



浪潮存储系统 Cinder 主机插件用户手册

文档版本 **1.9**

发布日期 **2020-09-28**

适用版本 **Cinder_V4.0.0 及以上**

尊敬的用户：

衷心感谢您选用浪潮存储系统！浪潮存储秉承“云存智用 运筹新数据”的新存储之道，致力于为您提供符合新数据时代需求的存储产品和解决方案。

本手册用于帮助您更详细地了解 and 便捷地使用存储系统，涉及的截图仅为示例，最终界面请以实际设备显示的界面为准。

由于产品版本升级或其他原因，本手册内容会不定期进行更新，如有变动恕不另行通知。除非另有约定，本手册仅作为使用指导，本手册中的所有陈述、信息和建议不构成任何明示或暗示的担保。

浪潮拥有本手册的版权，保留随时修改本手册的权利。未经浪潮许可，任何单位和个人不得以任何形式复制本手册的内容。

如果您对本手册有任何疑问或建议，请向浪潮电子信息产业股份有限公司垂询。

技术服务电话： 4008600011

地 址： 中国济南市浪潮路 1036 号
浪潮电子信息产业股份有限公司

邮 编： 250101

使用声明

在您正式使用本存储系统之前，请先阅读以下声明。只有您阅读并且同意以下声明后，方可正式开始使用本存储系统。如果您对以下声明有任何疑问，请和您的供货商联系或直接与我们联系。如您在开始使用本系统前未就以下声明向我们提出疑问，则默认您已经同意了以下声明。

1. 请不要自行拆卸本存储系统机箱及机箱内任何硬件设备。在本存储系统出现任何硬件故障或您希望对硬件进行任何升级时，请您将机器的详细硬件配置反映给我们的客户服务中心。
2. 请不要将本存储系统的设备与任何其他型号的相应设备混用。本存储系统的内存、CPU、CPU 散热片、风扇、硬盘托架、硬盘等都是特殊规格的。
3. 在使用本存储系统时遇到任何软件问题，请您首先和相应软件的提供商联系。由提供商和我们联系，以方便我们共同沟通和解决您遇到的问题。对于数据库、网络管理软件或其他网络产品的安装、运行问题，我们尤其希望您能够这样处理。
4. 上架安装本存储系统前，请先仔细阅读相关产品手册中的快速安装指南。我们致力于产品功能和性能的持续提升，部分功能及操作与手册描述可能会有所差异，但不会影响使用。如果您有任何疑问问题，请与我们的客户服务中心联系。
5. **我们特别提醒您：在使用过程中，注意对您的数据进行必要的备份。**
6. 本存储系统为 A 级产品，在生活环境中可能会造成无线电干扰，需要您对其干扰采取切实可行的措施。
7. 请仔细阅读并遵守本手册的安全声明和安全细则。
8. 本手册中涉及的各项、硬件产品的标识、名称版权归产品的相应公司拥有。

以上声明中，“我们”指代浪潮电子信息产业股份有限公司；浪潮电子信息产业股份有限公司拥有对以上声明的最终解释权。

安全声明

我们非常重视数据安全和隐私，且一如既往地严密关注产品和解决方案的安全性，为您提供更满意的服务。在您正式使用本存储系统之前，请先阅读以下安全声明。

1. 为了保护您的数据隐私，在调整存储产品用途或淘汰存储设备时，请您将存储系统软件恢复固件出厂设置、删除信息、清除日志。同时，建议采用第三方安全擦除工具对存储系统软件所在的系统盘进行全面安全擦除。
2. 您购买的存储产品业务运营或故障定位的过程中可能会获取或使用用户的某些个人数据（如告警邮件接收地址、IP 地址）。因此，您有义务根据所适用国家或地区的法律法规制定必要的用户隐私政策，并采取足够的措施以确保用户的个人数据受到充分的保护。
3. 如需获取存储系统开源软件声明，请直接联系浪潮客户服务人员。
4. 存储系统的某些安全特性需要您自行配置，如认证、传输加密、存储数据加密等，这些配置操作可能会对存储系统的性能和使用方便性造成一定影响。您可以根据应用环境，权衡是否进行安全特性配置。
5. 存储系统自带了部分用于生产、装备、返厂检测维修的接口、命令及定位故障的高级命令，如使用不当，可能会导致设备异常或者业务中断，不建议您自行使用。如需使用，请联系我们的客户服务人员。
6. 我们已全面建立产品安全漏洞应急和处理机制，确保第一时间处理产品安全问题。若您在存储产品使用过程中发现任何安全问题，或者寻求有关产品安全漏洞的必要支持，请直接联系我们的客户服务人员。

以上声明中，“我们”指代浪潮电子信息产业股份有限公司；浪潮电子信息产业股份有限公司拥有对以上声明的最终解释权。

安全细则

在使用本存储系统时，若操作不当，可能会危及您的人身安全。为避免发生意外，在正式使用本存储系统之前，请务必认真阅读以下安全细则，严格按照要求进行操作。

1. 本存储系统中的电源设备可能会产生高电压和危险电能，从而导致人身伤害。请勿自行卸下主机盖以拆装、更换系统内部的任何组件。除非另外得到我们的通知，否则只有经过我们培训的维修技术人员才有权拆开主机盖及拆装、更换内部组件。
2. 请将设备连接到适当的电源，仅可使用额定输入标签上指明的外部电源为设备供电。为保护您的设备免受电压瞬间升高或降低所导致的损坏，请使用相关的稳压设备或不间断电源设备。
3. 如果必须使用延长线缆，请使用配有正确接地插头的三芯线缆，并查看延长线缆的额定值，确保插入延长线缆的所有产品的额定电流总和不超过延长线缆额定电流限制的百分之八十。
4. 请务必使用随机配备的供电组件，如电源线、电源插座（如果随机配备）等。为了本存储系统及使用者的安全，切勿随意更换电源线缆或插头。
5. 为防止因系统漏电而造成电击危险，请务必将本存储系统和外围设备的电源电缆插入已正确接地的电源插座。在未安装接地导线及不确定是否已有适当接地保护的情况下，请勿操作和使用本存储系统，并及时与电工联系。
6. 切勿将任何物体塞入本存储系统的开孔处，否则，可能会导致内部组件短路而引起火灾或电击。
7. 请将本存储系统置于远离散热片和有热源的地方，切勿堵塞通风孔。
8. 切勿在高潮湿、高灰尘的环境中使用本存储系统，切勿让食物或液体散落在系统内部或其它组件上。
9. 使用错误型号的电池会有爆炸的危险，需要更换电池时，请先向制造商咨询并使用与制造商推荐型号相同或相近的电池。切勿拆开、挤压、刺戳电池或使其外部接点短路。不要将其丢入火中或水中，也不要暴露在温度超过 60 摄氏度的环境中。请勿尝试打开或维修电池，务必合理处置用完的电池，不要将用完的电池及可能包含电池的电路板及其它组件与其它废品放在一起。有关电池回收政策请与当地废品回收处理机构联系。

以上内容中，“我们”指代浪潮电子信息产业股份有限公司；浪潮电子信息产业股份有限公司拥有对以上内容的最终解释权。

目 录

使用声明	iii
安全声明	iv
安全细则	v
1 功能描述	1
1.1 基本介绍.....	1
1.2 约束与限制.....	2
1.3 应用场景.....	2
2 安装与部署	4
2.1 浪潮 Cinder 驱动安装.....	4
2.2 OpenStack 环境配置.....	6
2.3 All-In-One 部署示例.....	7
3 功能配置与管理.....	10
3.1 网络配置.....	10
3.2 配置存储池.....	11
3.3 配置驱动的用户身份验证信息.....	13
3.4 OpenStack 启用多路径.....	16
3.5 启用浪潮存储驱动程序.....	19
3.6 配置选项.....	21
4 故障分析与解决.....	24
5 术语&缩略语	28
附录一	29

