



存储资源盘活系统

用户手册 - 命令行

天翼云科技有限公司

修订记录

版本	发布日期	描述
3.9	2025 年 04 月 21 日	1. 支持快照功能，实现数据的快速备份与恢复。 2. 支持克隆卷功能，用于数据复制、测试验证等场景。
3.8	2025 年 02 月 14 日	1. 存储卷和缓存卷支持将数据上传至兼容 S3 的对象存储。 2. 支持设置基础服务的数据存储目录。 3. Target 增加回收策略，支持无卷关联后自动删除。 4. 针对智算、虚拟化以及高可用敏感度等场景，支持一键调整系统参数。 5. 支持设置折叠副本数，允许数据副本/分片放在同一个故障域中。
3.7	2024 年 8 月 08 日	1. 支持设置集群拓扑。 2. 支持创建和管理多存储池。 3. 支持机房和机架级别故障域。 4. 支持设置卷的高速缓存池。 5. 支持基础服务迁移。
3.6	2024 年 06 月 03 日	1. 支持存储卷和缓存卷，将数据从后端上传到 OOS。 2. 硬件及 HBlock 监控数据支持对接到 Prometheus。 3. 支持 HBlock 告警信息对接到智能运维平台。 4. 优化读写性能。 5. 增加对龙芯服务器的支持。
3.5	2024 年 03 月 04 日	1. 支持服务器、数据目录级别的故障域，支持磁盘级别的数据服务。 2. 支持指定基础服务的安装节点。 3. 支持数据目录配额，设置 HBlock 可写入的数据量上限。 4. Target 可被多个客户端发现并连接。 5. 设置卷的最小写入副本数，提高数据写入安全性。 6. 扩大纠删码 EC N+M 支持范围，满足 $N+M \leq 128$。
3.4	2023 年 07 月 12 日	1. 卷连接支持一主多备，提高业务可用性。

		2. 支持 IPv6 环境。 3. 控制台提供 Dashboard 一页式概览。 4. 支持通过命令行查询 CHAP 密码。
3.3	2022 年 12 月 23 日	支持安全移除服务器。
3.2	2022 年 09 月 26 日	1. “产品概述”章节新增“基本概念”。 2. 监控项增加，扩大覆盖范围。 3. 增加对告警、日志管理的支持。 4. 事件中增加对系统事件的支持。 5. “附录”章节新增：系统事件列表、HBlock 进程、监控指标、告警列表。
3.1	2022 年 06 月 14 日	1. 单机版支持添加多个数据目录。 2. 集群版支持创建 Target 时指定对应的服务器，支持 Target 迁移。 3. 支持用户事件的记录和查询。
3.0	2022 年 01 月 18 日	1. 命令行变更为非交互式 2. 支持 WEB、API 调用方式。 3. 卷操作：支持设置卷的高可用类型和卷的写策略。
2.1	2021 年 08 月 27 日	1. 增加对 ARM 服务器的支持。 2. 软件许可证：查看许可证时，可以显示允许的容量。 3. 卷操作：支持对卷进行主备切换，即卷对应的 ACTIVE Target 和 STANDBY Target 切换。
2.0	2021 年 05 月 28 日	1. 支持集群版部署。 2. 支持多副本和纠删码数据冗余。

目 录

1 产品概述	1
1.1 产品定义	1
1.2 应用场景	4
1.3 基本概念	6
1.3.1 iSCSI	6
1.3.2 卷	6
1.3.3 快照	6
1.3.4 克隆卷	7
1.3.5 iSCSI Target	7
1.3.6 iSCSI 目标门户	7
1.3.7 存储池	7
1.3.8 集群拓扑	8
1.3.9 故障域	8
1.3.10 数据服务	8
1.3.11 监控	8
1.3.12 事件	8
1.3.13 日志采集	9
1.3.14 告警	9
1.4 使用限制	10
1.5 术语与缩略语	11
2 服务器端部署	13
2.1 环境要求	13
2.2 HBlock 部署 - 单机版	15
2.2.1 配置环境	15
2.2.2 部署 HBlock	18
2.2.3 部署 HBlock 示例	21

2.3 HBlock 部署 - 集群版	26
2.3.1 配置环境	26
2.3.2 部署 HBlock	29
2.3.3 部署 HBlock 示例	33
3 客户端操作	40
3.1 Windows 客户端 - 单机版	40
3.2 Windows 客户端 - 集群版	48
3.3 Linux 客户端 - 单机版	53
3.3.1 客户端配置	53
3.3.2 配置举例	56
3.4 Linux 客户端 - 集群版	62
3.4.1 客户端配置	62
3.4.2 配置举例	67
4 管理操作	77
4.1 管理操作命令行格式说明	77
4.2 帮助命令	78
4.3 安装	82
4.4 初始化	84
4.5 软件许可证	94
4.5.1 加载软件许可证	94
4.5.2 查看软件许可证	96
4.6 卷操作	100
4.6.1 创建卷	100
4.6.2 创建克隆卷	110
4.6.3 断开克隆卷与快照的关系链	118
4.6.4 删除卷	119
4.6.5 扩容卷	121

4.6.6 修改卷	127
4.6.7 设置卷主备优先级或自动切换（集群版适用）	131
4.6.8 触发卷对应 Target 的主备切换（集群版适用）	133
4.6.9 还原卷（上云卷适用）	134
4.6.10 恢复还原中断的卷（上云卷适用）	142
4.6.11 查询卷信息	143
4.7 快照操作	163
4.7.1 创建快照	163
4.7.2 修改快照	165
4.7.3 回滚快照	167
4.7.4 删除快照	169
4.7.5 查询快照	170
4.8 一致性快照操作	174
4.8.1 创建一致性快照	174
4.8.2 修改一致性快照	176
4.8.3 回滚一致性快照	178
4.8.4 删除一致性快照	180
4.8.5 查询一致性快照	181
4.9 iSCSI Target 操作	184
4.9.1 创建 iSCSI Target	184
4.9.2 删除 iSCSI Target	187
4.9.3 设置 iSCSI Target 的 CHAP 认证	189
4.9.4 迁移 iSCSI Target（集群版适用）	190
4.9.5 修改 iSCSI Target 下每个 IQN 允许建立的最大会话数	192
4.9.6 修改 iSCSI Target 的回收策略	193
4.9.7 查询 iSCSI Target	194
4.10 存储池操作（集群版适用）	199

4. 10. 1 创建存储池	199
4. 10. 2 添加节点到存储池	201
4. 10. 3 修改存储池	203
4. 10. 4 移除存储池内的节点	204
4. 10. 5 删除非基础存储池	205
4. 10. 6 查询存储池	206
4. 11 集群拓扑（集群版适用）	212
4. 11. 1 创建拓扑节点	212
4. 11. 2 修改拓扑节点信息	213
4. 11. 3 删除拓扑节点	215
4. 11. 4 查询拓扑信息	217
4. 12 服务器操作	219
4. 12. 1 添加服务器（集群版适用）	219
4. 12. 2 修改服务器端口范围	222
4. 12. 3 设置服务器目标门户 IP	223
4. 12. 4 设置服务器默认数据目录（单机版适用）	225
4. 12. 5 迁移服务器上的基础服务（集群版适用）	226
4. 12. 6 移除服务器（集群版适用）	228
4. 12. 7 查询服务器	230
4. 12. 8 添加数据目录	241
4. 12. 9 修改数据目录的容量配额	243
4. 12. 10 移除数据目录	245
4. 13 查询 HBlock 信息	247
4. 14 查看 HBlock 服务状态	250
4. 15 监控	254
4. 15. 1 查询实时性能数据	254
4. 15. 2 导出性能数据	258

4.16 告警	260
4.16.1 查看告警信息	260
4.16.2 导出告警	266
4.16.3 手动解除告警	267
4.16.4 静默告警	268
4.16.5 解除告警静默	269
4.17 事件和日志管理	270
4.17.1 查看 HBlock 事件	270
4.17.2 导出 HBlock 事件	274
4.17.3 发起 HBlock 日志采集	275
4.17.4 查看 HBlock 采集的日志	277
4.17.5 删除 HBlock 采集的日志	280
4.18 HBlock 系统设置	281
4.18.1 修改管理员密码	281
4.18.2 邮件设置	282
4.18.3 远程协助	287
4.18.4 pushgateway 监控配置	289
4.18.5 智维推送告警配置	301
4.19 调整 HBlock 性能参数（集群版适用）	305
4.19.1 调整 HBlock 性能参数	305
4.19.2 查看性能调优配置	306
4.20 停止服务器上的 HBlock	307
4.21 启动服务器上的 HBlock	308
4.22 重启服务器上的 HBlock	309
4.23 卸载 HBlock	310
4.24 查看 HBlock 版本	311
4.25 升级 HBlock	312

4. 25. 1 升级 HBlock	312
4. 25. 2 查看升级状态	317
5 常见问题	320
5. 1 部署环境要求	320
5. 2 常见操作	322
5. 3 升级扩容	335
5. 4 数据迁移	336
5. 5 数据安全	337
5. 6 技术支持	338
5. 7 购买指南	339
5. 8 其他问题	340
6 故障处理	341
7 附录	342
7. 1 HBlock 服务	342
7. 2 用户事件列表	343
7. 3 系统事件列表	347
7. 4 监控指标	351
7. 5 告警列表	356
7. 6 OOS Endpoint 和 Region	362
7. 7 HBlock 可推送的监控指标	363
7. 8 集群拓扑文件	367

1 产品概述

1.1 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 & 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。

在非联网模式下，HBlock 可被视为本地盘阵的替代品，用于本地数据存储；在联网模式下，HBlock 还可以作为本地与云端存储之间的桥梁，将全量数据自动同步到对象存储中，本地仅保留热数据以节省本地存储空间，或者保留全量数据以保障本地 I/O 性能，实现混合云存储。

HBlock 可以像普通应用程序那样以非 root 方式安装在 Linux 操作系统中，与服务器中的其他应用混合部署，形成的高性能高可用的虚拟硬盘供业务使用。如此一来，HBlock 可以在不影响用户业务、无需额外采购设备的情况下，直接原地盘活存储资源！

传统的硬件存储阵列可以为每个逻辑卷提供低延迟和高可用性，但存在横向扩展性差、成本高的问题，并且可能形成许多“数据孤岛”，导致存储资源成本高和利用率低。传统的分布式存储虽然具有很强的吸引力，但通常存在部署复杂、性能差、稳定性差等问题。

HBlock 以完全不同的方式提供存储阵列：

- **易安装：**HBlock 安装包是一个 zip 包，可以安装在通用 64 位 x86 服务器、ARM 服务器、龙芯服务器上的主流 Linux 操作系统上，支持物理服务器、裸金属服务器、虚拟机。

HBlock 与硬件驱动程序完全解耦，用户可以自由使用市场上最新的硬件，减少供应商锁定。

- **绿色：**HBlock 作为一组用户态进程运行，不依赖特定版本的 Linux 内核或发行版，不依赖、不修改操作系统环境，不独占整个硬盘，不干扰任何其他进程的执行。因此，

HBlock可以与其他应用同时运行在同一Linux操作系统实例中。我们称此功能为“绿色”。一方面，它可以帮助用户提高现有硬件资源的利用率，另一方面，它也降低了用户使用HBlock的门槛——甚至不需要虚拟机。

- **高利用率：**HBlock支持异构硬件，集群中的每个Linux操作系统实例可以具有不同的硬件配置，例如不同数量的CPU、不同的内存大小、不同容量的本地硬盘等。因此可以提高现有硬件资源的利用率。
- **高性能：**HBlock采用分布式多控架构，提供像传统硬件存储阵列一样的低延迟和高可用性，以及像传统分布式存储一样的高扩展性和高吞吐量。支持在不中断业务的情况下，从3台服务器扩展到数千台服务器，并从数千台服务器逐台缩小到3台服务器。
- **高质量：**当集群中同时发生的磁盘故障数不大于逻辑卷冗余模式允许的故障数（对于3副本模式，允许的故障数为2；对于纠删码N+M模式，允许的故障数为M），不影响HBlock的数据持久性。在集群中发生单个服务器、链路或磁盘故障时，HBlock保证服务可用。HBlock是面向混沌（Chaos）环境设计的，可适用于弱网、弱算、弱盘等不确定环境，并在发布之前已经在复杂和大规模的环境进行了充分的测试。HBlock支持快照和克隆功能，用户可以为逻辑卷创建时间点快照，实现数据的快速备份和恢复。同时，克隆功能允许基于快照快速生成新的卷副本，显著提升数据复制和测试环境搭建的效率。

同时，作为本地与云端存储之间的桥梁，HBlock还具有如下特点：

- **支持数据上云：**HBlock可以与兼容S3的对象存储结合，创建本地、存储和缓存模式的卷。
 - 对于本地模式的卷：数据全部存储在本地，不仅具有传统硬件存储阵列的低延迟和高可用性，而且兼具传统分布式存储的高扩展性和高吞吐量。绿色特性使用户部署成本大大降低，异构硬件特性提高了现有硬件资源的利用率；
 - 对于存储模式的卷：全量数据不仅存储在本地，还会被异步地复制到对象存储，实现本地高性能和异地数据灾备；
 - 对于缓存模式的卷：最近读写的数据会缓存在本地以尽可能提高性能，全量数据将保存在对象存储上以降低成本，使得很小的本地容量可以存储海量的数据，特

别适合于数据备份、归档等对实时性要求不高的业务，以及文档卷宗、医疗影像、视频监控等写入多调阅少的业务。HBlock 可将本地应用与云端存储无缝连接，实现存储空间的按需使用，弹性扩展。

- **一致性：**HBlock 利用了对对象存储的原子操作，能够真正确保云上数据的一致性（即任何时候云上数据都是本地数据的一个快照），不会出现因云上数据的不一致而导致无法恢复整个业务的情况，从而保证数据安全。

1.2 应用场景

- **存储资源利旧：**利用HBlock的广泛兼容性，纳管各类服务器中的空闲存储空间，整合成存储池，并通过iSCSI协议向其他主机提供高可用高性能的虚拟盘。面对业务快速增长带来的存储容量需求，及各类型服务器资源闲置带来的资源浪费问题，HBlock提供的快速部署和扩容的解决方案，实现了无需额外成本投入，即可提升存储资源的利用率，并支持业务的滚动更新，满足未来业务在容量和性能上不断变化的需求。
- **小型分布系统存储高可用：**利用HBlock纳管应用节点的物理盘，并把所形成的虚拟盘再挂回到应用节点本地，使得应用程序访问的是高可用的虚拟盘。一方面让应用程序更容易实现高可用，另一方面盘活应用节点的存储资源，无需额外购买存储硬件，降低用户TCO
- **新建自主管控：**通过HBlock纳管新建的存储资源池，用户可以保持对存储服务器的实际管控权，意味着用户不仅能够使用HBlock进行存储管理，还可以在硬件上部署其他应用，以充分发挥硬件的价值。传统的软硬一体式存储产品，或者分布式存储方案，要求独占设备，用户只能通过管理界面进行有限的操作，失去了对设备的管理权。使用HBlock管理的存储集群，实现了用户对资源池的全面管控，提升了更多的操作自由度。后期对资源池进行升级扩容，可以任意选择任意规格和型号的服务器，无供应商锁定问题，客户结合自身的业务需求和预算灵活选择合适的硬件，从而保护了客户的投资。
- **混合云存储：**HBlock可以同时管理本地存储资源和对象存储资源，实现存储空间的统一管理，满足客户混合云存储的需求。对于需要存储海量数据的客户，可以通过HBlock将本地应用与云端存储无缝连接，将数据同步到云端，实现存储空间的按需使用，弹性扩展。对于数据安全性要求较高，敏感数据不适宜上云的场景，HBlock也可以帮助用户实现数据本地存储，提高数据访问速度。此外，HBlock简化了混合云存储环境中的数据管理。通过标准iSCSI协议为上层应用提供虚拟Target和逻辑卷，它除了可以部署在本地，还可以部署在私有云或公有云上。
- **快速构建类生产环境：**基于分布式块存储的多卷一致性快照能力，可瞬间捕获上层业务系统多个应用的数据状态。当发生数据误删或逻辑错误时，借助快照实现秒级恢复，显著降低数据丢失风险，是数据备份、恢复和版本管理的理想选择。此外，利用秒级创建

克隆卷的特性，可快速复制出多个独立的、与生产环境一致的测试环境。这极大提升了数据复制和环境搭建效率，可以便捷地验证应用的快速恢复与灾难恢复能力，同时在不干扰生产环境的前提下，高效进行业务测试、升级验证、数据分析等工作。

- **异地多活高可靠存储：**HBLOCK通过存储协议栈优化与分布式算法创新，实现跨可用区数据强一致性（RPO=0），并具备秒级故障恢复能力（RTO秒级）。系统可自动感知故障，触发跨可用区数据重建，结合智能数据冗余与副本折叠技术，在确保数据安全的同时降低存储开销。单可用区故障时，服务仍可用，业务无感知切换，全程无需人工干预，保障数据安全与业务连续性。

1.3 基本概念

1.3.1 iSCSI

iSCSI (Internet Small Computer System Interface, 互联网小型计算机系统接口) 是一种基于 TCP/IP 及 SCSI-3 协议下的存储技术, 用来建立和管理 IP 存储设备、主机和客户机等之间的相互连接, 并创建存储区域网络 (SAN)。

1.3.2 卷

LUN (Logical Unit Number, 卷) 是逻辑单元号, 用于标识逻辑单元的数字。可以根据需求, 创建本地卷 (本地模式) 和上云卷 (缓存模式和存储模式):

- 本地模式: 数据全部保留在本地。
- 缓存模式: 本地保留部分热数据, 全部数据异步存储到对象存储中。
- 存储模式: 本地保留全部数据, 并异步存储到对象存储中。

最小副本数: 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 $\leq X$), 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 $N \leq Y \leq N+M$), 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。

1.3.3 快照

快照 (Snapshot) 记录了 HBlock 卷在某一特定时间点的状态和内容。它类似于拍照, 能够捕捉并保存卷数据在该时刻的完整状态。当需要时, 用户可以通过快照进行回滚操作, 或者基于快照创建克隆卷, 从而快速恢复到该特定状态, 有效实现数据保护。

一致性快照是指在某一特定时刻, 同时对选中的所有卷创建快照, 从而确保这些快照反映的是同一时间点的数据状态。

1.3.4 克隆卷

克隆卷是基于快照创建的卷，与快照时刻的数据和状态一致。它可独立于源卷进行修改和操作，仅记录克隆后新增或修改的数据，原始数据依赖源卷存储。克隆卷与快照断开关系链后，快照时刻的数据将被全量复制到克隆卷，成为独立卷，与普通卷无异。

克隆深度是指从一个快照克隆出来的链式克隆卷的层级数，即从初始源卷开始，可以连续创建多少层克隆卷。例如：

- 以 lun1 为源卷，创建快照 snap1，再从 snap1 创建克隆卷 lun1-C1，此时克隆深度为 1。
- 以 lun1-C1 为新的源卷，创建快照 snap2，再从 snap2 创建克隆卷 lun1-C2，此时克隆深度为 2。

1.3.5 iSCSI Target

iSCSI Target 是位于 iSCSI 服务器上的存储资源。它是一个通过 IP 网络基础设施来连接数据存储设备的协议，允许将远程存储设备映射到本地主机，提供了一种基于网络的存储解决方案。

1.3.6 iSCSI 目标门户

iSCSI 目标门户即 HBlock 服务器的目标门户，用于与 HBlock 不在同一局域网的 Initiator 通信。

如果与 HBlock 不在同一局域网的 Initiator 想访问 HBlock 某一服务器，需要先进行网络配置（如 NAT 等），确保 Initiator 可以通过该 IP 地址访问 HBlock 服务器，然后将该地址配置为 HBlock 服务器的目标门户，之后 Initiator 即可通过配置的目标门户访问 HBlock 服务器。

1.3.7 存储池

存储池：硬件提供的存储资源的集合，物理上通常指的是同一种介质的硬盘(跨多台服务器)的集合。也可以根据自己的需要将不同机房、机柜的服务器上的数据目录组成存储池。

基础存储池：初始化时默认创建的存储池为基础存储池。通过 3.7 之前的版本升级上来的物理资源也都属于基础存储池。

1.3.8 集群拓扑

集群拓扑：对集群物理资源实际部署方式的可视化逻辑展示。

1.3.9 故障域

存储池支持设置故障域，故障域是指一组可能因共享某些基础设施（如机房、机架、服务器等）而同时发生故障的组件的集合，同一故障域内组件的故障仅影响本故障域内的数据。

HBlock 将副本模式数据的各个副本或者 EC 模式数据的各个分块，按照故障域分配并存储，以达到数据保护的目的。同一数据的各个副本，以及同一数据的 EC 分块，写入不同故障域中。

1.3.10 数据服务

一个数据目录对应一个数据服务，HBlock 通过数据服务管理数据目录内用户的文件数据块。

1.3.11 监控

监控是指对 HBlock 系统、存储池、服务器、数据目录、卷的性能指标进行监测记录，用户可以查看实时或者历史性能数据，关注存储服务的性能。监控指标详见附录**监控指标**。

1.3.12 事件

事件指系统记录的用户操作 HBlock 的行为或 HBlock 系统行为，方便排查故障、审计和跟踪，全方面掌控存储运行情况。

事件分为用户事件和系统事件：

- 用户事件：用户操作 HBlock 的行为，具体用户事件列表详见附录**用户事件列表**。

- 系统事件：HBlock 系统行为，具体系统事件列表详见附录**系统事件列表**。

1.3.13 日志采集

日志采集是指用户主动采集 HBlock 日志数据，生成日志文件后下载到本地查看日志详细内容，以便排查故障。可以通过指定时间段、日志类型和服务器，缩小日志采集范围加快采集进度。

1.3.14 告警

告警指系统检测到 HBlock 业务或系统异常时产生的信息。

告警分为三个级别：

- 警告（Warning）：指一般性的，系统检测到潜在的或即将发生的影响业务的故障，当前还没有影响业务的告警。维护人员需及时查找告警原因，消除故障隐患。
- 重要（Major）：指局部范围内的、对系统性能造成影响的告警。需要尽快处理，否则会影响重要功能运行。
- 严重（Critical）：指带有全局性的、已经造成业务中断，或者会导致瘫痪的告警。需立即处理，否则系统有崩溃危险。

告警列表详见附录**告警列表**。

1.4 使用限制

项目	描述
iSCSI Target	HBlock 上 iSCSI Target IQN 的数量最多 32766 个。一个 iSCSI Target 最多可以关联 256 个卷，但是一个卷只能被一个 iSCSI Target 关联。
存储池	HBlock 集群中最多可以创建 32768 个存储池。
快照	● 单卷支持的最大快照数：512。 ● 系统支持的最大快照数：100000。 ● 单个快照可创建的最大克隆卷数：512。 ● 系统支持的最大快照深度：512。
一致性快照	一致性快照添加的最大卷个数：512。
克隆	● 系统支持的最大克隆卷数：100000。 ● 单个快照可创建的最大克隆卷数：512。 ● 系统支持的最大克隆深度：16。
服务器	每台服务器最多只能添加 100 个数据目录。

1.5 术语与缩略语

术语与缩略语	描述
ALUA	Asymmetric Logical Unit Access，非对称逻辑单元访问。
CHAP	Challenge Handshake Authentication Protocol，质询握手认证协议。
CIDR	Classless Inter-Domain Routing，无类别域间路由，一个用于给用户分配 IP 地址以及在互联网上有效地路由 IP 数据包的对 IP 地址进行归类的方法。
DSM	Device Specific Module，设备特定模块。
EC	Erase Coding，纠删码。
interface IP address	接口 IP 地址，可以通过命令行 <code>ifconfig</code> 查看。
IQN	iSCSI Qualified Name，iSCSI 限定名。
iSCSI	Internet Small Computer System Interface，互联网小型计算机系统接口。
iSCSI initiator	iSCSI 发起程序。
IOPS	Input/Output Operations Per Second，每秒读写次数。
I/O	Input/Output，输入/输出。
LUN	Logical Unit Number，逻辑单元号。
MPIO	Multipath I/O，多路径 IO 管理。
NFS	Network File System，网络文件系统。
NTP	Network Time Protocol，网络时间协议。
OOS	Object-Oriented Storage，天翼云对象存储（经典版）I 型。
RAID	Redundant Arrays of Independent Disks，独立磁盘冗余阵列。
SAN	Storage Area Network，存储区域网络。
SPC	SCSI（Small Computer System Interface，小型计算机系统接口）Primary Commands，SCSI 基础命令。
SSD	Solid State Disk，固态硬盘。
SSL	Secure Sockets Layer，安全套接字协议。

Target	存储目标。
UUID	Universally Unique Identifier，通用唯一识别码。
WWID	The World Wide Identification Number，卷的唯一标识符。如果客户端连接卷的时候，HBlock 端有多个卷，可以通过 WWID 标识符来确认所要连接的卷。
数据目录	用于存储 HBlock 数据的目录。

2 服务器端部署

2.1 环境要求

项目	描述
支持 Linux OS	CentOS 7、8、9，CTyunOS 3。64 位操作系统。 ARM 架构的硬件环境下，推荐使用 PageSize 为 4k 的操作系统。
硬件	x86 服务器、ARM 服务器或者龙芯服务器。 最低配置：单核 CPU、2GB 内存。可根据实际业务需要增加配置。
带宽	● 客户端到 HBlock 的带宽：读写带宽能力大于业务读写带宽。 ● 数据目录对应磁盘分区的写带宽能力大于用户实际写入数据的带宽。 ● 上云带宽大于业务写入带宽。
安装目录所在盘	10GB 以上，建议配置为 RAID 1 或者 RAID 10。
数据目录	● 最小配置：5GB，可根据实际业务需要增加配置。 ● 根据存储容量和副本模式配置数据目录对应分区的容量。 对于 HBlock 使用到的目录，建议设置开机自动挂载，或使用已设置自动挂载的目录或子目录。
网络设定	● 可以与对象存储进行网络连接（部署本地模式的卷则不需要）。 ● 若 HBlock 与 OOS 之间配有专线，须在 HBlock 服务器 /etc/hosts 配置 OOS 资源池的内网域名解析（请联系我们获取），以确保是通过专线访问 OOS，保证访问速度。若您使用互联网，则不需要配置。 网络整体架构如下： 1. HBlock 内部各节点之间通过内网互联。 2. HBlock 与上层应用之间通过内网或专线或公网互联。 3. HBlock 与对象存储之间通过专线或公网互联（部署本地模式的卷则不需

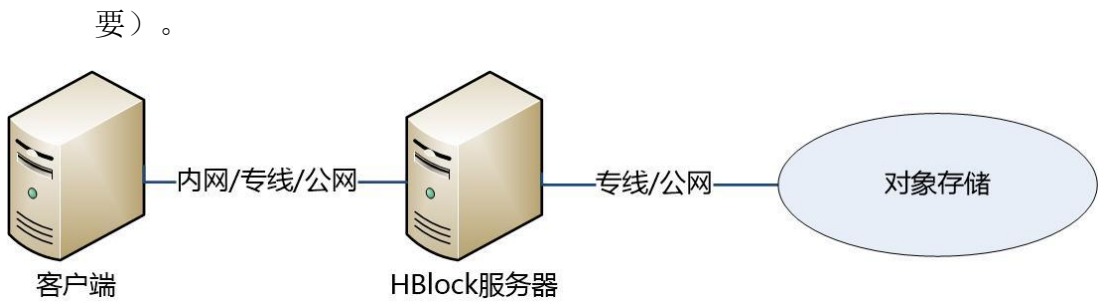


图1.单机版网络拓扑图

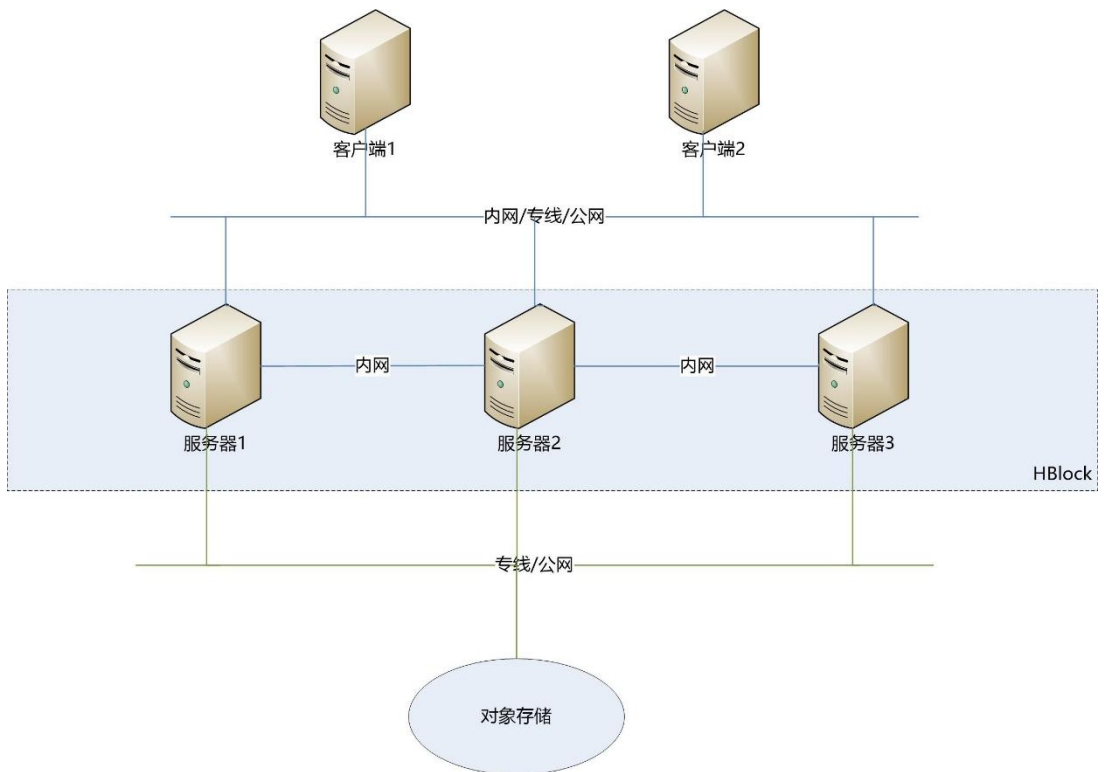


图2.HBlock 集群版网络拓扑图

注意：在部署 HBlock 前，需要明确使用**单机版**还是**集群版**，因为一旦部署后，不支持通过增减服务器进行模式切换。

2.2 HBlock 部署 – 单机版

2.2.1 配置环境

按照环境要求，准备 1 台服务器。

注意：确保 ping 命令和 ps 命令可用。Debian/Ubuntu 可以使用下列命令安装 ping 命令和 ps 命令。

```
apt-get update    #获取最新安装包
apt-get install iputils-ping    #安装 ping 命令
apt-get install procps    #安装 ps 命令安装
```

服务器按照下列操作步骤完成配置，以下操作以 CentOS 7.x 版本为例：

说明：如果已经安装操作系统，请忽略步骤一。如果磁盘已挂载，请忽略步骤二，可以使用挂载路径作为 HBlock 的数据目录，或者使用命令 `mkdir DIRECTORY` 在挂载路径下创建一个目录，将此目录作为 HBlock 数据目录。

(一)安装操作系统 CentOS 7.x 版本（可选）

(二)格式化硬盘并挂载（可选）

请参考下列示例将您服务器上的硬盘进行格式化并挂载，方便后续部署使用。

```
lsblk    #查看硬盘
mkfs.ext4 /dev/vdX    #将硬盘格式化为 ext4，如果已经格式化磁盘，请忽略此步骤。
mkdir DIRECTORY    #创建挂载路径，DIRECTORY 为路径名
mount /dev/vdX DIRECTORY    #挂载硬盘，挂载后，可以使用该路径作为 HBlock 数据目录
```

说明：mount 命令为临时挂载命令，服务器重启后，需要再次挂载。对于 HBlock 使用到的目录，建议设置开机自动挂载，或使用已设置自动挂载的目录或子目录。

注意：如果安装 HBlock 的用户为非 root 用户，需要对 HBlock 使用到的目录有读写权限，可以使用下列命令。

```
chown HBlock 用户:HBlock 用户所属组 DIRECTORY
```

(三)关闭 selinux 和 swap 分区（建议）

(四) 防火墙设定

若您的服务器未开启防火墙，可以忽略此步骤。若您的服务器已开启防火墙，请开启 iSCSI 端口，以便客户端连接到服务器的 Target。示例如下：

1. 开启 iSCSI 端口，如 iSCSI 端口为 3260 时：

```
firewall-cmd --permanent --add-port=3260/tcp
```

2. 重新加载防火墙使配置生效：

```
firewall-cmd --reload
```

(五) 设置资源限制

修改配置文件 `/etc/security/limits.conf`，在配置文件中增加下列内容，设置在 *domain* 中打开的最大文件数、同时运行的最大进程数。

说明：仅非 root 用户需要手动修改 `/etc/security/limits.conf`。

```
domain soft nfile 65536    # 参数 domain 根据情况设置为具体的值
domain hard nfile 65536    # 参数 domain 根据情况设置为具体的值
domain soft nproc 65535    # 参数 domain 根据情况设置为具体的值
domain hard nproc 65535    # 参数 domain 根据情况设置为具体的值
```

domain 为可变参数，可以根据情况设置为对应的 *username*、*groupname*、*uid*、*wildcard*。

注意：如果 *domain* 设置为对应的 *username*，则必须包含启动 HBlock 服务的用户。

例 1：例如 *domain* 取值为 `*`，表示所有用户打开的最大文件数为 65536，同时运行的最大进程数为 65535。

```
* soft nfile 65536    # *为参数 domain 的取值
* hard nfile 65536    # *为参数 domain 的取值
* soft nproc 65535    # *为参数 domain 的取值
* hard nproc 65535    # *为参数 domain 的取值
```

例 2：例如 *domain* 取值为 `root`，表示 `root` 用户打开的最大文件数为 65536，同时运行的最大进程数为 65535。

```
root soft nfile 65536    # root 为参数 domain 的取值
```

```
root hard nfile 65536    # root 为参数 domain 的取值
root soft nproc 65535    # root 为参数 domain 的取值
root hard nproc 65535    # root 为参数 domain 的取值
```

2.2.2 部署 HBlock

部署 HBlock 的主要步骤为：

(一)安装前准备：准备一个或多个目录作为 HBlock 数据目录，安装 HBlock 的用户对这些目录有读写权限，用来存储 HBlock 数据。

说明：为了避免相互影响，建议数据目录不要与操作系统共用磁盘或文件系统。

(二)解压缩安装包，并进入解压缩后的文件夹路径。

(三)安装并初始化 HBlock。

(四)获取软件证书并加载。

(五)创建 iSCSI Target 并查询。

(六)创建卷并查询。

说明：下面以 x86 服务器的 HBlock 安装部署举例，ARM 服务器或者龙芯服务器的安装部署与 x86 服务器的安装部署相同。

详细步骤如下：

(一)请先完成以下准备工作：在服务器上准备一个或多个目录作为 HBlock 数据目录，用来存储 HBlock 数据。如：/mnt/storage01，/mnt/storage02。

(二)将安装包放到服务器欲安装 HBlock 的目录下并解压缩，进入解压缩后的文件夹。

```
unzip CTYUN_HBlock_Plus_3.9.0_x64.zip
cd CTYUN_HBlock_Plus_3.9.0_x64
```

(三)安装并初始化 HBlock。

1. 安装 HBlock。

注意：安装 HBlock 和执行 HBlock 管理操作的应该属于同一用户。

在服务器上安装 HBlock。

```
./stor install [ { -a | --api-port } API_PORT ] [ { -w | --web-port } WEB_PORT ]
```

API_PORT：指定 API 端口号，默认端口号为 1443。

WEB_PORT: 指定 WEB 端口号，默认端口号为 2443。

2. 初始化 HBlock。

说明: 可以通过 web、命令行和 API 进行初始化 HBlock。

初始化 HBlock 具体命令行详见**初始化 HBlock**。

```
./stor setup { -n | --stor-name } STOR_NAME [ { -u | --user-name } USER_NAME ] { -p | --password } PASSWORD { -s | --server } { SERVER_IP[:PORT]:PATH <1-n> } [ { -P | --public-network } CIDR ] [ --iscsi-port ISCSI_PORT ] [ --port-range PORT1-PORT2 ] [ --management-port1 MANAGEMENT_PORT1 ] [ --management-port2 MANAGEMENT_PORT2 ] [ --management-port3 MANAGEMENT_PORT3 ] [ --management-port4 MANAGEMENT_PORT4 ] [ --management-port6 MANAGEMENT_PORT6 ]
```

3. 查询服务器。

```
./stor server ls [ { -n | --server } SERVER_ID ] [ --port ]
```

(四) 获取软件许可证并加载。

HBlock 软件提供 30 天试用期，过期后无法进行管理操作。您可以通过下列步骤获取软件许可证。

1. 获取 HBlock 序列号:

```
./stor info { -S | --serial-id }
```

2. 联系 HBlock 软件供应商获取软件许可证，获取的时候需要提供 HBlock 序列号。

3. 获取软件许可证后，执行加载。

```
./stor license add { -k | --key } KEY
```

(五) 创建 iSCSI Target 并查询。

1. 创建 iSCSI Target。

创建 iSCSI Target 命令行详见**创建 iSCSI Target**。

```
./stor target add { -n | --name } TARGET_NAME [ --max-sessions MAX_SESSIONS ] [ --reclaim-policy RECLAIM_POLICY ] [ { -c | --chap-name } CHAP_NAME { -p | --password } CHAP_PASSWORD { -s | --status } STATUS ]
```

说明：如果允许 Target iSCSI Target 下的 IQN 建立的多个会话，可以通过配置参数 **--max-sessions MAX_SESSIONS** 来实现。

2. 查询 iSCSI Target。

```
./stor target ls [ -c | --connection ] [ { -n | --name } TARGET_NAME ]
```

(六)创建卷并查询卷。

1. 创建卷。

创建卷命令行详见[创建卷](#)。

■ 本地卷

```
./stor lun add { -n | --name } LUN_NAME { -p | --capacity } CAPACITY { -t | --target } TARGET_NAME [ { -o | --sector-size } SECTOR_SIZE ] [ { -w | --write-policy } WRITE_POLICY ] [ { -P | --path } PATH ] [ { { -m | --mode } STORAGE_MODE ]
```

■ 上云卷

```
./stor lun add { -n | --name } LUN_NAME { -p | --capacity } CAPACITY { -t | --target } TARGET_NAME [ { -o | --sector-size } SECTOR_SIZE ] [ { -w | --write-policy } WRITE_POLICY ] [ { -P | --path } PATH ] { -m | --mode } STORAGE_MODE [ --cloud-provider CLOUD_PROVIDER ] { -B | --bucket } BUCKET_NAME { -A | --ak } ACCESS_KEY { -S | --sk } SECRET_KEY [ { -C | --cloud-storage-class } CLOUD_STORAGE_CLASS ] { -E | --endpoint } ENDPOINT [ --sign-version VERSION ] [ --region REGION ] [ { -M | --cloud-compression } CLOUD_COMPRESSION ] [ { -O | --object-size } OBJECT_SIZE ] [ { -X | --prefix } PREFIX ]
```

2. 查询卷。

```
./stor lun ls [ { -n | --name } LUN_NAME ]
```

2.2.3 部署 HBlock 示例

应用场景：

- 服务器 IP 地址为：192.168.0.32，API 端口号为 1443，WEB 端口号为 2443，安装路径为/mnt/stor（挂载磁盘/dev/vdd），数据目录为/mnt/storage01（挂载磁盘/dev/vdc）、/mnt/storage02（挂载磁盘/dev/vda）。
- 创建卷：本地模式的卷 luna1，对应 iSCSI Target 为 targeta，卷容量为 100G；缓存模式卷 lunb1，对应 iSCSI Target 为 targetb，卷容量为 200G；存储模式卷 lunc1，对应 iSCSI Target 为 targetc，卷容量为 300G。
- 缓存模式和存储模式卷对应的 OOS 存储桶为：hblocktest3。
- HBlock 系统名称为 stor1，HBlock 登录用户名为 storuser，登录密码为 hblock12@。

操作步骤：

(一)完成以下准备工作：安装路径/mnt/stor，数据目录/mnt/storage01、/mnt/storage02。OOS 存储桶 hblocktest3，AK/SK，前缀 stor1。

(二)将安装包放到服务器欲安装 HBlock 的目录下并解压缩，进入解压缩后的文件夹。

```
[root@hblockserver stor]# ls
CTYUN_HBlock_Plus_3.9.0_x64.zip
[root@hblockserver stor]# unzip CTYUN_HBlock_Plus_3.9.0_x64.zip
.....
[root@hblockserver stor]# cd CTYUN_HBlock_Plus_3.9.0_x64
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]#
```

(三)安装并初始化 HBlock。

1. 安装 HBlock。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor install -w 2443
Do you agree with HBlock User Agreement? [Yes/No]
Used in Chinese mainland, follow https://www.ctyun.cn/portal/protocol/10073150
Otherwise, follow https://www.esurfingcloud.com/portal/protocol/20692906
```

```
y
Installing HBlock...
Installed successfully.
When all servers are installed, please initialize HBlock in any of the following ways:
1. Use web portal to initialize HBlock. The https port is 2443.
2. Use management API (POST /rest/v1/system/setup) to initialize HBlock. The https port
is 1443.
3. Use command line (stor setup) to initialize HBlock. Type 'stor --help setup' for more
information.
```

2. 初始化 HBlock。

密码可以按交互式输入，使用交互式输入密码时，以密文形式显示。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor setup -n stor1 -u storuser -s
192.168.0.32:/mnt/storage01,/mnt/stor
Please enter password:
*****
Start to setup HBlock, please wait.
Processing...
Setup successfully and the HBlock services have been started.
Welcome to HBlock!
You are using a 30-day trial version. Please follow the steps to get a license.
1. Run "stor info --serial-id" to get the serial ID of the HBlock
2. Contact the software vendor to obtain a license
3. Run "stor license add -k KEY" to import the license

Type 'stor --help' to get more information, such as managing LUNs, targets, servers,
etc.
```

3. 查询服务器。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor server ls -n hblock_1

Server Name: hblockserver
Server ID:
hblock_1
Status: Connected
```

```
Public Address: 192.168.0.32:3260
```

```
Cluster Address: 192.168.0.32
```

```
Recent Start Time: 2025-01-17 10:18:23
```

```
Version: 3.9.0
```

```
Disk Path(s):
```

No.	Path	Used Capacity	Total Capacity	Used Capacity Quota	Capacity Quota	Health Status	Health Detail
1.	/mnt/storage01(*)	19.37 GiB	93.29 GiB	4 KiB	Unlimited	Healthy	
2.	/mnt/stor	1.17 GiB	93.29 GiB	4 KiB	Unlimited	Healthy	-

(四)获取软件许可证并加载。

HBlock 软件提供 30 天试用期，过期后无法进行管理操作。您可以通过下列步骤获取软件许可证并加载。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor info --serial-id
```

```
HBlock serial ID: HBlock serial ID: CF317E33-A0E7-433C-9D2B-774828642C4F-0201-030700
```

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor license add -k *****
```

```
Import license successfully.
```

```
The current system time: 2025-01-17 14:48:34
```

LicenseId	Account	Type	Status	MaximumLocalCapacity
ehc2b6a9-f3fb-4098-a6b3-3652a5d71232	testaccount@ctyun.cn	Subscription	Effective	1 PiB

```
License ehc2b6a9-f3fb-4098-a6b3-3652a5d71232 (Effective):
```

```
Usage:
```

MaximumLocalCapacity	EffectiveTime	ExpireTime	Status
1 PiB	2025-01-17 14:42:08	2025-08-16 14:42:08	Effective

(五)创建 iSCSI Target 并查询。

1. 创建 iSCSI Target。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor target add -n targeta --max-
```

```
sessions 3 -c chap-test -p ***** -s on
Created Target targeta successfully.
iqn = iqn.2012-08.cn.ctyunapi.oos:targeta.1(192.168.0.32:3260)
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor target add -n targetb --max-
sessions 2
Created Target targetb successfully.
iqn = iqn.2012-08.cn.ctyunapi.oos:targetb.2(192.168.0.32:3260)
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor target add -n targetc
Created Target targetc successfully.
iqn = iqn.2012-08.cn.ctyunapi.oos:targetc.3(192.168.0.32:3260)
```

2. 查询 iSCSI Target

查询所有 iSCSI Target。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor target ls
```

No.	Target Name	Max Sessions	iSCSI Target	CHAP
1.	targeta	3	iqn.2012-08.cn.ctyunapi.oos:targeta.1(192.168.0.32:3260)	chap-test,Enabled
2.	targetb	2	iqn.2012-08.cn.ctyunapi.oos:targetb.2(192.168.0.32:3260)	Disabled
3.	targetc	1	iqn.2012-08.cn.ctyunapi.oos:targetc.3(192.168.0.32:3260)	Disabled

查询指定 iSCSI Target，例如查询 targeta。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor target ls -n targeta
Target Name: targeta
Max Sessions: 3
Create Time: 2025-01-17 14:50:23
iSCSI Target: iqn.2012-08.cn.ctyunapi.oos:targeta.1(192.168.0.32:3260)
Reclaim Policy: Delete
CHAP: chap-test,test1test2024,Enabled
```

(六)创建卷并查询。

1. 创建卷。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun add -n luna1 -p 100 -t
targeta
```

```
Created LUN luna1 successfully.
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun add -n lunb1 -t targetb
-p 200 -m Cache -B hblocktest3 -E oos-cn.ctyunapi.cn -A cb22b08b1f9229f85874 -S
***** --sign-version v4 --region cn -X stor1
Created LUN lunb1 successfully.
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun add -n lunc1 -t targetc
-p 300 -m Storage -B hblocktest3 -E oos-cn.ctyunapi.cn -A cb22b08b1f9229f85874 -S
***** --sign-version v4 --region cn -X stor1
Created LUN lunc1 successfully.
```

2. 查询卷

查询所有卷。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun ls
```

No.	LUN Name	Storage Mode	Capacity	Status	Target	Snapshot Numbers	Is Clone
1.	luna1(LUN 0)	Local	100 GiB	Normal	iqn.2012-08.cn.ctyunapi.oos:targeta.1(192.168.0.32:3260,Active)	0	
2.	lunb1(LUN 0)	Cache	200 GiB	Normal	iqn.2012-08.cn.ctyunapi.oos:targetb.2(192.168.0.32:3260,Active)	0	
3.	lunc1(LUN 0)	Storage	300 GiB	Normal	iqn.2012-08.cn.ctyunapi.oos:targetc.3(192.168.0.32:3260,Active)	0	

查询指定卷，例如查询卷 luna1。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun ls -n luna1
LUN Name: luna1 (LUN 0)
Storage Mode: Local
Capacity: 100 GiB
Status: Normal
iSCSI Target: iqn.2012-08.cn.ctyunapi.oos:targeta.1(192.168.0.32:3260,Active)
Create Time: 2025-01-17 14:52:42
Local Sector Size: 4096 bytes
Write Policy: WriteBack
WWID: 33ffffffffce6bc2a6
UUID: lun-uuid-9885d5c6-23f7-4739-b580-fc5a233ea3bd
Path: /mnt/storage01
Snapshot Numbers: 0
```

2.3 HBlock 部署 – 集群版

2.3.1 配置环境

按照环境要求，准备 3 台或 3 台以上的服务器。

注意：确保 ping 命令和 ps 命令可用。Debian/Ubuntu 可以使用下列命令安装 ping 命令和 ps 命令。

```
apt-get update /*获取最新安装包
apt-get install iputils-ping /* 安装 ping 命令
apt-get install procps /* 安装 ps 命令安装
```

每台服务器按照下列操作步骤完成配置，以下操作以 CentOS 7.x 版本为例：

说明：如果已经安装操作系统，请忽略步骤一。如果磁盘已挂载，请忽略步骤二，可以使用挂载路径作为 HBlock 的数据目录，或者使用命令 `mkdir DIRECTORY` 在挂载路径下创建一个目录，将此目录作为 HBlock 数据目录。

(一)安装操作系统 CentOS 7.x 版本

(二)格式化硬盘并挂载

请参考下列示例将服务器上的硬盘进行格式化，方便后续部署使用。

```
lsblk #查看硬盘
mkfs.ext4 /dev/vdX #将硬盘格式化为 ext4,如果已经格式化磁盘，请忽略此步骤。
mkdir DIRECTORY #创建挂载路径，DIRECTORY 为路径名
mount /dev/vdX DIRECTORY #挂载硬盘，挂载后，可以使用该路径作为 HBlock 数据目录
```

说明：mount 命令为临时挂载命令，服务器重启后，需要再次挂载。对于 HBlock 使用到的目录，建议设置开机自动挂载，或使用已设置自动挂载的目录或子目录。

注意：如果安装 HBlock 的用户为非 root 用户，需要对 HBlock 使用到的目录有读写权限，可以使用下列命令。

```
chown HBlock 用户:HBlock 用户所属组 DIRECTORY
```

(三)关闭 selinux 和 swap 分区（建议）

(四) 防火墙设定

确保集群服务器之间可以相互访问，集群服务器之间相互添加白名单，另外请开启 iSCSI 端口，以便客户端连接到服务器的 Target。如果是在云主机上安装，安全组中也需要添加白名单。

若您的服务器未开启防火墙，可以忽略此步骤。

示例如下：

1. 开启 iSCSI 端口，如 iSCSI 端口为 3260 时：

```
firewall-cmd --permanent --add-port=3260/tcp
```

2. 集群各服务器的 IP 添加白名单：

- 添加 IPv4 地址

```
firewall-cmd --permanent --add-rich-rule="rule family=ipv4 source address=your_IP  
accept" // your_IP is IP address allowed to access
```

- 添加 IPv6 地址

```
firewall-cmd --permanent --add-rich-rule="rule family=ipv6 source address=your_IP  
accept" // your_IP is IP address allowed to access
```

3. 重新加载防火墙使配置生效：

```
firewall-cmd --reload
```

(五) 设置资源限制

修改配置文件 `/etc/security/limits.conf`，在配置文件中增加下列内容，设置在 *domain* 中打开的最大文件数、同时运行的最大进程数。

说明：仅非 root 用户需要手动修改 `/etc/security/limits.conf`。

```
domain soft nofile 65536 # 参数 domain 根据情况设置为具体的值  
domain hard nofile 65536 # 参数 domain 根据情况设置为具体的值  
domain soft nproc 65535 # 参数 domain 根据情况设置为具体的值  
domain hard nproc 65535 # 参数 domain 根据情况设置为具体的值
```

domain 为可变参数，可以根据情况设置为对应的 *username*、*groupname*、*uid*、*wildcard*。

注意：如果 *domain* 设置为对应的 *username*，则必须包含启动 HBlock 服务的用户。

例 1：例如 *domain* 取值为*，表示所有用户打开的最大文件数为 65536，同时运行的最大进程数为 65535。

```
* soft nfile 65536    # *为参数 domain 的取值
* hard nfile 65536    # *为参数 domain 的取值
* soft nproc 65535    # *为参数 domain 的取值
* hard nproc 65535    # *为参数 domain 的取值
```

例 2：例如 *domain* 取值为 root，表示 root 用户打开的最大文件数为 65536，同时运行的最大进程数为 65535。

```
root soft nfile 65536    # root 为参数 domain 的取值
root hard nfile 65536    # root 为参数 domain 的取值
root soft nproc 65535    # root 为参数 domain 的取值
root hard nproc 65535    # root 为参数 domain 的取值
```

2.3.2 部署 HBlock

部署 HBlock 的步骤主要步骤为：

(一)安装前准备：在每台服务器上准备一个或多个目录作为 HBlock 数据目录，用来存储 HBlock 数据。

说明：为了避免相互影响，建议数据目录不要与操作系统共用磁盘或文件系统。

(二)解压缩安装包，并进入解压缩后的文件夹路径。

(三)安装并初始化 HBlock。

(四)获取软件证书并加载。

(五)创建 iSCSI Target 并查询。

(六)创建卷并查询。

说明：下面以 x86 服务器的 HBlock 安装部署举例，ARM 服务器、龙芯服务器的安装部署与 x86 服务器的安装部署相同。

详细步骤如下：

(一)请先完成以下准备工作：在每台服务器上准备一个或多个目录作为 HBlock 数据目录（每台服务器上的目录可以不同），用来存储 HBlock 数据，如：/mnt/storage01，/mnt/storage02。

(二)将安装包放到服务器欲安装 HBlock 的目录下并解压缩，进入解压缩后的文件夹。

说明：建议安装目录不要与数据目录共用磁盘或文件系统。

```
unzip CTYUN_HBlock_Plus_3.9.0_x64.zip
cd CTYUN_HBlock_Plus_3.9.0_x64
```

(三)安装并初始化 HBlock：

1. 安装 HBlock。

注意：安装 HBlock 和执行 HBlock 管理操作的应该属于同一用户。

在每台服务器上安装 HBlock。

```
./stor install [ { -a | --api-port } API_PORT ] [ { -w | --web-port } WEB_PORT ]
```

API_PORT: 指定 API 端口号，默认端口号为 1443。

WEB_PORT: 指定 WEB 端口号，默认端口号为 2443。

2. 初始化 HBlock。

说明: 可以通过 web、命令行和 API 进行初始化 HBlock。可以在已经安装的 HBlock 的任一台服务器上执行初始化。

初始化 HBlock 具体命令行详见**初始化**。

```
./stor setup { -n | --stor-name } STOR_NAME [ { -u | --user-name } USER_NAME ] { -p | --password } PASSWORD { { -s | --server } { SERVER_IP[:PORT][:PATH] &<1-n> } &<1-n> | --topology-file TOPOLOGY_FILE } [ { -C | --cluster-network } CIDR ] [ { -P | --public-network } CIDR ] [ --fault-domain FAULT_DOMAIN ] [ --iscsi-port ISCSI_PORT ] [ --port-range PORT1-PORT2 ] [ --data-port1 DATA_PORT1 ] [ --management-port1 MANAGEMENT_PORT1 ] [ --management-port2 MANAGEMENT_PORT2 ] [ --management-port3 MANAGEMENT_PORT3 ] [ --management-port4 MANAGEMENT_PORT4 ] [ --management-port5 MANAGEMENT_PORT5 ] [ --management-port6 MANAGEMENT_PORT6 ] [ --metadata-port1 METADATA_PORT1 ] [ --metadata-port2 METADATA_PORT2 ] [ --metadata-port3 METADATA_PORT3 ] [ --metadata-port4 METADATA_PORT4 ] [ --metadata-port5 METADATA_PORT5 ] [ --metadata-port6 METADATA_PORT6 ] [ --metadata-port7 METADATA_PORT7 ] [ --metadata-port8 METADATA_PORT8 ] [ --cs SERVER_IP[:DIR],SERVER_IP[:DIR],SERVER_IP[:DIR] ] [ --mdm SERVER_IP[:DIR],SERVER_IP[:DIR] ] [ --ls SERVER_IP[:DIR],SERVER_IP[:DIR],SERVER_IP[:DIR] ]
```

3. 查询服务器。

```
./stor server ls [ { -n | --server } SERVER_ID ] [ --port ]
```

(四) 获取软件许可证并加载

HBlock 软件提供 30 天试用期，过期后无法进行管理操作。您可以通过下列步骤获取软件许可证。

1. 获取 HBlock 序列号：

```
./stor info { -S | --serial-id }
```

2. 联系 HBlock 软件供应商获取软件许可证，获取的时候需要提供 HBlock 序列号。
3. 获取软件许可证后，执行加载。

```
./stor license add { -k | --key } KEY
```

(五)创建 iSCSI Target 并查询。

1. 创建 iSCSI Target。

创建 iSCSI Target 命令行详见[创建 iSCSI Target](#)。

```
./stor target add { -n | --name } TARGET_NAME [ --max-sessions MAX_SESSIONS ] [ --reclaim-policy RECLAIM_POLICY ] [ { -c | --chap-name } CHAP_NAME { -p | --password } CHAP_PASSWORD { -s | --status } STATUS ] [--num SERVER_NUMBER] [ --server SERVER_ID &<1-n> ]
```

说明：如果允许 Target iSCSI Target 下的 IQN 建立的多个会话，可以通过配置参数 **--max-sessions MAX_SESSIONS** 来实现。

2. 查询 iSCSI Target。

```
./stor target ls [ -c | --connection ] [ { -n | --name } TARGET_NAME ]
```

(六)创建卷并查询。

1. 创建卷

创建卷命令行详见[创建卷](#)。

■ 创建本地卷

```
./stor lun add { -n | --name } LUN_NAME { -p | --capacity } CAPACITY { -t | --target } TARGET_NAME [ --priority SERVER_ID &<1-n> [ --auto-failback AUTO_FAILBACK ] ] [--pool
```

```
POOL ] [ --cache-pool CACHE_POOL ] [ { -a | --ha } HIGH_AVAILABILITY ] [ { -c | --local-storage-class } LOCAL_STORAGE_CLASS ] [ --min-replica MIN_REPLICA ] [ --redundancy-overlap REDUNDANCY_OVERLAP ] [ --ec-fragment-size EC_FRAGEMENT_SIZE ] [ { -o | --sector-size } SECTOR_SIZE ] [ { -w | --write-policy } WRITE_POLICY ] [ { -m | --mode } STORAGE_MODE ]
```

■ 创建上云卷

```
./stor lun add { -n | --name } LUN_NAME { -p | --capacity } CAPACITY { -t | --target } TARGET_NAME [ --priority SERVER_ID &<1-n> [ --auto-failback AUTO_FAILBACK ] ] [--pool POOL ] [ --cache-pool CACHE_POOL ] [ { -a | --ha } HIGH_AVAILABILITY ] [ { -c | --local-storage-class } LOCAL_STORAGE_CLASS ] [ --min-replica MIN_REPLICA ] [ --redundancy-overlap REDUNDANCY_OVERLAP ] [ --ec-fragment-size EC_FRAGEMENT_SIZE ] [ { -o | --sector-size } SECTOR_SIZE ] [ { -w | --write-policy } WRITE_POLICY ] { -m | --mode } STORAGE_MODE [ --cloud-provider CLOUD_PROVIDER ] { -B | --bucket } BUCKET_NAME { -A | --ak } ACCESS_KEY { -S | --sk } SECRET_KEY [ { -C | --cloud-storage-class } CLOUD_STORAGE_CLASS ] { -E | --endpoint } ENDPOINT [ --sign-version VERSION ] [ --region REGION ] [ { -M | --cloud-compression } CLOUD_COMPRESSION ] [ { -O | --object-size } OBJECT_SIZE ] [ { -X | --prefix } PREFIX ]
```

2. 查询卷

```
./stor lun ls [ { -n | --name } LUN_NAME ]
```

2.3.3 部署 HBlock 示例

应用场景：

- 三台服务器
 - 服务器 1：IP 地址为 192.168.0.110，API 端口号为 1443，WEB 端口号为 2443，安装路径为/mnt/storage01（磁盘/dev/vdc），数据目录为/mnt/stor（磁盘/dev/vdd）、/mnt/storage02（磁盘/dev/vdb）。
 - 服务器 2：IP 地址为 192.168.0.192，API 端口号为 1443，WEB 端口号为 2443，安装路径为/mnt/storage01（磁盘/dev/vda），数据目录为/mnt/stor（磁盘/dev/vdd）。
 - 服务器 3：IP 地址为 192.168.0.102，API 端口号为 1443，WEB 端口号为 2443，安装路径为/mnt/storage01（磁盘/dev/vda），数据目录为/mnt/stor（磁盘/dev/vdd）。
- 创建卷：本地模式的卷 lun01a，对应 iSCSI Target 为 target01，卷容量为 100G；缓存模式卷 lun02a，对应 iSCSI Target 为 target02，卷容量为 200G；存储模式卷 lun03a，对应 iSCSI Target 为 target03，卷容量为 300G。
- 缓存模式和存储模式卷对应的 OOS 存储桶为：hblocktest3。
- HBlock 系统名称为 stor2，HBlock 登录用户名为 storuser，登录密码为 hblock12@。

操作步骤：

(一)完成以下准备工作：

- 服务器 1 安装路径：/mnt/storage01，数据目录：/mnt/stor、/mnt/storage02。
- 服务器 2 安装路径：/mnt/storage01，数据目录：/mnt/stor。
- 服务器 3 安装路径：/mnt/storage01，数据目录：/mnt/stor。
- OOS 存储桶 hblocktest3，AK/SK，前缀 stor2。

(二)将安装包放到各个服务器的安装 HBlock 目录下并解压缩，进入解压缩后的文件夹。

```
[root@hblockserver storage01]# ls
CTYUN_HBlock_Plus_3.9.0_x64.zip
[root@hblockserver storage01]# unzip CTYUN_HBlock_Plus_3.9.0_x64.zip
```

```
.....  
[root@hblockserver storage01]# cd CTYUN_HBlock_Plus_3.9.0_x64  
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]#
```

(三) 安装并初始化 HBlock。

1. 在每台服务器上都安装 HBlock，执行一遍下列步骤。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor install -w 2443  
Do you agree with HBlock User Agreement? [Yes/No]  
Used in Chinese mainland, follow https://www.ctyun.cn/portal/protocol/10073150  
Otherwise, follow https://www.esurfingcloud.com/portal/protocol/20692906  
y  
Installing HBlock...  
Installed successfully.  
When all servers are installed, please initialize HBlock in any of the following ways:  
1. Use web portal to initialize HBlock. The https port is 2443.  
2. Use management API (POST /rest/v1/system/setup) to initialize HBlock. The https port  
is 1443.  
3. Use command line (stor setup) to initialize HBlock. Type 'stor --help setup' for more  
information.
```

