



存储资源盘活系统

用户手册 - 管理 API

天翼云科技有限公司

修订记录

版本	发布日期	描述
4.0	2026 年 03 月 24 日	1. 支持免费版本。 2. 新增调整 HBlock 服务占用服务器内存参数功能。 3. 上云卷新增挂起卷功能。 4. 支持通过 API 设置本地卷的扩展属性。 5. 增加设置鉴权方式。
3.10	2025 年 09 月 25 日	1. 支持 target 访问权限，实现客户端和 target 端权限管理。 2. 支持备份，基于快照生成独立于源卷的数据备份文件。 3. 支持设置 QoS 规则，从带宽和 IOPS 维度管控流量。
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. API 签名方法变更，提升安全性。 2. 存储卷和缓存卷支持将数据上传至兼容 S3 的对象存储。 3. 支持设置基础服务的数据存储目录。 4. target 增加回收策略，支持无卷关联后自动删除。 5. 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。 6. 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 08 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 备份	7
1.3.6 iSCSI target	7
1.3.7 iSCSI 目标门户	8
1.3.8 存储池	8
1.3.9 QoS 策略	8
1.3.10 集群拓扑	8
1.3.11 故障域	9
1.3.12 数据服务	9
1.3.13 监控	9
1.3.14 事件	9
1.3.15 日志采集	10
1.3.16 告警	10
1.4 商业版与免费版	11
1.5 软件许可证到期（试用期或订阅模式）后仍可使用的功能	14
1.6 使用限制	15
1.7 术语与缩略语	16
2 服务器部署要求	17
2.1 环境要求	17
2.2 HBlock 配置环境 - 单机版	19
2.3 HBlock 配置环境 - 集群版	22
3 安装访问	25
3.1 安装	25
3.2 访问地址	27
3.3 接口	28
3.3.1 URL 规则	28
3.3.2 通用请求头	30

3.3.3 签名方法	31
3.3.4 通用响应头	32
3.3.5 通用错误码	33
4 管理 API	38
4.1 初始化	38
4.1.1 初始化	38
4.1.2 查询初始化进度	62
4.1.3 错误码列表	64
4.2 软件许可证	72
4.2.1 加载软件许可证	72
4.2.2 批量查询软件许可证信息	78
4.2.3 查询指定软件许可证信息	84
4.2.4 错误码列表	90
4.3 卷管理	92
4.3.1 创建卷	92
4.3.2 创建克隆卷	119
4.3.3 断开克隆卷与快照的关系链	139
4.3.4 删除卷	141
4.3.5 扩容卷	144
4.3.6 修改单个卷	146
4.3.7 设置卷的主备优先级或自动切换（集群版适用）	154
4.3.8 触发卷对应 target 的主备切换（集群版适用）	159
4.3.9 还原卷（上云卷适用）	162
4.3.10 挂起卷（上云卷适用）	181
4.3.11 恢复还原中断或挂起的卷（上云卷适用）	183
4.3.12 清空卷（本地卷适用）	185
4.3.13 设置卷的扩展属性（本地卷适用）	188
4.3.14 删除卷的指定扩展属性（本地卷适用）	191
4.3.15 删除卷的所有扩展属性（本地卷适用）	194
4.3.16 查询卷的扩展属性（本地卷适用）	195
4.3.17 批量查询卷	199
4.3.18 查询指定卷	225
4.3.19 查询卷关联的 QoS	250
4.3.20 错误码列表	256
4.4 快照管理	264
4.4.1 创建快照	264
4.4.2 修改快照	268

4.4.3 回滚快照.....	271
4.4.4 删除快照.....	273
4.4.5 批量查询快照.....	275
4.4.6 查询指定快照.....	279
4.4.7 错误码列表.....	284
4.5 一致性快照管理.....	287
4.5.1 创建一致性快照.....	287
4.5.2 修改一致性快照.....	292
4.5.3 回滚一致性快照.....	295
4.5.4 删除一致性快照.....	298
4.5.5 批量查询一致性快照.....	300
4.5.6 查询指定一致性快照.....	304
4.5.7 错误码列表.....	308
4.6 备份管理（本地卷）.....	312
4.6.1 导出备份.....	312
4.6.2 导入备份.....	317
4.6.3 错误码列表.....	319
4.7 iSCSI target 管理.....	321
4.7.1 创建 iSCSI target.....	321
4.7.2 修改 iSCSI target 的 CHAP 认证.....	338
4.7.3 迁移 iSCSI target（集群版适用）.....	341
4.7.4 修改 iSCSI target 下每个 IQN 允许建立的最大会话数.....	346
4.7.5 修改 iSCSI target 的回收策略.....	348
4.7.6 设置 iSCSI target 的允许访问列表.....	350
4.7.7 删除 iSCSI target 的允许访问列表.....	356
4.7.8 删除 iSCSI target.....	358
4.7.9 删除 CHAP.....	360
4.7.10 批量查询 iSCSI target.....	361
4.7.11 查询指定 iSCSI target.....	375
4.7.12 查询 iSCSI target 连接.....	385
4.7.13 删除 iSCSI target 连接.....	389
4.7.14 错误码列表.....	391
4.8 存储池（集群版）.....	396
4.8.1 创建存储池.....	396
4.8.2 添加节点到存储池.....	403
4.8.3 修改存储池.....	406
4.8.4 移除存储池内的节点.....	408
4.8.5 删除非基础存储池.....	411

