天翼云
State Cloud

存储资源盘活系统

OpenStack Cinder 驱动使用手册

天翼云科技有限公司

修订记录

版本	发布日期	描述	适配 HBlock 版本
1.1	2024 年 09 月 11 日	第一次发版： 1. 支持本地卷和上云卷。 2. 支持设置卷相关参数。 3. 支持 Nova 和 Glance 使用 HBlock 作为后端存储。	3.5 及以上

目 录

1 产品定义

HBlock是中国电信天翼云自主研发的存储资源盘活系统（Storage Resource Reutilization System，简称SRRS），是一款轻量级存储集群控制器，实现了全用户态的软件定义存储，将通用服务器及其管理的闲置存储资源转换成高可用的虚拟磁盘，通过标准 iSCSI 协议提供分布式块存储服务，挂载给本地服务器（或其他远程服务器）使用，实现对资源的集约利用。同时，产品拥有良好的异构设备兼容性及场景化适配能力，无惧IT架构升级带来的挑战，助力企业用户降本增效和绿色转型。

OpenStack是一个开源的云计算管理平台项目，目标是提供实施简单、可大规模扩展、丰富、标准统一的云计算管理平台。Cinder为OpenStack的管理块设备，主要功能是为虚拟机实例提供虚拟磁盘管理服务。Cinder本身不是块设备源，当虚拟机需要块设备时，询问Cinder去哪里获取具体的块设备。它的插件驱动架构有利于块设备的创建和管理，如创建卷、删除卷、在实例上挂载和卸载卷。OpenStack Cinder Driver是部署在OpenStack Cinder模块上的一个插件程序，遵循了OpenStack的driver架构。通过安装和HBlock适配的Cinder Driver驱动（以下简称stor Driver），并进行配置资源对接，实现和OpenStack Cinder 和HBlock存储系统资源连通，通过iSCSI协议向OpenStack中的虚拟机提供HBlock的逻辑卷功能。本文档仅为插件的安装和使用提供建议。

stor Driver与OpenStack的块存储（Cinder）和计算（Nova）组件集成，为OpenStack提供高性能，可扩展，高可靠的持久化存储。

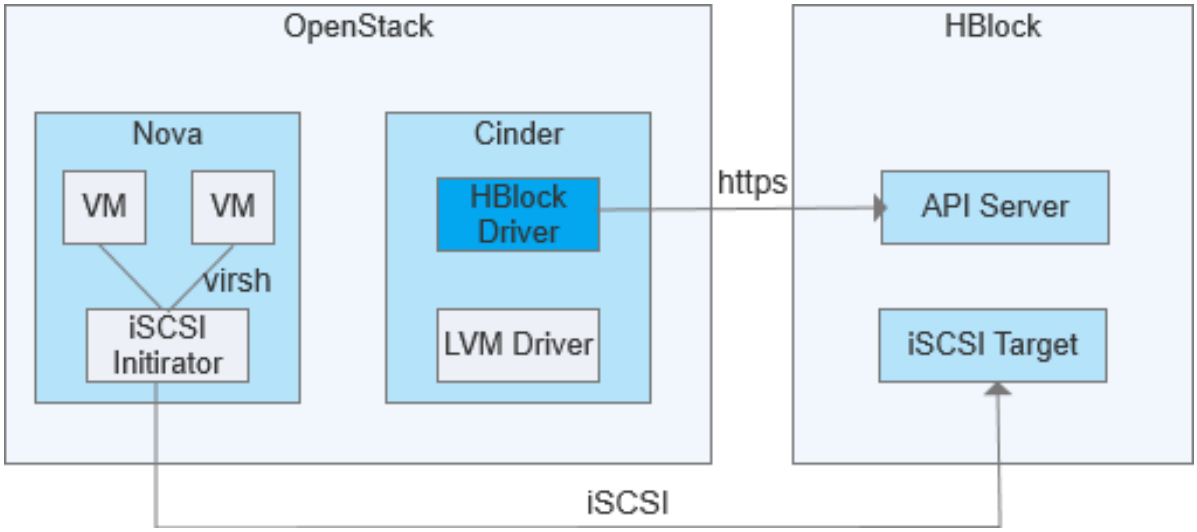


图1.HBlock对接OpenStack架构图

上图为HBlock对接OpenStack的架构图，其中：

- stor Driver：OpenStack Cinder接入HBlock的驱动程序，使用https协议和HBlock的API server进行交互，完成卷的创建、删除、查询、扩容等操作。
- nova-compute节点：完成iSCSI协议的接入，建立iSCSI卷后，通过virsh命令挂载到VM实例。

2 安装部署

2.1 前置条件

安装部署 HBlock 3.5.0 或以上版本，详细安装过程参见 HBlock 用户手册。

2.2 环境要求

环境项目	版本
OpenStack	Queens~Caracal
Python	2.7 及以上版本

2.3 安装驱动

在 Cinder 集群的节点上安装。

注意：非 root 用户安装，需要有其他用户的读权限。

1. 解压安装包

```
tar -zxvf stor-cinder-driver-1.1.0.tar.gz
```

2. 安装插件包

```
cd stor-cinder-driver-1.1.0
python setup.py install
```

说明：OpenStack 通常的安装路径位于下列目录：/usr，所以 Stor Cinder 驱动安装到此目录下。如果 OpenStack 安装到了别的路径，需要指定安装路径，与 OpenStack 所在的安装目录保持一致，例如：如果 openstack 安装在/user/lib/python_*version*/site-packages，那么 --prefix=/usr（path 为 OpenStack 所在的安装目录）。

3. 执行结束后，会在 stor-cinder-driver-1.1.0/dist 目录下生成 egg 文件(例如 stor_cinder_driver-1.1.0-py3.0.egg)，留存即可，通常无需手动分发此安装文件。

注意：安装包应在 Cinder 集群节点进行安装。如果 Cinder 使用了虚拟 python 环境，安装包需要在 Cinder 虚拟 python 环境中安装，使得 Cinder 可以发现插件的 Python 包。

3 配置插件

3.1 配置卷类型

可以按照下列步骤进行卷类型配置。

1. 修改 Cinder 的配置文件/etc/cinder/cinder.conf，使用 enabled_backends 启用 HBlcok 的卷，并且在配置文件中增加 HBlock 卷相关参数。