2. 初始化 HBlock

密码也可以按交互式输入，使用交互式输入密码时，以密文形式显示。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor setup -n stor2 -p hblock12@ -s  
192.168.0.110:/mnt/stor,/mnt/storage02 192.168.0.192:/mnt/stor 192.168.0.102:/mnt/stor  
Start to setup HBlock, please wait.  
Processing...  
Setup successfully and the HBlock services have been started.  
Welcome to HBlock!  
You are using a 30-day trial version. Please follow the steps to get a license.  
1. Run "stor info --serial-id" to get the serial ID of the HBlock  
2. Contact the software vendor to obtain a license  
3. Run "stor license add -k KEY" to import the license
```

Type 'stor --help' to get more information, such as managing LUNs, targets, servers, etc.

3. 查看服务器

查看所有服务器。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor server ls
```

No.	Server ID	Server Name	Status	Public Address	Cluster Address	Recent Start Time
1.	hblock_1(M**)	ecs-9689-0915140	Connected	192.168.0.102:3260	192.168.0.102	2025-01-17 13:58:21
2.	hblock_2(**)	pm-006	Connected	192.168.0.192:3260	192.168.0.192	2025-01-17 14:11:52
3.	hblock_3(**)	hblockserver	Connected	192.168.0.110:3260	192.168.0.110	2025-01-17 14:12:01

查看指定服务器。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor server ls -n hblock_1
```

Server Name: ecs-9689-0915140

Server ID: hblock_1

Node Name: default:server1

Parent Node: default

Status: Connected

Master Server: true

Base Server: true

Base Service: mdm(Up, /mnt/storage01/CTYUN_HBlock_Plus_3.9.0_x64)

ls(Up, /mnt/storage01/CTYUN_HBlock_Plus_3.9.0_x64)

cs(Up, /mnt/storage01/CTYUN_HBlock_Plus_3.9.0_x64)

Public Address: 192.168.0.102:3260

Cluster Address: 192.168.0.102

Recent Start Time: 2025-01-17 13:58:21

Version: 3.9.0

Disk Path(s):

No.	Path	Storage Pool	Used Capacity	Total Capacity	Used Capacity Quota	Capacity Quota	Health Status	Health Detail
1.	/mnt/stor	default	10.22 GiB	93.29 GiB	138.49 KiB	Unlimited	Healthy	-

(四) 获取软件许可证并加载。

HBlock 软件提供 30 天试用期，过期后无法进行管理操作。您可以通过下列步骤获取软件许可证并加载。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor info --serial-id
HBlock serial ID: 744B6ED7-6954-4BDA-8228-0772F574CA44-0202-030700
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor license add -k *****
Import license successfully.
The current system time: 2025-01-17 18:02:16
```

LicenseId	Account	Type	Status	MaximumLocalCapacity
ehc2b6a9-f3fb-4098-a6b3-3652a5d71232	testaccount@ctyun.cn	Subscription	Effective	1 PiB

```
License ehc2b6a9-f3fb-4098-a6b3-3652a5d71232 (Effective):
Usage:
```

MaximumLocalCapacity	EffectiveTime	ExpireTime	Status
1 PiB	2025-01-17 18:01:40	2025-08-17 18:01:40	Effective

(五) 创建 iSCSI Target 并查询。

1. 创建 iSCSI Target。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor target add -n target01 --
max-sessions 6 -c chap-test1 -p T12345678912 -s on --num 3 --server
hblock_1,hblock_2,hblock_3
Created Target target01 successfully.
iqn = iqn.2012-08.cn.ctyunapi.oos:target01.1(192.168.0.110:3260)
iqn = iqn.2012-08.cn.ctyunapi.oos:target01.2(192.168.0.192:3260)
iqn = iqn.2012-08.cn.ctyunapi.oos:target01.16(192.168.0.102:3260)
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor target add -n target02 --
max-sessions 2
Created Target target02 successfully.
iqn = iqn.2012-08.cn.ctyunapi.oos:target02.3(192.168.0.102:3260)
iqn = iqn.2012-08.cn.ctyunapi.oos:target02.4(192.168.0.110:3260)
```

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor target add -n target03
Created Target target03 successfully.
iqn = iqn.2012-08.cn.ctyunapi.oos:target03.5(192.168.0.102:3260)
iqn = iqn.2012-08.cn.ctyunapi.oos:target03.6(192.168.0.110:3260)
```

2. 查询 iSCSI Target。

查询所有 iSCSI Target。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor target ls
```

No.	Target Name	Max Sessions	ISCSI Target	CHAP
1.	target01	6	iqn.2012-08.cn.ctyunapi.oos:target01.1(192.168.0.110:3260)	chap-test1,Enabled
			iqn.2012-08.cn.ctyunapi.oos:target01.2(192.168.0.192:3260)	
			iqn.2012-08.cn.ctyunapi.oos:target01.16(192.168.0.102:3260)	
2.	target02	2	iqn.2012-08.cn.ctyunapi.oos:target02.3(192.168.0.102:3260)	Disabled
			iqn.2012-08.cn.ctyunapi.oos:target02.4(192.168.0.110:3260)	
3.	target03	1	iqn.2012-08.cn.ctyunapi.oos:target03.5(192.168.0.102:3260)	Disabled
			iqn.2012-08.cn.ctyunapi.oos:target03.6(192.168.0.110:3260)	

查询指定 iSCSI Target，例如 target01。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor target ls -n target01
Target Name: target01
Max Sessions: 6
Create Time: 2025-01-17 19:50:29
Number of Servers: 3
iSCSI Target: iqn.2012-08.cn.ctyunapi.oos:target01.1(192.168.0.110:3260)
               iqn.2012-08.cn.ctyunapi.oos:target01.2(192.168.0.192:3260)
               iqn.2012-08.cn.ctyunapi.oos:target01.16(192.168.0.102:3260)
Reclaim Policy: Retain
CHAP: chap-test1,T12345678912,Enabled
ServerID: hblock_1,hblock_2,hblock_3
```

(六)创建卷并查询。

1. 创建卷。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun add -n lun01a -t
target01 -c EC 2+1 --ec-fragment-size 32 -p 100 -m Local
```

```
Created LUN lun01a successfully.
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun add -n lun02a -t
target02 -c EC 2+1 --ec-fragment-size 32 -p 200 -m Cache -B hblocktest3 -E oos-
cn.ctyunapi.cn -A cb22b08b1f9229f85874 -S ***** --sign-version v4 --region
cn -X stor2
Created LUN lun02a successfully.
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun add -n lun03a -t
target03 -c EC 2+1 --ec-fragment-size 32 -p 300 -m Storage -B hblocktest3 -E oos-
cn.ctyunapi.cn -A cb22b08b1f9229f85874 -S ***** --sign-version v4 --region
cn -X stor2
Created LUN lun03a successfully.
```

2. 查询卷。

查询所有卷。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun ls
```

No.	LUN Name	Storage Mode	Capacity	Local Storage Class	Minimum Replica Number	Status	Target	Snapshot Numbers	Is Clone
1.	lun01a(LUN 0)	Local	100 GiB	EC 2+1+32KiB	2	Normal	iqn.2012-08.cn.ctyunapi.oos:target01.16(192.168.0.102:3260,Active)	0	
							iqn.2012-08.cn.ctyunapi.oos:target01.2(192.168.0.192:3260,Standby)		
							iqn.2012-08.cn.ctyunapi.oos:target01.1(192.168.0.110:3260,ColdStandby)		
2.	lun02a(LUN 0)	Cache	200 GiB	EC 2+1+32KiB	2	Normal	iqn.2012-08.cn.ctyunapi.oos:target02.3(192.168.0.102:3260,Active)	0	
							iqn.2012-08.cn.ctyunapi.oos:target02.4(192.168.0.110:3260,Standby)		
3.	lun03a(LUN 0)	Storage	300 GiB	EC 2+1+32KiB	2	Normal	iqn.2012-08.cn.ctyunapi.oos:target03.5(192.168.0.102:3260,Active)	0	
							iqn.2012-08.cn.ctyunapi.oos:target03.6(192.168.0.110:3260,Standby)		

查询指定卷，例如 lun02a。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun ls -n lun02a
LUN Name: lun02a (LUN 0)
Storage Mode: Cache
Capacity: 200 GiB
Status: Normal
Auto Failback: Enabled
iSCSI Target: iqn.2012-08.cn.ctyunapi.oos:target02.3(192.168.0.110:3260,Active)
               iqn.2012-08.cn.ctyunapi.oos:target02.4(192.168.0.102:3260,Standby)
Create Time: 2025-01-17 20:50:29
Local Storage Class: EC 2+1+32KiB
```

```
Minimum Replica Number: 2
Redundancy Overlap: 1
Local Sector Size: 4096 bytes
High Availability: ActiveStandby
Storage Pool: default
Write Policy: WriteBack
WWID: 33ffffffff805dcfc1
UUID: lun-uuid-9457ebfc-8440-4e69-bc9c-c054938e5551
Object Storage Info:
+-----+-----+
| Provider      | OOS                |
| Bucket Name   | hblocktest3       |
| Prefix        | stor2              |
| Endpoint      | https://oos-cn.ctyunapi.cn |
| Signature Version | v4                |
| Region        | cn                  |
| Storage Class  | STANDARD           |
| Access Key     | cb22b08b1f9229f85874 |
| Object Size    | 1024 KiB           |
| Compression    | Enabled            |
+-----+-----+
```

3 客户端操作

3.1 Windows 客户端 – 单机版

(一)准备客户端操作系统

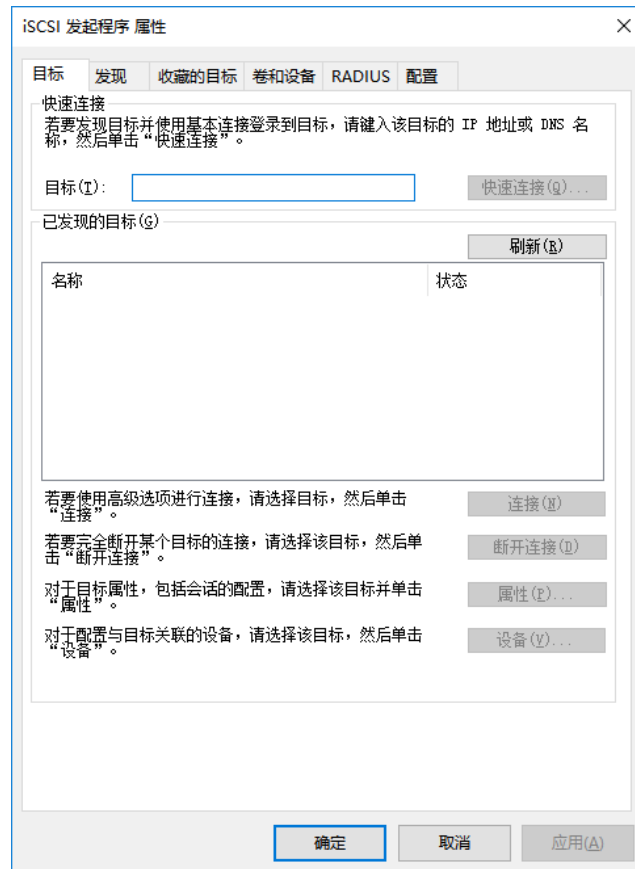
推荐使用 win10、windows server 2012R2、windows server 2016R2 等高版本的 Windows 操作系统，这些系统中自带了“iSCSI 发起程序”，无需单独安装组件。

不同版本的客户端支持单卷容量不同，请参考下表：

Windows 版本	Block Size	单卷最大容量
Windows Server 2008R2	512 bytes / 4KiB	256 TiB
Windows Server 2012R2	512 bytes / 4KiB	256 TiB
Windows Server 2016	512 bytes / 4KiB	256 TiB
Windows 10	512 bytes / 4KiB	1 PiB

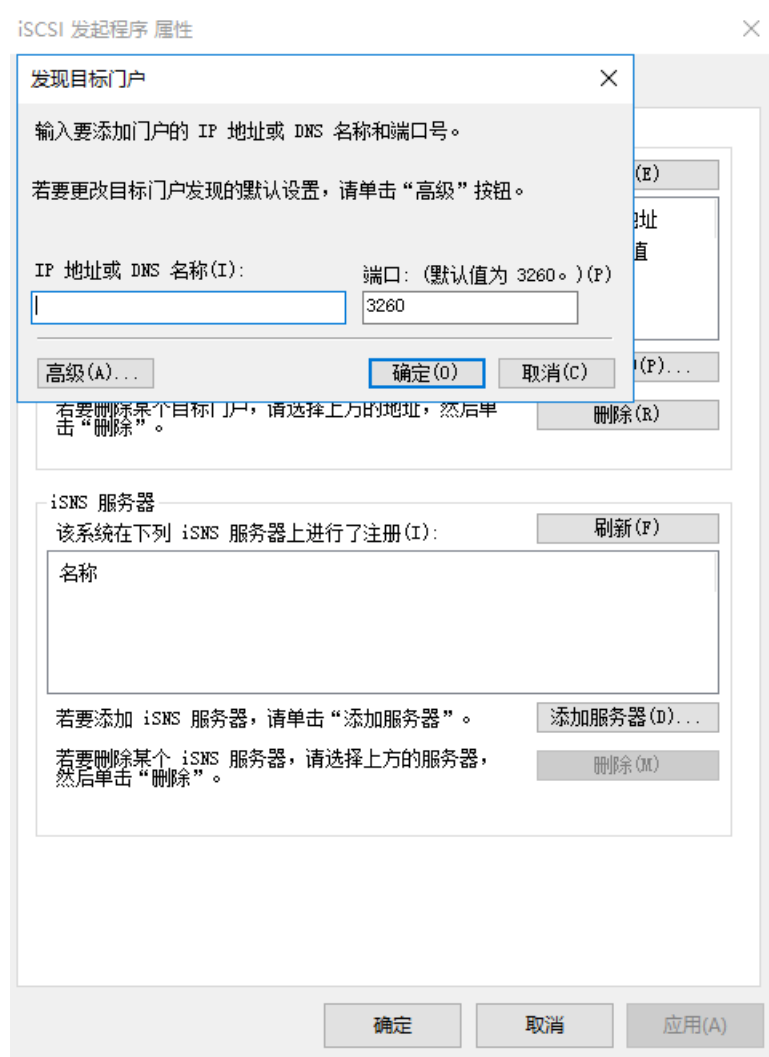
(二)运行 iSCSI 发起程序

Windows 客户端运行 iSCSI 发起程序，在“开始”>“搜寻程序和文件”输入“iSCSI”打开 iSCSI 发起程序，如下图所示：

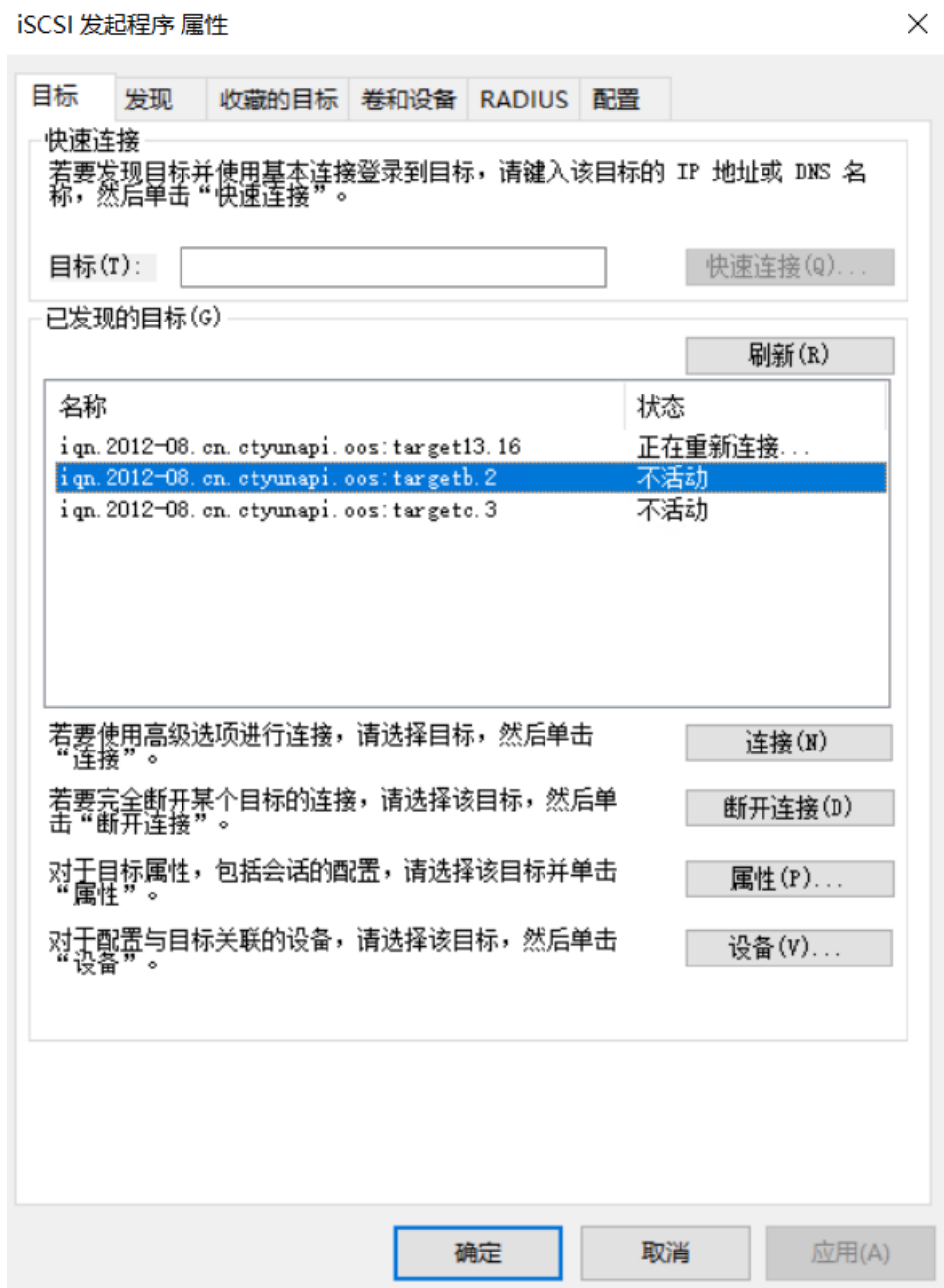


1. 配置 iSCSI 发起程序

在“发现”>“发现门户”中输入服务器 IP 和端口，如下图所示：

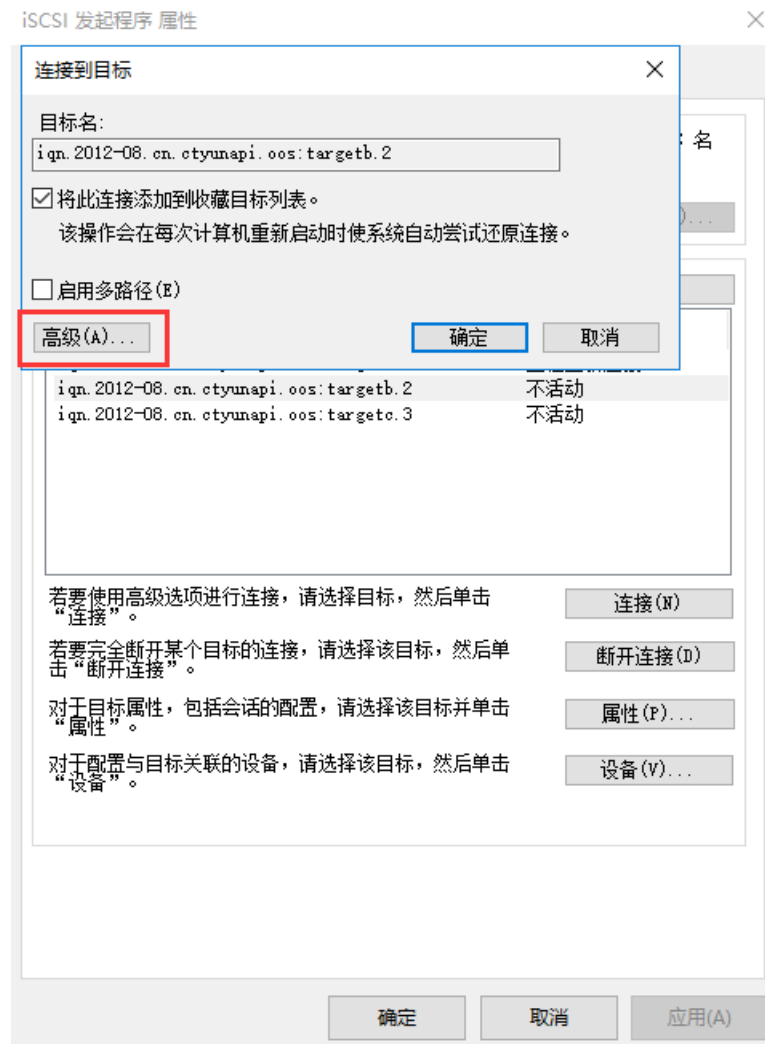


在“目标”>“已发现目标”中搜索到 HBlock 发布的 iSCSI Target，查看到状态是“不活动”，点击“连接”，如下图所示：



2. 启用 CHAP 认证

若您的 iSCSI Target 有开启 CHAP 认证，在弹出的连接到目标的对话框中，选择“高级”，如下图所示（没有开启请忽略此步骤直接连接即可）：



勾选“启用 CHAP 登录”，在“名称”中输入在 HBlock 系统中设置的 iSCSI 认证的用户名，在“目标机密”中输入设置的 iSCSI 认证的密码，然后点“确定”。如下图所示：



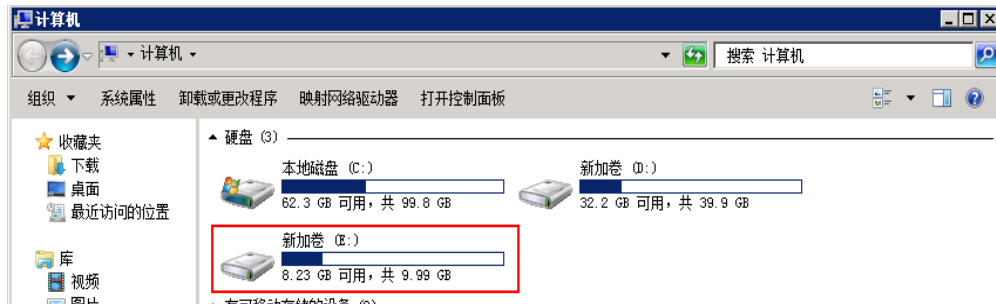
3. 客户端使用 iSCSI 共享磁盘

打开“服务器管理器”>“存储”>“磁盘管理”，将刚刚连接成功的状态是“脱机”的磁盘“联机”。

然后“初始化”，再“新建卷”，指定盘符并格式化，如下图所示。

注意：如果卷容量小于等于 2TiB 时，可以使用 MBR 和 GPT 中的任意一种进行分区；如果卷容量大于 2TiB，只能使用 GPT 分区。

打开“计算机”，可以看到新增的磁盘的盘符和容量，此时可以按使用本地磁盘的习惯使用 HBlock 发布的 iSCSI 磁盘了。如下图所示：



注意：

- 如果客户端需要断开连接或者删除磁盘，需要先打开“服务器管理器” > “存储” > “磁盘管理”，点击磁盘右键进行“脱机”，然后在“iSCSI 发起程序”中“断开 iSCSI 连接”。
- 如果客户端需要断开连接后再次接入，无需进行初始化、新建卷操作，重新连接后即可看到磁盘。

说明：如果想查询 HBlock 卷对应的磁盘，可以在客户端输入下列命令行查询。

```
wmic diskdrive get Name, Manufacturer, Model, InterfaceType, MediaType, SerialNumber
```

如下例所示，查询信息 Name 列对应的盘符号，对应“磁盘”上的“数目”列。

SerialName 对应 HBlock 的卷名称和 uuid。

```
C:\Users\Administrator>wmic diskdrive get Name, Manufacturer, Model, InterfaceType, MediaType, SerialNumber
InterfaceType Manufacturer MediaType Model Name SerialNumber
SCSI (标准磁盘驱动器) Fixed hard disk media Red Hat VirtIO SCSI Disk Device \\.\PHYSICALDRIVE0 230a6f34-09d9-440e-9
SCSI (标准磁盘驱动器) Fixed hard disk media CTYUN \\.\PHYSICALDRIVE1 lunbl-lun-uuid-7be39633-2790-42be-8439-74fd6428df27
```

服务器管理器 · 文件和存储服务 · 卷 · 磁盘

磁盘 所有磁盘 | 共 2 个

筛选器

数目	虚拟磁盘	状态	容量	未分配	分区	只读	已群集	子系统	总线类型
pm-0309699 (2)									
0		联机	40.0 GB	0.00 B	MBR				SCSI
1		联机	200 GB	0.00 B	GPT				iSCSI

上次刷新时间为 2024/5/20 18:07:09

卷 任务

选择一个磁盘以显示相关卷。

存储池 任务

未选择任何卷。

选择一个磁盘以显示其相关存储池。

3.2 Windows 客户端 – 集群版

Microsoft 在 Server 2008、2012、2016 中提供了通用的 DSM (Device Specific Module)，支持 ALUA (Asymmetric Logical Unit Access)，可与符合 SPC (SCSI Primary Commands) 规范的存储设备配置 MPIO (Multipath I/O) 环境。MPIO 保障了 Active Target、Standby Target 在进行切换时不会影响业务正常运行。因此，建议使用 Microsoft Server 2008、2012、2016 作为 HBlock 的客户端使用，并且配置 MPIO。Windows 7、8、10 不支持 MPIO，不建议使用此系统作为 HBlock 客户端。

(一) 安装 Native MPIO 软件

- **Windows Server 2008 R2**

1. 打开“服务器管理”。
2. 选择“功能”，打开“添加功能”。
3. 点击“下一步”选择“多路径 I/O”安装。
4. 重启 Windows。

- **Windows Server 2012 或 2016**

1. 打开“服务器管理器”，选择“添加角色和功能”。
2. 点击“下一步”，在“功能”步骤中勾选“多路径 I/O”。
3. 点击“下一步”，勾选“如果需要，自动重新启动目标服务器”。
4. 安装，点击“关闭”。

(二) 打开 MPIO 工具添加存储阵列

注意：在 Windows 系统使用 iSCSI MPIO 时，请通过“设备管理器”查看并设置 MPIO 的负载均衡策略。（iSCSI 发起程序未使用 MPIO_DSM_Path_V2 WMI 类来显示状态，因此通过 iSCSI 发起程序设置 MPIO 负载均衡策略后，会显示与预期不符的状态）。

1. 点击“管理工具”>“MPIO”。
2. 点击“发现多路径”，勾选“添加对 iSCSI 设备的支持”，点击“添加”>“确定”。
3. 重启 Windows。

(三) 调整 MPIO 配置

1. 打开 Powershell，开启路径检测和自定义路径恢复功能。

```
Get-MPIOSetting # 查看当前配置

Set-MPIOSetting -NewPathVerificationState Enabled # 开启路径检测

Set-MPIOSetting -CustomPathRecovery Enabled # 开启自定义路径恢复功能
```

2. 重启 Windows

(四) 运行 iSCSI 发起程序

1. Windows 客户端运行 iSCSI 发起程序，在“开始”>“搜寻程序和文件”输入 **iSCSI** 打开 iSCSI 发起程序。
2. 在“发现”>“发现门户”中输入 LUN 对应 Target 所在的服务器 IP 和 Port。可以在服务器上使用命令 **./stor lun ls** 查询卷的 ACTIVE Target 和 STANDBY Target。

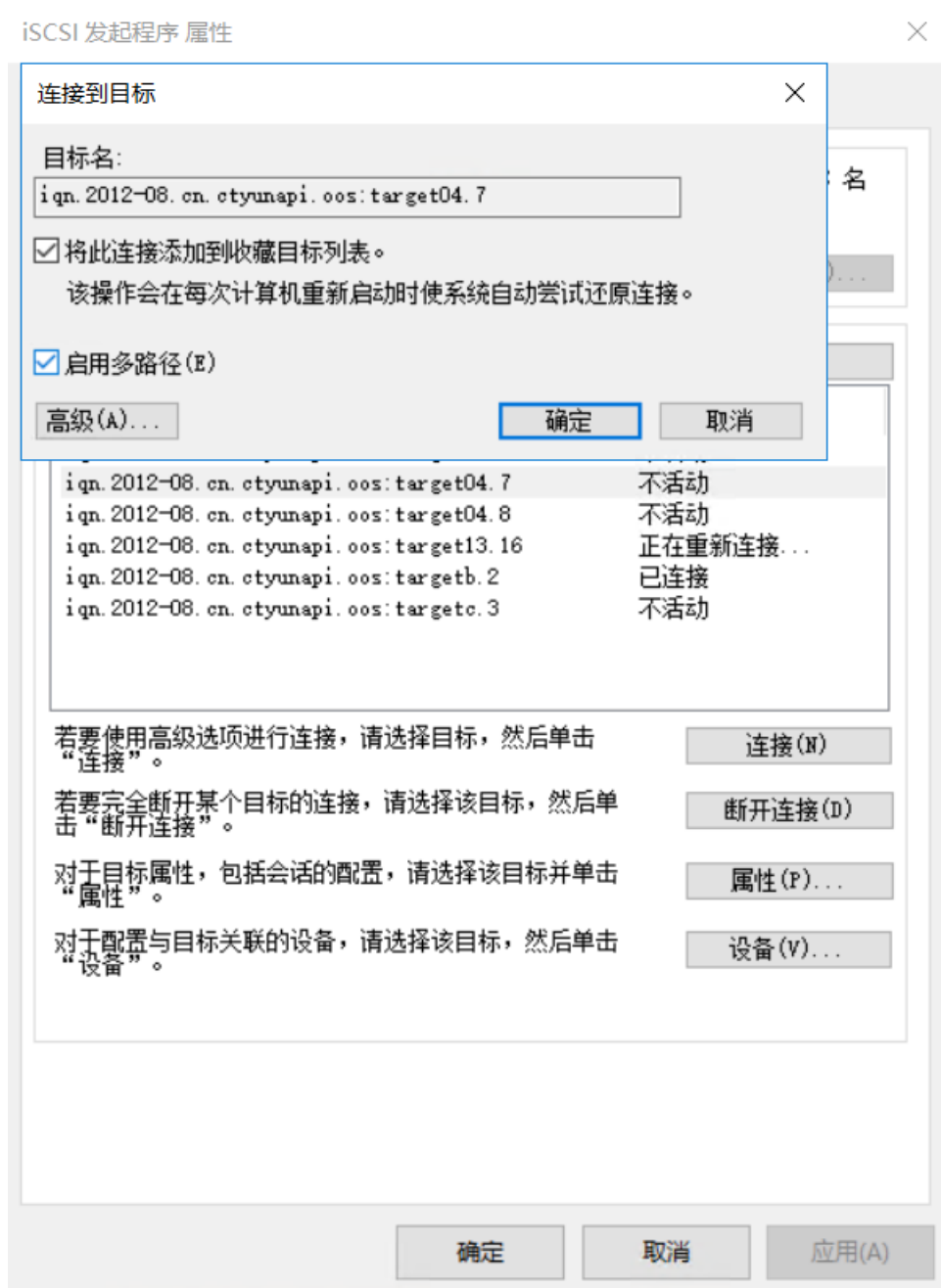
```
./stor lun ls #查看 LUN 所对应 Target 的服务器 IP 和 Port
```

3. 在“目标”>“已发现的目标”中搜索到 HBlock 发布的 iSCSI Target，查看到状态是“不活动”，点击“连接”，勾选“启用多路径”，点击“确定”。

说明：需要先连接 ACTIVE Target，然后连接 STANDBY Target。

注意：

- Windows Server 2012 或 2016：同一个 Target 可以对应多个卷。在一个 Target 可以对应多个卷时，如果不同卷对应的 ACTIVE Target 和 STANDBY Target 不同，iSCSI 连接时，需要等待一会才能识别出所有卷。故建议每个 Target 对应一个卷。
- Windows Server 2008：一个 Target 只能对应一个卷，且先建立的 iSCSI 连接必须为 ACTIVE Target，然后再建立 STANDBY Target 连接，否则无法正常操作 MPIO 设备。



4. 启用 CHAP 认证（没有开启请忽略此步骤直接连接即可）

若您的 iSCSI Target 有开启 CHAP 认证，在弹出的连接到目标的对话框中，选择“高级”，勾选“启用 CHAP 登录”，在“名称”中输入在 HBlock 系统中设置的 iSCSI 认证的用户名，在“目标机密”中输入已设置的 iSCSI 认证的密码，然后点“确定”。

5. 客户端使用 iSCSI 共享磁盘。打开“服务器管理器”>“存储”>“磁盘管理”，将刚刚连接成功的状态是“脱机”的磁盘“联机”。然后点击“初始化”，再点击“新建卷”，指定盘符并格式化，完成后即可看到新增的 iSCSI 设备。

注意：

- 如果卷容量小于等于 2TiB 时，可以使用 MBR 和 GPT 中的任意一种进行分区；如果卷容量大于 2TiB，只能使用 GPT 分区。
- 如果客户端需要断开连接或者删除磁盘，需要先打开“服务器管理器”>“存储”>“磁盘管理”，点击磁盘右键进行“脱机”，然后在“iSCSI 发起程序”中“断开 iSCSI 连接”。
- 如果客户端需要断开连接后再次接入，无需进行初始化、新建卷操作，重新连接后即可看到磁盘。

说明：如果想查询 HBlock 卷对应的磁盘，可以在客户端输入下列命令行查询。

```
wmic diskdrive get Name, Manufacturer, Model, InterfaceType, MediaType, SerialNumber
```

如下例所示，查询信息 Name 列对应的盘符号，对应“磁盘”上的“数目”列。

SerialName 对应 HBlock 的卷名称和 uuid。

```
C:\Users\Administrator>wmic diskdrive get Name, Manufacturer, Model, InterfaceType, MediaType, SerialNumber
InterfaceType  Manufacturer  MediaType  Model
SCSI           (标准磁盘驱动器)  Fixed hard disk media  Red Hat VirtIO SCSI Disk Device
SCSI           (标准磁盘驱动器)  Fixed hard disk media  CTYUN
SCSI           (标准磁盘驱动器)  Fixed hard disk media  CTYUN
```

Name	SerialNumber
\\.\PHYSICALDRIVE0	230a6f34-09d9-440e-9
\\.\PHYSICALDRIVE1	lunb1-lun-uuid-7be39633-2790-42be-8439-74fd6428df27
\\.\PHYSICALDRIVE2	lun04a-lun-uuid-c5aeb48b-f32c-48d0-a9b9-68d06a2a3f64

服务器管理器

服务器管理器 · 文件和存储服务 · 卷 · 磁盘

磁盘
所有磁盘 | 共 3 个

任务

筛选器

数目	虚拟磁盘	状态	容量	未分配	分区	只读	已群集	子系统	总线类型
pm-0309699 (3)									
0		联机	40.0 GB	0.00 B	MBR				SCSI
1		联机	200 GB	0.00 B	GPT				iSCSI
2		联机	440 GB	0.00 B	GPT				iSCSI

上次刷新时间为 2024/5/21 9:44:54

卷
相关卷 | 共 1 个

任务

筛选器

卷	状态	设置	容量	可用空间	重复数
pm-0309699 (1)					
E:	固定		440 GB	440 GB	

存储池
pm-0309699 上的 CTYUN iSCSI LUN Device

任务

不存在相关的存储池。

3.3 Linux 客户端 – 单机版

3.3.1 客户端配置

前置条件

- HBlock 服务器端，已经成功创建卷。
- 准备 Linux 客户端

注意：需要具有 root 权限才能配置 initiator。

若您客户端为 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。

若您客户端为 Ubuntu/Debian，安装命令如下：

```
apt install open-iscsi
```

操作步骤

- HBlock 服务器端：查询要连接的 LUN 及 LUN 对应 iSCSI Target 的详细信息

```
./stor lun ls { -n | --name } LUN_NAME  
./stor target ls { -n | --name } TARGET_NAME
```

- Linux 客户端

1. 发现 HBlock 的 Target

```
iscsiadm -m discovery -t st -p SERVER_IP
```

2. 登录 iSCSI 存储

若您的 iSCSI Target 没有开启 CHAP 认证，请直接执行步骤(4) **登录 Target**。

(1) 开启认证

```
iscsiadm -m node -T iSCSI_TARGET_IQN -o update --name node.session.auth.authmethod --  
value=CHAP
```

(2) 输入 CHAP 用户名

```
iscsiadm -m node -T iSCSI_TARGET_IQN -o update --name node.session.auth.username --value=USER
```

(3) 输入 CHAP 密码

```
iscsiadm -m node -T iSCSI_TARGET_IQN -o update --name node.session.auth.password --value=PASSWORD
```

(4) 登录 Target

```
iscsiadm -m node -T iSCSI_TARGET_IQN -p SERVER_IP:port -l # port 为 iSCSI 端口号
```

3. 显示会话情况，查看当前 iSCSI 连接。

```
iscsiadm -m session
```

4. 查看 iSCSI 磁盘、磁盘对应的 LUN（通过 **ls SCSI -i** 可以查看 LUN 的 wwid）

```
lsblk  
ls SCSI -i #磁盘对应的卷的 WWID
```

5. 格式化 iSCSI 磁盘

注意：如果用户之前已经连接过此磁盘并完成了格式化，重新连接后无需再次进行磁盘格式化，直接挂载 iSCSI 磁盘即可。

格式化当前新增的 iSCSI 磁盘分区，使用

```
mkfs.ext4 /dev/sdX
```

或

```
mkfs.xfs /dev/sdX
```

说明：常用的文件系统有 ext4、XFS，具体格式化成哪种文件系统要视用户文件系统决定。

6. 挂载 iSCSI 磁盘

将 iSCSI 磁盘分区挂载到本地目录上，挂载之后可以写入数据。

```
mount /dev/sdX PATH # PATH 为磁盘路径
```

注意：如果用户需要断开连接或者删除磁盘，执行下列步骤：

- (1) 确保在卸载文件系统之前，没有进程正在使用该文件系统的文件夹。
- (2) 使用 `sync` 命令来确保所有挂起的写操作都已写入磁盘。
- (3) 使用 `umount` 命令来正常卸载文件系统，断开 iSCSI 连接。

```
umount DIRECTORY_NAME_OR_PATH  
iscsiadm -m node -T iSCSI_TARGET_IQN -p SERVER_IP -u
```

示例

```
[root@client ~]# sync  
[root@client ~]# umount /mnt/disk_sda  
[root@client ~]# iscsiadm -m node -T iqn.2012-08.cn.ctyunapi.oos:target1.1 -p  
192.168.0.32 -u  
Logging out of session [sid: 1, target: iqn.2012-08.cn.ctyunapi.oos:target1.1, portal:  
192.168.0.32,3260]  
Logout of [sid: 1, target: iqn.2012-08.cn.ctyunapi.oos:target1.1, portal:  
192.168.0.32,3260] successful.
```

3.3.2 配置举例

应用场景

- Linux 客户端需要连接 HBlock 单机版的卷。
- 需要连接的 HBlock 单机版的卷为 lund1 和 lunf1，lund1 有 CHAP 认证。

前置条件

- 对于需要连接 HBlock 单机版的客户端，已经按照**客户端配置**的前置条件成准备工作。
- 对于 HBlock 服务器端，已经成功创建卷 lund1 和 lunf1。

操作步骤

- HBlock 服务器端：查询要连接的 LUN 及 LUN 对应 iSCSI Target 的详细信息。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun ls -n lund1
LUN Name: lund1 (LUN 0)
Storage Mode: Local
Capacity: 500 GiB
Status: Normal
iSCSI Target: iqn.2012-08.cn.ctyunapi.oos:targetd.4(192.168.0.32:3260,Active)
Create Time: 2024-05-21 10:00:34
Local Sector Size: 4096 bytes
Write Policy: WriteBack
WWID: 33000000068f2f320
UUID: lun-uuid-3ddcc779-bf34-42b9-ac5e-0339dae28821
Path: /mnt/storage01
Snapshot Numbers: 0
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor target ls -n targetd
Target Name: targetd
Max Sessions: 2
Create Time: 2024-05-21 09:59:12
iSCSI Target: iqn.2012-08.cn.ctyunapi.oos:targetd.4(192.168.0.32:3260)
LUN: lund1(LUN 0)
Reclaim Policy: Retain
CHAP: testd,T12345678912,Enabled
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun ls -n lunf1
LUN Name: lunf1 (LUN 0)
Storage Mode: Local
```

```
Capacity: 600 GiB
Status: Normal
iSCSI Target: iqn.2012-08.cn.ctyunapi.oos:targetf.5(192.168.0.32:3260,Active)
Create Time: 2024-05-21 10:00:56
Local Sector Size: 4096 bytes
Write Policy: WriteBack
WWID: 33000000030f798a5
UUID: lun-uuid-7b7f91d8-b75e-4de2-ac69-621e4be7a0cf
Path: /mnt/storage01
Snapshot Numbers: 0
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor target ls -n targetf
Target Name: targetf
Max Sessions: 2
Create Time: 2024-05-21 10:00:15
iSCSI Target: iqn.2012-08.cn.ctyunapi.oos:targetf.5(192.168.0.32:3260)
LUN: lunf1(LUN 0)
Reclaim Policy: Retain
```

- Linux 客户端:

1. 发现 lund1 和 lunf1 的 Target:

```
[root@client ~]# iscsiadm -m discovery -t st -p 192.168.0.32
192.168.0.32:3260,1 iqn.2012-08.cn.ctyunapi.oos:targetf.5
192.168.0.32:3260,1 iqn.2012-08.cn.ctyunapi.oos:targetd.4
192.168.0.32:3260,1 iqn.2012-08.cn.ctyunapi.oos:targetc.3
```

2. 登录 iSCSI 存储

- 登录 lund1 的 iSCSI 存储, 需要进行 CHAP 认证:

```
[root@client ~]# iscsiadm -m node -T iqn.2012-08.cn.ctyunapi.oos:targetd.4 -o update --
name node.session.auth.authmethod --value=CHAP
[root@client ~]# iscsiadm -m node -T iqn.2012-08.cn.ctyunapi.oos:targetd.4 -o update --
name node.session.auth.username --value=testd
[root@client ~]# iscsiadm -m node -T iqn.2012-08.cn.ctyunapi.oos:targetd.4 -o update --
name node.session.auth.password --value=*****
[root@client ~]# iscsiadm -m node -T iqn.2012-08.cn.ctyunapi.oos:targetd.4 -p
192.168.0.32:3260 -l
```

```
Logging in to [iface: default, target: iqn.2012-08.cn.ctyunapi.oos:targetd.4, portal:
192.168.0.32,3260] (multiple)
Login to [iface: default, target: iqn.2012-08.cn.ctyunapi.oos:targetd.4, portal:
192.168.0.32,3260] successful.
```

● 登录 lunfl 的 iSCSI 存储

```
[root@client ~]# iscsiadm -m node -T iqn.2012-08.cn.ctyunapi.oos:targetf.5 -p
192.168.0.32:3260 -l
Logging in to [iface: default, target: iqn.2012-08.cn.ctyunapi.oos:targetf.5, portal:
192.168.0.32,3260] (multiple)
Login to [iface: default, target: iqn.2012-08.cn.ctyunapi.oos:targetf.5, portal:
192.168.0.32,3260] successful.
```

3. 显示会话情况，查看当前 iSCSI 连接。

```
[root@client ~]# iscsiadm -m session
tcp: [1] 192.168.0.32:3260,1 iqn.2012-08.cn.ctyunapi.oos:targetd.4 (non-flash)
tcp: [2] 192.168.0.32:3260,1 iqn.2012-08.cn.ctyunapi.oos:targetf.5 (non-flash)
```

4. 查看 iSCSI 磁盘、磁盘对应的 LUN（通过 **lsscsi -i** 可以查看 LUN 的 wwid）。

```
[root@client ~]# lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
sda      8:0    0 500G  0 disk
sdb      8:16   0 600G  0 disk
vda     253:0    0  40G  0 disk
├─vda1 253:1    0   4G  0 part
└─vda2 253:2    0  36G  0 part /
vdb     253:16   0 100G  0 disk
└─vdb1 253:17   0 100G  0 part /mnt/storage01
vdc     253:32   0 100G  0 disk
vdd     253:48   0 100G  0 disk

[root@client ~]# lsscsi -i
[2:0:0:0]    disk      CTYUN      iSCSI LUN Device 1.00  /dev/sda  330000000068f2f320
[3:0:0:0]    disk      CTYUN      iSCSI LUN Device 1.00  /dev/sdb  330000000030f798a5
```

说明：可以看出/dev/sda 对应 HBlock 卷 lund1（卷 WWID 为 330000000068f2f320），
/dev/sdb 对应 HBlock 卷 lunf1（卷 WWID 为 330000000030f798a5）。

5. 格式化 iSCSI 磁盘。

```
[root@client ~]# mkfs.ext4 /dev/sda
mke2fs 1.42.9 (28-Dec-2013)
/dev/sda is entire device, not just one partition!
Proceed anyway? (y,n) y
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
32768000 inodes, 131072000 blocks
6553600 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2279604224
4000 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424, 20480000, 23887872, 71663616, 78675968,
    102400000

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

[root@client ~]# mkfs.ext4 /dev/sdb
mke2fs 1.42.9 (28-Dec-2013)
/dev/sdb is entire device, not just one partition!
Proceed anyway? (y,n) y
```

```
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
39321600 inodes, 157286400 blocks
7864320 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=230477048
4800 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424, 20480000, 23887872, 71663616, 78675968,
    102400000

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

6. 挂载 iSCSI 磁盘。

将 iSCSI 磁盘分区挂载到本地目录上，挂载之后可以写入数据。

```
[root@client ~]# mount /dev/sda /mnt/disk_sda
[root@client ~]# mount /dev/sdb /mnt/disk_sdb
[root@client ~]# lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda         8:0    0  500G  0 disk /mnt/disk_sda
sdb         8:16    0  600G  0 disk /mnt/disk_sdb
vda        253:0    0   40G  0 disk
├─vda1  253:1    0    4G  0 part
└─vda2  253:2    0   36G  0 part /
vdb        253:16   0  100G  0 disk
```

```
└─vdb1 253:17    0 100G 0 part /mnt/storage01
vdc    253:32    0 100G 0 disk
vdd    253:48    0 100G 0 disk
```

3.4 Linux 客户端 – 集群版

3.4.1 客户端配置

前置条件

- HBlock 服务器端，已经成功创建卷。
- 准备 Linux 客户端

1. 安装 Linux 客户端

注意：需要具有 root 权限才能配置 initiator。

若您客户端为 CentOS/RHEL，请安装 iscsi-initiator-utils，安装命令如下：

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

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

若您客户端为 Ubuntu/Debian，安装命令如下：

```
apt install open-iscsi
```

2. 安装 MPIO

- 对于 CentOS

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

- 对于 Ubuntu

```
apt install multipath-tools
```

3. 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）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
    }
}
```

说明: `user_friendly_names` 可以设置为 `yes`, 也可以设置为 `no`。

- `user_friendly_names yes`: 系统会使用文件 `/etc/multipath/bindings` 中的设置为多路径设备分配别名, 格式为 `mpathn` (例如 `mpatha`、`mpathb` 等)。
- `user_friendly_names no`: 系统会使用 `WWID` (全球唯一标识符) 作为多路径设备的别名。

4. 重启 multipathd 服务

- 对于 CentOS

```
systemctl restart multipathd    # CentOS
systemctl enable multipathd
```

- 对于 Ubuntu

```
systemctl restart multipath-tools.service    # Ubuntu
```

操作步骤

- HBlock 服务器端：查询要连接的 LUN 及 LUN 对应 iSCSI Target 的详细信息

```
./stor lun ls { -n | --name } LUN_NAME  
./stor target ls { -n | --name } TARGET_NAME
```

- Linux 客户端

1. 使用如下命令发现 Target IQN。

说明：如果卷对应多个 Target IQN，建议将这些 Target IQN 都连上。

```
iscsiadm -m discovery -t st -p ACTIVE_IP  
iscsiadm -m discovery -t st -p STANDBY_IP  
iscsiadm -m discovery -t st -p ColdStandby_IP
```

2. 登录 iSCSI 存储：建立多个 iSCSI 连接（按 Active Target、Standby Target、ColdStandby 顺序连接）

说明：若您的 iSCSI Target 没有开启 CHAP 认证，请直接执行步骤（4）**登录 Target**。

- (1) 开启认证

```
iscsiadm -m node -T iSCSI_TARGET_IQN -o update --name node.session.auth.authmethod --value=CHAP
```

- (2) 输入 CHAP 用户名

```
iscsiadm -m node -T iSCSI_TARGET_IQN -o update --name node.session.auth.username --value=USER
```

- (3) 输入 CHAP 密码

```
iscsiadm -m node -T iSCSI_TARGET_IQN -o update --name node.session.auth.password --value=PASSWORD
```

- (4) 登录 Target

```
iscsiadm -m node -T iSCSI_TARGET_IQN -p SERVER_IP:port -l # port 为 iSCSI 端口号
```

3. 设备显示会话情况，查看当前 iSCSI 连接。

```
iscsiadm -m session  
lsscsi #查看新增磁盘
```

4. 查看 MPIO 设备、磁盘对应的 LUN 的 WWID。

```
multipath -ll          # 可增加参数-v 3, 显示更详细的信息
ll /dev/mapper/mpathX
/lib/udev/scsi_id --whitelisted --device=/dev/sdX # 可以查看 iSCSI 磁盘对应 HBlock 卷的 WWID
```

5. 操作 MPIO 设备。

将 iSCSI 磁盘分区挂载到本地目录上, 挂载之后可以写入数据。

```
lsblk
mkfs -t ext4 /dev/mapper/mpathX      # 格式化成 ext4
mkdir DIRECTORY_NAME_OR_PATH        # 创建目录
mount /dev/mapper/mpathX DIRECTORY_NAME_OR_PATH # 将 mpathX 挂载到目录
lsblk
```

注意: 如果用户之前已经连接过此磁盘并完成了格式化, 重新连接后无需再次进行磁盘格式化, 直接挂载 iSCSI 磁盘即可。

说明: 常用的文件系统有 ext4、XFS, 具体格式化成哪种文件系统要视用户文件系统决定。

注意: 如果用户需要断开连接或者删除磁盘, 执行下列步骤:

- (1) 确保在卸载文件系统之前, 没有进程正在使用该文件的文件夹。
- (2) 使用 sync 命令来确保所有挂起的写操作都已写入磁盘。
- (3) 使用 umount 命令来正常卸载文件系统, 断开 iSCSI 连接。

```
umount DIRECTORY_NAME_OR_PATH
iscsiadm -m node -T iSCSI_TARGET_IQN -p SERVER_IP -u
```

示例

```
[root@client ~]# sync
[root@client ~]# umount /mnt/disk_mpatha
[root@client ~]# iscsiadm -m node -T iqn.2012-08.cn.ctyunapi.oos:target01.1 -p
192.168.0.102 -u
Logging out of session [sid: 2, target: iqn.2012-08.cn.ctyunapi.oos:target01.1, portal:
192.168.0.102,3260]
```

```
Logout of [sid: 2, target: iqn.2012-08.cn.ctyunapi.oos:target01.1, portal:
192.168.0.102,3260] successful.
[root@client ~]# iscsiadm -m node -T iqn.2012-08.cn.ctyunapi.oos:target01.2 -p
192.168.0.110 -u
Logging out of session [sid: 3, target: iqn.2012-08.cn.ctyunapi.oos:target01.2, portal:
192.168.0.110,3260]
Logout of [sid: 3, target: iqn.2012-08.cn.ctyunapi.oos:target01.2, portal:
192.168.0.110,3260] successful.
[root@client ~]# iscsiadm -m node -T iqn.2012-08.cn.ctyunapi.oos:target01.3 -p
192.168.0.192 -u
Logging out of session [sid: 4, target: iqn.2012-08.cn.ctyunapi.oos:target01.3, portal:
192.168.0.192,3260]
Logout of [sid: 4, target: iqn.2012-08.cn.ctyunapi.oos:target01.3, portal:
192.168.0.192,3260] successful.
```

3.4.2 配置举例

应用场景

- Linux 客户端需要连接 HBlock 集群版的卷。
- 需要连接的 HBlock 集群版的卷为 lun6a 和 lun7a，其 lun7a 有 CHAP 认证。

前置条件

- 对于需要连接 HBlock 集群版的客户端，已经按照**客户端配置**中的前置条件完成准备工作。
- 对于 HBlock 服务器端，已经成功创建卷 lun6a 和 lun7a。

操作步骤

- HBlock 服务器端：查询要连接的 LUN 及 LUN 对应 iSCSI Target 的详细信息。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun ls -n lun6a
LUN Name: lun6a (LUN 0)
Storage Mode: Cache
Capacity: 500 GiB
Status: Normal
Auto Failback: Enabled
iSCSI Target: iqn.2012-08.cn.ctyunapi.oos:target6.12(192.168.0.192:3260,Active)
               iqn.2012-08.cn.ctyunapi.oos:target6.11(192.168.0.110:3260,Standby)
               iqn.2012-08.cn.ctyunapi.oos:target6.13(192.168.0.102:3260,ColdStandby)
Create Time: 2024-05-21 14:14:48
Local Storage Class: EC 2+1+16KiB
Minimum Replica Number: 2
Redundancy Overlap: 1
Local Sector Size: 4096 bytes
Storage Pool: default
High Availability: ActiveStandby
Write Policy: WriteBack
WWID: 33ffffffffc69cbabb
UUID: lun-uuid-40731bfd-d0e5-49fb-9784-1d825635daf8
Object Storage Info:
+-----+-----+
| Provider          | OOS          |
```

```
| Bucket Name      | hblocktest3      |
| Prefix          | stor2             |
| Endpoint        | https://oos-cn.ctyunapi.cn |
| Signature Version | v2               |
| Region          |                   |
| Storage Class    | STANDARD          |
| Access Key       | cb22b08b1f9229f85874 |
| Object Size      | 1024 KiB          |
| Compression      | Enabled           |
+-----+-----+
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor target ls -n target6
Target Name: target6
Max Sessions: 2
Create Time: 2024-05-21 14:12:44
Number of Servers: 3
iSCSI Target: iqn.2012-08.cn.ctyunapi.oos:target6.11(192.168.0.110:3260)
               iqn.2012-08.cn.ctyunapi.oos:target6.12(192.168.0.192:3260)
               iqn.2012-08.cn.ctyunapi.oos:target6.13(192.168.0.102:3260)
LUN: lun6a(LUN 0)
Reclaim Policy: Retain
ServerID: hblock_1,hblock_2,hblock_3
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun ls -n lun7a
LUN Name: lun7a (LUN 0)
Storage Mode: Local
Capacity: 500 GiB
Status: Normal
Auto Failback: Enabled
iSCSI Target: iqn.2012-08.cn.ctyunapi.oos:target7.14(192.168.0.110:3260,Active)
               iqn.2012-08.cn.ctyunapi.oos:target7.15(192.168.0.192:3260,Standby)
Create Time: 2024-05-21 14:15:22
Local Storage Class: EC 2+1+16KiB
Minimum Replica Number: 2
Redundancy Overlap: 1
Local Sector Size: 4096 bytes
Storage Pool: default
High Availability: ActiveStandby
Write Policy: WriteBack
WWID: 330000000727497eb
UUID: lun-uuid-3429b79f-cd7d-47cb-9fb6-c79136deb237
Snapshot Numbers: 0
```

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor target ls -n target7
Target Name: target7
Max Sessions: 1
Create Time: 2024-05-21 14:13:27
Number of Servers: 2
iSCSI Target: iqn.2012-08.cn.ctyunapi.oos:target7.14(192.168.0.110:3260)
               iqn.2012-08.cn.ctyunapi.oos:target7.15(192.168.0.192:3260)
LUN: lun7a(LUN 0)
Reclaim Policy: Retain
CHAP: test2,T12345678912,Enabled
ServerID: hblock_1,hblock_2
```

- Linux 客户端:

1. 发现 lun6a 和 lun7a 的 Target:

```
[root@client ~]# iscsiadm -m discovery -t st -p 192.168.0.110
192.168.0.110:3260,1 iqn.2012-08.cn.ctyunapi.oos:target7.14
192.168.0.110:3260,1 iqn.2012-08.cn.ctyunapi.oos:target02.3
192.168.0.110:3260,1 iqn.2012-08.cn.ctyunapi.oos:target04.7
192.168.0.110:3260,1 iqn.2012-08.cn.ctyunapi.oos:target6.11
[root@client ~]# iscsiadm -m discovery -t st -p 192.168.0.192
192.168.0.192:3260,1 iqn.2012-08.cn.ctyunapi.oos:target7.15
192.168.0.192:3260,1 iqn.2012-08.cn.ctyunapi.oos:target6.12
192.168.0.192:3260,1 iqn.2012-08.cn.ctyunapi.oos:test.10
192.168.0.192:3260,1 iqn.2012-08.cn.ctyunapi.oos:target04.8
[root@client ~]# iscsiadm -m discovery -t st -p 192.168.0.102
192.168.0.102:3260,1 iqn.2012-08.cn.ctyunapi.oos:target02.4
192.168.0.102:3260,1 iqn.2012-08.cn.ctyunapi.oos:target6.13
192.168.0.102:3260,1 iqn.2012-08.cn.ctyunapi.oos:test.9
```

2. 登录 iSCSI 存储

- 登录 lun6a 的 iSCSI 存储（按 Active Target、Standby Target、ColdStandby 顺序连接）：

```
[root@client ~]# iscsiadm -m node -T iqn.2012-08.cn.ctyunapi.oos:target6.12 -p
192.168.0.192:3260 -l
```

```
Logging in to [iface: default, target: iqn.2012-08.cn.ctyunapi.oos:target6.12, portal:
192.168.0.192,3260] (multiple)
Login to [iface: default, target: iqn.2012-08.cn.ctyunapi.oos:target6.12, portal:
192.168.0.192,3260] successful.
[root@client ~]# iscsiadm -m node -T iqn.2012-08.cn.ctyunapi.oos:target6.11 -p
192.168.0.110:3260 -l
Logging in to [iface: default, target: iqn.2012-08.cn.ctyunapi.oos:target6.11, portal:
192.168.0.110,3260] (multiple)
Login to [iface: default, target: iqn.2012-08.cn.ctyunapi.oos:target6.11, portal:
192.168.0.110,3260] successful.
[root@client ~]# iscsiadm -m node -T iqn.2012-08.cn.ctyunapi.oos:target6.13 -p
192.168.0.102:3260 -l
Logging in to [iface: default, target: iqn.2012-08.cn.ctyunapi.oos:target6.13, portal:
192.168.0.102,3260] (multiple)
Login to [iface: default, target: iqn.2012-08.cn.ctyunapi.oos:target6.13, portal:
192.168.0.102,3260] successful.
```

- 登录 lun7a 的 iSCSI 存储，需要进行 CHAP 认证。

```
[root@client ~]# iscsiadm -m node -T iqn.2012-08.cn.ctyunapi.oos:target7.14 -o update -
-name node.session.auth.authmethod --value=CHAP
[root@client ~]# iscsiadm -m node -T iqn.2012-08.cn.ctyunapi.oos:target7.14 -o update -
-name node.session.auth.username --value=test2
[root@client ~]# iscsiadm -m node -T iqn.2012-08.cn.ctyunapi.oos:target7.14 -o update -
-name node.session.auth.password --value=*****
[root@client ~]# iscsiadm -m node -T iqn.2012-08.cn.ctyunapi.oos:target7.14 -p
192.168.0.110:3260 -l
Logging in to [iface: default, target: iqn.2012-08.cn.ctyunapi.oos:target7.14, portal:
192.168.0.110,3260] (multiple)
Login to [iface: default, target: iqn.2012-08.cn.ctyunapi.oos:target7.14, portal:
192.168.0.110,3260] successful.
[root@client ~]# iscsiadm -m node -T iqn.2012-08.cn.ctyunapi.oos:target7.15 -o update -
-name node.session.auth.authmethod --value=CHAP
[root@client ~]# iscsiadm -m node -T iqn.2012-08.cn.ctyunapi.oos:target7.15 -o update -
-name node.session.auth.username --value=test2
[root@client ~]# iscsiadm -m node -T iqn.2012-08.cn.ctyunapi.oos:target7.15 -o update -
-name node.session.auth.password --value=*****
```

```
[root@client ~]# iscsiadm -m node -T iqn.2012-08.cn.ctyunapi.oos:target7.15 -p
192.168.0.192:3260 -l
Logging in to [iface: default, target: iqn.2012-08.cn.ctyunapi.oos:target7.15, portal:
192.168.0.192,3260] (multiple)
Login to [iface: default, target: iqn.2012-08.cn.ctyunapi.oos:target7.15, portal:
192.168.0.192,3260] successful.
```

3. 显示会话情况，查看当前 iSCSI 连接。

```
[root@client ~]# iscsiadm -m session
tcp: [3] 192.168.0.192:3260,1 iqn.2012-08.cn.ctyunapi.oos:target6.12 (non-flash)
tcp: [4] 192.168.0.110:3260,1 iqn.2012-08.cn.ctyunapi.oos:target6.11 (non-flash)
tcp: [5] 192.168.0.102:3260,1 iqn.2012-08.cn.ctyunapi.oos:target6.13 (non-flash)
tcp: [6] 192.168.0.110:3260,1 iqn.2012-08.cn.ctyunapi.oos:target7.14 (non-flash)
tcp: [7] 192.168.0.192:3260,1 iqn.2012-08.cn.ctyunapi.oos:target7.15 (non-flash)
[root@client ~]# lsscsi
[4:0:0:0]    disk      CTYUN      iSCSI LUN Device 1.00  /dev/sdc
[5:0:0:0]    disk      CTYUN      iSCSI LUN Device 1.00  /dev/sdd
[6:0:0:0]    disk      CTYUN      iSCSI LUN Device 1.00  /dev/sde
[7:0:0:0]    disk      CTYUN      iSCSI LUN Device 1.00  /dev/sdf
[8:0:0:0]    disk      CTYUN      iSCSI LUN Device 1.00  /dev/sdg
```

4. 查看 MPIO、磁盘对应的 LUN 的 WWID。

```
[root@client ~]# multipath -ll
mpathc (0x3000000727497eb) dm-1 CTYUN      ,iSCSI LUN Device
size=500G features='1 queue_if_no_path' hwhandler='1 alua' wp=rw
|-+- policy='round-robin 0' prio=50 status=active
|  `- 7:0:0:0 sdf 8:80 active ready running
`-+- policy='round-robin 0' prio=1 status=enabled
   `- 8:0:0:0 sdg 8:96 active ghost running
mpathb (0x3ffffffffffc69cbabb) dm-0 CTYUN      ,iSCSI LUN Device
size=500G features='1 queue_if_no_path' hwhandler='1 alua' wp=rw
|-+- policy='round-robin 0' prio=50 status=active
|  `- 4:0:0:0 sdc 8:32 active ready running
|-+- policy='round-robin 0' prio=1 status=enabled
```

```
| ` - 5:0:0:0 sdd 8:48 active ghost running
`-+- policy='round-robin 0' prio=0 status=enabled
` - 6:0:0:0 sde 8:64 failed faulty running
[root@client ~]# ll /dev/mapper/mpathc
lrwxrwxrwx 1 root root 7 May 21 15:03 /dev/mapper/mpathc -> ../dm-1
[root@client ~]# ll /dev/mapper/mpathb
lrwxrwxrwx 1 root root 7 May 21 14:57 /dev/mapper/mpathb -> ../dm-0
[root@client ~]# # /lib/udev/scsi_id --whitelisted --device=/dev/sdc
33ffffffffc69cbabb
[root@client ~]# # /lib/udev/scsi_id --whitelisted --device=/dev/sdd
33ffffffffc69cbabb
[root@client ~]# # /lib/udev/scsi_id --whitelisted --device=/dev/sde
33ffffffffc69cbabb
[root@client ~]# # /lib/udev/scsi_id --whitelisted --device=/dev/sdf
330000000727497eb
[root@client ~]# # /lib/udev/scsi_id --whitelisted --device=/dev/sdg
330000000727497eb
```

说明：可以看出/dev/mapper/mpathb（/dev/sdc、/dev/sdd、/dev/sde）对应 HBlock 卷 lun6a（卷 WWID 为 33ffffffffc69cbabb），/dev/mapper/mpathc（/dev/sdf、/dev/sdg）对应 HBlock 卷 lun7a（卷 WWID 为 330000000727497eb）。

5. 操作 MPIO 设备。

将 iSCSI 磁盘分区挂载到本地目录上，挂载之后可以写入数据。

● 挂载 iSCSI 磁盘/dev/mapper/mpathb

```
[root@client ~]# lsblk
sdc          8:32    0 500G  0 disk
└─mpathb 252:0    0 500G  0 mpath
sdd          8:48    0 500G  0 disk
└─mpathb 252:0    0 500G  0 mpath
sde          8:64    0 500G  0 disk
└─mpathb 252:0    0 500G  0 mpath
sdf          8:80    0 500G  0 disk
└─mpathc 252:1    0 500G  0 mpath
```

```
sdg      8:96   0  500G  0 disk
└─mpathc 252:1   0   500G  0 mpath
vda      253:0   0   40G  0 disk
├─vda1   253:1   0    4G  0 part
└─vda2   253:2   0   36G  0 part  /
vdb      253:16  0  100G  0 disk
└─vdb1   253:17  0  100G  0 part  /mnt/storage01
vdc      253:32  0  100G  0 disk
vdd      253:48  0  100G  0 disk

[root@client ~]# mkfs -t ext4 /dev/mapper/mpathb
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
32768000 inodes, 131072000 blocks
6553600 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2279604224
4000 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424, 20480000, 23887872, 71663616, 78675968,
    102400000

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

[root@client ~]# mkdir /mnt/disk_mpathb
[root@client ~]# mount /dev/mapper/mpathb /mnt/disk_mpathb
```

```
[root@client ~]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
sdc          8:32   0  500G  0 disk
└─mpathb 252:0    0  500G  0 mpath /mnt/disk_mpathb
sdd          8:48   0  500G  0 disk
└─mpathb 252:0    0  500G  0 mpath /mnt/disk_mpathb
sde          8:64   0  500G  0 disk
└─mpathb 252:0    0  500G  0 mpath /mnt/disk_mpathb
sdf          8:80   0  500G  0 disk
└─mpathc 252:1    0  500G  0 mpath
sdg          8:96   0  500G  0 disk
└─mpathc 252:1    0  500G  0 mpath
vda         253:0    0   40G  0 disk
├─vda1  253:1    0    4G  0 part
└─vda2  253:2    0   36G  0 part /
vdb         253:16   0  100G  0 disk
└─vdb1  253:17   0  100G  0 part /mnt/storage01
vdc         253:32   0  100G  0 disk
vdd         253:48   0  100G  0 disk
```

● 挂载 iSCSI 磁盘/dev/mapper/mpathc

```
[root@client ~]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
sdc          8:32   0  500G  0 disk
└─mpathb 252:0    0  500G  0 mpath /mnt/disk_mpathb
sdd          8:48   0  500G  0 disk
└─mpathb 252:0    0  500G  0 mpath /mnt/disk_mpathb
sde          8:64   0  500G  0 disk
└─mpathb 252:0    0  500G  0 mpath /mnt/disk_mpathb
sdf          8:80   0  500G  0 disk
└─mpathc 252:1    0  500G  0 mpath
sdg          8:96   0  500G  0 disk
└─mpathc 252:1    0  500G  0 mpath
vda         253:0    0   40G  0 disk
```

```
|—vda1   253:1    0    4G  0 part
|—vda2   253:2    0   36G  0 part  /
vdb      253:16   0  100G  0 disk
|—vdb1   253:17   0  100G  0 part  /mnt/storage01
vdc      253:32   0  100G  0 disk
vdd      253:48   0  100G  0 disk

[root@client ~]# mkfs -t ext4 /dev/mapper/mpathc
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
32768000 inodes, 131072000 blocks
6553600 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2279604224
4000 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424, 20480000, 23887872, 71663616, 78675968,
    102400000

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

[root@client ~]# mkdir /mnt/disk_mpathc
[root@client ~]# mount /dev/mapper/mpathc /mnt/disk_mpathc
[root@client ~]# lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
sdc        8:32   0  500G  0 disk
```

```
└─mpathb 252:0    0  500G  0 mpath /mnt/disk_mpathb
sdd          8:48    0  500G  0 disk
└─mpathb 252:0    0  500G  0 mpath /mnt/disk_mpathb
sde          8:64    0  500G  0 disk
└─mpathb 252:0    0  500G  0 mpath /mnt/disk_mpathb
sdf          8:80    0  500G  0 disk
└─mpathc 252:1    0  500G  0 mpath /mnt/disk_mpathc
sdg          8:96    0  500G  0 disk
└─mpathc 252:1    0  500G  0 mpath /mnt/disk_mpathc
vda         253:0    0   40G  0 disk
├─vda1     253:1    0    4G  0 part
└─vda2     253:2    0   36G  0 part  /
vdb         253:16   0  100G  0 disk
└─vdb1     253:17   0  100G  0 part  /mnt/storage01
vdc         253:32   0  100G  0 disk
vdd         253:48   0  100G  0 disk
```

4 管理操作

4.1 管理操作命令行格式说明

在本手册中可能出现下列命令行格式形式，它们的意义如下：

格式	描述
粗体	命令行关键字。
<i>斜体</i>	命令行可变参数，即需要用实际值替代。
[]	表示 [] 括起来的部分，配置时可选。
{A B ...}	表示必须从 { } 括起来的关键字或参数中选取一个。
[A B ...]	表示从 [] 括起来的关键字或参数中选取一个，或者不选。
&<1-n>	表示符号&前面的参数可以重复输入 1-n 次，以英文逗号 (,) 分隔。

说明：后续命令行以 x86 服务器举例，ARM 服务器、龙芯服务器的命令与 x86 服务器的命令相同。

4.2 帮助命令

```
./stor { --help | -h }
```

此命令用来查看 HBlock 的命令帮助信息。

示例

- 查看 HBlock 整体命令。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor --help
Usage:
  stor <command> [options]
  stor <command> <subcommand> [options]

Type 'stor --help <command>' or 'stor --help <command> <subcommand>' to print help on
a specific subcommand.
Type 'stor --version' to print HBlock version.