4.8.6 批量查询存储池.....	413
4.8.7 查询指定存储池.....	425
4.8.8 查询存储池关联的 QoS	436
4.8.9 错误码列表.....	441
4.9 QoS 策略.....	447
4.9.1 创建 QoS 策略.....	447
4.9.2 修改 QoS 策略.....	453
4.9.3 设置卷关联的 QoS 策略.....	459
4.9.4 解除卷关联的 QoS 策略.....	461
4.9.5 设置存储池关联的 QoS 策略（集群版）.....	463
4.9.6 解除存储池关联的 QoS 策略（集群版）.....	465
4.9.7 设置存储池内卷的默认 QoS 策略（集群版）.....	467
4.9.8 解除存储池内卷的默认 QoS 策略（集群版）.....	469
4.9.9 删除 QoS 策略.....	471
4.9.10 批量查询 QoS 策略.....	472
4.9.11 查询指定 QoS 策略.....	478
4.9.12 查询 QoS 策略关联/可关联的对象信息.....	482
4.9.13 错误码列表.....	490
4.10 集群拓扑（集群版）.....	491
4.10.1 创建拓扑节点.....	491
4.10.2 修改拓扑节点信息.....	493
4.10.3 删除拓扑节点.....	495
4.10.4 查询整个拓扑.....	497
4.10.5 查询拓扑节点.....	506
4.10.6 错误码列表.....	514
4.11 服务器管理.....	517
4.11.1 添加服务器（集群版适用）.....	517
4.11.2 修改指定服务器属性.....	522
4.11.3 批量修改服务器属性.....	527
4.11.4 删除服务器属性.....	534
4.11.5 迁移服务器上的基础服务（集群版适用）.....	535
4.11.6 移除服务器（集群版适用）.....	538
4.11.7 重启服务器上的 HBlock 服务.....	543
4.11.8 批量查询服务器.....	546
4.11.9 查询指定服务器.....	574
4.11.10 添加数据目录.....	587
4.11.11 修改数据目录的容量配额.....	590
4.11.12 移除数据目录.....	593
4.11.13 错误码列表.....	596

4.12 查询 HBlock 信息	605
4.13 查询 HBlock 服务状态	612
4.14 监控	620
4.14.1 获取实时监控数据	620
4.14.2 获取历史性能数据	628
4.14.3 错误码列表	638
4.15 告警	639
4.15.1 批量查询 HBlock 告警信息	639
4.15.2 查询指定的 HBlock 告警	654
4.15.3 手动解除 HBlock 告警	659
4.15.4 静默告警/解除告警静默	661
4.15.5 导出告警	664
4.15.6 错误码列表	666
4.16 事件和日志管理	667
4.16.1 查看事件	667
4.16.2 导出事件	672
4.16.3 发起 HBlock 日志采集	675
4.16.4 批量查询采集的日志	680
4.16.5 查询单个采集的日志	686
4.16.6 下载采集的日志文件	689
4.16.7 删除指定的采集日志	691
4.16.8 删除所有的采集日志	692
4.16.9 错误码列表	693
4.17 HBlock 系统设置	695
4.17.1 修改管理员密码	695
4.17.2 邮件设置	697
4.17.3 远程协助	708
4.17.4 Pushgateway 监控配置	716
4.17.5 智维平台推送告警配置	741
4.17.6 鉴权方式	751
4.17.7 开启高级功能试用（免费版）	756
4.18 调整 HBlock 性能参数（集群版适用）	759
4.18.1 调整 HBlock 性能参数	759
4.18.2 查看性能调优配置	761
4.18.3 错误码	763
4.19 查询 HBlock 版本	764
4.20 升级 HBlock	766

4. 20. 1 升级 HBlock	766
4. 20. 2 查询升级进度	770
4. 20. 3 错误码	773
5 附录	777
5. 1 转义字符表	777
5. 2 HBlock 服务	778
5. 3 用户事件列表	779
5. 4 系统事件列表	784
5. 5 监控指标	788
5. 6 告警列表	795
5. 7 OOS Endpoint 和 Region	803
5. 7. 1 对象存储网络	803
5. 7. 2 对象存储网络 2	803
5. 7. 3 香港节点	804
5. 7. 4 其他地域	804
5. 8 HBlock 可推送的操作系统监控指标	805
5. 9 集群拓扑文件	810

