



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
1.3.0	2025 年 11 月 04 日	<ol style="list-style-type: none">支持查询存储池。支持 QoS 策略。支持查询快照大小和快照对应的源卷大小。	3.5-3.10
1.2.1	2025 年 06 月 04 日	<ol style="list-style-type: none">增加安装方式：复制 stor_driver 到 Cinder 驱动目录的方式安装驱动。	3.5-3.9
1.2.0	2025 年 04 月 27 日	<ol style="list-style-type: none">支持快照。支持通过快照方式创建新卷。支持通过卷方式创建新卷。支持备份。	3.5-3.9
1.1	2024 年 09 月 11 日	<p>第一次发版：</p> <ol style="list-style-type: none">支持本地卷和上云卷。支持设置卷相关参数。支持 Nova 和 Glance 使用 HBlock 作为后端存储。	3.5-3.7.2

目 录

1 产品定义	1
2 安装部署	3
2.1 前置条件	3
2.2 环境要求	5
2.3 安装驱动	6
2.3.1 驱动作为独立 Python 包安装	6
2.3.2 复制 stor_driver 到 Cinder 驱动目录的方式安装	7
3 配置插件	8
3.1 配置卷类型	8
4 卷操作	19
4.1 查看卷列表	19
4.2 查看卷详细信息	21
4.2.1 卷概况	21
4.2.2 卷快照	22
4.3 创建卷	24
4.4 连接卷	26
4.4.1 连接卷（卷页面）	26
4.4.2 连接卷（实例页面）	27
4.5 分离卷	28
4.5.1 分离卷（卷页面）	28
4.5.2 分离卷（实例页面）	29
4.6 扩展卷	30
4.7 删除卷	31
5 快照操作	32
5.1 命令行方式	32

5.1.1 创建快照	32
5.1.2 为快照重命名	35
5.1.3 修改快照描述信息	36
5.1.4 删除快照	37
5.1.5 回滚快照	38
5.1.6 查询所有快照	39
5.1.7 查询指定快照	41
5.2 WEB 方式	45
5.2.1 查看快照列表	45
5.2.2 查看快照详细信息	46
5.2.3 创建快照	47
5.2.4 编辑快照	48
5.2.5 删除快照	49
6 克隆卷操作	50
6.1 前置条件	50
6.2 命令行方式	54
6.2.1 通过快照方式创建新卷	54
6.2.2 通过卷方式创建新卷	57
6.3 WEB 方式	60
6.3.1 通过快照方式创建新卷	60
6.3.2 通过卷方式创建新卷	62
7 配置备份业务	64
7.1 配置备份驱动文件	64
7.2 全量备份卷	67
7.3 增量备份卷	69
7.4 查看备份列表	71
7.5 查看备份详情	72

7.6 删除备份	75
7.7 恢复备份	76
7.8 修改备份状态	77
8 存储池操作	78
8.1 查询存储池	78
9 QoS 策略	85
9.1 创建 QoS 策略	85
9.2 修改 QoS 策略	89
9.3 取消 QoS 策略的规则限制	92
9.4 将 QoS 策略绑定到卷类型	94
9.5 解除卷类型的 QoS 策略	95
9.6 删除 QoS 策略	96
9.7 查看卷类型关联的 QoS 策略	97
9.8 查看 QoS 策略列表	99
9.9 查看 QoS 策略详细信息	102
10 Nova 使用 HBlock 卷	105
10.1 前置条件	105
10.2 使用方法	105
11 Glance 使用 HBlock 卷	109
11.1 前置条件	109
11.2 使用方法	109
11.3 使用 HBlock 卷创建 Glance 镜像	116
12 常见问题	118
12.1 创建实例超时	118
12.2 Dashboard session 超时	118
12.3 设置 QoS 规格，消费者为前端或两者时，同时设置 total_bytes_sec 与 read_bytes_sec/write_bytes_sec 报错	118

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。

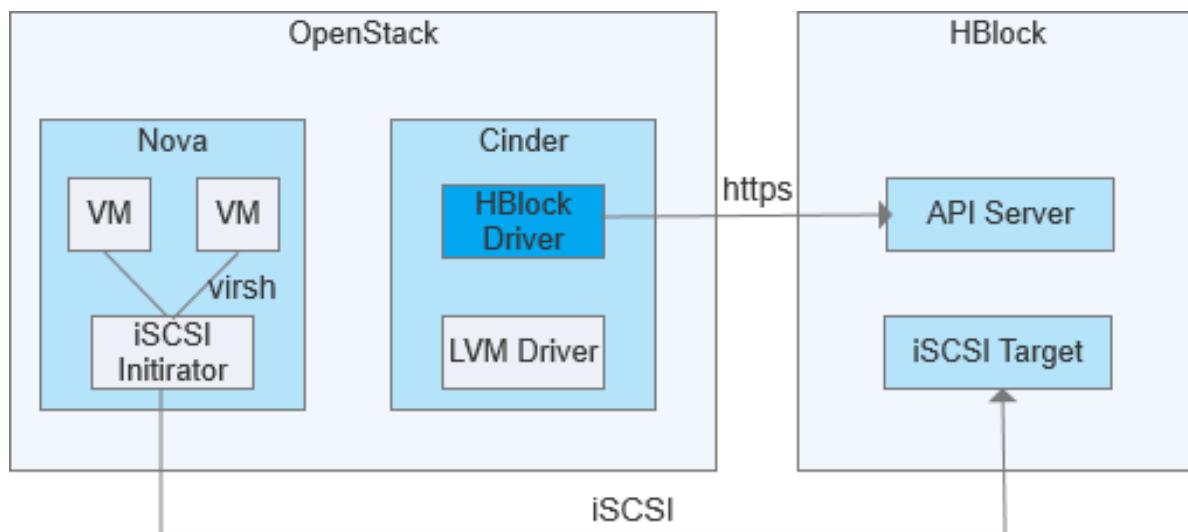


图1.HBlock对接OpenStack架构图

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。

如果对接 HBlock 集群版，Cinder、Glance、Nova 所在节点需要安装 iSCSI 工具及配置 MPIO。

安装 iSCSI 工具：

- 如果操作系统是 CentOS/RHEL，请安装 iscsi-initiator-utils，安装命令如下：

```
yum -y install iscsi-initiator-utils
```

注意：安装 iSCSI initiator 6.2.0-874-10 或以上版本。

- 如果操作系统是 Ubuntu/Debian，安装命令如下：

```
apt install open-iscsi
```

配置 MPIO 步骤：

1. 安装 MPIO

说明：如果已安装 MPIO，可以忽略此步骤。

- 对于 CentOS

```
yum install device-mapper-multipath device-mapper-multipath-libs      #CentOS
```

- 对于 Ubuntu

```
apt install multipath-tools      #Ubuntu
```

2. 配置 MPIO

- 1) 复制 `/usr/share/doc/device-mapper-multipath-X.Y.Z/multipath.conf` (其中 X.Y.Z 为 multipath 的实际版本号，请根据实际情况查找 multipath.conf) 到 `/etc/multipath.conf`。
- 2) 在 `/etc/multipath.conf` 中增加如下配置：

注意：配置文件 multipath.conf 中，如果 multipath 部分与 devices 部分中有相同参数，multipath 中的参数值会覆盖 devices 中的参数值。为了正常使用 HBlock 卷，需要删除 multipath 中的与下列字段相同的参数。

```
defaults {
    user_friendly_names yes
    find_multipaths yes
    uid_attribute "ID_WWN"
}

devices {
    device {
        vendor "CTYUN"
        product "iSCSI LUN Device"
        path_grouping_policy failover
        path_checker tur
        path_selector "round-robin 0"
        hardware_handler "1 alua"
        rr_weight priorities
        no_path_retry queue
        prio alua
    }
}
```

3. 重启 multipathd 服务

- 对于 CentOS

```
systemctl restart multipathd
systemctl enable multipathd
```

- 对于 Ubuntu

```
systemctl restart multipath-tools.service
systemctl enable multipath-tools.service
```

2.2 环境要求

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

2.3 安装驱动

在 Cinder 集群的节点上安装。可以通过两种方式安装驱动：驱动作为独立 Python 包安装、复制 stor_driver 到 Cinder 驱动目录的方式安装。不同的安装方式下，“HBlock 驱动所在路径”（对应配置卷类型中的参数 volume_driver）和“HBlock 备份驱动所在路径”（对应配置备份驱动文件中的参数 backup_driver）不同。

注意：非 root 用户安装，需要有其他用户的读权限。

2.3.1 驱动作为独立 Python 包安装

驱动作为独立 Python 包安装，执行下列步骤：

1. 解压安装包。

```
tar -zxvf stor-cinder-driver-1.2.1.tar.gz
```

2. 安装插件。

```
cd stor-cinder-driver-1.2.1
python setup.py install --prefix=prefix
```

说明：--prefix 为 OpenStack 安装路径前缀，OpenStack 安装在下列路径：

/xx/yy/lib/python_version/site-packages，那么--prefix=/xx/yy。例如：

OpenStack 安装的路径为：/usr/lib/python_version/site-packages，那么安装 Stor Cinder 驱动时需要指定 prefix 为/usr，即 python setup.py install --prefix=/usr。

执行结束后，会在 stor-cinder-driver-1.2.1/dist 目录下生成 egg 文件(例如：

stor_cinder_driver-1.2.1-py3.0.egg)，留存即可，通常无需手动分发此安装文件。

3. 重启 Cinder。

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

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。

2.3.2 复制 stor_driver 到 Cinder 驱动目录的方式安装

复制 stor_driver 到 Cinder 驱动目录的方式安装，执行下列步骤：

1. 解压安装包。

```
tar -zxvf stor-cinder-driver-1.2.1.tar.gz
```

2. 复制解压后的文件 stor_driver 至 site-packages/cinder/volume/drivers、site-packages/cinder/backup/drivers 下，作为 Cinder 库的一部分。

```
cp -r stor-cinder-driver-1.2.1/stor_driver /xx/yy/lib/python_version/site-packages/cinder/volume/drivers  
cp -r stor-cinder-driver-1.2.1/stor_driver /xx/yy/lib/python_version/site-packages/cinder/backup/drivers
```

说明：/xx/yy/lib 为 Python 的安装路径，*python_version* 为 Python 的安装版本。

3. 重启 Cinder。

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

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 HBlock 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://10.183.22.196:1443
write_policy = WriteBack
sector_size = 4096
max_sessions = 2
stor_path = /mnt/stor
flatten_volume_from_snapshot=false
max_clone_depth=5
```

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://10.183.22.197:1443,https://10.179.32.193:1443
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
flatten_volume_from_snapshot=false
max_clone_depth=5

[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://10.183.22.199:1443,https://10.179.32.193:1443
storage_mode = Local
```

```

local_storage_class = single-copy
high_availability = Disabled
write_policy = WriteAround
sector_size = 512
flatten_volume_from_snapshot=true
max_clone_depth=5

[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 = storuser
stor_api_password = *****
stor_chap_user = chap_user
stor_chap_password = *****
stor_api_endpoint = https://10.183.22.200:1443
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：

名称	描述	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：	是

	<ul style="list-style-type: none"> 如果驱动作为独立 Python 包进行安装, 取值为 stor_driver.driver.driver_stor.StorDriver。 如果复制 stor_driver 到 Cinder 驱动目录的方式安装驱动, 取值为: cinder.volume.drivers.stor_driver.driver_stor.StorDriver。 	
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 定义的后端存储名称保持一致。	是
stor_provider	产品名称。 取值: HBlock。	是
stor_api_user	HBlock 的管理员用户名。 说明: HBlock 初始化时, 默认用户名为 storuser。	是
stor_api_password	HBlock 的管理员密码。	是
storage_mode	HBlock 卷的模式。 取值: <ul style="list-style-type: none"> 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	<p>配置 HBlock RESTful API 地址和端口。</p> <p>对于集群版 HBlock，可以填写多对 HBlock RESTful API 地址和端口，以英文逗号隔开。</p> <p>说明：安装 HBlock 时，默认 API 端口为 1443。</p>	是
stor_path	<p>指定 HBlock 存储卷数据的数据目录（仅单机版支持）。</p> <p>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。</p>	否
server_numbers	<p>Target 所在的服务器数量（仅集群版支持）。</p> <p>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。</p>	否
fault_domains	<p>卷的服务端连接位置信息。根据存储池的故障域，创建 Target 所在服务器的列表（仅集群版支持），以便创建 LUN 时，LUN 关联的 Target 优先从该服务器列表中选择所在服务器。例如存储池为 rack 级别，其拓扑图涵盖 rack1、rack2、rack3、rack4 中的节点，且 faultDomains 指定 rack1、rack2，那么创建 LUN 时，LUN 关联的 Target 优先从 rack1、rack2 所包含的此存储池的服务器列表里进行选择。</p> <p>注意：</p> <ul style="list-style-type: none"> ● 存储池的故障域为 path 级别时，不能设置该参数。 ● 如果 LUN 指定了高速缓存池和最终存储池，则从高速缓存池中选择节点列表。如果 LUN 只指定了最终存储池，则从最终存储池中选择节点列表。 <p>取值：以节点的形式添加，节点的级别可以到 room、rack、server。可以指定多个节点，但是节点的个数要小于等于 serverNumbers。支持一个节点添加多次，但是每次只能选一个 server，并且选择的 server 不能与前面重复。如果一个节点出现的次数过多导致节点内的全部 server 都被选择，则系统会忽略此节点，从后面的节点中继续选择。</p>	否

pool	<p>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。</p> <p>注意：已用该卷类型创建卷，禁止修改 pool，否则将影响已创建的卷。如果需要修改 pool，需要创建新的卷类型。</p>	否
cache_pool	<p>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。</p> <p>注意：存储池与缓存存储池不能是同一个存储池。</p>	否
high_availability	<p>选择卷的高可用类型（仅集群版支持）：</p> <ul style="list-style-type: none"> ● 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。 <p>默认值为 ActiveStandby。</p>	否
local_storage_class	<p>卷冗余模式（仅集群版支持）。</p> <p>取值：</p> <ul style="list-style-type: none"> ● 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。 <p>默认值为 EC 2+1。</p>	否
min_replica	<p>最小副本数（仅集群版支持）。</p> <p>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。</p>	否

	取值：整数。对于副本卷，取值范围是[1, N]，N 为副本模式卷的副本数，默认值为 1。对于 EC 卷，取值范围是[N, N+M]，默认值为 N。	
redundancy_overlap	<p>卷的折叠副本数（仅集群版支持）。在数据冗余模式下，同一份数据的不同副本/分片默认分布在不同的故障域，当故障域损坏时，允许根据卷的冗余折叠原则，将多份数据副本放在同一个故障域中，但是分布在不同的 path 上。</p> <p>注意：如果存储池故障域级别为 path，此参数不生效。</p> <p>取值：对副本模式，取值范围是[1, 副本数]，默认值为 1；对于 EC 模式，取值范围是[1, M+N]，默认值为 1。</p>	否
ec_fragment_size	<p>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。</p> <p>取值：1、2、4、8、16、32、64、128、256、512、1024、2048、4096，单位是 KiB。默认值为 16。</p>	否
sector_size	<p>设置扇区大小。</p> <p>取值：512、4096，单位为字节。默认值为 4096。</p>	否
write_policy	<p>卷的写策略：</p> <ul style="list-style-type: none"> ● 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。 <p>默认值为 WriteBack (wb)。</p>	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
flatten_volume_from_snapshot	是否断开卷与快照的关系链： <ul style="list-style-type: none">● true：断开。● false：不断开。 默认值为 false。	否
max_clone_depth	最大克隆卷深度，当超过设置的最大深度时，新创建的卷会自动执行断链。 取值：整数，取值范围是[0, 15]，默认值为 5。0 表示不支持克隆卷。	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。 取值：字符串形式，长度范围是 1~256。	否
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。	否

	取值: 128、256、512、1024、2048、4096、8192, 单位是 KiB。 默认值为 1024。	
cloud_storage_class	设置 OOS 的存储类型: <ul style="list-style-type: none">● 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: <ul style="list-style-type: none">● 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: <ul style="list-style-type: none">● 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: <ul style="list-style-type: none">● true: 删除云上数据。● false: 不删除云上数据。 默认值为 true。	否

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. 验证卷类型是否配置正确

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

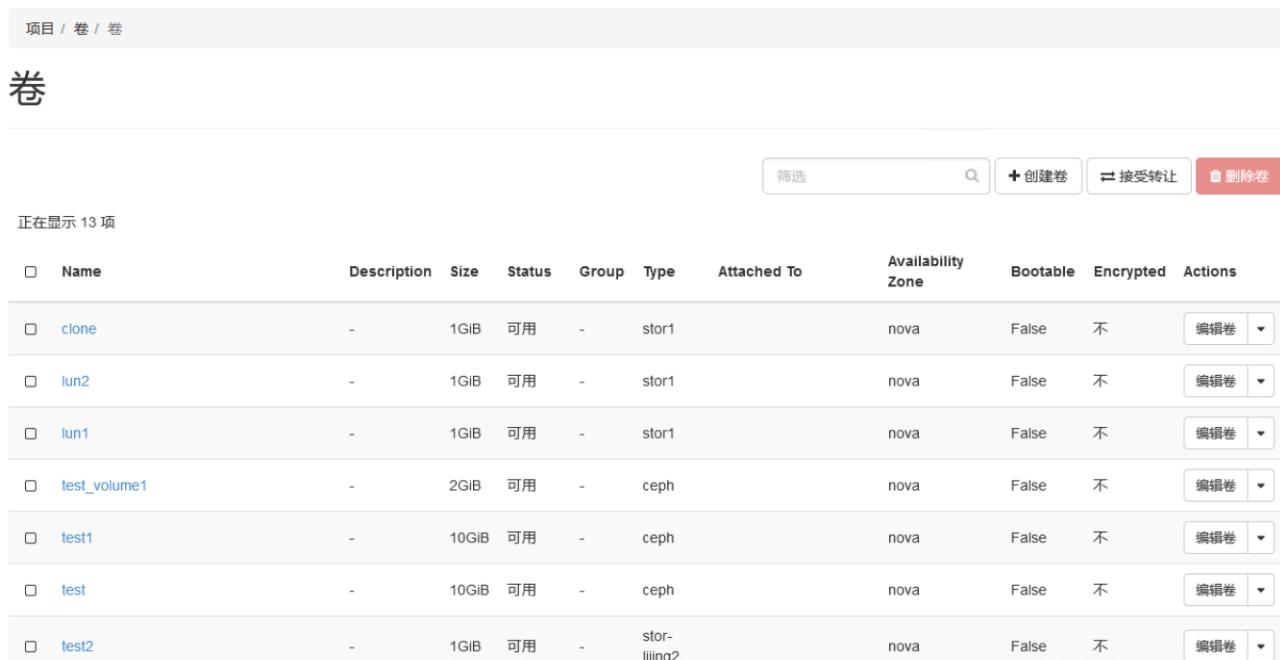
例如：