Available commands:
  alarm          View and manage alarms.
  config         Configure HBlock.
  conssnap       Manage consistency snapshot(s).
  event          Event information.
  info           Get HBlock information.
  install        Install HBlock.
  license        Manage HBlock license(s).
  logcollect     Log collection information.
  lun            Manage LUN(s).
  monitor        Get performance data.
  restart        Restart HBlock.
  server         Manage HBlock server(s).
  setup          Initialize HBlock.
  snapshot       Manage snapshot(s).
  start          Start HBlock.
  status         Get HBlock status.
  stop           Stop HBlock.
```

storagepool	Manage storage pools.
target	Manage Target(s).
topology	Manage topology.
tuning	Adjusting parameters to fit different scenarios.
uninstall	Uninstall HBlock on all servers.
upgrade	Upgrade HBlock.

General subcommands:

add	Add an entity, such as adding a LUN or server.
ls	List entities or the specified entity, such as listing servers or the specified server.
rm	Remove an entity, such as removing a LUN or server.
set	Set entity properties, such as setting LUN or server properties.

Available subcommands for alarm:

E[export]	Export alarm records to a file.
M[mute]	Set the mute status of a specified alarm record to Muted status, stop receiving its alarm emails and not list it by default.
UM[unmute]	Set the mute status of a specified alarm record to Normal status.
R[resolve]	Set a specified alarm record to Resolved status.

Available subcommands for conssnap:

R[rollback]	Rollback consistency snapshot.
-------------	--------------------------------

Available subcommands for lun:

C[clone]	Create clone LUN.
F[flatten]	Flatten clone LUN.
RC[recover]	Recover the specified LUN from the cloud.
RS[resume]	Resume recovery of a previously failed LUN.
S[switch]	Switch between active and standby iSCSI Targets.
X[expand]	Expand LUN.

Available subcommands for monitor:

```
E[export]          Export historical performance data to a file.
V[view]           Get real-time performance data.
```

Available subcommands for server:

```
A[addpath]        Add path(s) to store data.
R[rmpath]         Remove path(s).
S[setpath]        Set path(s) properties.
```

Available subcommands for snapshot:

```
R[rollback]       Rollback snapshot or a snapshot in consistency snapshot.
```

Available subcommands for storagepool:

```
A[addnode]        Add node(s) to storage pool.
R[rmnode]         Remove node(s) from storage pool.
```

Available subcommands for upgrade:

```
T[status]
```

● 查看卷的命令。

```
[root@hbblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun --help
```

lun: Manage LUN(s).

Usage: lun <subcommand> [options]

Type 'stor --help lun <subcommand>' to print help on a specific subcommand.

Available subcommands:

```
add              Create LUN.
C[clone]         Create clone LUN.
F[flatten]       Flatten clone LUN.
ls              List all iSCSI LUNs or the specified LUN.
RC[recover]      Recover the specified LUN from the cloud.
RS[resume]       Resume recovery of a previously failed LUN.
prefer          Set server affinity of active and standby iSCSI
```

Targets for LUN.

```
rm              Remove LUN.
```

S[switch]	Switch between active and standby iSCSI Targets.
set	Set LUN properties.
X[expand]	Expand LUN.

4.3 安装

```
./stor install [ { -a | --api-port } API_PORT ] [ { -w | --web-port } WEB_PORT ]
```

此命令用来安装 HBlock。

注意：

- 请确保 Linux 用户具有所需要端口的权限。Linux 系统默认小于 1024 的端口不对没有 root 权限的 Linux 普通用户开放。
- 安装 HBlock 和执行 HBlock 管理操作的应该属于同一用户。

说明：如果服务器未安装字体库，执行安装命令后会显示相应的提示信息。此外，服务器未安装字体库时，通过 WEB 登录会导致验证码无法显示，从而无法完成登录操作。

参数

参数	描述
-a API_PORT 或 --api-port API_PORT	指定 API 端口号。 取值：整型，取值范围是[1, 65535]，默认端口号是 1443。
-w WEB_PORT 或 --web-port WEB_PORT	指定 web 端口号。 取值：整型，取值范围是[1, 65535]，默认端口号是 2443。

示例

安装 HBlock。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor install
Do you agree with HBlock User Agreement? [Yes/No]
Used in Chinese mainland, follow https://www.ctyun.cn/portal/protocol/10073150
Otherwise, follow https://www.esurfingcloud.com/portal/protocol/20692906
y
Installing HBlock...
Installed successfully.
```

When all servers are installed, please initialize HBlock in any of the following ways:

1. Use web portal to initialize HBlock. The https port is 2443.
2. Use management API (POST /rest/v1/system/setup) to initialize HBlock. The https port is 1443.
3. Use command line (stor setup) to initialize HBlock. Type 'stor --help setup' for more information.

4.4 初始化

单机版：

```
./stor setup { -n | --stor-name } STOR_NAME [ { -u | --user-name } USER_NAME ] { -p | -  
-password } PASSWORD { -s | --server } { SERVER_IP[:PORT]:PATH &<1-n> } [ { -P | --  
public-network } CIDR ] [ --iscsi-port ISCSI_PORT ] [--port-range PORT1-PORT2 ] [ --  
management-port1 MANAGEMENT_PORT1 ] [ --management-port2 MANAGEMENT_PORT2 ] [ --  
management-port3 MANAGEMENT_PORT3 ] [ --management-port4 MANAGEMENT_PORT4 ] [ --  
management-port6 MANAGEMENT_PORT6 ]
```

集群版：

```
./stor setup { -n | --stor-name } STOR_NAME [ { -u | --user-name } USER_NAME ] { -p | -  
-password } PASSWORD { { -s | --server } { SERVER_IP[:PORT][:PATH] &<1-n> } &<1-n> | --  
topology-file TOPOLOGY_FILE } [ {-C | --cluster-network } CIDR ] [ { -P | --public-  
network } CIDR ] [ --fault-domain FAULT_DOMAIN ] [ --iscsi-port ISCSI_PORT ] [--port-  
range PORT1-PORT2 ] [ --data-port1 DATA_PORT1 ] [ --management-port1 MANAGEMENT_PORT1 ]  
[ --management-port2 MANAGEMENT_PORT2 ] [ --management-port3 MANAGEMENT_PORT3 ] [ --  
management-port4 MANAGEMENT_PORT4 ] [ --management-port5 MANAGEMENT_PORT5 ] [ --  
management-port6 MANAGEMENT_PORT6 ] [ --metadata-port1 METADATA_PORT1 ] [ --metadata-  
port2 METADATA_PORT2 ] [ --metadata-port3 METADATA_PORT3 ] [ --metadata-port4  
METADATA_PORT4 ] [ --metadata-port5 METADATA_PORT5 ] [ --metadata-port6 METADATA_PORT6 ]  
[ --metadata-port7 METADATA_PORT7 ] [ --metadata-port8 METADATA_PORT8 ] [ --cs  
SERVER_IP[:DIR],SERVER_IP[:DIR],SERVER_IP[:DIR] ] [ --mdm  
SERVER_IP[:DIR],SERVER_IP[:DIR] ] [ --ls  
SERVER_IP[:DIR],SERVER_IP[:DIR],SERVER_IP[:DIR] ]
```

此命令用来初始化 HBlock。

说明：对于集群版，在每台服务器上安装 HBlock 后，可以在集群内的任一服务器上进行初始化。

注意：

- 请确保 Linux 用户具有所需要端口的权限。Linux 系统默认小于 1024 的端口不对没有 root 权限的 Linux 普通用户开放。
- 设置端口范围 (`--port-range PORT1-PORT2`) 时，请避免和 Linux 系统的本地临时端口 (`ip_local_port_range`) 范围重合，否则可能会导致 HBlock 服务所用的端口被占用。使用命令行 `cat /proc/sys/net/ipv4/ip_local_port_range` 可以查看本地临时端口范围。

参数

参数	描述
<code>-n STOR_NAME</code> 或 <code>--stor-name STOR_NAME</code>	指定 HBlock 名称。 取值：字符串形式，长度范围是 1~64，可以包含字母、数字、下划线 (<code>_</code>) 和短横线 (<code>-</code>)，字母区分大小写，且仅支持以字母或数字开头。
<code>-u USER_NAME</code> 或 <code>--user-name USER_NAME</code>	HBlock 的管理员用户名。 取值：字符串形式，长度范围是 5~16，只能由数字和字母组成，字母区分大小。默认值为 <code>storuser</code> 。
<code>-p PASSWORD</code> 或 <code>--password PASSWORD</code>	设置新的管理员密码。初始化时必须修改密码。 取值：字符串形式，长度范围 8~16，至少包含以下字符中的 3 种：大写字母、小写字母、数字、特殊字符 (<code>~!@#\$%^&*()_+[]{} ;:.,/<>?</code>)，区分大小写。不能包含：3 个连续重复的字符，3 个连续或反序的数字、或字母（不区分大小写），3 个连续或反序的键盘序列（不区分大小写）。
<code>-s { SERVER_IP[:PORT][:PATH]}</code>	指定要初始化的服务器 IP (<code>SERVER_IP</code>)、API

<pre>&<1-n> } &<1-n>或--server { SERVER_IP[:PORT][:PATH] &<1- n> } &<1-n></pre>	端口号 (<i>PORT</i>) 和数据目录 (<i>PATH</i>)。 取值： ● 服务器 IP：一个服务器 IP 可以对应多个数据目录，用英文逗号 (,) 隔开。 ■ 对于单机版，只需要一台服务器即可。 ■ 对于集群版，至少需要三台服务器。 ● API 端口号：取值范围是[1, 65535]，默认值为 1443。需要和该服务器安装 HBlock 时设置的 API 端口号保持一致。 ● 数据目录：数据目录名不能含逗号。可以设置多个数据目录，以英文逗号 (,) 隔开。数据目录用于存储数据，建议不要与操作系统共用磁盘或文件系统。 说明：对于单机版，第一个数据目录是默认数据目录。对于集群版，需要至少有一个数据目录。
<pre>--topology-file TOPOLOGY_FILE</pre>	导入集群拓扑文件（仅集群版支持）。 拓扑文件为符合 UTF-8 编码格式的 JSON 文件，详见集群拓扑文件。
<pre>-C CIDR 或--cluster-network CIDR</pre>	集群网络（仅集群版支持），用于集群间的数据通信。IP CIDR 格式。 ● 如果指定集群网络，为了保证 HBlock 的各个服务器之间能够正常通信，请确保每个服务器上都有与指定网段相符的 IP，系统会自动选取该 IP 进行通信。 ● 如果未指定集群网络，默认使用服务器列表中 HBlock 服务器 IP，此时 HBlock 的服务器

	IP 不能指定为 localhost、127.0.0.1 或 0:0:0:0:0:0:0:1。
-P CIDR 或 --public-network CIDR	业务网络，用于客户端和服务端之间的数据传输。IP CIDR 格式。 ● 如果指定业务网络，请确保每个服务器上都有与指定网段相符的 IP，系统会自动选取该 IP 和客户端进行通信。 ● 如果未指定业务网络，或指定了业务网络，但指定的网段与服务器所有 IP 均不相符，默认使用服务器列表中 HBlock 服务器 IP，此时 HBlock 服务器 IP 不能指定为 localhost、127.0.0.1 或 0:0:0:0:0:0:0:1。
--fault-domain FAULT_DOMAIN	设置基础存储池的故障域级别（仅集群版支持）。初始化时设置的集群拓扑中，包含的数据目录节点都加入到基础存储池中。 取值： ● path：数据目录级别。 ● server：服务器级别。 ● rack：机架级别。 ● room：机房级别。 默认值为 server。 注意：如果故障域级别为 room 或者 rack，则必须使用拓扑文件导入方式进行初始化。
--iscsi-port ISCSI_PORT	指定 iSCSI 端口号。 取值：整型，取值为[1, 65535]，默认端口为 3260。
--port-range PORT1-PORT2	指定端口范围。存储服务以及未指定端口的服务

	将从此范围中自动取值。 取值：整型，取值范围为[1, 65535]，<i>PORT1</i> 为端口范围最小值，<i>PORT2</i> 为端口范围最大值，且 <i>PORT1</i><<i>PORT2</i>。<i>PORT1</i> 默认取值为 20000，<i>PORT2</i> 默认取值为 20500。 说明：建议指定的端口范围至少包含 500 个端口。
--data-port1 <i>DATA_PORT1</i>	指定数据端口 1（仅集群版支持）。 取值：整型，取值范围为[1, 65535]。
--management-port1 <i>MANAGEMENT_PORT1</i>	指定管理服务端口 1。 取值：整型，取值范围为[1, 65535]。
--management-port2 <i>MANAGEMENT_PORT2</i>	指定管理服务端口 2。 取值：整型，取值范围为[1, 65535]。
--management-port3 <i>MANAGEMENT_PORT3</i>	指定管理服务端口 3。 取值：整型，取值范围为[1, 65535]。
--management-port4 <i>MANAGEMENT_PORT4</i>	指定管理服务端口 4。 取值：整型，取值范围为[1, 65535]。
--management-port5 <i>MANAGEMENT_PORT5</i>	指定管理服务端口 5（仅集群版支持）。 取值：整型，取值范围为[1, 65535]。
--management-port6 <i>MANAGEMENT_PORT6</i>	指定管理服务端口 6。 取值：整型，取值范围为[1, 65535]。
--metadata-port1 <i>METADATA_PORT1</i>	指定元数据端口 1（仅集群版支持）。 取值：整型，取值范围为[1, 65535]。
--metadata-port2 <i>METADATA_PORT2</i>	指定元数据端口 2（仅集群版支持）。 取值：整型，取值范围为[1, 65535]。
--metadata-port3 <i>METADATA_PORT3</i>	指定元数据端口 3（仅集群版支持）。 取值：整型，取值范围为[1, 65535]。

<code>--metadata-port4 METADATA_PORT4</code>	指定元数据端口 4（仅集群版支持）。 取值：整型，取值范围为[1, 65535]。
<code>--metadata-port5 METADATA_PORT5</code>	指定元数据端口 5（仅集群版支持）。 取值：整型，取值范围为[1, 65535]。
<code>--metadata-port6 METADATA_PORT6</code>	指定元数据端口 6（仅集群版支持）。 取值：整型，取值范围为[1, 65535]。
<code>--metadata-port7 METADATA_PORT7</code>	指定元数据端口 7（仅集群版支持）。 取值：整型，取值范围为[1, 65535]。
<code>--metadata-port8 METADATA_PORT8</code>	指定元数据端口 8（仅集群版支持）。 取值：整型，取值范围为[1, 65535]。
--cs <code>SERVER_IP[:DIR],SERVER_IP[:DIR], SERVER_IP[:DIR]</code>	指定协调服务的地址和存储协调服务数据的数据目录（仅集群版支持）。 如果指定协调服务的地址，必须同时指定集群中的 3 个 IP 地址。 说明： 为了提升读写性能，建议协调服务的数据目录与安装目录、存储数据的数据目录相互独立。 取值： ● <code>SERVER_IP</code>（协调服务的地址）：IPv4 或 [IPv6]地址。 ● <code>DIR</code>（协调服务的数据目录）：数据目录名不能含逗号。默认目录为安装目录。
--mdm <code>SERVER_IP[:DIR],SERVER_IP[:DIR]</code>	指定元数据管理服务的地址和存储元数据管理服务数据的数据目录（仅集群版支持）。 如果指定元数据管理服务的地址，必须同时指定集群中的 2 个 IP 地址。 说明： 为了提升读写性能，建议元数据服务的数

	据目录与安装目录、存储数据的数据目录相互独立。 取值： ● SERVER_IP（元数据管理服务的地址）：IPv4 或[IPv6]地址。 ● DIR（元数据管理服务的数据目录）：数据目录名不能含逗号。默认目录为安装目录。
--ls SERVER_IP[:DIR],SERVER_IP[:DIR], SERVER_IP[:DIR]	指定日志服务的地址和存储日志服务数据的数据目录（仅集群版支持）。 如果指定日志服务的地址，必须同时指定集群中的 3 个 IP 地址。 说明：为了提升读写性能，建议日志服务的数据目录与安装目录、存储数据的数据目录相互独立。 取值： ● SERVER_IP（日志服务的地址）：IPv4 或[IPv6]地址。 ● DIR（日志服务的数据目录）：数据目录名不能含逗号。默认目录为安装目录。

示例

- 初始化 HBlock（单机版）。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor setup -n test -s
192.168.0.69:/mnt/storage01,/mnt/storage02
Please enter password:
*****
Start to setup HBlock, please wait.
Processing...
```

```
Setup successfully and the HBlock services have been started.
Welcome to HBlock!
You are using a 30-day trial version. Please follow the steps to get a license.
1. Run "stor info --serial-id" to get the serial ID of the HBlock
2. Contact the software vendor to obtain a license
3. Run "stor license add -k KEY" to import the license

Type 'stor --help' to get more information, such as managing LUNs, targets, servers,
etc.
```

● 初始化 HBlock（集群版）：集群拓扑使用服务器方式。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor setup -n test -s
192.168.0.209:/mnt/storage01,/mnt/storage02 192.168.0.121:/mnt/storage01
192.168.0.72:/mnt/storage01
Please enter password:
*****
Start to setup HBlock, please wait.
Processing...
Setup successfully and the HBlock services have been started.
Welcome to HBlock!
You are using a 30-day trial version. Please follow the steps to get a license.
1. Run "stor info --serial-id" to get the serial ID of the HBlock
2. Contact the software vendor to obtain a license
3. Run "stor license add -k KEY" to import the license

Type 'stor --help' to get more information, such as managing LUNs, targets, servers,
etc.
```

● 初始化 HBlock（集群版）：集群拓扑使用拓扑文件导入方式。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor setup -n stor -p ***** --
topology-file /mnt/storage01/topology.txt --fault-domain path
Start to setup HBlock, please wait.
Processing...
Setup successfully and the HBlock services have been started.
```

```
Welcome to HBlock!

You are using a 30-day trial version. Please follow the steps to get a license.

1. Run "stor info --serial-id" to get the serial ID of the HBlock
2. Contact the software vendor to obtain a license
3. Run "stor license add -k KEY" to import the license

Type 'stor --help' to get more information, such as managing LUNs, targets, servers,
etc.
```

拓扑文件源码如下:

```
{
  "name": "default",
  "childNodes": [
    {
      "name": "room1",
      "type": "room",
      "childNodes": [
        {
          "type": "server",
          "name": "server1",
          "ip": "192.168.0.192",
          "apiPort": 1443,
          "childNodes": [
            {
              "name": "/mnt/stor",
              "type": "path"
            },
            {
              "name": "/mnt/storage01",
              "type": "path"
            }
          ]
        }
      ]
    }
  ]
}
```

```
        "type": "server",
        "name": "server2",
        "ip": "192.168.0.110",
        "apiPort": 1443,
        "childNodes": [
            {
                "name": "/mnt/stor",
                "type": "path"
            }
        ]
    },
    {
        "type": "server",
        "name": "server3",
        "ip": "192.168.0.102",
        "apiPort": 1443,
        "childNodes": [
            {
                "name": "/mnt/stor",
                "type": "path"
            }
        ]
    }
]
}
```

4.5 软件许可证

4.5.1 加载软件许可证

```
./stor license add { -k | --key } KEY
```

此命令用来加载软件许可证。

说明：

- HBlock 软件初始化后提供 30 天试用期，如确定使用，建议尽快联系软件供应商获取正式版软件许可证，并加载入系统。
- 可以通过 `./stor info --serial-id` 获取 serial ID，将该 ID 提供给软件供应商用于获取软件许可证。
- 试用期过期，或软件许可证过期，均无法再使用管理功能。

参数

参数	描述
<code>-k KEY</code> 或 <code>--key KEY</code>	加载软件许可证。

示例

加载软件许可证。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor license add -k *****
Import license successfully.
The current system time: 2024-07-22 14:48:34
```

LicenseId	Account	Type	Status	MaximumLocalCapacity
ehc2b6a9-f3fb-4098-a6b3-3652a5d71232	testaccount@ctyun.cn	Subscription	Effective	1 PiB

```
License ehc2b6a9-f3fb-4098-a6b3-3652a5d71232 (Effective):
Usage:
```

MaximumLocalCapacity	EffectiveTime	ExpireTime	Status
1 PiB			Effective

1 PiB	2024-06-22 14:42:08	2024-08-21 14:42:08	Effective	
+-----+	+-----+	+-----+	+-----+	+

4.5.2 查看软件许可证

```
./stor license ls [ -r | --record ] [ -u | --usage ] [ { -n | --license }  
LICENSE_ID ]
```

此命令用来查看软件许可证信息。

参数

参数	描述
-r 或 --record	查看软件许可证购买信息。
-u 或 --usage	查看软件许可证使用信息。
-n LICENSE_ID 或 --license LICENSE_ID	查看指定 LICENSE_ID 的软件许可证信息。

示例

查询软件许可证信息。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor license ls  
License List:  
  
+-----+-----+-----+-----+  
| LicenseId | Type | Status | MaximumLocalCapacity |  
+-----+-----+-----+-----+  
| ehc2b6a9-f3fb-4098-a6b3-3652a5d76269 | Subscription | Invalid | 2 PiB |  
| qws2b6a9-f3fb-4098-a6b3-3652a5d76530 | Perpetual | Effective | 2 PiB |  
+-----+-----+-----+-----+  
  
License ehc2b6a9-f3fb-4098-a6b3-3652a5d76269 (Invalid):  
Record:  
  
+-----+-----+-----+-----+-----+-----+  
| PurchaseTime | Operation | LocalCapacity | EffectiveTime | ExpireTime | Status |  
+-----+-----+-----+-----+-----+-----+  
| 2022-04-21 16:16:15 | New | 2 PiB | 2022-0-21 16:16:15 | 2022-09-19 16:16:15 | Effective |  
+-----+-----+-----+-----+-----+-----+
```

Usage:

MaximumLocalCapacity	EffectiveTime	ExpireTime	Status
2 PiB	2022-04-21 16:16:15	2022-09-19 16:16:15	Effective

License qws2b6a9-f3fb-4098-a6b3-3652a5d76530 (Effective):

Record:

PurchaseTime	Operation	LocalCapacity	MaintenanceEffectiveTime	MaintenanceExpireTime	Status
2022-04-21 16:16:41	New	2 PiB	2022-04-21 16:16:41	2022-09-19 16:16:41	Effective

Usage:

MaximumLocalCapacity	MaintenanceEffectiveTime	MaintenanceExpireTime	Status
2 PiB	2022-04-21 16:16:41	2022-09-19 16:16:41	Effective

软件许可证信息描述

项目	描述
License List: 软件许可证列表。	
LicenseId	软件许可证 ID。
Type	软件许可证订购类型： - Subscription: 订阅模式。 - Perpetual: 永久许可模式。
Status	软件许可证状态： - Effective: 软件许可证已生效。 - Expired: 软件许可证已过期。

	● Invalid: 软件许可证无效。
MaximumLocalCapacity	允许的本地卷总容量。 “-”表示未限制本地卷容量。
Record: 软件许可证购买信息。	
PurchaseTime	软件许可证购买时间。
Operation	软件许可证的购买记录: ● New: 首次购买。 ● Expand: 扩容。 ● Renew: 续订/续保。
LocalCapacity	本次购买软件许可证, 容许本地卷的容量。
EffectiveTime	对于订阅模式软件许可证, 软件许可证生效时间。
ExpireTime	对于订阅模式软件许可证, 软件许可证过期时间。
MaintenanceEffectiveTime	对于永久许可模式软件许可证, 软件许可证维保生效时间。
MaintenanceExpireTime	对于永久许可模式软件许可证, 软件许可证维保过期时间。
Status	软件许可证状态: ● Effective: 软件许可证已生效。 ● Expired: 软件许可证已过期。 ● NotStart: 软件许可证未生效。
Usage: 软件许可证使用情况。	
MaximumLocalCapacity	允许的卷总容量。 “-”表示未限制本地卷容量。
EffectiveTime	对于订阅模式软件许可证, 软件许可证生效时间。
ExpireTime	对于订阅模式软件许可证, 软件许可证过期时间。
MaintenanceEffectiveTime	对于永久许可模式软件许可证, 软件许可证维保生效时间。
MaintenanceExpireTime	对于永久许可模式软件许可证, 软件许可证维保过期时间。
Status	软件许可证状态:

	● Effective: 软件许可证已生效。 ● Expired: 软件许可证已过期。 ● NotStart: 软件许可证未生效。
--	--

4.6 卷操作

4.6.1 创建卷

本地卷：

- 单机版：

```
./stor lun add { -n | --name } LUN_NAME { -p | --capacity } CAPACITY { -t | --target }  
TARGET_NAME [ { -o | --sector-size } SECTOR_SIZE ] [ { -w | --write-policy }  
WRITE_POLICY ] [ { -P | --path } PATH ] [ { { -m | --mode } STORAGE_MODE ]
```

- 集群版：

```
./stor lun add { -n | --name } LUN_NAME { -p | --capacity } CAPACITY { -t | --target }  
TARGET_NAME [ --priority SERVER_ID &<1-n> [ --auto-failback AUTO_FAILBACK ] ] [--pool  
POOL ] [ --cache-pool CACHE_POOL ] [ { -a | --ha } HIGH_AVAILABILITY ] [ { -c | --  
local-storage-class } LOCAL_STORAGE_CLASS ] [ --min-replica MIN_REPLICA ] [ --  
redundancy-overlap REDUNDANCY_OVERLAP ] [ --ec-fragment-size EC_FRAGEMENT_SIZE ] [ { -o  
| --sector-size } SECTOR_SIZE ] [ { -w | --write-policy } WRITE_POLICY ] [ { -m | --  
mode } STORAGE_MODE ]
```

上云卷：

- 单机版：

```
./stor lun add { -n | --name } LUN_NAME { -p | --capacity } CAPACITY { -t | --target }  
TARGET_NAME [ { -o | --sector-size } SECTOR_SIZE ] [ { -w | --write-policy }  
WRITE_POLICY ] [ { -P | --path } PATH ] { -m | --mode } STORAGE_MODE [ --cloud-provider  
CLOUD_PROVIDER ] { -B | --bucket } BUCKET_NAME { -A | --ak } ACCESS_KEY { -S | --sk }  
SECRET_KEY [ { -C | --cloud-storage-class } CLOUD_STORAGE_CLASS ] { -E | --endpoint }  
ENDPOINT [ --sign-version VERSION ] [ --region REGION ] [ { -M | --cloud-compression }  
CLOUD_COMPRESSION ] [ { -O | --object-size } OBJECT_SIZE ] [ { -X | --prefix } PREFIX ]
```

- 集群版：

```
./stor lun add { -n | --name } LUN_NAME { -p | --capacity } CAPACITY { -t | --target }
TARGET_NAME [ --priority SERVER_ID &<1-n> [ --auto-failback AUTO_FAILBACK ] ] [--pool
POOL ] [ --cache-pool CACHE_POOL ] [ { -a | --ha } HIGH_AVAILABILITY ] [ { -c | --
local-storage-class } LOCAL_STORAGE_CLASS ] [ --min-replica MIN_REPLICA ] [ --
redundancy-overlap REDUNDANCY_OVERLAP ] [ --ec-fragment-size EC_FRAGEMENT_SIZE ] [ { -o
| --sector-size } SECTOR_SIZE ] [ { -w | --write-policy } WRITE_POLICY ] { -m | --mode }
STORAGE_MODE [ --cloud-provider CLOUD_PROVIDER ] { -B | --bucket } BUCKET_NAME { -A | -
-ak } ACCESS_KEY { -S | --sk } SECRET_KEY [ { -C | --cloud-storage-class }
CLOUD_STORAGE_CLASS ] { -E | --endpoint } ENDPOINT [ --sign-version VERSION ] [ --
region REGION ] [ { -M | --cloud-compression } CLOUD_COMPRESSION ] [ { -O | --object-
size } OBJECT_SIZE ] [ { -X | --prefix } PREFIX ]
```

此命令用来创建卷。

说明：

- HBlock 上 Target IQN 的数量最多 32766 个。一个 Target 最多可以关联 256 个卷，但是一个卷只能被一个 Target 关联。
- 卷创建成功后，仅下列参数可以修改：卷容量、写策略、最小副本数（仅集群版支持）、折叠副本数（仅集群版支持）、主备优先级（仅集群版支持）、Endpoint、上云签名认证版本、区域、AK/SK、是否压缩数据。

参数

参数	描述
-n LUN_NAME 或--name LUN_NAME	指定卷名称。 取值：字符串形式，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。
-p CAPACITY 或--capacity	指定卷容量。

CAPACITY	取值：整数形式，数字后面可以输入单位简写 G/g、T/t 或 P/p，分别代表 GiB、TiB、PiB，如果不输入，默认为 GiB。 ● 如果单位是 GiB，取值为[1, 1048576]。 ● 如果单位是 TiB，取值为[1, 1024]。 ● 如果单位是 PiB，取值为 1。
-t TARGET_NAME 或 --target TARGET_NAME	指定 iSCSI Target 名称。 取值：字符串形式，长度范围 1~16，只能由小写字母、数字、句点(.)和短横线(-)组成，且仅支持以字母或数字开头。 说明：创建卷时，如果指定的 iSCSI Target 名称不存在，那么同时创建 iSCSI Target，新创建 iSCSI Target 的回收策略默认为 Delete。
--priority SERVER_ID &<1-n>	指定卷主备分布优先级的服务器 ID（仅集群版支持），系统会根据指定的服务器 ID 顺序来选择卷的主备 IQN。可以指定一个或者多个服务器 ID，以英文逗号分开。 前置条件：iSCSI Target 名称已经存在，且指定的服务器必须是 iSCSI Target 所在的服务器。
--auto-failback AUTO_FAILBACK	是否根据指定的卷主备分布优先级自动进行主备切换（仅集群版支持），即针对卷主备状态，当高优先级的服务器恢复正常后，是否自动进行主备状态切换。 取值： ● Enabled: 自动进行主备切换。 ● Disabled: 不自动进行主备切换。 默认值为 Enabled。
--pool 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。

--cache-pool <i>CACHE_POOL</i>	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。
-a <i>HIGH_AVAILABILITY</i> 或 --ha <i>HIGH_AVAILABILITY</i>	选择卷的高可用类型（仅集群版支持）： 取值： ● ActiveStandby (as)：启用主备，该卷关联对应 Target 下的所有 IQN。 ● Disabled (off)：不启用主备，该卷关联对应 Target 下的 1 个 IQN。客户端连接该类型的卷的方法可以参见 Windows 客户端 – 单机版、Linux 客户端 – 单机版。 默认值为 ActiveStandby (as)。
-c <i>LOCAL_STORAGE_CLASS</i> 或 --local-storage-class <i>LOCAL_STORAGE_CLASS</i>	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 $N + M \leq 128$。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。 说明（以下场景均为集群可用的前提下）： ● 创建 EC N+M 的卷后： ■ 卷所在存储池可用故障域数量大于等于卷的最小副本数时，可以向卷中写数据。卷所在存储池可用故障域数量小于最小副本数时，不能向卷写数据，且系统会产生告警。

	■ 卷所在存储池可用故障域数量大于等于 $N+M$，卷数据正常，不会产生降级。卷所在存储池可用故障域数量处于 $[N, N+M)$，卷数据将处于降级状态，建议尽快添加或修复故障域。卷所在存储池可用故障域数量小于 N 时，已写入的数据发生损毁。 ● 创建副本模式的卷后： ■ 卷所在存储池可用故障域数量大于等于卷的最小副本数时，可以向卷中写数据。卷所在存储池可用故障域数量小于最小副本数时，不能向卷写数据，且系统会产生告警。 ■ 卷所在存储池可用故障域数量大于等于副本数，卷数据正常，不会产生降级。对于两副本、三副本卷，卷存储池所在故障域大于等于 1，但小于副本数时，卷数据将处于降级状态，建议尽快添加或修复存储池故障域。卷所在存储池无可用故障域时，已写入的数据发生损毁。
--min-replica 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 $\leq X$），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 $N \leq Y \leq N+M$），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。 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 REDUNDANCY_OVERLAP	指定卷的折叠副本数（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。 注意：如果存储池故障域级别为 path，此参数不生效。 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[1, N+M]，默认值为 1。
--ec-fragment-size EC_FRAGMENT_SIZE	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。 取值：1、2、4、8、16、32、64、128、256、512、1024、2048、4096，单位是 KiB。默认值为 16。
-o SECTOR_SIZE 或 --sector-size SECTOR_SIZE	设置扇区大小。 取值：512、4096，单位为字节。默认值为 4096。 说明：扇区大小的选取：根据自身业务场景，一般情况下，单次 I/O 操作的数据大小大于或接近 4KiB，则推荐选择 4096；单次 I/O 操作的数据大小接近 512Bytes，则推荐选择 512。如果对接 VMware 等虚拟化平台，则推荐选择 512Bytes。
-w WRITE_POLICY 或 --write-policy 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)。
-P PATH 或 --path PATH	指定存储卷数据的数据目录（仅单机版支持）。 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。
-m STORAGE_MODE 或 --mode STORAGE_MODE	设置卷的存储类型： ● 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对象存储中。 ● Storage：存储模式，本地保留全部数据，并异步存储到对象存储中。 存储类型默认为 Local。
--cloud-provider CLOUD_PROVIDER	对象存储服务名称： ● OOS：天翼云对象存储经典版 I 型。 ● S3：兼容 S3 的其他对象存储。 默认值为 OOS。
-B BUCKET_NAME 或 --bucket BUCKET_NAME	已存在的存储桶的名称。 注意：请勿开启 Bucket 的生命周期设定和合规保留。
-A ACCESS_KEY 或 --ak ACCESS_KEY	对象存储服务的 Access Key。
-S SECRET_KEY 或 --sk SECRET_KEY	对象存储服务的 Secret Access Key。
-C CLOUD_STORAGE_CLASS 或 --cloud-storage-class	设置数据在对象存储的存储类型： ● STANDARD：标准存储。

<code>CLOUD_STORAGE_CLASS</code>	● STANDARD_IA: 低频访问存储。 默认为 STANDARD。
<code>-E ENDPOINT</code> 或 <code>--endpoint ENDPOINT</code>	设置对象存储的 Endpoint。 注意: ● Endpoint 必须与所使用的对象存储服务一一对应: 若使用 OOS 对象存储服务, 需填写 OOS 的 Endpoint; 若使用兼容 S3 的其他对象存储服务, 则需填写它对应的 Endpoint。 ● 如果仅输入域名将会使用 HTTPS 协议进行访问。 OOS Endpoint 详见 OOS Endpoint 和 Region。
<code>--sign-version VERSION</code>	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。
<code>--region REGION</code>	表示 Endpoint 资源池所在区域。 OOS region 详见 OOS Endpoint 和 Region。 V4 签名时, 此项必填。
<code>-M CLOUD_COMPRESSION</code> 或 <code>--cloud-compression CLOUD_COMPRESSION</code>	是否压缩数据上传至对象存储: ● Enabled (on): 压缩数据上传至对象存储。 ● Disabled (off): 不压缩数据上传至对象存储。 默认值为 Enabled (on)。
<code>-O OBJECT_SIZE</code> 或 <code>--object-size OBJECT_SIZE</code>	数据存储在对象存储的大小。 取值: 128、256、512、1024、2048、4096、8192, 单位是 KiB。默认值为 1024。
<code>-X PREFIX</code> 或 <code>--prefix PREFIX</code>	设置对象存储中的前缀名称, 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。

示例

- 单机版本地卷：创建卷 **lun1**，容量为 10 GiB，Target 为 **target1**，卷数据存储在数据目录 **/mnt/storage01** 上。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun add -n lun1 -t target1 -p 10
-P /mnt/storage01
Created LUN lun1 successfully.
```

- 集群版本地卷：创建卷 **lun1**，容量为 10 GiB，Target 为 **target1**，卷冗余模式为 3-copy。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun add -n lun1 -t target1 -p 10
-c 3-copy
Created LUN lun1 successfully.
```

- 集群版本地卷：创建卷 **lun01a**，容量为 100 GiB，Target 为 **target01**，卷冗余模式为 EC 2+1，分片为 32 KiB。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun add -n lun01a -t target01 -c
EC 2+1 --ec-fragment-size 32 -p 100
Created LUN lun01a successfully.
```

- 单机版上云卷：创建卷 **luna2**，容量为 110 GiB，Target 为 **targeta**，卷的存储类型为 **Storage**，云上数据存在 OOS，使用 V4 签名认证，存储前缀为 **hblock**。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun add -n luna2 -t targeta -p
110 -m Storage -B hblocktest3 -E oos-cn.ctyunapi.cn -A cb22b08b1f9229f85874 -S
***** --sign-version v4 --region cn -X hblock
Created LUN luna2 successfully.
```

- 集群版上云卷：创建卷 **lun03a**，容量为 1000 GiB，Target 为 **target03**，卷冗余模式为 EC 2+1，分片为 32 KiB，卷的存储类型为 **Cache**，云上数据存在 OOS，使用 V4 签名认证。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun add -n lun03a -t target03 -c
```

```
EC 2+1 --ec-fragment-size 32 -p 1000 -m Cache -B hblocktest1 -E oos-cn.ctyunapi.cn -A  
9fe711b8b89ad73c2a46 -S ***** --sign-version v4 --region cn  
Created LUN lun03a successfully.
```

4.6.2 创建克隆卷

单机版：

```
./stor lun { C | clone } { -n | --name } LUN_NAME { -s | --snapshot } SNAPSHOT_NAME { -t | --target } TARGET_NAME [ { -p | --capacity } CAPACITY ] [ { -o | --sector-size } SECTOR_SIZE ] [ { -w | --write-policy } WRITE_POLICY ] [ { -P | --path } PATH ]
```

集群版：

```
./stor lun { C | clone } { -n | --name } LUN_NAME { -s | --snapshot } SNAPSHOT_NAME { -t | --target } TARGET_NAME [ { -p | --capacity } CAPACITY ] [ --priority SERVER_ID <1-n> ] [ --auto-failback AUTO_FAILBACK ] [ --pool POOL ] [ --cache-pool CACHE_POOL ] [ { -a | --ha } HIGH_AVAILABILITY ] [ { -c | --local-storage-class } LOCAL_STORAGE_CLASS ] [ --min-replica MIN_REPLICA ] [ --redundancy-overlap REDUNDANCY_OVERLAP ] [ --ec-fragment-size EC_FRAGEMENT_SIZE ] [ { -o | --sector-size } SECTOR_SIZE ] [ { -w | --write-policy } WRITE_POLICY ]
```

此命令用来创建克隆卷。

前提条件：用于创建克隆卷的快照，其状态必须是 Normal。

说明：

- 系统支持的最大克隆卷数：100000。
- 单个快照可创建的最大克隆卷数：512。
- 系统支持的最大克隆深度：16。

参数

参数	描述
-n LUN_NAME 或--name LUN_NAME	指定克隆卷名称。 取值：字符串形式，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。
-s SNAPSHOT_NAME 或--	指定克隆卷关联的快照名称。

snapshot <i>SNAPSHOT_NAME</i>	取值：字符串形式，长度范围是 1~256，只能由字母、数字、短横线(-)、下划线(_)组成，字母区分大小写，且仅支持以字母或数字开头。
-t <i>TARGET_NAME</i> 或 --target <i>TARGET_NAME</i>	指定克隆卷关联的 iSCSI Target 名称，可以跟源卷的 iSCSI Target 不同。 取值：字符串形式，长度范围 1~16，只能由小写字母、数字、句点(.)和短横线(-)组成，且仅支持以字母或数字开头。 说明：创建卷时，如果指定的 iSCSI Target 名称不存在，那么同时创建 iSCSI Target，新创建 iSCSI Target 的回收策略默认为 Delete。
-p <i>CAPACITY</i> 或 --capacity <i>CAPACITY</i>	指定克隆卷的容量。 取值：整数形式，数字后面可以输入单位简写 G/g、T/t 或 P/p，分别代表 GiB、TiB、PiB，如果不输入，默认为 GiB。 ● 如果单位是 GiB，取值为[1, 1048576]。 ● 如果单位是 TiB，取值为[1, 1024]。 ● 如果单位是 PiB，取值为 1。 默认为快照时刻的源卷容量，如果重新设置，则必须大于等于快照时刻的源卷的容量。
--priority <i>SERVER_ID</i> &<1- <i>n</i> >	指定克隆卷主备分布优先级的服务器 ID（仅集群版支持），系统会根据指定的服务器 ID 顺序来选择卷的主备 IQN。可以指定一个或者多个服务器 ID，以英文逗号分开。 前置条件：iSCSI Target 名称已经存在，且指定的服务器必须是 iSCSI Target 所在的服务器。
--auto-failback	是否根据指定的克隆卷主备分布优先级自动进行主备切换

<code>AUTO_FAILBACK</code>	（仅集群版支持），即针对克隆卷主备状态，当高优先级的服务器恢复正常后，是否自动进行主备状态切换。 取值： ● Enabled: 自动进行主备切换。 ● Disabled: 不自动进行主备切换。 默认值为 Enabled。
<code>--pool POOL</code>	指定存储池（仅集群版支持）。默认值与源卷的配置一致。 注意： ● <code>POOL</code> 与 <code>CACHE_POOL</code> 不能设置为同一个存储池。 ● 当克隆卷的 <code>POOL</code> 与 <code>CACHE_POOL</code> 配置与源卷一致，无需额外设置。若单独设置了克隆卷的 <code>POOL</code> 或 <code>CACHE_POOL</code>，则不再沿用源卷的 <code>POOL</code> 和 <code>CACHE_POOL</code>，而是采用所设参数值。例如，源卷同时有 <code>CACHE_POOL</code> 和 <code>POOL</code>，若克隆卷仅重新设置了 <code>POOL</code> 而未设置 <code>CACHE_POOL</code>，则克隆卷仅使用新设置的 <code>POOL</code>，无 <code>CACHE_POOL</code>；反之，若克隆卷设置了 <code>CACHE_POOL</code>，则必须同时设置 <code>POOL</code>，或者让 <code>POOL</code> 使用基础存储池。
<code>--cache-pool CACHE_POOL</code>	指定缓存存储池（仅集群版支持）。默认值与源卷的配置一致。 注意： ● <code>POOL</code> 与 <code>CACHE_POOL</code> 不能设置为同一个存储池。 ● 当克隆卷的 <code>POOL</code> 与 <code>CACHE_POOL</code> 配置与源卷一致，无需额外设置。若单独设置了克隆卷的 <code>POOL</code> 或 <code>CACHE_POOL</code>，则不再沿用源卷的 <code>POOL</code> 和 <code>CACHE_POOL</code>，而是采用所设参数值。例如，源卷同时

	有 <code>CACHE_POOL</code> 和 <code>POOL</code>，若克隆卷仅重新设置了 <code>POOL</code> 而未设置 <code>CACHE_POOL</code>，则克隆卷仅使用新设置的 <code>POOL</code>，无 <code>CACHE_POOL</code>；反之，若克隆卷设置了 <code>CACHE_POOL</code>，则必须同时设置 <code>POOL</code>，或者让 <code>POOL</code> 使用基础存储池。
<code>-a HIGH_AVAILABILITY</code> 或 <code>--ha HIGH_AVAILABILITY</code>	指定克隆卷的高可用类型（仅集群版支持）： 取值： ● ActiveStandby (as)：启用主备，该卷关联对应 Target 下的所有 IQN。 ● Disabled (off)：不启用主备，该卷关联对应 Target 下的 1 个 IQN。客户端连接该类型的卷的方法可以参见 Windows 客户端 – 单机版、Linux 客户端 – 单机版。 默认值为源卷的高可用类型。
<code>-c LOCAL_STORAGE_CLASS</code> 或 <code>--local-storage-class LOCAL_STORAGE_CLASS</code>	指定克隆卷的冗余模式（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 $N + M \leq 128$。表示将数据分割成 N 个片段，并生成 M 个校验数据。 默认值为源卷的卷冗余模式。 说明（以下场景均为集群可用的前提下）： ● 创建 EC N+M 的卷后： ■ 卷所在存储池可用故障域数量大于等于卷的最小副本数时，可以向卷中写数据。卷所在存储池可

	用故障域数量小于最小副本数时，不能向卷写数据，且系统会产生告警。 ■ 卷所在存储池可用故障域数量大于等于 $N+M$，卷数据正常，不会产生降级。卷所在存储池可用故障域数量处于 $[N, N+M)$，卷数据将处于降级状态，建议尽快添加或修复故障域。卷所在存储池可用故障域数量小于 N 时，已写入的数据发生损毁。 ● 创建副本模式的卷后： ■ 卷所在存储池可用故障域数量大于等于卷的最小副本数时，可以向卷中写数据。卷所在存储池可用故障域数量小于最小副本数时，不能向卷写数据，且系统会产生告警。 ■ 卷所在存储池可用故障域数量大于等于副本数，卷数据正常，不会产生降级。对于两副本、三副本卷，卷存储池所在故障域大于等于 1，但小于副本数时，卷数据将处于降级状态，建议尽快添加或修复存储池故障域。卷所在存储池无可用故障域时，已写入的数据发生损毁。
--min-replica MIN_REPLICA	指定克隆卷的最小副本数（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 $\leq X$），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 $N \leq Y \leq N+M$），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。 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$[N, N+M]$。如果未设置克隆卷的最小副本数，默认值为源卷的最小副本数。
--redundancy-overlap REDUNDANCY_OVERLAP	指定克隆卷的折叠副本数（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。 注意：如果存储池故障域级别为 path，此参数不生效。 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$[1, N+M]$。如果未设置克隆卷的折叠副本数，默认值为源卷的折叠副本数。
--ec-fragment-size EC_FRAGMENT_SIZE	指定克隆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。 取值：1、2、4、8、16、32、64、128、256、512、1024、2048、4096，单位是 KiB。默认值为源卷的纠删码模式分片大小。
-o SECTOR_SIZE 或 --sector-size SECTOR_SIZE	指定克隆卷的扇区大小。 取值：512、4096，单位为字节。默认值为源卷的扇区大小。 说明：扇区大小的选取：根据自身业务场景，一般情况下，单次 I/O 操作的数据大小大于或接近 4KiB，则推荐选择 4096；单次 I/O 操作的数据大小接近 512Bytes，则推荐选择 512。如果对接 VMware 等虚拟化平台，则推荐选择 512Bytes。
-w WRITE_POLICY 或 --write-policy	克隆卷的写策略： ● WriteBack (wb)：回写，指数据写入到内存后即返

WRITE_POLICY	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。 默认为源卷的写策略。
-P PATH 或 --path PATH	指定存储克隆卷数据的数据目录（仅单机版支持）。 如果创建克隆卷时不指定数据目录，默认与源卷配置保持一致。

示例

- 单机版：创建克隆卷 **luna1-C1**，关联的快照为 **luna1-snapshot**，关联的 iSCSI Target 为 **targeta1**，卷容量为 200 GiB，其余配置与源卷一致。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun C -n luna1-C1 -s luna1-snapshot -t targeta1 -p 200
Confirm the information you set. If correct, enter Yes(Y). If not, enter No(N).
LUN Name: luna1-C1
Source Snapshot: luna1-snapshot
Capacity: 200 GiB
iSCSI Target: targeta1
Local Sector Size: 4096 bytes
Write Policy: WriteBack
Path: /mnt/stor01
yes
Created clone LUN luna1-C1 successfully.
```

- 集群版：创建克隆卷 **lun01a-C1**，关联的快照为 **lun01a-snap20250304170942**，关联的 iSCSI Target 为 **target01**，其余配置与源卷一致。

```
[root@hbblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun C -n lun01a-C1 -s lun01a-snap20250304170942 -t target01
Confirm the information you set. If correct, enter Yes(Y). If not, enter No(N).
LUN Name: lun01a-C1
Source Snapshot: lun01a-snap20250304170942
Capacity: 100 GiB
iSCSI Target: target01
Local Storage Class: EC 2+1 + 16KiB
Minimum Replica Number: 2
Redundancy Overlap: 1
Local Sector Size: 4096 bytes
Storage Pool: default
High Availability: ActiveStandby
Write Policy: WriteBack
yes
Created clone LUN lun01a-C1 successfully.
```

4.6.3 断开克隆卷与快照的关系链

```
./stor lun { F | flatten } { -n | --name } LUN_NAME
```

此命令用来断开克隆卷与快照的关系链。断开与快照的关联后，克隆卷为独立卷。

注意：

- 若源卷数据损坏，或因其他原因致使此操作无法读取源卷数据，这部分数据便无法复制到克隆卷，系统会持续重试，直至数据可读取后才进行复制。
- 若该克隆卷处于回滚中，无法执行此操作。

参数

参数	描述
-n LUN_NAME 或 --name LUN_NAME	指定要与快照断开关系链的克隆卷名称。

示例

断开克隆卷 clone-lun02-2 与快照的关系链。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun F -n clone-lun02-2
Start flattening the clone LUN clone-lun02-2.
```

4.6.4 删除卷

本地卷

```
./stor lun rm { -n | --name } LUN_NAME [ -f | --force ]
```

上云卷

```
./stor lun rm { -n | --name } LUN_NAME [ { -c | --del-cloud } ] [ -f | --force ]
```

此命令用来删除指定的卷。

说明：对于上云卷，如果删除卷的时候未删除云上数据，后期可以使用卷还原功能进行卷数据还原。删除卷时，应确保卷不包含快照、一致性快照。

注意：

- 对于上云卷，选择强制删除卷，且同时删除云上数据，可能会有云上数据残留，需要对云上数据进行手动删除。
- 如果卷存在关联克隆卷，且克隆卷未处于“Deleting”或“Flattening”状态，则禁止删除源卷。
- 如果卷有关联的快照或一致性快照，只能强制删除卷。强制删除卷时，将同时删除此卷的快照和一致性快照中的卷快照。同时也会产生数据残留风险，请谨慎操作。

参数

参数	描述
-n <i>LUN_NAME</i> 或 --name <i>LUN_NAME</i>	指定删除的卷名称。
-c 或 --del-cloud	删除卷时，删除云上数据。
-f 或 --force	强制删除卷。 注意： 强制删除卷，会产生数据残留风险，请谨慎操作。

示例

删除卷 lun1。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun rm -n lun1
```

Start deleting LUN lun1. You can list LUNs to check whether the deletion is completed.

4.6.5 扩容卷

```
./stor lun { X | expand } { -n | --name } LUN_NAME { -p | --capacity } CAPACITY
```

此命令用来扩容卷。

参数

参数	描述
-n LUN_NAME 或 --name LUN_NAME	指定要扩容卷的卷名称。
-p CAPACITY 或 --capacity CAPACITY	扩容卷后的卷容量，此次卷容量需要大于等于扩容前的卷容量。 取值：整数形式，数字后面可以输入单位简写 G/g、T/t 或 P/p，分别代表 GiB、TiB、PiB，如果不输入，默认为 GiB。 ● 如果单位是 GiB，取值为[1, 1048576]。 ● 如果单位是 TiB，取值为[1, 1024]。 ● 如果单位是 PiB，取值为 1。

示例

注意：如果 LUN 还未挂载到客户端，在服务器端扩容后，客户端挂载卷的时候会使用扩容后的卷容量。如果 LUN 已经挂载客户端，服务器端 LUN 扩容后，客户端也需要进行卷扩容操作，才能扩容成功。

- LUN 未挂载到客户端时，直接在服务器端执行卷扩容即可：

```
[root@server CTYUN_HBblock_Plus_3.9.0_x64]# ./stor lun X -n lun3 -p 11
Expanded LUN lun3 successfully.
```

- LUN 已挂载到客户端：
 - 对于 windows 系统：

服务器端：

```
[root@server CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun X -n lun3 -p 11  
Expanded LUN lun3 successfully.
```

客户端：

打开“服务器管理器”>“文件和存储服务”>“卷”>“磁盘”，在空白处点击右键，“重新扫描存储”（见下图），磁盘容量更新后，点击“卷”右键“扩展卷”。此时已完成客户端的扩容卷操作，查看对应的磁盘容量，为扩容过后卷的容量。



图1.重新扫描存储



图2.扩展卷

➤ 对于 Linux 系统：

服务端：执行卷扩容命令

```
./stor lun { X | expand } { -n | --name } LUN_NAME { -p | --capacity } CAPACITY
```

客户端：

1. 扫描磁盘：

- 如果是 HBlock 单机版客户端，找到 iSCSI 卷对应的盘符，执行扫描磁盘。
- 如果是 HBlock 集群版客户端，找到 iSCSI 卷对应的多个盘符，执行扫描磁盘。

```
echo 1 > /sys/class/block/sdX/device/rescan # 其中 sdX 为 iSCSI 卷在客户端的盘符。
```

2. 更新多路径大小（多控卷）。

```
multipathd resize map mpathX
```

3. 扩容文件系统。

```
resize2fs /dev/mapper/mpathX  
# ext4 扩容。如果是 XFS 扩容，使用 xfs_growfs /dev/mapper/mpathX
```

说明：单机版设备名称为 `/dev/sdX`，集群版设备名为 `/dev/mapper/mpathX`，用户可以根据情况进行执行相关命令。

4. 查看扩容后的磁盘。