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中，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。
- **应用系统存储高可用：**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。
- **关键数据永久独立备份：**HBlock支持通过快照技术生成全量和增量备份，确保备份数据长期保存且与生产系统完全隔离。用户可将备份文件灵活导出至各类存储介质（如对象存储、NAS等），实现数据与平台的彻底解耦。即使主系统完全瘫痪或遭受勒索软件攻击，这些独立保存的备份仍可作为可信恢复源，精准还原卷数据，确保关键业务数据零丢失。该方案尤其适用于金融、医疗等对数据长期归档和合规审计有严格要求的行业，为其提供可靠的数据保护保障。
- **零信任存储防护体系：**HBlock通过iSCSI target白名单、CHAP认证与QoS流控三重防护，打造端到端的零信任存储访问体系。系统基于卷级粒度严格校验客户端IQN身份，确保未授权主机完全不可见；采用CHAP认证进行连接握手，有效防御伪造身份和中间人攻击；结合QoS策略，以IOPS/带宽双维度动态调控，实现异常流量实时阻断。最终达成存储性能零干扰、数据与流量安全隔离、关键业务持续稳定的三重保障。
- **上云数据差额还原：**HBlock通过上云卷的挂起/恢复机制，实现跨集群容灾切换与数据流动管理，支持跨集群数据高效同步与快速恢复，构建了安全高效的云边数据协同体系。基于上云卷状态独占控制机制，通过挂起能力暂停卷和云端的交互。此后通过云端数据还原功能支持将云上数据在其他集群进行恢复，并继续进行数据读写，直至再次挂起。在挂起卷进行恢复的时候，通过增量差异智能比对和差量数据定向回传技术，仅同步云端增量数据即可完成卷恢复。确保同一业务卷在任意时刻仅归属于单一集群并保持可读写状态，从根源避免多地写入引发的数据冲突风险。HBlock在业务连续性保障、跨集群数据安全以及容灾切换流程自动化方面具备核心价值，为混合云与多活架构提供安全可靠、弹性灵活、可视可控的存储层容灾能力。

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 备份

备份记录特定时间点的数据状态，并生成可独立存储的备份文件，支持独立存放，按需导入。可以根据需要进行全量导出和增量导出，支持导入完整的备份文件、导入中断产生的备份文件。

可以应用下列场景：

- 关键数据备份：完整数据副本存储于独立介质，支持增量/全量策略，长期保留历史版本数据。
- 跨集群迁移/灾难恢复：将备份文件还原到其他存储池和其他集群的卷中。
- 镜像分发与部署：将预配置的系统镜像导入到多个卷中实现快速部署，在无网络连接的环境中也可通过导入文件部署系统。

1.3.6 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。

1.3.7 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。

1.3.8 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。

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。

1.3.9 QoS 策略

QoS（Quality of Service，服务质量）策略通过令牌桶算法精确调控 IOPS 与吞吐量，在拥塞出现前就完成流量整形，从源头避免网络拥塞。

该策略可以根据实际需求，应用下列场景：

- 关键业务保障：通过限制周边业务的 I/O，间接保障核心业务服务质量。
- 存储分层优化：在不同性能层(如 SSD 和 HDD)之间合理分配 I/O 资源，优化混合存储环境中的性能表现。
- 突发流量处理：允许短时间超出基准限制，平滑处理突发 I/O 而不直接拒绝请求。

1.3.10 集群拓扑

拓扑：对集群物理资源实际部署方式的可视化逻辑展示。

1.3.11 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。

1.3.12 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。

1.3.13 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**。

1.3.14 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**。

1.3.15 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。

1.3.16 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 商业版与免费版

初始化时，可以根据需要选择商业版或者免费版，商业版和免费版支持的功能如下：

功能		商业版	免费版
软件许可证	加载软件许可证、查看软件许可证	支持	支持
卷	创建卷、扩容卷、修改卷、设置卷主备优先级或自动切换、触发卷对应 target 的主备切换、删除卷、设置卷的扩展属性、删除卷的指定扩展属性、删除卷所有的扩展属性、查询卷的扩展属性、查询卷、查询卷关联的 QoS 策略	支持	支持
	克隆卷、断开克隆与快照的关系链、还原卷、挂起卷、恢复还原中断或挂起的卷、清空卷	支持	不支持
快照	创建快照、修改快照、回滚快照、删除快照、查询快照	支持	不支持
一致性快照	创建一致性快照、修改一致性快照、回滚一致性快照、删除一致性快照、查询一致性快照	支持	不支持
备份	导出备份、导入备份	支持	不支持
iSCSI target	创建 iSCSI target、删除 iSCSI target、设置 iSCSI target 的 CHAP 认证、删除 CHAP、迁移 iSCSI target、修改 iSCSI target 下每个 IQN 允许建立的最大会话数、修改 iSCSI target 的回收策略、设置 iSCSI target 的允许访问列表、删除 iSCSI target 的允许访问列表、查询 iSCSI target、查询 iSCSI target 连接、删除 iSCSI target 连接	支持	支持
存储池	创建存储池、添加节点到存储池、修改存储池、移除存储池内的节点、删除非基础存储池、查询存储池、查询存储池关联的 QoS 策略	支持	支持
QoS 策略	创建 QoS 策略、修改 QoS 策略、设置卷关联的 QoS	支持	不支持