1 功能描述

1.1 基本介绍

浪潮 Cinder 主机插件是 OpenStack 中 Cinder 的驱动，能使 Cinder 有更优的性能。Cinder 是 OpenStack 的块存储服务。它的作用是向最终用户提供在 OpenStack 的计算模块（Nova）中使用的存储资源。它通过使用逻辑卷管理（LVM）方式或者通过配置其他存储的插件形式的驱动连接其他存储设备使用。简单地说，Cinder 将块存储设备的管理进行了虚拟化，只对最终用户提供 REST API 来请求和使用存储资源，用户并不需要知道他们的存储实际上是什么类型或者是什么样的。浪潮开发了上述中的插件形式的**驱动**，通过配置浪潮的驱动可连接浪潮的存储，让 Cinder 可以使用我们的存储设备给用户所需的块存储服务。

通过配置浪潮的驱动，Cinder 可以使用浪潮存储给 OpenStack 提供块存储服务，并且可将 Cinder 的卷、快照、一致性组的操作在浪潮存储中实现。详细功能如下：

表 1-1 Cinder 实现的功能

序号	功能模块	操作
1	卷操作	创建卷
2		从已有卷创建卷（克隆）
3		扩展卷
4		删除卷
5	卷-虚拟机操作	挂载卷到虚拟机实例
6		从虚拟机实例分离卷
7	卷-快照操作	创建卷的快照
8		从已有卷快照创建卷
9		删除快照
10	卷-镜像操作	从镜像创建卷

11		从卷创建镜像
12	卷-一致性组操作	创建/删除一致性组
13		一致性组快照操作
14	其他操作	Qos
15		卷迁移
16	存储接口协议	iSCSI
17		FC

1.2 约束与限制

浪潮 Cinder 主机插件，并没有集成在所有的 OpenStack 发布版本中。当所使用的 OpenStack 发布版本中未包含浪潮 Cinder 主机插件驱动时，需要手动安装随机光盘中所带的 Cinder 插件，才可以与 OpenStack 环境进行集成。

随机光盘中所带的 Cinder 主机插件提供了对 OpenStack 平台的 Juno、Kilo、Liberty、Mitaka、Newton、Ocata、Pike、Queens 等版本的兼容性支持。

适用产品型号：

AS2150G2&AS2200G2&AS2600G2&AS5300G2&AS5500G2&AS5600G2&AS5800G2&AS6800G2

HF5500

AS2600G2-F&AS5300G2-F&AS5500G2-F&AS5600G2-F&AS5800G2-F&AS6800G2-F

AS5300G5&AS5500G5&AS5600G5&AS5800G5&HF5000G5&HF6000G5

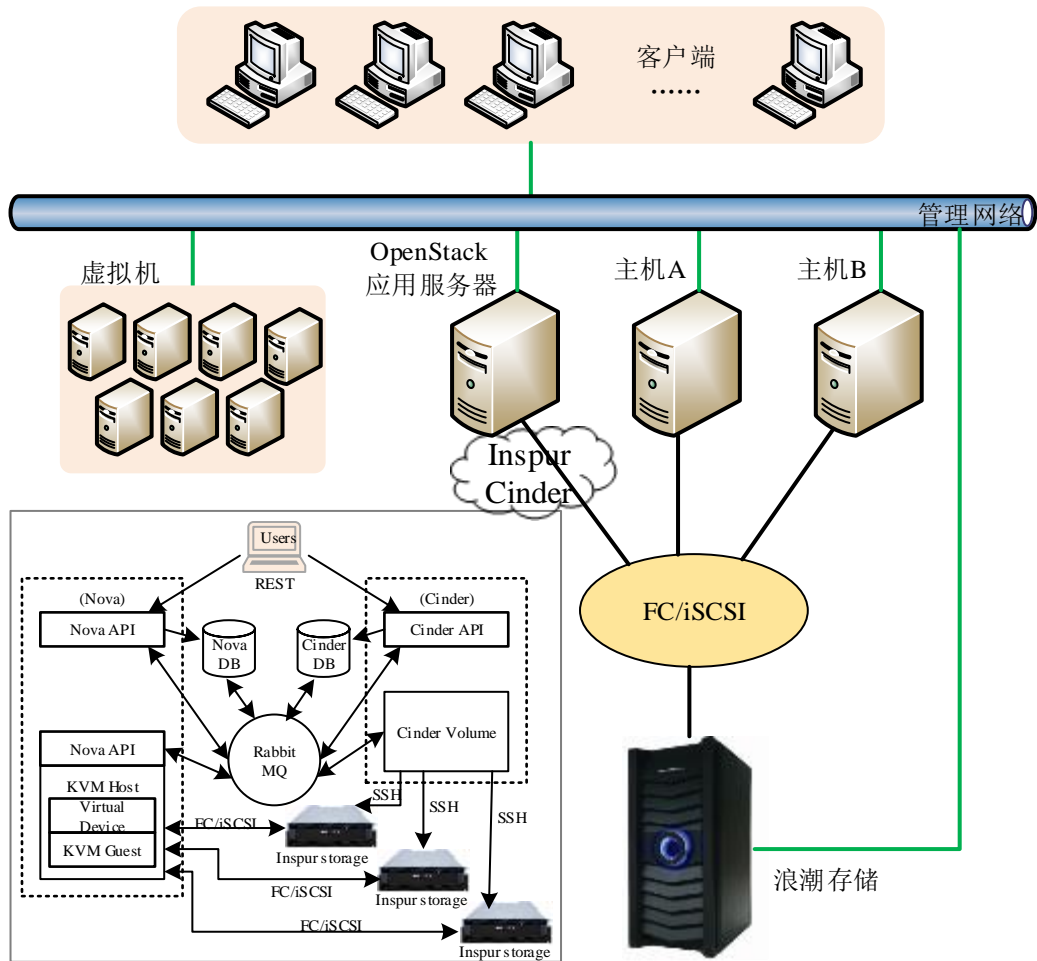
AS5500G5-C

1.3 应用场景

浪潮 Cinder 主机插件是 OpenStack 中 Cinder 的驱动，通过配置浪潮的驱动，能使 Cinder 有更优的性能。Cinder 可以使用浪潮存储给 OpenStack 提供块存储服务、并且可将 Cinder 的卷、快照、一致性组的操作在浪潮存储中实现。拓扑图如图 1-1 所

示。

图 1-1 Cinder 主机插件应用拓扑图



2 安装与部署

浪潮 Cinder 主机插件是 OpenStack 中 Cinder 服务的驱动，浪潮的驱动程序集成于 OpenStack 的发布版本中，存在于 Cinder 服务模块的代码目录中，具体路径为 `cinder/volume/drivers/inspur/instorage`，包含 iSCSI 和 FC 两种存储接口协议的驱动，分别为 `instorage_iscsi` 模块中的 `InStorageMCSISCSIDriver` 类和 `instorage_fc` 模块中的 `InStorageMCSFCDriver` 类。浪潮 Cinder 驱动会跟随 OpenStack 版本发布，但并非所有 OpenStack 版本均包含浪潮 Cinder 驱动，当 OpenStack 版本中未包含浪潮 Cinder 驱动或者所包含的浪潮 Cinder 驱动版本过低时，用户可以使用随机光盘中的发布版本进行安装，并根据用户需要进行相关配置即可。