```
lsblk
```

LINUX 集群版卷扩容示例（卷 lun01 容量从 30 GiB 扩展到 40 GiB）：

■ 服务器端：

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun ls -n lun01  
LUN Name: lun01 (LUN 0)  
Capacity: 30 GiB  
Storage Mode: Local  
Status: Normal  
Auto Failback: Enabled  
iSCSI Target: iqn.2012-08.cn.ctyunapi.oos:target01.1(192.168.0.102:3260,Active)  
               iqn.2012-08.cn.ctyunapi.oos:target01.2(192.168.0.110:3260,Standby)  
               iqn.2012-08.cn.ctyunapi.oos:target01.3(192.168.0.192:3260,ColdStandby)  
Create Time: 2024-01-02 13:47:11  
Local Storage Class: EC 2+1+16KiB  
Minimum Replica Number: 2  
Local Sector Size: 4096 bytes
```

```
High Availability: ActiveStandby
Write Policy: WriteBack
WWID: 33fffffffffaae6e920
UUID: lun-uuid-1d5ec154-7fb8-412a-b14c-b490eb33dad3
Snapshot Numbers: 0
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun X -n lun01 -p 40
Expanded LUN lun01 successfully.
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun ls -n lun01
LUN Name: lun01 (LUN 0)
Capacity: 40 GiB
Storage Mode: Local
Status: Normal
Auto Failback: Enabled
iSCSI Target: iqn.2012-08.cn.ctyunapi.oos:target01.1(192.168.0.102:3260,Active)
               iqn.2012-08.cn.ctyunapi.oos:target01.2(192.168.0.110:3260,Standby)
               iqn.2012-08.cn.ctyunapi.oos:target01.3(192.168.0.192:3260,ColdStandby)
Create Time: 2024-01-02 13:47:11
Local Storage Class: EC 2+1+16KiB
Minimum Replica Number: 2
Local Sector Size: 4096 bytes
High Availability: ActiveStandby
Write Policy: WriteBack
WWID: 33fffffffffaae6e920
UUID: lun-uuid-1d5ec154-7fb8-412a-b14c-b490eb33dad3
Snapshot Numbers: 0
```

■ **客户端：**在 lun01 扩容前，客户端已经连接 lun01。

```
[root@client ~]# lsblk
NAME            MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
sda              8:0    0   30G  0 disk
└─mpatha        252:3    0   30G  0 mpath /mnt/disk_mpatha
sdb             8:16    0  100G  0 disk
└─mpathb        252:2    0  100G  0 mpath
sdc             8:32    0   30G  0 disk
```

```

└─mpatha    252:3    0   30G  0 mpath /mnt/disk_mpatha
sdd         8:48    0  100G  0 disk
└─mpathb    252:2    0  100G  0 mpath
sde         8:64    0   30G  0 disk
└─mpatha    252:3    0   30G  0 mpath /mnt/disk_mpatha
sdf         8:80    0  100G  0 disk
└─mpathb    252:2    0  100G  0 mpath
vda         253:0    0   40G  0 disk
├─vda1      253:1    0    4G  0 part [SWAP]
└─vda2      253:2    0   36G  0 part /

[root@client ~]# echo 1 > /sys/class/block/sda/device/rescan
[root@client ~]# echo 1 > /sys/class/block/sdc/device/rescan
[root@client ~]# echo 1 > /sys/class/block/sde/device/rescan
[root@client ~]# multipathd resize map mpatha
ok

[root@client ~]# resize2fs /dev/mapper/mpatha
resize2fs 1.42.9 (28-Dec-2013)
Filesystem at /dev/mapper/mpatha is mounted on /mnt/disk_mpatha; on-line resizing required
old_desc_blocks = 4, new_desc_blocks = 5
The filesystem on /dev/mapper/mpatha is now 10485760 blocks long.

[root@client ~]# lsblk
NAME            MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
sda              8:0      0   40G  0 disk
└─mpatha        252:3    0   40G  0 mpath /mnt/disk_mpatha
sdb              8:16     0  100G  0 disk
└─mpathb       252:2    0  100G  0 mpath
sdc              8:32     0   40G  0 disk
└─mpatha       252:3    0   40G  0 mpath /mnt/disk_mpatha
sdd              8:48     0  100G  0 disk
└─mpathb       252:2    0  100G  0 mpath
sde              8:64     0   40G  0 disk
└─mpatha       252:3    0   40G  0 mpath /mnt/disk_mpatha
sdf              8:80     0  100G  0 disk
└─mpathb       252:2    0  100G  0 mpath
  
```

vda	253:0	0	40G	0	disk	
vda1	253:1	0	4G	0	part	[SWAP]
vda2	253:2	0	36G	0	part	/

4.6.6 修改卷

本地卷：

```
./stor lun set { -n | --name } LUN_NAME [ { -w | --write-policy } WRITE_POLICY ] [ --min-replica MIN_REPLICA ] [ --redundancy-overlap REDUNDANCY_OVERLAP ] [ --priority { SERVER_ID <1-n> } | CLEAR ] [ --auto-failback AUTO_FAILBACK ]
```

上云卷：

```
./stor lun set { -n | --name } LUN_NAME [ { -w | --write-policy } WRITE_POLICY ] [ --min-replica MIN_REPLICA ] [ --redundancy-overlap REDUNDANCY_OVERLAP ] [ --priority { SERVER_ID <1-n> } | CLEAR ] [ --auto-failback AUTO_FAILBACK ] [ { -A | --ak } ACCESS_KEY { -S | --sk } SECRET_KEY ] [ { -E | --endpoint } ENDPOINT ] [ --sign-version VERSION ] [ --region REGION ] [ { -M | --cloud-compression } CLOUD_COMPRESSION ]
```

此命令用来修改指定的卷。

说明：卷处于“删除中”、“回滚中”，不能修改卷的配置。

参数

参数	描述
-n LUN_NAME 或--name LUN_NAME	指定修改的卷名称。
-w WRITE_POLICY 或--write-policy 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。
--min-replica 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$\leq$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 $N \leq Y \leq N+M$），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。 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]。
--redundancy-overlap REDUNDANCY_OVERLAP	指定卷的折叠副本数（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。 注意：如果存储池故障域级别为 path，此参数不生效。 取值：整数。对副本模式，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[1, N+M]。
--priority SERVER_ID	指定卷主备分布优先级的服务器 ID（仅集群版支持），系统会根据指定的服务器 ID 顺序来选择卷的主备 IQN。可以指定一个或者多个服务器 ID，以英文逗号分开。
--priority CLEAR	取消卷的主备优先级设置（仅集群版支持）。
--auto-failback AUTO_FAILBACK	是否根据指定的卷主备分布优先级自动进行主备切换（仅集群版支持），即针对卷主备状态，当高优先级的服务器恢复正常后，是否自动进行主备状态切换。 取值：

	● Enabled: 自动进行主备切换。 ● Disabled: 不自动进行主备切换。
-A ACCESS_KEY 或 --ak ACCESS_KEY	修改对象存储服务的 Access Key。 注意 : 如果要修改对象存储服务的 Access Key、Secret Access Key, 必须同时修改。
-S SECRET_KEY 或 --sk SECRET_KEY	修改对象存储服务的 Secret Access Key。 注意 : 如果要修改对象存储服务的 Access Key、Secret Access Key, 必须同时修改。
-E ENDPOINT 或 --endpoint ENDPOINT	设置对象存储服务的 Endpoint。 注意 : ● Endpoint 必须与所使用的对象存储服务一一对应: 若使用 OOS 对象存储服务, 需填写 OOS 的 Endpoint; 若使用兼容 S3 的其他对象存储服务, 则需填写它的 Endpoint。 ● 如果仅输入域名将会使用 HTTPS 协议进行访问。 OOS Endpoint 详见 OOS Endpoint 和 Region 。
--sign-version VERSION	修改上云签名认证的类型: ● v2: V2 签名认证。 ● v4: V4 签名认证。
--region REGION	表示 Endpoint 资源池所在区域。V4 签名时, 此项必填。
-M CLOUD_COMPRESSION 或 --cloud-compression CLOUD_COMPRESSION	修改是否压缩数据上传至对象存储: ● Enabled (on): 压缩数据上传至对象存储。 ● Disabled (off): 不压缩数据上传至对象存储。

示例

修改卷的写策略为透写 (wt)。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun set -n lun01a -w wt
```

```
Set LUN lun01a successfully.
```

修改卷的最小副本数为 3。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun set -n lun01a --min-replica 3
Set LUN lun01a successfully.
```

修改卷的主备分布优先级的服务器 ID。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun set -n lun01a --priority
hblock_1
Set LUN lun01a successfully.
```

取消卷的主备分布优先级的服务器 ID。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun set -n lun01a --priority CLEAR
Set LUN lun01a successfully.
```

修改卷的签名为 v4 签名，不压缩数据上传至对象存储。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun set -n lun03a --sign-version
v4 --region cn -M off
Set LUN lun03a successfully.
```

4.6.7 设置卷主备优先级或自动切换（集群版适用）

```
./stor lun prefer [ { -n | --name } LUN_NAME &<1-n> ] [--priority { SERVER_ID &<1-n> |  
CLEAR } | --status-driven ] [ --auto-failback AUTO_FAILBACK ]
```

此命令用来设置卷的主备优先级或自动切换。

参数

参数	描述
-n LUN_NAME 或--name LUN_NAME	指定卷名称。可以指定一个或者多个卷，以英文逗号分开。 说明： 如果不指定卷名称，表示设置所有卷。
--priority SERVER_ID	指定卷主备分布优先级的服务器 ID。指定卷主备分布优先级的服务器 ID，系统会根据指定的服务器 ID 顺序来选择卷的主备 IQN。可以指定一个或者多个服务器 ID，以英文逗号分开。 注意： ● 如果指定卷主备分布优先级的服务器 ID，只能指定一个卷名称。 ● --priority SERVER_ID、--priority CLEAR 与 --status-driven 互斥。
--priority CLEAR	取消卷的主备优先级设置。 注意： --priority SERVER_ID 、 --priority CLEAR 与 --status-driven 互斥。
--status-driven	根据卷的当前主备 Target（Active Target、Standby Target、Cold Target）所在服务器，进行优先级配置。 注意： --priority SERVER_ID 、 --priority CLEAR 与 --status-driven 互斥。

--auto-failback AUTO_FAILBACK	是否根据指定的卷主备分布优先级自动进行主备切换，即针对卷主备状态，当高优先级的服务器恢复正常后，是否自动进行主备状态切换。 取值： ● Enabled: 自动进行主备切换。 ● Disabled: 不自动进行主备切换。 默认值为 Enabled。
--	---

示例

设置卷 lun04a 的主备优先级。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun prefer -n lun04a --priority hblock_2
Set LUN lun04a with server affinity successfully.
```

设置卷 lun04a、lun04b 的自动进行主备切换为 Disabled。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun prefer -n lun04a,lun04b --auto-failback Disabled
Set LUN lun04a,lun04b with server affinity successfully.
```

根据卷的当前主备 Target 所在服务器，设置卷 lun04a、lun04b 的主备优先级。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun prefer -n lun04a,lun04b --status-driven
Set LUN lun04a,lun04b with server affinity successfully
```

取消卷 lun04a 的主备优先级设置。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun prefer -n lun04a --priority CLEAR
Set LUN lun04a with server affinity successfully.
```

4.6.8 触发卷对应 Target 的主备切换（集群版适用）

```
./stor lun { S | switch } [ { -n | --name } LUN_NAME &<1-n> ]
```

此命令用来触发卷对应 Target 的主备切换。

说明：执行此操作后，客户端不需要进行任何操作。

参数

参数	描述
-n LUN_NAME 或 --name LUN_NAME	指定卷名称。可以指定一个或者多个卷，以英文逗号分开。 说明：如果不指定卷名称，表示设置所有卷。

示例

触发卷对应 Target 的主备切换。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun S -n lun01a
Triggered the active and standby switchover on LUN lun01a successfully.
iSCSI Target: iqn.2012-08.cn.ctyunapi.oos:target01.4(192.168.0.102:3260,Active)
iSCSI Target: iqn.2012-08.cn.ctyunapi.oos:target01.3(192.168.0.192:3260,ColdStandby)
```

触发所有卷对应 Target 的主备切换。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun S
Triggered the active and standby switchover on LUN lun01a,lun02a,lun03a successfully.
```

4.6.9 还原卷（上云卷适用）

单机版：

```
./stor lun { RC | --recover } { -n | --name } LUN_NAME { -t | --target } TARGET_NAME --  
uuid UUID [ { -w | --write-policy } WRITE_POLICY ] [ { -P | --path } PATH ] { -m | --  
mode } STORAGE_MODE [ --cloud-provider CLOUD_PROVIDER ] { -B | --bucket } BUCKET_NAME  
[ { -X | --prefix } PREFIX ] { -A | --ak } ACCESS_KEY { -S | --sk } SECRET_KEY { -E | --  
endpoint } ENDPOINT [ --sign-version VERSION ] [ --region REGION ] [ { -C | --cloud-  
storage-class } CLOUD_STORAGE_CLASS ] [ { -M | --cloud-compression } CLOUD_COMPRESSION ]
```

集群版：

```
./stor lun { RC | --recover } { -n | --name } LUN_NAME { -t | --target } TARGET_NAME  
[ --pool POOL ] [ --cache-pool CACHE_POOL ] --uuid UUID [ { -a | --ha }  
HIGH_AVAILABILITY ] [ { -c | --local-storage-class } LOCAL_STORAGE_CLASS ] [ --min-  
replica MIN_REPLICA ] [ --ec-fragment-size EC_FRAGMENT_SIZE ] [ --redundancy-overlap  
REDUNDANCY_OVERLAP ] [ { -w | --write-policy } WRITE_POLICY ] { -m | --mode }  
STORAGE_MODE [ --cloud-provider CLOUD_PROVIDER ] { -B | --bucket } BUCKET_NAME [ { -X |  
--prefix } PREFIX ] { -A | --ak } ACCESS_KEY { -S | --sk } SECRET_KEY [ { -C | --cloud-  
storage-class } CLOUD_STORAGE_CLASS ] { -E | --endpoint } ENDPOINT [ --sign-version  
VERSION ] [ --region REGION ] [ { -M | --cloud-compression } CLOUD_COMPRESSION ]
```

此命令用来还原 HBlock 上云卷。

以下场景适合还原卷功能：

- 原 HBlock 中存在上云卷，并且将数据上传到了云端。如果此时 HBlock 发生故障无法启动，可以通过还原卷的功能，在另一个 HBlock 中将该卷重新生成，并且从云端进行数据恢复。
- 原上云卷被删除，但云上数据保留，可以通过还原卷功能恢复卷数据。

注意：

- 在执行还原卷的操作前，请确保源卷已经将所有数据上传到云端，并且源卷不再继续使用（源卷已删除，或者源卷所在系统已经停止服务）。

- 待还原的卷名称在当前 HBlock 中不存在。
- 待还原的卷必须在指定的 Bucket/prefix 中找到，且数据完整。
- 还原操作异步执行，请通过查询卷的功能查看还原进度。

参数

参数	描述
-n LUN_NAME 或 --name LUN_NAME	源卷的名称。
-t TARGET_NAME 或 --target TARGET_NAME	指定还原卷的 iSCSI Target 名称。 取值：字符串形式，长度范围 1~16，只能由小写字母、数字、句点(.)和短横线(-)组成，且仅支持以字母或数字开头。 说明： 还原卷时，如果指定的 iSCSI Target 名称不存在，那么同时创建 iSCSI Target。
--pool 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。
--cache-pool 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。
--uuid UUID	源卷的唯一标识码。
-a HIGH_AVAILABILITY 或 --ha HIGH_AVAILABILITY	指定还原卷的高可用类型（仅集群版支持）： 取值： ● ActiveStandby (as)：启用主备，该卷关联对应 Target 下的所有 IQN。 ● Disabled (off)：不启用主备，该卷关联对应 Target 下的 1 个 IQN。客户端连接该类型的卷的方法可以参见 Windows 客户端 – 单机版、Linux 客户端 – 单

	机版。 默认值为源卷的高可用类型。
-c LOCAL_STORAGE_CLASS 或 --local-storage-class LOCAL_STORAGE_CLASS	设置还原卷的冗余模式（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 $N+M \leq 128$。表示将数据分割成 N 个片段，并生成 M 个校验数据。 默认值为源卷的冗余模式。 说明（以下场景均为集群可用的前提下）： ● 还原 EC N+M 的卷后： ■ 卷所在存储池可用故障域数量大于等于卷的最小副本数时，可以向卷中写数据。卷所在存储池可用故障域数量小于最小副本数时，不能向卷写数据，且系统会产生告警。 ■ 卷所在存储池可用故障域数量大于等于 N+M，卷数据正常，不会产生降级。卷所在存储池可用故障域数量处于 $[N, N+M)$，卷数据将处于降级状态，建议尽快添加或修复故障域。卷所在存储池可用故障域数量小于 N 时，已写入的数据发生损毁。 ● 还原副本模式的卷后： ■ 卷所在存储池可用故障域数量大于等于卷的最小副本数时，可以向卷中写数据。卷所在存储池可用故障域数量小于最小副本数时，不能向卷写数

	据，且系统会产生告警。 ■ 卷所在存储池可用故障域数量大于等于副本数，卷数据正常，不会产生降级。对于两副本、三副本卷，卷存储池所在故障域大于等于 1，但小于副本数时，卷数据将处于降级状态，建议尽快添加或修复存储池故障域。卷所在存储池无可用故障域时，已写入的数据发生损毁。
--min-replica MIN_REPLICA	设置还原卷的最小副本数（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$\leq$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 $N \leq Y \leq N+M$），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。 注意：如果指定了还原卷的最小副本数，必须指定还原卷的冗余模式。 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]。如果未设置还原卷的最小副本数，默认值为源卷的最小副本数。
--redundancy-overlap REDUNDANCY_OVERLAP	指定还原卷的折叠副本数（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。 注意：如果存储池故障域级别为 path，此参数不生效。如果指定了还原卷的折叠副本数，必须指定还原卷的冗余模

	式。 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[1, N+M]。如果未设置还原卷的折叠副本数，默认值为源卷的折叠副本数。
--ec-fragment-size <i>EC_FRAGEMENT_SIZE</i>	设置还原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。 取值：1、2、4、8、16、32、64、128、256、512、1024、2048、4096，单位是 KiB。默认值为源卷的纠删码模式分片大小。
-w WRITE_POLICY 或 --write-policy <i>WRITE_POLICY</i>	设置还原卷的写策略： ● 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。 默认值为源卷的写策略。
-P PATH 或 --path PATH	指定还原卷存储数据的数据目录（仅单机版支持）。 如果还原卷时不指定数据目录，使用服务器设置的默认数据目录。
-m STORAGE_MODE 或 --mode STORAGE_MODE	设置还原卷的存储类型： ● Cache：缓存模式，本地保留部分热数据，全部数据异步存储到对象存储中。

	● Storage: 存储模式，本地保留全部数据，并异步存储到对象存储中。 存储类型默认为源卷的存储类型。
--cloud-provider <i>CLOUD_PROVIDER</i>	指定还原卷的对象存储服务名称： ● OOS: 天翼云对象存储经典版 I 型。 ● S3: 兼容 S3 的其他对象存储。 默认值为 OOS。 说明： 源卷和还原卷必须使用相同的对象存储服务。
-B BUCKET_NAME 或 --bucket <i>BUCKET_NAME</i>	源卷所在存储桶的名称。 注意： 请勿开启 Bucket 的生命周期设定和合规保留。
-X PREFIX 或 --prefix <i>PREFIX</i>	源卷的前缀名称。如果源卷未指定前缀，则此处不设置。 取值：字符串形式，长度范围是 1~256。
-A ACCESS_KEY 或 --ak <i>ACCESS_KEY</i>	源卷的 Access Key。
-S SECRET_KEY 或 --sk <i>SECRET_KEY</i>	源卷的 Secret Access Key。
-C CLOUD_STORAGE_CLASS 或 --cloud-storage-class <i>CLOUD_STORAGE_CLASS</i>	设置还原卷上传数据至对象存储的存储类型： ● STANDARD: 标准存储。 ● STANDARD_IA: 低频访问存储。 默认为源卷上传数据至对象存储的存储类型。
-E ENDPOINT 或 --endpoint <i>ENDPOINT</i>	源卷的 Endpoint。 注意： 如果仅输入域名将会使用 HTTPS 协议进行访问。
--sign-version <i>VERSION</i>	指定还原卷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。
--region <i>REGION</i>	还原卷的 region，即 Endpoint 资源池所在区域。

	V4 签名时，此项必填。
-M CLOUD_COMPRESSION 或 --cloud-compression CLOUD_COMPRESSION	设置还原卷是否压缩数据上传至对象存储： ● Enabled (on)：压缩数据上传至对象存储。 ● Disabled (off)：不压缩数据上传至对象存储。 默认值为源卷的设置。

示例

还原卷 lun01a。

<pre>[root@songt-0004 CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun recover -n lun01a -t target01 --uuid lun-uuid-38c7678c-33fd-4abe-b80f-0f5cd3ec6b18 --pool pool1 -B hblocktest3 -A 8f129a5529f202811fd0 -S ***** -E oos-cn.ctyunapi.cn Confirm the information you set. If correct, enter Yes(Y). If not, enter No(N). UUID: lun-uuid-38c7678c-33fd-4abe-b80f-0f5cd3ec6b18 LUN Name: lun01a Local Storage Class: EC 2+1+16KiB Minimum Replica Number: 2 Redundancy Overlap:1 Capacity: 202 GiB iSCSI Target: target01 Local Sector Size: 4096 bytes High Availability: ActiveStandby Write Policy: WriteBack Object Storage Info: +-----+-----+ Provider OOS Bucket Name hblocktest3 Endpoint https://oos-cn.ctyunapi.cn Signature Version v2 Region Storage Class STANDARD Access Key 8f129a5529f202811fd0 Object Size 1024 KiB Compression Enabled </pre>	
---	--

```
+-----+
Y
Start to recover LUN lun01a.
You can list lun to check whether the recovery is completed.
```

4.6.10 恢复还原中断的卷（上云卷适用）

```
./stor lun { RS | --resume } { -n | --name } LUN_NAME
```

此命令用来恢复还原中断的 HBlock 上云卷。

前置条件：还原的卷因为某些原因中断，且没有对还原卷数据做任何修改。

参数

参数	描述
-n LUN_NAME 或 --name <i>LUN_NAME</i>	指定需要继续还原的卷的名称。

示例

恢复还原中断的卷。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun RS -n lun04a
Starting to recover LUN lun04a from the breakpoint.
You can list lun to check whether the recovery is completed.
```

4.6.11 查询卷信息

```
./stor lun ls [ { -n | --name } LUN_NAME [ { -s | --snapshot } SNAPSHOT_FORMAT ] ] [ --clone ] [ --wwid ]
```

此命令用来查询卷信息。

参数

参数	描述
-n LUN_NAME 或--name LUN_NAME	指定要查询的卷名称。
-s SNAPSHOT_FORMAT 或--snapshot SNAPSHOT_FORMAT	快照的展示方式： ● list: 展示卷快照的详细信息。 ● tree: 以树形展示快照信息。 说明 : 只有指定卷名称，才能输出卷快照信息。
--clone	查询所有的克隆卷。
--wwid	指定查询卷的唯一标识符。 如果客户端连接卷的时候，HBlock 端有多个卷，可以通过 WWID 标识符来确认所要连接的卷。

示例

- 单机版：查询所有卷信息。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun ls
```

No.	LUN Name	Storage Mode	Capacity	Status	Target	Snapshot Numbers	Is Clone
1.	luna1 (LUN 0)	Local	100 GiB	Normal	iqn.2012-08.cn.ctyunapi.oos:targeta.1 (192.168.0.66:3260,Active)	4	
2.	luna1-C1 (LUN 0)	Local	200 GiB	Normal	iqn.2012-08.cn.ctyunapi.oos:targeta.1.4 (192.168.0.66:3260,Active)	1	clone
3.	luna2 (LUN 1)	Local	88 GiB	Normal	iqn.2012-08.cn.ctyunapi.oos:targeta.1 (192.168.0.66:3260,Active)	1	
4.	lunb1 (LUN 0)	Cache	200 GiB	Normal	iqn.2012-08.cn.ctyunapi.oos:targetb.2 (192.168.0.66:3260,Active)	0	

5.	lunc1 (LUN 0)	Storage	300 GiB	Normal	iqn.2012-08.cn.ctyunapi.oos:targetc.3 (192.168.0.66:3260,Active)	0		
----	---------------	---------	---------	--------	--	---	--	--

● 单机版：查询所有的克隆卷。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun ls --clone
```

No.	LUN Name	Storage Mode	Capacity	Status	Target	Snapshot Numbers	Is Clone
1.	luna1-C1 (LUN 0)	Local	200 GiB	Normal	iqn.2012-08.cn.ctyunapi.oos:targeta1.4 (192.168.0.66:3260,Active)	1	clone

● 单机版：查询所有卷的标识符。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun ls --wwid
```

No.	LUN Name	Storage Mode	Capacity	Status	Target	Snapshot Numbers	Is Clone	WWID
1.	luna1 (LUN 0)	Local	100 GiB	Normal	iqn.2012-08.cn.ctyunapi.oos:targeta.1 (192.168.0.66:3260,Active)	4		3300000032b51acf
2.	luna1-C1 (LUN 0)	Local	200 GiB	Normal	iqn.2012-08.cn.ctyunapi.oos:targeta1.4 (192.168.0.66:3260,Active)	1	clone	3300000066d85e04
3.	luna2 (LUN 1)	Local	88 GiB	Normal	iqn.2012-08.cn.ctyunapi.oos:targeta.1 (192.168.0.66:3260,Active)	1		33fffffffd281600
4.	lunb1 (LUN 0)	Cache	200 GiB	Normal	iqn.2012-08.cn.ctyunapi.oos:targetb.2 (192.168.0.66:3260,Active)	0		33ffffff9449437a
5.	lunc1 (LUN 0)	Storage	300 GiB	Normal	iqn.2012-08.cn.ctyunapi.oos:targetc.3 (192.168.0.66:3260,Active)	0		3300000033c8867b

● 集群版：查询所有卷信息。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun ls
```

No.	LUN Name	Storage Mode	Capacity	Local Storage Class	Minimum Replica Number	Status	Target	Snapshot Numbers	Is Clone
1.	clone-lun02-2 (LUN 0)	Local	100 GiB	EC 2+1+16KiB	2	Normal	iqn.2012-08.cn.ctyunapi.oos:target02.3 (192.168.0.65:3260,Active)	1	
							iqn.2012-08.cn.ctyunapi.oos:target02.4 (192.168.0.64:3260,Standby)		
2.	clone-lun02-3 (LUN 2)	Local	100 GiB	EC 2+1+16KiB	2	Normal	iqn.2012-08.cn.ctyunapi.oos:target002.9 (192.168.0.65:3260,Active)	1	clone
							iqn.2012-08.cn.ctyunapi.oos:target002.10 (192.168.0.64:3260,Standby)		
3.	clone-lun02-4 (LUN 0)	Local	101 GiB	EC 2+1+16KiB	2	Normal	iqn.2012-08.cn.ctyunapi.oos:target2-4.17 (192.168.0.65:3260,Active)	0	
							iqn.2012-08.cn.ctyunapi.oos:target2-4.18 (192.168.0.64:3260,Standby)		
4.	clone-lun02-5 (LUN 1)	Local	110 GiB	EC 2+1+16KiB	2	Normal	iqn.2012-08.cn.ctyunapi.oos:target02.3 (192.168.0.65:3260,Active)	0	clone

								iqn.2012-08.cn.ctyunapi.oos:target02.4 (192.168.0.64:3260,Standby)			
5.	demo1un2 (LUN 1)	Local	210 GiB	EC 2+1+16KiB	2		Normal	iqn.2012-08.cn.ctyunapi.oos:target-demo.13 (192.168.0.65:3260,Active)	0		
								iqn.2012-08.cn.ctyunapi.oos:target-demo.14 (192.168.0.64:3260,Standby)			
6.	lun-test-01 (LUN 0)	Local	10 GiB	EC 2+1+16KiB	2		Normal	iqn.2012-08.cn.ctyunapi.oos:target-test-01.11 (192.168.0.65:3260,Active)	1		
								iqn.2012-08.cn.ctyunapi.oos:target-test-01.12 (192.168.0.67:3260,Standby)			
7.	lun001 (LUN 1)	Local	100 GiB	EC 2+1+16KiB	2		Normal	iqn.2012-08.cn.ctyunapi.oos:target01.2 (192.168.0.64:3260,Active)	0		
								iqn.2012-08.cn.ctyunapi.oos:target01.1 (192.168.0.67:3260,Standby)			
8.	lun01 (LUN 0)	Local	100 GiB	EC 2+1+16KiB	2		Normal	iqn.2012-08.cn.ctyunapi.oos:target001.8 (192.168.0.67:3260,Active)	1		
								iqn.2012-08.cn.ctyunapi.oos:target001.7 (192.168.0.65:3260,Standby)			
9.	lun01a (LUN 0)	Local	100 GiB	EC 2+1+16KiB	2		Normal	iqn.2012-08.cn.ctyunapi.oos:target01.2 (192.168.0.64:3260,Active)	3		
								iqn.2012-08.cn.ctyunapi.oos:target01.1 (192.168.0.67:3260,Standby)			
10.	lun01a-clone (LUN 8)	Local	100 GiB	EC 2+1+16KiB	2		Normal	iqn.2012-08.cn.ctyunapi.oos:target01.2 (192.168.0.64:3260,Active)	0	clone	
								iqn.2012-08.cn.ctyunapi.oos:target01.1 (192.168.0.67:3260,Standby)			
11.	lun01clon (LUN 7)	Local	100 GiB	EC 2+1+16KiB	2		Normal	iqn.2012-08.cn.ctyunapi.oos:target01.2 (192.168.0.64:3260,Active)	0	clone	
								iqn.2012-08.cn.ctyunapi.oos:target01.1 (192.168.0.67:3260,Standby)			
12.	lun01a-C1 (LUN 12)	Local	100 GiB	EC 2+1+16KiB	2		Normal	iqn.2012-08.cn.ctyunapi.oos:target01.1 (192.168.0.67:3260,Active)	0	clone	
								iqn.2012-08.cn.ctyunapi.oos:target01.2 (192.168.0.64:3260,Standby)			
13.	lun01a-C2 (LUN 13)	Local	100 GiB	EC 2+1+16KiB	2		Normal	iqn.2012-08.cn.ctyunapi.oos:target01.1 (192.168.0.67:3260,Active)	0	clone	
								iqn.2012-08.cn.ctyunapi.oos:target01.2 (192.168.0.64:3260,Standby)			
14.	lun01a-clone2 (LUN 2)	Local	100 GiB	EC 2+1+16KiB	2		Normal	iqn.2012-08.cn.ctyunapi.oos:target001.7 (192.168.0.65:3260,Active)	0	clone	
								iqn.2012-08.cn.ctyunapi.oos:target001.8 (192.168.0.67:3260,Standby)			
15.	lun01a20250304 (LUN 11)	Local	100 GiB	EC 2+1+16KiB	2		Normal	iqn.2012-08.cn.ctyunapi.oos:target01.2 (192.168.0.64:3260,Active)	0	clone	
								iqn.2012-08.cn.ctyunapi.oos:target01.1 (192.168.0.67:3260,Standby)			
16.	lun002 (LUN 2)	Local	200 GiB	EC 2+1+16KiB	2		Normal	iqn.2012-08.cn.ctyunapi.oos:target01.2 (192.168.0.64:3260,Active)	0		
								iqn.2012-08.cn.ctyunapi.oos:target01.1 (192.168.0.67:3260,Standby)			
17.	lun02 (LUN 0)	Local	100 GiB	EC 2+1+16KiB	2		Normal	iqn.2012-08.cn.ctyunapi.oos:target002.9 (192.168.0.65:3260,Active)	4		
								iqn.2012-08.cn.ctyunapi.oos:target002.10 (192.168.0.64:3260,Standby)			
18.	lun003 (LUN 3)	Local	300 GiB	EC 2+1+16KiB	2		Normal	iqn.2012-08.cn.ctyunapi.oos:target01.2 (192.168.0.64:3260,Active)	0		
								iqn.2012-08.cn.ctyunapi.oos:target01.1 (192.168.0.67:3260,Standby)			
19.	lun03 (LUN 1)	Local	100 GiB	EC 2+1+16KiB	2		Normal	iqn.2012-08.cn.ctyunapi.oos:target002.10 (192.168.0.64:3260,Active)	1		
								iqn.2012-08.cn.ctyunapi.oos:target002.9 (192.168.0.65:3260,Standby)			
20.	lun004 (LUN 4)	Local	400 GiB	EC 2+1+16KiB	2		Normal	iqn.2012-08.cn.ctyunapi.oos:target01.2 (192.168.0.64:3260,Active)	0		
								iqn.2012-08.cn.ctyunapi.oos:target01.1 (192.168.0.67:3260,Standby)			
21.	lun04a (LUN 0)	Cache	220 GiB	EC 2+1+16KiB	2		Normal	iqn.2012-08.cn.ctyunapi.oos:target04.15 (192.168.0.65:3260,Active)	0		
								iqn.2012-08.cn.ctyunapi.oos:target04.16 (192.168.0.64:3260,Standby)			

- 集群版：查询所有克隆卷。

```
[root@hblockserver CTYUN HBlock Plus_3.9.0 x64]# ./stor lun ls --clone
```

No.	LUN Name	Storage Mode	Capacity	Local Storage Class	Minimum Replica Number	Status	Target	Snapshot Numbers	Is Clone
1.	clone-lun02-3 (LUN 2)	Local	100 GiB	EC 2+1+16KiB	2	Normal	iqn.2012-08.cn.ctyunapi.oos:target002.9 (192.168.0.65:3260,Active)	1	clone
							iqn.2012-08.cn.ctyunapi.oos:target002.10 (192.168.0.64:3260,Standby)		
2.	clone-lun02-5 (LUN 1)	Local	110 GiB	EC 2+1+16KiB	2	Normal	iqn.2012-08.cn.ctyunapi.oos:target02.3 (192.168.0.65:3260,Active)	0	clone
							iqn.2012-08.cn.ctyunapi.oos:target02.4 (192.168.0.64:3260,Standby)		
3.	lun01a-clone (LUN 8)	Local	100 GiB	EC 2+1+16KiB	2	Normal	iqn.2012-08.cn.ctyunapi.oos:target01.2 (192.168.0.64:3260,Active)	0	clone
							iqn.2012-08.cn.ctyunapi.oos:target01.1 (192.168.0.67:3260,Standby)		
4.	lun01clon (LUN 7)	Local	100 GiB	EC 2+1+16KiB	2	Normal	iqn.2012-08.cn.ctyunapi.oos:target01.2 (192.168.0.64:3260,Active)	0	clone
							iqn.2012-08.cn.ctyunapi.oos:target01.1 (192.168.0.67:3260,Standby)		
5.	lun01a-C1 (LUN 12)	Local	100 GiB	EC 2+1+16KiB	2	Normal	iqn.2012-08.cn.ctyunapi.oos:target01.1 (192.168.0.67:3260,Active)	0	clone
							iqn.2012-08.cn.ctyunapi.oos:target01.2 (192.168.0.64:3260,Standby)		
6.	lun01a-C2 (LUN 13)	Local	100 GiB	EC 2+1+16KiB	2	Normal	iqn.2012-08.cn.ctyunapi.oos:target01.1 (192.168.0.67:3260,Active)	0	clone
							iqn.2012-08.cn.ctyunapi.oos:target01.2 (192.168.0.64:3260,Standby)		
7.	lun01a-clone2 (LUN 2)	Local	100 GiB	EC 2+1+16KiB	2	Normal	iqn.2012-08.cn.ctyunapi.oos:target001.7 (192.168.0.65:3260,Active)	0	clone
							iqn.2012-08.cn.ctyunapi.oos:target001.8 (192.168.0.67:3260,Standby)		
8.	lun01a20250304 (LUN 11)	Local	100 GiB	EC 2+1+16KiB	2	Normal	iqn.2012-08.cn.ctyunapi.oos:target01.2 (192.168.0.64:3260,Active)	0	clone
							iqn.2012-08.cn.ctyunapi.oos:target01.1 (192.168.0.67:3260,Standby)		

● 集群版：查询所有卷的标识符信息。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun ls --wwid
```

No.	LUN Name	Storage Mode	Capacity	Local Storage Class	Minimum Replica Number	Status	Target	Snapshot Numbers	Is Clone	WWID
1.	clone-lun02-2 (LUN 0)	Local	100 GiB	EC 2+1+16KiB	2	Normal	iqn.2012-08.cn.ctyunapi.oos:target02.3 (192.168.0.65:3260,Active)	1		33ffffffb26e7cbf
							iqn.2012-08.cn.ctyunapi.oos:target02.4 (192.168.0.64:3260,Standby)			
2.	clone-lun02-3 (LUN 2)	Local	100 GiB	EC 2+1+16KiB	2	Normal	iqn.2012-08.cn.ctyunapi.oos:target002.9 (192.168.0.65:3260,Active)	1	clone	330000000ba19dfa
							iqn.2012-08.cn.ctyunapi.oos:target002.10 (192.168.0.64:3260,Standby)			
3.	clone-lun02-4 (LUN 0)	Local	101 GiB	EC 2+1+16KiB	2	Normal	iqn.2012-08.cn.ctyunapi.oos:target2-4.17 (192.168.0.65:3260,Active)	0		33ffffff9eb8f604
							iqn.2012-08.cn.ctyunapi.oos:target2-4.18 (192.168.0.64:3260,Standby)			
4.	clone-lun02-5 (LUN 1)	Local	110 GiB	EC 2+1+16KiB	2	Normal	iqn.2012-08.cn.ctyunapi.oos:target02.3 (192.168.0.65:3260,Active)	0	clone	330000000209228f9
							iqn.2012-08.cn.ctyunapi.oos:target02.4 (192.168.0.64:3260,Standby)			
5.	demolun2 (LUN 1)	Local	210 GiB	EC 2+1+16KiB	2	Normal	iqn.2012-08.cn.ctyunapi.oos:target-demo.13 (192.168.0.65:3260,Active)	0		3300000004783f5f6
							iqn.2012-08.cn.ctyunapi.oos:target-demo.14 (192.168.0.64:3260,Standby)			
6.	lun-test-01 (LUN 0)	Local	10 GiB	EC 2+1+16KiB	2	Normal	iqn.2012-08.cn.ctyunapi.oos:target-test-01.11 (192.168.0.65:3260,Active)	1		33ffffffb3889df2
							iqn.2012-08.cn.ctyunapi.oos:target-test-01.12 (192.168.0.67:3260,Standby)			
7.	lun001 (LUN 1)	Local	100 GiB	EC 2+1+16KiB	2	Normal	iqn.2012-08.cn.ctyunapi.oos:target01.2 (192.168.0.64:3260,Active)	0		3300000001e4b5b12

查询所有卷信息描述

147

	LUN，LUN 的编号一般为 0。
Storage Mode	卷的存储类型： ● 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对象存储中。 ● Storage：存储模式，本地保留全部数据，并异步存储到对象存储中。
Capacity	卷容量。
Local Storage Class	卷冗余模式（仅集群版支持）： ● single-copy：单副本。 ● 2-copy：两副本。 ● 3-copy：三副本。 ● EC $N+M$：分片大小。
Minimum Replica Number	最小副本数（仅集群版支持）。
Status	卷的状态： ● Normal：正常。 ● Deleting：卷正在删除中。 ● DeleteFailed：卷删除失败。 ● Recovering：卷正在还原中。 ● RecoverFailed：卷还原失败。 ● Rollbacking：卷正在回滚。 ● Flattening：与快照的关系链断开过程中，表示克隆卷正在复制源卷的数据，复制完成之后将变成独立卷，不再依赖快照和源卷。
Target	卷关联的 Target，包括 Target IQN、Target IP、Target 端口号、卷对应的 Target 的状态（Active：主 Target；Standby：热备 Target；Offline：离线 Target；

	ColdStandby: 冷备 Target)。
Snapshot Numbers	卷的快照个数。
Is Clone	该卷是否为克隆卷: ● clone: 表示卷为克隆卷。 ● 不显示表示该卷不是克隆卷。
WWID	卷的唯一标识符。 如果客户端连接卷的时候, HBlock 端有多个卷, 可以通过 WWID 标识符来确认所要连接的卷。

- 单机版本地卷: 查询卷 luna1(非克隆卷)的信息。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun ls -n luna1
LUN Name: luna1 (LUN 0)
Storage Mode: Local
Capacity: 100 GiB
Status: Normal
iSCSI Target: iqn.2012-08.cn.ctyunapi.oos:targeta.1 (192.168.0.66:3260,Active)
Create Time: 2025-02-27 16:48:31
Local Sector Size: 4096 bytes
Write Policy: WriteBack
WWID: 33000000032b51acf
UUID: lun-uuid-2ffdaa5e-da3e-400f-84fd-7bd46ed34fb1
Path: /mnt/stor01
Snapshot Numbers: 4
```

- 单机版本地卷: 查询卷 luna1-C1(克隆卷)的信息。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun ls -n luna1-C1
LUN Name: luna1-C1 (LUN 0)
Storage Mode: Local
Capacity: 200 GiB
Status: Normal
iSCSI Target: iqn.2012-08.cn.ctyunapi.oos:targeta1.4 (192.168.0.66:3260,Active)
```

```
Create Time: 2025-03-10 11:55:32
Local Sector Size: 4096 bytes
Write Policy: WriteBack
WWID: 33000000066d85e04
UUID: lun-uuid-4beffea2-05a1-4ba9-8e21-06337a1e57e5
Path: /mnt/stor01
Source Snapshot: luna1-snapshot
Source LUN: luna1
Snapshot Numbers: 1
```

- 单机版上云卷：查询卷 lunb1 的信息。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun ls -n lunb1
LUN Name: lunb1 (LUN 0)
Storage Mode: Cache
Capacity: 200 GiB
Status: Normal
iSCSI Target: iqn.2012-08.cn.ctyunapi.oos:targetb.2 (192.168.0.66:3260,Active)
Create Time: 2025-02-27 18:04:14
Local Sector Size: 4096 bytes
Write Policy: WriteBack
WWID: 33ffffffff9449437a
UUID: lun-uuid-fc9a2f2c-c8bd-462b-8684-49e6e1650b1c
Path: /mnt/stor01
Object Storage Info:
+-----+-----+
| Provider      | OOS          |
| Bucket Name   | hblocktest3  |
| Prefix        | lunb         |
| Endpoint      | https://oos-cn.ctyunapi.cn |
| Signature Version | v4          |
| Region        | cn           |
| Storage Class | STANDARD     |
| Access Key    | 8f129a5529f202811fd0 |
| Object Size   | 1024 KiB     |
```

Compression	Enabled	
+-----+-----+		

- 集群版本地卷：查询卷 **lun002**（非克隆卷）的信息。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun ls -n lun002
LUN Name: lun002 (LUN 2)
Storage Mode: Local
Capacity: 200 GiB
Status: Normal
Auto Failback: Enabled
iSCSI Target: iqn.2012-08.cn.ctyunapi.oos:target01.2 (192.168.0.64:3260,Active)
               iqn.2012-08.cn.ctyunapi.oos:target01.1 (192.168.0.67:3260,Standby)
Create Time: 2025-02-19 17:07:55
Local Storage Class: EC 2+1+16KiB
Minimum Replica Number: 2
Redundancy Overlap: 1
Local Sector Size: 4096 bytes
Storage Pool: default
High Availability: ActiveStandby
Write Policy: WriteBack
WWID: 330000001088ec52
UUID: lun-uuid-62d68861-57c9-49c6-980e-6b67b2f053d2
Snapshot Numbers: 0
```

- 集群版本地卷：查询卷 **clone-lun02-3**（克隆卷）的信息。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun ls -n clone-lun02-3
LUN Name: clone-lun02-3 (LUN 2)
Storage Mode: Local
Capacity: 100 GiB
Status: Normal
Auto Failback: Enabled
iSCSI Target: iqn.2012-08.cn.ctyunapi.oos:target002.9 (192.168.0.65:3260,Active)
               iqn.2012-08.cn.ctyunapi.oos:target002.10 (192.168.0.64:3260,Standby)
Create Time: 2025-03-06 12:02:30
```

```
Local Storage Class: EC 2+1+16KiB
Minimum Replica Number: 2
Redundancy Overlap: 1
Local Sector Size: 4096 bytes
Storage Pool: default
High Availability: ActiveStandby
Write Policy: WriteBack
WWID: 3300000000ba19dfa
UUID: lun-uuid-c4f1e777-1c8a-49d1-b794-e17edcf7c3ef
Source Snapshot: snapshot-lun02-3
Source LUN: lun02
Snapshot Numbers: 1
```

- 集群版上云卷：查询卷 **lun04a** 的信息。

```
[root@hbblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun ls -n lun04a
LUN Name: lun04a (LUN 0)
Storage Mode: Cache
Capacity: 220 GiB
Status: Normal
Auto Failback: Enabled
iSCSI Target: iqn.2012-08.cn.ctyunapi.oos:target04.15 (192.168.0.65:3260,Active)
               iqn.2012-08.cn.ctyunapi.oos:target04.16 (192.168.0.64:3260,Standby)
Create Time: 2025-02-28 10:04:17
Local Storage Class: EC 2+1+16KiB
Minimum Replica Number: 2
Redundancy Overlap: 1
Local Sector Size: 4096 bytes
Storage Pool: default
High Availability: ActiveStandby
Write Policy: WriteBack
WWID: 33fffffffff4f9099b
UUID: lun-uuid-194c91ec-1dd5-4d12-94ce-7bab1a898362
Object Storage Info:
+-----+-----+
```

Provider	OOS	
Bucket Name	hblocktest3	
Endpoint	https://oos-cn.ctyunapi.cn	
Signature Version	v2	
Region		
Storage Class	STANDARD	
Access Key	8f129a5529f202811fd0	
Object Size	1024 KiB	
Compression	Enabled	
+-----+-----+		

- 集群版上云卷：查询卷 **lun04a**（还原失败）的信息。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun ls -n lun04a
LUN Name: lun04a (LUN 0)
Storage Mode: Storage
Capacity: 440 GiB
Status: RecoverFailed
Failed Reason: Unknown Error, please resume LUN recovery later or contact your software vendor.
Auto Failback: Enabled
iSCSI Target: iqn.2012-08.cn.ctyunapi.oos:target04.7(192.168.0.102:3260,Offline)
               iqn.2012-08.cn.ctyunapi.oos:target04.8(192.168.0.117:3260,Offline)
Create Time: 2024-12-17 17:25:37
Local Storage Class: EC 2+1+16KiB
Minimum Replica Number: 2
Redundancy Overlap: 1
Local Sector Size: 4096 bytes
Cache Storage Pool: pool3
Storage Pool: default
High Availability: ActiveStandby
Write Policy: WriteThrough
WWID: 3300000006d8b38ff
UUID: lun-uuid-2883e7a4-8a34-4ba2-9d4c-00374f43c1e6
Object Storage Info:
```

+-----+-----+		
Provider	OOS	
Bucket Name	hblocktest3	
Endpoint	https://oos-cn.ctyunapi.cn	
Signature Version	v2	
Region		
Storage Class	STANDARD	
Access Key	8f129a5529f202811fd0	
Object Size	1024 KiB	
Compression	Enabled	
+-----+-----+		

查询指定卷的信息描述

项目	描述
LUN Name	包括卷名称和卷编号。括号内容表示卷编号。卷编号： LUN 在 Target 下的编号，由存储系统分配，对应客户端挂载存储设备时设备地址中的 LUN ID。如果 Target 下只有一个 LUN，LUN 的编号一般为 0。
Storage Mode	卷的存储类型： ● 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对象存储中。 ● Storage：存储模式，本地保留全部数据，并异步存储到对象存储中。
Capacity	卷容量。
Status	卷的状态： ● Normal：正常。 ● Deleting：卷正在删除中。 ● DeleteFailed：卷删除失败。

	● Recovering: 卷正在还原中。 ● RecoverFailed: 卷还原失败。 ● Rollbacking: 卷正在回滚。 ● Flattening: 与快照的关系链断开过程中，表示克隆卷正在复制源卷的数据，复制完成之后将变成独立卷，不再依赖快照和源卷。
Failed Reason	失败原因：卷还原失败的原因（卷状态为 RecoverFailed 时显示），或者卷删除失败的原因（卷状态为 DeleteFailed 时显示）。
Auto Failback	针对卷主备状态，当高优先级的服务器恢复正常后，是否自动进行主备状态切换： ● Enabled: 自动进行主备切换。 ● Disabled: 不自动进行主备切换。
iSCSI Target	卷关联的 Target，包括 Target IQN、Target IP、Target 端口号、卷对应的 Target 的状态（ Active: 主 Target； Standby: 热备 Target； offline: 离线 Target； ColdStandby: 冷备 Target）。
Create Time	卷创建的时间。
Local Storage Class	卷冗余模式（仅集群版支持）： ● single-copy: 单副本。 ● 2-copy: 两副本。 ● 3-copy: 三副本。 ● EC $N+M$: 分片大小。
Minimum Replica Number	最小副本数（仅集群版支持）。
Redundancy Overlap	卷的折叠副本数（仅集群版支持）。
Local Sector Size	扇区大小。
Cache Storage Pool	缓存存储池（仅集群版支持）。

Storage Pool	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。
High Availability	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。
Write Policy	卷的写策： ● WriteBack: 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。 ● WriteThrough: 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。 ● WriteAround: 绕写，指数据直接写到磁盘，不写入内存。
WWID	卷的唯一标识符。 如果客户端连接卷的时候，HBlock 端有多个卷，可以通过 WWID 标识符来确认所要连接的卷。
UUID	卷的唯一识别码。
Path	存储卷数据的数据目录（仅单机版支持）。
Source Snapshot	克隆卷关联的快照（仅克隆卷支持）。
Source LUN	克隆卷的源卷名称（仅克隆卷支持）。
Snapshot Numbers	卷的快照个数（仅本地卷支持）。
Object Storage Info	卷上云信息，包括：Provider、Bucket Name、Prefix、Endpoint、Signature Version、Region、Storage Class、Access Key、Object Size、Compression。
Provider	对象存储服务名称： ● OOS: 天翼云对象存储经典版 I 型。

	● S3: 兼容 S3 的其他对象存储。
Bucket Name	存储桶名称。
Prefix	卷数据存储在对象存储的前缀名称。 如果未设置，则不显示此项。
Endpoint	对象存储服务的 Endpoint。
Signature Version	上云签名认证的类型： ● v2: V2 签名认证。 ● v4: V4 签名认证。
Region	对象存储服务的 Endpoint 资源池所在区域。如果是 V2 签名，此处为空。
Storage Class	数据在对象存储的存储类型： ● STANDARD: 标准存储。 ● STANDARD_IA: 低频访问存储。
Access Key	对象存储服务的 Access Key。
Object Size	数据存储在云端的大小。
Compression	是否压缩数据上传至对象存储： ● Enabled: 压缩数据上传至对象存储。 ● Disabled: 不压缩数据上传至对象存储。

- **本地卷**: 查询卷 **lun02** 的快照信息，快照以树形展示。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun ls -n lun02 -s tree
LUN Name: lun02 (LUN 0)
Storage Mode: Local
Capacity: 100 GiB
Status: Normal
Auto Failback: Enabled
iSCSI Target: iqn.2012-08.cn.ctyunapi.oos:target002.9 (192.168.0.65:3260,Active)
               iqn.2012-08.cn.ctyunapi.oos:target002.10 (192.168.0.64:3260,Standby)
Create Time: 2025-02-21 17:03:16
```

```
Local Storage Class: EC 2+1+16KiB
Minimum Replica Number: 2
Redundancy Overlap: 1
Local Sector Size: 4096 bytes
Storage Pool: default
High Availability: ActiveStandby
Write Policy: WriteBack
WWID: 33ffffffffc4dd6c0a
UUID: lun-uuid-996d444f-d824-49c6-abbd-e31863bfd90f
Snapshot Numbers: 4
Snapshots:
snapshot-lun02-1
└── snapshot-lun02-3
    ├── lun02-snap20250326164436 (*)
    └── lun02-snap-4
```

- 本地卷：查询卷 **lun02** 的快照信息，展示快照的详细信息。

```
[root@hbblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun ls -n lun02 -s list
LUN Name: lun02 (LUN 0)
Storage Mode: Local
Capacity: 100 GiB
Status: Normal
Auto Failback: Enabled
iSCSI Target: iqn.2012-08.cn.ctyunapi.oos:target002.9 (192.168.0.65:3260,Active)
               iqn.2012-08.cn.ctyunapi.oos:target002.10 (192.168.0.64:3260,Standby)
Create Time: 2025-02-21 17:03:16
Local Storage Class: EC 2+1+16KiB
Minimum Replica Number: 2
Redundancy Overlap: 1
Local Sector Size: 4096 bytes
Storage Pool: default
High Availability: ActiveStandby
Write Policy: WriteBack
WWID: 33ffffffffc4dd6c0a
UUID: lun-uuid-996d444f-d824-49c6-abbd-e31863bfd90f
```

Snapshot Numbers: 4						
Snapshots:						
No.	Name	Consistency Snapshot Name	LUN Capacity	Reclaim Policy	Status	Create Time
1.	snapshot-lun02-1	-	100 GiB	Retain	Normal	2025-03-04 15:18:10
2.	snapshot-lun02-3	-	100 GiB	Retain	Normal	2025-03-04 15:18:35
3.	lun02-snap20250326164436 (*)	consistencysnapshot8	100 GiB	Retain	Normal	2025-03-26 16:44:36
4.	lun02-snap-4	-	100 GiB	Retain	Normal	2025-03-10 16:03:27

查询指定卷的快照信息

项目	描述
LUN Name	包括卷名称和卷编号。括号内容表示卷编号。卷编号： LUN 在 Target 下的编号，由存储系统分配，对应客户端挂载存储设备时设备地址中的 LUN ID。如果 Target 下只有一个 LUN，LUN 的编号一般为 0。
Storage Mode	卷的存储类型： ● Local：本地模式，数据全部保留在本地。仅本地卷支持快照。
Capacity	卷容量。
Status	卷的状态： ● Normal：正常。 ● Deleting：卷正在删除中。 ● DeleteFailed：卷删除失败。 ● Recovering：卷正在还原中。 ● RecoverFailed：卷还原失败。 ● Rollbacking：卷正在回滚。 ● Flattening：与快照的关系链断开过程中，表示克隆

	卷正在复制源卷的数据，复制完成之后将变成独立卷，不再依赖快照和源卷。
Auto Failback	针对卷主备状态，当高优先级的服务器恢复正常后，是否自动进行主备状态切换： ● Enabled: 自动进行主备切换。 ● Disabled: 不自动进行主备切换。
iSCSI Target	卷关联的 Target，包括 Target IQN、Target IP、Target 端口号、卷对应的 Target 的状态（Active: 主 Target；Standby: 热备 Target；offline: 离线 Target；ColdStandby: 冷备 Target）。
Create Time	卷创建的时间。
Local Storage Class	卷冗余模式（仅集群版支持）： ● single-copy: 单副本。 ● 2-copy: 两副本。 ● 3-copy: 三副本。 ● EC N+M+分片大小。
Minimum Replica Number	最小副本数（仅集群版支持）。
Redundancy Overlap	卷的折叠副本数（仅集群版支持）。
Local Sector Size	扇区大小。
Cache Storage Pool	缓存存储池（仅集群版支持）。
Storage Pool	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。
High Availability	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。

Write Policy	卷的写策： ● WriteBack: 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。 ● WriteThrough: 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。 ● WriteAround: 绕写，指数据直接写到磁盘，不写入内存。
WWID	卷的唯一标识符。 如果客户端连接卷的时候，HBlock 端有多个卷，可以通过 WWID 标识符来确认所要连接的卷。
UUID	卷的唯一识别码。
Path	存储卷数据的数据目录（仅单机版支持）。
Source Snapshot	克隆卷关联的快照（仅克隆卷支持）。
Source LUN	克隆卷的源卷名称（仅克隆卷支持）。
Snapshot Numbers	卷的快照个数（仅本地卷支持）。
Snapshots	卷快照信息。
	快照以详细信息展示，包括下列信息： ● No.: 序号。 ● Name: 快照名称。*: 表示当前的读写是以此快照为基础。 ● Consistency Snapshot Name: 所属的一致性快照名称。仅一致性快照的卷快照显示，“-”表示不是一致性快照的卷快照。 ● LUN Capacity: 创建快照时刻，源卷的容量。 ● Reclaim Policy: 快照回收策略： ■ Delete: 当快照无关联克隆卷，且最多拥有 1 个子节点（子节点表示存在其他快照依赖此快照，或者

	当前写操作基于此快照），快照自动删除。 ■ Retain: 当快照无关联克隆卷，且最多拥有 1 个子节点，快照仍然保留。 ● Status: 快照的状态： ■ Normal: 正常。 ■ Error: 异常。 ■ Pending: 创建中。 ■ Deleting: 删除中。 ● Create Time: 快照的创建时间。
	快照以树形展示，仅包含快照名称，“*”表示当前的读写是以此快照为基础。

4.7 快照操作

4.7.1 创建快照

```
./stor snapshot add { -n | --name } SNAPSHOT_NAME { -l | --lun } LUN_NAME [ --reclaim-policy RECLAIM_POLICY ]
```

此命令用来创建单卷快照。

注意：

- 在执行此操作之前，请确保源卷的所有数据已持久化，即如果源卷已被客户端挂载，需确保客户端的数据都已经同步到卷上，创建快照前：
 - 对于 Linux 客户端：sync 版本大于等于 8.6 版本时（可以通过 `sync --version` 命令查看版本信息），在客户端执行 `sync -f` 命令；sync 版本低于 8.6 版本时，在客户端执行 `sync` 命令。
 - 对于 Windows 客户端：在客户端将源卷对应的磁盘脱机；创建快照后，在客户端将源卷对应的磁盘重新联机。
- 源卷需要处于正常状态。

说明：

- 单卷支持的最大快照数：512。
- 系统支持的最大快照数：100000。
- 单个快照可创建的最大克隆卷数：512。
- 系统支持的最大快照深度：512。

参数

参数	描述
<code>-n SNAPSHOT_NAME</code> 或 <code>--name SNAPSHOT_NAME</code>	指定快照的名称。 取值：字符串形式，长度范围是 1~256，只能由字母、数字、短横线(-)、下划线(_)组成，字母区分大小写，且仅支持以字母或数字开头。

	注意： 快照名称、一致性快照名称及一致性快照的卷快照名称必须唯一，不可相互重复。
-l LUN_NAME 或 --lun LUN_NAME	指定快照对应的源卷。 注意： 仅支持本地模式（Local）的卷。
--reclaim-policy RECLAIM_POLICY	指定快照回收策略。 取值： ● Delete：当快照无关联克隆卷，且最多拥有 1 个子节点（子节点表示存在其他快照依赖此快照，或者当前写操作基于此快照），快照自动删除。 ● Retain：当快照无关联克隆卷，且最多拥有 1 个子节点，快照仍然保留。 默认值为 Retain 。

示例

创建卷 lun01a 的快照 snapshot-lun01a。

说明：可以交互输入快照的描述信息，1~256 位字符串。如果不添加快照的描述信息，直接按回车即可。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor snapshot add -n snapshot-lun01a
-l lun01a
Enter the description for this snapshot, limited to 256 characters:
lun01a's snapshot
Start creating snapshot snapshot-lun01a of LUN lun01a. You can list snapshot to check
whether it is completed.
```

4.7.2 修改快照

```
./stor snapshot set { -n | --name } SNAPSHOT_NAME [ --change-description ] [ --new-name  
NEW_NAME ] [ --reclaim-policy RECLAIM_POLICY ]
```

此命令用来修改单卷快照（包括一致性快照中的卷快照）信息。

参数

参数	描述
-n SNAPSHOT_NAME 或 --name SNAPSHOT_NAME	指定要修改的快照名称。 取值：字符串形式，长度范围是 1~256，只能由字母、数字、短横线(-)、下划线(_)组成，字母区分大小写，且仅支持以字母或数字开头。
--change-description	修改快照的描述信息。
--new-name NEW_NAME	指定快照的新名称。 取值：字符串形式，长度范围是 1~256，只能由字母、数字、短横线(-)、下划线(_)组成，字母区分大小写，且仅支持以字母或数字开头。 注意： 快照名称、一致性快照名称及一致性快照的卷快照名称必须唯一，不可相互重复。
--reclaim-policy RECLAIM_POLICY	修改快照回收策略。 取值： ● Delete: 当快照无关联克隆卷，且最多拥有 1 个子节点（子节点表示存在其他快照依赖此快照，或者当前写操作基于此快照），快照自动删除。 ● Retain: 当快照无关联克隆卷，且最多拥有 1 个子节点，快照仍然保留。

示例

- 修改快照 `snapshot-lun01a` 的描述信息。

说明：交互输入快照的描述信息，1~256 位字符串。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor snapshot set -n snapshot-lun01a
--change-description
Enter the description for this snapshot, limited to 256 characters:
The first snapshot of lun01a.
Set snapshot snapshot-lun01a successfully
```

- 修改快照 `snapshot-lun01a` 的名称为 `snapshot-lun01a-1`，保留策略为 `Delete`。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor snapshot set -n snapshot-lun01a
--new-name snapshot-lun01a-1 --reclaim-policy Delete
Set snapshot snapshot-lun01a successfully.
```

4.7.3 回滚快照

```
./stor snapshot { R | rollback } { -n | --name } SNAPSHOT_NAME
```

此命令用来回滚单卷快照（包括一致性快照中的卷快照）。

注意：

- 此操作会将快照中的数据覆盖到源卷，建议对源卷创建新的快照进行数据备份。
- 如果快照的源卷是一个克隆卷，并且此克隆卷正在执行断开关系链的操作时，不能使用这个克隆卷的任何快照进行回滚。
- 如果源卷有快照正在创建，不能执行回滚操作。
- 如果源卷已被客户端挂载，需在客户端取消挂载卷后回滚，回滚后重新挂载卷：

■ 对于 Linux 客户端，可以执行下列步骤：

1. 回滚快照前，客户端执行命令：