```
[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 卷操作

4.1 查看卷列表

登录 OpenStack，点击“卷”，进入“卷”页面，可以查看卷列表。



正在显示 13 项

<input type="checkbox"/> Name	Description	Size	Status	Group	Type	Attached To	Availability Zone	Bootable	Encrypted	Actions
<input type="checkbox"/> clone	-	1GiB	可用	-	stor1		nova	False	不	<button>编辑卷</button> ▾
<input type="checkbox"/> lun2	-	1GiB	可用	-	stor1		nova	False	不	<button>编辑卷</button> ▾
<input type="checkbox"/> lun1	-	1GiB	可用	-	stor1		nova	False	不	<button>编辑卷</button> ▾
<input type="checkbox"/> test_volume1	-	2GiB	可用	-	ceph		nova	False	不	<button>编辑卷</button> ▾
<input type="checkbox"/> test1	-	10GiB	可用	-	ceph		nova	False	不	<button>编辑卷</button> ▾
<input type="checkbox"/> test	-	10GiB	可用	-	ceph		nova	False	不	<button>编辑卷</button> ▾
<input type="checkbox"/> test2	-	1GiB	可用	-	stor-liina2		nova	False	不	<button>编辑卷</button> ▾

名称	描述
Name	卷名称。
Description	对卷的说明。
Size	卷的容量。
Status	卷的状态： <ul style="list-style-type: none"> ● 可用 ● 正在使用 ● 错误 ● 删除中
Type	卷类型。

Attached To	连接到实例的具体位置。
Availability Zone	卷的可用域。
Bootable	卷是否可启动。
Actions	<p>对卷的操作，可以：</p> <ul style="list-style-type: none">● 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。

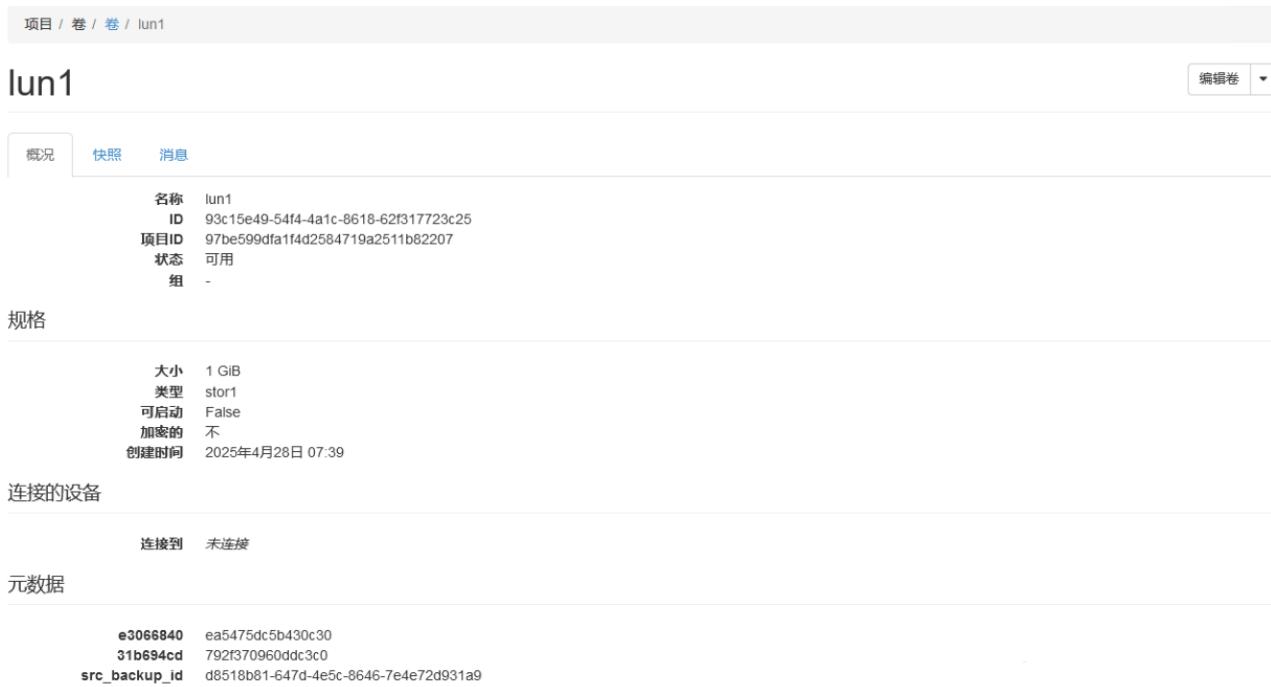
4.2 查看卷详细信息

登录 OpenStack，点击“卷”，进入“卷”页面，点击需要查看卷详细信息的卷，即可查看卷详细，包括卷的概况、快照、消息。

说明：本节主要介绍与 HBlock 相关的卷概况和快照信息。

4.2.1 卷概况

在卷详细信息页面，点击“概况”，即可查看卷概况信息。



The screenshot shows the 'Volume' details page for 'lun1'. The top navigation bar includes '项目 / 卷 / 卷 / lun1'. The main content area is titled 'lun1' with a '编辑卷' button. Below the title, there are three tabs: '概况' (selected), '快照', and '消息'. The '概况' tab displays the following information:

名称	lun1
ID	93c15e49-54f4-4a1c-8618-62f317723c25
项目ID	97be599dfa1f4d2584719a2511b82207
状态	可用
组	-

Below this, the '规格' section shows:

大小	1 GiB
类型	stor1
可启动	False
加密的	不
创建时间	2025年4月28日 07:39

The '连接的设备' section shows '连接到 未连接'.

The '元数据' section lists:

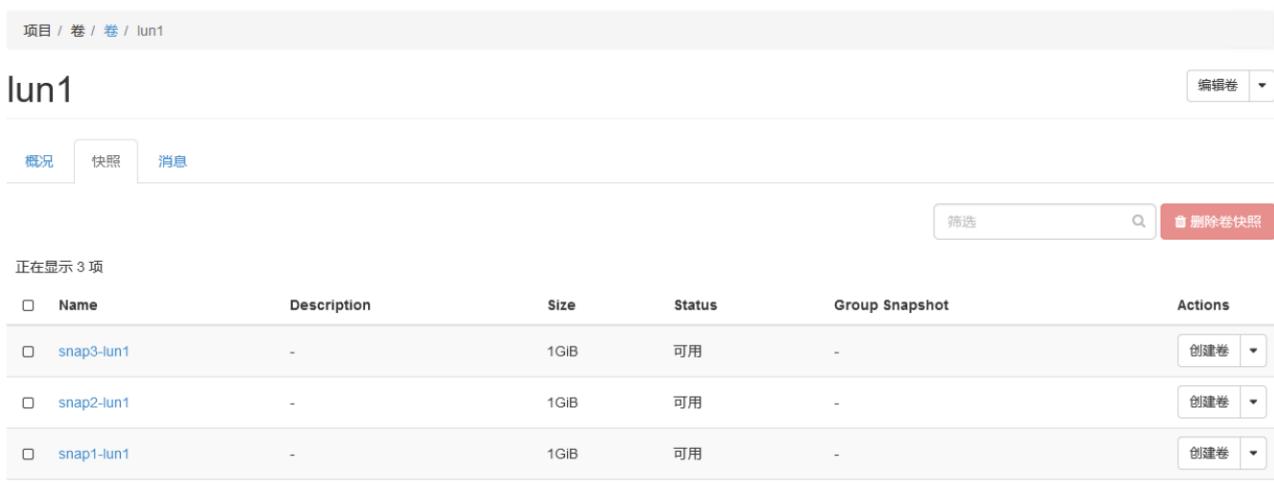
e3066840	ea5475dc5b430c30
31b694cd	792f370960ddc3c0
src_backup_id	d8518b81-647d-4e5c-8646-7e4e72d931a9

名称	描述
名称	卷名称。
ID	卷 ID。
项目 ID	所属项目的唯一标识符，标识该卷属于哪个项目。
状态	卷的状态： <ul style="list-style-type: none"> ● 可用。 ● 正在使用。 ● 错误。

	● 删除中。
组	卷所属的组，如果是“-”表示没有分配到特定的组。
大小	卷的容量。
类型	卷类型。
可启动	该卷是否可启动。
加密的	该卷是否加密。
创建时间	卷的创建时间。
连接到	连接到实例的具体位置。
元数据	卷的元数据。

4.2.2 卷快照

在卷详细信息页面，点击“快照”，即可查看卷快照相关信息。



正在显示 3 项

<input type="checkbox"/> Name	Description	Size	Status	Group Snapshot	Actions
<input type="checkbox"/> snap3-lun1	-	1GiB	可用	-	<input type="button" value="创建卷"/>
<input type="checkbox"/> snap2-lun1	-	1GiB	可用	-	<input type="button" value="创建卷"/>
<input type="checkbox"/> snap1-lun1	-	1GiB	可用	-	<input type="button" value="创建卷"/>

正在显示 3 项

名称	描述
Name	快照名称。
Description	快照的描述信息。
Size	快照的大小。
Status	快照的状态。
Group Snapshot	与卷快照相关的组快照信息。

Actions	<p>对快照的操作，可以：</p> <ul style="list-style-type: none">● 创建卷。● 编辑快照。● 删除卷快照。● 更新元数据。
---------	--

4.3 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。

创建卷

卷名称 说明：卷是可被连接到实例的块设备。

描述 卷类型描述：iscsi 没有可用的描述。

卷来源 卷限度 总大小 (GB) 已用 573, 共 1000 GiB

类型 卷数量 16 已使用, 共 1100

大小(GiB)*

可用域

组 取消 创建卷

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	<p>创建卷的来源：</p> <ul style="list-style-type: none">● 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。● 快照：通过快照创建卷，详见通过快照方式创建新卷。● 卷：通过卷创建新的卷，详见通过卷方式创建新卷。
使用镜像作为源	<p>选择一个镜像，作为卷的源。</p> <p>说明：如果“卷来源”项选择的是“镜像”，才会有该项。</p>
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）	<p>卷容量。</p> <p>取值：[1,1048576]，单位是 GiB。</p>
可用域	<p>可用域。</p> <p>说明：如果“卷来源”项选择的是“卷”，则不会显示该项。</p>

4.4 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。

4.4.1 连接卷（卷页面）

登录 OpenStack，点击“卷”，进入“卷”页面，选择需要连接的卷，然后点击其“Actions”列的“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。

管理已连接卷



Instance	Device	Actions
没有要显示的条目。		

连接到实例

连接到实例 

test-instance5 (cb3e1eb5-2f8e-4909-8043-b2ff9397145c) 

取消

连接卷

4.4.2 连接卷（实例页面）

登录 OpenStack，点击“计算”>“实例”，进入“实例”页面，选择要连接卷的实例，然后点击其“Actions”列的“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。

连接卷

卷 ID * ?