	策略、解除卷关联的 QoS 策略、设置存储池关联的 QoS 策略、解除存储池关联的 QoS 策略、设置存储池内卷的默认 QoS 策略、解除存储池内卷的默认 QoS 策略、删除 QoS 策略、查询 QoS 策略、查询 QoS 策略关联/可关联的对象信息		
集群拓扑	创建拓扑节点、修改拓扑节点信息、删除拓扑节点、查询拓扑信息	支持	支持
服务器	添加服务器、修改服务器属性（包括：修改服务器端口范围、设置服务器目标门户 IP、设置服务器中 HBlock 可使用的内存、设置服务器默认数据目录）、删除服务器属性（服务器的 targetPortalIP 配置）、移除服务器、查询服务器、添加数据目录、修改数据目录的容量配额、移除数据目录	支持	支持
	迁移服务器上的基础服务	支持	不支持
监控	查询实时性能数据、导出性能数据（命令行）、查看历史性能数据（WEB/API）	支持	支持
告警	查看告警信息、导出告警、手动解除告警、静默告警、解除告警静默	支持	支持
事件	查看 HBlock 事件、导出 HBlock 事件	支持	支持
日志	发起 HBlock 日志采集、查看 HBlock 采集的日志、下载采集的 HBlock 日志文件、删除 HBlock 采集的日志	支持	支持
管理员密码	修改管理员密码	支持	支持
邮件设置	设置邮件通知功能、发送测试邮件、删除邮件配置、查询邮件配置	支持	支持
远程协助	设置远程协助、删除远程协助配置、查询远程协助配置	支持	支持
Pushgateway	添加 Pushgateway 监控配置、修改 Pushgateway 监控	支持	不支持

监控配置	配置、删除 Pushgateway 监控配置、查询 Pushgateway 监控配置		
智维推送告警配置	添加向智维推送告警的配置、修改向智维推送告警的配置、删除向智维推送告警的配置、查询向智维推送告警的配置	支持	不支持
鉴权方式	设置鉴权方式、查询鉴权方式	支持	支持
高级试用功能	开启高级试用功能	不支持	支持
调整 HBlock 性能参数	调整 HBlock 性能参数、查看性能调优配置	支持	支持
服务管理	停止服务器上的 HBlock 服务、启动服务器上的 HBlock 服务、重启服务器上的 HBlock 服务	支持	支持
卸载 HBlock	卸载 HBlock	支持	支持
系统查询	查看 HBlock 信息、查看 HBlock 服务状态、查看 HBlock 版本	支持	支持
升级 HBlock	查看升级状态	支持	支持
	升级 HBlock	支持	2 年内支持

1.5 软件许可证到期（试用期或订阅模式）后仍可用的功能

软件许可证到期（试用期或订阅模式）后仍可用的功能，详见下表：

模块	功能
帮助命令	帮助命令（仅命令行）
软件许可证	加载软件许可证、查看软件许可证
卷	删除卷
服务器	查询服务器
服务管理	停止服务器上的 HBlock 服务、启动服务器上的 HBlock 服务、重启服务器上的 HBlock 服务
告警	查看告警信息
事件	查看 HBlock 事件、导出 HBlock 事件
日志	发起 HBlock 日志采集、查看 HBlock 采集的日志、删除 HBlock 采集的日志
管理员密码	修改管理员密码
远程协助	设置远程协助、删除远程协助配置、查询远程协助配置
系统查询	查看 HBlock 信息、查看 HBlock 服务状态、查看 HBlock 版本
卸载 HBlock	卸载 HBlock
其他	WEB 支持概览页面（许可证、数据目录、故障域、卷）

1.6 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。

1.7 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	Erasur Coding，纠删码。
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。
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。
数据目录	用于存储 HBlock 数据的目录。

2 服务器部署要求

2.1 环境要求

项目	描述
支持 Linux OS	CentOS 7/8/9 64 位，CTyunOS 2/4 64 位。
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要）。