```
umount DIRECTORY_NAME_OR_PATH
```

2. 回滚快照后，客户端执行命令

```
mount /dev/sdx DIRECTORY_NAME_OR_PATH
```

■ 对于 Windows 客户端，可以执行下列步骤：

1. 回滚快照前，在客户端将源卷对应的磁盘脱机。
2. 回滚快照后，在客户端将源卷对应的磁盘重新联机。

说明：处于回滚状态的卷：

- 不能创建快照。
- 不能再次回滚。
- 不能删除该卷正在回滚的快照。
- 不能编辑该快照。
- 不能修改该卷。
- 不能扩容该卷。
- 如果此卷是克隆卷，不能执行断开关系链操作。

参数

参数	描述
-n SNAPSHOT_NAME 或 --name SNAPSHOT_NAME	指定快照的名称。 取值：字符串形式，长度范围是 1~256，只能由字母、数字、短横线(-)、下划线(_)组成，字母区分大小写，且仅支持以字母或数字开头。

示例

回滚快照 snapshot-lun01a-2。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor snapshot R -n snapshot-lun01a-2
DANGER: This operation will overwrite data on the source LUN with that on the snapshot.
Start rollbacking snapshot snapshot-lun01a-2 on LUN lun01a.
```

4.7.4 删除快照

```
./stor snapshot rm { -n | --name } SNAPSHOT_NAME
```

此命令用来删除单卷快照（包括一致性快照中的卷快照）。

注意：

- 如果快照存在未断开关系链的克隆卷，则无法进行删除。
- 如果快照拥有 2 个及 2 个以上子节点，则无法删除。子节点表示存在其他快照依赖此快照，或者当前写操作基于此快照。
- 快照处于删除状态时，仅支持对该快照的查询、再次删除操作。

参数

参数	描述
-n SNAPSHOT_NAME 或 --name SNAPSHOT_NAME	要删除的卷快照名称。

示例

删除 snapshot-lun01a-2。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor snapshot rm -n snapshot-lun01a-2
WARNING: This operation cannot be undone and will delete all snapshot information.
Start deleting snapshot snapshot-lun01a-2.
```

4.7.5 查询快照

```
./stor snapshot ls [ { -n | --name } SNAPSHOT_NAME ]
```

此命令用来查询快照信息。

参数

参数	描述
-n SNAPSHOT_NAME 或 --name SNAPSHOT_NAME	指定要查询的快照名称。

示例

- 查询所有快照的信息。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor snapshot ls
```

No.	Name	LUN Name	LUN Capacity	Consistency Snapshot Name	Clone LUN Numbers	Reclaim Policy	Status
1.	lun01a-snap20250304161817	lun01a	100 GiB	consistencysnapshot1	0	Retain	Normal
2.	lun01a-snap20250304162045	lun01a	100 GiB	consistencysnapshot2	0	Retain	Normal
3.	lun02-snap20250304162045	lun02	100 GiB	consistencysnapshot2	0	Retain	Normal
4.	snapshot-lun01-1	lun01	100 GiB	-	0	Retain	Normal
5.	snapshot-lun01a-1	lun01a	100 GiB	-	0	Delete	Normal
6.	snapshot-lun02-1	lun02	100 GiB	-	0	Retain	Normal
7.	snapshot-lun02-2	lun02	100 GiB	-	1	Retain	Normal

查询所有快照信息描述

项目	描述
No.	序号。
Name	快照名称。
LUN Name	源卷的名称。
LUN Capacity	创建快照时刻，源卷的容量。
Consistency Snapshot	所属的一致性快照名称。

Name	说明：仅一致性快照的卷快照显示，“-”表示不是一致性快照的卷快照。
Clone LUN Numbers	克隆卷的个数。
Reclaim Policy	快照回收策略： ● Delete：当快照无关联克隆卷，且最多拥有 1 个子节点（子节点表示存在其他快照依赖此快照，或者当前写操作基于此快照），快照自动删除。 ● Retain：当快照无关联克隆卷，且最多拥有 1 个子节点，快照仍然保留。
Status	快照的状态： ● Normal：正常。 ● Error：异常。 ● Pending：创建中。 ● Deleting：删除中。

- 查询快照 snapshot-lun02-2 的信息。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor snapshot ls -n snapshot-lun02-2
Name: snapshot-lun02-2
Description: The second snapshot of lun02.
LUN Name: lun02
LUN Capacity: 100 GiB
Reclaim Policy: Retain
Status: Normal
Create Time: 2025-03-04 15:18:35
Clones:
+-----+-----+-----+-----+
| No. | Name           | Capacity | Status |
+-----+-----+-----+-----+
| 1.  | clone-lun02-2 | 100 GiB  | Normal |
+-----+-----+-----+-----+
```

查询一致性快照的卷快照 lun02-snap20250304162045 的信息。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor snapshot ls -n lun02-snap20250304162045
Name: lun02-snap20250304162045
LUN Name: lun02
LUN Capacity: 100 GiB
Reclaim Policy: Retain
Consistency Snapshot Name: consistencysnapshot2
Status: Normal
Create Time: 2025-03-04 16:20:45
```

查询单个快照信息描述

项目	描述
Name	快照名称。
Description	快照描述信息。
LUN Name	源卷的名称。
LUN Capacity	创建快照时刻，源卷的容量。
Reclaim Policy	快照回收策略： ● Delete: 当快照无关联克隆卷，且最多拥有 1 个子节点（子节点表示存在其他快照依赖此快照，或者当前写操作基于此快照），快照自动删除。 ● Retain: 当快照无关联克隆卷，且最多拥有 1 个子节点，快照仍然保留。
Consistency Snapshot Name	所属的一致性快照名称。 说明：仅一致性快照的卷快照显示。
Status	快照的状态： ● Normal: 正常。 ● Error: 异常。 ● Pending: 创建中。 ● Deleting: 删除中。
Create Time	快照的创建时间。

Clones	快照的克隆卷信息： ● No.: 序号。 ● Name: 克隆卷名称。 ● Capacity: 克隆卷容量。 ● Status: 克隆卷的状态： ■ Normal: 正常。 ■ Deleting: 卷正在删除中。 ■ DeleteFailed: 卷删除失败。 ■ Flattening: 与快照的关系链断开过程中，表示克隆卷正在复制源卷的数据，复制完成之后将变成独立卷，不再依赖快照和源卷。
--------	---

4.8 一致性快照操作

4.8.1 创建一致性快照

```
./stor conssnap add { -n | --name } CONSISTENCYSNAPSHOT_NAME { -l | --lun } LUN_NAME
&<1-n> [ --reclaim-policy RECLAIM_POLICY ]
```

此命令用来创建一致性快照。

一致性快照是指在某一特定时刻，同时对选中的所有卷创建快照，从而确保这些快照反映的是同一时间点的数据状态。

通过一致性快照创建的卷快照，默认命名为：卷名称-snap 创建快照的时间戳（秒级），例如：lun01-snap20240601120000。用户无法在创建时自定义名称，但可以通过**修改快照**命令修改快照名称。

说明：一致性快照添加的最大卷个数是 512。

注意：

- 在执行此操作之前，请确保源卷的所有数据已持久化，即如果源卷已被客户端挂载，需确保客户端的数据都已经同步到卷上，创建一致性快照前：
 - 对于 Linux 客户端：sync 版本大于等于 8.6 版本时（可以通过 `sync --version` 命令查看版本信息），在客户端执行 `sync -f` 命令；sync 版本低于 8.6 版本时，在客户端执行 `sync` 命令。
 - 对于 Windows 客户端：在客户端将源卷对应的磁盘脱机；创建快照后，在客户端将源卷对应的磁盘重新联机。
- 创建一致性快照时，源卷需要处于 **Normal** 状态。

参数

参数	描述
<code>-n CONSISTENCYSNAPSHOT_NAME</code> 或 <code>--name</code> <code>CONSISTENCYSNAPSHOT_NAME</code>	指定一致性快照的名称。 取值：字符串形式，长度范围是 1~256，只能由字母、数字、短横线(-)、下划线(_)组成，字母区分大小写，且仅

	支持以字母或数字开头。 注意：快照名称、一致性快照名称及一致性快照的卷快照名称必须唯一，不可相互重复。
-l LUN_NAME &<1-n>或--lun LUN_NAME &<1-n>	指定一次性快照对应的源卷，一次可以指定多个卷，以英文逗号隔开。 说明：最多可以指定 512 个源卷。 注意：仅支持本地模式（Local）的卷。
--reclaim-policy RECLAIM_POLICY	一致性快照内卷快照的回收策略。 取值： ● Delete: 当快照无关联克隆卷，且最多拥有 1 个子节点（子节点表示存在其他快照依赖此快照，或者当前写操作基于此快照），快照自动删除。 ● Retain: 当快照无关联克隆卷，且最多拥有 1 个子节点，快照仍然保留。 默认值为 Retain。

示例

创建卷 lun01、lun01a、lun02、lun03 的一致性快照 consistencysnapshot3。

说明：可以交互输入快照的描述信息，1~256 位字符串。如果不添加一致性快照的描述信息，直接按回车即可。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor conssnap add -n
consistencysnapshot3 -l lun01,lun01a,lun02,lun03
Enter the description for this consistency snapshot, limited to 256 characters:
The consistencysnapshot of lun01,lun01a,lun02,lun03.
Start creating consistency snapshot consistencysnapshot3. You can list consistency
snapshot to check whether it is completed.
```

4.8.2 修改一致性快照

```
./stor conssnap set { -n | --name } CONSISTENCYSNAPSHOT_NAME [ --change-description ]  
[ --new-name NEW_NAME ] [ --reclaim-policy RECLAIM_POLICY ]
```

此命令用来修改快照信息。

参数

参数	描述
-n <i>CONSISTENCYSNAPSHOT_NAME</i> 或 --name <i>CONSISTENCYSNAPSHOT_NAME</i>	指定一致性快照的名称。 取值：字符串形式，长度范围是 1~256，只能由字母、数字、短横线(-)、下划线(_)组成，字母区分大小写，且仅支持以字母或数字开头。
--change-description	修改一致性快照的描述信息。
--new-name <i>NEW_NAME</i>	指定一致性快照的新名称。 取值：字符串形式，长度范围是 1~256，只能由字母、数字、短横线(-)、下划线(_)组成，字母区分大小写，且仅支持以字母或数字开头。
--reclaim-policy <i>RECLAIM_POLICY</i>	修改一致性快照内单卷快照的回收策略。 取值： ● Delete 当快照无关联克隆卷，且最多拥有 1 个子节点（子节点表示存在其他快照依赖此快照，或者当前写操作基于此快照），快照自动删除。 ● Retain：当快照无关联克隆卷，且最多拥有 1 个子节点，快照仍然保留。

示例

- 修改一致性快照 `consistencysnapshot5` 的描述信息。

说明：交互输入快照的描述信息，1~256 位字符串。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor conssnap set -n  
consistencysnapshot5 --change-description  
Enter the description for this consistency snapshot, limited to 256 characters:  
The consistencysnapshot of lun01a, lun02.  
Set consistency snapshot consistencysnapshot5 successfully.
```

- 修改快照 `consistencysnapshot3` 的名称为 `consistencysnapshot4`，一致性快照内卷快照的回收策略为 `Delete`。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor conssnap set -n  
consistencysnapshot3 --new-name consistencysnapshot4 --reclaim-policy Delete  
Set consistency snapshot consistencysnapshot3 successfully.
```

4.8.3 回滚一致性快照

```
./stor conssnap { R | rollback } { -n | --name } CONSISTENCYSNAPSHOT_NAME
```

此命令用来回滚一致性快照。

注意：

- 此操作会将一致性快照中的数据覆盖到所有源卷的数据，建议操作前，对源卷数据通过创建快照或者一致性快照的方式进行备份。
- 如果源卷已被客户端挂载，需在客户端取消挂载卷后回滚，回滚后重新挂载卷：
 - 对于 Linux 客户端，可以执行下列步骤：
 1. 回滚一致性快照前，客户端执行命令：


```
umount DIRECTORY_NAME_OR_PATH
```
 2. 回滚一致性快照后，客户端执行命令


```
mount /dev/sdx DIRECTORY_NAME_OR_PATH
```
 - 对于 Windows 客户端，可以执行下列步骤：
 1. 回滚一致性快照前，在客户端将源卷对应的磁盘脱机。
 2. 回滚一致性快照后，在客户端将源卷对应的磁盘重新联机。
- 如果一致性快照的卷快照的源卷是一个克隆卷，并且此克隆卷正在执行断开关系链的操作时，不能使用这个克隆卷的任何快照进行回滚。

说明：如果一致性快照中有一个单卷快照处于删除中的状态，可以回滚，对其余正常状态的单卷快照进行回滚，删除中状态的快照不回滚。

参数

参数	描述
-n	指定一致性快照的名称。
CONSISTENCYSNAPSHOT_NAME	取值：字符串形式，长度范围是 1~256，只能由字母、数字、短横线(-)、下划线(_)组成，字母区分大小写，且仅支持以字母或数字开头。
或 --name	
CONSISTENCYSNAPSHOT_NAME	

示例

回滚 consistencysnapshot4。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor conssnap R -n  
consistencysnapshot4  
DANGER: This operation will overwrite data on source LUNs with that on the consistency  
snapshot.  
Start rollbacking consistency snapshot consistencysnapshot4 on LUN  
lun01,lun01a,lun02,lun03.
```

4.8.4 删除一致性快照

```
./stor conssnap rm { -n | --name } CONSISTENCYSNAPSHOT_NAME
```

此命令用来删除一致性快照。

注意：

- 若一致性快照的单卷快照有未断开关系链的克隆卷，则无法进行删除。
- 删除一致性快照会删除一致性快照下所有的单卷快照。

说明：处于删除状态中的一致性快照，仅能进行查询操作，不能执行其他操作。

参数

参数	描述
-n <i>CONSISTENCYSNAPSHOT_NAME</i> 或 --name <i>CONSISTENCYSNAPSHOT_NAME</i>	指定要删除的一致性快照名称。

示例

删除 consistencysnapshot2。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor conssnap rm -n
consistencysnapshot2
WARNING: This operation cannot be undone and will delete all consistency snapshot
information.
Start deleting consistency snapshot consistencysnapshot2.
```

4.8.5 查询一致性快照

`./stor conssnap ls [ { -n | --name } CONSISTENCYSNAPSHOT_NAME ]`

此命令用来查询一致性快照信息。

参数

参数	描述
<code>-n CONSISTENCYSNAPSHOT_NAME</code> 或 <code>--name CONSISTENCYSNAPSHOT_NAME</code>	指定要查询的一致性快照名称。

示例

- 查询所有快照的信息。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor conssnap ls
+-----+-----+-----+-----+-----+
| No. | Name | Status | LUN Snapshot Numbers | Create Time |
+-----+-----+-----+-----+-----+
| 1. | consistencysnapshot1 | Normal | 1 | 2025-03-04 16:18:17 |
| 2. | consistencysnapshot4 | Normal | 4 | 2025-03-04 17:09:42 |
+-----+-----+-----+-----+-----+
```

查询所有快照信息描述

项目	描述
No.	序号。
Name	一致性快照的名称。
Status	一致性快照的状态： ● Normal: 正常。 ● Pending: 创建中。 ● Deleting: 删除中。 ● Error: 异常。
LUN Snapshot Numbers	一致性快照的卷快照数。

Create Time	一致性快照的创建时间。
-------------	-------------

- 查询一致性快照 `consistencysnapshot4` 的信息。

```
[root@hbblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor conssnap ls -n consistencysnapshot4
Name: consistencysnapshot4
Status: Normal
Description: The consistencysnapshot of lun01,lun01a,lun02,lun03.
Create Time: 2025-03-04 17:09:42
LUN Snapshot Numbers: 3
Lun Snapshots:
+-----+-----+-----+-----+-----+-----+-----+
| No. | Name | LUN Name | LUN Capacity | Clone LUN Numbers | Reclaim Policy | Status |
+-----+-----+-----+-----+-----+-----+-----+
| 1. | lun01-snap20250304170942 | lun01 | 100 GiB | 1 | Delete | Normal |
| 2. | lun01a-snap20250304170942 | lun01a | 100 GiB | 1 | Retain | Normal |
| 3. | lun03-snap20250304170942 | lun03 | 100 GiB | 0 | Delete | Normal |
+-----+-----+-----+-----+-----+-----+-----+
```

查询指定一致性快照的信息描述

项目	描述
Name	一致性快照名称。
Status	一致性快照的状态： ● Normal: 正常。 ● Pending: 创建中。 ● Deleting: 删除中。 ● Error: 异常。
Description	一致性快照的描述信息。
Create Time	一致性快照的创建时间。
LUN Snapshot Numbers	一致性快照的卷快照数。
Lun Snapshots	一致性快照的卷快照信息： ● No.: 序号

	● Name: 卷快照的名称。 ● LUN Name: 源卷名称。 ● LUN Capacity: 创建一致性快照时刻，源卷的容量。 ● Clone LUN Numbers: 克隆卷数量。 ● Reclaim Policy: 卷快照回收策略： ■ Delete: 当快照无关联克隆卷，且最多拥有 1 个子节点（子节点表示存在其他快照依赖此快照，或者当前写操作基于此快照），快照自动删除。 ■ Retain: 当快照无关联克隆卷，且最多拥有 1 个子节点，快照仍然保留。 ● Status: 卷快照状态： ■ Normal: 正常。 ■ Error: 异常。 ■ Pending: 创建中。 ■ Deleting: 删除中。
--	---

4.9 iSCSI Target 操作

4.9.1 创建 iSCSI Target

单机版

```
./stor target add { -n | --name } TARGET_NAME [ --max-sessions MAX_SESSIONS ] [ --reclaim-policy RECLAIM_POLICY ] [ { -c | --chap-name } CHAP_NAME { -p | --password } CHAP_PASSWORD { -s | --status } STATUS ]
```

集群版

```
./stor target add { -n | --name } TARGET_NAME [ --max-sessions MAX_SESSIONS ] [ --reclaim-policy RECLAIM_POLICY ] [ { -c | --chap-name } CHAP_NAME { -p | --password } CHAP_PASSWORD { -s | --status } STATUS ] [--num SERVER_NUMBER] [ --server SERVER_ID <1-n> ]
```

此命令用来创建 iSCSI Target。

说明：HBlock 上 iSCSI Target IQN 的数量最多 32766 个。一个 iSCSI Target 最多可以关联 256 个卷，但是一个卷只能被一个 iSCSI Target 关联。

参数

参数	描述
-n TARGET_NAME 或 --name TARGET_NAME	iSCSI Target 名称。 取值：字符串形式，长度范围 1~16，只能由小写字母、数字、句点(.)和短横线(-)组成，且仅支持以字母或数字开头。 注意： 一个 iSCSI Target 最多可以关联 256 个卷，但是一个卷只能被一个 iSCSI Target 关联。
--max-sessions MAX_SESSIONS	iSCSI Target 下每个 IQN 允许建立的最大会话数。 取值：整数，取值范围是[0, 1024]，默认值为 1。

	0 表示客户端无法发现该 iSCSI Target。 注意：如果多个客户端连接同一 iSCSI Target IQN，客户端可以同时读，但不能同时写。
--reclaim-policy RECLAIM_POLICY	指定 iSCSI Target 的回收策略。 取值： ● Delete：当 iSCSI Target 关联的卷全部删除后，iSCSI Target 自动删除。 ● Retain：当 iSCSI Target 关联的卷全部删除后，iSCSI Target 仍然保留。 默认值为 Retain。 说明：如果创建卷时指定不存在的 iSCSI Target，那么同时创建 iSCSI Target，新创建 iSCSI Target 的回收策略默认为 Delete。
-c CHAP_NAME 或 --chap-name CHAP_NAME	客户端 CHAP 认证名称。 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。
-p CHAP_PASSWORD 或 --password CHAP_PASSWORD	客户端 CHAP 认证密码。 取值：字符串形式，长度范围是 12~16，必须包含大写字母、小写字母、数字、下划线(_)中的至少两种字符，字母区分大小写。
-s STATUS 或 --status STATUS	指定 CHAP 认证的状态。 取值： ● Enabled (on)：启用 CHAP 认证。 ● Disabled (off)：禁用 CHAP 认证。
--num SERVER_NUMBER	Target 所在的服务器数量（仅集群版支持）。

	取值：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。
--server <i>SERVER_ID</i> &<1- n >	指定服务器 ID（仅集群版支持）。 最多可以指定 n 个服务器 ID，以英文逗号隔开。 n 为集群内服务器的数量。

示例

- 单机版：创建 **targetc**，允许建立的最多 3 个会话数，开启 CHAP 认证。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor target add -n targetc --max-sessions 3 -c chap-test -p ***** -s on
Created Target targetc successfully.
iqn = iqn.2012-08.cn.ctyunapi.oos:targetc.2(192.168.0.32:3260)
```

- 集群版：创建 **target02**，允许建立的最多 6 个会话数，开启 CHAP 认证，指定服务器 ID 为 **hblock_1**、**hblock_2**、**hblock_3**。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor target add -n target02 --max-sessions 6 -c chap-test -p ***** -s on --num 3 --server hblock_1,hblock_2,hblock_3
Created Target target02 successfully.
iqn = iqn.2012-08.cn.ctyunapi.oos:target02.5(192.168.0.110:3260)
iqn = iqn.2012-08.cn.ctyunapi.oos:target02.6(192.168.0.192:3260)
iqn = iqn.2012-08.cn.ctyunapi.oos:target02.7(192.168.0.102:3260)
```

4.9.2 删除 iSCSI Target

```
./stor target rm { -n | --name } TARGET_NAME [ { -c | --connection }  
ISCSI_INITIATOR_NAME ] [ { -i | --target-ip } TARGET_IP ]
```

此命令用来删除指定 iSCSI Target 或断开指定 iSCSI Target 连接。

注意：只有未关联任何卷的 iSCSI Target 才能被删除。

参数

参数	描述
-n TARGET_NAME 或 --name TARGET_NAME	iSCSI Target 名称。
-c ISCSI_INITIATOR_NAME 或 --connection ISCSI_INITIATOR_NAME	要删除连接所属的 initiator 名称。 字符串形式，取值： ● 若使用 Windows iSCSI 发起程， ISCSI_INITIATOR_NAME 为“发起程序名称”。 ● 如使用 Linux 发起程序， ISCSI_INITIATOR_NAME 为“cat /etc/iscsi/initiatorname.iscsi”指令取得的“发起程序名称”。
-i TARGET_IP 或 --target-ip TARGET_IP	删除指定 Target IP 的 initiator 连接。

示例

- 删除名为 targettest 的 Target。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor target rm -n targettest  
Removed Target targettest successfully.
```

- 删除 Target 的 initiator 连接。并需要在客户端断开与 iSCSI Target 的连接，详见[客户端操作](#)。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor target rm -n target01 -c  
iqn.1994-05.com.redhat:265a95d81ae  
Removed the connection 'iqn.1994-05.com.redhat:265a95d81ae' with Target target01  
successfully.
```

4.9.3 设置 iSCSI Target 的 CHAP 认证

```
./stor target set { -i | --item } chap { -n | --name } TARGET_NAME [ { -c | --chap-name } CHAP_NAME ] [ { -p | --password } CHAP_PASSWORD ] { -s | --status } STATUS
```

此命令用来设置 iSCSI Target 的认证。

参数

参数	描述
-i chap 或--item chap	设置 CHAP 认证。
-n TARGET_NAME 或--name TARGET_NAME	iSCSI Target 名称。
-c CHAP_NAME 或--chap-name CHAP_NAME	客户端 CHAP 认证名称。 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。
-p CHAP_PASSWORD 或--password CHAP_PASSWORD	客户端 CHAP 认证密码。 取值：字符串形式，长度范围是 12~16，必须包含大写字母、小写字母、数字、下划线(_)中的至少两种字符，字母区分大小写。
-s STATUS 或--status STATUS	指定 CHAP 认证的状态： ● Enabled (on)：启用 CHAP 认证。 ● Disabled (off)：禁用 CHAP 认证。

示例

设置 target02 的 CHAP 认证名及认证密码。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor target set -i chap -n target02 -c chap3 -p ***** -s on
Set Target target02 successfully.
```

4.9.4 迁移 iSCSI Target（集群版适用）

```
./stor target set { -i | --item } server { -n | --name } TARGET_NAME { -m | --migrate }  
{ SOURCE_SERVER_ID:DEST_SERVER_ID }&<1-n> { -f | --force }
```

此命令用来迁移 iSCSI Target，修改 Target 对应的服务器。

说明：可以只迁移 iSCSI Target 对应的一个服务器，也可以同时迁移对应的多个服务器。

注意：

- 执行迁移 iSCSI Target 之前，需要保证集群处于 working 状态，同时目的服务器需要处于正常已连接状态。
- 目前仅支持强制迁移 iSCSI Target，强制迁移 iSCSI Target 可能造成数据丢失。
- 如果被迁移的 iSCSI Target 已被卷连接，且该卷已经被客户端挂载，迁移 iSCSI Target 前，需要客户端与原 iSCSI Target IQN 断开；迁移后，确保原 iSCSI Target IQN 不能被发现，客户端重新连接迁移后的 iSCSI Target IQN。

参数

参数	描述
-i server 或--item server	设置 iSCSI Target 迁移。
-n TARGET_NAME 或--name TARGET_NAME	iSCSI Target 名称。
-m { SOURCE_SERVER_ID:DEST_SERVER_ID }&<1-n>或--migrate { SOURCE_SERVER_ID:DEST_SERVER_ID }&<1-n>	指定 Target 对应的源服务器 ID 和目标服务器 ID。 说明： 可以指定多组源服务器 ID 和目标服务器 ID，以英文逗号隔开。
-f 或--force	强制迁移 iSCSI Target。 注意： 目前仅支持强制迁移 iSCSI Target，强制迁移 iSCSI Target 可能造成数据丢失。

示例

迁移 **target01** 的一个服务器。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor target set -i server -n target01 -m
hblock_2:hblock_1 --force

When migrating the Target from server forcibly, the connections on this Target will be
immediately disconnected.

If there are connections on the Target, you should disconnect the connections first.

After migrating is complete, the LUN(s) on the Target should be reconnected.

Are you sure want to migrate the Target? [Yes/No]
y
Target target01 migrated successfully from server hblock_2 to server hblock_1
You can reconnect the following iSCSI LUN(s):
```

No.	LUN Name	iSCSI Target
1.	lun01a	iqn.2012-08.cn.ctyunapi.oos:target01.1(192.168.0.102:3260,Active)
		iqn.2012-08.cn.ctyunapi.oos:target01.2(192.168.0.110:3260,ColdStandby)

说明：卷关联的 Target 处于 **ColdStandby** 状态，表示 Target 正在从一个服务器迁移到集群其他服务器上，迁移过程，该 Target 处于 **ColdStandby** 状态。迁移完成后，Target 将变为 **Active** 或者 **Standby** 状态。

4.9.5 修改 iSCSI Target 下每个 IQN 允许建立的最大会话数

```
./stor target set { -i | --item } session { -n | --name } TARGET_NAME --max-sessions  
MAX_SESSIONS
```

此命令用来修改 iSCSI Target 下每个 IQN 允许建立的最大会话数。

参数

参数	描述
-i session 或 --item session	表示设置 iSCSI Target 下每个 IQN 允许建立的最大会话数。
-n TARGET_NAME 或 --name TARGET_NAME	iSCSI Target 名称。
--max-sessions MAX_SESSIONS	iSCSI Target 下每个 IQN 允许建立的最大会话数。 整数，取值范围是[0, 1024]，默认值为 1。0 表示客户端无法发现该 Target。 说明：建议修改为比现有会话数大的数字，因为如果修改为比现有会话数小，可能会导致现有客户端连接断开之后，不能再次建立连接。 注意：如果多个客户端连接同一 iSCSI Target IQN，客户端可以同时读，但不能同时写。

示例

将 target01 下每个 IQN 允许建立的最大会话数修改为 10。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor target set -i session -n target01 --max-  
sessions 10  
Set Target target01 successfully.
```

4.9.6 修改 iSCSI Target 的回收策略

```
./stor target set { -i | --item } generic { -n | --name } TARGET_NAME --reclaim-policy  
RECLAIM_POLICY
```

此命令用来修改 iSCSI Target 的回收策略。

参数

参数	描述
-i generic 或--item generic	表示设置 iSCSI Target 的回收策略。
-n TARGET_NAME 或--name TARGET_NAME	iSCSI Target 名称。
--reclaim-policy RECLAIM_POLICY	指定 iSCSI Target 的回收策略。 取值： ● Delete: 当 iSCSI Target 关联的卷全部删除后，iSCSI Target 自动删除。 ● Retain: 当 iSCSI Target 关联的卷全部删除后，iSCSI Target 仍然保留。

示例

将 target05 的回收策略修改为 Retain。

```
[root@hbblockserver CTYUN_HBlock_Plus_3.8._x64]# ./stor target set -i generic -n target05 --  
reclaim-policy Retain  
Set Target target05 successfully.
```

4.9.7 查询 iSCSI Target

```
./stor target ls [ -c | --connection ] [ { -n | --name } TARGET_NAME ]
```

此命令用来查询 iSCSI Target 信息。

参数

参数	描述
-c 或--connection	查询 iSCSI Target 连接。
-n TARGET_NAME 或--name TARGET_NAME	查询指定 iSCSI Target 的信息。

示例

- 查询 HBlock 服务中所有 iSCSI Target 信息。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor target ls
```

No.	Target Name	Max Sessions	ISCSI Target	CHAP
1.	target01	1	iqn.2012-08.cn.ctyunapi.oos:target01.3(192.168.0.117:3260)	test,Disabled
			iqn.2012-08.cn.ctyunapi.oos:target01.4(192.168.0.102:3260)	
2.	target02	1	iqn.2012-08.cn.ctyunapi.oos:target02.5(192.168.0.192:3260)	Disabled
			iqn.2012-08.cn.ctyunapi.oos:target02.6(192.168.0.117:3260)	
3.	target03	1	iqn.2012-08.cn.ctyunapi.oos:target03.7(192.168.0.192:3260)	Disabled
			iqn.2012-08.cn.ctyunapi.oos:target03.8(192.168.0.102:3260)	
4.	target05(Deleting)	1	iqn.2012-08.cn.ctyunapi.oos:target05.9(192.168.0.117:3260)	Disabled
			iqn.2012-08.cn.ctyunapi.oos:target05.10(192.168.0.192:3260)	
5.	tg001	2	iqn.2012-08.cn.ctyunapi.oos:tg001.1(192.168.0.117:3260)	Disabled
			iqn.2012-08.cn.ctyunapi.oos:tg001.2(192.168.0.192:3260)	

iSCSI Target 查询信息描述

项目	描述
No.	序号。

Target Name	iSCSI Target 名称。 说明：如果 iSCSI Target 处于删除中，iSCSI Target 名称后面会显示“Deleting”。
Max Sessions	iSCSI Target 下每个 IQN 允许建立的最大会话数。
ISCSI Target	iSCSI Target IQN、IP 和端口号。
CHAP	CHAP 认证信息，包含 CHAP 名称、CHAP 密码、CHAP 状态。 只有配置了 CHAP 认证信息，才会显示 CHAP 名称。 CHAP 状态： ● Enabled: 开启 CHAP 认证。 ● Disabled: 未开启 CHAP 认证。

- 单机版：查询 iSCSI Target 名称为 **targetc** 的信息。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor target ls -n targetc
Target Name: targetc
Max Sessions: 3
Create Time: 2023-09-27 17:29:24
iSCSI Target: iqn.2012-08.cn.ctyunapi.oos:targetc.1(192.168.0.32:3260)
LUN: lunc1(LUN 0),lunc2(LUN 1)
Reclaim Policy: Retain
CHAP: chap-test,***** ,Enabled
```

- 集群版：查询 iSCSI Target 名称为 **target01** 的信息。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor target ls -n target01
Target Name: target01
Max Sessions: 10
Create Time: 2023-09-27 17:17:32
Number of Servers: 3
iSCSI Target: iqn.2012-08.cn.ctyunapi.oos:target01.3(192.168.0.117:3260)
               iqn.2012-08.cn.ctyunapi.oos:target01.4(192.168.0.102:3260)
               iqn.2012-08.cn.ctyunapi.oos:target01.5(192.168.0.192:3260)
LUN: lun01a(LUN 0),lun01b(LUN 1)
Reclaim Policy: Retain
```

```
CHAP: chap-test, *****,Enabled
ServerID: hblock_1,hblock_2,hblock_3
```

具体 iSCSI Target 查询信息描述

项目	描述
Target Name	iSCSI Target 名称。
Status	iSCSI Target 的状态： ● Deleting: iSCSI Target 正在删除。 仅 iSCSI Target 处于删除中会返回此项。
Max Sessions	iSCSI Target 下每个 IQN 允许建立的最大会话数。
Create Time	iSCSI Target 创建时间。
Number of Servers	iSCSI Target 所在的服务器数量（仅集群版支持）。
iSCSI Target	iSCSI Target IQN、客户端 IP 和端口号。
LUN	iSCSI Target 对应的卷。括号内容表示卷编号。
Reclaim Policy	iSCSI Target 的回收策略： ● Delete: 当 iSCSI Target 关联的卷全部删除后，iSCSI Target 自动删除。 ● Retain: 当 iSCSI Target 关联的卷全部删除后，iSCSI Target 仍然保留。
CHAP	CHAP 认证信息，包含 CHAP 名称、CHAP 密码、CHAP 状态。 说明： 只有配置了 CHAP 认证信息，才会显示 CHAP 名称。 CHAP 状态： ● Enabled: 开启 CHAP 认证。 ● Disabled: 未开启 CHAP 认证。
ServerID	iSCSI Target 对应的服务器 ID（仅集群版支持）。

- 查询 iSCSI Target 连接信息。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor target ls -c
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
```

SessionId	Target IQN	Target IP	Initiator name	Client IP	Client Port
0x8	iqn.2012-08.cn.ctyunapi.oos:target01.3	192.168.0.102	iqn.1991-05.com.microsoft:ecs-e16f-0915299	192.168.0.116	61314
0x3	iqn.2012-08.cn.ctyunapi.oos:target01.4	192.168.0.192	iqn.1991-05.com.microsoft:ecs-e16f-0915299	192.168.0.116	61416
0x2	iqn.2012-08.cn.ctyunapi.oos:target03.5	192.168.0.192	iqn.1994-05.com.redhat:ca375039f35f	127.0.0.1	58978
0x1	iqn.2012-08.cn.ctyunapi.oos:target1.1	192.168.0.192	iqn.1994-05.com.redhat:ca375039f35f	127.0.0.1	58976

iSCSI Target 连接查询信息描述

项目	描述
SessionId	iSCSI 会话 ID。
Target IQN	iSCSI Target IQN。
Target IP	iSCSI Target 对应的 IP。
Initiator name	连接所属的 initiator 名称。
Client IP	客户端 IP。
Client Port	客户端端口号。

- 查询 target02 的 iSCSI Target 连接信息。

[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor target ls -c -n target02					
SessionId	Target IQN	Target IP	Initiator name	Client IP	Client Port
0x5	iqn.2012-08.cn.ctyunapi.oos:target02.7	192.168.0.121	iqn.1991-05.com.microsoft:ecs-28f3	192.168.0.46	52594
0x7	iqn.2012-08.cn.ctyunapi.oos:target02.8	192.168.0.72	iqn.1991-05.com.microsoft:ecs-28f3	192.168.0.46	52795

具体 iSCSI Target 连接查询信息描述

项目	描述
SessionId	iSCSI 会话 ID。
Target IQN	iSCSI Target IQN。
Target IP	iSCSI Target 对应的 IP。

Initiator name	连接所属的 initiator 名称。
Client IP	客户端 IP。
Client Port	客户端端口号。

4.10 存储池操作（集群版适用）

4.10.1 创建存储池

```
./stor storagepool add { -n | --name } POOL_NAME [ --fault-domain FAULT_DOMAIN ]
```

此命令用来创建存储池。

说明：HBlock 集群中最多可以创建 32768 个存储池。

参数

参数	描述
-n POOL_NAME 或 --name POOL_NAME	指定存储池名称。 取值：字符串形式，长度范围是 1~16，只能由字母、数字、短横线（-）、下划线（_）组成，字母区分大小写，且仅支持以字母和数字开头。
--fault-domain FAULT_DOMAIN	指定存储池的故障域级别。 取值： ● path：数据目录级别。 ● server：服务器级别。 ● rack：机架级别。 ● room：机房级别。 默认值为 server 。

示例

创建存储池 pool4，并添加描述信息。

说明：如果不添加存储池描述信息，直接按回车即可。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor storagepool add -n pool4 --fault-domain path
Enter the description for this pool, limited to 50 characters:
pool4's fault domain is path level.
```

Created storage pool pool4 successfully.

4.10.2 添加节点到存储池

```
./stor storagepool { A | addnode } { -n | --name } POOL_NAME --node NODE <1-n>
```

此命令用来添加节点到存储池。

说明：初始化时添加的数据目录节点都属于基础存储池节点。后续如果需要将基础存储池中的节点添加到其他存储池，需先从基础存储池移除。

注意：

- 待添加的节点必须属于集群拓扑节点，且每个集群拓扑节点只能添加到一个存储池中。
- 添加 path 级别的节点时，该 path 不能属于其他存储池，不能处在移除状态，path 所属的服务器也不能处在移除状态。
- 当添加的节点类型低于存储池故障域级别，该节点的祖先节点中需要存在和该存储池故障域级别相同的映射节点，否则报错。如存储池故障域为 rack，针对节点 root:room1:server1，将 server1 加入到该存储池时报错。
- 当添加的节点类型高于存储池故障域级别，该节点的子孙节点中需要存在和该存储池故障域级别相同的映射节点，否则报错。例如存储池故障域为 rack，存在拓扑节点 root:room1:server1:path1，将 room1 加入到该存储池时报错。

参数

参数	描述
-n POOL_NAME 或 --name POOL_NAME	指定存储池名称。
--node NODE <1-n>	指定加入存储池的节点，该节点必须属于集群拓扑节点。一次可以添加多个节点至存储池，以英文逗号隔开。 说明： ● 如果待添加的节点下含有子节点，则尚未被分配到任何存储池的子节点，将会被统一添加到该存储池中。

	● 节点名称可以使用在集群拓扑中的全路径，格式为：<i>name:name:name</i>，从根节点开始逐级指定；也可以使用部分路径，但该路径需在集群拓扑中唯一。例如 default:room4:hblock_4、room4:hblock_4、hblock_4 在集群拓扑中均指向同一节点，且节点名在集群拓扑中唯一，则任选其一即可。
--	---

示例

添加节点 default:hblock_4:/mnt/storage01（此处 default 为根节点名称）至存储池 pool4。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor storagepool addnode -n pool4 --node
default:hblock_4:/mnt/storage01
Added node default:hblock_4:/mnt/storage01 to storage pool pool4 successfully.
```

4.10.3 修改存储池

```
./stor storagepool set { -n | --name } POOL_NAME [ --new-name NEW_NAME ] [ --change-description ]
```

此命令用来修改存储池信息。

参数

参数	描述
-n POOL_NAME 或 --name POOL_NAME	指定存储池名称。
--new-name NEW_NAME	指定存储池的新名称。 取值：字符串形式，长度范围是 1~16，只能由字母、数字、短横线（-）、下划线（_）组成，字母区分大小写，且仅支持以字母和数字开头。
--change-description	修改存储池描述信息。

示例

- 修改存储池 pool3 的名称为 rackpool3。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor storagepool set -n pool3 --new-name rackpool3
Set storage pool pool3 successfully.
```

- 修改存储池 pool4 的描述信息。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor storagepool set -n pool4 --change-description
Enter the description for this pool, limited to 50 characters:
pool4's fault domain is path level.
Set storage pool pool4 successfully.
```

4.10.4 移除存储池内的节点

```
./stor storagepool { R | rmnode } { -n | --name } POOL_NAME --node NODE &<1-n> [ -d | -  
-del-data ] [ -f | --force ]
```

此命令用来移除存储池内的指定节点。

注意：

- 基础存储池仅剩一个可用故障域时，无法移除故障域内的任何节点。
- 节点涉及存储池的多个故障域，只能强制移除，强制移除可能造成数据丢失。

参数

参数	描述
-n POOL_NAME 或 --name POOL_NAME	指定要移除节点的存储池名称。
--node NODE &<1-n>	指定节点名称。 注意： 一次可以移除同一故障域内的多个节点，以英文逗号隔开。一次不能移除涉及多个故障域的节点。
-d 或 --del-data	将节点从存储池移除，并删除存储在该节点上的 HBlock 数据。
-f 或 --force	强制将节点从存储池中移除。 注意： 强制将节点从存储池中移除，可能造成数据丢失。

示例

将节点 root:room1:rack1:hblock_2:/mnt/storage01 从存储池 rackpool3 中移除。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor storagepool rmnode -n rackpool3 --node  
root:room1:rack1:hblock_2:/mnt/storage01  
Start removing node root:room1:rack1:hblock_2:/mnt/storage01 from storage pool rackpool3.
```

4.10.5 删除非基础存储池

```
./stor storagepool rm { -n | --name } POOL_NAME
```

此命令用来删除非基础存储池。

注意：

- 不能删除基础存储池。
- 存储池内有卷，无法删除。
- 删除存储池之后，存储池的拓扑节点不再属于任何存储池，但是还在集群拓扑结构内。

参数

参数	描述
-n POOL_NAME 或 --name POOL_NAME	指定要删除的存储池名称。

示例

删除存储池 pool4。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor storagepool rm -n pool4
Start deleting storage pool pool4.
```

4.10.6 查询存储池

```
./stor storagepool ls [ { -n | --name } POOL_NAME ]
```

此命令用来查询存储池信息。

参数

参数	描述
-n POOL_NAME 或 --name POOL_NAME	查询指定存储池。

示例

- 查询所有存储池信息。

```
[root@hbblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor storagepool ls
```

No.	Pool Name	Pool Status	Fault Domain	Total Capacity	Used Capacity	LUN for CachePool	LUN for Pool
1.	default(*)	Normal	server	313.36 GiB	35.79 GiB	lun03a	lun02a,lun01a
2.	pool2	Normal	path	119.17 GiB	26.95 GiB	lun02a	lun03a

查询所有存储池信息描述

项目	描述
No.	序号。
Pool Name	存储池名称。 *: 表示是基础存储池。
Pool Status	存储池状态: ● Normal: 正常。 ● Deleting: 删除中。
Fault Domain	故障域级别: ● path: 数据目录级别。

	● server: 服务器级别。 ● rack: 机架级别。 ● room: 机房级别。
Total Capacity	存储池总容量。
Used Capacity	存储池已用容量。
LUN for CachePool	缓存存储池卷。
LUN for Pool	存储池卷。

● 查询指定存储池信息。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor storagepool ls -n pool2

Storage Pool Name: pool2

Pool Status: Normal

Fault Domain: path

Base Pool: false

Total Capacity: 122.83 GiB

Used Capacity: 25.11 GiB

Created Time: 2024-08-01 16:00:35

LUN:

    LUN for CachePool: lun02a

    LUN for Pool: lun03a

status: 2 total, 500 GiB (cache mode:1 total, 200 GiB; storage mode:1 total, 300 GiB; local mode:0 total, 0 B)

Topology:

Node Name: default:rack1:server4:/mnt/storage02

Node Type: path

Status: Healthy

Disk Path(s):

+-----+-----+-----+-----+-----+-----+-----+
| No. | Path                                | Used Capacity | Total Capacity | Used Capacity Quota | Capacity Quota | Health Status | Health Detail |
+-----+-----+-----+-----+-----+-----+-----+
| 1.  | default:rack1:server4:/mnt/storage02 | 54.57 MiB    | 52.18 GiB     | 64 KiB              | Unlimited      | Healthy      |               |
+-----+-----+-----+-----+-----+-----+-----+

Node Name: default:rack1:server4:/mnt/storage03

Node Type: path

Status: Error

Disk Path(s):

+-----+-----+-----+-----+-----+-----+-----+
```

No.	Path	Used Capacity	Total Capacity	Used Capacity Quota	Capacity Quota	Health Status	Health Detail
1.	(Removing) default:rack1:server4:/mnt/storage03	-	-	-	Unlimited	Error	DataServiceConnectionFailed;DataServiceExitAbnormally
Removing Details:							
No.	Path	Stage	Details				
1.	default:rack1:server4:/mnt/storage03	Executing	-				
Node Name: default:rack1:server4:/mnt/storage04							
Node Type: path							
Status: Healthy							
Disk Path(s):							
No.	Path	Used Capacity	Total Capacity	Used Capacity Quota	Capacity Quota	Health Status	Health Detail
1.	default:rack1:server4:/mnt/storage04	12.52 GiB	33.49 GiB	64 KiB	Unlimited	Healthy	
Node Name: default:rack1:server4:/mnt/storage05							
Node Type: path							
Status: Healthy							
Disk Path(s):							
No.	Path	Used Capacity	Total Capacity	Used Capacity Quota	Capacity Quota	Health Status	Health Detail
1.	default:rack1:server4:/mnt/storage05	12.52 GiB	33.49 GiB	24 KiB	Unlimited	Healthy	

查询指定存储池信息的描述

项目	描述
Storage Pool Name	存储池名称。
Pool Status	存储池状态： ● Normal: 正常。 ● Deleting: 删除中。
Fault Domain	存储池故障域级别：

		● path: 数据目录级别。 ● server: 服务器级别。 ● rack: 机架级别。 ● room: 机房级别。
Base Pool		是否是基础存储池： ● true: 是基础存储池。 ● false: 不是基础存储池。
Total Capacity		存储池总容量。
Used Capacity		存储池已用容量。
Created Time		存储池创建时间。
Description		存储池描述信息。
LUN	卷相关信息，包含 LUN for CachePool、LUN for Pool、status。	
	LUN for CachePool	缓存存储池卷列表。
	LUN for Pool	存储池卷列表。
	status	卷的状态信息，包括卷的总数、总容量，以及各类型卷的个数及容量。
Topology	存储池拓扑图，包括：Node Name、Node Type、Status、Disk Path(s)、Removing Details。	
	Node Name	节点名称。
	Node Type	节点类型： ● path: 数据目录。 ● server: 服务器。 ● rack: 机架。 ● room: 机房。
	Status	节点健康状态： ● Healthy: 节点处于健康状态，可正常读写。 ● Warning: 节点处于警告状态，可读。

		● Error: 节点处于错误状态，无法访问。
	Disk Path(s)	数据目录信息： ● No.: 编号。 ● Path: 数据目录。 ● Used Capacity: HBlock 数据目录对应分区的已用容量。 ● Total Capacity: HBlock 数据目录对应分区的总容量。 ● Used Capacity Quota: 已用数据目录容量配额。 ● Capacity Quota: 数据目录容量配额，当 HBlock 的使用空间一旦达到配额，就立刻阻止数据写入，不允许再使用超出配额的空间。 ● Health Status: 数据目录的健康状态： ■ Healthy: 数据目录处于健康状态，可正常读写，且数据目录所在磁盘使用率未超过阈值（系统默认值为 95%）。 ■ Warning: 数据目录处于警告状态，可读，但存在以下情况的任意一种：慢盘；数据目录所在磁盘使用率超过阈值（系统默认值为 95%）；磁盘剩余空间不足 1GiB；或者 HBlock 对这个目录停写。 ■ Error: 数据目录错误状态，无法访问，原因可能是：所在磁盘出现 I/O 错误导致无法读写，数据目录未正确挂载等。 ● Health Detail: 数据目录健康状态详情。 ■ 如果健康状态为 Healthy，此列为空。

		■ 如果健康状态为 Warning 或 Error，显示警告或错误的详细信息。
	Removing Details	被移除节点的详细信息： ● No.: 编号。 ● Path: 节点的具体数据目录。 ● Stage: 移除节点所处的阶段： ■ Reconfiguration: 重置中。 ■ CheckingData: 检查数据。 ■ Executing: 执行移除节点。 ● Details: 详细数据： ■ FaultDomains: 故障域详情，包括： healthy（健康个数），warning（告警个数），error（错误个数）。 ■ Data: safe（安全数据百分比），await reconstruction（需要重建的数据百分比），await more faultdomains（需要额外故障域才能够重建的数据百分比），single-copy（单副本数据百分比），corrupted（已经损坏的数据百分比）。

4.11 集群拓扑（集群版适用）

4.11.1 创建拓扑节点

```
./stor topology add { -n | --name } NODE_NAME { -t | --type } NODE_TYPE [ --parent-node PARENT_NODE ]
```

此命令用来创建拓扑节点。

参数

参数	描述
-n <i>NODE_NAME</i> 或 --name <i>NODE_NAME</i>	指定拓扑节点名称。 取值：字符串形式，长度范围 1~63，只能由字母、数字、句点(.)、下划线(_)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。
-t <i>NODE_TYPE</i> 或 --type <i>NODE_TYPE</i>	指定拓扑节点的类型。 取值： ● rack: 机架类型。 ● room: 机房类型。
--parent-node <i>PARENT_NODE</i>	指定父节点。 默认为根节点。

示例

创建类型为 room 的拓扑节点 room3。

说明：如果不添加节点的描述信息，直接按回车即可。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor topology add --name room3 --type room
room
Enter the description for this node, limited to 50 characters:
room3 is new node.
Created node room3 successfully.
```

4.11.2 修改拓扑节点信息

```
./stor topology set { -n | --name } NODE_NAME [ --change-description ] [ --new-name  
NEW_NAME ] [ --parent-node PARENT_NODE ]
```

此命令用来修改拓扑节点信息。

注意：不支持修改 path 级别的节点信息。

参数

参数	描述
-n NODE_NAME 或 --name NODE_NAME	指定拓扑节点名称。 取值：字符串形式，长度范围 1~63，只能由字母、数字、句点(.)、下划线(_)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。
--change-description	修改拓扑节点的描述信息。
--new-name NEW_NAME	指定拓扑节点的新名称。 取值：字符串形式，长度范围 1~63，只能由字母、数字、句点(.)、下划线(_)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。
--parent-node PARENT_NODE	修改拓扑节点的父节点。

示例

- 将节点 hblock_4 的父节点修改为 room3。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor topology set --name hblock_4 --  
parent-node room3  
Set node hblock_4 successfully.
```

- 修改节点 room3 的名称为 room-3。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor topology set --name room3 --new-  
name room-3
```

Set node room3 successfully.

4.11.3 删除拓扑节点

```
./stor topology rm { -n | --name } NODE_NAME [ -d | --del-data ] [ -f | --force ]
```

此命令用来删除拓扑节点。

注意：

- 对于 server 节点，该节点的所有 path 都不属于任何存储池，可以从集群中删除这个 server，否则只能强制移除该节点，强制移除节点可能造成数据丢失。
- 对于 room、rack 节点，该节点没有任何子节点，才能从集群中删除，否则不能移除该节点，强制移除也不可以。
- 对于同名的节点，如 room1:rack1 以及 room2:rack1，*NODE_NAME* 可以带上父节点的名称，否则可能找不到对应的节点。
- 此命令不支持删除 path 节点，如果需要删除 path 节点，请使用移除数据目录。

参数

参数	描述
-n <i>NODE_NAME</i> 或 --name <i>NODE_NAME</i>	指定要删除的拓扑节点名称。 取值：字符串形式，长度范围 1~63，只能由字母、数字、句点(.)、下划线(_)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。
-d 或 --del-data	删除拓扑节点，并删除拓扑节点上的 HBlock 数据。仅删除 server 类型的节点时支持此参数。
-f 或 --force	强制删除拓扑节点。仅删除 server 类型的节点时支持此参数。 注意： 强制删除拓扑节点，可能造成数据丢失。

示例

- 删除拓扑节点 room3。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor topology rm -n room3
```

```
Deleted node room3 successfully.
```

- 删除拓扑节点 hblock_4。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor topology rm -n  
room2:rack2:hblock4  
Deleted node room2:rack2:hblock4 successfully.
```

4.11.4 查询拓扑信息

```
./stor topology ls [ { -n | --name } NODE_NAME ]
```

此命令用来查询拓扑信息。

参数

参数	描述
-n <i>NODE_NAME</i> 或 --name <i>NODE_NAME</i>	查询指定节点的拓扑图。 说明：不支持查询 path 节点。

示例

- 查询整个拓扑图信息。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor topology ls
Type  Name
root  default
      room  room1
          rack  rack1
              server  hblock_2
                  path  /mnt/stor
room  room2
      rack  rack2
          server  hblock_1
              path  /mnt/stor
          server  hblock_3
              path  /mnt/stor
              path  /mnt/storage01
```

- 查询节点 rack2 的拓扑信息。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor topology ls -n rack2
Type  Name
rack  rack2
```

```
server hblock_1
  path /mnt/stor
server hblock_3
  path /mnt/stor
  path /mnt/storage01
```

拓扑信息描述

项目	描述
Type	节点类型： ● root: 根节点。 ● room: 机房级别的节点。 ● rack: 机架级别的节点。 ● server: 服务器级别的节点。 ● path: 数据目录级别的节点。
Name	具体节点名称。

4.12 服务器操作

4.12.1 添加服务器（集群版适用）

```
./stor server add { -s | --server } SERVER_IP[:PORT ] [ --parent-node PARENT_NODE ]
[ { -n | --name } NODE_NAME ] [ { -p | --path } PATH<1-n> [ --capacity-quota
CAPACITY_QUOTA ] ] [ --port-range PORT1-PORT2 ] [ --data-port1 DATA_PORT1 ] [ --iscsi-
port ISCSI_PORT ] [ --management-port1 MANAGEMENT_PORT1 ] [ --management-port2
MANAGEMENT_PORT2 ] [ --management-port3 MANAGEMENT_PORT3 ] [ --management-port4
MANAGEMENT_PORT4 ] [ --management-port5 MANAGEMENT_PORT5 ] [ --management-port6
MANAGEMENT_PORT6 ]
```

此命令用来添加服务器。

注意：

- 待添加到集群的服务器安装 HBlock 后，才可以在集群服务器上使用此命令添加该服务器。
- 请确保 Linux 用户具有所需要端口的权限。Linux 系统默认小于 1024 的端口不对没有 root 权限的 Linux 普通用户开放。
- 设置端口范围（--port-range PORT1-PORT2）时，请避免和 Linux 系统的本地临时端口（ip_local_port_range）范围重合，否则可能会导致 HBlock 服务所用的端口被占用。使用命令行 `cat /proc/sys/net/ipv4/ip_local_port_range` 可以查看本地临时端口范围。

参数

参数	描述
<code>-s SERVER_IP[:PORT]</code> 或 <code>--server SERVER_IP[:PORT]</code>	添加服务的 IP（ <code>SERVER_IP</code> ）及 API 端口号（ <code>PORT</code> ）。 服务器 IP：IPv4 或[IPv6]。 API 端口号：取值范围是[1, 65535]，默认值为 1443。

	需要和该服务器安装 HBlock 时设置的 API 端口号保持一致。
--parent-node <i>PARENT_NODE</i>	指定父节点。 默认为根节点。
-n <i>NODE_NAME</i> 或 --name <i>NODE_NAME</i>	指定服务器节点名称。 取值：字符串形式，长度范围 1~63，只能由字母、数字、句点(.)、下划线(_)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。 默认使用服务器 ID 作为节点名称。
-p <i>PATH</i> 或 --path <i>PATH</i>	指定数据目录，支持多个数据目录。数据目录用于存储数据，建议不要与操作系统共用磁盘或文件系统。 注意： 数据目录中不能有逗号(,)。
--capacity-quota <i>CAPACITY_QUOTA</i>	指定数据目录的容量配额，即针对加入到服务器中的每个数据目录，HBlock 可写入的数据总量。当 HBlock 的使用空间一旦达到配额，就立刻阻止数据写入，不允许再使用超出配额的空间。 取值：整型。小于数据目录所在磁盘的总容量，单位是 K/k、M/m、G/g、T/t、P/p，默认单位是 G/g。负整数表示无限制写入，0 表示禁止写入。默认不限制写入。 注意： 如果一次添加多个数据目录，只能配置一个容量配额，配置的容量配额适用添加的所有目录。
--port-range <i>PORT1-PORT2</i>	指定端口范围。存储服务以及未指定端口的服务将从此范围中自动取值。 取值：整型，取值范围为[1, 65535]， <i>PORT1</i> 为端口范围最小值， <i>PORT2</i> 为端口范围最大值，且 <i>PORT1</i> < <i>PORT2</i> 。 <i>PORT1</i> 默认取值为 20000， <i>PORT2</i> 默

	认取值为 20500。 说明： 建议指定的端口范围至少包含 500 个端口。
--data-port1 <i>DATA_PORT1</i>	数据端口 1，整型，取值为[1, 65535]。
--iscsi-port <i>ISCSI_PORT</i>	iSCSI 端口，整型，取值为[1, 65535]，默认端口为 3260。
--management-port1 <i>MANAGEMENT_PORT1</i>	管理服务端口 1，整型，取值为[1, 65535]。
--management-port2 <i>MANAGEMENT_PORT2</i>	管理服务端口 2，整型，取值为[1, 65535]。
--management-port3 <i>MANAGEMENT_PORT3</i>	管理服务端口 3，整型，取值为[1, 65535]。
--management-port4 <i>MANAGEMENT_PORT4</i>	管理服务端口 4，整型，取值为[1, 65535]。
--management-port5 <i>MANAGEMENT_PORT5</i>	管理服务端口 5，整型，取值为[1, 65535]。
--management-port6 <i>MANAGEMENT_PORT6</i>	管理服务端口 6，整型，取值为[1, 65535]。

示例

- 添加服务器 192.168.0.202，并指定/mnt/storage01 为数据目录。

说明： 如果不添加新增服务器的描述信息，直接按回车即可。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor server add -s 192.168.0.202 -p /mnt/storage01
Enter the description for this node, limited to 50 characters:
new server
Added server 192.168.0.202 with path /mnt/storage01 successfully.
```

- 添加服务器 192.168.0.117，父节点是 rack1，服务器节点命名为 server4，并指定 /mnt/storage01 为数据目录。

说明： 如果不添加新增服务器的描述信息，直接按回车即可。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor server add -s 192.168.0.117 --parent-node rack1 -n server4 -p /mnt/storage01
Enter the description for this node, limited to 50 characters:
rack1
Added server 192.168.0.117 with path /mnt/storage01 successfully.
```

4.12.2 修改服务器端口范围

```
./stor server set { -i | --item } port [ { -n | --server } SERVER_ID ] --port-range
PORT1-PORT2
```

此命令用来修改 HBlock 服务器用于 HBlock 相关服务的端口范围。

说明：后续新增的服务端口会从修改后的端口范围中选择，已使用的端口值不变。

注意：修改端口范围（--port-range PORT1-PORT2）时，请避免和 Linux 系统的本地临时端口（ip_local_port_range）范围重合，否则可能会导致 HBlock 服务所用的端口被占用。使用命令行 `cat /proc/sys/net/ipv4/ip_local_port_range` 可以查看本地临时端口范围。

参数

参数	描述
-i port 或--item port	修改 HBlock 服务器用于 HBlock 相关服务的端口范围。
-n SERVER_ID 或--server SERVER_ID	HBlock 服务器的 ID。 默认值为当前服务器 ID。
--port-range PORT1-PORT2	指定端口范围，用于相关服务。 取值：整型，取值范围为[1, 65535]， <i>PORT1</i> 为端口范围最小值， <i>PORT2</i> 为端口范围最大值，且 <i>PORT1</i> < <i>PORT2</i> 。 说明： 建议指定的端口范围至少包含 500 个端口。

示例

修改 HBlock 服务器用于 HBlock 相关服务的端口范围为 19000-20500。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor server set -i port -n hblock_1 -
-port-range 19000-20500
Set server hblock_1 successfully.
```

4.12.3 设置服务器目标门户 IP

```
./stor server set { -i | --item } target-portal-ip [ { -n | --server } SERVER_ID ] { -s | --status } STATUS [ { -e | --target-portal-ip } TARGET_PORTAL_IP { -P | --port } PORT ]
```

此命令用来设置 HBlock 服务器的目标门户 IP。

若服务器与客户端不在同一网段（如服务器位于内网，客户端位于外网），通过 NAT 设备（如路由器）进行连接，则需要将 NAT 设备的外网地址和端口添加到服务器，从而使得外网的客户端可以正常与该服务器的 Target 建立 iSCSI 连接。

参数

参数	描述
-i target-portal-ip 或--item target-portal-ip	HBlock 服务器的目标门户 IP。
-n SERVER_ID 或--server SERVER_ID	HBlock 服务器的 ID。 默认值为当前服务器 ID。
-s STATUS 或--status STATUS	目标门户 IP 的状态： ● Enabled (on)：开启目标门户 IP。 ● Disabled (off)：不开目标门户 IP。
-e TARGET_PORTAL_IP 或--target-portal-ip TARGET_PORTAL_IP	目标门户 IP，IPv4 或[IPv6]格式。
-P PORT 或--port PORT	目标门户端口号，整型，取值为[1,65535]。

示例

设置服务器 hblock_1 的目标门户：目标门户 IP 为 101.89.213.5，目标门户端口号为 3261。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor server set -i target-portal-ip -n hblock_1 -s on -e 101.89.213.5 -P 3261
```

Set server hblock_1 successfully.

4.12.4 设置服务器默认数据目录（单机版适用）

```
./stor server set { -i | --item } default-path { -p | --path } DEFAULT_PATH
```

此命令用来设置 HBlock 服务器的默认数据目录。

参数

参数	描述
-i default-path 或 --item default-path	设置 HBlock 服务器的默认数据目录。
-p DEFAULT_PATH 或 --path DEFAULT_PATH	指定默认数据目录。 数据目录必须已经添加到 HBlock 系统中，并且状态为 Normal。

示例

设置 HBlock 服务器的默认数据目录为 /mnt/storage01。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor server set -i default-path -p /mnt/storage01
Set server hblock_1 successfully.
```

4.12.5 迁移服务器上的基础服务（集群版适用）

```
./stor server set { -i | --item } service { -n | --server } SERVER_ID --dest-server  
DEST_SERVER_ID --migrate SERVICE [ --meta-dir META_DIR ]
```