2.1 浪潮 Cinder 驱动安装

当所部署的 OpenStack Cinder 服务中未包含浪潮 Cinder 驱动时，或者所部署的 OpenStack Cinder 服务中包含的浪潮 Cinder 驱动版本过低时，可以通过手动方式安装随机光盘中包含的 Cinder 驱动，过程如下。

1. 检查 Cinder 服务中是否包含浪潮 Cinder 驱动。

在 Cinder 服务安装目录中检查是否存在 `cinder/volume/drivers/inspur` 目录，如果存在，说明 Cinder 服务已经包含了浪潮 Cinder 驱动。

2. 打开浪潮存储随机光盘中所携带的 Cinder 驱动安装包。

浪潮 Cinder 驱动安装包中包含如下目录和文件：

```
ci@devstack-32:~/Cinder_V4.0.0.Build20180301$ tree
.
├── cinder
│   ├── opts-mitaka.py
│   ├── opts-newton.py
│   ├── opts-ocata.py
│   ├── opts-pike.py
│   ├── opts.py
│   ├── opts-queens.py
│   └── volume
│       └── drivers
│           └── inspur
│               ├── __init__.py
│               └── instorage
│                   ├── __init__.py
│                   ├── instorage_common.py
│                   ├── instorage_const.py
│                   ├── instorage_fc.py
│                   ├── instorage_iscsi.py
│                   └── replication.py
├── instorage_mpio_cfg.sh
├── mkpackage.sh
└── README.md

5 directories, 16 files
```

3. 进入驱动安装包目录，根据所部署 OpenStack 版本执行 `mkpackage.sh` 命令，生成对应 OpenStack 版本的驱动包，新生成的驱动包与 `mkpackage.sh` 同级，目录名为 `InStorage_XXX_cinder`。`mkpackage.sh` 工具通过 `-t` 参数来指定 OpenStack 版本，参数值为具体 OpenStack 版本的代码首字母，当前支持 `j/k/l/m/n/o/p/q` 等版本，以 `Ocata` 版本为例，具体过程如下：

```
ci@devstack-32:~/Cinder_V4.0.0.Build20180301$ ./mkpackage.sh -t o
Generate Package for OpenStack version OCATA
ci@devstack-32:~/Cinder_V4.0.0.Build20180301$ ls
cinder  instorage_mpio_cfg.sh  InStorage_OCATA_cinder  mkpackage.sh  README.md
```

4. 进入生成的 `InStorage_XXX_cinder` 目录，目录包含了对应版本的驱动文件，其中 `opts.py` 文件从 OpenStack Mitaka 版本开始存在。

```
ci@devstack-32:~/Cinder_V4.0.0.Build20180301$ cd InStorage_OCATA_cinder/
ci@devstack-32:~/Cinder_V4.0.0.Build20180301/InStorage_OCATA_cinder$ tree
.
├── inspur
│   ├── __init__.py
│   └── instorage
│       ├── __init__.py
│       ├── instorage_common.py
│       ├── instorage_const.py
│       ├── instorage_fc.py
│       ├── instorage_iscsi.py
│       └── replication.py
├── opts.py
└── README.md
2 directories, 9 files
```

5. 将该驱动包中的 `inspur` 目录拷贝到 Cinder 服务安装目录下的 `cinder/volume/drivers` 目录。
6. 用该驱动包中的 `opts.py` 文件替换 Cinder 服务安装目录中的 `cinder/opts.py` 文件。
7. 至此，浪潮 Cinder 驱动安装完毕，后续可参考功能配置与管理部分对驱动进行合理配置并重新启动服务，其中 `README` 文件中包含了简单的使用范例。

2.2 OpenStack 环境配置

由于 OpenStack 的开源性，所以目前市面上有许多不同的 OpenStack 发行版，以及不同规模的部署方式：

- All-In-One 方式，使用 Devstack 将所需要的各个 OpenStack 组件安装在一台物理机或虚拟机上，一般用于开发测试；
- Mirantis Fuel 方式，通过相对直观的图形化界面，定制自己的 OpenStack 环境，可通过简单的配置各个节点参数，一键部署 OpenStack 环境。

说明： 相关信息请参考 <https://wiki.openstack.org/wiki/Fuel>。

原生部署，OpenStack 官网有各个组件的部署教程，用户可根据自己的需求部署自己的环境，根据自己的情况，可将各个服务部署在自己指定的节点，或者将所有服务安装在单独的节点上，具体安装教程请参考：<https://docs.openstack.org/project-install-guide/newton/>。

各个第三方 OpenStack 云平台服务提供商，目前市面上有不少公司提供 OpenStack 服务，包括云平台的部署、管理、维护等，他们通过对 OpenStack 的二次开发，为

客户提供自己定制的 OpenStack 服务。

2.3 All-In-One 部署示例



注意

由于 OpenStack 环境的复杂性，本手册提供开发测试环境的搭建示例，仅供参考，实际生产环境请按 OpenStack 环境运维人员要求部署。

本测试环境使用 Devstack 部署方式进行部署。采用虚拟机单节点安装，使用配置 4VCPU, 8GB 内存, 60GB 硬盘，需要虚拟机能够访问互联网，建议使用桥接模式。

- Ubuntu Server 16.04 最小化安装，并选择安装 openSSH server。
- 创建 stack 用户，并给定 stack 用户 root 权限。
- `echo "stack ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers;`

或者直接编辑/etc/sudoers 找到 `root ALL=(ALL:ALL) ALL` 这一行，在下边添加如下：`stack ALL=(ALL:ALL) ALL`

- 配置固定 IP，修改/etc/network/interfaces

```
auto eth0

iface eth0 inet static (网卡 IP 地址为固定配置)
address 1.2.3.4 (选择合适的 IP 地址)
netmask 255.255.255.0 (选择合适的子网掩码)
gateway 1.2.3.254 (选择合适的网关)
```

- 配置可用的 DNS，修改 /etc/resolvconf/resolv.conf.d/base

```
nameserver 8.8.8.8 (选择合适的 DNS)
```

- 更新软件并安装 GIT，执行以下命令：

```
sudo apt-get update (更新软件列表)
sudo apt-get upgrade (更新软件)
apt-get install git -y (安装 git)
```

- 下载并安装 devstack

1. 执行命令：

```
git clone https://git.openstack.org/openstack-dev/devstack
```

说明：示例采用了最新的稳定版本 newton 并使用了国内的 trystack 镜像源。

2. 执行命令：

```
git clone http://git.trystack.cn/openstack-dev/devstack.git -b stable/newton
```

devstack 已经不支持使用 root 用户运行，需要创建 stack 用户运行。修改 devstack 目录权限，让 stack 用户可以运行以下命令：

```
chown -R stack /home/stack/devstack  
chmod 777 /dev/pts/0
```

3. 打开 devstack 目录，创建 local.conf 文件并根据如下地址配置所需参数：

<https://docs.openstack.org/developer/devstack/configuration.html>

也可通过 wget 命令来获取本示例采用的配置文件，执行以下命令：

```
wget -O - https://raw.githubusercontent.com/inspur-  
storage/devstack/master/local.conf > /home/stack/devstack/local.conf
```