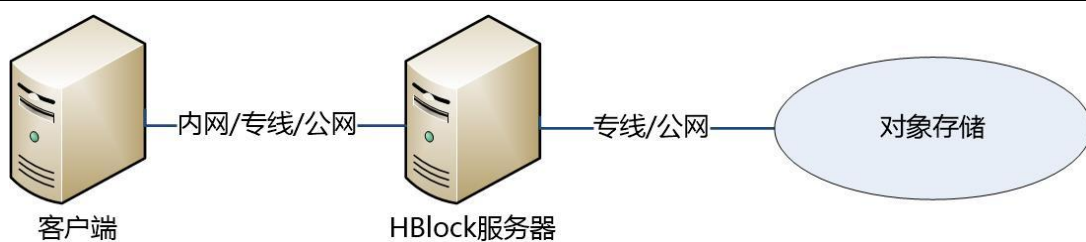


图1. 单机版网络拓扑图

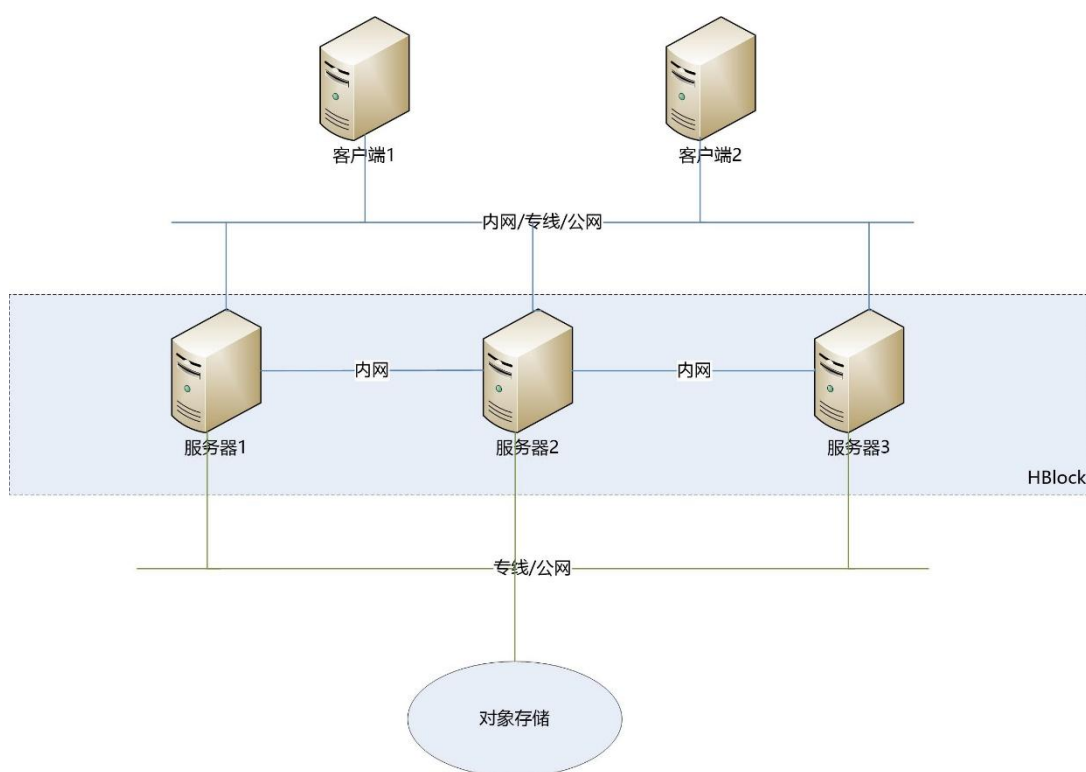


图2. HBlock 集群版网络拓扑图

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 配置环境 – 单机版

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，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
mkdir DIRECTORY #创建挂载路径，DIRECTORY 为路径名
mount /dev/vdX DIRECTORY #挂载硬盘
```

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。

如果防火墙是 `firewall`，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
```

如果防火墙是 `iptables`，示例如下：

1. 开启 iSCSI 端口，如 iSCSI 端口为 3260 时：

```
iptables -I INPUT -p tcp --dport 3260 -j ACCEPT
```

2. 保存配置。

```
iptables-save
```

(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
```

可以根据情况，将 `domain` 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 nofile 65536    # root 为参数 domain 的取值
root hard nofile 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 HBlock 配置环境 – 集群版

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（可选）

(二)格式化硬盘并挂载（可选）

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
mkdir DIRECTORY #创建挂载路径，DIRECTORY 为路径名
mount /dev/vdX DIRECTORY #挂载硬盘
```

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。

如果防火墙是 firewall，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
```

如果防火墙是 iptables，示例如下：

1. 开启 iSCSI 端口，如 iSCSI 端口为 3260 时：

```
iptables -I INPUT -p tcp --dport 3260 -j ACCEPT
```

2. 集群各服务器间相互访问：

- 允许本地环路地址

