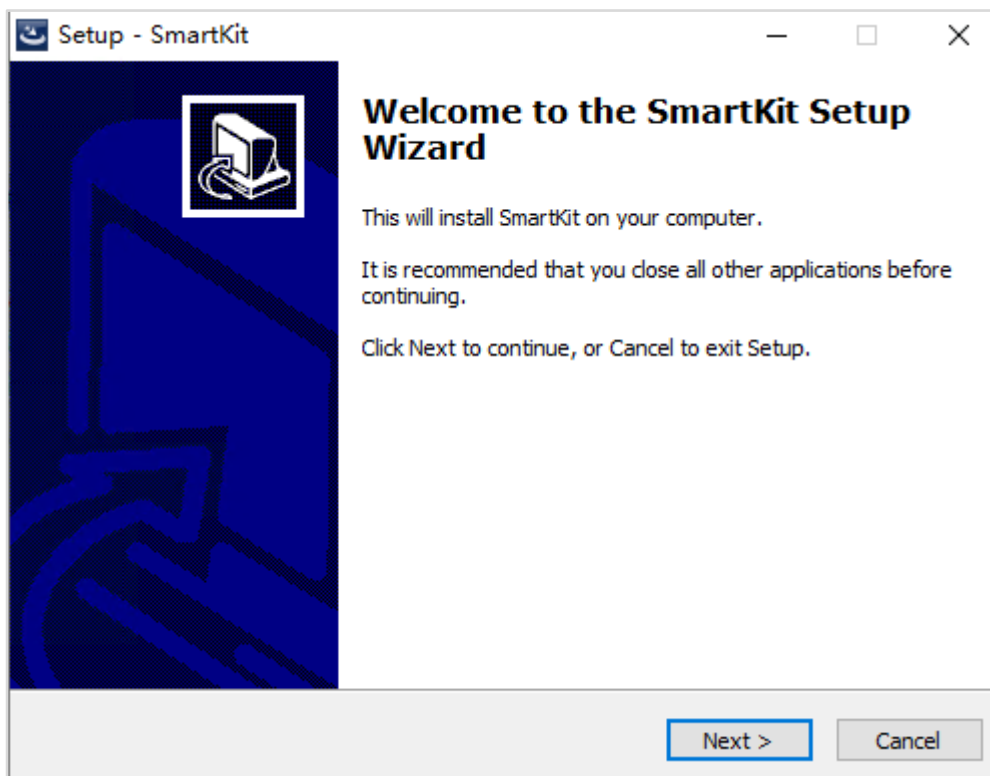
- 在“硬盘日志”下，单击“DHA 运行日志”、“HSSD 日志”或“硬盘数据擦除报告”，在弹出的“警告”对话框中勾选“我已阅读上述信息，了解执行此操作带来的后果。”，并单击“确定”，系统开始收集日志，并展开日志列表。
- 选择“配置信息”，单击“导出”，在弹出的“警告”对话框中勾选“我已阅读上述信息，了解执行此操作带来的后果。”，并单击“确定”，导出配置信息。
- 在“诊断文件”下，单击“导出”，在弹出的“警告”对话框中勾选“我已阅读上述信息，了解执行此操作带来的后果。”，并单击“确定”，导出设备的故障信息。
- 在“防病毒日志”下，单击“导出”，在弹出的“警告”对话框中勾选“我已阅读上述信息，了解执行此操作带来的后果。”，并单击“确定”，导出设备的防病毒扫描信息。
- 在“FTDS 日志”下，单击“导出”，在弹出的“警告”对话框中勾选“我已阅读上述信息，了解执行此操作带来的后果。”，并单击“确定”，导出设备的 FTDS 日志。
- 在“性能文件”下，选择文件日期区间，单击“导出”，在弹出的“警告”对话框中勾选“我已阅读上述信息，了解执行此操作带来的后果。”，并单击“确定”，导出设备的性能文件。