   说明：一次可以在 cinder.conf 添加多个卷的类型。

   单机版示例如下：

```
[DEFAULT]
enabled_backends = lvmdriver-1,stor-1

[stor-1]
volume_group=stack_volumes-lvmdriver-2
volume_driver=stor_driver.driver.driver_stor.StorDriver
volume_backend_name = stor-1
stor_provider = HBlock
stor_api_user = storuser
stor_api_password = *************
stor_chap_user = chap_user
stor_chap_password = ************
stor_api_endpoint = https://10.183.22.196:31443
write_policy = WriteBack
sector_size = 4096
max_sessions = 2
stor_path = /mnt/stor
```

   集群版示例如下：

```
[DEFAULT]
enabled_backends = lvmdriver-1,stor,stor1,stor2

[stor]
```

```
volume_group=stack_volumes-lvmdriver-2
volume_driver=stor_driver.driver.driver_stor.StorDriver
volume_backend_name = stor
stor_provider = HBlock
stor_api_user = storuser
stor_api_password = *************
stor_chap_user = chap_user
stor_chap_password = ************
stor_api_endpoint = https://10.183.22.197:31443
stor_chap_user = chap_user
stor_chap_password = ************
storage_mode = Local
local_storage_class = EC 2+1
high_availability = ActiveStandby
write_policy = WriteBack
sector_size = 512
ec_fragment_size = 16
max_sessions = 2
min_replica = 2

[stor1]
volume_group=stack_volumes-lvmdriver-2
volume_driver=stor_driver.driver.driver_stor.StorDriver
volume_backend_name = stor1
stor_provider = HBlock
stor_api_user = storuser
stor_api_password = *************
stor_chap_user = chap_user
stor_chap_password = ************
stor_api_endpoint = https://10.183.22.199:31443
storage_mode = Local
local_storage_class = single-copy
high_availability = Disabled
write_policy = WriteAround
sector_size = 512
```

```
[stor2]
volume_group=stack_volumes-lvmdriver-2
volume_driver=stor_driver.driver.driver_stor.StorDriver
volume_backend_name = stor2
stor_provider = HBlock
stor_api_user = admin
stor_api_password = *************
stor_chap_user = chap_user
stor_chap_password = ************
stor_api_endpoint = https://10.183.22.200:31443
storage_mode = Local
local_storage_class = 3-copy
high_availability = ActiveStandby
write_policy = WriteThrough
sector_size = 4096
```

文件中的参数解释如下：

参数名称	参数说明	是否必填
enabled_backends	需要在系统生效的后端存储名称。 如果有多个需要生效的存储，名称之间用英文逗号隔开。	是
volume_group	卷的组。	是
volume_driver	HBlock 驱动所在路径。 取值为：stor_driver.driver.driver_stor.StorDriver。	是
volume_backend_name	卷对应的后端存储名称，存储节点的 Volume Provider 命名。与 enabled_backends 定义的后端存储名字保持一致。	是
stor_provider	产品名称。 取值：HBlock。	是

stor_api_user	HBlock 的管理员用户名。	是
stor_api_password	HBlock 的管理员密码。	是
storage_mode	HBlock 卷的模式。 取值： ● `Local`：本地模式，数据全部保留在本地。 ● `Cache`：缓存模式，本地保留部分热数据，全部数据异步存储到 OOS 中。 ● `Storage`：存储模式，本地保留全部数据，并异步存储到 OOS 中。 默认值为 Local。	否
stor_chap_user	CHAP 认证用名称。 取值：字符串形式，长度范围是 3~64，只能由字母、数字、句点(.)、短横线(-)、下划线(_)、冒号(:)组成，字母区分大小写，且仅支持以字母或数字开头。	否
stor_chap_password	CHAP 认证密码。 取值：字符串形式，长度范围是 12~16，只能由字母、数字或下划线(_)组成，字母区分大小写。	否
stor_api_endpoint	配置 HBlock RESTful API 地址和端口。	是
stor_path	指定 HBlock 存储卷数据的数据目录（仅单机版支持）。 **说明**：如果创建卷时不指定数据目录，使用服务器设置的默认数据目录。	否
server_numbers	Target 所在的服务器数量（仅集群版支持）。 整数形式，取值为[2, n]，n 为集群内服务器的数量。默认值为 2。	否
pool	指定存储池（仅集群版支持），表示最终存储池，卷数据最终落在该存储池内。默认使用集群的基础存储池。	否

cache_pool	指定缓存存储池（仅集群版支持）。如果指定了缓存存储池，卷数据首先写入缓存存储池，然后再存入存储池。 **注意**：存储池与缓存存储池不能是同一个存储池。	否
high_availability	选择卷的高可用类型（仅集群版支持）： ● ActiveStandby：启用主备，该卷关联对应 Target 下的所有 IQN。 ● Disabled：不启用主备，该卷关联对应 Target 下的 1 个 IQN。 默认值为 ActiveStandby。	否
local_storage_class	卷冗余模式（仅集群版支持）。 取值： ● single-copy：单副本。 ● 2-copy：两副本。 ● 3-copy：三副本。 ● EC *N*+*M*：纠删码模式。其中 N、M 为正整数，N>M，且 N+M≤128。表示将数据分割成 N 个片段，并生成 M 个校验数据。 默认值为 EC 2+1。	否
min_replica	最小副本数（仅集群版支持）。 对于副本模式的卷，假设卷副本数为 X，最小副本数为 Y（Y 必须≤X），该卷每次写入时，至少 Y 份数据写入成功，才视为本次写入成功。对于 EC *N*+*M* 模式的卷，假设该卷最小副本数设置为 Y（必须满足 N≤Y≤N+M），必须满足总和至少为 Y 的数据块和校验块写入成功，才视为本次写入成功。 取值：整数。对于副本卷，取值范围是[1, N]，N 为副本	否