说明：
连接卷到正在运行的实例上

选择一个卷

取消 连接卷

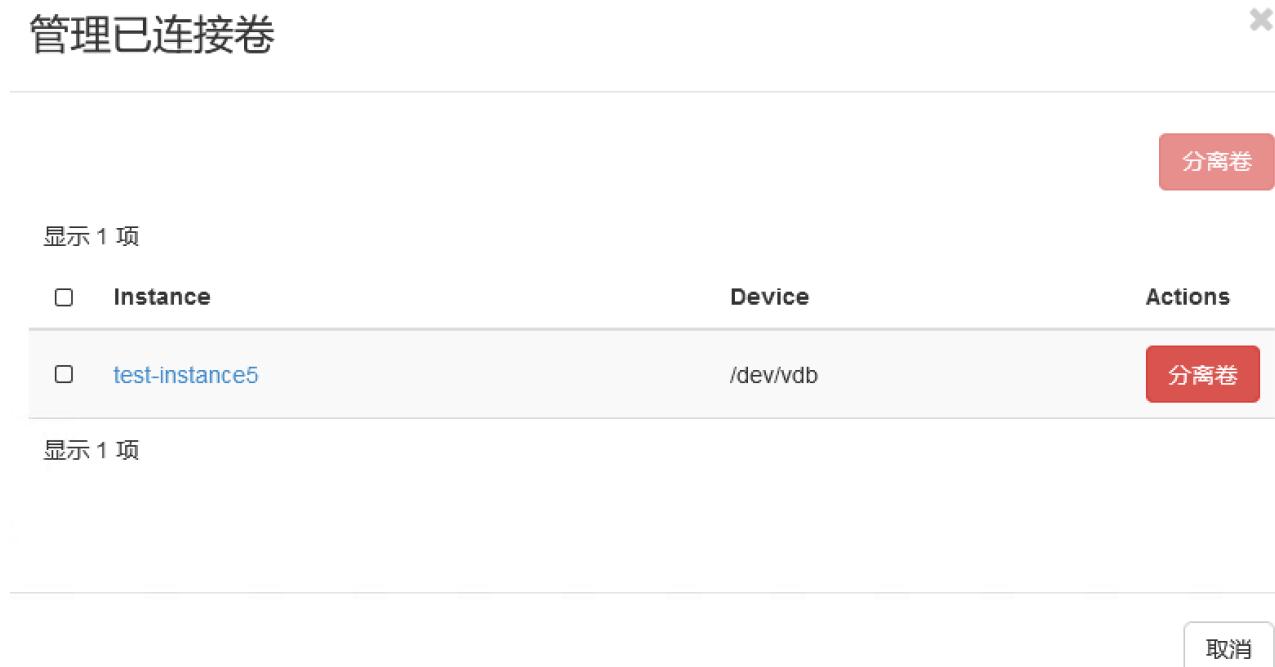
4.5 分离卷

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。

4.5.1 分离卷（卷页面）

登录 OpenStack，点击“卷”，进入“卷”页面，选择需要分离卷的卷，然后点击其“Actions”列的“管理连接”，弹出“管理已连接卷”页面，可在点击“Actions”列的“分离卷”按钮，即可分离卷。



Instance	Device	Actions
<input type="checkbox"/> test-instance5	/dev/vdb	分离卷

4.5.2 分离卷（实例页面）

登录 OpenStack，点击“计算”>“实例”，进入“实例”页面，选择要分离卷的实例，然后点击其“Actions”列的“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。



4.6 扩展卷

登录 OpenStack，点击“卷”，进入“卷”页面，选择需要扩展卷，然后点击其“Actions”列“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。

扩展卷

×

卷名称	说明：
clone-snap5	扩展卷的大小
当前大小(GiB)	卷限度
30	总大小 (GB) 已用 543, 共 1000 GiB
新大小(GiB) *	
<input type="text" value="30"/>	<input type="button" value="▲"/> <input type="button" value="▼"/>

取消 扩展卷

4.7 删除卷

登录 OpenStack，点击“卷”，进入“卷”页面，选择需要删除卷，点击其“Actions”列的“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。
- 如果卷有关联的快照，通过 WEB 页面删除卷时，必须先删除关联的快照后，才能再删除卷。

确认删除卷



您已选择了 "lun1"。请确认您的选择。删除卷是不可恢复的。卷中所有的数据都会被移除

 取消 删除卷

5 快照操作

5.1 命令行方式

5.1.1 创建快照

OpenStack 命令行：

```
openstack volume snapshot create --volume { volume_id | volume_name } [ snapshot_name ]  
[ --description snapshot_description ]
```

Cinder 命令行：

```
cinder snapshot-create { volume_id | volume_name } [ --name snapshot_name ] [ --  
description snapshot_description ]
```

此命令用来创建快照。

参数

参数	描述
<code>volume_id</code>	要创建快照的卷的 ID。
<code>volume_name</code>	要创建快照的卷的名称。
<code>--name snapshot_name</code>	指定快照的名称。 注意： 这里的快照名称与 HBlock 中的快照名称不同，HBlock 中的快照名称，对应 Cinder 中的快照 ID 前面增加 <code>snapshot-</code> 。
<code>--description</code> <code>snapshot_description</code>	指定快照的描述信息。 说明： 此描述信息仅为快照在 Cinder 上的描述信息，不是快照在 HBlock 端的描述信息。

示例

创建快照 snapshot-w5。

```
[root@controller cinder(keystone_admin)]# openstack volume snapshot create --volume lun_001
snapshot-w5 --description "This is a test snapshot"
+-----+-----+
| Field      | Value
+-----+-----+
| created_at | 2025-04-25T02:01:35.780265
| description | This is a test snapshot
| id          | 8f65856b-c5b6-4f45-b171-472f284f2fd7
| name        | snapshot-w5
| properties  |
| size        | 1
| status      | creating
| updated_at  | None
| volume_id   | 6b1649ea-48ec-4392-b4e8-f919cf484f43
+-----+-----+
```

创建快照 snapshot-w3。

```
[root@controller cinder(keystone_admin)]# cinder snapshot-create lun_001 --name snapshot-w3 --description "This is a test snapshot"
+-----+-----+
| Property      | Value
+-----+-----+
| consumes_quota | True
| created_at     | 2025-04-25T02:02:04.712369
| description     | This is a test snapshot
| group_snapshot_id | None
| id             | ff273f8d-a35f-44da-8be4-1f92b7749ed2
| metadata        | {}
| name            | snapshot-w3
| size            | 1
| status          | creating
| updated_at      | None
+-----+-----+
```

user_id	d60fa5458fda4f96bbf581bf6c2dd63e	
volume_id	6b1649ea-48ec-4392-b4e8-f919cf484f43	
+-----+-----+-----+		

5.1.2 为快照重命名

OpenStack 命令行：

```
openstack volume snapshot set --name new_name { snapshot_id | snapshot_name }
```

Cinder 命令行：

```
cinder snapshot-rename { snapshot_id | snapshot_name } new_name
```

此命令用来为快照重命名。

参数

参数	描述
<i>snapshot_id</i>	快照 ID。
<i>snapshot_name</i>	快照名称。
<i>new_name</i>	快照的新名称。

示例

为 ID 是 d6929c22-2c90-40a7-bf9d-f5eac1d80a22 的快照重命名。

```
[root@controller cinder(keystone_admin)]# openstack volume snapshot set --name snapshot-w2new  
d6929c22-2c90-40a7-bf9d-f5eac1d80a22
```

为 ID 是 6ebbd38a-98f3-49e9-95ea-63ccfa567490 的快照重命名。

```
[root@controller cinder(keystone_admin)]# cinder snapshot-rename 6ebbd38a-98f3-49e9-95ea-63ccfa567490  
snapshot-w5new  
Request to rename snapshot '6ebbd38a-98f3-49e9-95ea-63ccfa567490' has been accepted.
```

为名称是 snapshot-w3 的快照重命名。

```
[root@controller cinder(keystone_admin)]# cinder snapshot-rename snapshot-w3 snapshot-w3new  
Request to rename snapshot 'snapshot-w3' has been accepted.
```

5.1.3 修改快照描述信息

```
openstack volume snapshot set --description new_description { snapshot_id |  
snapshot_name }
```

此命令用来修改快照描述信息。

说明：此描述信息仅为快照在 Cinder 上的描述信息，不是快照在 HBlock 端的描述信息。

参数

参数	描述
<i>new_description</i>	快照的新描述信息。
<i>snapshot_id</i>	快照 ID。
<i>snapshot_name</i>	快照名称。

示例

为名称是 snapshot-w2 的快照修改描述信息。

```
[root@controller cinder(keystone_admin)]# openstack volume snapshot set --description "This  
is the second snapshot" snapshot-w2
```

5.1.4 删除快照

OpenStack 命令行：

```
openstack volume snapshot delete { snapshot_id | snapshot_name }
```

Cinder 命令行：

```
cinder snapshot-delete { snapshot_id | snapshot_name }
```

此命令用来删除快照。

参数

参数	描述
<i>snapshot_id</i>	快照 ID。
<i>snapshot_name</i>	快照名称。

示例

删除名称为 snapshot-w5 的快照。

```
[root@controller cinder(keystone_admin)]# openstack volume snapshot delete snapshot-w5
```

删除名称为 snapshot-w3 的快照。

```
[root@controller cinder(keystone_admin)]# cinder snapshot-delete snapshot-w3
```

删除 ID 为 4ae02758-7857-4bd1-864b-7541d37b81bc 的快照。

```
[root@controller cinder(keystone_admin)]# cinder snapshot-delete 4ae02758-7857-4bd1-864b-7541d37b81bc
```

5.1.5 回滚快照

```
cinder revert-to-snapshot { snapshot_id | snapshot_name }
```

此命令用来回滚快照。

说明：仅支持卷通过最新的快照进行回滚。

注意：如果卷在打快照后进行了扩容，则不能回滚快照。

参数

参数	描述
<i>snapshot_id</i>	快照 ID。
<i>snapshot_name</i>	快照名称。

示例

回滚快照 21ae5c47-5363-4c05-8bd6-213d06f30a97。

```
[root@controller cinder(keystone_admin)]# cinder revert-to-snapshot 21ae5c47-5363-4c05-8bd6-213d06f30a97
```

5.1.6 查询所有快照

OpenStack 命令行：

```
openstack volume snapshot list
```

Cinder 命令行：

```
cinder snapshot-list
```

此命令用来查询所有快照。

示例

- 使用 OpenStack 命令行查询所有快照。

```
[root@controller cinder(keystone_admin)]# openstack volume snapshot list
+-----+-----+-----+-----+-----+
| ID | Name | Description | Status | Size |
+-----+-----+-----+-----+-----+
| 636251bf-3cce-4356-a327-b3ae8a1bc18e | snapshot-w2 | None | available | 1 |
| ef6ff9c7-5480-4744-8d19-a644d5d4ef78 | snapshot-w1 | None | error | 1 |
| 87c848eb-4427-4164-9b41-e0429117683a | snap1 | None | available | 1 |
| 7a154b59-909c-4635-85c1-8a02abf3ce3c | snap1 | None | available | 1 |
+-----+-----+-----+-----+-----+
```

- 使用 Cinder 命令行查询所有快照。

```
[root@controller cinder(keystone_admin)]# cinder snapshot-list
+-----+-----+-----+-----+-----+-----+-----+
| ID | Volume ID | Status | Name | Size | Consumes Quota | User ID |
+-----+-----+-----+-----+-----+-----+-----+
| 636251bf-3cce-4356-a327-b3ae8a1bc18e | 6c339b63-5432-4e61-b980-c601cb103398 | available | snapshot-w2 | 1 | True | d60fa5458fda4f96bbf581bf6c2dd63e |
| 7a154b59-909c-4635-85c1-8a02abf3ce3c | 6c339b63-5432-4e61-b980-c601cb103398 | available | snap1 | 1 | True | d60fa5458fda4f96bbf581bf6c2dd63e |
| 87c848eb-4427-4164-9b41-e0429117683a | c55c284c-7f44-4fe4-806b-04d96b2cb86b | available | snap1 | 1 | True | d60fa5458fda4f96bbf581bf6c2dd63e |
| ef6ff9c7-5480-4744-8d19-a644d5d4ef78 | 0e87db4b-f6a3-457f-99d8-e31109554c22 | error | snapshot-w1 | 1 | True | d60fa5458fda4f96bbf581bf6c2dd63e |
+-----+-----+-----+-----+-----+-----+-----+
```

快照信息描述

项目	描述
ID	快照的 ID，唯一标识符（UUID）。 说明： 在 HBlock 显示的快照名称为 ID 前面增加 snapshot-。
Volume ID	快照的源卷 ID。
Status	快照的状态。
Name	快照的名称。
Size	创建快照时刻，源卷的容量。单位是 GiB。
Consumes Quota	是否消耗配额，通常表示快照是否计入用户配额限制。
User ID	创建快照的用户的 UUID。

5.1.7 查询指定快照

OpenStack 命令行：

```
openstack volume snapshot show { snapshot_id | snapshot_name }
```

Cinder 命令行：

```
cinder snapshot-show { snapshot_id | snapshot_name }
```

此命令用来查询指定快照信息。

参数

参数	描述
<i>snapshot_id</i>	要查询快照的 ID。
<i>snapshot_name</i>	要查询快照的名称。

示例

查询快照 s1（快照 ID 为 83919052-835c-441a-bf3b-8c38e01d0978）。

[root@controller cinder(keystone_admin)]# openstack volume snapshot show 83919052-835c-441a-bf3b-8c38e01d0978
<pre>+-----+-----+ Field Value +-----+-----+ created_at 2025-09-29T10:01:22.000000 description None id 83919052-835c-441a-bf3b-8c38e01d0978 name s1 os-extended-snapshot-attributes:progress 100% os-extended-snapshot-attributes:project_id 97be599dfa1f4d2584719a2511b82207 properties snapshot_size='4096' size 1 status available updated_at 2025-09-29T10:03:18.000000 volume_id 7d9aefaa-6a93-4b21-a30d-1cfe30c622e1 </pre>

Field	Value
-------	-------

查询快照 s1 (快照 ID 为 83919052-835c-441a-bf3b-8c38e01d0978)。

```
[root@controller cinder(keystone_admin)]# openstack volume snapshot show s1
```

Field	Value
created_at	2025-09-29T10:01:22.000000
description	None
id	83919052-835c-441a-bf3b-8c38e01d0978
name	s1
os-extended-snapshot-attributes:progress	100%
os-extended-snapshot-attributes:project_id	97be599dfa1f4d2584719a2511b82207
properties	snapshot_size='4096'
size	1
status	available
updated_at	2025-09-29T10:03:18.000000
volume_id	7d9aefaa-6a93-4b21-a30d-1cfe30c622e1

查询快照 s1 (快照 ID 为 83919052-835c-441a-bf3b-8c38e01d0978)。

```
[root@controller cinder(keystone_admin)]# cinder snapshot-show s1
```

Property	Value
consumes_quota	True
created_at	2025-09-29T10:01:22.000000
description	None
group_snapshot_id	None
id	83919052-835c-441a-bf3b-8c38e01d0978
metadata	{'snapshot_size': '4096'}
name	s1
os-extended-snapshot-attributes:progress	100%
os-extended-snapshot-attributes:project_id	97be599dfa1f4d2584719a2511b82207

size	1	
status	available	
updated_at	2025-09-29T10:03:18.000000	
user_id	d60fa5458fda4f96bbf581bf6c2dd63e	
volume_id	7d9aefaa-6a93-4b21-a30d-1cfe30c622e1	
+	-----+	-----+