```
iptables -I INPUT -i lo -j ACCEPT
```

- 允许内部网段访问

```
iptables -I INPUT -s your_IP -j ACCEPT
```

3. 保存配置。

```
iptables-save
```

(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
```

可以根据情况，将 `domain` 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
```

3 安装访问

3.1 安装

1. 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。对于集群版，每台服务器准备的目录可以不同。建议数据目录不要与操作系统共用磁盘或文件系统。

2. 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_4.0.0_x64.zip
cd CTYUN_HBlock_Plus_4.0.0_x64
```

示例：在服务器上执行解压缩安装包，并进入解压缩后的文件夹。

```
[root@hblockserver opt]# unzip CTYUN_HBlock_Plus_4.0.0_x64.zip
.....
[root@hblockserver opt]# cd CTYUN_HBlock_Plus_4.0.0_x64/
```

3. 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。

说明：您可以根据业务需要设置 API 端口号和 WEB 端口号。

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。

示例：在服务器上安装 HBlock。

```
[root@hblockserver1 CTYUN_HBlock_Plus_4.0.0_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. 安装完成：安装完成后，可以使用 WEB、命令行或者 API 对 HBlock 进行初始化及管理操作。本文后续将介绍使用 API 对 HBlock 进行初始化及管理操作。

3.2 访问地址

管理 API 服务启动后，可以通过 `https://SERVER_IP:PORT` 地址进行访问。仅支持 HTTPS 协议，`SERVER_IP` 可以是已安装 HBlock 的任意一台服务器 IP 地址（IPv4 或者 IPv6）。

参数	描述
<code>SERVER_IP</code>	安装 HBlock 的服务器 IP。 确保该服务器能够被您访问到，只有 API 客户端和被访问服务器在同一局域网内， <code>SERVER_IP</code> 才能使用服务器的内网 IP。
<code>PORT</code>	安装时配置的 API 端口，如果未配置，默认为 1443。

3.3 接口

3.3.1 URL 规则

采用 HTTP Restful API，URL 规则为：