	模式卷的副本数，默认值为 1。对于 EC 卷，取值范围是[N, N+M]，默认值为 N。	
ec_fragment_size	纠删码模式分片大小（仅集群版支持）。卷冗余模式为 EC 模式时，此设置才生效，否则忽略。 取值：1、2、4、8、16、32、64、128、256、512、1024、2048、4096，单位是 KiB。默认值为 16。	否
sector_size	设置扇区大小。 取值：512、4096，单位为字节。默认值为 4096。	否
write_policy	卷的写策略： ● WriteBack（wb）：回写，指数据写入到内存后即返回客户端成功，之后再异步写入磁盘。适用于对性能要求较高，稳定性要求不高的场景。 ● WriteThrough（wt）：透写，指数据同时写入内存和磁盘，并在都写成功后再返回客户端成功。适用于稳定性要求较高，写性能要求不高，且最近写入的数据会较快被读取的场景。 ● WriteAround（wa）：绕写，指数据直接写到磁盘，不写入内存。适用于稳定性要求较高，性能要求不高，且写多读少的场景。 默认值为 WriteBack（wb）。	否
max_sessions	iSCSI Target 允许建立的最大会话数。 取值：整数，取值范围是[1, 1024]，默认值为 1。	否
cloud_bucket_name	已存在的 OOS 存储桶的名称。 **注意**：请勿开启 Bucket 的生命周期设定和合规保留。	上云卷必填
cloud_prefix	设置 OOS 中的前缀名称，设置前缀名称后，卷数据会存在存储桶以前缀命名的类文件夹中。如果未指定前缀，则直接存储在以卷名称命名的类文件夹中。	否

	取值：字符串形式，长度范围是 1~256。	
cloud_access_key	OOS AccessKeyID。	上云卷必填
cloud_secret_key	OOS SecretAccessKey。	上云卷必填
cloud_endpoint	设置对 OOS Endpoint。	上云卷必填
cloud_object_size	数据存储在 OOS 中的大小。 取值：128、256、512、1024、2048、4096、8192，单位是 KiB。默认值为 1024。	否
cloud_storage_class	设置 OOS 的存储类型： ● STANDARD：标准存储。 ● STANDARD_IA：低频访问存储。 默认为 STANDARD。	否
cloud_compression	是否压缩数据上传至 OOS： ● Enabled（on）：压缩数据上传至 OOS。 ● Disabled（off）：不压缩数据上传至 OOS。 默认值为 Enabled（on）。	否
cloud_sign_version	指定上云签名认证的类型： ● v2：V2 签名认证。 ● v4：V4 签名认证。 默认值为 v2。	否
cloud_region	表示 Endpoint 资源池所在区域。 V4 签名时，此项必填。	条件
delete_cloud_data	删除卷时，是否删除云上的数据： ● true：删除云上数据。 ● false：不删除云上数据。	否

	默认值为 true。	

2. 使用下列命令，使用配置生效：

```
cinder type-create volume_backend_name
cinder type-key volume_backend_name set volume_backend_name=volume_backend_name
```

示例 1：使步骤 1 中单机版配置生效

```
cinder type-create stor-1
cinder type-key stor-1 set volume_backend_name=stor-1
```

示例 2：使步骤 1 中配置的集群版配置生效

```
cinder type-create stor
cinder type-key stor set volume_backend_name=stor
cinder type-create stor1
cinder type-key stor1 set volume_backend_name=stor1
cinder type-create stor2
cinder type-key stor2 set volume_backend_name=stor2
```

3. 重启 Cinder 服务

- 如果使用 DevStack 方式安装 OpenStack，重启 Cinder 服务命令如下：

```
systemctl restart devstack@c-*
```

- 如果使用 Packstack 安装 OpenStack，重启 Cinder 服务命令如下：

```
systemctl restart openstack-cinder*
```

4. 配置 Multipath（集群版）

1) 参考《存储资源盘活系统用户手册-命令行》“客户端配置”-“Linux 客户端 – 集群版”-“客户端配置”章节中的“配置 MPIO”，确保 MPIO 正确配置。

2) 启用 Nova 的 multipath 功能（如果是多节点部署，需要到对应的计算节点修改配置）：如果使用 DevStack 方式安装 OpenStack，修改/etc/nova-cpu.conf；如果使用 Packstack 安装 OpenStack，修改/etc/nova/nova.conf。

```
[libvirt]
```

```
iscsi_use_multipath = true
```

5. 重启 Nova 服务（集群版）

说明：如果是多节点部署，需要到对应的计算节点重启 Nova 服务。

- 如果使用 DevStack 方式安装 OpenStack，重启 Nova 服务命令如下：

```
systemctl restart devstack@n-*
```

- 如果使用 Packstack 安装 OpenStack，重启 Nova 服务命令如下：

```
systemctl restart openstack-nova*
```

6. 验证卷是否配置正确

```
cinder type-list
cinder type-show ID  #ID 为 cinder type-list 中查询到的 ID
```

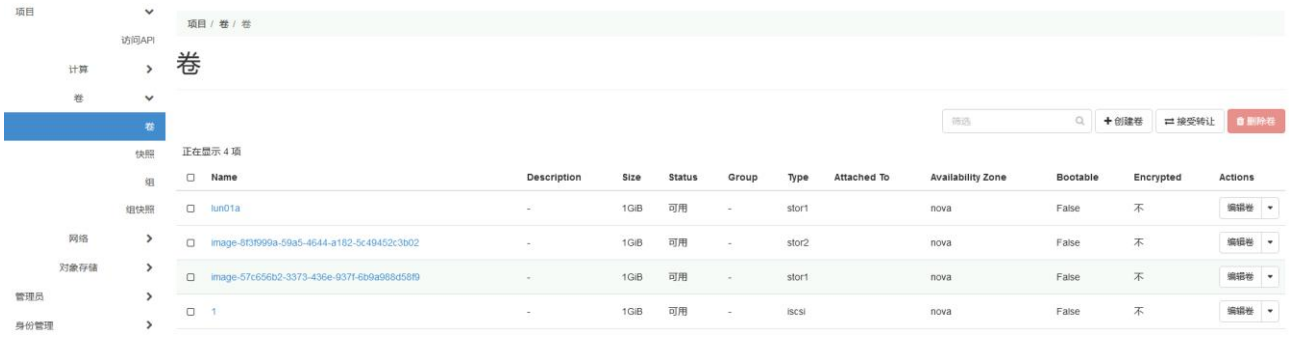
例如:

```
[root@server ~(keystone_admin)]# cinder type-list
+--------------------------------------+-------------+---------------------+-----------+
| ID                                   | Name        | Description         | Is_Public |
+--------------------------------------+-------------+---------------------+-----------+
| 40dda59f-f524-4f5d-99a8-da2fc4650463 | stor        | -                   | True      |
| 693b27f2-c59d-45a8-9870-9cc7283de3bd | stor1       | -                   | True      |
| 97a3c362-738f-41cf-8723-f84454707361 | stor2       | -                   | True      |
| f292d76d-0de6-447f-b294-418399ba4b5e | __DEFAULT__ | Default Volume Type | True      |
+--------------------------------------+-------------+---------------------+-----------+
[root@server ~(keystone_admin)]# # cinder type-show 40dda59f-f524-4f5d-99a8-da2fc4650463
+---------------------------------+--------------------------------------+
| Property                        | Value                                |
+---------------------------------+--------------------------------------+
| description                     | None                                 |
| extra_specs                     | volume_backend_name : stor           |
| id                              | 40dda59f-f524-4f5d-99a8-da2fc4650463 |
| is_public                       | True                                 |
| name                            | stor                                 |
| os-volume-type-access:is_public | True                                 |
| qos_specs_id                    | None                                 |
+---------------------------------+--------------------------------------+
```