快照信息描述

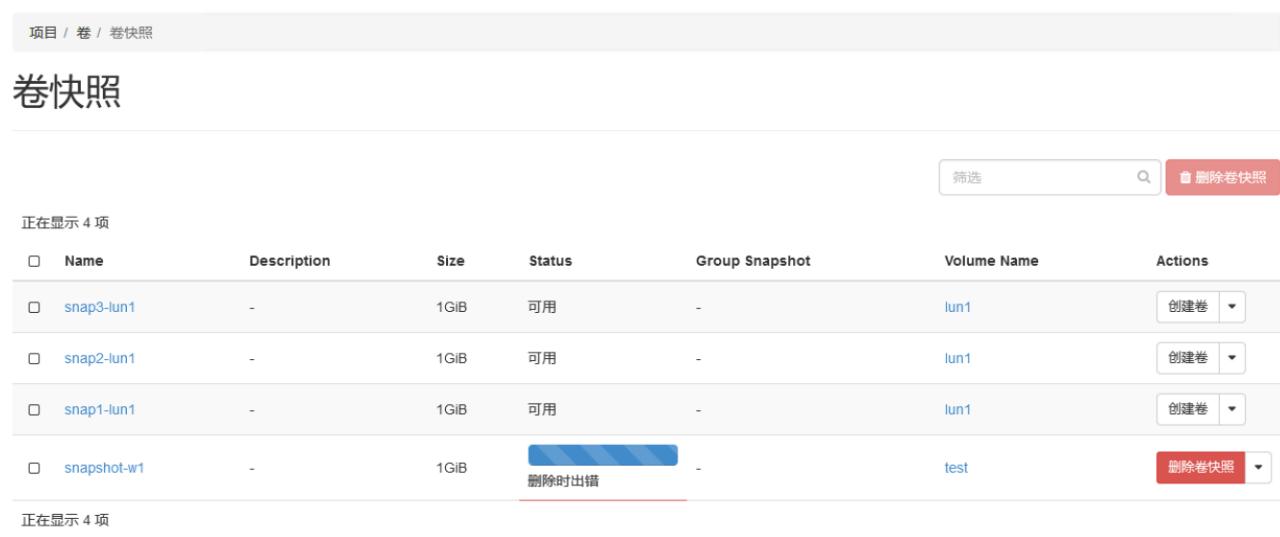
项目	描述
<code>consumes_quota</code>	是否消耗配额，通常表示快照是否计入用户配额限制。
<code>created_at</code>	快照创建的时间戳。
<code>description</code>	快照的描述，如果未指定则 None。
<code>group_snapshot_id</code>	该快照所属的组快照（Group Snapshot）的 ID。
<code>id</code>	快照的 ID，唯一标识符（UUID）。 说明： 在 HBlock 显示的快照名称为 ID 前面增加 snapshot-。
<code>properties、metadata</code>	表示快照的元数据。 <code>snapshot_size</code> 为快照的大小，单位是 Byte。 注意： 卷异常或上游快照删除等因素可能导致快照大小波动。
<code>name</code>	快照的名称。
<code>os-extended-snapshot-attributes:progress</code>	快照创建的进度。
<code>os-extended-snapshot-attributes:project_id</code>	快照所属项目 ID。
<code>size</code>	创建快照时刻，源卷的容量。单位是 GiB。
<code>status</code>	快照的状态。
<code>updated_at</code>	表示快照最后被更新的时间戳。
<code>user_id</code>	创建快照的用户的 UUID。

volume_id	快照的源卷 ID。
-----------	-----------

5.2 WEB 方式

5.2.1 查看快照列表

登录 OpenStack，点击“卷”>“快照”，进入“快照”页面，可以查看快照列表。



The screenshot shows the 'Volume Snapshot' list page in the OpenStack interface. The top navigation bar includes '项目 / 卷 / 卷快照'. The main title is '卷快照'. On the right, there are search and filter buttons, and a red '删除卷快照' (Delete Snapshot) button. The table lists four snapshots with the following details:

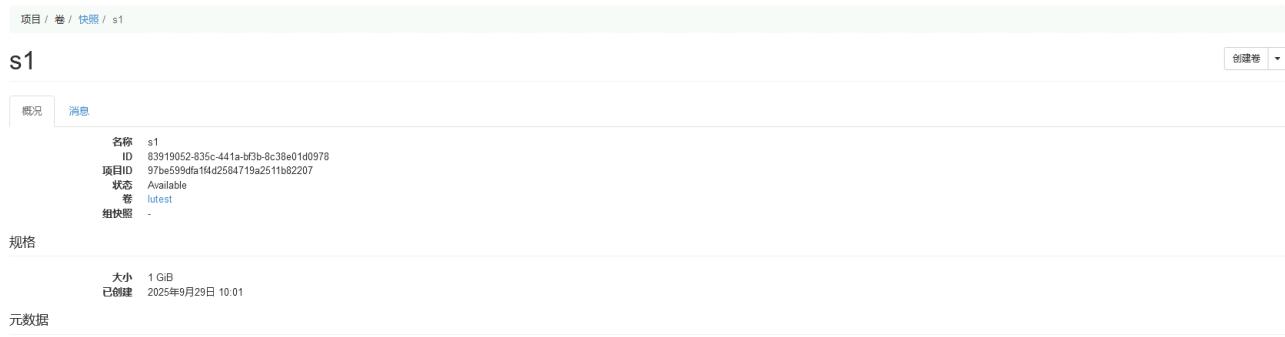
正在显示 4 项	Name	Description	Size	Status	Group Snapshot	Volume Name	Actions
□ snap3-lun1	-	1GiB	可用	-	-	lun1	<button>创建卷</button>
□ snap2-lun1	-	1GiB	可用	-	-	lun1	<button>创建卷</button>
□ snap1-lun1	-	1GiB	可用	-	-	lun1	<button>创建卷</button>
□ snapshot-w1	-	1GiB	删除时出错	-	-	test	<button>删除卷快照</button>

At the bottom, it says '正在显示 4 项'.

名称	描述
Name	快照名称。
Description	快照的描述信息。
Size	创建快照时刻，源卷的容量。
Status	快照的状态。
Group Snapshot	与卷快照相关的组快照信息。
Volume Name	源卷名称。
Actions	对快照的操作，可以 <ul style="list-style-type: none"> ● 创建卷。 ● 编辑快照。 ● 删除卷快照。 ● 更新元数据。

5.2.2 查看快照详细信息

登录 OpenStack，点击“卷”>“快照”，进入“快照”页面，点击需要查看快照详细信息的快照，即可查看快照详细。



项目 / 卷 / 快照 / s1

s1

创建卷

概况 消息

名称 s1
 ID 83919052-835c-441a-bf3b-8c38e01d0978
 项目ID 97be599dfa1f4d2584719a2511b82207
 状态 Available
 卷 lu-test
 组快照 -

规格

大小 1 GiB
 已创建 2025年9月29日 10:01

元数据

snapshot_size 4096

名称	描述
名称	快照名称。
ID	快照 ID。
项目 ID	快照所属项目 ID。
状态	快照的状态。
卷	快照的源卷名称。
组快照	与卷快照相关的组快照信息。
大小	创建快照时刻，源卷的容量。
已创建	快照创建的时间。
snapshot_size	snapshot_size 为快照的大小，单位是 Byte。 注意：卷异常或上游快照删除等因素可能导致快照大小波动。

5.2.3 创建快照

登录 OpenStack，点击“卷”，选择需要创建快照的卷，再点击其“Actions”列的“创建快照”，弹出“创建卷快照”页面，即可为该卷创建快照。

创建卷快照

快照名称

说明：
为一个卷创建一个快照

描述

快照限制

总大小 (GB) 已用 542, 共 1000 GiB

快照数量 2 已使用, 共 10

名称	描述
快照名称	要创建的卷快照名称。
描述	快照的描述信息。

5.2.4 编辑快照

登录 OpenStack，点击“卷”>“快照”，选择需要编辑的快照，再点击其“Actions”列的“编辑快照”，弹出“编辑快照”页面，即可为该快照重命名、修改描述信息。

编辑快照



快照名称

说明：

编辑快照的名称和描述

描述

名称	描述
快照名称	快照的新名称。
描述	快照的描述信息。

5.2.5 删除快照

登录 OpenStack，点击“卷”>“快照”，选择需要删除的快照，再点击其“Actions”列的“删除卷快照”，弹出“确认删除卷快照”页面，即可删除该快照。

确认删除卷快照



您已选择了 "snap-w2"。请确认您的选择。删除卷快照是不可恢复的

取消

删除卷快照

6 克隆卷操作

6.1 前置条件

说明：仅支持本地卷创建克隆卷。

创建克隆卷时，可以选择是否修改克隆卷的参数。如果不修改，继承源卷的参数，如果修改，可以按照下列步骤执行。

1. 在 Cinder 的配置文件/etc/cinder/cinder.conf 中修改源卷对应的卷类型参数。支持修改的参数详见下表，如果不修改，默认与源卷的参数保持一致。

参数	描述
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。
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。
fault_domains	卷的服务端连接位置信息。根据存储池的故障域，创建 Target 所在服务器的列表（仅集群版支持），以便创建 LUN 时，LUN 关联的 Target 优先从该服务器列表中选择所在服务器。 例如存储池为 rack 级别，其拓扑图涵盖 rack1、rack2、rack3、rack4 中的节点，且 faultDomains 指定 rack1、rack2，那么创建 LUN 时，LUN 关联的 Target 优先从 rack1、rack2 所包含的此存储池的服务器列表里进行选择。 注意： <ul style="list-style-type: none">● 存储池的故障域为 path 级别时，不能设置该参数。● 如果 LUN 指定了高速缓存池和最终存储池，则从高速缓存池池中选择节点列表。如果 LUN 只指定了最终存储

	<p>池，则从最终存储池中选择节点列表。</p> <p>取值：以节点的形式添加，节点的级别可以到 room、rack、server。可以指定多个节点，但是节点的个数要小于等于 serverNumbers。支持一个节点添加多次，但是每次只能选一个 server，并且选择的 server 不能与前面重复。如果一个节点出现的次数过多导致节点内的全部 server 都被选择，则系统会忽略此节点，从后面的节点中继续选择。</p>
cache_pool	<p>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。</p> <p>注意：存储池与缓存存储池不能是同一个存储池。</p>
high_availability	<p>选择卷的高可用类型（仅集群版支持）：</p> <ul style="list-style-type: none"> ● 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。
local_storage_class	<p>卷冗余模式（仅集群版支持）。</p> <p>取值：</p> <ul style="list-style-type: none"> ● 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。
min_replica	<p>最小副本数（仅集群版支持）。</p> <p>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</p>

	<p>本数设置为 Y (必须满足 $N \leq Y \leq N+M$)，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。</p> <p>取值：整数。对于副本卷，取值范围是[1, N]，N 为副本模式卷的副本数。对于 EC 卷，取值范围是[N, N+M]。</p>
redundancy_overlap	<p>卷的折叠副本数（仅集群版支持）。在数据冗余模式下，同一份数据的不同副本/分片默认分布在不同的故障域，当故障域损坏时，允许根据卷的冗余折叠原则，将多份数据副本放在同一个故障域中，但是分布在不同的 path 上。</p> <p>注意：如果存储池故障域级别为 path，此参数不生效。</p> <p>取值：对副本模式，取值范围是[1, 副本数]，默认值为 1；对于 EC 模式，取值范围是[1, M+N]，默认值为 1。</p>
ec_fragment_size	<p>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。</p> <p>取值：1、2、4、8、16、32、64、128、256、512、1024、2048、4096，单位是 KiB。</p>
sector_size	<p>设置扇区大小。</p> <p>取值：512、4096，单位为字节。</p>
write_policy	<p>卷的写策略：</p> <ul style="list-style-type: none"> ● 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。
--	--------

2. 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*
```

3. 通过快照方式创建新卷或者通过卷方式创建新卷。

6.2 命令行方式

6.2.1 通过快照方式创建新卷

OpenStack 命令行：

```
openstack volume create --snapshot { snapshot_id | snapshot_name } volume_name [ --size size ]
```

Cinder 命令行：

```
cinder create --snapshot-id snapshot_id --name volume_name [ size ]
```

此命令用来通过快照创建新卷。

参数

参数	描述
<i>snapshot_id</i>	快照 ID。
<i>snapshot_name</i>	快照名称。
<i>volume_name</i>	卷的名称。
<i>size</i>	卷的容量。 说明： 如果指定，必须大于等于快照时刻的源卷容量。

示例

创建卷 clone-w1，容量为 100 GiB。

```
[root@controller cinder(keystone_admin)]# openstack volume create --snapshot 025a1fbc-5423-44b0-85fc-913f07479842 clone-w1 --size 100
+-----+-----+
| Field          | Value           |
+-----+-----+
| attachments    | []              |
| availability_zone | nova           |
| bootable       | false          |
```

consistencygroup_id	None
created_at	2025-04-23T10:17:44.907350
description	None
encrypted	False
id	d579d547-ef69-4653-835a-66e7d30ce6bc
migration_status	None
multiattach	False
name	clone-w1
properties	
replication_status	None
size	100
snapshot_id	025a1fbc-5423-44b0-85fc-913f07479842
source_volid	None
status	creating
type	stor1
updated_at	None
user_id	d60fa5458fda4f96bbf581bf6c2dd63e
-----+-----+	

创建卷 clone-w4，容量为 100 GiB。

[root@controller cinder(keystone_admin)]# cinder create --snapshot-id 025a1fbc-5423-44b0-85fc-913f07479842 --name clone-w4 100	
-----+-----+	
Field	Value
-----+-----+	
attachments	[]
availability_zone	nova
bootable	false
consistencygroup_id	None
created_at	2025-04-23T10:17:44.907350
description	None
encrypted	False
id	d579d547-ef69-4653-835a-66e7d30ce6bc
migration_status	None
multiattach	False

name	clone-w1	
properties		
replication_status	None	
size	100	
snapshot_id	025a1fbc-5423-44b0-85fc-913f07479842	
source_volid	None	
status	creating	
type	stor1	
updated_at	None	
user_id	d60fa5458fda4f96bbf581bf6c2dd63e	
+-----+-----+-----+		

6.2.2 通过卷方式创建新卷

OpenStack 命令行：

```
Openstack volume create --source { source_volume_id | source_volume_name } [ --size
size ] volume_name
```

Cinder 命令行：

```
cinder create --source-volid source_volume_id --name volume_name [ size ]
```

此命令用来通过卷方式创建新卷。

参数

参数	描述
<i>source_volume_id</i>	源卷的 ID。 说明： 仅支持使用本地模式的 HBlock 卷作为源卷。
<i>source_volume_name</i>	源卷的名称。 说明： 仅支持使用本地模式的 HBlock 卷作为源卷。
<i>volume_name</i>	新卷的名称。
<i>size</i>	新卷的容量。 说明： 如果指定，必须大于等于源卷容量。

示例

创建卷 clone-w2，容量为 100 GiB。