此命令用来迁移 HBlock 服务器上的基础服务，包括：**mdm**（元数据管理服务）、**ls**（日志服务）、**cs**（协调服务）。

应用场景：基础服务所在的服务器出现损坏、宕机等时，为了不影响 HBlock 服务，可以迁移服务器上的基础服务。

说明：可以通过命令行 `./stor server ls { -n | --server } SERVER_ID` 查询基础服务的状态。一次只能迁移一个基础服务，如果有正在迁移的基础服务进程，必须等待迁移完成，才可以执行其他服务迁移操作。

注意：

- 迁移 **ls** 服务的时候，要确保两个 **mdm** 服务和其他两个 **ls** 服务都是 **up** 状态，除了源服务器之外的所有的其他节点的 **ms** 服务（管理服务）正常，服务状态可以通过查询服务器命令获取。
- 迁移 **mdm** 服务的时候，要确保另一个 **mdm** 服务是 **up**，除了源服务器之外的所有的其他节点的 **ps** 服务（协议解析服务）和 **ms** 服务（管理服务）正常，服务状态可以通过查询服务器命令获取。
- 迁移 **cs** 服务的时候，要确保其他两个 **cs** 服务都 **up**，除了源服务器之外的所有的其他节点的 **ps** 服务（协议解析服务）和 **ms** 服务（管理服务）正常，服务状态可以通过查询服务器命令获取。

参数

参数	描述
-i service 或 --item service	迁移服务器上的基础服务。
-n SERVER_ID 或 --server SERVER_ID	源 HBlock 服务器的 ID。
--dest-server DEST_SERVER_ID	目的服务器 ID。

<code>--migrate SERVICE</code>	需要迁移的基础服务。 取值： ● <code>mdm</code>：元数据管理服务。 ● <code>ls</code>：日志服务。 ● <code>cs</code>：协调服务。
<code>--meta-dir META_DIR</code>	迁移服务的数据目录，用于存储基础服务的相关数据信息。 说明： 为了提升读写性能，建议各基础服务的数据目录、安装目录、存储数据的数据目录相互独立。 取值：数据目录名不能含逗号，默认目录为安装目录。

示例

将 `mdm` 服务从服务器 `hblock_1` 迁移至服务器 `hblock_3`。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor server set -i service -n
hblock_1 --dest-server hblock_3 --migrate mdm --meta-dir /mnt/stor
Start migrating service mdm on hblock_1 to hblock_3. You can list server to check
whether it is completed.
```

4.12.6 移除服务器（集群版适用）

```
./stor server rm { -n | --server } SERVER_ID [ -d | --del-data ] [ -f | --force ]
```

此命令用来从 HBlock 集群中移除服务器。

注意：

- 如果移除服务器上有 Target，该 Target 对应卷的高可用类型是 ActiveStandby，移除服务器时，业务不会中断，此卷对应的 Target 会切换到其他服务器上，客户端需要重新连接 Target 对应的新服务器 IP。
- 如果移除服务器上有 Target，该 Target 对应卷的高可用类型是 Disabled，移除服务器时，业务会中断，此卷对应的 Target 会切换到其他服务器上，客户端需要重新连接 Target 对应的新服务器 IP。但服务器移除时，会有数据丢失风险。
- 如果执行日志采集后，产生的日志保存在服务器安装目录下，在服务器移除之后，该日志将被删除。如果产生的日志保存在 HBlock 的数据目录内，并且移除服务器时删除服务器 HBlock 数据目录中的数据，该日志也将被删除。
- 有服务器正在移除时，不能再移除其他服务器。如果必须移除，请使用强制移除，但有丢数据风险。
- 该节点的所有数据目录不属于任何存储池，允许移除该服务器。否则不能移除，如果必须移除，请使用强制移除，但有丢数据风险。
- 如果要移除服务器的某个数据目录属于基础存储池，且是基础存储池中仅剩的一个可用故障域中的节点，不允许移除。

参数

参数	描述
-n SERVER_ID 或 --server SERVER_ID	要移除服务器的 ID。
-d 或 --del-data	移除服务器时，删除服务器上 HBlock 数据目录中的数据。
-f 或 --force	强制移除服务器。

	注意： 强制移除服务器，可能造成数据丢失。
--	------------------------------

示例

移除服务器 ID 为 hblock_4 的服务器。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor server rm -n hblock_4
The software will be deleted on the specified server. Are you sure to remove this
server? [Yes/No]
y
Processing...
Start removing server hblock_4. You can list server to check whether it is completed.
You can reconnect the following iSCSI LUN(s):
1.iSCSI LUN name:lun04a
iSCSI Target: iqn.2012-08.cn.ctyunapi.oos:target04.7(192.168.0.209:3260,ColdStandby)
iSCSI Target: iqn.2012-08.cn.ctyunapi.oos:target04.8(192.168.0.72:3260,Active)
```

说明：卷关联的 iSCSI Target 处于 **ColdStandby** 状态，表示 iSCSI Target 正在从被删除的服务器上迁移到集群其他服务器上，迁移过程，该 iSCSI Target 处于 **ColdStandby** 状态。迁移完成后，iSCSI Target 将变为 **Active** 或者 **Standby** 状态。

4.12.7 查询服务器

`./stor server ls [ { -n | --server } SERVER_ID ] [ --port ]`

此命令用来查询 HBlock 服务器的相关信息。

参数

参数	描述
<code>-n SERVER_ID</code> 或 <code>--server SERVER_ID</code>	指定 HBlock 服务器的 ID。
<code>--port</code>	指定查询服务器中 HBlock 使用的端口号。

示例

- 单机版：查询 HBlock 服务器的信息。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor server ls
```

No.	Server ID	Server Name	Status	Public Address	Cluster Address	Recent Start Time
1.	hblock_1	hblockserver	Connected	192.168.0.66:3260	192.168.0.66	2025-02-17 17:51:27

- 集群版：查询 HBlock 服务器的信息。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor server ls
```

No.	Server ID	Server Name	Status	Public Address	Cluster Address	Recent Start Time
1.	hblock_1(M**)	hblockserver	Connected	192.168.0.192:3260	192.168.0.192	2025-01-03 10:47:27
2.	hblock_2(**)	server2	Connected	192.168.0.202:3260	192.168.0.202	2025-01-03 10:46:02
3.	hblock_3(**)	server3	Connected	192.168.0.102:3260	192.168.0.102	2025-01-03 10:47:16

HBlock 服务器信息描述

项目	描述
No.	序号。

Server ID	服务器 ID。 - 服务器 ID 后带 (M) 表示为主服务器。 - 服务器 ID 后带 (**) 表示为基础节点服务器。 - 如果只有服务器 ID，则属于非基础节点服务器。
Server Name	服务器名称。
Status	服务器状态： - Connected: 已连接。 - Disconnected: 未连接。 - Removing: 移除中。
Public Address	业务网的 IP 和端口号。
Cluster Address	集群网的 IP 和端口号。
Recent Start Time	HBlock 服务在该节点上最近一次成功启动的时间。 -: 表示 HBlock 服务处于停止状态。

- 单机版：查询服务器 ID 为 hblock_1 的服务器信息。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor server ls -n hblock_1

Server Name: hblockserver

Server ID: hblock_1

Status: Connected

Public Address: 192.168.0.66:3260

Cluster Address: 192.168.0.66

Recent Start Time: 2025-02-17 17:51:27

Version: 3.9.0

Disk Path(s):
```

No.	Path	Used Capacity	Total Capacity	Used Capacity Quota	Capacity Quota	Health Status	Health Detail
1.	/mnt/stor01(*)	284 KiB	92.86 GiB	100.09 KiB	Unlimited	Healthy	-

- 集群版：查询服务器 ID 为 hblock_1 的服务器信息。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor server ls -n hblock_1
```

Server Name: ecs-9689-0915143

Server ID: hblock_1

Node Name: default:hblock_1

Parent Node: default

Status: Connected

Master Server: true

Base Server: true

Base Service: mdm(Up, /mnt/stor02)

ls(Up, /mnt/stor02)

cs(Up, /mnt/stor02)

Public Address: 192.168.0.192:3260

Cluster Address: 192.168.0.192

Recent Start Time: 2024-11-25 15:15:23

Version: 3.9.0

Disk Path(s):

No.	Path	Storage Pool	Used Capacity	Total Capacity	Used Capacity Quota	Capacity Quota	Health Status	Health Detail
1.	/mnt/stor01	default	107.65 MiB	93.29 GiB	47.16 MiB	Unlimited	Healthy	-
2.	/mnt/stor02	pool2	152.19 MiB	93.29 GiB	152 KiB	Unlimited	Healthy	-

查询指定服务器信息描述

项目	描述
Server Name	服务器名称。
Server ID	服务器 ID。
Node Name	节点名称。
Parent Node	父节点名称。
Description	服务器描述信息，如果没有描述信息，不显示。
Status	服务器状态： ● Connected: 已连接。

	● Disconnected: 未连接。 ● Removing: 移除中。
Master Server	该服务器是否为 Master 节点（仅集群版支持）： ● true: 该服务器为 Master 节点。 ● false: 该服务器不是 Master 节点。
Base Server	该服务器是否为基础节点（仅集群版支持）： ● true: 该服务器为基础节点。 ● false: 该服务器不是基础节点。
Base Service	服务器上的基础服务（仅集群版支持），包括基础服务名称、状态及对应的数据目录。
Public Address	业务网的 IP 和端口号。
Cluster Address	集群网的 IP。
Target portal IP and Port	目标门户和目标端口。
Recent Start Time	HBlock 服务在该节点上最近一次成功启动的时间。 -: 表示 HBlock 服务处于停止状态。
Version	HBlock 版本号。
Disk Path(s)	数据目录信息： ● No.: 编号。 ● Path: 数据目录。单机版中，如果数据目录后有 (*), 表示为 HBlock 服务器的默认数据目录。 ● Storage Pool: 数据目录所属资源池。 ● Used Capacity: HBlock 数据目录对应分区的已用容量。 ● Total Capacity: HBlock 数据目录对应分区的总容量。 ● Used Capacity Quota: 数据目录已用容量配额。 ● Capacity Quota: 数据目录容量配额，当 HBlock

	的使用空间一旦达到配额，就立刻阻止数据写入，不允许再使用超出配额的空间。 ● Health Status: 数据目录的健康状态： ■ Healthy: 数据目录处于健康状态，可正常读写，且数据目录所在磁盘使用率未超过阈值（系统默认值为 95%）。 ■ Warning: 数据目录处于警告状态，可读，但存在以下情况的任何一种：慢盘；数据目录所在磁盘使用率超过阈值（系统默认值为 95%）；磁盘剩余空间不足 1GiB；或者 HBlock 对这个目录停写。 ■ Error: 数据目录错误状态，无法访问，原因可能是：所在磁盘出现 I/O 错误导致无法读写，数据目录未正确挂载等。 ● Health Detail: 数据目录健康状态详情。 ■ 如果健康状态为 Healthy，此列为空。 ■ 如果健康状态为 Warning 或 Error，显示警告或错误的详细信息。
--	--

- 集群版：集群中有 3 台服务器，移除 hblock_1 的一个数据目录，查询 hblock_1 的服务信息。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor server ls -n hblock_1

Server Name: pm-006

Server ID: hblock_1

Node Name: default:room2:rack2:hblock_1

Parent Node: default:room2:rack2

Status: Connected

Master Server: true

Base Server: true

Base Service: ls(Up, /mnt/storage01/CTYUN_HBlock_Plus_3.9.0)
```

cs(Up, /mnt/storage01/CTYUN_HBlock_Plus_3.9.0)

Public Address: 192.168.0.192:3260

Cluster Address: 192.168.0.192

Recent Start Time: 2025-01-13 13:46:11

Version: 3.9.0

Disk Path(s):

No.	Path	Storage Pool	Used Capacity	Total Capacity	Used Capacity Quota	Capacity Quota	Health Status	Health Detail
1.	/mnt/stor	default	10.31 GiB	93.29 GiB	869.26 MiB	Unlimited	Healthy	-
2.	(Removing) /tmp/pathnCV9V	default	62.1 MiB	5 GiB	62.09 MiB	Unlimited	Warning	StopWritingWhenRemoving

Removing Details (from storage pool):

No.	Path	Stage	Details
1.	/tmp/pathnCV9V	CheckingData	FaultDomains: 2 healthy, 1 warning, 0 error
			Data: 33.33% safe, 66.67% await reconstruction, 0% await more faultdomains, 0% single-copy, 0% corrupted

- 集群版：集群中有 4 台服务器，移除服务器 hblock_4，查询服务器 hblock_4 信息。

[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor server ls -n hblock_4								
Server Name: ecs-9689-0915142								
Server ID: hblock_4								
Node Name: default:room2:rack2:stor								
Parent Node: default:room2:rack2								
Status: Removing								
Master Server: false								
Base Server: false								
Public Address: 192.168.0.117:20050								
Cluster Address: 192.168.0.117								
Recent Start Time: 2024-06-26 11:30:04								
Version: 3.9.0								
Disk Path(s):								
+-----+-----+-----+-----+-----+-----+-----+-----+-----+								
No.	Path	Storage Pool	Used Capacity	Total Capacity	Used Capacity Quota	Capacity Quota	Health Status	Health Detail
+-----+-----+-----+-----+-----+-----+-----+-----+-----+								
1.	(Removing) /mnt/storage02	-	52.13 MiB	52.18 GiB	-	Unlimited	-	-
+-----+-----+-----+-----+-----+-----+-----+-----+-----+								

查询指定服务器信息描述

项目	描述
Server Name	服务器名称。
Server ID	服务器 ID。
Node Name	节点名称。
Parent Node	父节点名称。
Description	服务器描述信息，如果没有描述信息，不显示。
Status	服务器状态： ● Connected: 已连接。 ● Disconnected: 未连接。 ● Removing: 移除中。
Master Server	该服务器是否为 Master 节点： ● true: 该服务器为 Master 节点。 ● false: 该服务器不是 Master 节点。
Base Server	该服务器是否为基础节点： ● true: 该服务器为基础节点。 ● false: 该服务器不是基础节点。
Base Service	服务器上的基础服务（仅集群版支持），包括基础服务名称、状态及对应的数据目录。
Public Address	业务网的 IP 和端口号。
Cluster Address	集群网的 IP。
Recent Start Time	HBLOCK 服务在该节点上最近一次成功启动的时间。 -: 表示 HBLOCK 服务处于停止状态。
Version	HBLOCK 版本号。
Disk Path(s)	数据目录信息： ● No.: 编号。 ● Path: 数据目录。单机版中，如果数据目录后有

	(*)，表示为 HBlock 服务器的默认数据目录。 ● Storage Pool: 数据目录所属资源池。 ● Used Capacity: HBlock 数据目录对应分区的已用容量。 ● Total Capacity: HBlock 数据目录对应分区的总容量。 ● Used Capacity Quota: 数据目录已用容量配额。 ● Capacity Quota: 数据目录容量配额，当 HBlock 的使用空间一旦达到配额，就立刻阻止数据写入，不允许再使用超出配额的空间。 ● Health Status: 数据目录的健康状态： ■ Healthy: 数据目录处于健康状态，可正常读写，且数据目录所在磁盘使用率未超过阈值（系统默认值为 95%）。 ■ Warning: 数据目录处于警告状态，可读，但存在以下情况的任意一种：慢盘；数据目录所在磁盘使用率超过阈值（系统默认值为 95%）；磁盘剩余空间不足 1GiB；或者 HBlock 对这个目录停写。 ■ Error: 数据目录错误状态，无法访问，原因可能是：所在磁盘出现 I/O 错误导致无法读写，数据目录未正确挂载等。 ● Health Detail: 数据目录健康状态详情。 ■ 如果健康状态为 Healthy，此列为空。 ■ 如果健康状态为 Warning 或 Error，显示警告或错误的详细信息。
Removing Details	移除服务器或者移除数据目录时，被移除数据目录的详

	细信息： ● No.: 编号。 ● Path: 具体数据目录。 ● Stage: 移除数据目录所处的阶段： ■ Reconfiguration: 重置中。 ■ CheckingData: 检查数据。 ■ Executing: 执行移除数据目录。 ● Details: 详细数据： ■ FaultDomains: 故障域详情，包括：healthy（健康个数），warning（告警个数），error（错误个数）。 ■ Data: safe（安全数据百分比），await reconstruction（需要重建的数据百分比），await more faultdomains（需要额外故障域才能够重建的数据百分比），single-copy（单副本数据百分比），corrupted（已经损坏的数据百分比）。
--	--

- 单机版：查询服务器中 HBlock 使用的端口号。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor server ls --port
```

No.	Server ID	Server Name	Public Address	Cluster Address	Port Range	iSCSI Port	API&Web Port	Management Port
1.	hblock_1	hblockserver	192.168.0.32	192.168.0.32	20000-20500	3260	1443,2443	20000,20001,20002
								20003,20004

- 集群版：查询服务器中 HBlock 使用的端口号。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor server ls --port
```

No.	Server ID	Server Name	Public Address	Cluster Address	Port Range	iSCSI Port	API&Web Port	Data Port	Storage Port	Management Port	Metadata Port
1.	hblock_1(M)	pm-006	192.168.0.192	192.168.0.192	20000-20500	3260	1443,2443	20008	20015,20016,20017	20000,20003,20007	20001,20002,20004
									20018,20019,20020	20009,20010,20013	20005,20006,20011
											20012,20014
2.	hblock_2(**)	hblockserver	192.168.0.110	192.168.0.110	20000-20500	3260	1443,2443	20008	20015,20016,20017	20000,20003,20007	20001,20002,20004
										20009,20010,20013	20005,20006,20011
											20012,20014
3.	hblock_3(**)	ecs-9689-0915140	192.168.0.102	192.168.0.102	20000-20500	3260	1443,2443	20008	20002,20004,20005	20000,20003,20007	20001,20006,20011
										20009,20010,20013	20012,20014

- 集群版：查询服务器 hblock_1 中 HBlock 的端口号。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor server ls -n hblock_1 --port
```

Server Name: pm-006

Server ID: hblock_1

Node Name: default:room2:rack2:hblock_1

Parent Node: default:room2:rack2

Public Address: 192.168.0.192

Cluster Address: 192.168.0.192

Ports:

Port Classification	Port
Port Range	20000-20500
iSCSI Port	3260
API Port	1443
Web Port	2443
Data Port	20008
Storage Port	20015,20016,20017,20018,20019,20020
Management Port	20000,20003,20007,20009,20010,20013
Metadata Port	20001,20002,20004,20005,20006,20011,20012,20014

+-----+-----+

查询指定服务器端口号描述

项目	描述
No.	编号。
Server Name	服务器名称。
Server ID	服务器 ID。
Node Name	节点名称。
Parent Node	父节点名称。
Public Address	业务网的 IP 和端口号。
Cluster Address	集群网的 IP 和端口号。
Port Range	端口范围。
iSCSI Port	iSCSI 端口。
API Port	API 端口。
Web Port	Web 端口。
Data Port	数据端口（仅集群版支持）。
Storage Port	数据存储端口（仅集群版支持）。
Management Port	管理端口。
Metadata Port	元数据端口（仅集群版支持）。

4.12.8 添加数据目录

```
./stor server { A | addpath } { -p | --path } PATH &<1-n> [ { -n | --server }  
SERVER_ID ] [ --capacity-quota CAPACITY_QUOTA ]
```

此命令用来为 HBlock 的指定服务器添加数据目录。一次可以添加多个数据目录，以英文逗号（,）分开。

说明：对于新增的数据目录，建议设置开机自动挂载，或使用已设置自动挂载的目录或子目录。

注意：每台服务器最多只能添加 100 个数据目录。

参数

参数	描述
-p PATH 或--path PATH	要添加的数据目录，数据目录中不能包含逗号。一次可以添加多个数据目录，以英文逗号（,）分开。 该数据目录用于存储数据，建议不要与操作系统共用磁盘或文件系统。
-n SERVER_ID 或--server SERVER_ID	要添加的数据目录所属服务器 ID。 如果未指定服务器 ID，则为执行该命令的服务器添加数据目录。
--capacity-quota CAPACITY_QUOTA	指定数据目录的容量配额，即针对加入到服务器中的每个数据目录，HBlock 可写入的数据总量。当 HBlock 的使用空间一旦达到配额，就立刻阻止数据写入，不允许再使用超出配额的空间。 取值：整型。小于数据目录所在磁盘的总容量，单位是 K/k、M/m、G/g、T/t、P/p，默认单位是 G/g。负整数表示无限制写入，0 表示禁止写入。默认无限制写入。 注意： 如果一次添加多个数据目录，只能配置一个容量配额，配置的容量配额适用添加的所有目录。

示例

为 HBlock 服务器 hblock_1 添加数据目录/mnt/storage01。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor server A -p /mnt/storage01 -n  
hblock_1 --capacity-quota 20G  
Added path /mnt/storage01 on server hblock_1 successfully.
```

4.12.9 修改数据目录的容量配额

```
./stor server { S | setpath } { -p | --path } PATH &<1-n> [ { -n | --server }  
SERVER_ID ] --capacity-quota CAPACITY_QUOTA
```

此命令用来修改数据目录的容量配额。

说明：对于新增的数据目录，建议设置开机自动挂载，或使用已设置自动挂载的目录或子目录。

参数

参数	描述
-p PATH 或--path PATH	数据目录，数据目录中不能包含逗号。一次可以添加多个数据目录，以英文逗号（,）分开。 该数据目录用于存储数据，建议不要与操作系统共用磁盘或文件系统。
-n SERVER_ID 或--server SERVER_ID	要修改的数据目录所属服务器 ID。 如果未指定服务器 ID，则为执行该命令的服务器修改数据目录。
--capacity-quota CAPACITY_QUOTA	指定数据目录的容量配额，即针对加入到服务器中的每个数据目录，HBlock 可写入的数据总量。当 HBlock 的使用空间一旦达到配额，就立刻阻止数据写入，不允许再使用超出配额的空间。 取值：整型。小于数据目录所在磁盘的总容量，单位是 K/k、M/m、G/g、T/t、P/p，默认单位是 G/g。负整数表示无限制写入，0 表示禁止写入。默认无限制写入。 注意： 如果一次修改多个数据目录，只能配置一个容量配额，配置的容量配额适用所有指定的数据目录。

示例

为 HBlock 服务器 hblock_1 修改数据目录/mnt/storage01 的容量配额。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor server S -p /mnt/storage01 -n  
hblock_1 --capacity-quota 80G  
Set path /mnt/storage01 on server hblock_1 successfully.
```

4.12.10 移除数据目录

```
./stor server { R | rmpath } { -p | --path } PATH &<1-n> [ { -n | --server } SERVER_ID ]  
[ { -d | --del-data } ] [ -f | --force ]
```

此命令用来移除 HBlock 服务器的数据目录。一次可以移除多个数据目录。

注意：

- 对于单机版，如果要移除的数据目录为默认数据目录，需要先指定其他数据目录为默认数据目录，才可以移除。
- 如果执行日志采集后，产生的日志保存在 HBlock 的数据目录内，当移除该数据目录时选择删除数据目录上的 HBlock 数据，该日志将被删除。
- 有数据目录正在移除时，不能再移除其他数据目录。如果必须移除，请使用强制移除，但有丢数据风险。
- 对于集群版，该数据目录不属于任何存储池，才允许从移除该数据目录。否则不能移除，如果必须移除，请使用强制移除，但有丢数据风险。

参数

参数	描述
-p PATH &<1-n> 或 --path PATH &<1-n>	HBlock 服务器数据目录，一次可以移除多个数据目录，以英文逗号（,）分开。
-n SERVER_ID 或 --server SERVER_ID	指定移除数据目录的 HBlock 服务器 ID。 如果未指定服务器 ID，则为执行该命令的服务器移除数据目录。
-d 或 --del-data	移除 HBlock 服务器数据目录，并删除数据目录上的 HBlock 数据。
-f 或 --force	强制移除 HBlock 服务器数据目录。 注意： 强制移除数据目录，可能造成数据丢失。

示例

移除服务器 ID 为 hblock_2 的数据目录/home/stor01。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor server R -p /home/stor01 -n  
hblock_2  
Start removing path /home/stor01.
```

4.13 查询 HBlock 信息

```
./stor info [ { -i | --stor-id } | { -n | --stor-name } | { -u | --user-name } | { -S | --serial-id } | { -T | --trial } ]
```

此命令用来查看 HBlock 相关信息。

参数

参数	描述
-i 或--stor-id	查看 HBlock 的 ID。
-n 或--stor-name	查看 HBlock 的名称。
-u 或--user-name	查看 HBlock 的管理员用户名。
-S 或--serial-id	查看 HBlock 的序列号。 获取 License 时需要此序列号进行申请。
-T 或--trial	查看 HBlock 试用期信息。

示例

- 单机版：查看 HBlock 的信息。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor info
HBlock ID: B914B0E1-CF85-4954-8547-4902AE2B8B44
HBlock name: stor
HBlock serial ID: B914B0E1-CF85-4954-8547-4902AE2B8B44-0201-030500
User name: storuser
Setup date: 2024-01-24
You are using a 30-day trial version, the expiration date is 2024-04-23. After it
expires, the management function will not be able to use. Please contact your software
vendor to get a license.
```

- 集群版：查看 HBlock 的信息。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor info
HBlock ID: 89F43ADB-73BC-4D4F-A0BD-03F0D14ABB81
```

```
HBlock name: stor1
HBlock serial ID: 89F43ADB-73BC-4D4F-A0BD-03F0D14ABB81-0202-030700
User name: storuser
Base Pool: default
Fault domain of base pool: server
Setup date: 2024-06-19
You are using a 30-day trial version, the expiration date is 2024-09-17. After it
expires, the management function will not be able to use. Please contact your software
vendor to get a license.
```

HBlock 信息描述

项目	描述
HBlock ID	HBlock 的 ID。
HBlock name	HBlock 的名字。
HBlock serial ID	HBlock 的序列号。
User name	HBlock 的管理员用户名。
Base Pool	基础存储池（仅集群版）。
Fault domain of base pool	基础存储池的故障域类型（仅集群版）： ● path: 数据目录级别。 ● server: 服务器级别。 ● rack: 机架级别。 ● room: 机房级别。
Setup date	HBlock 初始化的时间。

- 查看 HBlock 的 ID。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor info --stor-id
HBlock ID: CECD1458-7405-4825-BF6A-75C04F1114ED
```

- 查看 HBlock 的名称。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor info --stor-name  
HBlock name: test
```

- 查看 HBlock 的管理员用户名。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor info --user-name  
User name: storuser
```

- 查看 HBlock 的序列号。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor info --serial-id  
HBlock serial ID: CECD1458-7405-4825-BF6A-75C04F1114ED-00000003-00000001-00000015
```

- 查看 HBlock 试用期信息（已导入 license）。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor info --trial  
It is not trial version. You can run "stor license ls" to view the license information
```

- 查看 HBlock 试用期信息（未导入 license）。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor info --trial  
You are using a 30-day trial version, the expiration date is 2024-08-06. After it  
expires, the management function will not be able to use. Please contact your software  
vendor to get a license.
```

4.14 查看 HBlock 服务状态

```
./stor status [ { -t | --type } TYPE ]
```

此命令用来查看 HBlock 服务状态。

参数

参数	描述
-t TYPE 或 --type TYPE	查看指定服务的状态， <i>TYPE</i> 取值： ● system: 查看系统状态。 ● storagepool: 存储池状态（仅集群版支持）。 ● server: 查看服务器状态。 ● disk: 查看磁盘状态（仅单机版支持）。 ● LUN: 查看卷的状态。 未指定状态，则显示全部信息。

示例

- 单机版：查看 HBlock 服务状态。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor status
System:
  status: Working, licenseStatus: Effective
Server:
  status: 1 total, 1 connected, 0 disconnected, 0 removing
Disk:
  usage: 60.15 MiB used, 93.29 GiB total
  status: 1 total, 1 healthy, 0 warning, 0 error
LUN:
  status: 3 total, 230 GiB (cache mode:0 total, 0 B; storage mode:1 total, 110 GiB;
local mode:2 total, 120 GiB)
  data: 100% normal, 0% low redundancy, 0% error
```

- 集群版：查看 HBlock 服务状态。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor status
System:
    status: Working, licenseStatus: None
Server:
    status: 3 total, 3 connected, 0 disconnected, 0 removing
LUN:
    status: 7 total, 1.17 TiB (cache mode:2 total, 450 GiB; storage mode:1 total, 200
GiB; local mode:4 total, 550 GiB)
    data: 3% normal, 97% low redundancy, 0% error
Storage Pool:
    number: 3
    base pool: default
    fault domain of base pool:
        level: server
        status: 1 total, 1 healthy, 0 warning, 0 error
    disk of base pool:
        usage: 20.81 GiB used, 126.78 GiB total
        status: 2 total, 2 healthy, 0 warning, 0 error
```

- 查看 HBlock 服务器状态。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor status -t server
Server:
    status: 3 total, 2 connected, 1 disconnected, 0 removing
```

HBlock 服务状态信息描述

项目	描述
System	系统状态，包括： ● status：系统运行状态。 ● licenseStatus：软件许可证状态。
Server	服务器状态，包括：服务器个数、状态为连接的服务器个数、状态为断开连接的服务器个数、状态为正在移除的服

	务器个数。
Disk	磁盘状态（仅单机版支持），包括： ● usage: 数据目录所在磁盘使用情况，包括磁盘已用容量、磁盘总容量。 ● status: 数据目录所在磁盘状态，包括磁盘总个数，状态为健康的磁盘个数、状态为告警的磁盘个数、状态为错误的磁盘个数。
LUN	卷的状态，包括： ● status: 卷的总数、总容量，以及各类型卷的个数及容量。 ● data: 卷的数据状态，包括：卷数据正常的比例，卷数据是低冗余的比例，卷数据异常的比例。
Storage Pool	存储池状态（仅集群版支持）： ● number: 存储池个数。 ● base pool: 基础存储池名称。 ● fault domain of base pool: 基础存储池的故障域： ■ level: 故障域级别： ◆ path: 数据目录级别。 ◆ server: 服务器级别。 ◆ rack: 机架级别。 ◆ room: 机房级别。 ■ status: 故障域状态： ◆ Healthy: 健康状态，可正常读写。 ◆ Warning: 警告状态，可读。 ◆ Error: 错误状态，无法访问。 ● disk of base pool: 基础存储池的数据目录状态：

	■ usage: 数据目录所在磁盘使用情况，包括磁盘已用容量、磁盘总容量。 ■ status: 数据目录所在磁盘状态，包括磁盘总个数，状态为健康的磁盘个数、状态为告警的磁盘个数、状态为错误的磁盘个数。
--	--

4.15 监控

4.15.1 查询实时性能数据

```
./stor monitor { V | view } [ { -d | --dimension } DIMENSION ] [ { -i | --instance }  
INSTANCE_ID <1-n> ] [ { -m | --metric } METRIC <1-n> ]
```

此命令用来查看 HBlock 实时性能数据。

说明：监控数据以服务器系统时间为准进行记录。时间被调整，或集群中服务器时间不统一，都可能导致监控数据不准确。但用户的业务数据不会受到影响。

参数

参数	描述
-d <i>DIMENSION</i> 或 --dimension <i>DIMENSION</i>	指定监控对象。 取值： ● system：系统。 ● pool：存储池（仅集群版支持）。 ● server：服务器。 ● diskPath：数据目录。 ● LUN：卷。 单机版默认值为 system 。集群版默认值为 pool 。
-i <i>INSTANCE_ID</i> 或 --instance <i>INSTANCE_ID</i>	指定监控对象实例的唯一标识，每次可以填写多个实例，以英文逗号隔开： ● 监控对象为 system，没有实例。 ● 监控对象为 pool，实例为存储池名称。 ● 监控对象为 server，实例取值为服务器 ID，必填。 ● 监控对象为 diskPath，实例取值为 <i>serverId:diskPath</i>。

	● 监控对象为 LUN，实例取值为卷名称。 如果不指定对象实例，默认查询监控对象的所有实例。
-m METRIC 或 --metric METRIC	指定指标名称。 ● 监控对象为 system，取值：PATH、IOPS、BANDWIDTH、LATENCY、CLOUD。 ● 监控对象为 pool，取值：PATH、IOPS、BANDWIDTH、LATENCY。 ● 监控对象为 server，取值：MEM、CPU、PATH、IOPS、BANDWIDTH、LATENCY、CLOUD。 ● 监控对象为 diskPath，取值为 PATH。 ● 监控对象为 LUN，取值：IOPS、BANDWIDTH、LATENCY、CLOUD。 默认查询监控对象的所有指标。

示例

- 查询系统的实时性能数据。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor monitor V -d system
```

No.	Time	IOPS(T/W/R)	Bandwidth(T/W/R)	Latency(T/W/R)	Path(Total/Used/Rate)	Quota(Total/Used/Rate)	Cloud Bandwidth(T/U/D)
1	2024-05-07 17:58:40	0/ 0/ 0	0 B/s/ 0 B/s/ 0 B/s	10.67 ms/ 10.67 ms/ -	279.87 GiB/ 91.84 GiB/ 32.81%	279.87 GiB/ 169.28 MiB/ 0.06%	0 B/s/ 0 B/s/ 0 B/s

- 查询存储池的实时性能数据。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor monitor V -d pool
```

No.	Time	Pool Name	IOPS(T/W/R)	Bandwidth(T/W/R)	Latency(T/W/R)	Path(Total/Used/Rate)	Quota(Total/Used/Rate)
1	2024-06-26 16:03:20	default(*)	0/ 0/ 0	0 B/s/ 0 B/s/ 0 B/s	-/ -/ -	126.78 GiB/ 20.84 GiB/ 21.24%	126.78 GiB/ 1.01 GiB/ 0.54%
2	2024-06-26 16:03:00	pool2	0/ 0/ 0	0 B/s/ 0 B/s/ 0 B/s	-/ -/ -	93.29 GiB/ 6.75 GiB/ 7.23%	93.29 GiB/ 1.03 GiB/ 1.11%
3	2024-06-26 16:03:00	pool3	0/ 0/ 0	0 B/s/ 0 B/s/ 0 B/s	-/ -/ -	93.29 GiB/ 9.06 GiB/ 9.71%	93.29 GiB/ 88 KiB/ 0%

- 查询服务器的实时性能数据。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor monitor V -d server
```

No.	Time	Server ID	CPU Rate	Memory(Total/Used/Rate)	IOPS(T/W/R)	Bandwidth(T/W/R)	Latency(T/W/R)	Path(Total/Used/Rate)	Quota(Total/Used/Rate)	Cloud Bandwidth(T/U/D)
1	2024-05-07 17:58:40	hblock_1	56.72%	15.51 GiB/ 10.53 GiB/ 67.9%	19.44/ 19.44/ 0	4.63 MiB/s/ 4.63 MiB/s/ 0 B/s	3.38 ms/ 3.38 ms/ -	93.29 GiB/ 28.39 GiB/ 30.43%	93.29 GiB/ 56.55 MiB/ 0.06%	0 B/s/ 0 B/s/ 0 B/s
2	2024-05-07 17:59:00	hblock_2	61.35%	15.51 GiB/ 11.4 GiB/ 73.52%	0/ 0/ 0	0 B/s/ 0 B/s/ 0 B/s	-/ -/ -	93.29 GiB/ 30.85 GiB/ 33.07%	93.29 GiB/ 56.18 MiB/ 0.06%	0 B/s/ 0 B/s/ 0 B/s
3	2024-05-07 17:59:00	hblock_3	45.06%	15.51 GiB/ 10.49 GiB/ 67.6%	0/ 0/ 0	0 B/s/ 0 B/s/ 0 B/s	-/ -/ -	93.29 GiB/ 32.6 GiB/ 34.95%	93.29 GiB/ 56.55 MiB/ 0.06%	0 B/s/ 0 B/s/ 0 B/s

- 查询数据目录的实时性能数据。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor monitor V -d diskPath
```

No.	Time	DiskPath	Total Path Capacity	Used Path Capacity	Path Rate	Path Quota Capacity	Used Path Quota	Quota Rate
1	2024-05-07 17:55:00	hblock_1:/mnt/stor	93.29 GiB	28.39 GiB	30.43%	93.29 GiB	56.55 MiB	0.06%
2	2024-05-07 17:55:00	hblock_2:/mnt/stor	93.29 GiB	30.85 GiB	33.07%	93.29 GiB	56.18 MiB	0.06%
3	2024-05-07 17:55:00	hblock_3:/mnt/stor	93.29 GiB	32.6 GiB	34.95%	93.29 GiB	56.55 MiB	0.06%

- 查询 LUN 的实时性能信息。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor monitor V -d LUN
```

No.	Time	LUN Name	IOPS(T/W/R)	Bandwidth(T/W/R)	Latency(T/W/R)	Cloud Bandwidth(T/U/D)	Wait Upload
1	2024-05-07 18:12:20	lun01a	0/ 0/ 0	0 B/s/ 0 B/s/ 0 B/s	-/ -/ -	-/ -/ -	-
2	2024-05-07 18:12:20	lun03a	0/ 0/ 0	0 B/s/ 0 B/s/ 0 B/s	-/ -/ -	0 B/s/ 0 B/s/ 0 B/s	0.0 B
3	2024-05-07 18:12:20	lun04a	0/ 0/ 0	0 B/s/ 0 B/s/ 0 B/s	-/ -/ -	0 B/s/ 0 B/s/ 0 B/s	0.0 B
4	2024-05-07 18:12:20	lun05a	0/ 0/ 0	0 B/s/ 0 B/s/ 0 B/s	-/ -/ -	0 B/s/ 0 B/s/ 0 B/s	0.0 B
5	2024-05-07 18:12:20	lun06a	0/ 0/ 0	0 B/s/ 0 B/s/ 0 B/s	-/ -/ -	0 B/s/ 0 B/s/ 0 B/s	0.0 B
6	2024-05-07 18:12:20	lun10a	0/ 0/ 0	0 B/s/ 0 B/s/ 0 B/s	1.34 ms/ 1.34 ms/ -	0 B/s/ 0 B/s/ 0 B/s	537.08 MiB

实时性能数据描述

项目	描述
----	----

No.	序号。
Time	查询时间点。
IOPS(T/W/R)	每秒读写总次数、写次数、读次数。
Bandwidth(T/W/R)	总带宽、写带宽、读带宽。
Latency(T/W/R)	总时延、写时延、读时延。
Path(Total/Used/Rate)	数据目录所在磁盘总容量、已用容量、平均使用率。
Quota(Total/Used/Rate)	HBlock 总容量配额、已用容量配额、容量配额使用率。
Cloud Bandwidth(T/U/D)	上云总带宽、上传带宽、下载带宽。
Server ID	服务器 ID。
Pool Name	存储池名称。
CPU Rate	CPU 使用率。
Memory(Total/Used/Rate)	服务器总内存、已用内存、内存使用率。
DiskPath	数据目录。
Total Path Capacity	数据目录所在磁盘总容量。
Used Path Capacity	数据目录所在磁盘已用容量。
Path Quota Capacity	HBlock 的容量配额。
Used Path Quota	HBlock 已使用的容量。
Quota Rate	HBlock 的容量配额使用率。
LUN Name	卷名称。
Wait Upload	待上传云的数据量。

4.15.2 导出性能数据

```
./stor monitor { E | export } [ { -d | --dimension } DIMENSION ] [ { -i | --instance }  
INSTANCE_ID ] [ --start-time START_TIME ] [ --end-time END_TIME ] [ { -o | --out }  
DIRECTORY ]
```

此命令用来导出 HBlock 性能数据。

参数

参数	描述
-d DIMENSION 或 --dimension DIMENSION	指定监控对象。 取值： ● system: 系统。 ● pool: 存储池（仅集群版支持）。 ● server: 服务器。 ● diskPath: 数据目录。 ● LUN: 卷。 单机版默认值为 system 。集群版默认值为 pool 。
-i INSTANCE_ID 或 --instance INSTANCE_ID	指定监控对象实例的唯一标识： ● 监控对象为 system，没有实例。 ● 监控对象为 pool，必填，实例为存储池名称。 ● 监控对象为 server，必填，实例取值为服务器 ID。 ● 监控对象为 diskPath，必填，实例取值为 serverId:diskPath。 ● 监控对象为 LUN，必填，实例取值为卷名称。
--start-time START_TIME	性能数据导出的起始时间。格式为 yyyy-MM-dd HH:mm:ss ，如果不指定，默认为性能数据导出结束时间

	2 小时之前的时间点。 注意：起始时间必须早于结束时间，且起始时间不能早于服务器当前时间一年。
--end-time <i>END_TIME</i>	性能数据导出的结束时间。格式为 <i>yyyy-MM-dd HH:mm:ss</i>，默认值为当前时间。 注意：起始时间必须早于结束时间。
-o <i>DIIRECTORY</i> 或 --out <i>DIIRECTORY</i>	指定导出文件存放的目录，为绝对路径。默认存放在被请求服务器的 HBlock 安装目录下，以 <i>monitordata_instanceId_yyyyMMddHHmmss.csv</i> 命名，其中： ● <i>instanceId</i>：监控对象的实例，若未指定，此处显示为空。 ● <i>yyyyMMddHHmmss</i>：文件生成的时间。

示例

导出卷 lun01a 2023-08-10 14:00 至 2024-01-23 14:00 的性能数据至

/home/stor/monitor。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor monitor E -d LUN -i lun01a --start-time 2023-08-10 14:00:00 --end-time 2024-01-23 14:00:00 --out /mnt/stor/monitor
Exported successfully.
```

4.16 告警

注意：告警状态为告警中的数据存储上限为 10000 条，达到上限后，新的告警无法触发。届时请尽快解决相关故障，或者尝试手动解除告警。

4.16.1 查看告警信息

```
./stor alarm ls [ -S { Resolved | Expired } [--alarm-severity ALARM_SEVERITY] [ --num NUMBER ]
```

此命令用来查看已解除或者已失效的告警。

```
./stor alarm ls [ -S Unresolved ] [ --alarm-severity ALARM_SEVERITY ] [ --mute-status MUTE_STATUS ] [ --num NUMBER ]
```

此命令用来查看告警中的告警。

```
./stor alarm ls [ { -n | --alarm } ALARM_ID ]
```

此命令用来查看告警状态为告警中的单条告警。

说明：告警数据以服务器系统时间为准进行记录。时间被调整，或集群中服务器时间不统一，都可能导致告警数据不准确。但用户的业务数据不会受到影响。

参数

参数	描述
-S Resolved	查看告警状态为已解除的告警。
-S Expired	查看告警状态为已失效的告警。
-S Unresolved	查看告警状态为告警中的告警。 如果未指定告警状态，默认查看告警状态为告警中的告警。
--alarm-severity ALARM_SEVERITY	告警级别： ● Warning: 警告。 ● Major: 重要。 ● Critical: 严重。

	默认查看所有级别的告警。
--num NUMBER	指定查询的告警数量。取值为[2, 1000]。 如果不指定查询告警数量，则最多列出 10000 条告警记录。
--mute-status MUTE_STATUS	告警的静默状态： ● Normal: 正常 ● Muted: 已静默 默认值为 Normal。
-n ALARM_ID 或--alarm ALARM_ID	告警 ID。

示例

● 查询已解除的告警。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor alarm ls -S Resolved
```

Alarm ID	Instance ID	Alarm Rule	Severity	Alarm Time	Alarm Value	Resolve Time	Resolve Value	Duration	Resolve Type
4j1Snw1d	hblock_3:/mnt/stor	CapacityQuotaUsageExceedsThreshold	Warning	2024-01-08 11:21:11	93.69%	2024-01-08 11:31:14	5.06%	10m 3s	Auto
4j1SnJ9h	hblock_3	CapacityQuotaUsageExceedsThreshold	Warning	2024-01-08 11:21:11	93.69%	2024-01-08 11:31:14	5.06%	10m 3s	Auto
4j1SmXuZ	hblock_3:/mnt/stor	FaultDomainWarning	Warning	2024-01-08 11:21:10	-	2024-01-08 11:31:14	-	10m 3s	Auto
4j1SmPPo	hblock_3:/mnt/stor	DataServiceHealthStatusWarning	Warning	2024-01-08 11:21:10	-	2024-01-08 11:31:13	-	10m 2s	Auto
4i1ZH7wX	hblock_1	ProtocolServiceAbnormal	Major	2024-01-05 17:16:37	-	2024-01-08 09:20:46	-	2d 16h 4m	Auto
4hDwJnaH	hblock_3	ProtocolServiceAbnormal	Major	2024-01-03 17:47:45	-	2024-01-03 17:53:47	-	6m 1s	Auto
4hD1BMWS	hblock_4	ProtocolServiceAbnormal	Major	2024-01-03 17:15:35	-	2024-01-03 17:53:46	-	38m 11s	Auto
4hD2AgSt	hblock_1:/mnt/stor	FaultDomainError	Major	2024-01-03 17:16:36	-	2024-01-03 17:17:37	-	1m 1s	Auto
4hD2zeSY	hblock_1:/mnt/stor	DataServiceHealthStatusError	Major	2024-01-03 17:16:35	-	2024-01-03 17:17:36	-	1m 1s	Auto
4hCtBCK5	hblock_4:/mnt/storage01	DataServiceHealthStatusError	Major	2024-01-03 16:40:27	-	2024-01-03 16:41:27	-	1m 0s	Auto

● 查询告警中的告警。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor alarm ls
```

Alarm ID	Instance ID	Alarm Rule	Severity	Alarm Time	Alarm Value	Resolve Time	Resolve Value	Duration	Resolve Type
----------	-------------	------------	----------	------------	-------------	--------------	---------------	----------	--------------

Alarm ID	Instance ID	Alarm Rule	Severity	Alarm Time	Alarm Value	Current Value	Duration	Mute Status
4jlhtUe1	hblock_4	ProtocolServiceAbnormal	Major	2024-01-08 10:43:03	-	-	20s	Normal

- 查询告警级别为 Warning 的告警。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor alarm ls -S Unresolved --alarm-severity Warning
```

Alarm ID	Instance ID	Alarm Rule	Severity	Alarm Time	Alarm Value	Current Value	Duration	Mute Status
1kyvM3zJ	ehc2b6a9-f3fb-4098-a6b3-3652a5d76269	LicenseWillExpire	Warning	2022-08-16 17:26:02	-	-	16h 26m 41s	Normal

- 查询告警 ID 为 4jlhtUe1 的告警。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor alarm ls -n 4jlhtUe1
```

Alarm Id: 4jlhtUe1

Instance Id: hblock_4

Instance Snapshot: hblock_4,ecs-9689-0915141,192.168.0.202

Severity: Major

Alarm Rule: ProtocolServiceAbnormal

Alarm Time: 2024-01-08 10:43:03

Alarm Value: -

Current Value: -

Duration: 2m 0s

Alarm Status: Unresolved

Mute Status: Normal

Mute Operations: -

告警信息描述

项目	描述
Alarm ID	告警 ID。
Instance ID	告警实例 ID。

Alarm Rule	告警规则。告警规则具体描述详见附录告警列表。
Severity	告警级别： ● Warning: 警告。 ● Major: 重要。 ● Critical: 严重。
Alarm Time	告警发生时间。
Alarm Value	告警时数值。仅有数值型指标对应的告警会有数值显示，以下告警规则会涉及数值型指标： ● AlarmNumberApproachingLimit: 告警中的告警条数接近上限，显示告警中的告警总条数/告警中条数上限，百分数，单位是%。计算公式：（告警中的告警总条数/告警中条数上限）*100%。 ● ResourceUsageApproachingLimit: 资源用量接近使用上限，显示许可证已用容量使用率，百分数，单位是%。计算公式：（本地卷总容量/许可证允许的容量）*100%。 ● CapacityQuotaUsageExceedsThreshold: 配额使用率超阈值，显示存储池或数据目录关联磁盘的配额使用率（Path_Cap_Quota_Rate），百分数，单位是%。 ● CapacityQuotaUsageApproachLimit: 配额用尽，显示存储池或数据目录关联磁盘的配额使用率（Path_Cap_Quota_Rate），百分数，单位是%。 ● DiskUsageExceedsThreshold: 磁盘使用率超阈值，显示存储池或数据目录关联磁盘的使用率（Path_Rate），百分数，单位是%。 ● InsufficientSpaceonInstallationPath: 安装目

	录剩余空间不足，显示目录所在磁盘的文件系统剩余空间，单位是 GiB。 ● InsufficientSpaceonMetaDir: 基础服务数据目录剩余空间不足，显示目录所在磁盘的文件系统剩余空间，单位是 GiB。
Current Value	当前数值。仅有数值型指标对应的告警会有数值显示，以下告警规则会涉及数值型指标： ● AlarmNumberApproachingLimit: 告警中的告警条数接近上限，显示显示告警中的告警总条数/告警中条数上限，百分数，单位是%。计算公式：$(\text{告警中的告警总条数} / \text{告警中条数上限}) * 100\%$。 ● ResourceUsageApproachingLimit: 资源用量接近使用上限，显示许可证已用容量使用率，百分数，单位是%。计算公式：$(\text{本地卷总容量} / \text{许可证允许的容量}) * 100\%$。 ● CapacityQuotaUsageExceedsThreshold: 配额使用率超阈值，显示存储池或数据目录关联磁盘的配额使用率（Path_Cap_Quota_Rate），百分数，单位是%。 ● CapacityQuotaUsageApproachLimit: 配额用尽，显示存储池或数据目录关联磁盘的配额使用率（Path_Cap_Quota_Rate），百分数，单位是%。 ● DiskUsageExceedsThreshold: 磁盘使用率超阈值，显示存储池或数据目录关联磁盘的使用率（Path_Rate），百分数，单位是%。 ● InsufficientSpaceonInstallationPath: 安装目录剩余空间不足，显示目录所在磁盘的文件系统剩余空间，单位是 GiB。

	● InsufficientSpaceonMetaDir: 基础服务数据目录剩余空间不足, 显示目录所在磁盘的文件系统剩余空间, 单位是 GiB。
Duration	告警持续时长。
Mute Status	静默状态: ● Muted: 静默。 ● Normal: 正常。
Resolve Time	告警解除时间。
Resolve Value	告警解除值。
Resolve Type	告警解除方式: ● Auto: 自动解除。 ● Manual: 手动解除。
Instance Snapshot	告警实例快照, 即告警发生时告警实例的详细信息。
Alarm Status	告警状态: ● Resolved: 已解除。 ● Expired: 已失效。 ● Unresolved: 告警中。
Mute Operations	静默操作记录, 包括: ● operTime: 操作时间。 ● operType: 静默操作类型: ■ Mute: 静默。 ■ AutoUnmute: 自动解除静默。 ■ ManualUnmute: 手动解除静默。 ● reason: 静默/解除静默的原因。 ● dueTime: 静默截止时间。

4.16.2 导出告警

```
./stor alarm { export | E } [ { -S | --status } STATUS ] [ { -o | --out } DIRECTORY ]
```

此命令用来导出告警数据。

参数

参数	描述
-s STATUS 或--status STATUS	告警状态： ● Unresolved: 告警中。 ● Resolved: 已解除。 ● Expired: 已失效。 默认值为 Unresolved 。
-o DIRECTORY 或--out DIRECTORY	导出告警的存放的目录，为绝对路径。默认存放在被请求服务器的 HBlock 安装目录下，以 alarm_Status_yyyyMMddHHmmss.csv 命名，其中： ● Status: 告警状态。 ● yyyymmddHHmmss: 文件生成的时间。

示例

导出告警中的告警信息至 `/home/stor/alarms`。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor alarm E -o /home/stor/alarms
Exported successfully.
```

4.16.3 手动解除告警

```
./stor alarm { resolve | R } { -n | --alarm } ALARM_ID
```

此命令用来手动解除告警。

说明：手工解除告警时，需要输入解除告警的原因。填写的原因不能超过 50 字符。如果告警提示的问题未解决，系统会再次发送告警。

参数

参数	描述
-n ALARM_ID 或 --alarm ALARM_ID	告警 ID。

示例

手动解除告警 10DR4186。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor alarm R -n 10DR4186
Enter the troubleshooting instruction of this alarm record, limited to 50 characters:
Restart server.
Successfully set the alarm status to 'Resolved', make sure the related problem is fixed,
otherwise it will alarm again.
```

4.16.4 静默告警

```
./stor alarm { M | mute } { -n | --alarm } ALARM_ID --due-time DUE_TIME
```

此命令用来静默告警。

说明：静默告警时，需要输入静默告警的原因，填写的原因不能超过 50 字符。

参数

参数	描述
-n ALARM_ID 或 --alarm ALARM_ID	告警 ID。
--due-time DUE_TIME	静默截止时间。格式为 <i>yyyy-MM-dd HH:mm:ss</i> 。

示例

静默告警 1kUyXPFy。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor alarm M -n 1kUyXPFy --due-time 2022-08-17 19:00:00
Enter the reason for this operation, limited to 50 characters:
I will apply new license.
Muted successfully.
```

4.16.5 解除告警静默

```
./stor alarm { UM | unmute } { -n | --alarm } ALARM_ID
```

此命令用来解除告警静默。

说明：解除告警静默时，需要输入静默告警的原因及静默截止时间。填写的原因不能超过 50 字符。

参数

参数	描述
-n ALARM_ID 或 --alarm ALARM_ID	告警 ID。

示例

解除告警 1kUyXPfY 静默。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor alarm unmute -n 1kUyXPfY
Enter the reason for this operation, limited to 50 characters:
No new license.
Unmuted successfully.
```

4.17 事件和日志管理

事件名称描述详见附录用户事件列表和系统事件列表。

说明：系统可以保留 6 个月的事件。

4.17.1 查看 HBlock 事件

```
./stor event ls [ --type TYPE ] [ --num NUMBER ]
```

此命令用来查看 HBlock 事件信息。

参数

参数	描述
--type TYPE	指定查看 HBlock 事件的类型。 取值： ● system: 系统事件。 ● user: 用户事件。 默认值为 user 。
--num NUMBER	指定最近事件的查询个数。 取值：[1, 1000]，默认值为 1000。当取值为 1 时，显示最近一次事件的详细信息，当取值大于 1 时，显示最近事件的主要信息。

示例

- 查询最近发生的 1 条用户事件信息。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor event ls --num 1
event ID: f0ffa38c-7a8e-4253-9352-b19eb4af906c
module: Server
name: RemoveServer
event time: 2024-01-23 14:07:06
```

```
request ID: 1db0b648560144849ce63cfc59453d82
requester IP: 192.168.0.110
status code: 202
error code: -
error message: -
details: {method:DELETE,
URL:/rest/v1/system/server/hblock_4?force=false&deleteLocalData=false}
```

用户事件描述

项目	描述
event ID	用户事件 ID。
moudle	用户事件所属模块。
name	用户事件名称。 说明： 用户事件名称描述详见 用户事件列表 。
event time	HBlock 接收到用户事件请求的时间。
requester ID	用户事件请求 ID。
requester IP	发起请求的源 IP 地址。
status code	响应状态码。
error code	错误码。 -: 表示无错误码。
error message	错误信息。 -: 表示无错误信息。
details	用户事件详情。

- 查询最近发生的 5 条用户事件。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor event ls --type user --num 5

+-----+-----+-----+-----+-----+-----+-----+
| Event Time | Event ID | Requester IP | Module | Event Name | Status | Error Code |
+-----+-----+-----+-----+-----+-----+-----+
| 2024-01-23 14:07:06 | f0ffa38c-7a8e-4253-9352-b19eb4af906c | 192.168.0.110 | Server | RemoveServer | 202 | - |
| 2024-01-23 13:55:09 | 1f312b19-49e5-440f-a132-e1fe2cf99385 | 1.202.233.200 | System | Login | 200 | - |
```

2024-01-23 11:10:46 26f47555-b9e6-4a76-bdfe-cbd4a459b4aa 192.168.0.110 Server SetPath 204 -
2024-01-23 11:07:09 a97ae9e8-9d92-429f-a947-c36ccf43fac8 192.168.0.110 Server AddPath 200 -
2024-01-23 11:01:08 69f5f9f0-57a7-487d-b495-0391d4da8856 192.168.0.110 Server SetServer 204 -
+-----+-----+-----+-----+-----+-----+-----+

用户事件描述

项目	描述
Event Time	HBlock 接收到用户事件请求的时间。
Event ID	用户事件 ID。
Requester IP	发起请求的源 IP 地址。
Moudle	用户事件所属模块。
Event Name	用户事件名称。 说明： 用户事件名称描述详见附录用户事件列表。
Status	响应状态码。
Error Code	错误码。

- 查询最近发生的 1 条系统事件信息。

<pre>[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor event ls --type system --num 1 event ID: d490beea-fc3e-423e-8ceb-c5d07b1d4f95 module: LUN instance ID: lun02a event time: 2022-11-04 16:12:16 name: ActiveStandbySwitched details: {"Target": "iqn.2012-08.cn.ctyunapi.oos:target02.3(192.168.0.72:3260,hblock_1)","status": "Transition to Active successfully"}</pre>
--

系统事件描述

项目	描述
event ID	系统事件 ID。
moudle	系统事件所属模块。
instance ID	系统事件针对的实例 ID。

	-: 表示无实例 ID。
event time	系统事件发生的时间。
name	系统事件名称。 说明：系统事件名称描述详见系统事件列表。
details	系统事件详情。 -: 表示无事件详情。

● 查询最近发生的 5 条系统事件。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor event ls --type system --num 5
```

Event Time	Event ID	Module	Event Name	Instance ID	Details
2022-08-01 14:57:14	ad9a053d-fa77-4494-b0c5-ad3c2131cf2a	Disk	PathAdded	hblock_1:/mnt/storage03	-
2022-08-01 09:51:24	fad76652-fc51-4384-8c83-e0fdd2ad0ecb	Disk	PathAdded	hblock_1:/mnt/storage02	-
2022-07-28 18:18:05	2f131000-7593-4b2f-a3ed-0b924e285e6c	Server	ServiceAvailable	hblock_1	{"message": "Service up", "service": "ws"}
2022-07-28 18:16:01	1eb1d604-e2eb-4018-bcfc-385e9c28bfff	Server	ServiceAvailable	hblock_1	{"message": "Service up", "service": "ms"}

系统事件描述

项目	描述
Event Time	系统事件发生的时间。
Event ID	系统事件 ID。
Moudle	事件所属模块。
Event Name	系统事件名称。 说明：事件名称描述详见系统事件列表。
Instance ID	系统事件针对的实例 ID。 -: 表示无实例 ID。
Details	系统事件详情。 -: 表示无系统事件详情。

4.17.2 导出 HBlock 事件

```
./stor event ls [ --type TYPE ] [ --num NUMBER ] { -o | --out } DIRECTORY
```

此命令用来导出 HBlock 事件信息。

参数

参数	描述
--type TYPE	指定导出 HBlock 事件的类型。 取值： ● system: 系统事件。 ● user: 用户事件。 默认值为 user 。
--num NUMBER	指定导出最近 HBlock 事件的个数。 取值: [1, 10000]。 说明 : 如果不指定, 默认导出系统记录的所有 HBlock 事件, 系统记录至少 6 个月的用户事件信息。
-o DIRECTORY 或 --out DIRECTORY	导出 HBlock 事件至指定文件夹。

示例

- 导出最近的 1000 条用户事件至文件夹 `/home/stor/events`。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor event ls --num 1000 -o /home/stor/events  
The output is in /home/stor/events/events_20220803172056.csv.
```

- 导出最近的 1000 条系统事件至文件夹 `/home/stor/events`。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor event ls --type system --num 1000 -o /home/stor/events  
The output is in /home/stor/events/events_20220803172118.csv.
```

4.17.3 发起 HBlock 日志采集

```
./stor logcollect add [ --start-time START_TIME ] [ --end-time END_TIME ] [ { -o | --out } DIIRECTORY ] [ --server SERVER_ID <1-n> ] [ --type LOG_TYPE <1-n> ]
```

此命令用来发起 HBlock 日志采集。

注意：日志采集的进程不能超过 10 个。

参数

参数	描述
--start-time <i>START_TIME</i>	HBlock 日志采集的起始时间。格式为 <i>yyyy-MM-dd HH:mm:ss</i>，默认值为 HBlock 日志采集结束时间 2 小时之前的时间点。 注意： ● HBlock 日志采集的起始时间必须早于 HBlock 日志采集的结束时间。 ● 如果 HBlock 日志采集的起始时间早于 HBlock 初始化时间，则 HBlock 初始化时间为日志采集起始时间。 ● 如果修改系统时间，可能导致日志文件的最后修改时间出现波动甚至错误。
--end-time <i>END_TIME</i>	HBlock 日志采集的结束时间。格式为 <i>yyyy-MM-dd HH:mm:ss</i>，默认值为当前时间。 注意： ● HBlock 日志采集的起始时间必须早于 HBlock 日志采集的结束时间。 ● HBlock 日志采集的结束时间必须晚于 HBlock 初始化时间。 ● 日志文件的最后修改时间大于结束时间时，如果存在多个大于结束时间的同类型日志，则 HBlock 日志采集时间值最小的那个日志文件。

-o <i>DIIRECTORY</i> 或 --out <i>DIIRECTORY</i>	HBlock 日志采集后存放的目录，为绝对路径。默认存放在被请求服务器的 HBlock 安装目录下，以 <code>collected_logs/hblock_logs_id_yyyyMMddHHmmss_yyyyMMddHHmmss.zip</code> 命名： ● id: 本次日志请求的唯一标识符。 ● yyyyMMddHHmmss: 日志采集的起始时间和结束时间，UTC+0 时间。
--server <i>SERVER_ID</i> &<1-n>	日志采集的服务器 ID，默认采集所有服务器的 HBlock 日志。可以采集 1 到 n 个服务器的日志，n 为集群中服务器的数量。
--type <i>LOG_TYPE</i>	采集的 HBlock 日志类型： ● Config: 配置相关的日志。 ● System: 系统相关的日志。 ● Data: 数据处理相关的日志（仅集群版支持）。 ● Coordination: 内部协调服务相关的日志（仅集群版支持）。 默认采集所有类型的日志。

示例

- 采集截止到当前时间 2 小时内的所有服务器的所有类型日志。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor logcollect add
Start collecting logs with ID f5dc5f0599a74be3af03441726707f01. After the logs are collected, they will be stored as a zip file in the /mnt/storage01/CTYUN_HBlock_Plus_3.9.0/collected_logs directory. You can check status with "./stor logcollect ls".
```