4 WEB 使用

4.1 查看卷

登录 OpenStack，点击“卷”，进入“卷”页面，可以查看已创建的卷。



名称	描述
名称	卷名称。
描述	对卷的说明。
大小	卷的容量。
状态	卷的状态： ● 可用 ● 正在使用 ● 错误 ● 删除中
类型	卷类型，卷的类型，配置卷类型中配置的卷。
连接到	连接到实例的具体位置。
可用域	卷的可用域。
可启动	卷是否可启动： ● Yes

	● No
动作	可以对卷进行编辑卷： ● 扩展卷 ● 管理连接 ● 创建快照 ● 修改卷类型 ● 上传镜像 ● 创建转让 ● 删除卷 ● 更新元数据

4.2 创建卷

登录 OpenStack，点击“卷”，进入“卷”页面，点击“创建卷”按钮，可以进行卷创建。

说明：主要介绍 HBlock 卷相关的内容。

名称	描述
卷名称	HBlock 卷名称。 取值：字符串，长度范围是 1~16，只能由字母、数字和短横线（-）组成，字母区分大小写，且仅支持以字母或数字开头。
描述	对卷的说明。
卷来源	创建卷的来源： ● 没有源，空白卷 ● 镜像：如果选择该项，则需要选择一个镜像作为源。 卷：复制其他的卷信息，创建新的卷。目前该驱动不支持。
使用镜像作为源	选择一个镜像，作为卷的源。 **说明**：如果“卷来源”项选择的是“镜像”，才会有该项。
类型	卷类型，根据**配置卷类型**中配置的卷，选择新建卷的类型。
大小（GiB）	卷容量。 取值：[1,1048576]，单位是 GiB。