```
[root@controller cinder(keystone_admin)]# openstack volume create --source 6b1649ea-
48ec-4392-b4e8-f919cf484f43 --size 100 clone-w2
+-----+-----+
| Field          | Value           |
+-----+-----+
| attachments    | []              |
| availability_zone | nova           |
| bootable       | false           |
```

consistencygroup_id	None
created_at	2025-04-24T02:10:25.527708
description	None
encrypted	False
id	82e538f2-25cb-49c8-935e-817a8a52d96d
migration_status	None
multiattach	False
name	clone-w2
properties	
replication_status	None
size	100
snapshot_id	None
source_volid	6b1649ea-48ec-4392-b4e8-f919cf484f43
status	creating
type	stor1
updated_at	None
user_id	d60fa5458fda4f96bbf581bf6c2dd63e

创建卷 clone-w5，容量为 200 GiB。

[root@controller cinder(keystone_admin)]# cinder create --source-volid 6b1649ea-48ec-4392-b4e8-f919cf484f43 --name clone-w5 200	
+-----+-----+-----+	
Property	Value
attachments	[]
availability_zone	nova
bootable	false
cluster_name	None
consistencygroup_id	None
consumes_quota	True
created_at	2025-04-24T02:28:48.000000
description	None
encrypted	False

group_id	None	
id	ed6503f3-7c4c-4f17-8e1d-ab0693ac2747	
metadata	{}	
migration_status	None	
multiattach	False	
name	clone-w5	
os-vol-host-attr:host	None	
os-vol-mig-status-attr:migstat	None	
os-vol-mig-status-attr:name_id	None	
os-vol-tenant-attr:tenant_id	97be599dfa1f4d2584719a2511b82207	
provider_id	None	
replication_status	None	
service_uuid	None	
shared_targets	True	
size	200	
snapshot_id	None	
source_valid	6b1649ea-48ec-4392-b4e8-f919cf484f43	
status	creating	
updated_at	None	
user_id	d60fa5458fda4f96bbf581bf6c2dd63e	
volume_type	stor1	
volume_type_id	104def3d-abf3-4a3b-b850-5ee7b8eaba3e	
+-----+-----+-----+		

6.3 WEB 方式

6.3.1 通过快照方式创建新卷

登录 OpenStack，点击“卷”，进入“卷”页面，点击“创建卷”按钮，弹出“创建卷”页面，在“卷来源”处选择“快照”，即可通过快照创建新卷。

说明：主要介绍 HBlock 卷相关的内容。

创建卷

卷名称

说明：

卷是可被连接到实例的块设备。

描述

卷限度

总大小 (GB)

已用 543, 共 1000 GiB

卷数量

15 已使用, 共 1100

卷来源

快照

使用快照作为源*

选择快照

大小(GiB)*

1

组②

没有组

取消

创建卷

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	快照：通过快照方式创建卷。
使用快照作为源	选择一个快照，作为创建卷的源。
大小 (GiB)	卷容量。 默认为源卷的容量，如果修改，必须大于等于快照时刻源卷容量。
组	新卷所属的组。

6.3.2 通过卷方式创建新卷

登录 OpenStack，点击“卷”，进入“卷”页面，点击“创建卷”按钮，弹出“创建卷”页面，在“卷来源”处选择“卷”，即可通过快照创建卷。

说明：主要介绍 HBlock 卷相关的内容。

创建卷

卷名称

说明：
卷是可被连接到实例的块设备。

描述

卷限度

总大小 (GB) 已用 573, 共 1000 GiB

卷数量 16 已使用, 共 1100

卷来源

卷

将卷作为源*

选择卷

大小(GiB)*

组②

没有组

取消 **创建卷**

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	卷：通过卷方式创建卷。
将卷作为源	选择一个卷，作为创建新卷的源卷。 说明： 仅支持使用本地模式的 HBlock 卷作为源卷。
大小 (GiB)	卷容量。 默认为源卷的容量，如果修改，必须大于等于源卷的容量。
组	新卷所属的组。

7 配置备份业务

7.1 配置备份驱动文件

前置条件：HBlock 集群版已经正确配置，且可以正常启动。

注意：

- 仅 HBlock 集群版支持配置备份驱动，且仅能配置一个集群。
- 仅支持本地模式的 HBlock 卷备份。

可以按照下列步骤配置备份驱动文件

1. 在 cinder.conf 中修改添加备份的配置文件：找到默认的 backup_driver ([default] 标签下），然后修改并扩充配置。

```
backup_driver=stor_driver.backup_driver.driver_stor.StorBackupDriver
backup_stor_stor_provider = HBlock
backup_stor_stor_api_user = storuser
backup_stor_stor_api_password = password
backup_stor_stor_api_endpoint = https://101.91.214.45:1443
backup_stor_pool = default
backup_stor_max_clone_depth = 5
backup_stor_local_storage_class = local_storage_class
backup_stor_min_replica = min_replica
backup_stor_ec_fragment_size = ec_fragment_size
backup_stor_max_sessions = max_sessions
```

参数说明

参数	描述	是否必填
backup_driver	<p>HBlock 备份驱动所在路径。</p> <p>取值：</p> <ul style="list-style-type: none">● 如果驱动作为独立 Python 包进行安装，取值为 stor_driver.backup_driver.driver_stor.StorBackupDriver。	是

	<ul style="list-style-type: none"> 如果复制 stor_driver 到 Cinder 驱动目录的方式安装驱动，取值为 cinder.backup.drivers.stor_driver.backup_driver.driver_stor.StorBackupDriver。 	
backup_stor_stor_provider	产品名称。 取值：HBlock。	是
backup_stor_stor_api_user	HBlock 的管理员用户名。 说明：HBlock 初始化时，默认用户名为 storuser。	是
backup_stor_stor_api_password	HBlock 的管理员密码。	是
backup_stor_stor_api_endpoint	配置 HBlock RESTful API 地址和端口。 说明：安装 HBlock 时，默认 API 端口为 1443。	是
backup_stor_pool	备份存储池，对应 HBlock 集群中的存储池名称。 注意： <ul style="list-style-type: none"> 一旦备份存储池配置完成并存储了备份数据，则不允许进行任何更改，否则将导致业务异常。 只支持同存储池增量备份卷，暂不支持跨存储池增量备份卷。 	是
backup_stor_max_clone_depth	备份驱动的最大克隆长度。 整数形式，取值为[1, 15]。默认值为 5。	否
backup_stor_local_storage_class	备份卷的冗余模。 取值： <ul style="list-style-type: none"> 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。 	否

	默认值为 EC 2+1。	
backup_stor_min_replica	<p>备份卷的最小副本数。</p> <p>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。</p> <p>取值：整数。对于副本卷，取值范围是[1, N]，N 为副本模式卷的副本数，默认值为 1。对于 EC 卷，取值范围是[N, N+M]，默认值为 N。</p>	否
backup_stor_ec_fragment_size	<p>备份卷的纠删码模式分片大小。卷冗余模式为 EC 模式时，此设置才生效，否则忽略。</p> <p>取值：1、2、4、8、16、32、64、128、256、512、1024、2048、4096，单位是 KiB。默认值为 16。</p>	否
backup_stor_max_sessions	<p>备份卷 iSCSI Target 允许建立的最大会话数。</p> <p>取值：整数，取值范围是[0, 1024]，默认值为 0，表示不允许客户端连接。</p>	否

2. 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*
```

7.2 全量备份卷

OpenStack 命令行：

```
openstack volume backup create --name backup-name { volume_id | volume_name }
```

Cinder 命令行：

```
cinder backup-create volume_id --name backup_name
```

此命令用来对卷进行全量备份。

参数

参数	描述
<i>backup-name</i>	备份名称。
<i>volume_id</i>	源卷 ID。
<i>volume_name</i>	源卷名称。

示例

为卷进行全量备份，备份名称为 back-w1。

```
[root@controller cinder(keystone_admin)]# openstack volume backup create --name back-w1  
da139514-d19f-4567-8070-c04db3a66de9  
+-----+-----+  
| Field | Value |  
+-----+-----+  
| id | b686b7b9-7783-4cda-a2cb-32f8cdeccbb4 |  
| name | back-w1 |  
| volume_id | da139514-d19f-4567-8070-c04db3a66de9 |  
+-----+-----+
```

为卷进行全量备份，备份名称为 back-w4。

```
[root@controller cinder(keystone_admin)]# cinder backup-create da139514-d19f-4567-8070-  
c04db3a66de9 --name back-w4  
+-----+-----+  
| Property | Value |  
+-----+-----+
```

id	2545a20d-cbd3-4005-89fa-54a60bef341b
name	back-w4
volume_id	da139514-d19f-4567-8070-c04db3a66de9

7.3 增量备份卷

OpenStack 命令行：

```
openstack volume backup create --incremental --name backup-name { volume_id |  
volume_name }
```

Cinder 命令行：

```
cinder backup-create --incremental volume_id --name backup_name
```

此命令用来对卷进行增量备份。

注意：只支持同存储池增量备份卷，暂不支持跨存储池增量备份卷。

参数

参数	描述
<i>backup-name</i>	备份名称。
<i>volume_id</i>	源卷 ID。
<i>volume_name</i>	源卷名称。

示例

为卷进行增量备份，名称为 back-w11。

```
[root@controller cinder(keystone_admin)]# openstack volume backup create --incremental -  
-name back-w11 lun_001  
+-----+-----+  
| Field | Value |  
+-----+-----+  
| id | 597ae1b2-fa19-46a0-beb5-4514af37228a |  
| name | back-w11 |  
| volume_id | da139514-d19f-4567-8070-c04db3a66de9 |  
+-----+-----+
```

为卷进行增量备份，名称为 back-w9。

```
[root@controller cinder(keystone_admin)]# cinder backup-create --incremental da139514-  
d19f-4567-8070-c04db3a66de9 --name back-w9
```

Property	Value
id	59bb78be-0099-4b2c-8955-b4d25af84c89
name	back-w9
volume_id	da139514-d19f-4567-8070-c04db3a66de9

7.4 查看备份列表

`openstack volume backup list`

此命令用来查看备份列表。

示例

查看备份列表。

```
[root@controller cinder(keystone_admin)]# openstack volume backup list

+-----+-----+-----+-----+-----+
| ID      | Name    | Description | Status    | Size | Incremental |
+-----+-----+-----+-----+-----+
| 5f2fdb1a-de0c-4c51-9326-43a4450fc6d9 | back-w6 | None        | available | 1   | True        |
| 6a51f310-9bd2-4c11-b0fe-df9e84bb87ee | back-w5 | None        | available | 1   | True        |
| b43d92d8-d6b5-4178-8c2f-312a25eddfa7 | None    | None        | available | 1   | True        |
| abac89da-7abd-4076-a983-b5b6782c913d | back-w3 | None        | available | 1   | False       |
+-----+-----+-----+-----+-----+
```

备份信息描述

项目	描述
ID	备份 ID。
Name	备份名称。
Description	备份卷的描述信息。 None 表示没有表示信息。
Status	备份卷的状态。
Size	备份卷的大小，以 GiB 为单位。
Incremental	表示该备份是否为增量备份： ● True: 增量备份。 ● False: 全量备份。

7.5 查看备份详情

```
openstack volume backup show backup_id
```

此命令用来查看备份的详细信息。

参数

参数	描述
<i>backup_id</i>	备份 ID。

示例

查看备份 ID 为 5f2fdb1a-de0c-4c51-9326-43a4450fc6d9 的详细信息。

```
[root@controller cinder(keystone_admin)]# openstack volume backup show 5f2fdb1a-de0c-4c51-9326-43a4450fc6d9
```

Field	Value
availability_zone	nova
container	None
created_at	2025-04-24T11:30:18.000000
data_timestamp	2025-04-24T11:30:18.000000
description	None
encryption_key_id	None
fail_reason	None
has_dependent_backups	False
id	5f2fdb1a-de0c-4c51-9326-43a4450fc6d9
is_incremental	True
metadata	{}
name	back-w6
object_count	0
project_id	97be599dfa1f4d2584719a2511b82207
size	1
snapshot_id	None
status	available

updated_at	2025-04-24T11:30:23.000000	
user_id	d60fa5458fda4f96bbf581bf6c2dd63e	
volume_id	6b1649ea-48ec-4392-b4e8-f919cf484f43	
+-----+-----+-----+		

备份信息描述

项目	描述
Field	备份的属性字段名称。
Value	对应字段的具体值。
availability_zone	备份的可用分区。
container	备份存储的容器名称。 None 表示未指定。
created_at	备份的创建时间。
data_timestamp	备份数据的时间戳。
description	备份卷的描述信息。 None 表示没有表示信息。
encryption_key_id	备份的加密密钥 ID。 None 表示未加密。
fail_reason	备份失败的原因。 None 表示备份未失败。
has_dependent_backups	是否还有其他备份依赖于该备份。
id	备份 ID。
is_incremental	该备份是否为增量备份： <ul style="list-style-type: none"> ● True: 增量备份。 ● False: 全量备份。
metadata	备份的元数据。 {} 表示没有元数据。
name	备份名称。

object_count	备份中的对象数量。
project_id	备份所属项目的 ID。
size	备份的大小。
snapshot_id	用于创建备份的快照 ID。 None 表示未使用快照。
status	备份的状态。
updated_at	备份最后更新的时间。
user_id	备份所属用户的 ID。
volume_id	创建备份的源卷 ID。

7.6 删除备份

openstack volume backup delete *backup_id*

此命令用来删除指定的备份。

参数

参数	描述
<i>backup_id</i>	备份 ID。

示例

删除备份。

```
[root@controller cinder(keystone_admin)]# openstack volume backup delete 5f2fdb1a-de0c-4c51-9326-43a4450fc6d9
```

7.7 恢复备份

```
cinder backup-restore backup_id --volume volume_id
```

此命令用通过备份恢复卷。

注意：

- 待恢复卷的大小必须大于等于备份大小。
- 通过备份恢复卷时，该卷不能有关联的快照，否则会恢复失败。

参数

参数	描述
<i>backup_id</i>	备份 ID。
<i>volume_id</i>	恢复卷的 ID。

示例

通过备份恢复卷。

```
[root@controller cinder(keystone_admin)]# cinder backup-restore ad9f5d37-e275-4f12-8ba1-41488eb36218 --volume 6b1649ea-48ec-4392-b4e8-f919cf484f43
+-----+-----+
| Property | Value           |
+-----+-----+
| backup_id | ad9f5d37-e275-4f12-8ba1-41488eb36218 |
| volume_id | 6b1649ea-48ec-4392-b4e8-f919cf484f43 |
| volume_name | lun_001          |
+-----+-----+
```

7.8 修改备份状态

```
openstack volume backup set [--state state] backup_id
```

此命令用来修改备份状态。

注意：

- 修改备份状态时要非常小心，因为错误的状态可能会导致备份数据不一致或无法使用
- 执行此命令通常需要管理员权限。

参数

参数	描述
<i>state</i>	备份的状态。
<i>backup_id</i>	备份 ID。

示例

修改备份状态。

```
[root@controller cinder(keystone_admin)]# openstack volume backup set --state error  
ad9f5d37-e275-4f12-8ba1-41488eb36218
```

8 存储池操作

8.1 查询存储池

OpenStack 命令行：

```
openstack volume backend pool list --long
```

Cinder 命令行：

```
cinder get-pools --detail
```

此命令用来查询存储池信息。

注意：

- 修改/etc/cinder/cinder.conf 配置文件中的参数 local_storage_class，若与现有卷的存储类型属性不匹配，将会导致计算错误，影响查询结果的准确性。
- 多个存储类型共享同一存储池时，可能因存储空间交叉分配而导致查询信息出现偏差。
- 系统计算存储池容量时自动过滤缓存池资源。

示例

查询存储池详细信息。

```
[root@controller ~(keystone_admin)]# cinder get-pools --detail
```