```
https://${ip}:${port}/rest/${version}/${prefix}/${resourceName}/${name}[?(${parameterName}=${value})+]
```

参数描述

参数	描述	是否必须
ip	服务器 IP。	是
port	管理 API 的端口号。	是
version	版本号。 取值： ● v1。 ● v2：目前仅下列接口支持 v2，同时兼容 v1：批量查询卷、查询指定卷、查询 HBlock 服务状态。对于这些接口，建议使用 v2。	是
prefix	前缀，取值： ● system：系统相关接口。 ● block：块操作相关接口。	是
resourceName	资源名称。	是
name	具体实例的名称或 ID。	否
parameter	具体参数。	否

3.3.1.1 filter 条件

批量查询时，可用使用 filter 过滤条件查询。

过滤条件由查询属性和属性值组成，其中属性和属性值之间用单个冒号分隔（**key:value**）表示模糊匹配，用两个冒号分隔（**key::value**）表示精确匹配。可以指定多个过滤条件，如果是或的关系，使用“or”将查询条件分隔开；如果是与的关系，使用“and”将查询条件分隔开。

例如：

查询卷的过滤条件：**filter=lunName::lunb or lunName:lun01 and status:Normal**。

表示查询卷名称等于 **lunb** 或卷名称包含 **lun01**，且状态为 **Normal** 的所有卷。

3.3.2 通用请求头

请求头	描述	是否必须
Host	请求的 IP 和端口号。	是
Connection	是否需要持久连接，取值： ● keep-alive: 持续连接。 ● close: 关闭连接。 默认值为keep-alive。	否
User-Agent	请求的客户端信息。	否
Date	请求发送的日期和时间，格式：EEE, dd MMM yyyy HH:mm:ss 'GMT'。	是
Content-Type	请求体MIME类型，取值为：application/json; charset=utf-8。	否
Content-Length	请求体的长度。	是
Content-MD5	按照RFC 1864，使用base64编码格式生成信息的128位MD5值。此请求头可以用作数据完整性检查，以验证数据是否与客户端发送的数据相同。	否
Authorization	包含请求的签名认证信息，格式是： HBLOCK+" "+userName+": "+signature。	是

3.3.3 签名方法

Authorization 请求头中，计算签名的方法为：

Authorization: *HBlock*+" "+*userName*+": "+*signature*，其中：

- *HBlock*：固定字符串，取值为 HBlock。
- *userName*：HBlock 的管理员用户名。

注意：使用 API 初始化时，需要使用默认用户名 **storuser** 进行签名，初始化时可以修改用户名。

- $signature = \text{HMAC-SHA256}(\text{SHA256}(password), Date + "\n" + \text{SHA256}(password))$
password 为用户密码；*Date* 为请求头 *Date* 的值，GMT 格式。

注意：使用 API 初始化时，需要使用初始密码进行签名，初始密码为 123456。初始化时必须重新设置密码，后续使用其他接口时使用新设置的密码进行签名。

3.3.4 通用响应头

响应头	描述
Server	服务名。
Connection	表示是否需要持久连接： ● keep-alive: 持续连接。 ● close: 关闭连接。
Date	响应时间。
Content-Type	响应体的MIME类型。
Content-Length	响应体的长度。
x-hblock-request-id	唯一标示请求的一个32位字符串，用于排查定位问题。
Retry-After	当服务端backoff时，返回此响应头，值为90。建议客户端90秒后重试。 backoff 说明 ：当系统服务繁忙，暂不可用时，会返回客户端503响应状态码和Retry-After响应头。

3.3.5 通用错误码

HTTP status	错误码	错误信息	说明
400	BadDigest	The Content-MD5 you specified does not match what HBlock received.	Content-MD5 的值和请求体的 MD5 不一致。
400	ClusterModeNotAllowed	'operation' is not supported by cluster mode of HBlock.	集群版本 HBlock 不支持该操作。
400	IncompleteBody	You did not provide the number of bytes specified by the Content-Length HTTP header.	请求体长度不足 Content-Length 请求头指定的长度。
400	InsufficientServerSpace	This method is not allowed because the free space of <i>directory</i> must be greater than or equal to <i>value</i> for server <i>serverId</i> [, <i>serverId</i> ...].	服务器存储空间不足，无法执行操作。
400	InvalidArgument	Value empty at ' <i>argument</i> ' failed to satisfy constraint: Argument must not be empty.	Json 数组长度不能为 0。
400	InvalidBoolean	Value <i>value</i> at ' <i>argument</i> ' failed to satisfy constraint: Argument must be boolean.	布尔类型不合法。
400	InvalidFilter	Value <i>value</i> at ' <i>filter</i> ' failed to satisfy constraint: Argument must use a single colon (':' means fuzzy match) or two colons '::' means	filter 参数不合法。

		exact match) to separate the attribute key and value. If you query multiple attributes, you can use 'and' or 'or' keywords to separate them, and there must be spaces before and after 'and' or 'or'.	
400	InvalidHeader	Invalid HTTP header, <i>headerName</i> .	请求头格式不合法。
400	InvalidInteger	Value <i>value</i> at ' <i>argument</i> ' failed to satisfy constraint: Argument must be integer.	值必须为整数。
400	InvalidArgumentLength	Value <i>value</i> at ' <i>argument</i> ' failed to satisfy constraint: Argument must not exceed <i>value</i> characters.	参数长度错误。
400	InvalidPackage	Invalid package. Please get the correct package.	安装包无效。
400	InvalidRequest	Unable to parse request <i>url</i> .	URL 解析失败。
400	InvalidRequestBody	Invalid request body.	请求无效。
400	InvalidStatus	Value <i>value</i> at ' <i>argument</i> ' failed to satisfy constraint: Argument must satisfy enum value set: [Enabled, Disabled].	输入的参数值不对。
400	InvalidString	Value " at ' <i>argument</i> ' failed to satisfy constraint: Argument must not be an empty string.	参数不能是空字符串。
400	LoadConfigFailed	Failed to load configuration files.	加载配置文件失败。

			加载配置文件需要满足以下条件： ● 需要基础服务器至少有 2 台满足 cs 和 ls 服务成功启动。 ● 需要基础服务器至少有 1 台满足 fc、mdm 和 ds 服务成功启动。 ● 需要保证待启动的服务器和主服务器（Master）网络通信正常。
400	MissingArgument	Value null at ' <i>argument</i> ' failed to satisfy constraint: Argument must not be null.	Json 参数不能为空。
400	MissingHeader	Missing HTTP header <i>headerName</i> .	请求头 Host 不能为空。
400	NotInitialized	HBlock is not initialized.	HBlock 未初始化。
400	NotInstalled	HBlock is not installed.	HBlock 未安装。
400	RequestTimeout	No response received. Please check if there are any network issues and if the operation was successfully executed.	未收到响应，请检查是否存在网络问题，以及操作是否执行成功。
400	StandaloneModeNotAllowed	' <i>operation</i> ' is not supported by standalone mode of HBlock.	单机版不支持该操作。

403	LicenseExpired	The software license has expired.	软件许可证过期。
403	PermissionDenied	Failed to <i>action</i> because user has no permission.	用户的权限不足导致操作失败。
403	SignatureDoesNotMatch	Signature does not match. Please check your user name, password and signing method for server(s) [, <i>serverIP</i> ...].	签名不正确。请检查用户名、密码和签名方法。
400	SystemDataDegraded	There is low redundancy or error data in the system, operation is not allowed. Please repair fault domains and disk paths.	系统中存在降级或错误数据，不允许操作。请修复故障域和磁盘路径。
403	TrialExpired	The software trial version has expired.	软件试用期已过期。
405	InvalidMethod	The specified method does not exist.	指定的方法不存在。
409	InvalidLUNStatus	The status of LUN ' <i>lunname</i> ' is ' <i>status</i> ', the request is invalid.	LUN当前的状态不正确，请求无效。
409	InvalidServerStatus	The server <i>serverId</i> status is ' <i>status</i> ', the request is invalid.	服务器当前的状态不正确，请求无效。
409	InvalidStorStatus	The HBlock status is ' <i>status</i> ', the request is invalid.	HBlock当前的状态不正确，请求无效。
409	InvalidtargetStatus	The target status is ' <i>status</i> ', the request is invalid.	target当前的状态不正确，请求无效。
500	InsufficientResource	Operation failed due to insufficient resources.	内存或其他资源不足，操作失败。
500	InternalError	HBlock encountered an internal error. Please try again or contact	请重试或者联系软件供应商。

		the software vendor.	
503	SlowDown	Please reduce your request rate.	服务端繁忙，请降低 请求频率。

4 管理 API

4.1 初始化

4.1.1 初始化

此操作用来初始化 HBlock，如果服务端返回 202 响应码，则表示开始执行初始化操作。

说明：安装完所有服务器后，需要初始化 HBlock。

注意：

- 使用 API 初始化时，需要使用默认用户名和初始密码进行签名，默认用户名为 **storuser**，初始密码为 123456。初始化时必须重新设置密码，后续使用其他接口时使用新设置的密码进行签名。
- 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（portRange）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。

请求语法

- 单机版