可用域	可用域。 **说明：**如果“卷来源”项选择的是“卷”，则不会显示该项。

4.3 连接卷

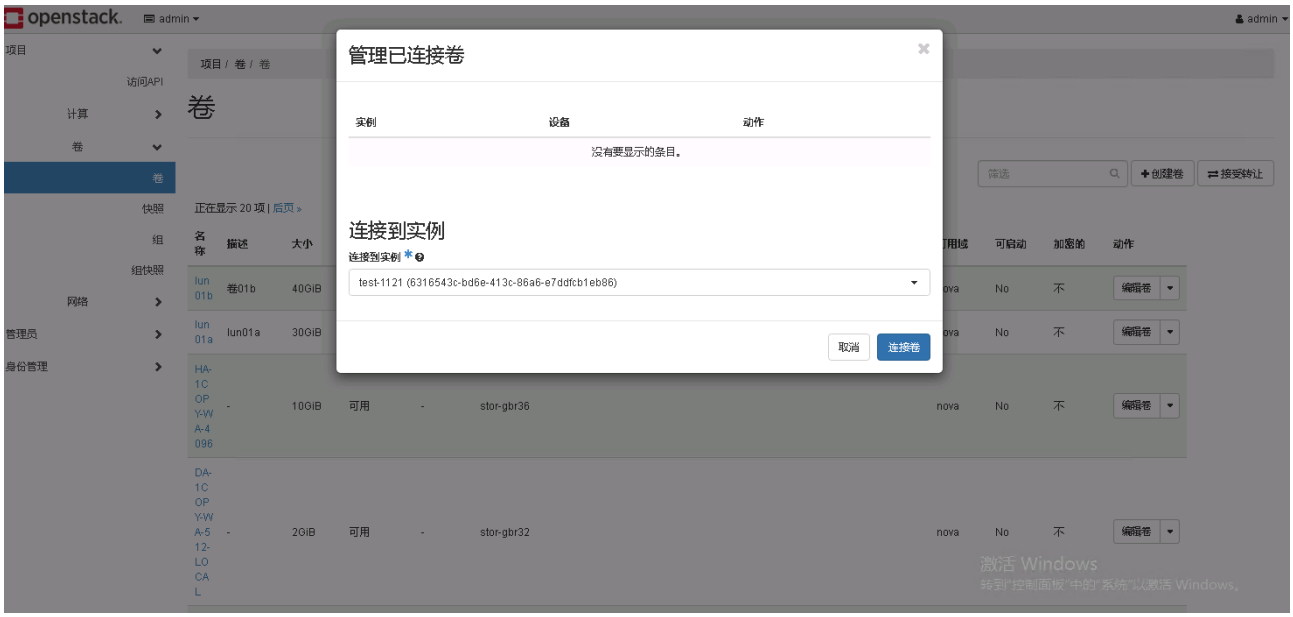
连接卷成功后，可以登录实例，进行 mount 等操作。

可以通过下列两种方式中的一种连接卷：

- 在卷页面连接卷
- 在实例页面连接卷

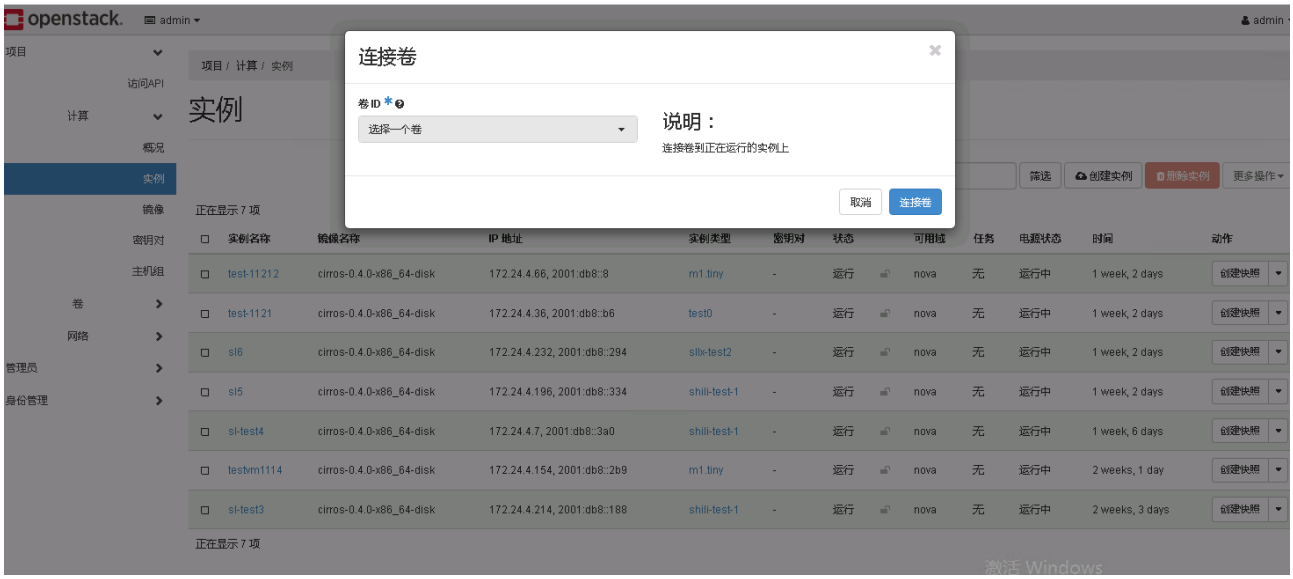
4.3.1 连接卷（卷页面）

登录 OpenStack，点击“卷”，进入“卷”页面，点击需要连接卷“动作”列的“编辑卷”>“管理连接”，弹出“管理已连接卷”页面，可在“连接到实例”下拉框中选择需要连接的实例。



4.3.2 连接卷（实例页面）

登录 OpenStack，点击“计算”>“实例”，进入“实例”页面，在具体实例后的“动作”列选项中选择“连接卷”，弹出“连接卷”页面，在“卷 ID”下拉框中选择需要连接的卷即可。



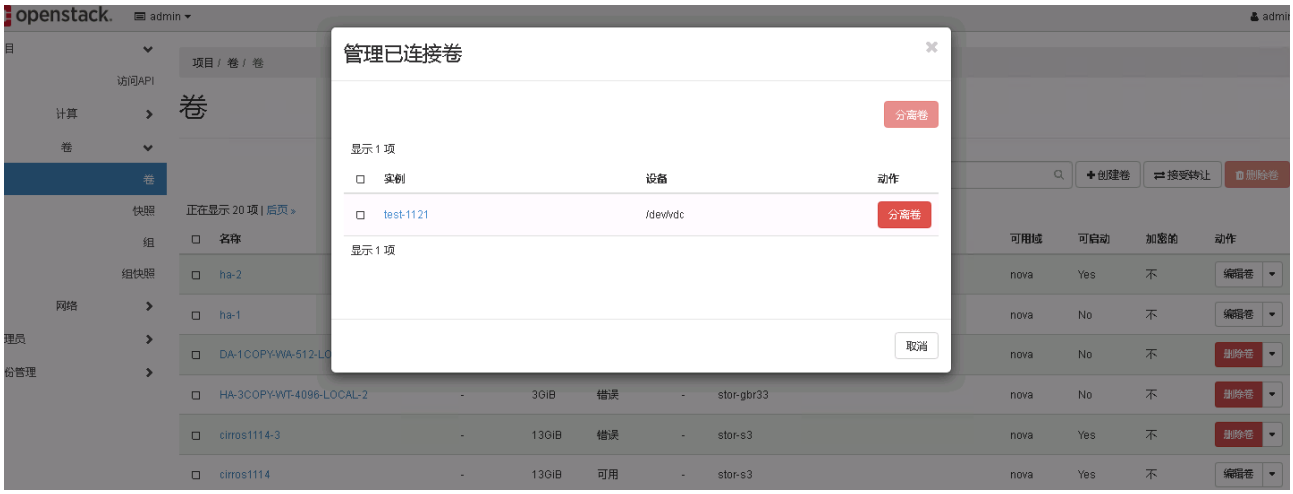
4.4 分离卷

可以通过下列两种方式中的一种分离已连接的卷：

- 在卷页面分离卷
- 在实例页面分离卷

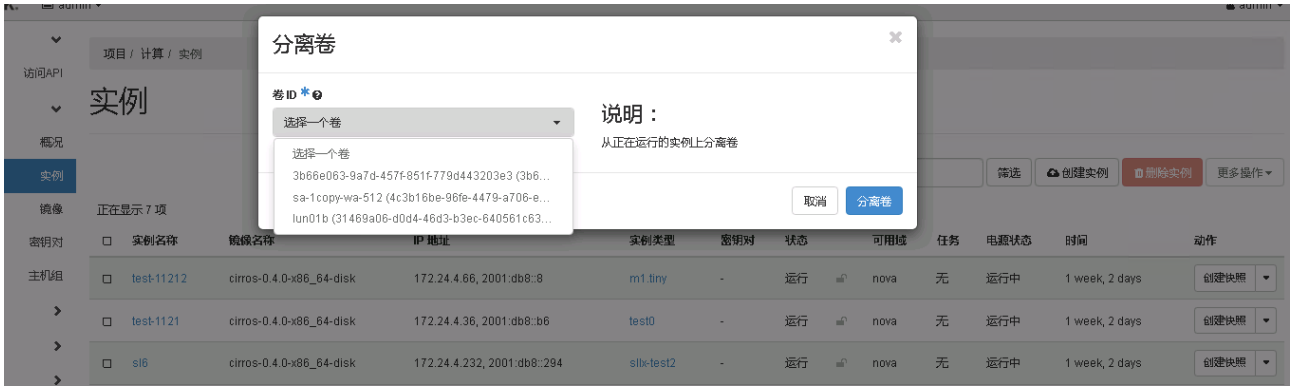
4.4.1 分离卷（卷页面）

登录 OpenStack，点击“卷”，进入“卷”页面，点击需要分离卷“动作”列的“编辑卷”>“管理连接”，弹出“管理已连接卷”页面，可在点击动作列的“分离卷”按钮，即可分离卷。