2.3 SmartKit

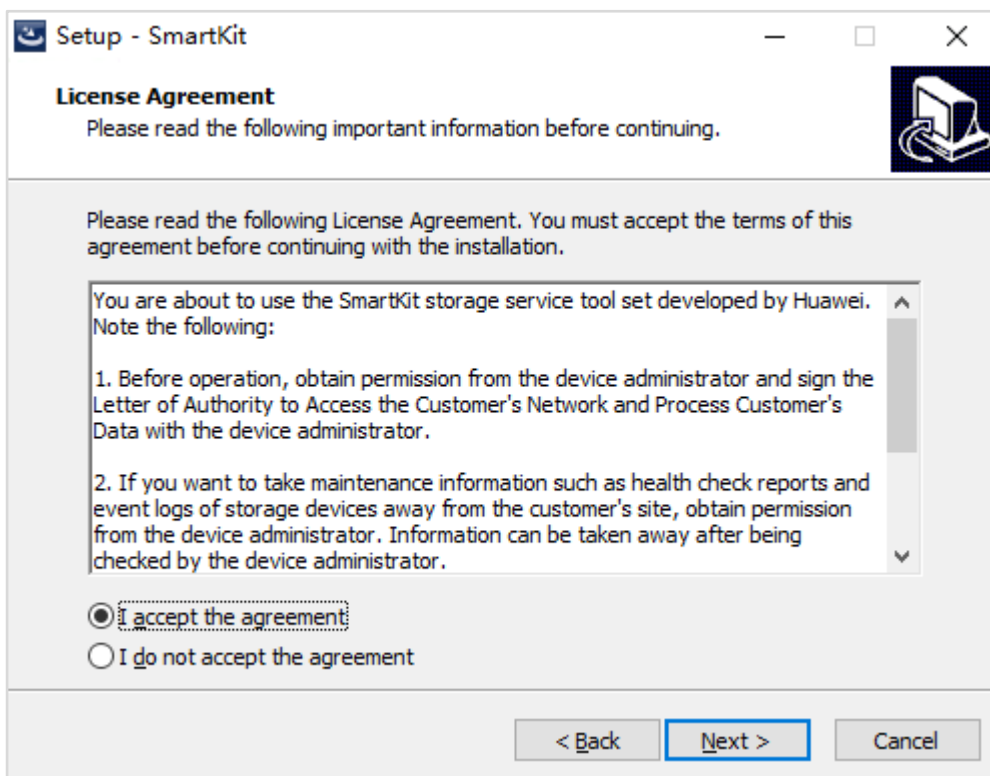
2.3.1 安装 SmartKit

登录 DCA-Win 主机，下载 SmartKit，并双击安装软件，开始安装 SmartKit。

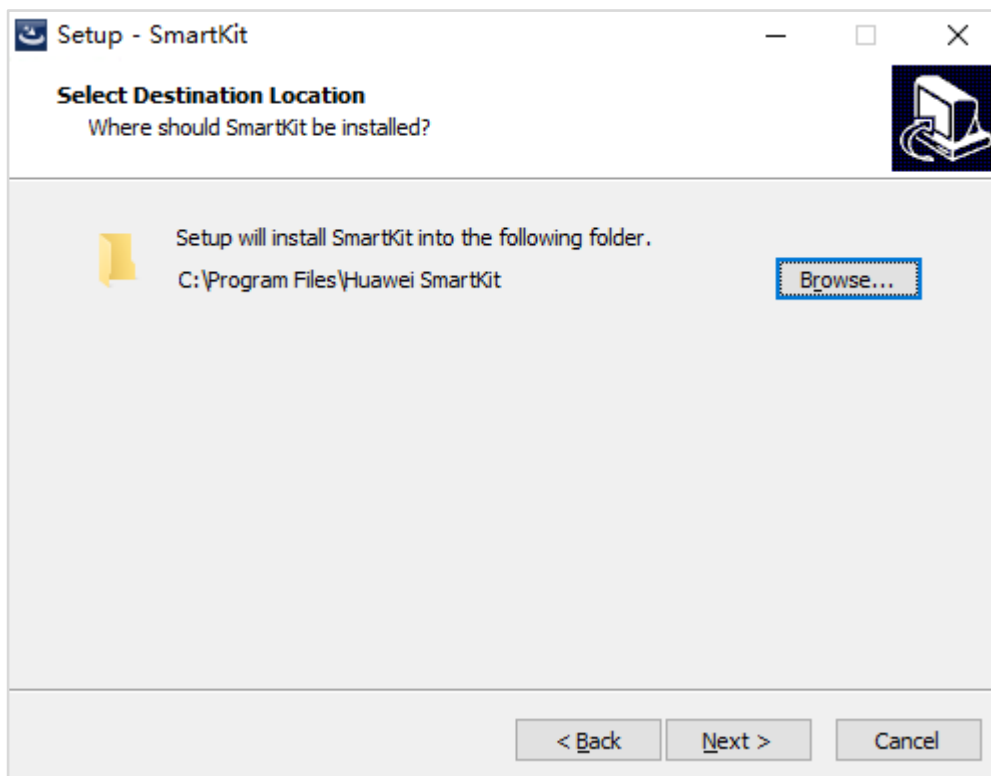
系统弹出安装向导欢迎页面，单击“Next”。



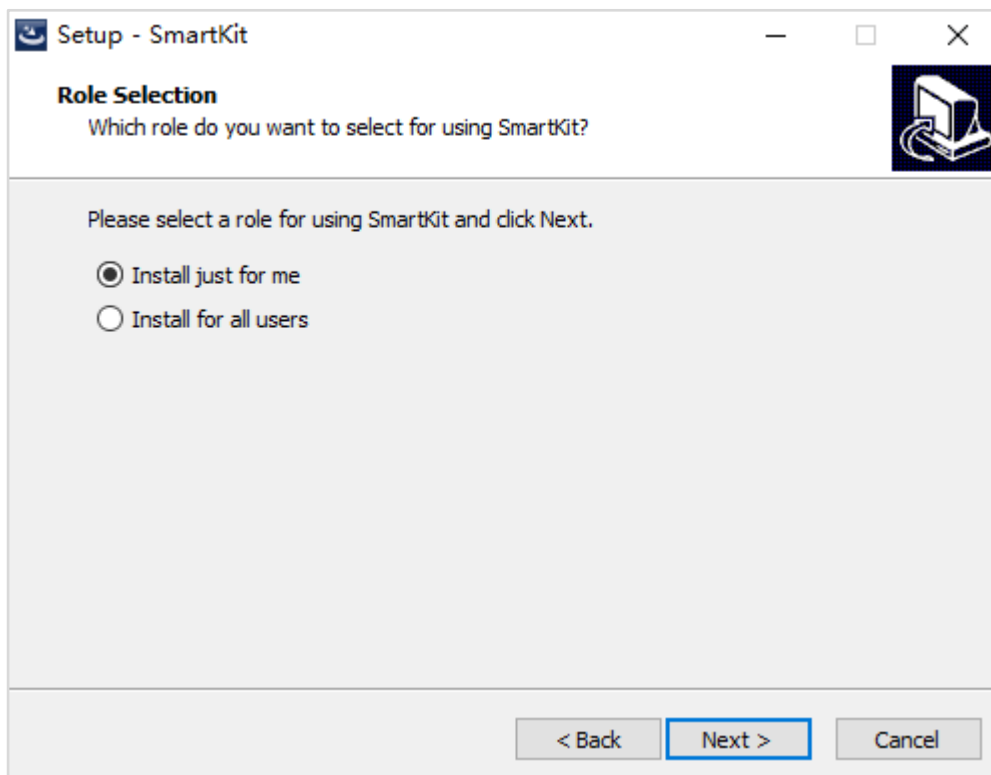
系统弹出“License Agreement”页面，选择“I accept the agreement”，并单击“Next”。