4. 修改 HOST_IP 为虚拟机 IP，Credentials 信息为 Neutron 中的 FLOATING_RANGE ， Q_FLOATING_ALLOCATION_POOL ， PUBLIC_NETWORK_GATEWAY。

5. 执行 stack.sh 脚本以安装 devstack:

```
./stack.sh
```

安装过程中，可能会遇到错误，一般都是网络问题，执行 unstack.sh 脚本，再执行 stack.sh 脚本即可。

安装完成后如图 2-1 所示。

图 2-1 devstack 安装成功

```
=====
DevStack Component Timing
=====
Total runtime          1732
run_process            92
test_with_retry        5
apt-get-update         16
pip_install            283
restart_apache_server  10
wait_for_service       18
apt-get                20
=====

This is your host IP address: 10.180.207.101
This is your host IPV6 address: fe80::20c:29ff:fe25:e7bf
Horizon is now available at http://10.180.207.101/dashboard
Keystone is serving at http://10.180.207.101/identity/
The default users are: admin and demo
The password: 111111
stack@devstack:/home/devstack$ █
```

6. 这时可登陆虚拟机 IP 访问 OpenStack 的 Dashboard 使用 OpenStack 了。

图 2-2 OpenStack Dashboard 登录页面



The image shows the OpenStack Dashboard login page. At the top center is the OpenStack logo, a red square with a white 'O' inside. Below the logo is the text 'openstack.' in a bold, sans-serif font. Underneath that is the Chinese word '登录' (Login). There are two input fields: the first is labeled '用户名' (Username) and the second is labeled '密码' (Password). The password field has a small eye icon to its right, indicating it can be toggled between visible and hidden. At the bottom right of the form is a blue button with the Chinese characters '连接' (Connect).

3 功能配置与管理

为了在 Cinder 服务中使用浪潮存储，首先需要对 cinder-volume 服务进行配置，增加浪潮存储的相关配置。

3.1 网络配置

管理 IP

驱动需要访问 Inspur 存储的管理接口，驱动使用 SSH 的方式与管理接口通信。驱动需要配置浪潮存储系统的 IP、SSH 端口（默认为 22）、还可以设置辅助管理 IP。上述三个信息通过在/etc/cinder/cinder.conf 中配置 san_ip、san_ssh_port、instorage_san_secondary_ip。



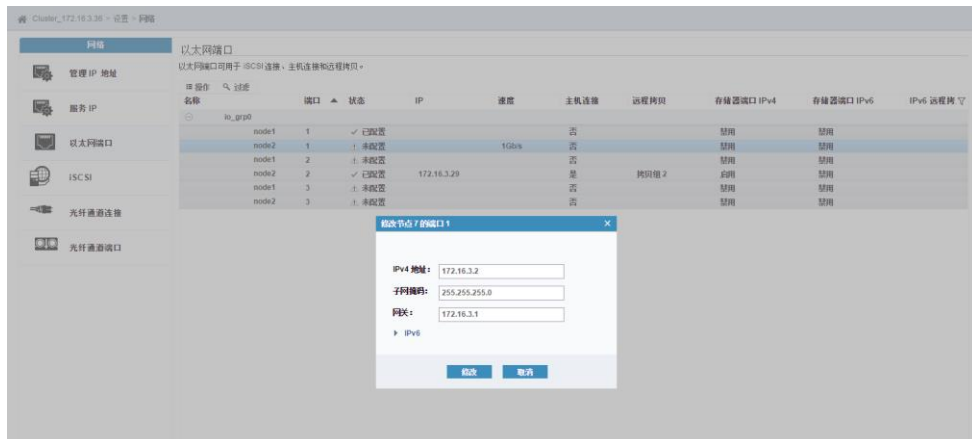
注意

- 确保运行 Cinde-volume 管理驱动程序 Cinder 节点具有到存储系统的 SSH 网络访问。
- Inspur 存储设备必须配置有 iSCSI、FC，两者至少有一种，或者两者兼有。

iSCSI 网络配置

如果使用 iSCSI,则需要每个浪潮存储节点至少有一个 iSCSI 的 IP 地址。浪潮 Cinder 主机插件（驱动）会优先使用卷的优先节点（如果存在）的 iSCSI 的 IP 来挂载卷到实例，否则会使用系统的第一个可用的 iSCSI 的 IP。驱动会直接从存储系统中获取 iSCSI IP，用户不需要给驱动单独提供 iSCSI IP。虽然不需要单独给驱动分配 iSCSI IP，但是要在存储管理系统中设定 iSCSI 端口的 IP，在“设置 > 网络 > 以太网端口”中，选择已经连通的端口，单击鼠标右键，选择“修改 IP”，填写有效的 IP、子网掩码、网关。如图 3-1 所示。

图 3-1 添加 iSCSI 端口 IP

**注意**

如果使用 iSCSI，确保 compute 节点与存储系统有畅通的 iSCSI 网络可供访问。

FC 网络配置

如果使用 FC，则需要每个浪潮存储节点至少配置有一个 WWPN 端口。驱动会使用所有可用的 WWPN 端口将卷挂载到虚拟机实例。驱动会直接从存储系统中获取 WWPN，用户不需要给驱动单独提供 WWPN。

**注意**

如果使用 FC，确保 compute 节点与存储系统有 FC 连接。

3.2 配置存储池

驱动可以在多个池中分配卷，通过在 `cinder.conf` 中 `instorage_mcs_volpool_name` 后配置，并且池和池用逗号分隔。详情参考配置表。驱动配置的存储池是应该事先创建好的，存储池可以在浪潮存储的管理系统中创建并添加存储器，具体操作如图 3-2 图 3-3、图 3-4、图 3-5 所示。

图 3-2 存储池操作



图 3-3 创建存储池

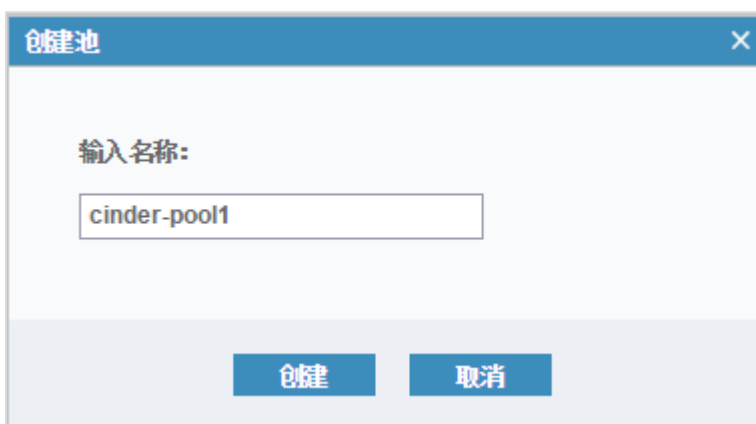


图 3-4 存储池列表

名称	状态	容量
cinder-pool1	✓ 联机	<input type="text"/> 0 B / 5.45 TiB (0%)
Pool0	✓ 联机	<input type="text"/> 0 B / 59.00 GiB (0%)

图 3-5 添加存储器到池



3.3 配置驱动的用户身份验证信息

为了让驱动能与浪潮存储设备进行通信，需要给驱动提供一个账户。驱动有两种验证方式，一种是口令认证，一种是 SSH 密钥方式。建议用户通过管理员创建单独的管理驱动程序的账户，并且确保账户属于管理员角色。