4.4.2 分离卷（实例页面）

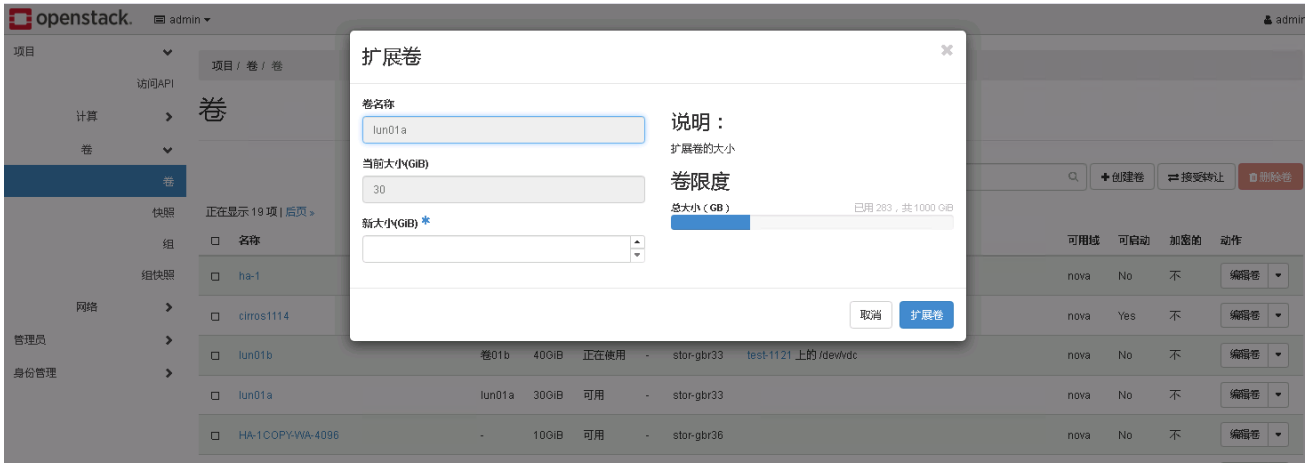
登录 OpenStack，点击“计算”>“实例”，进入“实例”页面，在具体实例后的“动作”列选项中选择“分离卷”，弹出“分离卷”页面，在“卷 ID”下拉框中选择需要分离的卷即可。


分离卷
卷 ID *
选择一个卷
选择一个卷
3b66e063-9a7d-457f-851f-779d443203e3 (3b6...
sa-1copy-wa-512 (4c3b16be-96fe-4479-a706-e...
lun01b (31469a06-d0d4-46d3-b3ec-640561c63...
说明：
从正在运行的实例上分离卷
取消
分离卷
项目 / 计算 / 实例
实例
访问API
概况
实例
镜像
密钥对
主机组
正在显示 7 项
筛选
创建实例
删除实例
更多操作
实例名称
镜像名称
IP 地址
实例类型
密钥对
状态
可用域
任务
电源状态
时间
动作
test-11212
cirros-0.4.0-x86_64-disk
172.24.4.66, 2001:db8::8
m1.tiny
运行
nova
无
运行中
1 week, 2 days
创建快照
test-1121
cirros-0.4.0-x86_64-disk
172.24.4.36, 2001:db8::b6
test0
运行
nova
无
运行中
1 week, 2 days
创建快照
sl6
cirros-0.4.0-x86_64-disk
172.24.4.232, 2001:db8::294
sllx-test2
运行
nova
无
运行中
1 week, 2 days
创建快照

4.5 扩展卷

登录 OpenStack，点击“卷”，进入“卷”页面，点击需要扩展卷“动作”列的“编辑卷”>“扩展卷”，弹出“扩展卷”页面，在新大小（GiB）中填写扩展后的容量。

注意：卷在没有连接实例的情况下才能进行卷扩展。如果已经连接，请先分离卷。

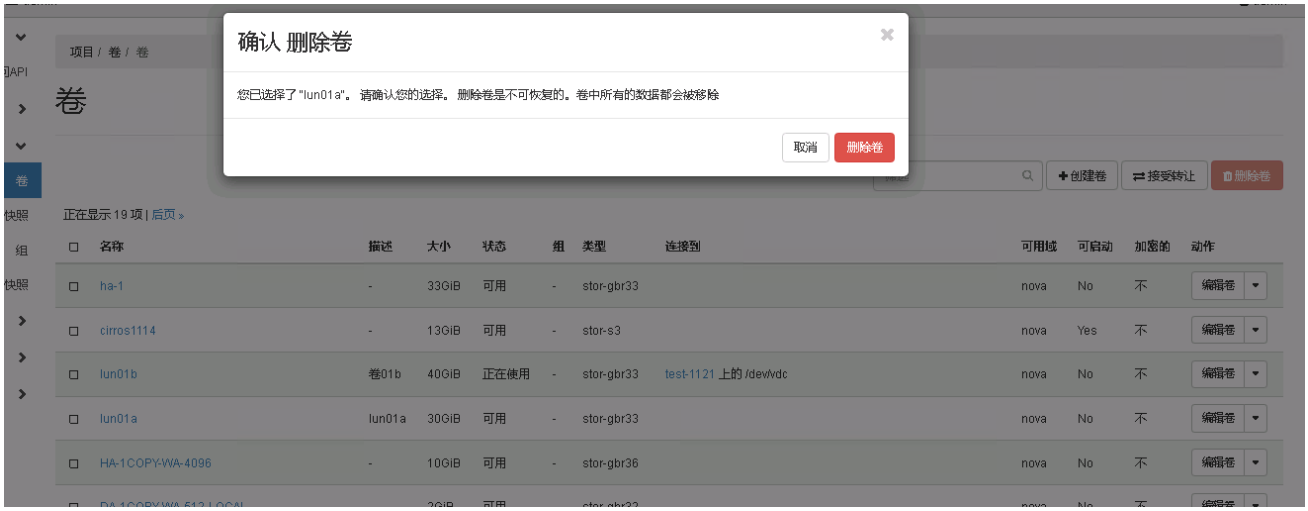


4.6 删除卷

登录 OpenStack，点击“卷”，进入“卷”页面，点击需要扩展卷“动作”列的“编辑卷”>“删除卷”，弹出“确认删除卷”页面，点击“删除卷”按钮，即删除卷。

注意：

- 删除卷是不可恢复的，卷中所有的数据都会被移除。
- 卷在没有连接实例的情况下才能进行删除。如果已经连接，请先分离卷。



5 Nova 使用 HBlock 卷

5.1 前置条件

- HBlock 服务已经正确配置，且可以正常启动。
- 正确安装 OpenStack,，支持的 OpenStack 版本详见**正确安装 OpenStack，支持的 OpenStack 版本详见。**
- 。
- 正确安装 OpenStack 的接入插件，且功能正常。