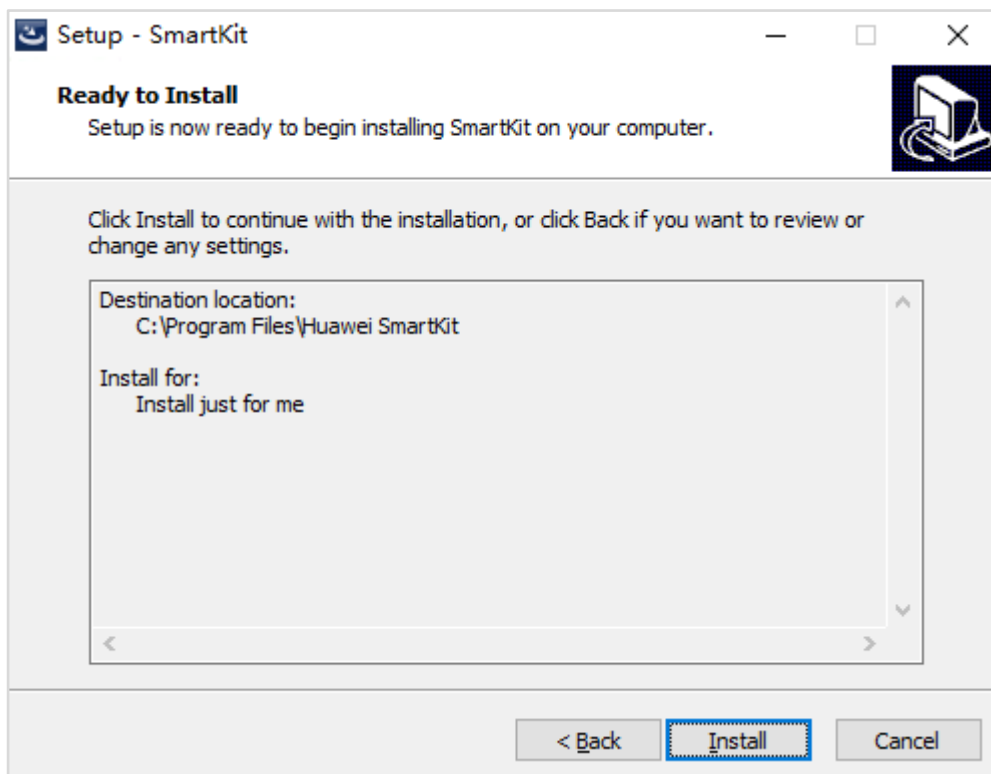
系统弹出“Select Destination Location”页面，单击“Next”。



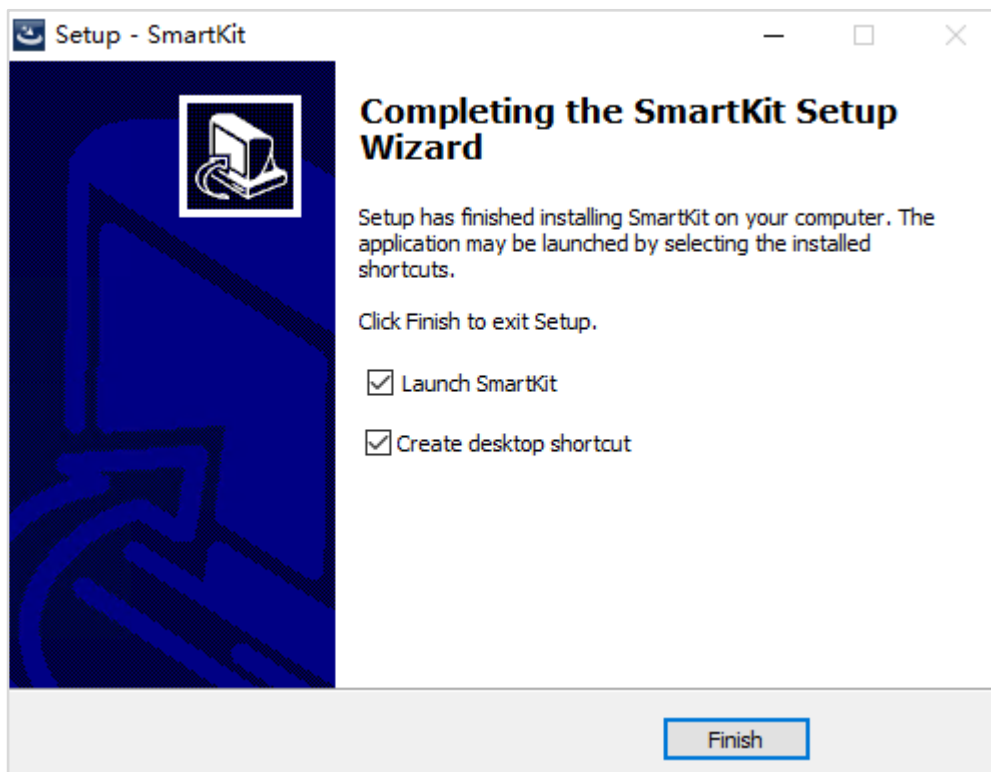
系统弹出“Role Selection”页面，选择“Install just for me”或“Install for all users”并单击“Next”。