1. 如果使用口令认证，需要在 `/etc/cinder/cinder.conf` 中 `san_login`、`san_password` 配置用户名和密码。
2. 如果使用 SSH 密钥方式，需要在主机侧创建密钥，可通过执行 `ssh-keygen -t rsa` 创建密钥。将 key 文件内容配置到 `/etc/cinder/cinder.conf` 中的 `san_private_key` 中，将 `key.pub` 上传到 Inspur 存储的管理系统中去。

具体操作方法如下：

- 1) 首先要在要配置 Cinder 驱动的环境中生成 key 文件，执行 `ssh-keygen -t rsa`。

图 3-6 生成密钥文件 1

```

stack@devstack-102:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/stack/.ssh/id_rsa):
Created directory '/home/stack/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/stack/.ssh/id_rsa.
Your public key has been saved in /home/stack/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:WNwYmrSiQn6hDmc2uVDj6VyaTvsnsMNHGADcKT3v9TQ stack@devstack-102
The key's randomart image is:
+---[RSA 2048]-----+
|+ 0 . . . |
|.o = . = + |
| 0+. + + + . |
|000=.o + E |
|+oX+o 0 S . |
|o0+*.. . |
| oB+ |
| 0+.o . |
| 0+.o |
+-----[SHA256]-----+
stack@devstack-102:~$ █

```

- 2) 打开 key 文件所在位置:

图 3-7 生成密钥文件 2

```

stack@devstack-102:~$ cd /home/stack/.ssh/
stack@devstack-102:~/.ssh$ ls
id_rsa id_rsa.pub
stack@devstack-102:~/.ssh$ █

```

- 3) 保存公钥文件（即图中 id_rsa.pub 文件）到一个合适的地方，以便后续上传到存储管理系统中。
- 4) 配置 cinder.conf 文件，将私钥的存放位置填写到 san_private_key。

图 3-8 配置驱动密钥

```

[indigo-driver]
volume_driver = cinder.volume.drivers.inspur.instorage_mcs_iscsi.InStorageMCSISCSIDriver
san_ip = 172.16.3.38
san_login = superuser
san_private_key = /home/stack/.ssh/id_rsa
instorage_mcs_volpool_name = Pool0
volume_backend_name = indigo-driver

```

- 5) 打开浪潮存储管理系统的“访问 > 用户”页面，选择指定的用户并双击，打开配置对话框，将公钥导入。

图 3-9 导入公钥到管理系统中

6) 导入过后如图所示：

图 3-10 导入密钥成功

名称	用户组	密码	SSH 密钥
grsuser	SecurityAdmin	已配置	否
grsuser1	SecurityAdmin	已配置	否
mytest	SecurityAdmin	已配置	否
superuser	SecurityAdmin	已配置	是

3.4 OpenStack 启用多路径

为了提升 SAN 存储卷的可靠性，在生产环境中，通常会启用多路径。

在 OpenStack 环境中，如果多路径相关配置设置不恰当，在实际使用中有可能使用到单路径设备，如果该路径出现损坏，则可能会产生 I/O 错误，导致虚拟机文件系统变为只读或损坏。

在 Linux 环境下，浪潮存储利用 Linux 系统自带的 multipathd 服务进行多路径聚合，如果需要启用多路径，首先需要保证 OpenStack 各节点均按照要求部署安装 multipathd 服务。然后可以参考下面的描述，对 OpenStack 环境中的各节点多路径相关的配置进行设置，来正确的使用多路径。

1. 在 nova.conf 配置中正确配置虚拟机使用多路径

nova.conf 配置文件中 libvirt 配置组下，启用 volume_use_multipath 选项（旧版本 openstack 选项名称为 iscsi_use_multipath），值设置为 true。

使能该配置选项后，在卷挂载，卸载过程中，openstack 会使用聚合后的多路径设备为上层虚拟机提供设备。

```
[libvirt]
volume_use_multipath = true
或
iscsi_use_multipath = true
```

说明：

nova.conf 配置文件的具体路径以及配置项名称，请咨询 OpenStack 云提供商。

2. 在 cinder.conf 配置中正确设置启动多路径

cinder.conf 配置文件中每个存储后端均可以通过将 use_multipath_for_image_xfer 选项设置为 true，来启用多路径。当启用多路径后，在通过镜像创建卷过程中，会使用映射到控制节点上的多路径设备进行镜像写入。具体配置可参考[启用浪潮存储驱动程序](#)章节中的配置描述。

说明：

cinder.conf 配置文件的具体路径，请咨询 OpenStack 云提供商。

3. 在 multipath.conf 配置中正确设置设备黑名单

在 openstack 环境中，一个计算节点上，会同时挂载非常多的卷，每个卷又会

有多条路径，从而使得节点上的 sdX 设备会非常多。多路径的配置文件中 blacklist 配置组中的参数会过滤符合 blacklist 条件的路径。如果该配置过滤的设备不恰当，可能会导致部分路径无法进行多路径聚合。

例如，如果 devnode 参数配置成类似“^sda”时，会导致 sdaa,sdab 等等路径不会进行路径聚合，导致路径缺失，正确的应该是“^sda\$”，即仅将 sda 设备屏蔽掉。

针对 multipath.conf 配置文件的具体用法，Linux 用户可以通过 man 5 multipath.conf 获取帮助。请确保不会将浪潮存储提供的设备路径加入设备黑名单。

说明：

multipath.conf 配置文件的具体路径，请咨询 OpenStack 云提供商。

4. 在 multipath.conf 配置中正确设置设备名称选项

openstack 在使用多路径后，在卸载多路径卷时，使用的是 wwn 名称的设备进行 flush 操作。但是 multipath -f 后面指定的即使是不正确的设备时，该命令仍然会执行成功，会给人一种错误的现象，以为设备已经正常 flush 了。但是实际上数据可能并没有真的全部刷到存储上，导致多路径设备无法正常卸载，产生异常。可以通过将 defaults 选项组中的 user_friendly_names 选项设置为 no，来使得多路径设备使用设备的 wwn 进行命名。

```
defaults {  
    user_friendly_names "no"  
}
```

针对 multipath.conf 配置文件的具体用法，Linux 用户可以通过 man 5 multipath.conf 获取帮助。

说明：

multipath.conf 配置文件的具体路径，请咨询 OpenStack 云提供商。

5. 在 multipath.conf 配置文件中，正确设置浪潮存储推荐的设备配置参数

目前浪潮存储的推荐多路径配置已经合入到多路径工具的社区版本中，针对使用旧版本多路径工具的场景，需要在配置文件中加入浪潮推荐的多路径配置。即在 devices 配置组中增加浪潮存储的 device 配置内容。通常配置信息如下：

```
devices{
```



```
device {  
    vendor          "INSPUR"  
    product         "MCS"  
    path_grouping_policy  group_by_prio  
    path_selector    "round-robin 0"  
    path_checker     tur  
    features         "1 queue_if_no_path"  
    hardware_handler "0"  
    prio            alua  
    failback        immediate  
    rr_weight       uniform  
    rr_min_io       1000  
}  
}
```

针对 `multipath.conf` 配置文件的具体用法，Linux 用户可以通过 `man 5 multipath.conf` 获取帮助。

说明：

- `multipath.conf` 配置文件的具体路径，请咨询 OpenStack 云提供商。
- `devices` 配置组中可能会存在多个存储厂商的 `device` 配置。