Property	Value
QoS_support	True
allocated_capacity_gb	0
backend_state	unknown
cacheable	True

cluster_license_max_local_capacity_gb	unknown	
driver_version	1.3.0	
fault_domain_count	unknown	
fault_domain_level	unknown	
filter_function	None	
free_capacity_gb	unknown	
goodness_function	None	
location_info	unknown	
max_over_subscription_ratio	20.0	
multiattach	False	
name	controller@stor1#default	
path_capacity_total_gb	unknown	
path_capacity_used_gb	unknown	
path_quota_total_gb	unknown	
path_quota_used_gb	unknown	
pool_name	default	
provisioned_capacity_gb	unknown	
snapshot_used_size_gb	unknown	
stor_id	DDD978E1-14F1-4387-9841-FCC40BE47C66	
storage_protocol	iSCSI	
thin_provisioning_support	True	
timestamp	2025-11-03T03:29:03.320675	
total_capacity_gb	unknown	
vendor_name	CLOUD.STOR	
volume_backend_name	stor1	
+-----+-----+-----+		
+-----+-----+-----+		

Property	Value
QoS_support	True
allocated_capacity_gb	0
backend_state	up
cacheable	True
cluster_license_max_local_capacity_gb	unknown
driver_version	1.3.0
fault_domain_count	3
fault_domain_level	server
filter_function	None
free_capacity_gb	unknown
goodness_function	None
location_info	StorDriver:test:0FBD184B-EBBD-4022-8455-F2559B860F67:cinder:default
max_over_subscription_ratio	20.0
multiattach	False
name	controller@stor8#default
path_capacity_total_gb	736.891972
path_capacity_used_gb	32.060497
path_quota_total_gb	736.891972
path_quota_used_gb	0.000168
pool_name	default
provisioned_capacity_gb	0
snapshot_used_size_gb	0.000000
stor_id	0FBD184B-EBBD-4022-8455-F2559B860F67
storage_protocol	iSCSI
thin_provisioning_support	True

timestamp	2025-11-03T03:29:22.989870
total_capacity_gb	491.26
vendor_name	CLOUD.STOR
volume_backend_name	stor8

查询存储池详细信息。

[root@controller ~]# openstack volume backend pool list -long	
Name	Capabilities
controller@stor1#unknown	QoS_support='True', allocated_capacity_gb='6', backend_state='up', cacheable='True', cluster_license_max_local_capacity_gb='unknown', driver_version='1.3.0', fault_domain_count='unknown', fault_domain_level='unknown', filter_function=, free_capacity_gb='unknown', goodness_function=, location_info='StorDriver:stor1:ABFFE025-3A43-43F6-B061-8DE90288937F:cinder:/mnt/cache1', max_over_subscription_ratio='20.0', multiattach='False', path_capacity_total_gb='416.668736', path_capacity_used_gb='0.061180', path_quota_total_gb='416.668736', path_quota_used_gb='0.000240', pool_name='unknown', provisioned_capacity_gb='6', snapshot_used_size_gb='unknown', stor_id='ABFFE025-3A43-43F6-B061-8DE90288937F', storage_protocol='iSCSI', thin_provisioning_support='True', timestamp='2025-11-03T03:30:32.443281', total_capacity_gb='416.67', vendor_name='CLOUD.STOR', volume_backend_name='stor1'
controller@stor8#default	QoS_support='True', allocated_capacity_gb='0', backend_state='up', cacheable='True', cluster_license_max_local_capacity_gb='unknown', driver_version='1.3.0', fault_domain_count='3', fault_domain_level='server', filter_function=, free_capacity_gb='unknown', goodness_function=, location_info='StorDriver:test:0FBD184B-EBBD-4022-8455-F2559B860F67:cinder:default', max_over_subscription_ratio='20.0', multiattach='False', path_capacity_total_gb='736.891972', path_capacity_used_gb='32.060051', path_quota_total_gb='736.891972', path_quota_used_gb='0.000168', pool_name='default', provisioned_capacity_gb='0', snapshot_used_size_gb='0.000000', stor_id='0FBD184B-EBBD-4022-8455-F2559B860F67', storage_protocol='iSCSI', thin_provisioning_support='True', timestamp='2025-11-03T03:30:22.980963', total_capacity_gb='491.26', vendor_name='CLOUD.STOR', volume_backend_name='stor8'

存储池信息描述

项目	描述
QoS_support	是否支持 QoS: <ul style="list-style-type: none"> ● True: 支持。 ● False: 不支持。
allocated_capacity_gb	返回值为 0。
backend_state	Cinder 后端服务是否可用：

	<ul style="list-style-type: none"> ● up: 服务可用。 ● down: 服务不可用。
cacheable	是否支持数据缓存到本机: <ul style="list-style-type: none"> ● True: 支持。 ● False: 不支持。
cluster_license_max_local_capacity_gb	当前存储池所属集群的有效许可证容量, 单位是 GiB。
driver_version	驱动的版本。
fault_domain_count	故障域数量。 仅集群版支持, 单机版显示为 unknown 。
fault_domain_level	故障域级别。 仅集群版支持, 单机版显示为 unknown 。
filter_function	用户自定义的布尔表达式。
free_capacity_gb	返回值为 unkown 。
goodness_function	用户自定义的评分表达式。
location_info	存储后端的详细信息。
max_over_subscription_ratio	瘦供给 (thin provisioning) 场景下存储资源超售比例。
multiattach	存储池是否支持将单个卷同时挂载到多个虚拟机实例的功能: <ul style="list-style-type: none"> ● True: 支持。 ● False: 不支持。
name	存储池的唯一标识符
path_capacity_total_gb	存储池中所有数据目录所在磁盘的总容量, 单位是 GiB。 仅集群版支持, 单机版显示为 unknown 。
path_capacity_used_gb	存储池中所有数据目录所在磁盘的已用容

	量, 单位是 GiB。 仅集群版支持, 单机版显示为 <code>unknown</code> 。
<code>path_quota_total_gb</code>	存储池中所有数据目录所在磁盘中, 用户给 HBlock 分配的容量配额的总和, 单位是 GiB。 仅集群版支持, 单机版显示为 <code>unknown</code> 。
<code>path_quota_used_gb</code>	存储池中所有数据目录所在磁盘中, HBlock 数据占用的空间大小, 单位是 GiB。 仅集群版支持, 单机版显示为 <code>unknown</code> 。
<code>pool_name</code>	存储池名称。
<code>provisioned_capacity_gb</code>	<ul style="list-style-type: none"> ● 集群版: HBlock 存储池中已经分配的卷的总空间大小, 即已创建的卷容量和, 单位是 GiB。 ● 单机版: 对应 <code>path</code> 中包含的卷的容量总和, 即已创建的卷容量和, 单位是 GiB。
<code>snapshot_used_size_gb</code>	存储池内所有快照大小总和, 即以当前存储池作为最终存储池的所有卷的快照大小的总和。单位是 GiB。 仅集群版支持, 单机版显示为 <code>unknown</code> 。 注意: 卷异常或上游快照删除等因素可能导致快照大小波动。
<code>stor_id</code>	HBlock 集群 ID。
<code>storage_protocol</code>	存储协议。
<code>thin_provisioning_support</code>	否支持 thin provisioning: <ul style="list-style-type: none"> ● <code>True</code>: 支持。

	<ul style="list-style-type: none">● False: 不支持。
timestamp	最后状态更新时间戳。
total_capacity_gb	<ul style="list-style-type: none">● 集群版：存储池总空间可以分配的虚拟卷总容量，单位是 GiB。 说明：如果卷的冗余模式变更，会重新计算该值。● 单机版：path 对应磁盘可以分配的虚拟卷总容量，单位是 GiB。
vendor_name	驱动的供应商。
volume_backend_name	存储后端的逻辑标识符。

9 QoS 策略

9.1 创建 QoS 策略

OpenStack 命令行：

```
openstack volume qos create --consumer back-end qosName
```

Cinder 命令行：

```
cinder qos-create qosName consumer=back-end key1=value1 key2=value2 keyN=valueN
```

此命令用来创建 QoS 策略。

说明：使用 **openstack volume qos create --consumer back-end qosName** 创建 QoS 策略后，可以再使用 **openstack volume qos set --property key1=value1 [--property key2=value2 ...] qosName** 添加 QoS 策略限制规则。

参数

参数	描述
<i>qosName</i>	QoS 策略名称。
<i>consumer</i>	指定消费端位置。 取值：back-end，表示在存储设备侧限速。
<i>Key=value</i>	QoS 限制规则。一次可以写多个 QoS 限制规则。 total_iops_sec 每秒能够进行读写操作次数的最大值。 取值：整型，取值范围为[-1, 999999999]，默认值为-1。-1 表示不限制。 read_iops_sec 每秒能够进行读操作次数的最大值。 取值：整型，取值范围为[-1, 999999999]，默认值为-1。-1 表示不限制。 write_iops_sec 每秒能够进行写操作次数的最大值。 取值：整型，取值范围为[-1, 999999999]，默认值为-1。-1 表示不限制。

	<p>-1。-1 表示不限制。</p>
<code>total_bytes_sec</code>	<p>每秒可传输数据量的最大值。</p> <p>取值：整型，取值范围为[-1, 40960000000000]，默认值为-1，单位是 Bytes/s。-1 表示不限制。</p>
<code>read_bytes_sec</code>	<p>读带宽上限。</p> <p>取值：整型，取值范围为[-1, 40960000000000]，默认值为-1，单位是 Bytes/s。-1 表示不限制。</p>
<code>write_bytes_sec</code>	<p>写带宽上限。</p> <p>取值：整型，取值范围为[-1, 40960000000000]，默认值为-1，单位是 Bytes/s。-1 表示不限制。</p>
<code>total_iops_sec_max</code>	<p>使用 Burst 功能时，每秒能够进行读写操作次数的最大值。</p> <p>取值：只有当 <code>total_iops_sec</code> 大于等于 1 时，此项设置为-1 或(<code>total_iops_sec</code>, 99999999]内的正整数方可生效。默认值为-1，表示不限制。</p>
<code>read_iops_sec_max</code>	<p>使用 Burst 功能时，每秒能够进行读操作次数的最大值。</p> <p>取值：只有当 <code>read_iops_sec</code> 大于等于 1 时，此项设置为-1 或(<code>read_iops_sec</code>, 99999999]内的正整数方可生效。默认值为-1，表示不限制。</p>
<code>write_iops_sec_max</code>	<p>使用 Burst 功能时，每秒能够进行写操作次数的最大值。</p> <p>取值：只有当 <code>write_iops_sec</code> 大于等于 1 时，此项设置为-1 或(<code>write_iops_sec</code>, 99999999]内的正整数方可生效。默认值为-1，表示不限制。</p>
<code>total_bytes_sec_max</code>	<p>使用 Burst 功能时，每秒可传输的数据量最大值。</p> <p>取值：只有当 <code>total_bytes_sec</code> 大于等于 1 时，此</p>

		项设置为-1 或(<i>total_bytes_sec</i> , 4096000000000]内的正整数方可生效， 默认值为-1， 单位是 Bytes/s。 -1 表示不限制。
	<code>read_bytes_sec_max</code>	使用 Burst 功能时， 读带宽上限。 取值： 只有当 <code>read_bytes_sec</code> 大于等于 1 时， 此项设置为-1 或(<i>read_bytes_sec</i> , 4096000000000]内的正整数方可生效， 默认值为-1， 单位是 Bytes/s。 -1 表示不限制。
	<code>write_bytes_sec_max</code>	使用 Burst 功能时， 写带宽上限。 取值： 只有当 <code>write_bytes_sec</code> 大于等于 1 时， 此项设置为-1 或(<i>write_bytes_sec</i> , 4096000000000]内的正整数方可生效， 默认值为-1， 单位是 Bytes/s。 -1 表示不限制。

示例

创建 QoS 策略 `qos-w1`， 并添加限制规则。

```
[root@controller cinder(keystone_admin)]# openstack volume qos create --consumer back-end qos-w1
+-----+-----+
| Field      | Value
+-----+-----+
| consumer   | back-end
| id          | 3af30a66-edb2-4776-8a24-42063e5a7ec7
| name        | qos-w1
| properties  |
+-----+-----+
[root@controller cinder(keystone_admin)]# openstack volume qos set --property
total_iops_sec=90000000000 --property read_iops_sec=6000000000 --property
write_iops_sec=6000000000 qos-w1
```

创建 QoS 策略 qos-w2。

```
[root@controller cinder(keystone_admin)]# cinder qos-create qos-w2 consumer=back-end
total_iops_sec=90000000000 read_iops_sec=60000000000
+-----+-----+
| Property | Value |
+-----+-----+
| consumer | back-end |
| id | e34815a1-81d4-4c1f-bc72-d54be24a3562 |
| name | qos-w2 |
| specs | read_iops_sec : 60000000000 |
| | total_iops_sec : 90000000000 |
+-----+-----+
```

9.2 修改 QoS 策略

OpenStack 命令行：

```
openstack volume qos set --property key1=value1 [--property key2=value2 ...] qos-spec
```

Cinder 命令行：

```
cinder qos-key qos-id set key1=value1 key2=value2 keyN=valueN
```

此命令用来修改 QoS 策略。

说明：卷类型绑定 QoS 策略后，修改 QoS 策略，仅对后续新创建的卷生效，现存的卷不受影响。

参数

参数	描述								
<i>qos-spec</i>	QoS 策略名称或者 QoS 策略 ID。								
<i>qos-id</i>	QoS 策略 ID。								
--property	可以修改的 QoS 限制规则。一次可以写多个 QoS 限制规则。								
<i>key=value</i> 或 <i>Key=value</i>	<table border="1"> <tr> <td>total_iops_sec</td> <td>每秒能够进行读写操作次数的最大值。 取值：整型，取值范围为[-1, 9999999999]，默认值为-1。-1 表示不限制。</td> </tr> <tr> <td>read_iops_sec</td> <td>每秒能够进行读操作次数的最大值。 取值：整型，取值范围为[-1, 9999999999]，默认值为-1。-1 表示不限制。</td> </tr> <tr> <td>write_iops_sec</td> <td>每秒能够进行写操作次数的最大值。 取值：整型，取值范围为[-1, 9999999999]，默认值为-1。-1 表示不限制。</td> </tr> <tr> <td>total_bytes_sec</td> <td>每秒可传输数据量的最大值。 取值：整型，取值范围为[-1, 4096000000000]，默认值为-1，单位是 Bytes/s。-1 表示不限制。</td> </tr> </table>	total_iops_sec	每秒能够进行读写操作次数的最大值。 取值：整型，取值范围为[-1, 9999999999]，默认值为-1。-1 表示不限制。	read_iops_sec	每秒能够进行读操作次数的最大值。 取值：整型，取值范围为[-1, 9999999999]，默认值为-1。-1 表示不限制。	write_iops_sec	每秒能够进行写操作次数的最大值。 取值：整型，取值范围为[-1, 9999999999]，默认值为-1。-1 表示不限制。	total_bytes_sec	每秒可传输数据量的最大值。 取值：整型，取值范围为[-1, 4096000000000]，默认值为-1，单位是 Bytes/s。-1 表示不限制。
total_iops_sec	每秒能够进行读写操作次数的最大值。 取值：整型，取值范围为[-1, 9999999999]，默认值为-1。-1 表示不限制。								
read_iops_sec	每秒能够进行读操作次数的最大值。 取值：整型，取值范围为[-1, 9999999999]，默认值为-1。-1 表示不限制。								
write_iops_sec	每秒能够进行写操作次数的最大值。 取值：整型，取值范围为[-1, 9999999999]，默认值为-1。-1 表示不限制。								
total_bytes_sec	每秒可传输数据量的最大值。 取值：整型，取值范围为[-1, 4096000000000]，默认值为-1，单位是 Bytes/s。-1 表示不限制。								