系统弹出“Ready to Install”页面，单击“Install”，执行安装操作。



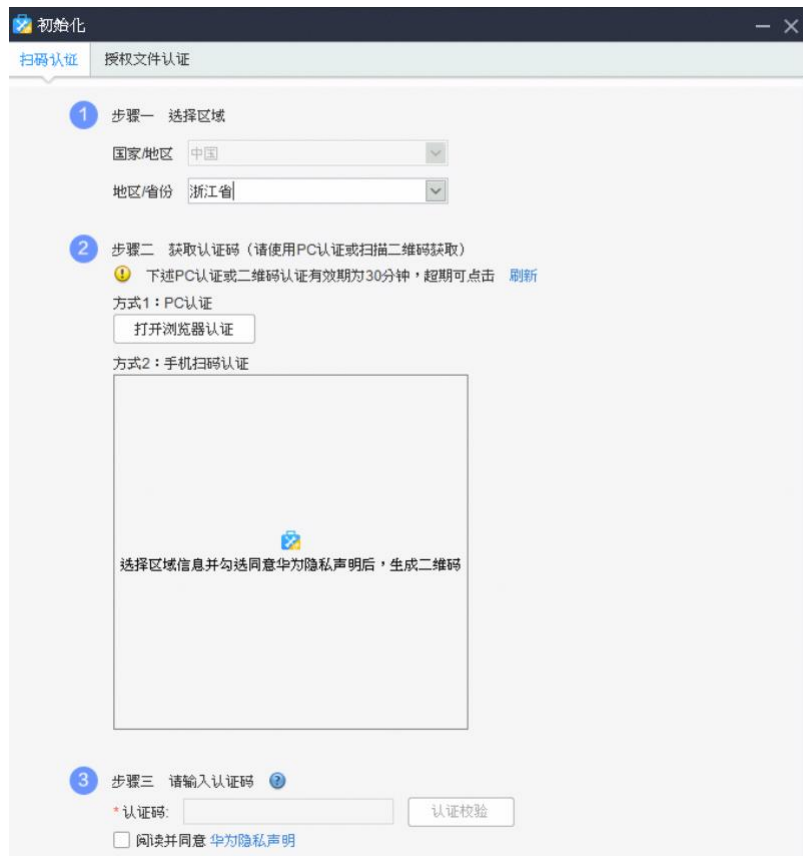
安装完成后，系统弹出“Completing the SmartKit Setup Wizard”页面，单击“Finish”。



2.3.2 运行 SmartKit

运行维护终端桌面上的“SmartKit”，进入登录认证。

一般情况下，需要选择区域、扫描二维码、输入验证码等操作进行登录认证。在没有外网的情况下，可以选择“PC 认证”或“手机扫码认证”使用华为账号进行认证。



The image shows the SmartKit login authentication interface. It has a title bar with '初始化' (Initialization) and a close button. Below the title bar are two tabs: '扫码认证' (QR Code Authentication) and '授权文件认证' (Authorization File Authentication). The main content area is divided into three steps:

- 步骤一 选择区域** (Step 1: Select Region): This step contains two dropdown menus. The first is labeled '国家/地区' (Country/Region) and has '中国' (China) selected. The second is labeled '地区/省份' (Region/Province) and has '浙江省' (Zhejiang Province) selected.
- 步骤二 获取认证码** (Step 2: Obtain Authentication Code): This step includes a warning icon and text: '下述PC认证或二维码认证有效期为30分钟，过期可点击 刷新' (The validity of the following PC authentication or QR code authentication is 30 minutes. After expiration, you can click Refresh). Below this, there are two options: '方式1: PC认证' (Method 1: PC Authentication) with a button '打开浏览器认证' (Open Browser Authentication), and '方式2: 手机扫码认证' (Method 2: Mobile QR Code Authentication) with a large rectangular area for scanning a QR code. Inside this area, there is a small QR code icon and the text '选择区域信息并勾选同意华为隐私声明后，生成二维码' (After selecting the region information and checking 'I agree to Huawei Privacy Policy', a QR code will be generated).
- 步骤三 请输入认证码** (Step 3: Please enter the authentication code): This step has a text input field labeled '* 认证码:' (Authentication Code) and a button '认证校验' (Authentication Check). Below the input field, there is a checkbox labeled '阅读并同意 华为隐私声明' (Read and agree to Huawei Privacy Policy).