6. 在 `lvm.conf` 配置文件中正确设置设备过滤

在制作 linux 虚拟机镜像时，由于 linux 操作系统在安装时，通常会默认使用 LVM 对磁盘进行管理，使得创建出来的 linux 系统镜像的系统盘，通常都是有 LVM 信息。通常来说，主机上也会使用 LVM 管理器对主机上的磁盘进行管理。这样，当客户机的系统盘挂载到主机上以后，由于主机的 LVM 管理器并没有办法区分这个磁盘是主机的还是客户机的，主机的 LVM 管理器就可能会对该磁盘进行管理，为了避免客户机上的信息被主机 LVM 管理器识别，需要对主机的 LVM 管理器的配置文件进行修改，通过修改 `lvm.conf` 配置文件中的 `filter`, `global_filter` 配置项，确保 LVM 管理器不会对客户机的磁盘进行管理。

针对 `lvm.conf` 配置文件的具体用法，Linux 用户可以通过阅读 `lvm.conf` 配置文件中的注释信息来获取帮助。请确保由浪潮存储提供的卷不会被主机上的 LVM 管理器进行管理。

说明：

lvm.conf 配置文件的具体路径，请咨询 OpenStack 云提供商。

7. 正确设置 multipathd 服务 service 文件中的启动超时时间

当使用主机 multipathd 服务时，在该主机进行重启时，由于 multipathd 服务需要对所有的已映射卷进行识别并聚合，如果已映射的卷非常多，路径非常多，会导致 multipathd 服务使用大量的时间来处理路径聚合，导致启动过程占用时间长，导致操作系统 systemd 管理服务认为 multipathd 超时，并将其杀死。（默认超时时间 90s，如果卷数量非常多，可能会超过这个时间，就会被系统杀死）。如果有这种情况，可以通过设置 multipath 的 service 文件，在[Service]配置节中增加 TimeoutSet=900 配置，增加处理超时时间，避免被 systemd 服务超时杀死。

```
...  
[Service]  
...  
TimeoutSet=900  
ExecStartPre=/sbin/modprobe dm-multipath  
ExecStart=/sbin/multipathd -d -s  
...
```

说明：

multipathd 服务的 systemd service 配置文件的具体路径，请咨询 OpenStack 云提供商。如果 multipathd 通过容器形式进行安装，可以不用设置该参数。

3.5 启用浪潮存储驱动程序

修改 Cinder 服务配置文件/etc/cinder/cinder.conf，在配置文件中增加浪潮存储设备作为存储后端的配置部分。如下文件所示：

iSCSI:

```
[mcs1234]  
volume_driver =  
cinder.volume.drivers.inspur.instorage.instorage_iscsi.InStorageMCSISCSIDriver  
san_ip = 1.2.3.4
```

```
san_login = superuser
san_password = passw0rd
instorage_mcs_volpool_name = cinder_pool1
volume_backend_name = mcs1234
use_multipath_for_image_xfer = True
```

FC:

```
[mcs1234]
volume_driver =
cinder.volume.drivers.inspur.instorage.instorage_fc.InStorageMCSFCDriver
san_ip = 1.2.3.4
san_login = superuser
san_password = passw0rd
instorage_mcs_volpool_name = cinder_pool1
volume_backend_name = mcs1234
use_multipath_for_image_xfer = True
```

如果使用 replication 需要在驱动配置下配置相关的参数，下边为 replication 的配置示例：

Replication 配置:

在存储后端的配置部分，增加 replication_device 配置参数，指定 replication 中所使用的另一个存储：

```
[mcs1234]
volume_driver =
cinder.volume.drivers.inspur.instorage.instorage_iscsi.InStorageMCSISCSIDriver
san_ip = 1.2.3.4
san_login = superuser
san_password = passw0rd
instorage_mcs_volpool_name = cinder_pool1
volume_backend_name = mcs1234
use_multipath_for_image_xfer = True
replication_device = backend_id:rep_mcs,
                    san_ip:1.2.3.5,
                    san_login:superuser,
                    san_password:passw0rd,
```

```
pool_name:cinder_pool1
```

其中 `backend_id` 是远端存储系统的唯一名称。`san_ip`、`san_login`、`san_password` 为远端存储系统的认证信息。`pool_name` 为复制目标卷所在池的名称。

说明：

参数 `use_multipath_for_image_xfer` 是 `openstack` 内置的配置选项，用于在控制节点启用多路径。

参考“[OpenStack 启用多路径](#)”章节中的相关说明了解启用多路径所需的相关配置。

配置存储后端

在完成各存储后端的配置部分以后，需要修改 `[DEFAULT]` 配置部分中的 `enabled_backends` 参数，将需要启用的存储后端名称作为该参数的值，该参数支持多个存储后端名称，多个名称之间采用“，”号分割。

```
[DEFAULT]
...
enabled_backends = mcs1234,mcs1235
...
```

完成 `/etc/cinder/cinder.conf` 配置参数的修改以后，需要重新启动 `cinder-volume` 服务，可以使用 `ps` 命令查询 `cinder-volume` 进程号，采用 `kill` 命令杀死相应进程，并重新启动。

3.6 配置选项

表 3-1 浪潮存储驱动程序配置信息的列表

标志名称	类型	默认	描述
<code>san_ip</code>	必须	-	管理 IP 或主机名
<code>san_ssh_port</code>	可选	22	管理端口
<code>san_login</code>	必须	-	管理登录用户名
<code>san_password</code>	必须[1]	-	管理登录密码
<code>san_private_key</code>	必须[1]	-	管理登录 SSH 私钥

instorage_mcs_volpool_name	必须	-	默认的卷所在的存储池
instorage_mcs_vol_rsize	可选	2	初始物理分配（百分比） [2]
instorage_mcs_vol_warning	可选	0（禁用）	空间分配警告阈值（百分比）
instorage_mcs_vol_autoexpand	可选	TRUE	启用或禁用自动扩展卷 [3]
instorage_mcs_vol_grainsize	可选	256	卷颗粒度（KB）
instorage_mcs_vol_compression	可选	FALSE	启用或禁用实时压缩 [4]
instorage_mcs_vol_intier	可选	TRUE	启用或禁用自动分层 [5]
instorage_mcs_vol_iogrp	可选	0	给虚拟磁盘分配的 I/O 组
instorage_mcs_localcopy_timeout	可选	120	本地复制超时阈值 [6]（秒）
instorage_mcs_iscsi_chap_enabled	可选	TRUE	配置 iSCSI 连接 CHAP 认证
instorage_mcs_multihost_enabled	可选	TRUE	启用虚拟磁盘映射到多个主机 [7]
instorage_mcs_vol_nofmtdisk	可选	FALSE	启用或禁用快速格式 [8]
max_over_subscription_ratio	可选	20	预配置容量，自动精简配置卷所涉及的比值 [9]

[1] 认证要求密码（san_password）或 SSH 私钥（san_private_key）。必须指定其中一个，如果同时指定，则驱动程序只使用 SSH 私钥。

[2] 该驱动程序默认创建精简配置卷。instorage_mcs_vol_rsize 参数配置精简卷初始物理分配比，如果设置为 -1，则驱动程序会创建完整大小的卷。