	<code>read_bytes_sec</code>	读带宽上限。 取值：整型，取值范围为[-1, 40960000000000]，默认值为-1，单位是 Bytes/s。-1 表示不限制。
	<code>write_bytes_sec</code>	写带宽上限。 取值：整型，取值范围为[-1, 40960000000000]，默认值为-1，单位是 Bytes/s。-1 表示不限制。
	<code>total_iops_sec_max</code>	使用 Burst 功能时，每秒能够进行读写操作次数的最大值。 取值：只有当 <code>total_iops_sec</code> 大于等于 1 时，此项设置为-1 或(<code>total_iops_sec</code> , 99999999]内的正整数方可生效。默认值为-1，表示不限制。
	<code>read_iops_sec_max</code>	使用 Burst 功能时，每秒能够进行读操作次数的最大值。 取值：只有当 <code>read_iops_sec</code> 大于等于 1 时，此项设置为-1 或(<code>read_iops_sec</code> , 99999999]内的正整数方可生效。默认值为-1，表示不限制。
	<code>write_iops_sec_max</code>	使用 Burst 功能时，每秒能够进行写操作次数的最大值。 取值：只有当 <code>write_iops_sec</code> 大于等于 1 时，此项设置为-1 或(<code>write_iops_sec</code> , 99999999]内的正整数方可生效。默认值为-1，表示不限制。
	<code>total_bytes_sec_max</code>	使用 Burst 功能时，每秒可传输的数据量最大值。 取值：只有当 <code>total_bytes_sec</code> 大于等于 1 时，此项设置为-1 或(<code>total_bytes_sec</code> , 40960000000000]内的正整数方可生效，默认值为-1，单位是 Bytes/s。-1 表示不限制。
	<code>read_bytes_sec_max</code>	使用 Burst 功能时，读带宽上限。

		取值：只有当 <code>read_bytes_sec</code> 大于等于 1 时，此项设置为-1 或(<code>read_bytes_sec</code> , 4096000000000]内的正整数方可生效，默认值为-1，单位是 Bytes/s。-1 表示不限制。
	<code>write_bytes_sec_max</code>	使用 Burst 功能时，写带宽上限。 取值：只有当 <code>write_bytes_sec</code> 大于等于 1 时，此项设置为-1 或(<code>write_bytes_sec</code> , 4096000000000]内的正整数方可生效，默认值为-1，单位是 Bytes/s。-1 表示不限制。

示例

修改 QoS 策略，QoS 策略 qos-w1。

```
[root@controller cinder(keystone_admin)]# openstack volume qos set --property
total_iops_sec_max=990000000 --property write_iops_sec_max=880000000 qos-w1
```

修改 QoS 策略，QoS 策略 ID 为 e34815a1-81d4-4c1f-bc72-d54be24a3562。

```
[root@controller cinder(keystone_admin)]# cinder qos-key e34815a1-81d4-4c1f-bc72-
d54be24a3562 set total_iops_sec=8000000000 read_iops_sec=7000000000
total_iops_sec_max=90000000100
```

9.3 取消 QoS 策略的规则限制

```
openstack volume qos set --no-property qos-spec
```

此命令用来取消指定 QoS 策略的所有规则限制。

```
openstack volume qos unset --property key [--property key ...] qos-spec
```

此命令用来取消指定 QoS 策略的指定规则限制。

参数

参数	描述
--no-property	删除 QoS 策略的所有限速。
qos-spec	QoS 策略名称或者 QoS 策略 ID。
--property	QoS 限制规则。限制规则如下:
key	
total_iops_sec	每秒能够进行读写操作次数的最大值。
read_iops_sec	每秒能够进行读操作次数的最大值。
write_iops_sec	每秒能够进行写操作次数的最大值。
total_bytes_sec	每秒可传输数据量的最大值。
read_bytes_sec	读带宽上限。
write_bytes_sec	写带宽上限。
total_iops_sec_max	使用 Burst 功能时, 每秒能够进行读写操作次数的最大值。
read_iops_sec_max	使用 Burst 功能时, 每秒能够进行读操作次数的最大值。
write_iops_sec_max	使用 Burst 功能时, 每秒能够进行写操作次数的最大值。
total_bytes_sec_max	使用 Burst 功能时, 每秒可传输的数据量最大值。
read_bytes_sec_max	使用 Burst 功能时, 读带宽上限。
write_bytes_sec_max	使用 Burst 功能时, 写带宽上限。

示例

取消 QoS 策略 qos-w1 的限制规则 total_iops_sec_max、write_iops_sec_max。

```
[root@controller cinder(keystone_admin)]# openstack volume qos unset --property  
total_iops_sec_max --property write_iops_sec_max qos-w1
```

取消 QoS 策略 qos-w1 的所有的限制规则。

```
[root@controller cinder(keystone_admin)]# openstack volume qos set --no-property qos-w1
```

9.4 将 QoS 策略绑定到卷类型

OpenStack 命令行：

```
openstack volume qos associate qos-spec volume-type
```

Cinder 命令行：

```
cinder qos-associate qos-id volume-type-id
```

此命令用来将 QoS 策略绑定到卷类型。

说明：卷类型绑定 QoS 策略后，仅对后续新创建的卷生效，现存的卷不受影响。

参数

参数	描述
<i>qos-spec</i>	QoS 策略名称或者 QoS 策略 ID。
<i>volume-type</i>	卷类型名称或者卷类型 ID。
<i>qos-id</i>	QoS 策略 ID。
<i>volume-type-id</i>	卷类型 ID。

示例

将 QoS 策略 qos-w1 绑定到卷类型 stor3。

```
[root@controller cinder(keystone_admin)]# openstack volume qos associate qos-w1 stor3
```

将 QoS 策略 3af30a66-edb2-4776-8a24-42063e5a7ec7 绑定到卷类型 ed4465da-0046-4f89-9154-562e2d452043。

```
[root@controller cinder(keystone_admin)]# cinder qos-associate 3af30a66-edb2-4776-8a24-42063e5a7ec7 ed4465da-0046-4f89-9154-562e2d452043
```

9.5 解除卷类型的 QoS 策略

OpenStack 命令行：

```
openstack volume qos disassociate --volume-type volume-type qos-spec
```

Cinder 命令行：

```
cinder qos-disassociate qos-id volume-type-id
```

此命令用来解除卷类型的 QoS 策略。

说明：解除卷类型的 QoS 策略，仅对后续新建卷的 QoS 策略有影响，现存卷不受影响。

参数

参数	描述
<i>qos-spec</i>	QoS 策略名称或者 QoS 策略 ID。
<i>volume-type</i>	卷类型名称或者卷类型 ID。
<i>qos-id</i>	QoS 策略 ID。
<i>volume-type-id</i>	卷类型 ID。

示例

解除卷类型 stor3 的 QoS 策略。

```
[root@controller cinder(keystone_admin)]# openstack volume qos disassociate --volume-type stor3 qos-w1
```

解除卷类型 ed4465da-0046-4f89-9154-562e2d452043 的 QoS 策略。

```
[root@controller cinder(keystone_admin)]# cinder qos-disassociate 3af30a66-edb2-4776-8a24-42063e5a7ec7 ed4465da-0046-4f89-9154-562e2d452043
```

9.6 删除 QoS 策略

OpenStack 命令行：

```
openstack volume qos delete [ --force ] qos-spec
```

Cinder 命令行：

```
cinder qos-delete qos-spec
```

说明：如果删除的 QoS 策略已经被卷类型绑定，删除该 QoS 策略，仅对后续新创建的卷生效，现存的卷的 QoS 策略不受影响。

参数

参数	描述
<i>qos-spec</i>	QoS 策略名称或者 QoS 策略 ID。

示例

删除 QoS 策略 qos-w1。

```
[root@controller cinder(keystone_admin)]# openstack volume qos delete qos-w1
```

删除 QoS 策略 qos-w2。

```
[root@controller cinder(keystone_admin)]# cinder qos-delete qos-w2
```

9.7 查看卷类型关联的 QoS 策略

OpenStack 命令行：

```
openstack volume type show volume-type -c qos_specs_id
```

Cinder 命令行：

```
cinder type-show volume-type
```

此命令用来查看卷类型关联的 QoS 策略。

参数

参数	描述
<i>volume-type</i>	卷类型名称或者卷类型 ID。

示例

查看卷类型 stor2 关联的 QoS 策略。

```
[root@controller cinder(keystone_admin)]# openstack volume type show stor2 -c qos_specs_id
+-----+-----+
| Field      | Value          |
+-----+-----+
| qos_specs_id | 960aa9ef-9596-45da-9799-5ab6bac8b3ef |
+-----+-----+
```

查看卷类型 stor2 关联的 QoS 策略。

```
[root@controller cinder(keystone_admin)]# cinder type-show stor2
+-----+-----+
| Property          | Value          |
+-----+-----+
| description       | None          |
| extra_specs       | volume_backend_name : stor2 |
| id                | d91a3e01-5bad-484b-b661-ad7e242b4080 |
| is_public         | True          |
| name              | stor2         |
+-----+-----+
```

os-volume-type-access:is_public True	
qos_specs_id	960aa9ef-9596-45da-9799-5ab6bac8b3ef
+-----+-----+	

卷类型关联的 QoS 策略描述

项目	描述
<code>description</code>	卷类型的描述。
<code>extra_specs</code>	键值型“调度元数据”。
<code>id</code>	卷类型 ID。
<code>is_public</code>	卷类型是否对所有项目适用： ● <code>True</code> : 对所有项目可用。 ● <code>False</code> : 仅指定项目可使用。
<code>name</code>	卷类型名称。
<code>os-volume-type-access:is_public</code>	与 <code>is_public</code> 相同。
<code>qos_specs_id</code>	QoS 策略 ID。

9.8 查看 QoS 策略列表

OpenStack 命令行：

```
openstack volume qos list
```

Cinder 命令行：

```
cinder qos-list
```

此命令用来查看 QoS 策略列表。

示例

查看 QoS 策略列表。

[root@controller cinder(keystone_admin)]# openstack volume qos list				
ID	Name	Consumer	Associations	Properties
a628a018-078c-4df5-82c1-e4e3cd0f324e	qos-w1	back-end		read_iops_sec='6000000000', total_iops_sec='90000000000', write_iops_sec='6000000000'
fe535284-fcd8-43aa-ae9c-df230625ad22	qos-w2	back-end		read_iops_sec='6000000000', total_iops_sec='90000000000'
fbc45ded-ed9e-4d0c-a5a2-927b97af385	qos1	back-end	stor8	read_bytes_sec='10485760', read_bytes_sec_max='15240000', read_bytes_sec_time='10485760', read_iops_sec='80', read_iops_sec_max='110', test='1', total_bytes_sec='10M', total_bytes_sec_max='20M', total_iops_sec='100', total_iops_sec_max='120', write_bytes_sec='8M', write_bytes_sec_max='16M', write_iops_sec='90', write_iops_sec_max='120'

查看 QoS 策略列表。

[root@controller cinder(keystone_admin)]# cinder qos-list				

ID	Name	Consumer	specs
+-----+-----+-----+			
+-----+-----+-----+			
a628a018-078c-4df5-82c1-e4e3cd0f324e qos-w1 back-end {'write_iops_sec': '6000000000', 'read_iops_sec': '6000000000', 'total_iops_sec': '90000000000'}			
fbc45ded-ed9e-4d0c-a5a2-927b97af385 qos1 back-end {'read_iops_sec': '80', 'total_bytes_sec_max': '20M', 'write_bytes_sec_max': '16M', 'test': '1', 'read_iops_sec_max': '110', 'total_bytes_sec': '10M', 'read_bytes_sec_time': '10485760', 'total_iops_sec': '100', 'read_bytes_sec_max': '15240000', 'total_iops_sec_max': '120', 'write_bytes_sec': '8M', 'write_iops_sec_max': '120', 'write_iops_sec': '90', 'read_bytes_sec': '10485760'}			
fe535284-fcd8-43aa-ae9c-df230625ad22 qos-w2 back-end {'read_iops_sec': '6000000000', 'total_iops_sec': '90000000000'}			
+-----+-----+-----+			
+-----+-----+-----+			
+-----+-----+-----+			

QoS 策略列表信息描述

项目	描述	
Associations	关联的卷类型。	
id	QoS 策略 ID。	
name	QoS 策略名称。	
specs	QoS 策略规则限制。	
	total_iops_sec	每秒能够进行读写操作次数的最大值。 -1 表示不限制。
	read_iops_sec	每秒能够进行读操作次数的最大值。

		-1 表示不限制。
	<code>write_iops_sec</code>	每秒能够进行写操作次数的最大值。 -1 表示不限制。
	<code>total_bytes_sec</code>	每秒可传输数据量的最大值, 单位是 Bytes/s。 -1 表示不限制。
	<code>read_bytes_sec</code>	读带宽上限, 单位是 Bytes/s。 -1 表示不限制。
	<code>write_bytes_sec</code>	写带宽上限, 单位是 Bytes/s。 -1 表示不限制。
	<code>total_iops_sec_max</code>	使用 Burst 功能时, 每秒能够进行读写操作次数的最大值。 -1 表示不限制。
	<code>read_iops_sec_max</code>	使用 Burst 功能时, 每秒能够进行读操作次数的最大值。 -1 表示不限制。
	<code>write_iops_sec_max</code>	使用 Burst 功能时, 每秒能够进行写操作次数的最大值。 -1 表示不限制。
	<code>total_bytes_sec_max</code>	使用 Burst 功能时, 每秒可传输的数据量最大值, 单位是 Bytes/s。 -1 表示不限制。
	<code>read_bytes_sec_max</code>	使用 Burst 功能时, 读带宽上限, 单位是 Bytes/s。 -1 表示不限制。
	<code>write_bytes_sec_max</code>	使用 Burst 功能时, 写带宽上限, 单位是 Bytes/s。 -1 表示不限制。

9.9 查看 QoS 策略详细信息

OpenStack 命令行：

```
openstack volume qos show qos-spec
```

Cinder 命令行：

```
cinder qos-show qos-spec
```

此命令用来查看指定 QoS 策略的详细信息。

参数

参数	描述
<i>qos-spec</i>	QoS 策略名称或者 QoS 策略 ID。
<i>qos-id</i>	QoS 策略 ID。

示例

查看 QoS 策略 qos1。

```
[root@controller cinder(keystone_admin)]# openstack volume qos show qos-w1
+-----+-----+
| Field      | Value
+-----+-----+
| associations | stor3
| consumer     | back-end
| id           | a628a018-078c-4df5-82c1-e4e3cd0f324e
| name         | qos-w1
| properties   | read_iops_sec='6000000000', total_iops_sec='90000000000', write_iops_sec='6000000000' |
+-----+-----+
```