系统弹出“操作安全注意事项”界面。

首次登录，请勾选“我已阅读“安全声明”，了解执行此操作带来的风险。”，根据提示在校验框中输入文字“我确认只在中国区使用”，单击“继续”。

×

操作安全注意事项

安全声明

您将使用 SmartKit 中国区版本，该版本将默认自动回传以下工具执行结果到华为技术支持中心：

- 1、工具使用行为（按钮点击、检查项结果记录等）
- 2、存储设备巡检报告（巡检项、巡检标准、巡检结果等）
- 3、存储设备硬盘健康分析报告（硬盘信息、硬盘打分结果等）
- 4、存储设备日志信息（设备日志、告警、配置信息、电子标签、硬盘 S.M.A.R.T 信息、硬盘 LOG、IO 错误码、CLI 信息等）
- 5、服务器设备健康检查报告（设备健康状态、电子标签、版本信息、配置、兼容性等的巡检结果和统计信息）
- 6、服务器设备日志信息（设备的运行日志、告警、配置信息、电子标签、操作系统的运行日志）

该版本仅限中国区使用，在使用中国区版本之前，请先阅读相关安全声明，确保您已了解相关法律风险。

在其他国家或地区或者不允许回传的场景请使用通用版（[企业网站](#) | [运营商网站](#)），以避免引起相关信息安全纠纷。

For information security, use the universal version ([Enterprise Support](#) | [Carrier Support](#)) outside China or not sent back.

☒ 我已阅读“安全声明”，了解执行此操作带来的风险。

中国区版本校验

请输入以下文字，以便系统确认您将使用中国区版本。如果不能输入中文，请点击该[链接](#)。

我确认只在中国区使用

我确认只在中国区使用

下载授权书

请务必事先征得设备管理员的许可，并与设备管理员签署《客户网络接入操作&数据处理授权书》后方可进行操作。

《客户网络接入操作&数据处理授权书》

下载(D)

《个人数据说明文档》

下载(P)

继续(C)

退出(E)

首次使用 SmartKit，系统将弹出使用向导界面，为您介绍 SmartKit 工具箱的主要功能。

向导界面将弹出欢迎使用 SmartKit，概述工具的使用领域及获取路径。



单击 ，向导界面将为您展示 SmartKit 完全兼容原 OceanStor Toolkit 功能管理功能，并提供务场景下的运维功能，涵盖存储、服务器和云计算领域。



单击 ，向导界面将为您介绍功能管理界面，您可以一键安装或升级所需功能，对于不能联网的情况，可以从其他 SmartKit 环境导出功能包后进行导入，实现离线安装或升级。



功能管理

在SmartKit中可以一键安装或升级所需功能；

对于不能联网的情况，可以从其他SmartKit环境导出功能包后进行导入，实现离线安装或升级。

单击，向导界面将为您介绍设备列表。将设备添加到设备列表中，则后续各项功能操作时无需单独添加设备，且支持批量导入设备。



设备列表

将设备添加到设备列表中，则后续各项功能操作时无需单独添加设备。

支持批量导入设备。

单击，向导界面将为您介绍更新提醒功能。当工具可升级、用户认证即将过期或已经到期时，系统会自动进行提醒，您也可以在高级设置里修改提醒频率。

单击“开始使用”。

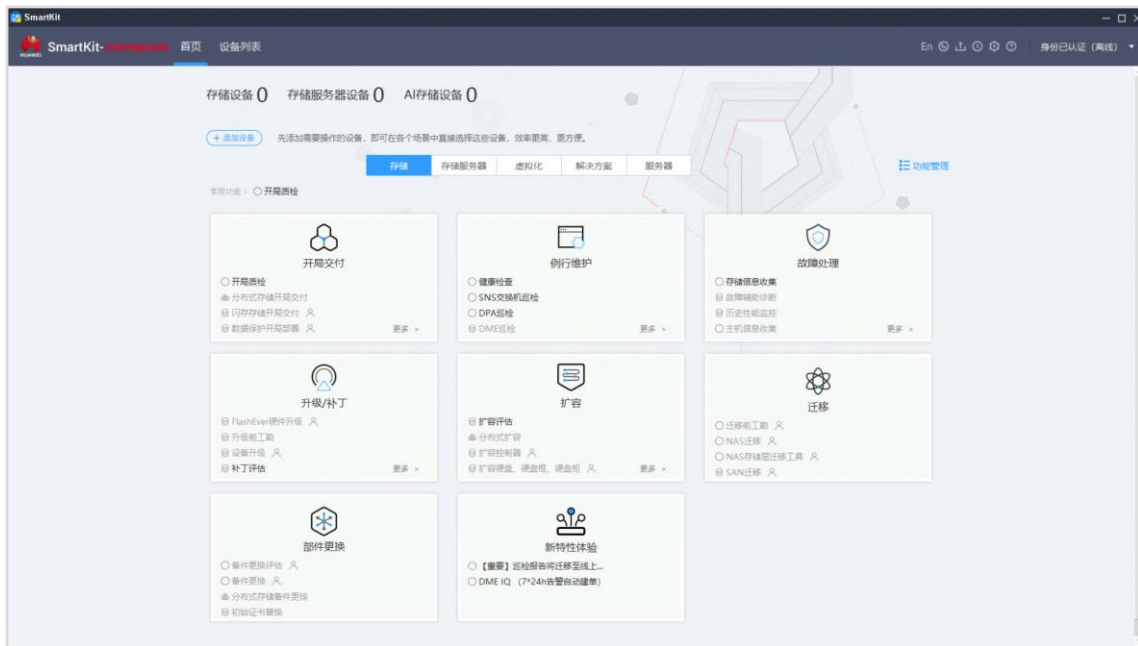


进入软件界面。

2.3.3 使用 SmartKit 进行巡检

步骤 1 运行 SmartKit 工具

启动 SmartKit 工具，进入主界面。



添加设备 步骤 2-2：配置信息

配置信息

请输入设备的用户名、密码和端口。

登录信息

用户名:

admin

密码:

端口:

22

是否需要校验证书:

?