[3] 定义精简卷是否可以由存储系统自动扩展，True 表示已启用自动扩展，False 禁用自动扩展。

[4] 定义 OpenStack 创建卷时是否开启实时压缩。并且存储设备开启此服务时此功

能才能使用。

- [5] 定义 OpenStack 创建卷时是否开启自动分层。并且存储设备开启此服务时此功能才能使用。
- [6] 创建 OpenStack 快照时的驱动程序等待超时的阈值，这实际上是驱动程序等待存储设备准备新的 LocalCopy 映射的最长时间。驱动程序接受的最大等待时间为 600 秒（10 分钟）。
- [7] 此选项用于定义驱动程序是否允许一次将虚拟磁盘映射到多个主机。这种情况在迁移已经挂载虚拟机的卷的时候发生；卷将同时映射到源计算主机和目标计算主机。如果不需要将虚拟磁盘附加到多个主机，将此标志设置为 **False** 将更安全。
- [8] 定义在创建时是否禁用配置普通卷（非精简卷）的快速格式化。默认值为 **False**，值为 **True** 表示禁用快速格式。
- [9] 此选项允许所有卷的预配置容量的总和大于池的总容量。默认值为 20.0，这意味着预配置的容量可以是总物理容量的 20 倍。

4 故障分析与解决

1. 网络不通，当 cinder-volume 所在存储节点与浪潮存储设备之间的网络不通时，会报如下错误，无法连接到存储 IP，这时需要检查网络或 IP 是否正确。

图 4-1 网络不通时提示错误信息

```
2016-11-14 23:54:18.870 TRACE cinder.volume.manager File "/opt/stack/cinder/cinder/ssh_utils.py", line 153,
in create
2016-11-14 23:54:18.870 TRACE cinder.volume.manager raise paramiko.SSHException(msg)
2016-11-14 23:54:18.870 TRACE cinder.volume.manager SSHException: Error connecting via ssh: [Errno None] Unab
le to connect to port 22 on 172.16.3.38
2016-11-14 23:54:18.870 TRACE cinder.volume.manager
2016-11-14 23:54:18.885 DEBUG cinder.service [req-34f19fc7-0f04-4fda-88f3-cc0fe4cb49e1 None None] Creating RP
C server for service cinder-volume from (pid=1067) start /opt/stack/cinder/cinder/service.py:236
2016-11-14 23:54:18.893 DEBUG oslo_db.sqlalchemy.engines [req-34f19fc7-0f04-4fda-88f3-cc0fe4cb49e1 None None]
MySQL server mode set to STRICT_TRANS_TABLES,STRICT_ALL_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISI
ON_BY_ZERO,TRADITIONAL,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION from (pid=1067) _check_effective_sql_mode /
usr/local/lib/python2.7/dist-packages/oslo_db/sqlalchemy/engines.py:261
2016-11-14 23:54:18.900 DEBUG cinder.service [req-34f19fc7-0f04-4fda-88f3-cc0fe4cb49e1 None None] Pinging obj
ect versions for RPC server serializer to 1.11 from (pid=1067) start /opt/stack/cinder/cinder/service.py:244
2016-11-14 23:54:18.929 INFO cinder.volume.manager [req-34f19fc7-0f04-4fda-88f3-cc0fe4cb49e1 None None] Initi
alizing RPC dependent components of volume driver InStorageMCSFCDriver (1.0)
2016-11-14 23:54:18.930 ERROR cinder.utils [req-34f19fc7-0f04-4fda-88f3-cc0fe4cb49e1 None None] Volume driver
InStorageMCSFCDriver not initialized
```

2. 认证失败，无法连接到浪潮存储设备，请检查用户名、密码或 ssh-key。

图 4-2 认证失败时提示错误信息

```
2016-11-15 00:09:43.007 TRACE cinder.volume.manager File "/opt/stack/cinder/cinder/ssh_utils.py", line 99,
in __init__
2016-11-15 00:09:43.007 TRACE cinder.volume.manager super(SSHPool, self).__init__(*args, **kwargs)
2016-11-15 00:09:43.007 TRACE cinder.volume.manager File "/usr/local/lib/python2.7/dist-packages/eventlet/p
ools.py", line 77, in __init__
2016-11-15 00:09:43.007 TRACE cinder.volume.manager self.free_items.append(self.create())
2016-11-15 00:09:43.007 TRACE cinder.volume.manager File "/opt/stack/cinder/cinder/ssh_utils.py", line 153,
in create
2016-11-15 00:09:43.007 TRACE cinder.volume.manager raise paramiko.SSHException(msg)
2016-11-15 00:09:43.007 TRACE cinder.volume.manager SSHException: Error connecting via ssh: Authentication fa
iled.
2016-11-15 00:09:43.007 TRACE cinder.volume.manager
2016-11-15 00:09:43.154 DEBUG paramiko.transport [-] EOF in transport thread from (pid=3521) _log /usr/local/
lib/python2.7/dist-packages/paramiko/transport.py:1563
2016-11-15 00:09:43.155 DEBUG cinder.service [req-c140318d-e09b-471c-af7f-adc29b831b57 None None] Creating RP
C server for service cinder-volume from (pid=3521) start /opt/stack/cinder/cinder/service.py:236
2016-11-15 00:09:43.164 DEBUG oslo_db.sqlalchemy.engines [req-c140318d-e09b-471c-af7f-adc29b831b57 None None]
MySQL server mode set to STRICT_TRANS_TABLES,STRICT_ALL_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISI
ON_BY_ZERO,TRADITIONAL,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION from (pid=3521) _check_effective_sql_mode /
usr/local/lib/python2.7/dist-packages/oslo_db/sqlalchemy/engines.py:261
2016-11-15 00:09:43.174 DEBUG cinder.service [req-c140318d-e09b-471c-af7f-adc29b831b57 None None] Pinging obj
ect versions for RPC server serializer to 1.11 from (pid=3521) start /opt/stack/cinder/cinder/service.py:244
2016-11-15 00:09:43.201 INFO cinder.volume.manager [req-c140318d-e09b-471c-af7f-adc29b831b57 None None] Initi
alizing RPC dependent components of volume driver StorwizeSVCFCDriver (2.1.1)
2016-11-15 00:09:43.202 ERROR cinder.utils [req-c140318d-e09b-471c-af7f-adc29b831b57 None None] Volume driver
StorwizeSVCFCDriver not initialized
2016-11-15 00:09:43.202 ERROR cinder.volume.manager [req-c140318d-e09b-471c-af7f-adc29b831b57 None None] Cann
ot complete RPC initialization because driver isn't initialized properly.
```