查看 QoS 策略 qos-w2。

```
[root@controller cinder(keystone_admin)]# cinder qos-show qos-w2
+-----+-----+
| Property | Value
+-----+-----+
```

consumer	back-end	
id	e34815a1-81d4-4c1f-bc72-d54be24a3562	
name	qos-w2	
specs	read_iops_sec : 7000000000	
	total_iops_sec : 80000000000	
	total_iops_sec_max : 90000000100	

查看 QoS 策略, ID 为 e34815a1-81d4-4c1f-bc72-d54be24a3562。

```
[root@controller cinder(keystone_admin)]# cinder qos-show e34815a1-81d4-4c1f-bc72-d54be24a3562
```

Property	Value
consumer	back-end
id	e34815a1-81d4-4c1f-bc72-d54be24a3562
name	qos-w2
specs	read_iops_sec : 7000000000
	total_iops_sec : 80000000000
	total_iops_sec_max : 90000000100

QoS 策略详细信息描述

项目	描述
Associations	关联的卷类型。
consumer	消费端位置。
id	QoS 策略 ID。
name	QoS 策略名称。
properties 或 specs	QoS 策略规则限制。 total_iops_sec 每秒能够进行读写操作次数的最大值。 -1 表示不限制。 read_iops_sec 每秒能够进行读操作次数的最大值。

	-1 表示不限制。
<code>write_iops_sec</code>	每秒能够进行写操作次数的最大值。 -1 表示不限制。
<code>total_bytes_sec</code>	每秒可传输数据量的最大值, 单位是 Bytes/s。 -1 表示不限制。
<code>read_bytes_sec</code>	读带宽上限, 单位是 Bytes/s。 -1 表示不限制。
<code>write_bytes_sec</code>	写带宽上限, 单位是 Bytes/s。 -1 表示不限制。
<code>total_iops_sec_max</code>	使用 Burst 功能时, 每秒能够进行读写操作次数的最大值。 -1 表示不限制。
<code>read_iops_sec_max</code>	使用 Burst 功能时, 每秒能够进行读操作次数的最大值。 -1 表示不限制。
<code>write_iops_sec_max</code>	使用 Burst 功能时, 每秒能够进行写操作次数的最大值。 -1 表示不限制。
<code>total_bytes_sec_max</code>	使用 Burst 功能时, 每秒可传输的数据量最大值, 单位是 Bytes/s。 -1 表示不限制。
<code>read_bytes_sec_max</code>	使用 Burst 功能时, 读带宽上限, 单位是 Bytes/s。 -1 表示不限制。
<code>write_bytes_sec_max</code>	使用 Burst 功能时, 写带宽上限, 单位是 Bytes/s。 -1 表示不限制。

10 Nova 使用 HBlock 卷

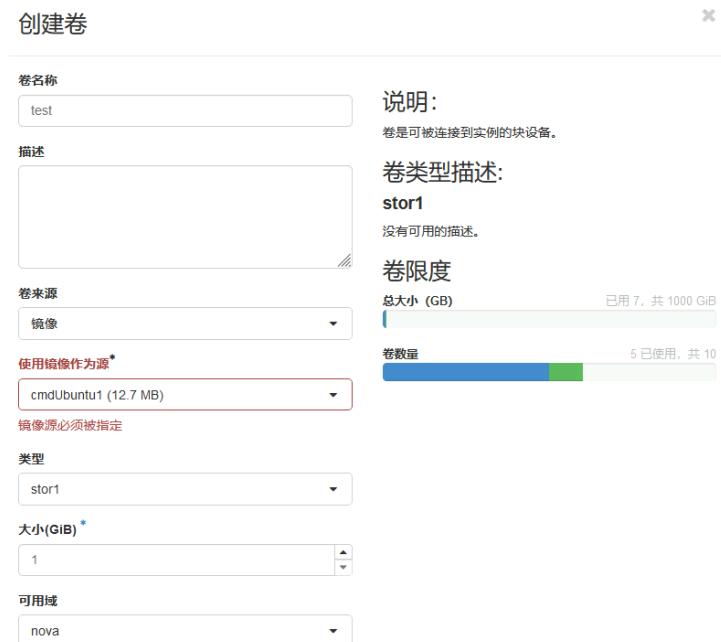
10.1 前置条件

- 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[环境要求](#)。
- 正确安装 OpenStack 的接入插件，且功能正常。

10.2 使用方法

以镜像为源创建的 HBlock 卷 test 为例。

创建卷



卷名称: test

说明: 卷是可被连接到实例的块设备。

描述:

卷类型描述: stor1

卷限度

总大小 (GB): 已用 7, 共 1000 GiB

卷数量: 5 已使用, 共 10

卷来源: 镜像

使用镜像作为源: cmdUbuntu1 (12.7 MB)

镜像源必须被指定

类型: stor1

大小(GiB): 1

可用域: nova

- 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			

11 Glance 使用 HBlock 卷

11.1 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[环境要求](#)。
- 正确安装 OpenStack 的接入插件，且功能正常。

说明：

在使用 Stor Cinder 进行镜像创建卷的场景下，它支持采用克隆方式进行底层处理。当镜像文件较大时，利用克隆方式可以快速高效地完成卷的创建任务，大大提升了操作效率和用户体验。使用此方式的前提条件是：

- Glance 配置了 HBlock 存储。
- cinder.conf 中配置： allowed_direct_url_schemes=cinder，配置位置在[DEFAULT]标签下，否则不生效。
- glance-api.conf 中配置： glance-api.conf: show_image_direct_url = True，配置位置在[DEFAULT]标签下，否则不生效。
- 对应的 image 必须是基于下列条件创建的： HBlock 的卷，格式为 raw，容器为 bare。

11.2 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]
show_image_direct_url = True
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://10.179.32.198:5000/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://10.179.32.198:5000/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
show_image_direct_url	是否允许通过直接 url 的方式表示镜像。 取值： <ul style="list-style-type: none">● True。● False。 说明：在使用 Stor Cinder 进行镜像创建卷的场景下，支持采用克隆方式进行底层处理。如果使用克隆进行底层处理，必须配

	置此项，且该项取值为 True。
enabled_backends	<p>存储标识符和存储类型。配置形式为 <i>key:value</i>。 <i>value</i> 的合法值为 file、rbd、http、swift、cinder。 <i>key</i> 为 default_backend 的值。</p> <p>说明：可以一次配置多个 <i>key:value</i>，以英文逗号隔开，如 <i>key1:value1,key1:value2</i>。enabled_backends 中的每一个 <i>key</i> 都需要有一个单独的一个配置组[<i>key</i>]，配置组中需要配置该配置的所有详细配置项。</p>
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，此值为类似地址： http://10.179.32.198:5000/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	<p>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。</p> <p>取值：</p> <ul style="list-style-type: none"> ● True。 ● False。 <p>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。</p>
cinder_use_multipath	<p>在部署中支持用于识别 Mutipath 的标记。</p> <p>如果不支持多路径，则将其设置为 False。</p>

	<ul style="list-style-type: none"> ● True: Cinder 使用多路径。 ● False: Cinder 不使用多路径。 <p>说明: 如果 HBlock 集群版, 需要配置为 True, 如果 HBlock 单机版, 配置为 False。</p>
rootwrap_config	<p>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。</p>

注意: 以下仅为示例, 请按照实际的 OpenStack 环境信息进行填写。

```

[DEFAULT]
show_image_direct_url = True
enabled_backends="file:file,cinder:cinder,cinder-stor8: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://10.179.32.198:5000/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

[cinder-stor8]
cinder_os_region_name=RegionOne
cinder_store_auth_address=http://10.179.32.198:5000/v3
cinder_store_user_name=admin
cinder_store_password=0ac3ffd8157c4692
cinder_store_project_name=admin
cinder_volume_type=stor8
cinder_enforce_multipath=False
cinder_use_multipath=False
rootwrap_config = /etc/cinder/rootwrap.conf
  
```

2. 命令行创建 images。

```

[root@controller gw(keystone_admin)]# glance image-create --name centos-instance-test1 --visibility public --
disk-format raw --container-format bare --file /home/gw/output.raw --store cinder-stor8
+-----+
| Property      | Value
+-----+
| checksum      | 56730d3091a764d5f8b38feef0bfcef
| container_format | bare
| created_at    | 2025-05-09T06:15:25Z
| direct_url    | cinder://cinder-stor8/313eaedf-3e11-4da3-b5ca-0723fc356c14
| disk_format    | raw
| id            | 12072d0b-b47d-435b-a5c4-cc77f9ed9742
| min_disk       | 0
| min_ram        | 0
| name          | centos-instance-test1
| os_hash_algo   | sha512
| os_hash_value  | 34f5709bc2363eafe857ba1344122594a90a9b8cc9d80047c35f7e34e8ac28ef1e14e2e3c13d55a4
|                  | 3b841f533435e914b01594f2c14dd597ff9949c8389e3006
| os_hidden      | False
| owner          | 97be599dfa1f4d2584719a2511b82207
| protected      | False
| size           | 41126400
| status          | active
| stores          | cinder-stor8
| tags            | []
| updated_at     | 2025-05-09T06:15:30Z
  
```

virtual_size	41126400
visibility	public
+-----+-----+	
[root@controller gw(keystone_admin)]# glance image-create --name centos-instance-test2 --visibility public --disk-format qcow2 --container-format bare --file /home/gw/cirros-0.3.4-x86_64-disk.img --store cinder-stor8	
+-----+-----+	
Property	Value
+-----+-----+	
checksum	ee1eca47dc88f4879d8a229cc70a07c6
container_format	bare
created_at	2025-05-09T08:02:01Z
direct_url	cinder://cinder-stor8/4c3d3910-4fbf-4a01-b434-2ce65263f02c
disk_format	qcow2
id	c1995b47-d3a8-4d34-9bd0-d904388d88ae
min_disk	0
min_ram	0
name	centos-instance-test2
os_hash_algo	sha512
os_hash_value	1b03ca1bc3fafe448b90583c12f367949f8b0e665685979d95b004e48574b953316799e23240f4f7 39d1b5eb4c4ca24d38fdc6f4f9d8247a2bc64db25d6bbdb2
os_hidden	False
owner	97be599dfa1f4d2584719a2511b82207
protected	False
size	13287936
status	active
stores	cinder-stor8
tags	[]
updated_at	2025-05-09T08:02:04Z
virtual_size	41126400
visibility	public
+-----+-----+	

3. 检查创建的 images。

[root@controller gw(keystone_admin)]# glance image-list	
+-----+-----+	
ID	Name
+-----+-----+	
12072d0b-b47d-435b-a5c4-cc77f9ed9742	centos-instance-test1
c1995b47-d3a8-4d34-9bd0-d904388d88ae	centos-instance-test2
+-----+-----+	

4. 创建卷。

```
[root@controller gw(keystone_admin)]# cinder create --name stor8-test3 --image-id 12072d0b-b47d-435b-a5c4-cc77f9ed9742 --volume-type stor8 100
+-----+-----+
| Property          | Value
+-----+-----+
| attachments       | []
| availability_zone | nova
| bootable          | false
| cluster_name      | None
| consistencygroup_id | None
| consumes_quota    | True
| created_at        | 2025-05-09T08:48:00.000000
| description        | None
| encrypted          | False
| group_id          | None
| id                | d4f7fa8a-4090-4b34-8084-cde24755a56b
| metadata          | {}
| migration_status  | None
| multiattach       | False
| name              | stor8-test3
| os-vol-host-attr:host | None
| os-vol-mig-status-attr:migstat | None
| os-vol-mig-status-attr:name_id | None
| os-vol-tenant-attr:tenant_id | 97be599dfa1f4d2584719a2511b82207
| provider_id       | None
| replication_status | None
| service_uuid      | None
| shared_targets    | True
| size              | 100
| snapshot_id       | None
| source_volid      | None
| status             | creating
| updated_at        | None
| user_id           | d60fa5458fda4f96bbf581bf6c2dd63e
| volume_type        | stor8
| volume_type_id    | 997c4eb1-4473-4c0a-a1f6-4f4967fd2c3a
+-----+-----+
```

11.3 使用 HBlock 卷创建 Glance 镜像

OpenStack 命令行：

```
openstack image create image-name --volume volume-id
```

Cinder 命令行：

```
cinder upload-to-image volume-id image-name --disk-format format
```

此命令用来使用 HBlock 卷创建 Glance 镜像。

参数

参数	描述
<i>image-name</i>	镜像名称。
<i>volume-id</i>	HBlock 卷 ID。
<i>format</i>	镜像的磁盘格式。

示例

创建镜像 image-w2。

```
[root@controller cinder(keystone_admin)]# openstack image create image-w2 --volume fa1f035e-0eec-497d-9ada-7b19a9403426
+-----+-----+
| Field          | Value          |
+-----+-----+
| container_format | bare          |
| disk_format     | raw           |
| display_description |          |
| id              | fa1f035e-0eec-497d-9ada-7b19a9403426 |
| image_id        | 5178e77c-d1f2-4a62-9c4f-1a0e7f6b0b8b |
| image_name      | image-w2      |
| protected       | False         |
| size            | 200          |
| status          | uploading    |
| updated_at      | 2025-10-13T09:06:15.000000 |
```

visibility	shared	
volume_type	stor1	

创建镜像 image-w1。

```
[root@controller cinder(keystone_admin)]# cinder upload-to-image 914ad72c-5f77-4af2-b820-b830950583b4 image-w1 --disk-format qcow2
```

Property	Value
container_format	bare
disk_format	qcow2
display_description	
id	914ad72c-5f77-4af2-b820-b830950583b4
image_id	74aebbab-cdde-4d9d-a116-373c6f891996
image_name	image-w1
protected	False
size	100
status	uploading
updated_at	2025-10-13T08:46:55.000000
visibility	private
volume_type	stor1

12 常见问题

12.1 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
```

12.2 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
```

12.3 设置 QoS 规格，消费者为前端或两者时，同时设置 total_bytes_sec 与 read_bytes_sec/write_bytes_sec 报错

设置 QoS 规格，消费者选择前端 (front-end) 或两者 (both) 时，如果同时设置 total_bytes_sec 与 read_bytes_sec/write_bytes_sec，由于 openstack 本身的限制，会出现报错。创建 QoS 规格时，您必须选择所需的消费者。消费者决定您要应用 QoS 限制的位置，并确定哪些 QoS 属性键可用于定义 QoS 限值。cinder 支持以下 QoS 规范使用者：

- 前端：当卷附加到实例时，计算服务(nova)将应用 QoS 限制。计算服务支持块存储服务提供的所有 QoS 属性键。

- 后端：关联卷类型的后端驱动程序应用 QoS 限制。每个后端驱动程序都支持自己的一组 QoS 属性键。
- 两者：使用者尽可能应用 QoS 限制。因此，这个消费者类型支持以下 QoS 属性键：
 - 当卷附加到实例时，您可以使用计算服务和后端驱动程序都支持的每个 QoS 属性键。
 - 当卷未附加到实例时，您只能使用后端驱动程序支持的 QoS 属性键。

所以，如果选择前端（front-end）或两者（both）时，设置 QoS 相关规则报错，请查看 openstack 官网关于 QoS 规格的限制。