是

否

需要调试密码

(查看需要调试密码的设备)

调试密码:

切换到root用户

(仅BCManager需要)

root密码:

上一步(P)

完成(E)

取消(C)

添加完成后，可在设备列表中看到添加的存储设备。

添加

修改

删除

更新

批量导入

批量导入数据

总数: 1

选中数: 0

请输入关键字搜索

型号	版本号	名称	IP地址	序列号	用户名	端口
OceanStor Dorado 5300 V6	V700R001C10	Huawei Storage	192.168.5.200	2102353UBF10H5100003	admin	22

步骤 3 巡检设备

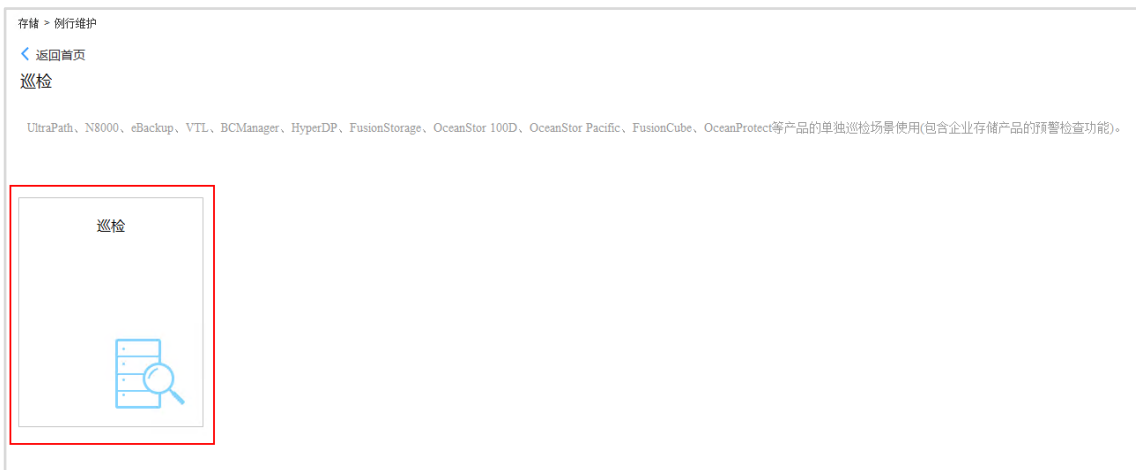