5.2 使用方法

以镜像为源创建的 HBlock 卷 test 为例。

- **卷查询**

```
[root@server devstack]# cinder list
```

```
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
| ID                                   | Status    | Name | Size | Volume Type | Bootable | Attached to |
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
| 0be87b9f-f1aa-429f-af81-75795dae2de8 | available | 1    | 1    | iscsi       | false    |             |
| c4e8af38-2b4f-48a2-9f54-6cf4c8f26806 | available | test | 1    | stor1       | true     |             |
| efb3b49a-c06f-44d5-937e-9fa1370a2785 | available | 6    | 1    | stor1       | false    |             |
+--------------------------------------+-----------+------+------+-------------+----------+-------------+
```

- **flavor 查询**

```
[root@server devstack]# nova flavor-list
+----+-----------+------------+------+-----------+------+-------+-------------+-----------+-------------+
| ID | Name      | Memory_MiB | Disk | Ephemeral | Swap | VCPUs | RXTX_Factor | Is_Public | Description |
+----+-----------+------------+------+-----------+------+-------+-------------+-----------+-------------+
| 1  | m1.tiny   | 512        | 1    | 0         | 0    | 1     | 1.0         | True      | -           |
| 2  | m1.small  | 2048       | 20   | 0         | 0    | 1     | 1.0         | True      | -           |
| 3  | m1.medium | 4096       | 40   | 0         | 0    | 2     | 1.0         | True      | -           |
| 4  | m1.large  | 8192       | 80   | 0         | 0    | 4     | 1.0         | True      | -           |
| 5  | m1.xlarge | 16384      | 160  | 0         | 0    | 8     | 1.0         | True      | -           |
+----+-----------+------------+------+-----------+------+-------+-------------+-----------+-------------+
```

- **image 查询**

```
[root@server devstack]# glance image-list
+--------------------------------------+------------+
| ID                                   | Name       |
+--------------------------------------+------------+
| 7a1586ec-f9a3-4be5-921c-565136d9573b | cmdUbuntu1 |
+--------------------------------------+------------+
```

- **net-id 查询**

```
[root@server devstack]# openstack network list
+--------------------------------------+------+--------------------------------------+
| ID                                   | Name | Subnets                              |
+--------------------------------------+------+--------------------------------------+
| 53dced88-bc11-462d-878c-50373d45e7d4 | net1 | 216fe08b-973e-4779-971c-e81a7dbba0f5 |
+--------------------------------------+------+--------------------------------------+
```

- 使用命令行创建虚拟机实例

```
[root@server devstack]# openstack server create --flavor m1.small --nic net-id=net1 --volume test
test-instance8
+-------------------------------------+--------------------------------------+
| Field                               | Value                                |
+-------------------------------------+--------------------------------------+
| OS-DCF:diskConfig                   | MANUAL                               |
| OS-EXT-AZ:availability_zone         |                                      |
| OS-EXT-SRV-ATTR:host                | None                                 |
| OS-EXT-SRV-ATTR:hypervisor_hostname | None                                 |
| OS-EXT-SRV-ATTR:instance_name       |                                      |
| OS-EXT-STS:power_state              | NOSTATE                              |
| OS-EXT-STS:task_state               | scheduling                           |
| OS-EXT-STS:vm_state                 | building                             |
| OS-SRV-USG:launched_at              | None                                 |
| OS-SRV-USG:terminated_at            | None                                 |
| accessIPv4                          |                                      |
| accessIPv6                          |                                      |
| addresses                           |                                      |
| adminPass                           | jAW7hUEvhbnd                         |
| config_drive                        |                                      |
| created                             | 2024-08-27T07:34:49Z                 |
| flavor                              | m1.small (2)                         |
| hostId                              |                                      |
| id                                  | 2c37fb17-abe3-4f7e-857a-0c162440f57e |
| image                               |                                      |
| key_name                            | None                                 |
| name                                | test-instance8                       |
| progress                            | 0                                    |
| project_id                          | 7984fa99a42b4da28c2e8efb1c0fda98     |
| properties                          |                                      |
| security_groups                     | name='default'                       |
| status                              | BUILD                                |
| updated                             | 2024-08-27T07:34:49Z                 |
| user_id                             | 9fc8c542deae487d898575abd7a43be5     |
| volumes_attached                    |                                      |
+-------------------------------------+--------------------------------------+
```

- 检查创建好的虚拟机

```
[root@server devstack]# nova list
+--------------------------------------+----------------+--------+----------------------+-------------+------------------+
| ID                                   | Name           | Status | Task State           | Power State | Networks         |
+--------------------------------------+----------------+--------+----------------------+-------------+------------------+
| 10c71001-9db2-4232-8a50-031d04f386c3 | test-instance  | ACTIVE | -                    | Running     | net1=172.24.4.88 |
| 2c37fb17-abe3-4f7e-857a-0c162440f57e | test-instance8 | BUILD  | block_device_mapping | NOSTATE     |                  |
+--------------------------------------+----------------+--------+----------------------+-------------+------------------+
```

6 Glance 使用 HBlock 卷

6.1 前置条件

- HBlock 服务已经正确配置，且可以正常启动。
- 正确安装 OpenStack，支持的 OpenStack 版本详见 **2.2 环境要求**。
- 正确安装 OpenStack 的接入插件，且功能正常。