- 采集日志，并将日志文件存在路径/mnt/storage02 下。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor logcollect add --start-time 2022-08-05 10:00:00 --end-time 2022-08-06 00:00:00 -o /mnt/storage02
Start collecting logs with ID a745d931721f4ec48bcc389b00109a39. After the logs are collected, they will be stored as a zip file in the /mnt/storage02 directory. You can check status with "./stor logcollect ls".
```

4.17.4 查看 HBlock 采集的日志

```
./stor logcollect ls [ --id LOG_ID ]
```

此命令用来查看 HBlock 采集的日志。

参数

参数	描述
<code>--id LOG_ID</code>	日志 ID。如果不指定，则查看 HBlock 采集的所有日志。

示例

- 查看 HBlock 采集的所有日志。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor logcollect ls
```

No.	Log ID	File	Type	ServerNum	Status
1.	a745d931721f4ec48bcc389b00109a39	192.168.0.209:/mnt/storage02/hblock_logs_a745d931721f4ec48bcc389b00109a39_20220805020000_20220805160000.zip	All	3	Succeeded (1.48 GiB)
2.	9a5d6f75eae401c99787068a70be6cc	192.168.0.209:/mnt/storage01/CTYUN_HBlock_Plus_3.9.0/collected_logs/hblock_logs_9a5d6f75eae401c99787068a70be6cc_20220807020000_20220807160000.zip	All	3	Succeeded (1.3 GiB)
3.	f5dc5f0599a74be3af03441726707f01	192.168.0.209:/mnt/storage01/CTYUN_HBlock_Plus_3.9.0/collected_logs/hblock_logs_f5dc5f0599a74be3af03441726707f01_20220807231550_20220808011550.zip	All	3	Succeeded (434.35 MiB)
4.	061de53058f14e42a8bcc597173c2587	192.168.0.72:/mnt/logs/hblock_logs_061de53058f14e42a8bcc597173c2587_20220805020000_20220903020000.zip	All	3	Succeeded (656.5 MiB)
5.	6c709f434a97480486fb77226d1e6509	192.168.0.72:/mnt/storage01/CTYUN_HBlock_Plus_3.9.0/collected_logs/hblock_logs_6c709f434a97480486fb77226d1e6509_20220804073723_20220804093723.zip	All	3	Succeeded (431.43 MiB)
6.	547b0976c22f49319889e7e27b3f5f0d	192.168.0.72:/mnt/storage01/CTYUN_HBlock_Plus_3.9.0/collected_logs/hblock_logs_547b0976c22f49319889e7e27b3f5f0d_20220802063842_20220802083842.zip	All	3	Succeeded (346.79 MiB)

查看所有采集日志的信息描述

项目	描述
No.	序号。
Log ID	日志 ID。
File	日志文件存储路径。
Type	日志类型。
ServerNum	日志采集的服务器个数。
Status	日志采集的状态：

	● Processing: 日志采集中。 ● Succeeded: 采集成功。 ● PartiallySucceeded: 部分采集成功。 ● Failed: 采集失败。
--	--

- 查看日志 ID 为 **a745d931721f4ec48bcc389b00109a39** 的日志文件信息。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor logcollect ls --id a745d931721f4ec48bcc389b00109a39
Log ID: a745d931721f4ec48bcc389b00109a39
Create Time: 2022-08-08 09:45:37
File: 192.168.0.209:/mnt/storage02/hblock_logs_a745d931721f4ec48bcc389b00109a39_20220805020000_20220805160000.zip
Size: 1.48 GiB
Type: All
Servers: hblock_1, hblock_2, hblock_3
Status: Succeeded
Message: -
```

查看指定日志的信息描述

项目	描述
Log ID	日志 ID。
Create Time	采集日志请求的创建时间。
File	日志文件存储路径。
Size	采集到的日志文件大小。
Type	日志采集的类型。
Servers	日志采集的服务器 ID。
Status	日志采集的状态： ● Processing: 日志采集中。 ● Succeeded: 采集成功。 ● PartiallySucceeded: 部分采集成功。 ● Failed: 采集失败。
Message	如果日志采集失败或者部分采集成功，则会给出原因。

	-: 代表无信息返回。
--	-------------

4.17.5 删除 HBlock 采集的日志

```
./stor logcollect rm [ --id LOG_ID ]
```

此命令用来删除 HBlock 采集的日志。

参数

参数	描述
--id LOG_ID	日志 ID。如果不指定，则删除 HBlock 采集的所有日志。

示例

- 删除日志为 a2682b3a89b94c68924bd7e46703658f 的日志。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor logcollect rm --id  
a2682b3a89b94c68924bd7e46703658f  
Removed successfully.
```

4.18 HBlock 系统设置

4.18.1 修改管理员密码

```
./stor config set { -i | --item } auth { -p | --password } PASSWORD
```

此命令用来修改管理员密码。

参数

参数	描述
-i auth 或--item auth	修改管理员密码。
-p PASSWORD 或--password PASSWORD	管理员密码。 取值：字符串形式，长度范围 8~16，至少包含以下字符中的 3 种：大写字母、小写字母、数字、特殊字符 (~!@#\$%^&*()_+[]{} ;:.,/<>?), 区分大小写。不能包含：3 个连续重复的字符，3 个连续或反序的数字、或字母（不区分大小写），3 个连续或反序的键盘序列（不区分大小写）。

示例

修改管理员密码。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor config set -i auth -p *****  
Set successfully.
```

4.18.2 邮件设置

4.18.2.1 设置邮件通知功能

```
./stor config set { -i | --item } email [ { -H | --smtp-host } SMTP_HOST ] [ { -L | --ssl } SSL_STATUS ] [ { -P | --smtp-port } SMTP_PORT ] [ { -R | --receiver-email } RECEIVER_EMAIL &<1-n> ] [ { -S | --sender-email } SENDER_EMAIL { -p | --password } PASSWORD ] { -s | --status } STATUS [ -T | --test-email ]
```

此命令用来设置邮件通知功能。

注意：如果邮件服务器地址是 IPv6 地址，为了确保邮件能发送成功，建议每台 HBlock 服务器上都有一个能连接到邮件服务器的 IPv6 地址。

参数

参数	描述
-i email 或--item email	设置邮件通知功能。
-H SMTP_HOST 或--smtp-host SMTP_HOST	设置 SMTP 服务器。 取值：SMTP 服务器域名或 IP。
-L SSL_STATUS 或--ssl SSL_STATUS	是否开启 SSL 功能： ● Enabled (on)：开启 SSL 功能。 ● Disabled (off)：禁用 SSL 功能。 默认值为 Disabled (off) 。
-P SMTP_PORT 或--smtp-port SMTP_PORT	设置 SMPT 端口号。整型，取值为[1, 65535]，如果开启了 SSL，默认端口号为 465；如果未开启 SSL，默认端口号为 25。
-R RECEIVER_EMAIL 或--receiver-email RECEIVER_EMAIL	设置收件箱。可以设置多个收件箱，邮箱之间使用英文逗号隔开。 邮箱格式 <i>local-part@domain</i> ： ● <i>local-part</i>：字符串形式，长度范围是

	1~64，可包含字母、数字、特殊字符（!#\$%&*+ - / = ? ^ _ ` { } ~ .），字母区分大小写。句点（.）不能作为首尾字符，也不能连续出现。 ● domain：以句点（.）分隔的字符串形式，长度范围是 1~255。通过句点（.）分隔开的每个字符串需要满足如下要求： ■ 长度 1~63。 ■ 可包含字母、数字、短横线（-），字母区分大小写。 ■ 顶级域名不能是纯数字。 ■ 短横线（-）不能作为首尾字符。
-S SENDER_EMAIL 或 --sender-email SENDER_EMAIL	设置发件箱。 邮箱格式 <i>local-part@domain</i>： ● local-part：字符串形式，长度范围是 1~64，可包含字母、数字、特殊字符（!#\$%&*+ - / = ? ^ _ ` { } ~ .），字母区分大小写。句点（.）不能作为首尾字符，也不能连续出现。 ● domain：以句点（.）分隔的字符串形式，长度范围是 1~255。通过句点（.）分隔开的每个字符串需要满足如下要求： ■ 长度 1~63。 ■ 可包含字母、数字、短横线（-），字母区分大小写。 ■ 顶级域名不能是纯数字。 ■ 短横线（-）不能作为首尾字符。

-p <i>PASSWORD</i> 或 --password <i>PASSWORD</i>	邮箱授权码。 说明： 授权码是邮箱推出的，用于第三方客户端登录的专用密码。
-s <i>STATUS</i> 或 --status <i>STATUS</i>	是否开启邮件通知功能： ● Enabled (on)：开启邮件通知功能。 ● Disabled (off)：禁用邮件通知功能。
-T 或 --test-email	开启发送测试邮件功能。

示例

设置开启邮件通知功能。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor config set -i email -H smtp.
ctyun.cn -P 475 -R test-1@ctyun.cn,test.2@ ctyun.cn -S test-2@chinatelecom.cn -p *****
-s on
Set successfully.
```

4.18.2.2 删除邮件配置

```
./stor config rm { -i | --item } email
```

此命令用来删除邮件配置。

示例

删除邮件配置信息。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor config rm -i email  
Removed successfully.
```

4.18.2.3 邮件配置查询

```
./stor config ls { -i | --item } email
```

此命令用来查询邮件配置信息。

示例

邮件配置信息查询。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor config ls -i email
Status: Enabled
SMTP server: smtp.ctyun.cn
SMTP port: 475
Sender Email: test@ctyun.cn
Enable SSL(Yes/No): No
Receiver Email: test1@chinatelecom.cn
```

邮件配置查询信息描述

项目	描述
Status	是否开启邮件通知功能： ● Enabled: 开启邮件通知功能。 ● Disabled: 禁用邮件通知功能。
SMTP server	SMTP 服务器。
SMTP port	SMTP 端口号。
Sender Email	发件箱。
Enable SSL(Yes/No)	是否开启 SSL 功能： ● Yes: 开启 SSL 功能。 ● No: 禁用 SSL 功能。
Receiver Email	收件箱。

4.18.3 远程协助

4.18.3.1 设置远程协助

```
./stor config set { -i | --item } remote { -H | --host } HOST { -P | --port } PORT { -s | --status } STATUS [ { -n | --server } SERVER_ID ]
```

此命令用来设置远程协助功能。

参数

参数	描述
-i remote 或--item remote	设置远程协助功能。
-H HOST 或--host HOST	设置远程协助 HOST，远程协助的服务端 IP 或域名。 请联系软件供应商获取 HOST 信息。
-P PORT 或--port PORT	设置远程协助端口。整型，取值为[1,65535]，默认端口号为 18100。 请联系软件供应商获取 PORT 信息。
-s STATUS 或--status STATUS	是否开启远程协助功能： ● Enabled (on)：开启远程协助功能。 ● Disabled (off)：禁用远程协助功能。
-n SERVER_ID 或--server SERVER_ID	要配置远程协助的服务器 ID。 如果未指定服务器 ID，则为执行该命令的服务器配置远程协助。

示例

设置远程协助。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor config set -i remote -H 120.71.10.106 -P 18100 -s on
Remote access to 120.71.10.106:18100 has been enabled, and the remote access code is: 763589. Please tell the HBlock assistant the code or the HBlock ID (run 'stor info --stor-id' to get it).
```

4.18.3.2 删除远程协助配置

```
./stor config rm { -i | --item } remote [ { -n | --server } SERVER_ID ]
```

此命令用来删除远程协助配置。

参数

参数	描述
-n SERVER_ID 或 --server SERVER_ID	要删除远程协助的服务器 ID。 如果未指定服务器 ID，则为执行该命令的服务器删除远程协助。

示例

删除远程协助配置。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor config rm -i remote  
Removed successfully.
```

4.18.4 pushgateway 监控配置

4.18.4.1 添加 pushgateway 监控配置

```
./stor config add { -i | --item } monitor --pushgateway PUSHGATEWAY [ --pushgateway-labels LABEL <1-n> ] [ { -n | --server } SERVER_ID <1-n> ]
```

此命令用来添加 pushgateway 监控配置。

说明：添加 pushgateway 监控配置时，默认携带下列 label: job、集群 ID、集群名称、ServerIP、hostname。用户有权限修改或删除集群 ID、集群名称、ServerIP、hostname，但不建议修改和删除。用户添加的自定义 label 与默认 label 同名时，则 label 取值以用户设置的为准。

参数

参数	描述
--pushgateway PUSHGATEWAY	pushgateway 的地址和接口，格式为 <i>IPv4:port</i> 、 <i>[IPv6]:port</i> 或者 <i>domain-name:port</i> 。
--pushgateway-labels LABEL	pushgateway 对应的标签值，格式为 <i>KEY:VALUE</i> 。一次可以添加多个标签值，以英文逗号（,）分开。
-n SERVER_ID 或 --server SERVER_ID	指定添加 pushgateway 监控配置的 HBlock 服务器 ID。一次可以指定多个 HBlock 服务器的 ID，以英文逗号（,）分开。 默认为所有 HBlock 服务器添加 pushgateway 监控配置。

示例

添加 pushgateway 监控配置。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor config add -i monitor --
```

```
pushgateway 192.168.0.1:9091 --pushgateway-labels agent:agentname,idc:idcname -n  
hblock_1,hblock_2
```

Added pushgateway 192.168.0.1:9091 with pushgateway labels agent:agentname,idc:idcname
on server hblock_1,hblock_2 successfully.

4.18.4.2 修改 pushgateway 监控配置

```
./stor config set { -i | --item } monitor --pushgateway PUSHGATEWAY [ --pushgateway-labels LABEL <1-n> ] [ --all ] [ --pushgateway-timeout PUSHGATEWAY_TIMEOUT ] [ --collect-interval COLLECT_INTERVAL ] [ --collect-metric COLLECT_METRIC [ --collect-metric-items ITEM <1-n> ] ] [ { -n | --server } SERVER_ID <1-n> ]
```

此命令用来修改 pushgateway 监控配置。

说明：不建议修改默认携带的 label: job、集群 ID、集群名称、ServerIP、hostname。

参数

参数	描述
--pushgateway <i>PUSHGATEWAY</i>	需要修改 pushgateway 监控配置的地址和接口，格式为 <i>IPv4:port</i> 、 <i>[IPv6]:port</i> 或者 <i>domain-name:port</i> 。
--pushgateway-labels <i>LABEL</i>	pushgateway 对应的标签值，格式为 <i>KEY:VALUE</i> 。一次可以修改多个标签值，以英文逗号（,）分开。
--all	覆盖所有的 labels 或者所有的 metric 配置。如果 --pushgateway-labels <i>LABEL</i> 或 --collect-metric-items <i>ITEM</i> 未指定，此参数不生效。
--pushgateway-timeout <i>PUSHGATEWAY_TIMEOUT</i>	指定向 pushgateway 发送数据的超时时间。 取值：如果单位是 s，取值为[1, 1000]范围的整数；如果单位是 ms，取值为[1000, 1000000]范围的整数。默认单位是 s。
--collect-interval <i>COLLECT_INTERVAL</i>	指定采集时间间隔。 取值：如果单位是 s，取值为负整数、[2, 1000]范围的整数；如果单位是 ms，取值为负整数、

	[2000, 1000000]范围的整数。默认单位是 s。 注意： ● 如果与--collect-metric <i>COLLECT_METRIC</i> 同时指定时，取值可以为负整数，表示不采集指定的监控指标。 ● 如果不指定--collect-metric <i>COLLECT_METRIC</i> 时，表示设置所有监控指标的采集间隔，取值只能是允许范围内的正整数。
--collect-metric <i>COLLECT_METRIC</i>	指定采集的监控指标。 取值为：server、fileSystem、interface、load、disk、tcp、os。 默认采集的上述所有监控指标。
--collect-metric-items <i>ITEM</i>	指定监控指标下的配置项，目前仅监控指标 disk、fileSystem、interface 可以指定配置项。 说明： ● 如果指定 collectMetric，未指定配置项，默认修改指定 collectMetric 下的所有配置项。 ● 如果未指定 collectMetric，不能指定此参数。
-n <i>SERVER_ID</i> 或 --server <i>SERVER_ID</i>	指定要修改 pushgateway 监控配置的 HBlock 服务器 ID。一次可以修改多个 HBlock 服务器的 pushgateway 监控配置，以英文逗号（,）分开。 默认修改所有 HBlock 服务器的 pushgateway 监控配置。

示例

修改 pushgateway 监控配置。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor config set -i monitor --  
pushgateway 192.168.0.1:9091 --pushgateway-labels agent:agentname -all -n hblock_1  
Set all pushgateway labels {agent=agentname} of pushgateway 192.168.0.1:9091 on server  
hblock_1 successfully.
```

4.18.4.3 删除 pushgateway 监控配置

```
./stor config rm { -i | --item } monitor --pushgateway PUSHGATEWAY [ --pushgateway-labels LABEL <1-n> ] [ --collect-metric COLLECT_METRIC [ --collect-metric-items ITEM <1-n> ] ] [ { -n | --server } SERVER_ID <1-n> ]
```

此命令用来删除 pushgateway 监控配置。

注意：如果删除了“指标必须具备的 label”，可能会造成相关监控数据无法识别的风险。

参数

参数	描述
--pushgateway <i>PUSHGATEWAY</i>	需要删除 pushgateway 监控配置的地址和接口，格式为 <i>IPv4:port</i> 、 <i>[IPv6]:port</i> 或者 <i>domain-name:port</i> 。
--pushgateway-labels <i>LABEL</i>	pushgateway 对应的标签值，取值是 label 项。一次可以删除多个 labels 值，以英文逗号 (,) 分开。 取值：key 值。
--collect-metric <i>COLLECT_METRIC</i>	指定删除的监控指标。 取值为：server、fileSystem、interface、load、disk、tcp、os。 默认删除上述所有监控指标。
--collect-metric-items <i>ITEM</i>	指定监控指标下的要删除的具体 items 配置项。一次删除多个 items 配置项，以英文逗号 (,) 分开。
-n <i>SERVER_ID</i> 或 --server <i>SERVER_ID</i>	指定要删除 pushgateway 监控配置的 HBlock 服务器 ID。一次可以删除多个 HBlock 服务器的 pushgateway 监控配置，以英文逗号 (,) 分开。

	默认删除所有 HBlock 服务器的 pushgateway 监控配置。
--	--------------------------------------

示例

删除监控配置信息。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor config rm -i monitor --  
pushgateway 192.168.0.1:9091  
Removed pushgateway 192.168.0.1:9091 successfully.
```

4.18.4.4 查询 pushgateway 监控配置

```
./stor config ls { -i | --item } monitor [ { -n | --server } SERVER_ID <1-n> ]
```

此命令用来查询 pushgateway 监控配置信息。

参数

参数	描述
-n <i>SERVER_ID</i> 或 --server <i>SERVER_ID</i>	指定要查询 pushgateway 监控配置的 HBlock 服务器 ID。一次可以查询多个 HBlock 服务器的 pushgateway 监控配置，以英文逗号（,）分开。 默认查询所有 HBlock 服务器的 pushgateway 监控配置。

示例

查询所有服务器的 pushgateway 监控配置。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor config ls -i monitor
{
  "monitors" : [ {
    "serverId" : "hblock_1",
    "send" : {
      "interval" : 20000,
      "timeout" : 2000,
      "pushgatewayTimeout" : 10000,
      "pushgateways" : [ {
        "address" : "192.168.0.1",
        "port" : 9091,
        "labels" : {
          "agent" : "agentname",
          "idc" : "idcname",
          "hostname" : "hblockserver",
          "serverIp" : "192.168.0.110",
          "clusterId" : "B8BFE310-8DF6-429E-8C8A-ACD22310AC08",
```

```
        "clusterName" : "stor2",
        "job" : "hblock"
    }
} ]
},
"collect" : {
    "interval" : 14000,
    "metrics" : [ {
        "name" : "storFs",
        "interval" : 20000,
        "items" : [ "ds-1", "ds-2", "mdm" ]
    } ]
}
}, {
    "serverId" : "hblock_3",
    "send" : {
        "interval" : 20000,
        "timeout" : 2000,
        "pushgatewayTimeout" : 10000,
        "pushgateways" : [ ]
    },
    "collect" : {
        "interval" : 14000,
        "metrics" : [ {
            "name" : "storFs",
            "interval" : 20000,
            "items" : [ "ds-1" ]
        } ]
    }
}, {
    "serverId" : "hblock_2",
    "send" : {
        "interval" : 20000,
        "timeout" : 2000,
        "pushgatewayTimeout" : 10000,
```

```
"pushgateways" : [ {
  "address" : "192.168.0.1",
  "port" : 9091,
  "labels" : {
    "agent" : "agentname",
    "idc" : "idcname",
    "hostname" : "pm-006",
    "serverIp" : "192.168.0.192",
    "clusterId" : "B8BFE310-8DF6-429E-8C8A-ACD22310AC08",
    "clusterName" : "stor2",
    "job" : "hblock"
  }
} ]
},
"collect" : {
  "interval" : 14000,
  "metrics" : [ {
    "name" : "storFs",
    "interval" : 20000,
    "items" : [ "ds-1", "mdm" ]
  } ]
}
} ]
}
```

查询服务器 hblock_1、hblock_2 的 pushgateway 监控配置。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor config ls -i monitor -n
hblock_1,hblock_2
{
  "monitors" : [ {
    "serverId" : "hblock_1",
    "send" : {
      "interval" : 20000,
      "timeout" : 2000,
      "pushgatewayTimeout" : 10000,
      "pushgateways" : [ {
```

```
"address" : "192.168.0.1",
"port" : 9091,
"labels" : {
  "agent" : "agentname",
  "idc" : "idcname",
  "hostname" : "hblockserver",
  "serverIp" : "192.168.0.110",
  "clusterId" : "B8BFE310-8DF6-429E-8C8A-ACD22310AC08",
  "clusterName" : "stor2",
  "job" : "hblock"
}
} ]
},
"collect" : {
  "interval" : 14000,
  "metrics" : [ {
    "name" : "storFs",
    "interval" : 20000,
    "items" : [ "ds-1", "ds-2", "mdm" ]
  } ]
}
}, {
  "serverId" : "hblock_2",
  "send" : {
    "interval" : 20000,
    "timeout" : 2000,
    "pushgatewayTimeout" : 10000,
    "pushgateways" : [ {
      "address" : "192.168.0.1",
      "port" : 9091,
      "labels" : {
        "agent" : "agentname",
        "idc" : "idcname",
        "hostname" : "pm-006",
        "serverIp" : "192.168.0.192",
```

```

        "clusterId" : "B8BFE310-8DF6-429E-8C8A-ACD22310AC08",
        "clusterName" : "stor2",
        "job" : "hblock"
    }
} ]
},
"collect" : {
    "interval" : 14000,
    "metrics" : [ {
        "name" : "storFs",
        "interval" : 20000,
        "items" : [ "ds-1", "mdm" ]
    } ]
}
} ]
}

```

pushgateway 监控配置查询信息描述

项目	描述
serverId	服务器 ID。
send.pushgatewayTimeout	向所有 pushgateway 推送数据的超时时长，超时即认为推送失败。
send.pushgateways.adress	pushgateway 的 IP 或域名。
send.pushgateways.labels	pushgateway 对应的标签。
collect.interval	硬件指标采集的采集间隔。
collect.metrics.name	采集间隔或采集对象的硬件指标类别名称。
collect.metrics.interval	硬件指标类别单独设置的采集间隔。
collect.metrics.items	硬件指标类别单独设置的采集对象。

4.18.5 智维推送告警配置

说明：使用天翼云智维平台做运维的用户，可以通过在 HBlock 上配置此功能，将 HBlock 告警推送至智维平台。

4.18.5.1 添加向智维推送告警的配置

```
./stor config add { -i | --item } ctyuncms --url URL --labels LABEL &<1-n>
```

此命令用来添加向智维平台推送告警的配置。

参数

参数	描述
-i ctyuncms 或 --item ctyuncms	增加向智维平台推送告警的配置。
--url URL	智维平台第三方告警的推送地址。
--labels LABEL	推送至智维平台的告警数据中携带的标签信息，格式为 <i>KEY:VALUE</i> 。一次可以添加多个 Label 信息，以英文逗号（,）分开。 注意： 标签 <i>idc</i> 和 <i>mid</i> 必须添加。

示例

增加向智维平台推送告警的配置。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor config add -i ctyuncms --url
https://ctyuncms:9527/blackhole/api/v1/issue --labels
mid:65de96adf2faca1f3a75b33a,idc:neimeng
Added ctyuncms https://ctyuncms:9527/blackhole/api/v1/issue with labels
mid:65de96adf2faca1f3a75b33a,idc:neimeng successfully.
```

4.18.5.2 修改向智维平台推送告警的配置

```
./stor config set { -i | --item } ctyuncms --url URL --labels LABEL &<1-n> [ --all ]
```

此命令用来修改向智维平台推送告警的配置。

参数

参数	描述
-i ctyuncms 或 --item ctyuncms	修改向智维平台推送告警的配置。
--url URL	智维平台第三方告警的推送地址。
--labels LABEL	推送至智维平台的告警数据中携带的标签信息，格式为 KEY:VALUE 。一次可以修改多个 Label 信息，以英文逗号（,）分开。
--all	使用此次指定的 Label 参数替换原来所有的 Lable 信息。

示例

修改向智维平台推送告警的配置标签。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor config set -i ctyuncms --url https://ctyuncms:9527/blackhole/api/v1/issue --labels mid:65de96adf2faca1f3a75b33a
Set labels mid:65de96adf2faca1f3a75b33a of ctyuncms
https://ctyuncms:9527/blackhole/api/v1/issue successfully.
```

使用此次的 Label 参数 **mid:65de96adf2faca1f3a75b33a,idc:neimeng** 替换原来所有的 Lable 信息。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor config set -i ctyuncms --url https://ctyuncms:9527/blackhole/api/v1/issue --labels mid:65de96adf2faca1f3a75b33a,idc:neimeng --all
Set all labels mid:65de96adf2faca1f3a75b33a,idc:neimeng of ctyuncms
https://ctyuncms:9527/blackhole/api/v1/issue successfully.
```

4.18.5.3 删除向智维平台推送告警的配置

```
./stor config rm { -i | --item } ctyuncms --url URL [ --labels LABEL <1-n> ]
```

此命令用来删除向智维平台推送告警的配置。

参数

参数	描述
-i ctyuncms 或 --item ctyuncms	删除向智维平台推送告警的配置。
--url URL	智维平台第三方告警的推送地址。
--labels LABEL	推送至智维平台的告警数据中携带的标签信息，取值为标签的 <i>KEY</i> 。一次可以删除多个标签信息，以英文逗号（,）分开。 注意： idc 和 mid 不能删除。

示例

- 删除向智维平台推送告警的配置标签：hblockname、clusterID。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor config rm -i ctyuncms --url
https://ctyuncms:9527/blackhole/api/v1/issue --labels hblockname,clusterID
Removed labels hblockname,clusterID of ctyuncms
https://ctyuncms:9527/blackhole/api/v1/issue successfully.
```

- 删除向智维平台 https://ctyuncms:9527/blackhole/api/v1/issue 推送告警的所有配置。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor config rm -i ctyuncms --url
https://ctyuncms:9527/blackhole/api/v1/issue
Removed ctyuncms https://ctyuncms:9527/blackhole/api/v1/issue successfully.
```

4.18.5.4 查询向智维平台推送告警的配置

```
./stor config ls { -i | --item } ctyuncms
```

此命令用来查询向智维平台推送告警的配置信息。

参数

参数	描述
-i ctyuncms 或 --item ctyuncms	查询向智维平台推送告警的配置信息。

示例

查询向智维平台推送告警的配置信息。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor config ls -i ctyuncms
CtyunCMS url: https://ctyuncms:9527/blackhole/api/v1/issue
CtyunCMS labels:
    mid: 65de96adf2faca1f3a75b33a
    idc: neimeng
```

向智维平台推送告警的配置查询信息描述

项目	描述
CtyunCMS url	智维平台第三方告警的推送地址。
CtyunCMS labels	推送至智维平台的告警数据中携带的标签信息。

4.19 调整 HBlock 性能参数（集群版适用）

4.19.1 调整 HBlock 性能参数

```
./stor tuning set [ --perf-scen PERF_SCEN ] [ --ha-sens HA_SENS ]
```

根据硬件环境和业务场景，可以使用此命令进行性能调优。

参数

参数	描述
<code>--perf-scen PERF_SCEN</code>	设置性能调优针对的场景。 取值： ● ICv1：智算场景。 ● VMv1：虚拟化场景。 ● default：系统默认的类型。
<code>--ha-sens HA_SENS</code>	设置性能调优针对的场景敏感型。 取值： ● High：高敏感型场景。 ● Middle：中等敏感型场景。 ● Low：低敏感型场景。 默认值为 Middle。

示例

HBlock 性能调优。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor tuning set --perf-scen ICv1 --ha-sens High
Start adjusting parameters for optimization. You can use 'stor tuning ls' to check whether it is
completed.
```

4.19.2 查看性能调优配置

`./stor tuning ls`

此命令用来查看性能调优的配置。

示例

查看性能调优的配置。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor tuning ls
The current optimization is adapted to the following scenarios.
Scenario for performance optimization: ICv1
Sensitivity for high availability: High
```

性能调优查询信息描述

项目	描述
Scenario for performance optimization	性能调优针对的场景： ● ICv1: 智算场景。 ● VMv1: 虚拟化场景。 ● default: 系统默认的类型。
Sensitivity for high availability	性能调优针对的场景敏感型： ● High: 高敏感型。 ● Middle: 中等敏感型。 ● Low: 低敏感型。 ● custom: 用户设置的场景。 ● Unknown: 场景未知。

4.20 停止服务器上的 HBlock

```
./stor stop [ { -f | --force } ]
```

此命令用来停止当前服务器上的 HBlock 服务。

参数

参数	描述
-f 或 --force	强制停止当前服务器上的 HBlock 服务。 注意： 当停止服务器上的 HBlock 失败时，可以强制停止 HBlock 服务，但可能会造成数据丢失。

示例

- 停止当前服务器上的 HBlock 服务。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor stop
Stopping HBlock...
Stop completely.
```

- 强制停止当前服务器上的 HBlock 服务。

注意：强制停止 HBlock 服务，有可能造成数据丢失，请慎用。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor stop --force
Stopping HBlock...
Stop completely.
```

4.21 启动服务器上的 HBlock

```
./stor start [ { -t | --timeout } TIMEOUT ]
```

此命令用来启动当前服务器上的 HBlock 服务。

参数

参数	描述
-t <i>TIMEOUT</i> 或 --timeout <i>TIMEOUT</i>	设置启动当前服务器上 HBlock 服务的最大等待时间。如果在最大时间内 HBlock 服务没有启动成功，则报启动失败。 取值：[1, 2147483647]，单位是秒（s）。

示例

启动当前服务器上的 HBlock 服务。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor start
Starting HBlock...
Start completely.
```

4.22 重启服务器上的 HBlock

```
./stor restart [ { -f | --force } ]
```

此命令用来重启当前服务器上的 HBlock 服务。

参数

参数	描述
-f 或 --force	强制重启当前服务器上的 HBlock 服务。 注意： 强制重启 HBlock 服务，可能会造成数据丢失。

示例

- 重启当前服务器上的 HBlock 服务。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor restart
Stopping HBlock...
Stop completely.
Starting HBlock...
Start completely.
```

- 强制重启当前服务器上的 HBlock。

注意：强制重启 HBlock 服务，可能会造成数据丢失。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor restart --force
Stopping HBlock...
Stop completely.
Starting HBlock...
Start completely.
```

4.23 卸载 HBlock

```
./stor uninstall [ -l | --del-local ] [ -c | --del-cloud ]
```

此命令用来卸载 HBlock。

注意：对于集群版，使用此命令，会卸载集群中所有服务器上的 HBlock。

参数

参数	描述
-l 或--del-local	卸载 HBlock，并删除本地 HBlock 数据。
-c 或--del-cloud	卸载 HBlock，并删除云上 HBlock 数据。 说明： 卸载 HBlock 时，如果选择删除云上 HBlock 数据，卸载过程可能因各种原因（如网络问题、权限问题等）而无法成功完成。在这种情况下，用户可以选择删除云上数据，先进行 HBlock 卸载操作。卸载成功后，用户再登录到云服务，手动删除 HBlock 数据。

示例

卸载 HBlock。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor uninstall
All servers will be uninstalled. The data on the local or in the cloud will not be
deleted by default. If you want to delete, please use the -l or -c option. The deleted
data cannot be recovered, please exercise caution. Are you sure you want to uninstall
HBlock? [Yes/No]
[0]
y
Start uninstalling HBlock, please wait.
Processing...
Uninstalled successfully.
```

4.24 查看 HBlock 版本

```
./stor { --version | -v }
```

此命令用来查看 HBlock 版本。

示例

查看 HBlock 版本。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor --version  
3.9.0
```

4.25 升级 HBlock

4.25.1 升级 HBlock

```
./stor upgrade { --filename | -f } file [ --archfile PACKAGE <1-n> ] [ --key VALUE ]
```

此命令用来升级 HBlock。对于集群，只需要在一台服务器上执行升级即可。

升级过程中系统会做检查，如果不符合升级条件，可能会导致升级失败，升级失败的原因可以在执行升级操作的服务器上查看日志 `upgrade.log`（日志路径：*HBlock 安装目录* `/logs/ops/upgrade.log`）。建议升级之前对系统进行检查，确保：

- 所有状态正常：
 - 所有 HBlock 服务器连接正常，没有处于删除状态的服务器。
 - HBlock 处于 `working`、`upgrading` 状态。
 - 软件许可证证书在有效期内，且本地卷总容量未超过许可证容量限制；或者试用期未结束。
- 如果卷是高可用类型，至少保证卷的主备连接正常。
- 没有处于还原状态、删除状态的卷。
- 系统整体数据冗余度不降级，正常数据百分比为 100%。并且，可用故障域数量和健康数量大于所有卷的写入需求。
- 升级监听服务（`stor:ua`）正常。
- 协议解析服务（`stor:ps`）正常。
- 基础服务正常：元数据管理服务（`stor:mdm`）、日志服务（`stor:ls`）、协调服务（`stor:cs`）。
- HBlock 使用的磁盘不存在 `Error` 状态。

注意：执行升级 HBlock 前，确保每个服务器的 HBlock 安装路径对应的文件系统，存在至少 1 GiB 的可用空间。

参数

参数	描述
--filename <i>file</i> 或 -f <i>file</i>	当前服务器升级的目标版本 HBlock 安装包具体路径和具体文件名称。 说明：当前服务器升级使用的安装包，必须保证与该服务器使用的安装包类型匹配。 注意：安装包的具体路径不要含有 -、--及空格。
--archfile <i>PACKAGE</i>	需升级的集群中其他不同架构服务器的安装包路径及文件名（仅集群版支持）。如果有多种不同类型服务器，应添加多个安装包，使用英文逗号隔开。 注意：当集群内存在与执行升级服务器架构不同的服务器时，才需指定此参数，用于指定相应架构服务器的 HBlock 安装包。
--key <i>VALUE</i>	升级至 HBlock 目标版本时，若该版本需导入参数，则须设置此项。具体参数设置应依据目标版本而定。若目标版本需多个参数，可多次按 “--key VALUE” 格式添加，其中 “key” 为参数名，“VALUE” 为参数值。

示例

升级 HBlock 服务：从 HBlock 3.8.0 升级至 HBlock 3.9.0。

```
[root@hblockserver CTYUN_HBlock_Plus_3.8.0_x64]# ./stor upgrade -f
/mnt/storage01/CTYUN_HBlock_Plus_3.9.0_x64.zip
Starting upgrade, the current version is 3.8.0, the target version is 3.9.0
Start time: 2025-04-15 18:50:05
Server info:
hblock_1, 192.168.0.65, hblockserver
hblock_2, 192.168.0.64, k8s-master
hblock_3, 192.168.0.67, songt-0006
```

```
2025-04-15 18:50:06 CST+0800 [Step 1/5] Checking for system status...
2025-04-15 18:50:06 CST+0800 Checking system info
2025-04-15 18:50:06 CST+0800 Checking data status

2025-04-15 18:50:36 CST+0800 [Step 2/5] Uploading update files...
2025-04-15 18:50:39 CST+0800 Upload [100%] for hblock_1 ###[server 1/3]
2025-04-15 18:50:42 CST+0800 Upload [100%] for hblock_2 ###[server 2/3]
2025-04-15 18:50:44 CST+0800 Upload [100%] for hblock_3 ###[server 3/3]

2025-04-15 18:50:44 CST+0800 [Step 3/5] Upgrade preparing...

2025-04-15 18:51:04 CST+0800 [Step 4/5] Installing updates...
2025-04-15 18:51:04 CST+0800 Upgrading config - Update[✓] ### [module 1/10]
2025-04-15 18:51:13 CST+0800 Upgrading cs for hblock_1 - Check[✓], Stop[✓], Update[✓], Start[✓] ###
[server 1/3] [module 2/10]
2025-04-15 18:52:07 CST+0800 Upgrading cs for hblock_2 - Check[✓], Stop[✓], Update[✓], Start[✓] ###
[server 2/3] [module 2/10]
2025-04-15 18:53:01 CST+0800 Upgrading cs for hblock_3 - Check[✓], Stop[✓], Update[✓], Start[✓] ###
[server 3/3] [module 2/10]
2025-04-15 18:53:54 CST+0800 Upgrading ls for hblock_1 - Check[✓], Stop[✓], Update[✓], Start[✓] ###
[server 1/3] [module 3/10]
2025-04-15 18:55:18 CST+0800 Upgrading ls for hblock_2 - Check[✓], Stop[✓], Update[✓], Start[✓] ###
[server 2/3] [module 3/10]
2025-04-15 18:56:41 CST+0800 Upgrading ls for hblock_3 - Check[✓], Stop[✓], Update[✓], Start[✓] ###
[server 3/3] [module 3/10]
2025-04-15 18:57:33 CST+0800 Upgrading mdm for hblock_2 - Check[✓], Stop[✓], Update[✓], Start[✓] ###
[server 1/3] [module 4/10]
2025-04-15 18:58:06 CST+0800 Upgrading mdm for hblock_1 - Check[✓], Stop[✓], Update[✓], Start[✓] ###
[server 2/3] [module 4/10]
2025-04-15 18:58:40 CST+0800 Upgrading mdm for hblock_3 - Update[✓] ### [server 3/3] [module 4/10]
2025-04-15 18:58:44 CST+0800 Upgrading ds-1 for hblock_1 - Check[✓], Stop[✓], Update[✓], Start[✓] ###
[service 1/1] [server 1/3] [module 5/10]
2025-04-15 18:59:05 CST+0800 Upgrading ds-1 for hblock_2 - Check[✓], Stop[✓], Update[✓], Start[✓] ###
[service 1/2] [server 2/3] [module 5/10]
2025-04-15 18:59:54 CST+0800 Upgrading ds-2 for hblock_2 - Check[✓], Stop[✓], Update[✓], Start[✓] ###
```

```
[service 2/2] [server 2/3] [module 5/10]
2025-04-15 19:00:14 CST+0800 Upgrading ds-1 for hblock_3 - Check[✓], Stop[✓], Update[✓], Start[✓] ###
[service 1/1] [server 3/3] [module 5/10]
2025-04-15 19:00:35 CST+0800 Upgrading rb for hblock_1 - Check[✓], Stop[✓], Update[✓] ### [server 1/3]
[module 6/10]
2025-04-15 19:00:46 CST+0800 Upgrading rb for hblock_2 - Check[✓], Stop[✓], Update[✓] ### [server 2/3]
[module 6/10]
2025-04-15 19:00:57 CST+0800 Upgrading rb for hblock_3 - Check[✓], Stop[✓], Update[✓] ### [server 3/3]
[module 6/10]
2025-04-15 19:01:08 CST+0800 Upgrading ps for hblock_1 - Check[✓], Stop[✓], Update[✓], Start[✓] ###
[server 1/3] [module 7/10]
2025-04-15 19:01:34 CST+0800 Upgrading ps for hblock_2 - Check[✓], Stop[✓], Update[✓], Start[✓] ###
[server 2/3] [module 7/10]
2025-04-15 19:01:58 CST+0800 Upgrading ps for hblock_3 - Check[✓], Stop[✓], Update[✓], Start[✓] ###
[server 3/3] [module 7/10]
2025-04-15 19:02:28 CST+0800 Upgrading tool - Update[✓] ### [module 8/10]
2025-04-15 19:02:34 CST+0800 Upgrading monitor for hblock_1 - Check[✓], Stop[✓], Update[✓], Start[✓]
### [server 1/3] [module 9/10]
2025-04-15 19:02:51 CST+0800 Upgrading monitor for hblock_2 - Check[✓], Stop[✓], Update[✓], Start[✓]
### [server 2/3] [module 9/10]
2025-04-15 19:03:07 CST+0800 Upgrading monitor for hblock_3 - Check[✓], Stop[✓], Update[✓], Start[✓]
### [server 3/3] [module 9/10]
2025-04-15 19:03:23 CST+0800 Upgrading ws for hblock_1 - Check[✓], Stop[✓], Update[✓], Start[✓] ###
[service 1/2] [server 1/3] [module 10/10]
2025-04-15 19:03:38 CST+0800 Upgrading ms for hblock_1 - Check[✓], Stop[✓], Update[✓], Start[✓] ###
[service 2/2] [server 1/3] [module 10/10]
2025-04-15 19:03:56 CST+0800 Upgrading ws for hblock_2 - Check[✓], Stop[✓], Update[✓], Start[✓] ###
[service 1/2] [server 2/3] [module 10/10]
2025-04-15 19:04:10 CST+0800 Upgrading ms for hblock_2 - Check[✓], Stop[✓], Update[✓], Start[✓] ###
[service 2/2] [server 2/3] [module 10/10]
2025-04-15 19:04:28 CST+0800 Upgrading ws for hblock_3 - Check[✓], Stop[✓], Update[✓], Start[✓] ###
[service 1/2] [server 3/3] [module 10/10]
2025-04-15 19:04:42 CST+0800 Upgrading ms for hblock_3 - Check[✓], Stop[✓], Update[✓], Start[✓] ###
[service 2/2] [server 3/3] [module 10/10]
```

2025-04-15 19:05:21 CST+0800 [Step 5/5] Updating system info...

2025-04-15 19:05:21 CST+0800 [Post Upgrade] Cleaning...

2025-04-15 19:05:21 CST+0800 Cleaning hblock_2 ###[server 1/3]

2025-04-15 19:05:21 CST+0800 Cleaning hblock_3 ###[server 2/3]

2025-04-15 19:05:21 CST+0800 Cleaning hblock_1 ###[server 3/3]

2025-04-15 19:05:21 CST+0800 [Upgrade Complete] The system has been successfully upgraded to 3.9.0 version

4.25.2 查看升级状态

`./stor upgrade status`

此命令用来查看升级 HBlock 状态。

示例

查看升级状态。

```
[root@hblockserver CTYUN_HBlock_Plus_3.8.0_x64]# ./stor upgrade status
The last upgrade operation was successfully executed. The target version is 3.9.0.
End time: 2025-04-15 19:05:21
Server info:
hblock_1, 192.168.0.65, hblockserve
hblock_2, 192.168.0.64, k8s-master
hblock_3, 192.168.0.67, songt-0006

2025-04-15 18:50:06 CST+0800 [Step 1/5] Checking for system status...
2025-04-15 18:50:06 CST+0800 Checking system info
2025-04-15 18:50:06 CST+0800 Checking data status

2025-04-15 18:50:36 CST+0800 [Step 2/5] Uploading update files...
2025-04-15 18:50:39 CST+0800 Upload [100%] for hblock_1 ###[server 1/3]
2025-04-15 18:50:42 CST+0800 Upload [100%] for hblock_2 ###[server 2/3]
2025-04-15 18:50:44 CST+0800 Upload [100%] for hblock_3 ###[server 3/3]

2025-04-15 18:50:44 CST+0800 [Step 3/5] Upgrade preparing...

2025-04-15 18:51:04 CST+0800 [Step 4/5] Installing updates...
2025-04-15 18:51:04 CST+0800 Upgrading config - Update[✓] ### [module 1/10]
2025-04-15 18:51:13 CST+0800 Upgrading cs for hblock_1 - Check[✓], Stop[✓], Update[✓], Start[✓] ###
[server 1/3] [module 2/10]
2025-04-15 18:52:07 CST+0800 Upgrading cs for hblock_2 - Check[✓], Stop[✓], Update[✓], Start[✓] ###
[server 2/3] [module 2/10]
2025-04-15 18:53:01 CST+0800 Upgrading cs for hblock_3 - Check[✓], Stop[✓], Update[✓], Start[✓] ###
[server 3/3] [module 2/10]
```

```
2025-04-15 18:53:54 CST+0800 Upgrading ls for hblock_1 - Check[✓], Stop[✓], Update[✓], Start[✓] ###  
[server 1/3] [module 3/10]  
2025-04-15 18:55:18 CST+0800 Upgrading ls for hblock_2 - Check[✓], Stop[✓], Update[✓], Start[✓] ###  
[server 2/3] [module 3/10]  
2025-04-15 18:56:41 CST+0800 Upgrading ls for hblock_3 - Check[✓], Stop[✓], Update[✓], Start[✓] ###  
[server 3/3] [module 3/10]  
2025-04-15 18:57:33 CST+0800 Upgrading mdm for hblock_2 - Check[✓], Stop[✓], Update[✓], Start[✓] ###  
[server 1/3] [module 4/10]  
2025-04-15 18:58:06 CST+0800 Upgrading mdm for hblock_1 - Check[✓], Stop[✓], Update[✓], Start[✓] ###  
[server 2/3] [module 4/10]  
2025-04-15 18:58:40 CST+0800 Upgrading mdm for hblock_3 - Update[✓] ### [server 3/3] [module 4/10]  
2025-04-15 18:58:44 CST+0800 Upgrading ds-1 for hblock_1 - Check[✓], Stop[✓], Update[✓], Start[✓] ###  
[service 1/1] [server 1/3] [module 5/10]  
2025-04-15 18:59:05 CST+0800 Upgrading ds-1 for hblock_2 - Check[✓], Stop[✓], Update[✓], Start[✓] ###  
[service 1/2] [server 2/3] [module 5/10]  
2025-04-15 18:59:54 CST+0800 Upgrading ds-2 for hblock_2 - Check[✓], Stop[✓], Update[✓], Start[✓] ###  
[service 2/2] [server 2/3] [module 5/10]  
2025-04-15 19:00:14 CST+0800 Upgrading ds-1 for hblock_3 - Check[✓], Stop[✓], Update[✓], Start[✓] ###  
[service 1/1] [server 3/3] [module 5/10]  
2025-04-15 19:00:35 CST+0800 Upgrading rb for hblock_1 - Check[✓], Stop[✓], Update[✓] ### [server 1/3]  
[module 6/10]  
2025-04-15 19:00:46 CST+0800 Upgrading rb for hblock_2 - Check[✓], Stop[✓], Update[✓] ### [server 2/3]  
[module 6/10]  
2025-04-15 19:00:57 CST+0800 Upgrading rb for hblock_3 - Check[✓], Stop[✓], Update[✓] ### [server 3/3]  
[module 6/10]  
2025-04-15 19:01:08 CST+0800 Upgrading ps for hblock_1 - Check[✓], Stop[✓], Update[✓], Start[✓] ###  
[server 1/3] [module 7/10]  
2025-04-15 19:01:34 CST+0800 Upgrading ps for hblock_2 - Check[✓], Stop[✓], Update[✓], Start[✓] ###  
[server 2/3] [module 7/10]  
2025-04-15 19:01:58 CST+0800 Upgrading ps for hblock_3 - Check[✓], Stop[✓], Update[✓], Start[✓] ###  
[server 3/3] [module 7/10]  
2025-04-15 19:02:28 CST+0800 Upgrading tool - Update[✓] ### [module 8/10]  
2025-04-15 19:02:34 CST+0800 Upgrading monitor for hblock_1 - Check[✓], Stop[✓], Update[✓], Start[✓]  
### [server 1/3] [module 9/10]  
2025-04-15 19:02:51 CST+0800 Upgrading monitor for hblock_2 - Check[✓], Stop[✓], Update[✓], Start[✓]
```

```
### [server 2/3] [module 9/10]
2025-04-15 19:03:07 CST+0800 Upgrading monitor for hblock_3 - Check[ ✓ ], Stop[ ✓ ], Update[ ✓ ], Start[ ✓ ]
### [server 3/3] [module 9/10]
2025-04-15 19:03:23 CST+0800 Upgrading ws for hblock_1 - Check[ ✓ ], Stop[ ✓ ], Update[ ✓ ], Start[ ✓ ] ###
[service 1/2] [server 1/3] [module 10/10]
2025-04-15 19:03:38 CST+0800 Upgrading ms for hblock_1 - Check[ ✓ ], Stop[ ✓ ], Update[ ✓ ], Start[ ✓ ] ###
[service 2/2] [server 1/3] [module 10/10]
2025-04-15 19:03:56 CST+0800 Upgrading ws for hblock_2 - Check[ ✓ ], Stop[ ✓ ], Update[ ✓ ], Start[ ✓ ] ###
[service 1/2] [server 2/3] [module 10/10]
2025-04-15 19:04:10 CST+0800 Upgrading ms for hblock_2 - Check[ ✓ ], Stop[ ✓ ], Update[ ✓ ], Start[ ✓ ] ###
[service 2/2] [server 2/3] [module 10/10]
2025-04-15 19:04:28 CST+0800 Upgrading ws for hblock_3 - Check[ ✓ ], Stop[ ✓ ], Update[ ✓ ], Start[ ✓ ] ###
[service 1/2] [server 3/3] [module 10/10]
2025-04-15 19:04:42 CST+0800 Upgrading ms for hblock_3 - Check[ ✓ ], Stop[ ✓ ], Update[ ✓ ], Start[ ✓ ] ###
[service 2/2] [server 3/3] [module 10/10]

2025-04-15 19:05:21 CST+0800 [Step 5/5] Updating system info...

2025-04-15 19:05:21 CST+0800 [Post Upgrade] Cleaning...
2025-04-15 19:05:21 CST+0800 Cleaning hblock_2 ###[server 1/3]
2025-04-15 19:05:21 CST+0800 Cleaning hblock_3 ###[server 2/3]
2025-04-15 19:05:21 CST+0800 Cleaning hblock_1 ###[server 3/3]

2025-04-15 19:05:21 CST+0800 [Upgrade Complete] The system has been successfully upgraded to 3.9.0 version
```

5 常见问题

5.1 部署环境要求

Q: 部署 HBlock 的基础设施环境和网络环境有什么基本要求?

A: HBlock 要求部署在具有 Linux 操作系统环境的服务器上, 需满足最低配置要求, HBlock 中的每台服务器均可通过网络访问其他服务器。另外用户需要保证内部应用可以访问 HBlock 服务。

Q: 部署 HBlock 集群版, 至少需要几台服务器?

A: 至少需要 3 台服务器。HBlock 在存储数据副本时, 会将副本存储在多个故障域, 这样就可以保证当任意节点故障时, 应用还可以通过访问其他故障域的副本数据来保证可靠性, 同时系统会对故障服务器和损坏磁盘上的数据进行重建。

Q: 部署了 HBlock 后, 客户现存应用系统对 HBlock 进行数据读写, 是否需要进行端口改造?

A: 不需要。HBlock 提供标准 iSCSI 端口, 支持 Windows 和 Linux 客户端, 一般情况下, 客户应用无需进行端口改造。

Q: 部署了 HBlock 后, 集群内的服务器 IP 地址变更会有什么影响?

A: 如果部署 HBlock 后:

- 如果 HBlock 集群内服务器通信使用的 IP 地址变更, 变更 IP 地址服务器上的 HBlock 服务不可用。
- 如果 HBlock 服务器与客户端连接的 IP 地址变更, 则会导致客户端无法连接到服务器。所以部署 HBlock 后, 已经使用的服务器 IP 地址不能变更。

Q: 部署 HBlock 是否要求配置 NTP 时间同步?

A: 不需要。通过 HBlock 进行数据存储的时候不依赖于系统内部时钟，不需要提前配置 NTP。

Q: 部署 HBlock 是否要求配置虚拟 IP 地址（VIP）？

A: 不需要。HBlock 采用分布式多控制器体系结构，不依赖于传统的 VIP 模式，只需要确保客户端连接到主备 Target 所在的服务器并启用多路径 I/O（MPIO），即可实现秒级故障切换。

5.2 常见操作

Q: 服务器重启后，HBlock 服务能否自动启动？

A: 可以执行下列步骤，实现在服务器重启后 HBlock 服务自动启动。

前提条件: HBlock 的安装目录和数据目录都已经完成开机自动挂载。

1. 复制 systemd 服务单元文件：

```
cp HBlock 安装目录/apps/tool/systemd/HBlock.service /usr/lib/systemd/system/
```

例如：

```
[root@hblockserver]# cp
/mnt/storage01/CTYUN_HBlock_Plus_3.9.0_x64/apps/tool/systemd/HBlock.service
/usr/lib/systemd/system/
[root@hblockserver]# cat /usr/lib/systemd/system/HBlock.service
[Unit]
Description=HBlock - Storage Resource Reutilization System
Documentation=https://www.ctyun.cn/products/hblock/
After=network-online.target
Wants=network-online.target
Before=iscsi.service iscsid.service

[Service]
Type=forking
KillMode=none
User=root
Group=root
TimeoutStartSec=660
TimeoutStopSec=660
RemainAfterExit=no

ExecStart=/mnt/storage01/CTYUN_HBlock_Plus_3.9.0_x64/stor start -t 600
ExecReload=/mnt/storage01/CTYUN_HBlock_Plus_3.9.0_x64/stor restart
ExecStop=/mnt/storage01/CTYUN_HBlock_Plus_3.9.0_x64/stor stop
PIDFile=/mnt/storage01/CTYUN_HBlock_Plus_3.9.0_x64/run/pid/ms.pid

Restart=on-failure
```

```
RestartSec=30
```

```
[Install]
```

```
WantedBy=multi-user.target
```

说明：/mnt/storage01/CTYUN_HBlock_Plus_3.9.0_x64 为安装路径。如果使用 root 用户安装，User、Group 使用默认值 root，如果其他用户安装的，请修改 User、Group 为对应的名称。

2. 重载 systemd 配置并启用服务：

```
sudo systemctl daemon-reload
```

```
sudo systemctl enable HBlock.service
```

Q：重启 HBlock 集群？

A：用户需要手动重启集群中每台服务器的 HBlock 服务。为了保证不影响用户的数据，建议按照以下步骤进行处理：

1. 停止客户端应用的读写操作。

2. 客户端断开 iSCSI 连接：

◆ Windows 客户端：点击磁盘右键进行脱机，然后在 **iSCSI 发起程序** 中断开 iSCSI 连接。

◆ Linux 客户端：执行下列命令

```
umount DIRECTORY_NAME_OR_PATH #断开连接
```

```
iscsiadm -m node -T iSCSI_TARGET_IQN -p SERVER_IP -u #如何注销到 target 的连接
```

3. 在 HBlock 所在的每台服务器上分别执行命令 **./stor stop**。

4. 执行重启 HBlock 服务，可以使用下列方法中的一种：

◆ 在 HBlock 所在的每台服务器上执行命令 **./stor start**。

◆ 直接重启 HBlock 所在的所有服务器：可以参考“**服务器重启后，HBlock 服务能否自动启动？**”，确保服务器重启后 HBlock 的所有服务自动重启。

Q：如何配置 HBlock 访问权限？

A: 为加强 HBlock 的安全性，可通过配置防火墙权限，限制 iSCSI 端口（如 iSCSI 端口为 3260）的访问来源 IP。请参考以下步骤操作：

1. 开启防火墙：**systemctl start firewalld**。
2. 配置允许 IP：
 - 对于 IPv4 地址：**firewall-cmd --permanent --add-rich-rule="rule family=ipv4 source address=IP port protocol=tcp port=3260 accept"**。
 - 对于 IPv6 地址：**firewall-cmd --permanent --add-rich-rule="rule family=ipv6 source address=IP port protocol=tcp port=3260 accept"**。
3. 重启防火墙：**firewall-cmd --reload**。
4. 开机自动启动：**systemctl enable firewalld.service**。

Q: 配置 Linux 集群版时，如何更改客户端连接到 HBlock 的超时时间？

A: 可以通过更改 `/etc/iscsi/iscsid.conf` 和 `/etc/multipath.conf` 配置文件，来控制客户端到 HBlock 的超时时间。

1) iSCSI: `/etc/iscsi/iscsid.conf`

➤ **node.conn[0].timeo.noop_out_interval**

iSCSI Initiator 发送 NOP OUT 的间隔时间，默认值 5，单位为秒。

➤ **node.conn[0].timeo.noop_out_timeout**

iSCSI Initiator 发出 NOP OUT 后，收到 NOP IN 响应的超时时间，默认值 5，单位为秒。

➤ **node.session.timeo.replacement_timeout**

达到 `noop_out_timeout` 后，iSCSI Initiator 向 SCSI 层或 multipath 层返回 IO 失败前等待 iSCSI session 重建的超时时间。

默认值 200，单位为秒。可选值如下：

- 0: 不等待 session 重建，立即返回 IO 失败。
- <0: 一直等待，直到 session 成功重建，或用户执行的 `iscsi session logout`。
- >0: 等待指定时间后，返回 IO 失败。

查看运行时 `replacement_timeout`:

```
cat /sys/class/iscsi_session/sessionXX/recovery_tmo
```

修改运行时的 replacement_timeout:

```
echo X > /sys/class/iscsi_session/sessionXX/recovery_tmo
```

注意：若 iSCSI 设备配置了 multipath,

则 /sys/class/iscsi_session/sessionXX/recovery_tmo 会被 multipath.conf 中的 fast_io_fail_tmo 覆盖。

2) Multipath: /etc/multipath.conf

➤ polling_interval

路径检测的间隔时间。默认值 5，单位为秒。路径正常时，间隔时间会每次翻倍，增加到 4*polling_interval。

路径检测方式由 path_checker 决定，一般为发送 TUR 命令，或 READ 第 0 个扇区。

➤ fast_io_fail_tmo

SCSI 传输层出现问题后，将 IO 失败返回到 multipath 层之前等待传输层重建的超时时间。

默认值 5，单位为秒。可选值：

- ≥ 0 : 超时时间，会覆盖
/sys/class/iscsi_session/sessionXX/recovery_tmo。
- off: 不修改 /sys/class/iscsi_session/sessionXX/recovery_tmo。

➤ no_path_retry

当某条路径失败后，IO 重试的次数，默认值 0。可选值：

- 0/fail: 立即失败。
- >0: 重试次数。
- queue: 一直重试。multipath -ll 查看设备信息，会显示 features "1 queue_if_no_path"。

若 no_path_retry=queue，且对应的路径失败，则该路径上已经在处理的 SCSI 命令会被阻塞，一直等待直到路径恢复，即上层应用感知不到 IO 失败。对于上层是集群的应用场景，可能会影响集群的故障检测及切换。

对于已经被阻塞的 SCSI 命令，可执行 **dmsetup message mpathX 0**

"fail_if_no_path"，使得该路径上的 SCSI 命令立即返回失败，避免上层应用无限期等待。

Q: 如果让客户端和 HBlock 服务端断开连接，应如何操作？

A: 对于 Windows 客户端，需要先进行脱机才能断开连接，先断开备连接，再断开主连接，否则可能丢失数据。

对于 Linux 客户端，需要先执行 sync 命令才能断开连接，否则可能丢失数据。

Q: 若 iSCSI Initiator 已经与某个 iSCSI Target 建立连接，之后 HBlock 再新建关联该 iSCSI Target 的 LUN，iSCSI Initiator 如何在不断开已有连接的情况下发现新的 LUN？

A: 根据客户端不同，可以使用如下方式发现新的 LUN：

- Windows: 在服务器管理->文件和存储服务器->卷->磁盘，点击刷新即可完成 LUN 的添加。
- Linux: 在挂载新建 LUN 前，需要在 Linux 客户端执行下列命令：

```
rescan-scsi-bus.sh          # 使用此命令前，系统需要安装 sg3_utils
```

例如：客户端已经连接 lun01，lun01 和 lun01-a 都对应 target01，Linux 客户端需要挂载 lun01-a。

- 服务器端：查询 LUN。

```
[root@hblockserver CTYUN_HBlock_Plus_3.9.0_x64]# ./stor lun ls
```

No.	LUN Name	Storage Mode	Capacity	Local Storage Class	Minimum Replica Number	Status	Target
1.	lun01a(LUN 0)	Local	1000 GiB	EC 2+1+16KiB	2	Enabled	iqn.2012-08.cn.ctyunapi.oos:target01.1(192.168.0.72:3260,Active)
							iqn.2012-08.cn.ctyunapi.oos:target01.2(192.168.0.209:3260,Standby)
2.	lun001-a(LUN 0)	Local	101 GiB	EC 2+1+16KiB	2	Enabled	iqn.2012-08.cn.ctyunapi.oos:target01.1(192.168.0.72:3260,Active)
							iqn.2012-08.cn.ctyunapi.oos:target01.2(192.168.0.209:3260,Standby)

- 客户端：

1. 执行命令 **rescan-scsi-bus.sh** 前:

```
[root@client ~]# lsscsi
[8:0:0:0]    disk    CTYUN    iSCSI LUN Device 1.00  /dev/sda
[9:0:0:0]    disk    CTYUN    iSCSI LUN Device 1.00  /dev/sdb
[root@client ~]# lsblk
NAME          MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
sda            8:0      0 1000G  0 disk
├─mpathh       252:0      0 1000G  0 mpath
│ └─mpathh1    252:1      0 1000G  0 part  /mnt/disk_mpathh
sdb            8:16      0 1000G  0 disk
├─mpathh       252:0      0 1000G  0 mpath
│ └─mpathh1    252:1      0 1000G  0 part  /mnt/disk_mpathh
vda            253:0      0   40G  0 disk
├─vda1         253:1      0    4G  0 part  [SWAP]
└─vda2         253:2      0   36G  0 part  /
vdb            253:16     0    1T  0 disk
└─vdb1         253:17     0 1024G  0 part
vdc            253:32     0    1T  0 disk
└─vdc1         253:33     0 1024G  0 part  /mnt/storage01
```

2. 执行 **rescan-scsi-bus.sh**, 并查看

```
[root@client ~]# lsscsi
[8:0:0:0]    disk    CTYUN    iSCSI LUN Device 1.00  /dev/sda
[9:0:0:0]    disk    CTYUN    iSCSI LUN Device 1.00  /dev/sdb
[root@client ~]# rescan-scsi-bus.sh
Scanning SCSI subsystem for new devices
Scanning host 0 for  SCSI target IDs  0 1 2 3 4 5 6 7, all LUNs
Scanning host 1 for  SCSI target IDs  0 1 2 3 4 5 6 7, all LUNs
Scanning host 8 for  SCSI target IDs  0 1 2 3 4 5 6 7, all LUNs
  Scanning for device 8 0 0 0 ...
OLD: Host: scsi8 Channel: 00 Id: 00 Lun: 00
      Vendor: CTYUN      Model: iSCSI LUN Device Rev: 1.00
      Type:   Direct-Access                      ANSI SCSI revision: 06
  Scanning for device 8 0 0 1 ...
NEW: Host: scsi8 Channel: 00 Id: 00 Lun: 01
```

```

Vendor: CTYUN    Model: iSCSI LUN Device Rev: 1.00
Type:   Direct-Access          ANSI SCSI revision: 06
Scanning host 9 for  SCSI target IDs  0 1 2 3 4 5 6 7, all LUNs
Scanning for device 9 0 0 0 ...
OLD: Host: scsi9 Channel: 00 Id: 00 Lun: 00
Vendor: CTYUN    Model: iSCSI LUN Device Rev: 1.00
Type:   Direct-Access          ANSI SCSI revision: 06
..... Scanning for device 9 0 0 1 ...
NEW: Host: scsi9 Channel: 00 Id: 00 Lun: 01
Vendor: CTYUN    Model: iSCSI LUN Device Rev: 1.00
Type:   Direct-Access          ANSI SCSI revision: 06
2 new or changed device(s) found.
[8:0:0:1]
[9:0:0:1]
0 remapped or resized device(s) found.
0 device(s) removed.
[root@client ~]# lsscsi
[8:0:0:0]    disk    CTYUN    iSCSI LUN Device 1.00  /dev/sda
[8:0:0:1]    disk    CTYUN    iSCSI LUN Device 1.00  /dev/sdc
[9:0:0:0]    disk    CTYUN    iSCSI LUN Device 1.00  /dev/sdb
[9:0:0:1]    disk    CTYUN    iSCSI LUN Device 1.00  /dev/sdd
[root@client ~]# lsblk
NAME          MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
sda           8:0      0 1000G  0 disk
└─mpathh      252:0      0 1000G  0 mpath
  └─mpathh1    252:1      0 1000G  0 part  /mnt/disk_mpathh
sdb           8:16      0 1000G  0 disk
└─mpathh      252:0      0 1000G  0 mpath
  └─mpathh1    252:1      0 1000G  0 part  /mnt/disk_mpathh
sdc           8:32      0  101G  0 disk
└─mpathi      252:2      0  101G  0 mpath
sdd           8:48      0  101G  0 disk
└─mpathi      252:2      0  101G  0 mpath
vda           253:0      0   40G  0 disk
└─vda1        253:1      0    4G  0 part  [SWAP]

```

```
└─vda2      253:2    0   36G  0 part  /
vdb         253:16   0    1T  0 disk
└─vdb1      253:17   0 1024G  0 part
vdc         253:32   0    1T  0 disk
└─vdc1      253:33   0 1024G  0 part  /mnt/storage01
```

此时，可以按照正常挂载 LUN 的步骤进行挂载新建的 LUN。

Q：如何确保 Linux 客户端在重启服务器之后能够直接挂载 HBlock 创建的 LUN？

A：在将 HBlock 创建的 LUN 挂载到客户端之后，可以参考以下步骤：

说明：

- 如果配置文件/etc/iscsi/iscsid.conf 中 node.startup=manual，可以使用下列方法任意一种：
 - 修改配置文件/etc/iscsi/iscsid.conf 中的 node.startup=automatic，然后依据/etc/multipath.conf 配置文件中的设置，选择挂载步骤。
 - 对每个 target 配置，在客户端重启时自动登录，执行下面的命令：**iscsiadm -m node -T Target-IQN -p SERVER_IP --op update -n node.startup -v automatic**，然后依据/etc/multipath.conf 配置文件中的设置，选择挂载步骤。
- 如果配置文件/etc/iscsi/iscsid.conf 中 node.startup = automatic，依据/etc/multipath.conf 配置文件中的设置，选择挂载步骤。

若/etc/multipath.conf 中配置了 user_friendly_names yes，可以按照下列步骤执行：

1. 在客户端使用命令 **lsblk -f** 查看挂载设备的文件系统信息，找到文件系统对应的 UUID：

```
[root@client ~]# lsblk -f
NAME            FSTYPE      LABEL UUID                                MOUNTPOINT
sda              mpath_member
└─mpatha
  └─mpatha1 ext4              7269eef6-e401-454a-acb4-503d33337f21 /mnt/disk_mpatha
sdb              mpath_member
└─mpatha
```

```

└─mpatha1 ext4          7269eef6-e401-454a-acb4-503d33337f21 /mnt/disk_mpatha
vda
├─vda1      swap        9e33bd6f-c68c-41c7-95c8-703f4fe8c3d4 [SWAP]
└─vda2      xfs         a83f4fdc-2ea1-4fec-a1e2-a42016ce0afe /
vdb
└─vdb1      ext4        74296a9e-8cfd-4708-89b1-08086f71175b
vdc
└─vdc1      ext4        a9fedea4-391e-4d2a-8824-c9a3a6853394

```

2. 在`/etc/fstab`文件中新增 HBlock 创建的 LUN 挂载信息，下次开机启动时可以自动挂载该 LUN。

```

UUID=7269eef6-e401-454a-acb4-503d33337f21 /mnt/disk_mpatha ext4 defaults,_netdev 0 0

```

若`/etc/multipath.conf`中配置了`user_friendly_names no`，可以按照下列步骤执行：

1. 在客户端使用命令 `lsblk -f` 查看挂载设备的文件系统信息：

```

[root@client ~]# lsblk -f
NAME                                FSTYPE     FSVER      LABEL    UUID                                  FSAVAIL  FSUSE%  MOUNTPOINTS
sda                                mpath_member
└─0x300000001e908caf ext4        1.0        9aa0f703-f0db-4db7-bda5-9989aaba0269  615.6G   0% /mnt/disk_lun06a
sdb                                mpath_member
└─0x300000001e908caf ext4        1.0        9aa0f703-f0db-4db7-bda5-9989aaba0269  615.6G   0% /mnt/disk_lun06a
sr0                                iso9660     Joliet Extension config-2 2025-01-08-11-16-53-00
vda
└─vda1      xfs         a74fe630-4089-4ae3-b9f8-95a915cf42d7  34.1G    15% /
vdb         ext4        1.0        f7728367-9021-4225-9fe6-129c36a4766a  91.5G    1% /mnt/storage01

```

2. 在`/etc/fstab`文件中新增 HBlock 创建的 LUN 挂载信息，下次开机启动时可以自动挂载该 LUN。

```

/dev/mapper/0x300000001e908caf /mnt/disk_lun06a ext4 defaults,_netdev 0 0

```

Q：HBlock 使用的数据目录，如何在服务器中设置开机自动挂载？

A：服务器使用 `mount` 命令挂载目录后，可以参考如下步骤开机自动挂载：

1. 在客户端使用命令 **lsblk -f** 查看挂载设备的文件系统信息，找到文件系统对应的 UUID:

```
[root@server ~]# lsblk -f
```

NAME	FSTYPE	LABEL	UUID	MOUNTPOINT
vda				
└─vda1	swap		9e33bd6f-c68c-41c7-95c8-703f4fe8c3d4	[SWAP]
└─vda2	xfss		a83f4fdc-2ea1-4fec-a1e2-a42016ce0afe	/
vdb				
└─vdb1	ext4		c62d513e-c3cf-4719-b15c-4366e4b52664	
vdc				
└─vdc1	ext4		1c47025a-6028-42ce-90aa-59d6f5106818	/mnt/storage01

2. 在 **/etc/fstab** 文件中新增挂载目录的信息，下次开机启动时会自动挂载该目录。

```
UUID=1c47025a-6028-42ce-90aa-59d6f5106818 /mnt/storage01 ext4 defaults 1 1
```

Q: 在 Linux 客户端，如何删除未与服务器端连接 iSCSI Target?