3. 找不到驱动程序，驱动名称写错，请按正确信息填写驱动路径及名称。

图 4-3 找不到驱动程序时提示错误信息

```

2016-11-15 00:25:33.055 WARNING cinder.keymgr.com key_mgr [req-57aec0fc-76c6-4ee0-8c84-b311e0b87da5 None None] This key manager is insecure and is not recommended for production deployments
2016-11-15 00:25:33.157 ERROR cinder.cmd.volume [req-57aec0fc-76c6-4ee0-8c84-b311e0b87da5 None None] Volume service devstack-102@indigo-driver failed to start.
2016-11-15 00:25:33.157 TRACE cinder.cmd.volume Traceback (most recent call last):
2016-11-15 00:25:33.157 TRACE cinder.cmd.volume File "/opt/stack/cinder/cinder/cmd/volume.py", line 100, in
main
2016-11-15 00:25:33.157 TRACE cinder.cmd.volume cluster=cluster)
2016-11-15 00:25:33.157 TRACE cinder.cmd.volume File "/opt/stack/cinder/cinder/service.py", line 382, in create
create
2016-11-15 00:25:33.157 TRACE cinder.cmd.volume cluster=cluster)
2016-11-15 00:25:33.157 TRACE cinder.cmd.volume File "/opt/stack/cinder/cinder/service.py", line 202, in __
init__
2016-11-15 00:25:33.157 TRACE cinder.cmd.volume *args, **kwargs)
2016-11-15 00:25:33.157 TRACE cinder.cmd.volume File "/opt/stack/cinder/cinder/volume/manager.py", line 242
, in __init__
2016-11-15 00:25:33.157 TRACE cinder.cmd.volume active_backend_id=curr_active_backend_id)
2016-11-15 00:25:33.157 TRACE cinder.cmd.volume File "/usr/local/lib/python2.7/dist-packages/oslo_utils/importutils.py", line 44, in import_object
return import_class(import_str)(*args, **kwargs)
2016-11-15 00:25:33.157 TRACE cinder.cmd.volume File "/usr/local/lib/python2.7/dist-packages/oslo_utils/importutils.py", line 36, in import_class
traceback.format_exc_info(*sys.exc_info()))
2016-11-15 00:25:33.157 TRACE cinder.cmd.volume ImportError: Class InstorageMCSISCSIDriver cannot be found (['Traceback (most recent call last):\n', ' File "/usr/local/lib/python2.7/dist-packages/oslo_utils/importutils.py", line 32, in import_class\n return getattr(sys.modules[mod_str], class_str)\n', "AttributeError: 'module' object has no attribute 'InstorageMCSISCSIDriver'\n"])
2016-11-15 00:25:33.157 TRACE cinder.cmd.volume
2016-11-15 00:25:33.160 ERROR cinder.cmd.volume [req-57aec0fc-76c6-4ee0-8c84-b311e0b87da5 None None] No volume service(s) started successfully, terminating.
stack@devstack-102:/usr/bin$

```

4. 存储池不存在，请检查存储设备上存储池是否存在及存储池名称是否正确。

图 4-4 存储池不存在时提示错误信息

```

665-bb0c-534d0753e9f3 None None] CLI Exception output:
command: ['svcinfo', 'lsmdiskgrp', '-bytes', '-delim', '!', 'Pool"]
stdout:
stderr: CMWVC5804E The action failed because an object that was specified in the command does not exist.
2016-11-15 00:15:45.401 ERROR cinder.volume.manager [req-82c0aceb-9d62-4665-bb0c-534d0753e9f3 None None] Failed to initialize driver.

```

5. 主机上设备路径异常，虚拟机没有正确使用多路径设备。

目前 OpenStack 通常利用 Linux 作为平台，使用 libvirt 对虚拟机进行管理，底层使用 KVM 虚拟化技术，用户态使用 qemu 虚拟化环境。当启动了一个虚拟机以后，可以通过检查 libvirt 的配置文件，来确定虚拟机是否使用的是多路径设备。通过 libvirt 创建的虚拟机均会有一个配置文件，通常位于 /var/run/libvirt/qemu 目录下（具体路径跟操作系统有关）。配置文件中的 domstatus.domain.devices.disk[type='block'] 节点包含了块设备的相关配置，其中 <source dev='/dev/xxx'> 表示虚拟机使用的主机上的具体设备。如果该设备不是一个多路径设备，说明多路径的配置存在问题。如下标红的内容，该虚拟机使用了单路经设备。

```
<devices>
```



```

<emulator>/usr/bin/qemu-system-x86_64</emulator>
<disk type='block' device='disk'>
  <driver name='qemu' type='raw' cache='none' io='native'/>
  <source dev='/dev/sdd'/>
  <backingStore/>
  <target dev='vda' bus='virtio'/>
  ...
</disk>
</devices>

```

另外，通过 `ps | grep qemu`，查看系统中所有的 `qemu` 虚拟机进程，可以通过进程启动时的 `-drive` 参数的值，判断虚拟机启动时是否使用了多路径设备。

（但是这种方式只能看到进程启动时挂载的卷，无法查看到虚拟机启动之后再挂载的设备）。如下标红的内容，虚拟机使用了单路径设备。

```

libvirt+ 31103 0.1 1.3 658000 164988 ? S1 Aug07 2:12
/usr/bin/qemu-system-x86_64 -name guest=instance-00000003,debug-
threads=on ... -drive file=/dev/sdd,format=raw,if=none,id=drive-virtio-
disk0,serial=4a5d62de-4238-41e6-8fca-17f9c47caa89,cache=none,aio=native ...

```

当确认虚拟机只使用了单路径设备时，可以对“[OpenStack 启用多路径](#)”章节提供的相关配置进行检查，确保配置正确。

6. 主机上设备路径异常，多路径中的路径信息与预期不符。

当主机上的多路径路径信息与预期不符时，如路径状态，路径数量与设计的不一致时，可能原因及解决方法可参考以下内容：

- 1) 在使用 FC 链路情况下，如果 ZONE 配置在 OpenStack 使用过程中出现来回变动，有可能会导导致新的路径被发现，如果 ZONE 配置将原有的路径屏蔽，主机上的多路径信息会标识该路径为 Failed 状态，但是并不会自动删除该路径。该现象会在问题排查过程中产生干扰。
- 2) 在使用过程中，如果出现控制器损坏，则该控制器上的路径均会变为 Failed 状态，在控制器损坏后，进行卷的挂载动作，损坏控制器上的路径并不会被发现，也就不会聚合到多路径设备下。
- 3) 当系统中多路径相关配置出现问题时，可能会导致主机上多路径设备出现

遗留，或者设备出现遗留现象。简单的方法是通过重启，然后系统会重新加载路径设备，并进行多路径聚合，也可以通过手动执行 `multipath -f XXX`，`echo 1 > /sys/block/devX/device/delete` 等动作将多路径，路径设备进行删除。另外通过 `scsi_id` 可以获取 `scsi` 设备的 ID 号，多路径服务也是通过该 ID 号对设备进行聚合的，如果通过 `scsi_id` 获取到的多路径设备的路径的 ID 号并不一样，这种现象出现时，需要对主机进行重启，虚拟机进行重建。（目前无法解释出现这种现象的原因）。

`scsi_id` 使用方法如下：

```
/lib/udev/scsi_id --whitelist --page=0x83 /dev/sdb
```

- 4) 在一个有多套存储的云环境中，当需要根据一个设备，查找该设备所在的存储时，可以通过 `scsi_id` 命令获取到设备的 ID 号，然后在存储上根据该 ID 号操作是否有对应卷。（注意，`scsi_id` 命令读出的 ID 号与存储上的 ID 号相比，在最前面多了一个“3”）
7. 其他故障请根据错误码查找原因。

5 术语&缩略语

A		
API	Application Program Interface	应用程序接口
C		
-	Cinder	OpenStack 的一个重要组件
F		
FC	Fiber Channel	光纤通道
I		
iSCSI	Internet Small Computer System Interface	互联网小型计算机接口
InStorage	Inspur Storage	浪潮存储
L		
LVM	Logical Volume Manager	逻辑卷管理
O		
-	OpenStack	云计算管理平台
S		
SSH	Secure Shell Protocol	一种安全协议
W		
WWPN	World Wide Port Name	全球端口名称

附录一

浪潮所开发的产品属于 Cinder 的驱动，具体配置请查看 Cinder 安装部署方法。关于 Cinder 的使用方法请访问 OpenStack 社区。

OpenStack 用户指南：<https://docs.openstack.org/user-guide/>

命令行使用方法：<https://docs.openstack.org/cli-reference/cinder.html>