6.2 使用方法

1. 修改 Glance 的配置文件/etc/glance/glance-api.conf，修改 enabled_backends，支持使用 Cinder 作为 Glance 的后端存储，同时通过 cinder_volume_type 来指定卷类型。

```
[DEFAULT]

enabled_backends="file:file,cinder-a:cinder,cinder-b:cinder"

[glance_store]

stores=file,cinder-a,cinder-b

default_backend=file

[file]

filesystem_store_datadir = /opt/stack/data/glance/images/

[os_glance_tasks_store]

filesystem_store_datadir = /opt/stack/data/glance/tasks_work_dir

[os_glance_staging_store]

filesystem_store_datadir = /opt/stack/data/glance/staging

[cinder-a]

cinder_os_region_name=RegionOne

cinder_store_auth_address=http://10.0.132.12/identity/v3

cinder_store_user_name=cinder
```

```
cinder_store_password=nomoresecret

cinder_store_project_name=service

cinder_volume_type=hblocka

cinder_enforce_multipath=True

cinder_use_multipath=True

rootwrap_config = /etc/cinder/rootwrap.conf

[cinder-b]

cinder_os_region_name=RegionOne

cinder_store_auth_address=http://10.0.132.12/identity/v3

cinder_store_user_name=cinder

cinder_store_password=nomoresecret

cinder_store_project_name=service

cinder_volume_type=hblockb

cinder_enforce_multipath=True

cinder_use_multipath=True

rootwrap_config = /etc/cinder/rootwrap.conf
```

部分参数描述，具体详细参数请参考 OpenStack 官网。

参数	描述
enabled_backends	存储标识符和存储类型。配置形式为 *key*:*value*。*value* 的合法值为 file、rbd、http、swift、cinder。*key* 为 default_backend 的值。 **说明**：可以一次配置多个 *key*:*value*，以英文逗号隔开，如 *key1*:*value1*,*key1*:*value2*。enabled_backends 中的每一个 *key* 都需要有一个单独的一个配置组[*key*]，配置组中需要配置该配置的所有详细配置项。
default_backend	必须为 enabled_backends 中的 key。
cinder_os_region_name	从服务目录中查找 cinder 服务的区域名称。

cinder_store_auth_address	Openstack 实际的 identity 服务 url，对应的账户密码需找管理员获取。在 PackStack 部署模式下，此值为类似的地址：http://10.0.132.12:5000/v3。
cinder_store_user_name	OpenStack 鉴权账户名。
cinder_store_password	OpenStack 鉴权密码。
cinder_store_project_name	镜像卷存储在 Cinder 中的项目名称。
cinder_volume_type	在 Cinder 中创建卷的卷类型。
cinder_enforce_multipath	如果它被设置为 True，则当 Multipathd 未运行时，镜像传输的卷附加将中止。 取值： ● True。 ● False。 **说明**：如果 HBlock 集群版，需要配置为 True，如果 HBlock 单机版，配置为 False。仅在 cinder_use_multipath 设置为 True 时，才将此字段设置为 True。
cinder_use_multipath	在部署中支持用于识别 Mutipath 的标记。 如果不支持多路径，则将其设置为 False。 ● True：Cinder 使用多路径。 ● False：Cinder 不使用多路径。 **说明**：如果 HBlock 集群版，需要配置为 True，如果 HBlock 单机版，配置为 False。
rootwrap_config	用于以 root 用户身份运行命令的 rootwrap 配置文件的路径。 Cinder 存储需要 root 特权才能运行镜像卷（用于连接到 iSCSI/FC 卷以及读/写卷数据等）。配置文件应允许 cinder store 和 os-brick 库所需的命令。

注意：以下仅为示例，请按照实际的 OpenStack 环境信息进行填写。

```
[DEFAULT]
enabled_backends="file:file,cinder:cinder"

[glance_store]
stores=file,cinder
default_backend=file

[file]
filesystem_store_datadir = /opt/stack/data/glance/images/
[os_glance_tasks_store]
filesystem_store_datadir = /opt/stack/data/glance/tasks_work_dir
[os_glance_staging_store]
filesystem_store_datadir = /opt/stack/data/glance/staging

[cinder]
cinder_os_region_name=RegionOne
cinder_store_auth_address=http://10.0.132.12/identity/v3
# cinder_store_auth_address 为 openstack 实际的 identity 服务 url，对应的账户密码需找管理员获取。
cinder_store_user_name=cinder
cinder_store_password=nomoresecret
cinder_store_project_name=service
cinder_volume_type=stor
cinder_enforce_multipath=True
cinder_use_multipath=True  rootwrap_config = /etc/cinder/rootwrap.conf
```

2. 命令行创建 images。

```
[root@server devstack]# glance image-create --name centos-instance-test --visibility
public --disk-format qcow2 --container-format bare --file /home/gw/cirros-0.3.4-x86_64-
disk.img --store cinder
+------------------+--------------------------------------+
| Property         | Value                                |
+------------------+--------------------------------------+
| checksum         | None                                 |
| container_format | bare                                 |
```

```
| created_at       | 2024-07-04T09:40:34Z                 |
| disk_format      | qcow2                                |
| id               | a59ef2b0-c462-41e0-a1b8-64a663e445bc |
| min_disk         | 0                                    |
| min_ram          | 0                                    |
| name             | centos-instance-test                 |
| os_hash_algo     | None                                 |
| os_hash_value    | None                                 |
| os_hidden        | False                                |
| owner            | 590fadcbc3a0414385ac1b6b7e24e6f9     |
| protected        | False                                |
| size             | None                                 |
| status           | queued                               |
| tags             | []                                   |
| updated_at       | 2024-07-04T09:40:34Z                 |
| virtual_size     | Not available                        |
| visibility       | public                               |
+------------------+--------------------------------------+
```

3. 检查创建的 images。

```
[root@server devstack]# glance image-list
+--------------------------------------+----------------------+
| ID                                   | Name                 |
+--------------------------------------+----------------------+
| a59ef2b0-c462-41e0-a1b8-64a663e445bc | centos-instance-test |
| 9c770cac-8a99-44ce-9791-19f0e57b256c | centos-instance1     |
| 34482177-1de6-4e3b-a394-5dd092d22272 | centos7-iso          |
| 4ddafeb0-3d4f-45ef-b566-2536577a0331 | centos7-qcow2        |
+--------------------------------------+----------------------+
```

7 常见问题

7.1 创建实例超时

修改计算节点的/etc/nova/nova.conf 配置文件。

```
[DEFAULT]
# 块设备允许重试最大次数。
block_device_allocate_retries = 600
# 块设备允许每次重试间隔时间，单位是秒。
block_device_allocate_retries_interval = 6
```

7.2 Dashboard session 超时

修改/etc/keystone/keystone.conf 配置文件。

```
[DEFAULT]
# 令牌应保持有效的时间，单位是秒。
expiration=7200
```