A:

- HBlock 未卸载，Linux 客户端删除与服务器端连接的 iSCSI Target:

1. Linux 客户端使用下列命令，删除磁盘断开连接:

```
umount DIRECTORY_NAME_OR_PATH
```

```
iscsiadm -m node -T iSCSI_TARGET_IQN -p SERVER_IP -u
```

2. Linux 客户端使用下列命令中的一种，删除与服务器断开的 iSCSI Target:

- 删除所有未与服务器端连接的 iSCSI Target

```
iscsiadm -m node --op delete
```

- 删除未与服务器端连接的指定的 iSCSI Target

```
iscsiadm -m node --targetname iSCSI_TARGET_IQN -p SERVER_IP:port -o delete
```

- 卸载 HBlock 后，Linux 客户端重启后，可以通过以下命令删除未与服务器端连接 iSCSI Target:

- 删除所有未与服务器端连接的 iSCSI Target

```
iscsiadm -m node --op delete
```

- 删除未与服务器端连接的指定的 iSCSI Target

```
iscsiadm -m node --targetname iSCSI_TARGET_IQN -p SERVER_IP:port -o delete
```

Q: 在 Linux 客户端，如何确认盘符与 HBlock 卷的对应关系？

A: 可以使用下列命令查询

```
sg_inq /dev/sdX # 输出信息中会展示 LUN 名称
```

```
udevadm info --query=all --path=/block/sdX #输出信息中会展示 Target、LUN 编号
```

或

```
iscsiadm -m session -P 3 #显示 Target portal 和对应的盘符
```

或

```
ll /dev/disk/by-path #列出所有磁盘
```

Q: HBlock 某些服务重启失败，怎么排查？

A: 请检查该服务使用的端口是否已被其他应用占用。HBlock 服务使用的端口值可以使用命令行 **./stor server ls --port** 查看，请避免系统内的其他服务占用这些端口号。并且确保 Linux 的本地临时端口（`ip_local_port_range`）范围不包含 HBlock 服务使用的端口号。使用命令行 **cat /proc/sys/net/ipv4/ip_local_port_range** 可以查看本地临时端口范围。

Q: 当 HBlock 服务端和客户端都重启后，且服务端比客户端延迟一段时间才启动，导致客户端无法自动重连，怎么处理？

A: 此问题是由于 HBlock 服务端比客户端延迟重启导致，尽量让 HBlock 客户端重启时间晚于 HBlock 服务器。如果不能避免，可以使用下列方法设置：

- 新创建的 Target：在/etc/iscsi/iscsid.conf 修改 iscsiadm 的登录超时重试次数

（node.session.initial_login_retry_max），如：

```
node.session.initial_login_retry_max=172800
```

默认每次重启时间间隔是 15s，用户可以根据实际情况设置重试次数。

- 针对已经发现过的 Target：修改

/etc/iscsi/nodes/iqn.2012-08.cn.ctyunapi.oos/IP,PORT/default 中的登录超时重试次数

（node.session.initial_login_retry_max）：

IP,PORT：分别为 Target 的 IP 地址和 iSCSI 端口号。

```
node.session.initial_login_retry_max=172800
```

默认每次重启时间间隔是 15s，用户可以根据实际情况设置重试次数。

说明：以上两个配置文件，系统不同，可能路径不同。

Q：当客户端已挂载 HBlock 卷，且使用了 XFS 文件系统格式化，此时在 HBlock 服务器端创建以此卷为源卷的克隆卷，挂载源卷的客户端同时挂载该卷的克隆卷，挂载不成功，怎么解决？

A：可用通过以下两种方式解决：

- 在客户端修改克隆卷的 UUID，然后再重新挂载。

```
xfs_admin -U generate /dev/sdX
mount /dev/sdX PATH
```

- 不修改克隆卷的 UUID，挂载时使用 nouuid 选项，忽略 UUID 检查。

```
mount -o nouuid /dev/sdX PATH
```

说明：/dev/sdX 为克隆卷设备名称，PATH 为磁盘路径。

示例：sda 为源卷，sdj、sdo 为克隆卷，直接挂载会报错，sdo 使用修改 UUID 方法挂载，sdj 使用忽略 UUID 检查方式挂载。

```
[root@songt-0007 /]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
sda          8:0    0  200G  0 disk  /mnt/sda
sdj          8:144   0   21G  0 disk
sdo          8:224   0  233G  0 disk
```

```
vda      253:0    0   40G  0 disk
└─vda1   253:1    0   40G  0 part  /
vdb      253:16   0  100G  0 disk  /mnt/local_stor01
vdc      253:32   0  100G  0 disk
vdd      253:48   0  100G  0 disk
[root@songt-0007 /]# mount /dev/sdo /mnt/sdo
mount: wrong fs type, bad option, bad superblock on /dev/sdo,
       missing codepage or helper program, or other error

       In some cases useful info is found in syslog - try
       dmesg | tail or so.
[root@songt-0007 /]# xfs_admin -U generate /dev/sdo
Clearing log and setting UUID
writing all SBs
new UUID = fe99e47e-8091-4a12-860c-f60e02b07b41
[root@songt-0007 /]# mount /dev/sdo /mnt/sdo
[root@songt-0007 /]# mount /dev/sdj /mnt/sdj
mount: wrong fs type, bad option, bad superblock on /dev/sdj,
       missing codepage or helper program, or other error

       In some cases useful info is found in syslog - try
       dmesg | tail or so.
[root@songt-0007 /]# mount -o nouuid /dev/sdj /mnt/sdj
[root@songt-0007 /]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
sda          8:0     0   200G  0 disk  /mnt/sda
sdg          8:96    0   555G  0 disk  /mnt/sdg
sdj          8:144   0   211G  0 disk  /mnt/sdj
vda         253:0    0    40G  0 disk
└─vda1      253:1    0    40G  0 part  /
vdb         253:16   0   100G  0 disk  /mnt/local_stor01
vdc         253:32   0   100G  0 disk
vdd         253:48   0   100G  0 disk
```

5.3 升级扩容

Q: HBlock 软件如何升级?

A: 在满足升级前提条件下，依照**升级 HBlock** 的升级步骤，即可完成升级操作。升级过程中可以执行 HBlock 查询类操作，无法执行 HBlock 修改、删除类的操作，但不会影响客户的正常读写业务。升级需要执行一段时间，如果升级失败，请按照提示尽快解决问题后重新发起升级请求，不建议不同版本的服务器长时间运行。升级遇到任何问题需要支撑，请联系天翼云客服。

Q: HBlock 支持哪些扩容方式?

A: HBlock 支持通过扩容新存储服务器和扩容现有节点中硬盘的方式，来扩容本地存储空间。并且不要求各个服务器上的磁盘数量一样。

Q: HBlock 扩容节点时，是否需要新节点的配置与原有节点一样?

A: 可以不一样。只要新扩容节点满足最低配置要求即可。各存储服务器节点的 CPU，内存，硬盘数量，硬盘大小可以不一样。

Q: HBlock 的自动化部署实现到什么程度?

A: 服务器上的操作系统、虚拟机、网络配置等工作需要借助第三方工具安装，或者人工操作完成。完成上述操作后，HBlock 软件可以通过命令行方式进行安装。需要在所有待安装服务器上执行 `install` 安装命令，然后在其中某一台服务器上执行 `setup` 初始化命令。

5.4 数据迁移

Q: HBlock 是否支持同传统 SAN 存储产品之间的数据迁移?

A: 可以通过虚拟化平台完成 HBlock 和其他 SAN 产品之间的数据迁移。对于非虚拟化场景，比如数据库，可以采用专门的数据迁移工具进行 SAN 与 HBlock 之间的数据迁移。此外，Linux 的内核模块 **dm-clone**，可以用于克隆块设备，将数据从原有的 LUN 切到新的 LUN，实现原有 SAN 存储到 HBlock 的数据迁移。Windows 操作系统可以使用**用户状态迁移工具**实现数据迁移。

5.5 数据安全

Q: 若出现磁盘毁损情况，是否导致数据丢失？

A: 由于 HBlock 支持多副本和纠删码冗余数据模式，数据可以通过分片的方式，存储在不同故障域上，当有磁盘损坏时，可以从同故障域其他磁盘或者不同故障域上自动恢复数据，大幅降低数据丢失风险。但因数据存储为用户本地的服务器中，用户需保证本地服务器和磁盘的稳定运行，当系统中的磁盘损毁情况过多时，还是会存在数据丢失风险。

Q: 若在传输过程中，出现断电，或者 HBlock 服务器宕机情况，是否导致数据丢失？

A: HBlock 支持集群版，可以部署在多台服务器上，在客户端正确配置 MPIO 的情况下，在单点故障发生时，服务会被自动切换到备节点，可以继续正常的数据读写。

Q: HBlock 的数据安全性、完整性和可靠性方面有哪些保障措施？

A: HBlock 支持多副本和纠删码数据冗余策略，采用时序传输、双重校验和数据缓存机制，尽可能保障数据安全、完整。同时，HBlock 支持集群版，可以快速完成故障切换，尽可能保证用户业务的连续性。

Q: 针对断电、磁盘毁损等意外事件，有哪些解决方案？

A: 当部署为单机版时，建议您配置 UPS 不间断电源，硬盘做 RAID，保证磁盘上数据的可靠性。当部署为集群版时，在配置合理的情况下，单服务器掉电、单盘损坏等单点故障会被自动进行切换。

Q: 集群中的服务器是否可以长时间处于宕机或者不启动 HBlock 服务？

A: 不建议服务器处于长时间宕机或者不启动 HBlock 服务器状态。

因为服务器再启动时，可能需要比较长的恢复时间，可能会影响业务。故建议如果集群中有服务器宕机，尽快重启或修复，并启动 HBlock 服务。

5.6 技术支持

Q: 使用 HBlock 后如果遇到问题，会有人上门解决吗？

A: 若安装和使用过程中有任何疑问，您可以通过下列方式联系我们：

- 天翼云官网咨询热线 400-810-9889。
- 天翼云官网工单管理系统。

如果需要远程协助：

- 已安装 HBlock 但未初始化：与天翼云工作人员联系后，根据天翼云工作人员提供的 HOST、PORT（默认端口号是 18100），使用命令 `./stor config set { -i | --item } remote {-H | --host} HOST {-P | --port} PORT { -s | --status } STATUS` 开启远程协助功能。然后将命令返回的远程协助码再告知天翼云工作人员，此时天翼云工作人员可以进行远程协助。
- 已完成安装、初始化 HBlock：您可以使用已安装 HBlock 但未初始化时的方式操作。也可以先通过 `./stor info --stor-id` 获取 HBlock ID，将问题和 HBlock ID 一起发给天翼云工作人员。并根据天翼云工作人员提供的 HOST、PORT（默认端口号是 18100），使用命令 `./stor config set { -i | --item } remote {-H | --host} HOST {-P | --port} PORT { -s | --status } STATUS` 开启远程协助功能，此时天翼云工作人员可以进行远程协助。

默认情况下，远程协助功能处于禁用状态，用户可以随时启用或禁用此功能。启用后，天翼云工作人员有权登录用户环境中的 HBlock 系统以诊断问题。天翼云工作人员远程协助登录时，具有以下两种用户的权限：开启远程协作操作的用户、安装 HBlock 的用户。用户可以通过 `logs/remoteaccess/remote_access.log` 查看工作人员远程协助时的所有操作。

注意：如果用户启用了远程协助功能，则意味着用户相信天翼云工作人员并授权天翼云工作人员访问 HBlock 系统中的所有数据。天翼云工作人员将尽力诊断问题并确保数据安全，但是由于系统环境的复杂性，我们对远程协助引起的任何后果不承担任何责任。

5.7 购买指南

Q: 如何获取软件许可证?

A: HBlock 提供 30 天试用期，到期后请联系 HBlock 软件供应商获取软件许可证，获取的时候需要提供 HBlock 序列号（可以通过命令 `./stor info --serial-id` 获取）。取得软件许可证后，执行 `./stor license add { -k | --key } KEY` 加载软件许可证。

5.8 其他问题

Q: 支持部署虚拟化应用吗?

A: 支持。支持用户部署的虚拟化应用包括: KVM、VMware。

Q: 支持部署数据库应用吗?

A: 支持。支持部署的数据库应用包括: Oracle (Oracle rac 除外)、MySQL、SQL Server、PostgreSQL、MongoDB、DB2。

Q: 怎么快速判断集群中各服务器网络连通状态?

A: 除了使用传统检测各服务器间网络连通状态的方法外, 在集群间网络中未禁用 ICMP 时, 还可以通过查看网络日志来查看近期服务器间通信状态, 日志地址: HBlock 安装目录 /logs/network/network.log。

Q: 是否支持 NAT 访问?

A: 支持。通常, iSCSI initiator 通过内网访问 HBlock 的 Target。如果内网路由器上配置了网络地址转换 (Network Address Translation, NAT), iSCSI initiator 可以通过 NAT 的外网 IP 连接到 Target 所在服务器, 从而通过 HBlock 将 iSCSI 作为云服务进行远程提供。

Q: 为什么系统的数据目录使用率不等于数据目录已用总量/数据目录总额, 数据目录容量配额使用率不等于数据目录已用配额/数据目录容量配额总容量?

A: HBlock 系统、服务器的数据目录使用率含义为所有数据目录使用率的平均值。HBlock 系统、服务器的数据目录容量配额使用率为所有数据目录容量配额使用率的平均值, 详情参见监控指标中的 Path_Rate 和 Path_Cap_Quota_Rate。

6 故障处理

怎么手动卸载 HBlock 服务器？

如果遇到 HBlock 服务器出现故障，不能自动卸载，可以通过手动卸载的方法将 HBlock 进行卸载，步骤如下：

1. 使用命令 **ps -aux|grep stor** 查询 HBlock 进程。
2. 根据查到的进程号，使用命令 **kill Stor_进程号** 删除 Stor 进程。
3. 使用命令 **rm -rf Stor_安装目录**，删除 Stor 相关安装文件。
4. 使用命令 **rm -rf Stor_数据目录**，删除 Stor 数据。

客户端通过 NFS 挂载 HBlock 创建的卷，读写数据时，出现文件校验错误，如何处理？

出现文件校验错误，可能与 NFS 挂载参数或客户端实现有关，请尝试刷新 NFS 客户端缓存，然后重新读取数据。

7 附录

7.1 HBlock 服务

服务	服务名称	作用
stor:mdm	元数据管理服务	管理整个系统的元数据。
stor:fc	故障转移控制服务	进行系统健康检测，实现故障转移控制。
stor:ls	日志服务	提供基于日志的数据同步功能。
stor:ds	数据服务	管理用户的文件数据块。
stor:cs	协调服务	监视各服务器的状态，触发通知事件，确保集群服务高可用。
stor:ms	管理服务	处理请求信息，维护集群运行状态。
stor:ws	监控服务	监控各个服务的状态，并负责服务的启动。
stor:ps	协议解析服务	负责 iSCSI 协议解析与数据存储。
stor:ag	数据采集服务	负责采集性能数据。
stor:ua	升级监听服务	负责接收升级请求，执行升级相关操作。

7.2 用户事件列表

● 服务器

事件	描述
AddServer	添加服务器
RemoveServer	移除服务器
SetServer	设置服务器属性
DeleteTargetPortalIP	删除服务器 Target 门户 IP
RestartService	重启服务
AddPath	添加数据目录
RemovePath	移除数据目录
StartService	启动服务
StopService	停止服务
SetPath	修改数据目录
MigrateService	迁移服务

● iSCSI Target

事件	描述
CreateTarget	创建 Target
DeleteTarget	删除 Target
SetTarget	设置 Target 属性
MigrateTarget	迁移 Target
DeleteCHAP	删除 CHAP
DeleteConnection	删除连接

● 卷

事件	描述
CreateLUN	创建卷
DeleteLUN	删除卷
SetLUN	设置卷属性
ExpandLUN	扩容卷
SwitchLUN	卷主备切换
RecoverLUN	还原卷
ResumeLUNRecovery	继续还原卷
SetBatchLUN	批量卷操作
CreateCloneLUN	创建克隆卷
FlattenCloneLUN	断开关系链
LUNDataResidue	卷数据残留

● 系统

事件	描述
Login	登录
SetMailConfig	设置邮件通知
DeleteMailConfig	删除邮件通知
SendTestMail	发送测试邮件
SetRemoteAccess	设置远程协助
DeleteRemoteAccess	删除远程协助
ImportLicense	导入软件许可证
SetPassword	设置密码

StartLogCollect	发起日志收集请求
DeleteLogCollect	删除日志收集请求
SetAlarmMuteStatus	修改告警静默状态
ManuallyResolveAlarm	手动解除告警
Setup	初始化
AddMonitorConfig	新增监控配置
SetMonitorConfig	修改监控配置
DeleteMonitorConfig	删除监控配置
AddCtyunCMSConfig	新增智维平台告警推送配置
SetCtyunCMSConfig	修改智维平台告警推送配置
DeleteCtyunCMSConfig	删除智维平台告警推送配置

● 集群拓扑

事件	描述
SetNode	修改节点信息
CreateNode	创建节点
DeleteNode	删除节点

● 存储池

事件	描述
CreateStoragePool	创建存储池
DeleteStoragePool	删除存储池
SetStoragePool	修改存储池信息
AddNodeToPool	添加节点入池

事件	描述
RemoveNodeFromPool	移除池中节点

- 升级

事件	描述
StartUpgrade	开始升级

- 快照

事件	描述
CreateSnapshot	创建快照
SetSnapshot	修改快照
DeleteSnapshot	删除快照
RollbackSnapshot	回滚快照
CreateConsistencySnapshot	创建一致性快照
SetConsistencySnapshot	修改一致性快照
RollbackConsistencySnapshot	回滚一致性快照
DeleteConsistencySnapshot	删除一致性快照

7.3 系统事件列表

● 服务器（Server）

事件	描述
ServiceUnavailable	服务不可用
ServiceAvailable	服务可用
ServerAdded	服务器添加
ServerRemoved	服务器移除
ProtocolServiceAbnormal	协议解析服务异常
ProtocolServiceResumed	协议解析服务恢复
InsufficientSpaceonInstallationPath	安装目录剩余空间不足
SpaceonInstallationPath	安装目录剩余空间满足服务运行要求
BaseServiceAbnormal	基础服务异常
BaseServiceResumed	基础服务恢复
ServiceMigrated	服务迁移完成
ServiceMigrateAbnormal	服务迁移异常
InsufficientSpaceonMetaDir	基础服务数据目录剩余空间不足
SufficientSpaceonMetaDir	基础服务数据目录剩余空间满足服务运行要求

● 数据目录（Disk）

事件	描述
DiskIOError	磁盘 IO 错误
DiskIOResumed	磁盘 IO 恢复
DiskWriteSpeedTooSlow	磁盘写入速度慢
DiskWriteSpeedResumed	磁盘写入速度恢复正常
PathAdded	数据目录添加
PathRemoved	数据目录移除

CapacityQuotaUsageExceedsThreshold	配额使用率超阈值
CapacityQuotaUsageBelowThreshold	配额使用率恢复正常
CapacityQuotaUsageApproachLimit	配额用尽
CapacityQuotaUsageBelowLimit	配额使用率低于上限
DiskUsageExceedsThreshold	磁盘使用率超阈值
DiskUsageBelowThreshold	磁盘使用率恢复正常
DiskPathHealthStatusWarning	数据目录健康状态警告
DiskPathHealthStatusError	数据目录健康状态错误
DiskPathHealthStatusResumed	数据目录健康状态恢复
DataServiceHealthStatusWarning	数据服务健康状态警告
DataServiceHealthStatusError	数据服务健康状态错误
DataServiceHealthStatusResumed	数据服务健康状态恢复

● 卷（LUN）

事件	描述
ActiveStandbySwitched	卷主备切换
InsufficientFDForLUNToWrite	可用故障域数量不满足卷写入要求
SufficientFDForLUNToWrite	可用故障域数量满足卷写入要求
LUNRecovered	卷还原
CannotConnectToCloud	无法连接到云
ConnectWithCloudResumed	上云连接已恢复
CloudAccountAbnormal	云账户异常
CloudAccountNormal	云账户正常
LUNCloudDataConflict	卷云端数据冲突
LUNCloudDataConflictResolved	卷云端数据冲突解除
LUNCloudHeartbeatConflict	卷云端心跳冲突

LUNCloudHeartbeatNormal	卷云端心跳正常
LUNFlattened	断开关系链完成

● 目标（Target）

事件	描述
InitiatorConnectionFailed	客户端连接断开
InitiatorConnectionNormal	客户端连接恢复

● 系统（System）

事件	描述
ReachLicenseMaxCapacity	许可证容量达到上限
LicenseMaintenanceExpired	许可证过保
LicenseExpired	许可证过期
LicenseImported	许可证导入
DataResumed	数据恢复
DataLowRedundancy	数据降级
DataBalanceStart	数据均衡开始
DataBalanceProgress	数据均衡过程
DataBalanceFailed	数据均衡失败
DataBalanceEnd	数据均衡结束
DataAccessFailed	数据无法访问

● 故障域（FaultDomain）

事件	描述
FaultDomainWarning	故障域状态变为警告
FaultDomainError	故障域状态变为错误
FaultDomainResumed	故障域状态恢复正常

● 存储池（StoragePool）

事件	描述
CapacityQuotaUsageExceedsThreshold	配额使用率超阈值
CapacityQuotaUsageBelowThreshold	配额使用率恢复正常
CapacityQuotaUsageApproachLimit	配额即将用尽
CapacityQuotaUsageBelowLimit	配额使用率低于上限
DiskUsageExceedsThreshold	磁盘使用率超阈值
DiskUsageBelowThreshold	磁盘使用率恢复正常

7.4 监控指标

● 数据粒度

监控指标的粒度可以分为“精细”和“粗糙”，具体含义如下：

精细类别	数据粒度	数据保留时长	说明
精细	20 秒	2 小时	每 20s 采集一次实时数据，生成 1 个数据点，每个数据点保留 2 小时。
	1 分钟	6 小时	基于 20s 粒度采集数据聚合 1 分钟粒度数据，保留时长 6 小时。
	5 分钟	1 天	基于 20s 粒度采集数据聚合 5 分钟粒度数据，保留时长 1 天。
	1 小时	7 天	基于 20s 粒度采集数据聚合 1 小时粒度数据，保留时长 7 天。
	1 天	1 年	基于 20s 粒度采集数据聚合 1 天粒度数据，保留时长 1 年。
粗糙	5 分钟	2 小时	基于 20s 粒度采集数据聚合 5 分钟粒度数据，保留时长 2 小时。
	1 小时	1 天	基于 20s 粒度采集数据聚合 1 小时粒度数据，保留时长 1 天。
	1 天	1 个月	基于 20s 粒度采集数据聚合 1 天粒度数据，保留时长 1 个月。
	1 周	6 个月	基于 20s 粒度采集数据聚合 1 周粒度数据，保留时长 6 个月。
	1 个月	1 年	基于 20s 粒度采集数据聚合 1 个月粒度数据，保留时长 1 年。

● 监控指标

监控对象 (dimension)	监控指标 (metric)	说明	单位	数据粒度
system	IOPS	客户端与 HBlock 之间的总 IOPS。	无	精细
	R_IOPS	客户端从 HBlock 读取数据的 IOPS。	无	精细
	W_IOPS	客户端向 HBlock 写入数据的 IOPS。	无	精细
	Bandwidth	客户端与 HBlock 之间的总带宽。	字节/s	精细
	R_Bandwidth	客户端从 HBlock 读取数据的带宽。	字节/s	精细

	W_Bandwidth	客户端向 HBlock 写入数据的带宽。	字节/s	精细
	Latency	客户端与 HBlock 之间的总时延。系统在一个采集周期内，读写操作平均时延，反映 HBlock 处理读写请求的时长。	ms	精细
	W_Latency	客户端向 HBlock 写入数据的时延，系统在一个采集周期内，写操作平均时延，反映 HBlock 处理写请求的时长。	ms	精细
	R_Latency	客户端从 HBlock 读取数据的时延，集群在一个采集周期内，读操作平均时延，反映 HBlock 处理读请求的时长。	ms	精细
	Path_Cap	数据目录总容量。	字节	粗糙
	Path_Used	数据目录已用容量。	字节	粗糙
	Path_Rate	数据目录平均使用率，即所有数据目录使用率的平均值。	%	粗糙
	Path_Cap_Quota	HBlock 可用空间大小，即用户给 HBlock 分配的所有目录容量配额的总和。	字节	粗糙
	Path_Cap_Quota_Used	磁盘文件系统中，HBlock 数据占用的空间大小。	字节	粗糙
	Path_Cap_Quota_Rate	数据目录容量配额使用率，即所有数据目录 Path_Cap_Quota_Used/Path_Cap_Quota 的平均值。	%	粗糙
	Cloud_Bandwidth	HBlock 与云之间的总带宽。	字节/s	精细
	Cloud_U_Bandwidth	HBlock 向云上传数据的带宽。	字节/s	精细
	Cloud_D_Bandwidth	HBlock 从云下载数据的带宽。	字节/s	精细
server	CPU_Rate	服务器 CPU 使用率。	%	精细
	Mem_Rate	服务器内存使用率。	%	精细
	Mem_Total	服务器内存总量。	字节	精细
	Mem_Used	服务器内存使用量。	字节	精细
	IOPS	客户端与 HBlock 之间的总 IOPS。	无	精细
	R_IOPS	客户端从 HBlock 读取数据的 IOPS。	无	精细
	W_IOPS	客户端向 HBlock 写入数据的 IOPS。	无	精细
	Bandwidth	客户端与 HBlock 之间的总带宽。	字节/s	精细
	R_Bandwidth	客户端从 HBlock 读取数据的带宽。	字节/s	精细

	W_Bandwidth	客户端向 HBlock 写入数据的带宽。	字节/s	精细
	Latency	客户端与 HBlock 之间的总时延。采集周期内，服务器关联卷的读写时延平均值。	ms	精细
	W_Latency	客户端到 HBlock 写时延，采集周期内，服务器关联卷的写时延平均值。	ms	精细
	R_Latency	客户端从 HBlock 读取数据的时延，采集周期内，服务器关联卷的平均读时延。	ms	精细
	Path_Cap	服务器的数据目录总容量。	字节	粗糙
	Path_Used	服务器的数据目录已用容量。	字节	粗糙
	Path_Rate	服务器的数据目录平均使用率，即服务器上所有数据目录使用率的平均值。	%	粗糙
	Path_Cap_Quota	HBlock 可用空间大小，即用户给 HBlock 分配的所有目录容量配额的总和。	字节	粗糙
	Path_Cap_Quota_Used	磁盘文件系统中，HBlock 数据占用的空间大小。	字节	粗糙
	Path_Cap_Quota_Rate	数据目录容量配额使用率，即服务器上所有数据目录 Path_Cap_Quota_Used/Path_Cap_Quota 的平均值。	%	粗糙
	Cloud_Bandwidth	HBlock 服务器与云之间的总带宽。	字节/s	精细
	Cloud_U_Bandwidth	HBlock 服务器向云上传数据的带宽。	字节/s	精细
	Cloud_D_Bandwidth	HBlock 服务器从云下载数据的带宽。	字节/s	精细
disk	Path_Cap	数据目录总容量。	字节	粗糙
	Path_Used	数据目录已用容量。	字节	粗糙
	Path_Rate	数据目录平均使用率。	%	粗糙
	Path_Cap_Quota	HBlock 可用空间大小，即用户给 HBlock 分配的容量配额。	字节	粗糙
	Path_Cap_Quota_Used	磁盘文件系统中，HBlock 数据占用的空间大小。	字节	粗糙
	Path_Cap_Quota_Rate	数据目录容量配额使用率，即 Path_Cap_Quota_Used/Path_Cap_Quota。	%	粗糙
LUN	IOPS	客户端与 HBlock 卷之间的总 IOPS	无	精细
	R_IOPS	客户端从 HBlock 卷读取数据的 IOPS。	无	精细
	W_IOPS	客户端向 HBlock 卷写入数据的 IOPS。	无	精细

	Bandwidth	客户端与 HBlock 卷的之间的总带宽。	字节/s	精细
	R_Bandwidth	客户端从 HBlock 卷读取数据的带宽。	字节/s	精细
	W_Bandwidth	客户端向 HBlock 卷写入数据的带宽。	字节/s	精细
	Latency	客户端与 HBlock 卷之间的总时延，卷在一个采集周期内，读写操作平均时延，反映 HBlock 单卷处理读写请求的时长。	ms	精细
	W_Latency	客户端向 HBlock 卷写入数据的时延，卷在一个采集周期内，写操作平均时延，反映 HBlock 单卷处理写请求的时长。	ms	精细
	R_Latency	客户端从 HBlock 卷读取数据的时延，卷在一个采集周期内，读操作平均时延，反映 HBlock 单卷处理读请求的时长。	ms	精细
	Cloud_Bandwidth	HBlock 卷与云之间的总带宽。	字节/s	精细
	Cloud_U_Bandwidth	HBlock 卷向云上传数据的带宽。	字节/s	精细
	Cloud_D_Bandwidth	HBlock 卷从云下载数据的带宽。	字节/s	精细
	Wait_Upload	待从 HBlock 卷上传云的数据量。	字节	精细
pool (仅集群版支持)	IOPS	客户端与 HBlock 之间的总 IOPS。	无	精细
	R_IOPS	客户端从 HBlock 读取数据的 IOPS。	无	精细
	W_IOPS	客户端向 HBlock 写入数据的 IOPS。	无	精细
	Bandwidth	客户端与 HBlock 之间的总带宽。	字节/s	精细
	R_Bandwidth	客户端从 HBlock 读取数据的带宽。	字节/s	精细
	W_Bandwidth	客户端向 HBlock 写入数据的带宽。	字节/s	精细
	Latency	客户端与 HBlock 之间的总时延。系统在一个采集周期内，读写操作平均时延，反映 HBlock 处理读写请求的时长。	ms	精细
	W_Latency	客户端向 HBlock 写入数据的时延，系统在一个采集周期内，写操作平均时延，反映 HBlock 处理写请求的时长。	ms	精细
	R_Latency	客户端从 HBlock 读取数据的时延，集群在一个采集周期内，读操作平均时延，反映 HBlock 处理读请求的时长。	ms	精细
	Path_Cap	数据目录总容量，即存储池中所有数据目录所在磁盘分区的文件系统的总容量。	字节	粗糙

	Path_Used	数据目录已用容量，即存储池中所有数据目录所在磁盘分区的文件的已用容量	字节	粗糙
	Path_Rate	数据目录平均使用率，即存储池所有数据目录使用率的平均值。	%	粗糙
	Path_Cap_Quota	HBlock 可用空间大小，即存储池中用户给 HBlock 分配的所有目录容量配额的和。	字节	粗糙
	Path_Cap_Quota_Used	存储池中所有已加入 HBlock 的数据目录中的 HBlock 数据大小。	字节	粗糙
	Path_Cap_Quota_Rate	数据目录容量配额平均使用率，存储池中所有数据目录 $\text{Path_Cap_Quota_Used}/\text{Path_Cap_Quota}$ 的平均值。	%	粗糙

7.5 告警列表

告警规则名称	告警级别	告警条件	自动解除条件	告警失效条件	是否允许手动解除	告警邮件发送频率
数据目录读写错误 PathIOError	重要	数据目录状态为坏盘	数据目录状态为正常	数据目录被移除，数据目录从存储池中移除，或数据目录所在服务器被移除	是	每天 1 次
数据目录所在磁盘写入速度慢 DiskWriteSlow	警告	数据目录所在磁盘写入速度慢	数据目录所在盘恢复正常	数据目录被移除，数据目录从存储池中移除，或数据目录所在服务器被移除	是	每天 1 次
许可证即将到期 LicenseWillExpire	警告	当前时间（告警模块所在服务器的系统时间）距离最后导入的许可证的到期时间 ≤ 15 天且 > 0 天	当前时间（告警模块所在服务器的系统时间）距离告警许可证的到期时间 > 15 天	许可证过期，或导入新的许可证（不同 id 的许可证）	是	每天 1 次
许可证过期 LicenseExpired	严重	当前时间（告警模块所在服务器的系统时间）距离最后导入的许可证的到期时间 ≤ 0 天	当前时间（告警模块所在服务器的系统时间）距离告警许可证的到期时间 > 0 天	导入新的许可证（不同 id 的许可证）	是	发送 1 次
许可证维保即将到期 LicenseMaintenanceWillExpire	警告	当前时间（告警模块所在服务器的系统时间）距离最后导入的许可证的维保到期时间 ≤ 15 天且 > 0 天	当前时间（告警模块所在服务器的系统时间）距离告警许可证的维保到期时间 > 15 天	许可证过期，或导入新的许可证（不同 id 的许可证）	是	每天 1 次
许可证过保 LicenseMaintenanceExpired	警告	当前时间（告警模块所在服务器的系统时间）距离最后导入的许可证的维保到期时间 ≤ 0 天	当前时间（告警模块所在服务器的系统时间）距离告警许可证的维保到期时间 > 0 天	导入新的许可证	是	发送 1 次
试用期即将到期 TrialVersionWillExpire	警告	当前未导入生效的许可证，并且当前时间（告警模块所在服务器的系统时间）距离试用期过期时间 ≤ 15 天且 ≥ 0 天	无解除条件，只能手动解除	导入新的许可证	是	每天 1 次

资源用量接近使用上限 ResourceUsageApproachingLimit	重要	本地卷总容量>=许可证容量的 80%	本地卷总容量<许可证容量的 75%	导入新的许可证	是	发送 1 次
告警中的告警条数接近上限 AlarmNumberApproachingLimit	严重	告警中的告警条数>=8000	告警条数<7500	无	是	每天 1 次
告警邮件发送失败 FailToSendAlarmEmail	严重	告警邮件发送失败	告警邮件发送成功	邮件配置被删除, 或邮件发送设置为 disable	是	每天 1 次
配额使用率超阈值 CapacityQuotaUsageExceedsThreshold	警告	存储池中数据目录关联磁盘的 Path_Cap_Quota_Rate>=80%, 数据目录层级的数据目录关联磁盘的 Path_Cap_Quota_Rate>=80% 说明: 数据目录未设置容量配额, 则按容量配额=磁盘总容量计算。	存储池中数据目录关联磁盘的 Path_Cap_Quota_Rate<75%, 或数据目录层级的数据目录关联磁盘的 Path_Cap_Quota_Rate<75% 说明: 数据目录未设置容量配额, 则按容量配额=磁盘总容量计算。	- 存储池名称变更 - 存储池中的数据目录被全部移除	是	每天 1 次
配额用尽 CapacityQuotaUsageApproachLimit	严重	基础存储池中数据目录对应磁盘总配额使用率>=95%	基础存储池中数据目录对应磁盘总配额使用率<90%	基础存储池名称变更	是	每天 1 次
配额用尽 CapacityQuotaUsageApproachLimit	警告	非基础存储池中数据目录对应磁盘总配额使用率>=95%, 或数据目录对应磁盘配额使用率>=95%	非基础存储池中数据目录对应磁盘总配额<90%, 或数据目录对应磁盘配额使用率<90%	- 存储池名称变更 - 存储池中的数据目录被全部移除	是	每天 1 次
磁盘使用率超阈值 DiskUsageExceedsThreshold	警告	存储池中数据目录对应磁盘的 Path_Rate>=80%, 或数据目录对应磁盘的 Path_Rate>=80%	存储池中数据目录对应磁盘的 Path_Rate<75%, 或数据目录对应磁盘的 Path_Rate<75%	- 存储池名称变更 - 存储池中的数据目录被全部移除	是	每天 1 次
可用故障域数量不满足卷写入要求 InsufficientFDForLUNToWrite	警告	卷所在缓存存储池或存储池的可用故障域数量及健康数据目录数量不满足卷的最小副本数要求	告警存储池的可用故障域数量及健康数据目录数量满足卷的最小副本数要求	- 卷删除 - 存储池名称变更 - 卷禁用 - 卷删除失败	是	每天 1 次
数据目录健康状态变为警告 DiskPathHealthStatusWarning	警告	数据目录健康状态变为警告	数据目录健康状态恢复正常	- 数据目录被移除 - 数据目录从存	是	每天 1 次

				储池中移除 ● 数据目录健康状态变为“Error”		
数据目录健康状态变为错误 DiskPathHealthStatusError	重要	数据目录健康状态变为错误	数据目录健康状态恢复正常	● 数据目录被移除。 ● 数据目录从存储池中移除	是	每天 1 次
数据服务健康状态变为警告 DataServiceHealthStatusWarning	警告	数据服务健康状态变为警告	数据服务健康状态恢复正常	● 服务器被移除 ● 数据目录被移除 ● 数据目录从存储池中移除 ● 数据目录健康状态变为“Error” ● 存储池名称变更	是	每天 1 次
数据服务健康状态变为错误 DataServiceHealthStatusError	重要	数据服务健康状态变为错误	数据服务健康状态恢复正常	● 服务器被移除 ● 数据目录被移除 ● 数据目录从存储池中移除 ● 存储池名称变更	是	每天 1 次
协议解析服务异常 ProtocolServiceAbnormal	重要	协议解析服务异常	协议解析服务恢复正常	● 服务器被移除 ● Target 被删除 ● Target 被迁移	是	每天 1 次
故障域状态变为警告 FaultDomainWarning	警告	故障域状态变为警告	故障域状态恢复正常	● 服务器被移除 ● 数据目录被移除 ● 数据目录从存储池中移除 ● 故障域健康状态变为“Error” ● 存储池名称变更或故障域全路径名称中任一节点名称变更	是	每天 1 次

				故障域中的数据目录被全部移除		
故障域状态变为错误 FaultDomainError	重要	故障域状态变为 Error	故障域状态变为非 Error	- 服务器被移除 - 数据目录被移除 - 数据目录从存储池中移除 - 存储池名称变更或故障域全路径名称中任一节点名称变更 - 故障域中的数据目录被全部移除	是	每天 1 次
无法连接到云 CannotConnectToCloud	严重	与云端连接断开超过 10 分钟	卷通过此服务器向云端读取或写入成功一次	- 卷被删除或服务器被移除，卷禁用，卷删除失败 - Target 发生迁移 - 还原中的卷产生的此告警，在卷还原完成后，告警失效	是	每天 1 次
云账户异常 CloudAccountAbnormal	严重	云端读取或写入失败：欠费冻结、欠费冻结或违规冻结	云端读取或写入成功一次	卷被删除、卷禁用或卷删除失败	是	每天 1 次
卷云端数据冲突 LUNCloudDataConflict	严重	卷对应的云端数据出现比本地更新的版本	卷对应的云端数据版本全部比本地旧	卷被删除、卷禁用或卷删除失败	是	每天 1 次
卷云端心跳冲突 LUNCloudHeartbeatConflict	严重	卷的云端出现来自非本集群的心跳	卷的云端心跳仅来自本集群	卷被删除、卷禁用或卷删除失败	是	每天 1 次
卷数据残留 LUNDataResidue	警告	强制删除卷，卷数据残留： - 本地数据残留：由于本地磁盘故障，导致卷数据无法同步删除 - 云端数据残留：	无法自动解除，只能手动解除告警	- 本地数据残留：数据目录从机器中移除 - 云端数据残留：不会自动解除，清理完云端残留数据	是	每天 1 次

		删除云上数据时，由于云端数据存储空间无法访问（包括网络连接失败、账号异常等），导致数据无法同步删除		后，可手动解除告警		
客户端连接断开 InitiatorConnectionFailed	警告	因为客户端的原因，HBlock 无法收到客户端心跳，导致 HBlock 认为和客户端连接断开，立即告警。但是客户端主动断开连接的情况除外。	该客户端与 Target 连接成功。	- 告警的 IQN 所在 Target 被删除 - Target 被迁移	是	每天 1 次
安装目录剩余空间不足 InsufficientSpaceonInstallationPath	严重	安装目录所在磁盘的文件系统剩余空间 $\leq 4\text{GiB}$	安装目录所在磁盘的文件系统剩余空间 $> 5\text{GiB}$	服务器被移除	是	每天 1 次
基础服务异常 BaseServiceAbnormal	严重	发生以下任一情况时，分别发出相应告警实例的告警： - 元数据管理服务（mdm）异常：集群中 stor:mdm 服务有 2 个，仅允许一个故障，故障时发出告警 - 故障转移控制服务（fc）异常：集群中 stor:fc 服务有 2 个，仅允许一个故障，故障时发出告警 - 日志服务（ls）异常：集群中 stor:ls 服务有 3 个，仅允许一个故障，故障时发出告警 - 协调服务（cs）	匹配的告警实例恢复到以下程度： - 元数据管理服务（mdm）在告警机器上恢复正常：告警机器上的 stor:mdm 服务恢复正常 - 故障转移控制服务（fc）在告警机器上恢复正常：告警机器上的 stor:fc 服务恢复正常 - 日志服务（ls）在告警机器上恢复正常：告警机器上的 stor:ls 服务恢复正常 - 协调服务（cs）在告警机器上恢复正常：告警机	告警机器上的基础服务迁移完成	是	每天 1 次

		异常：集群中 stor:cs 服务有 3 个，仅允许一个故障，故障时发出告警	器上的 stor:cs 服务恢复正常			
基础服务数据目录剩余空间不足 InsufficientSpaceonMetaDir	严重	基础服务数据目录所在磁盘的文件系统剩余空间≤4GiB	基础服务数据目录所在磁盘的文件系统剩余空间>5GiB	服务开始迁移	是	每天 1 次

7.6 OOS Endpoint 和 Region

中国大陆区域对象存储网络中的各个地区，使用统一的 OOS Endpoint: oos-cn.ctyunapi.cn，region 为 cn。

香港节点分为**香港精品网络**和**香港普通网络**两种方式：香港精品网 OOS Endpoint: oos-cn-hk-hqnet.ctyunapi.cn，region 为 cnhk-hqnet；香港普通网 OOS Endpoint: oos-cn-hk-nqnet.ctyunapi.cn，region 为 cnhk-nqnet。

说明：如果您的数据存储在中国大陆区域对象存储网络中的某个资源池，可以直接使用该资源池的 OOS Endpoint。OOS Endpoint 列表如下（Endpoint 列表仅为资源池 Endpoint 访问信息描述，与资源状态无关联）：

地区	OOS Endpoint	region
郑州	oos-hazz.ctyunapi.cn	hazz
沈阳	oos-lnsy.ctyunapi.cn	lnsy
四川成都	oos-sccd.ctyunapi.cn	sccd
乌鲁木齐	oos-xjwlmq.ctyunapi.cn	xjwlmq
甘肃兰州	oos-gslz.ctyunapi.cn	gslz
山东青岛	oos-sdqd.ctyunapi.cn	sdqd
贵州贵阳	oos-gzgy.ctyunapi.cn	gzgy
湖北武汉	oos-hbwh.ctyunapi.cn	hbwh
西藏拉萨	oos-xzls.ctyunapi.cn	xzls
安徽芜湖	oos-ahwh.ctyunapi.cn	ahwh
广东深圳	oos-gdsz.ctyunapi.cn	gdsz
江苏苏州	oos-jssz.ctyunapi.cn	jssz
上海 2	oos-sh2.ctyunapi.cn	sh2

7.7 HBlock 可推送的监控指标

类别	指标名称	含义
server	hblock_cpu_seconds_user	用户态时间
	hblock_cpu_seconds_nice	nice 用户态时间
	hblock_cpu_seconds_system	内核态时间
	hblock_cpu_seconds_idle	空闲时间
	hblock_cpu_seconds_iowait	I/O 等待时间
	hblock_cpu_seconds_irq	硬中断时间
	hblock_cpu_seconds_softirq	软中断时间
	hblock_cpu_seconds_steal	强制等待另外虚拟的 CPU 处理完毕时花费的时间
	hblock_cpu_guest_seconds_user	运行虚拟机所花费的 CPU 时间。当系统在虚拟化环境中运行虚拟机时，这个字段会统计虚拟机所使用的 CPU 时间。
	hblock_cpu_guest_seconds_nice	运行低优先级虚拟机所花费的 CPU 时间。与 guest 字段类似，这个字段统计的是运行低优先级虚拟机所使用的 CPU 时间。
	hblock_memory_MemTotal_bytes	系统中所有可用的内存大小
	hblock_memory_MemFree_bytes	系统尚未使用的内存大小
	hblock_memory_MemAvailable_bytes	真正的系统可用内存大小
	hblock_memory_Buffers_bytes 等操作系统的 meminfo 指标	其余 meminfo 指标。可在开启数据推送后，在 prometheus 查看具体指标名称
	hblock_memory_oom_kill	操作系统发生 oom killer 的数量
load	hblock_load1	最近 1 分钟的平均负载情况
	hblock_load5	最近 5 分钟的平均负载情况
	hblock_load15	最近 15 分钟的平均负载情况

interface	hblock_network_receive_bytes	端口接收的总字节数
	hblock_network_transmit_bytes	端口发送的总字节数
	hblock_network_receive_packets	端口接收的数据包数量
	hblock_network_transmit_packets	端口发送的数据包数量
	hblock_network_receive_errs	端口接收过程中发生的错误数据包数量
	hblock_network_transmit_errs	端口发送过程中发生的错误数据包数量
	hblock_network_receive_drop	端口接收过程中被丢弃的数据包数量
	hblock_network_transmit_drop	端口发送过程中被丢弃的数据包数量
	hblock_network_bandwidth	端口理论带宽
	hblock_network_status	端口状态。端口状态值含义如下： ● 0：表示状态为 down ● 1：表示状态为 up ● 2：表示状态为 unknown ● 3：表示状态为 notpresent ● 4：表示状态为 lowerlayerdown ● 5：表示状态为 testing ● 6：表示状态为 dormant ● -2：表示未识别的状态
tcp	hblock_network_up_count	端口 up 次数
	hblock_network_down_count	端口 down 次数
	hblock_netstat_tcp_RetransSegs	TCP 重传的报文数量
	hblock_netstat_tcp_OutSegs	TCP 输出的报文数量
	hblock_netstat_tcp_InSegs	TCP 接收的报文数量
	hblock_netstat_tcp_ActiveOpens	当前 ActiveOpen 状态的 TCP 连接数
	hblock_netstat_tcp_CurrEstab	当前 CurrEstab 状态的 TCP 连接数
	hblock_netstat_tcp_PassiveOpens	当前 PassiveOpens 状态的 TCP 连接数

	hblock_sockstat_tcp_mem	当前 mem 状态的 TCP 连接数
	hblock_sockstat_tcp_alloc	当前 alloc 状态的 TCP 连接数
	hblock_sockstat_tcp_inuse	当前 inuse 状态的 TCP 连接数
	hblock_sockstat_tcp_orphan	当前 orphan 状态的 TCP 连接数
	hblock_sockstat_tcp_tw	当前 tw 状态的 TCP 连接数
disk	hblock_disk_read_bytes	硬盘读请求数据量
	hblock_disk_written_bytes	写请求数据量
	hblock_disk_reads_completed	读请求次数
	hblock_disk_read_time_seconds	读请求的时间
	hblock_disk_writes_completed	写请求次数
	hblock_disk_write_time_seconds	写请求的时间
	hblock_disk_io_time_seconds	处理 I/O 操作的时间
	hblock_disk_io_time_weighted_seconds	处理 I/O 操作的加权时间
	hblock_disk_io_now	当前正在运行的实际 I/O 请求数
fileSystem	hblock_fileSystem_size_bytes	文件系统总容量
	hblock_fileSystem_free_bytes	文件系统剩余容量
	hblock_fileSystem_free_inode_count	空闲的 inode 数量
	hblock_fileSystem_total_inode_count	总的 inode 数量
	hblock_fileSystem_readonly	文件系统是否为只读
OS	hblock_os_boot_time_seconds	服务器最近一次启动时间
Cloud	hblock_cloud_wait_upload_bytes	待上传云的数据
	hblock_cloud_upload_size_bytes	已上传云的数据
	hblock_cloud_download_size_bytes	从云上已下载的数据
LUN	hblock_blockDevice_read_count	IO 读次数（包含读成功、读失败）
	hblock_blockDevice_read_fail_count	IO 读失败次数
	hblock_blockDevice_read_time_seconds	IO 读累积耗时（包含读成功、读失败）

	hblock_blockDevice_read_fail_time_seconds	IO 读失败累积耗时
	hblock_blockDevice_read_bytes	IO 读成功数据量大小
	hblock_blockDevice_write_count	IO 写次数（包含写成功、写失败）
	hblock_blockDevice_write_fail_count	IO 写失败次数
	hblock_blockDevice_write_time_seconds	IO 写累积耗时（包含写成功、写失败）
	hblock_blockDevice_write_fail_time_seconds	IO 写失败累积耗时
	hblock_blockDevice_write_bytes	IO 写成功数据量大小
diskpath	hblock_dataDir_total_bytes	数据目录对应磁盘的总容量
	hblock_dataDir_avail_bytes	数据目录对应磁盘的可用容量
	hblock_dataDir_quota_total_bytes	数据目录的配额空间
	hblock_dataDir_quota_used_bytes	数据目录的配额已用空间

7.8 集群拓扑文件

集群拓扑向用户展示集群内所有节点的拓扑结构，方便用户查看节点名称、节点状态、节点组成等信息。用户可以自己构造集群拓扑文件，在初始化的时候进行导入。拓扑文件为符合 UTF-8 编码格式的 JSON 文档。

说明：

- 父节点是 root，子节点可以是 room、rack 或者 server。
- 父节点是 room，子节点可以是 rack 或者 server。
- 父节点是 rack，子节点只能是 server。
- 父节点是 server，子节点只能是 path。

拓扑文件的参数说明如下：

参数	描述	是否必须
name	根节点名称。 类型：字符串 取值：字符串形式，长度范围 1~63，只能由字母、数字、句点(.)、下划线(_)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。默认为 default。	否
type	根节点的类型。 类型：字符串 取值：root：根节点。 默认为 root。	否
childNodes	子节点信息集合，根据子节点类型不同，需要的参数不同。 类型：数组	是

子节点类型为 room	name	子节点名称。 类型：字符串 取值：字符串形式，长度范围 1~63，只能由字母、数字、句点(.)、下划线（_）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。	是
	type	子节点类型。 类型：字符串 取值：room：机房类型。	是
	description	子节点描述信息。 类型：字符串 取值：1~50 位字符串。	否
	childNodes	子节点信息集合，根据子节点类型不同，需要的参数不同，子节点可以为 rack 或者 server。 类型：数组	否
子节点类型为 rack	name	子节点名称。 类型：字符串 取值：字符串形式，长度范围 1~63，只能由字母、数字、句点(.)、下划线（_）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。	是
	type	子节点类型。 类型：字符串 取值：rack：机架类型。	是
	description	子节点描述信息。 类型：字符串	否

		取值：1~50 位字符串。	
	childNodes	子节点信息集合，子节点只能为 server。 类型：数组	否
子节点类型为 server	name	子节点名称。 类型：字符串 取值：字符串形式，长度范围 1~63，只能由字母、数字、句点(.)、下划线(_)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。默认为服务器 ID。	否
	type	子节点类型。 类型：字符串 取值：server：服务器类型。	是
	description	子节点描述信息。 类型：字符串 取值：1~50 位字符串。	否
	ip	HBlock 的服务器 IP。 取值：IPv4 或[IPv6]地址。	是
	apiPort	管理 API 端口。 类型：整型 取值：[1, 65535]，默认值为 1443。需要和该服务器安装 HBlock 时设置的 API 端口号保持一致。	否
	childNodes	子节点信息集合。子节点的类型为 path。	是
子节点类型为 path	name	子节点名称。 类型：字符串 取值：数据目录具体路径。	是

	type	子节点类型。 类型：字符串 取值：path：数据目录类型。	是
	capacityQuota	指定数据目录的容量配额，即针对此数据目录，HBlock 可写入的数据总量。当 HBlock 的使用空间一旦达到配额，就立刻阻止数据写入，不允许再使用超出配额的空间。 类型：长整型 取值：小于数据目录所在磁盘的总容量，单位是字节。负整数表示无限制写入，0 表示禁止写入。默认无限制写入。 注意： 如果相同的数据目录出现多次，以第一次出现的数据目录的容量配额为准。	否

拓扑文件示例：

```
{
  "name": "default",
  "childNodes": [
    {
      "name": "room1",
      "type": "room",
      "childNodes": [
        {
          "type": "server",
          "name": "server1",
          "ip": "192.168.0.1",
          "apiPort": 1443,
          "childNodes": [
            {
              "name": "/mnt/storage01",
```

```
        "type": "path",
        "capacityQuota": 96636764160
    },
    {
        "name": "/mnt/storage02",
        "type": "path"
    }
]
},
{
    "type": "server",
    "name": "server2",
    "ip": "192.168.0.2",
    "apiPort": 1443,
    "childNodes": [
        {
            "name": "/mnt/storage01",
            "type": "path",
            "capacityQuota": 96636764160
        }
    ]
},
{
    "type": "server",
    "name": "server3",
    "ip": "192.168.0.3",
    "apiPort": 1443,
    "childNodes": [
        {
            "name": "/mnt/storage01",
            "type": "path"
        },
        {
            "name": "/mnt/storage02",
            "type": "path"
        }
    ]
}
]
```

```
]
}
```