在主界面上选择“例行维护 > 更多”。



系统弹出“例行维护”界面，点击“巡检”。



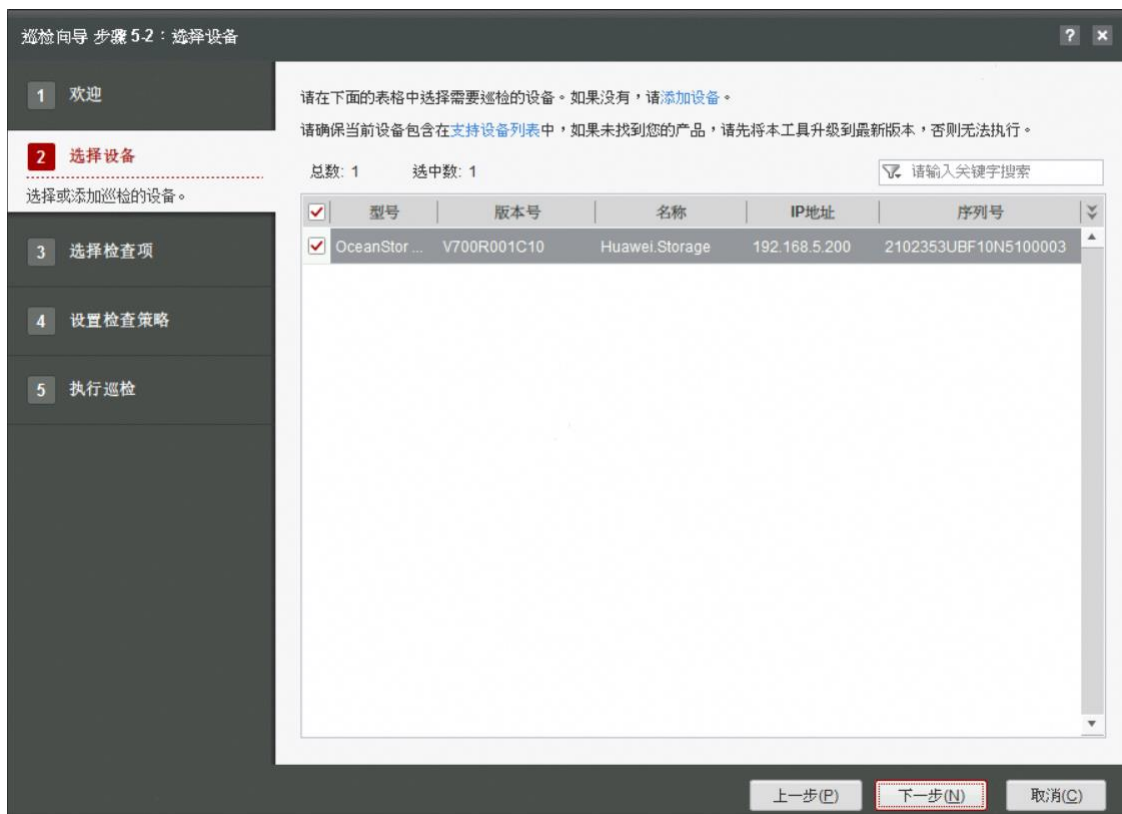
系统进入“巡检”界面，点击“巡检”。



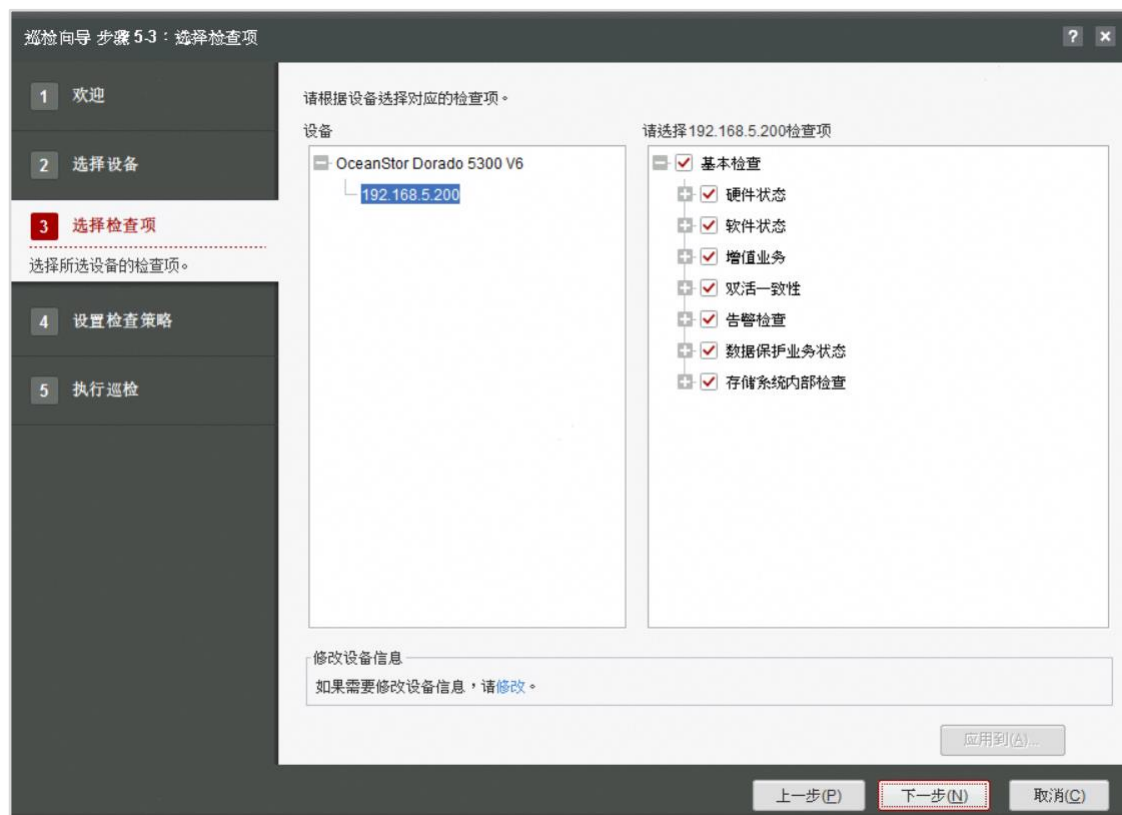
进入“巡检向导 步骤 5-1：欢迎”界面，选择“日常巡检”，点击“下一步”。



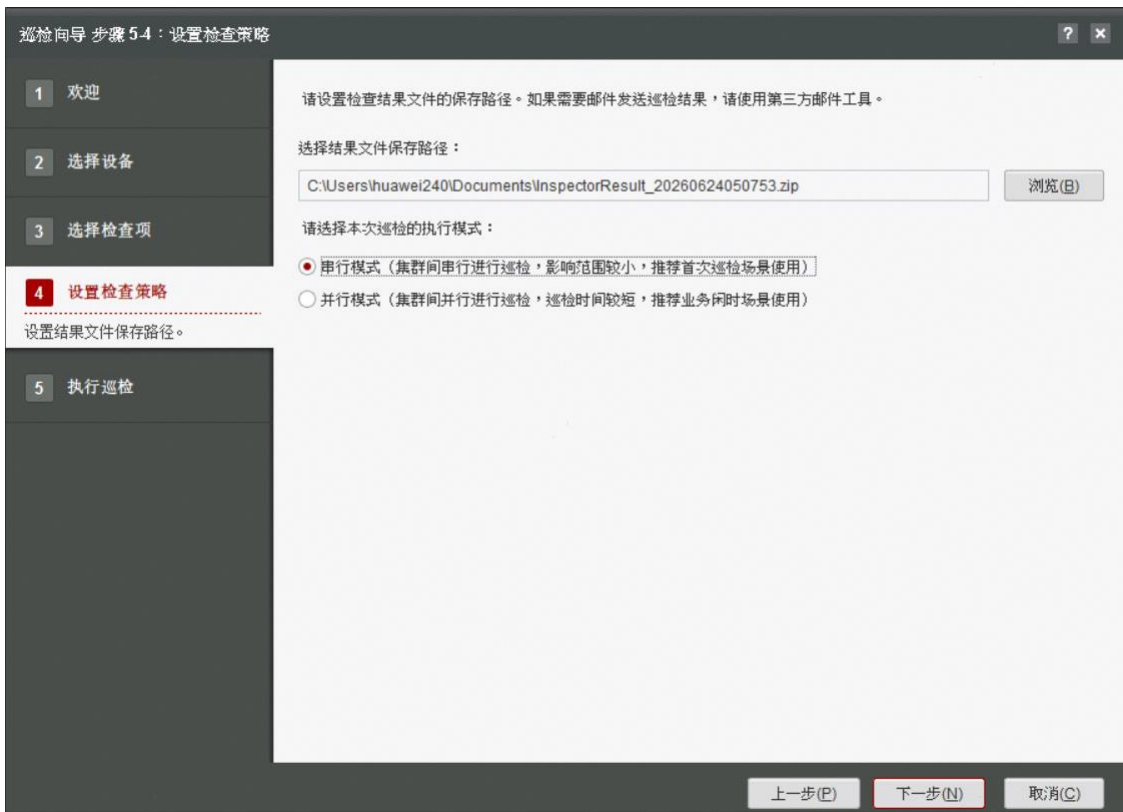
进入“巡检向导 步骤 5-2：选择设备”界面，勾选存储设备，点击“下一步”。



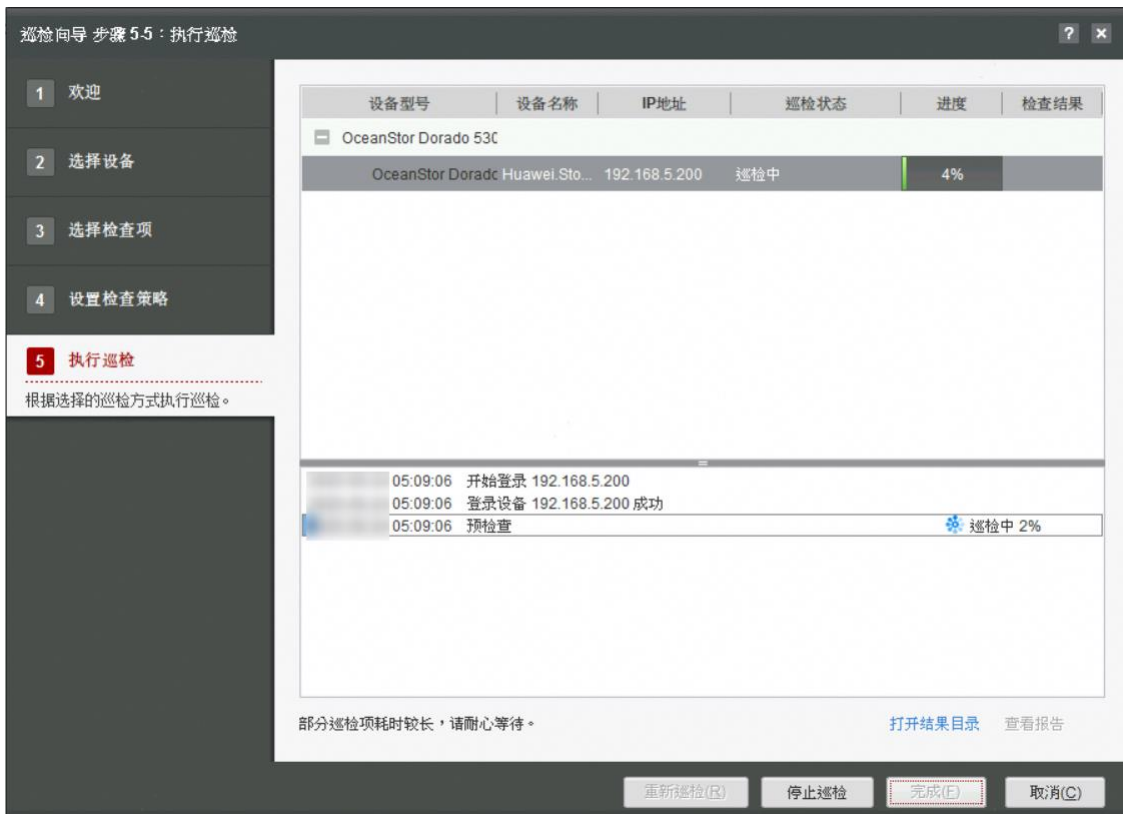
进入“巡检向导 步骤 5-3：选择检查项”界面，勾选需要的检查项，点击“下一步”。



进入“巡检向导 步骤 5-4：设置检查策略”界面，选择结果文件保存路径，点击“下一步”。



进入“巡检向导 步骤 5-5：执行巡检”界面，等待巡检完成。在巡检执行中，可看到进度以及已执行完的检查项的结果。点击已执行完的检查项，可查看详情。



巡检完成后，可看到检查结果，可打开结果目录查看报告进行分析。

2.4 思考题

问题：

SmartKit 在存储系统升级场景中如何保障操作可靠性？

参考答案：

SmartKit 通过标准化流程和风险预检保障升级可靠性：

升级前巡检：检查存储池状态、LUN 配置及兼容性，识别潜在风险（如固件版本冲突）。

分步操作引导：提供“传包 > 升级前检查 > 升级 > 升级后检查”全流程向导，避免跳步操作。

热补丁安装：支持一键式补丁安装，减少停机时间。