```
POST /rest/v1/system/setup HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "storName": storName,
  "userName": userName,
```



```
"newPassword": newPassword,
"edition": edition,
"iSCSIPort": iSCSIPort,
"portRange": port1-port2,
"ports":{
  "managementPort1": managementPort1,
  "managementPort2": managementPort2,
  "managementPort3": managementPort3,
  "managementPort4": managementPort4,
  "managementPort6": managementPort6
},
"servers":[
  {
    "ip": ip,
    "apiPort": apiPort,
    "diskPaths": [
      {
        "path": path,
        "capacityQuota": capacityvalue
      },
      {
        "path": path,
        "capacityQuota": capacityvalue
      },
      .....
    ]
  }
],
"publicNetwork": publicNetwork
}
```

- 集群版（服务器 IP 导入）：

POST /rest/v1/system/setup HTTP/1.1

Date: *date*

Content-Type: application/json; charset=utf-8

```
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "storName": storName,
  "userName": userName,
  "newPassword": newPassword,
  "edition": edition,
  "iSCSIPort": iSCSIPort,
  "portRange": port1-port2,
  "faultDomain": "faultDomain",
  "ports": {
    "dataPort1": dataPort1,
    "managementPort1": managementPort1,
    "managementPort2": managementPort2,
    "managementPort3": managementPort3,
    "managementPort4": managementPort4,
    "managementPort5": managementPort5,
    "managementPort6": managementPort6,
    "metadataPort1": metadataPort1,
    "metadataPort2": metadataPort2,
    "metadataPort3": metadataPort3,
    "metadataPort4": metadataPort4,
    "metadataPort5": metadataPort5,
    "metadataPort6": metadataPort6,
    "metadataPort7": metadataPort7,
    "metadataPort8": metadataPort8
  },
  "servers": [
    {
      "nodeName": servername,
      "ip": ip,
      "apiPort": apiPort,
      "diskPaths": [
```

```
        {
            "path": path,
            "capacityQuota": capacityvalue
        },
        {
            "path": path,
            "capacityQuota": capacityvalue
        },
        .....
    ]
},
.....
{
    "nodeName": servername,
    "ip": ip,
    "apiPort": apiPort,
    "diskPaths": [
        {
            "path": path,
            "capacityQuota": capacityvalue
        },
        {
            "path": path,
            "capacityQuota": capacityvalue
        },
        .....
    ]
}
],
"services": {
    "mdm": [
        {
            "ip": ip,
            "metaDir": path
        },
    ],
}
```

```
{
  "ip": ip,
  "metaDir": path
},
"ls": [
  {
    "ip": ip,
    "metaDir": path
  },
  {
    "ip": ip,
    "metaDir": path
  },
  {
    "ip": ip,
    "metaDir": path
  }
],
"cs": [
  {
    "ip": ip,
    "metaDir": path
  },
  {
    "ip": ip,
    "metaDir": path
  },
  {
    "ip": ip,
    "metaDir": path
  }
]
},
"clusterNetwork": clusterNetwork,
```

```
"publicNetwork": publicNetwork
}
```

● 集群版（导入拓扑文件）

```
POST /rest/v1/system/setup HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "storName": storName,
  "userName": userName,
  "newPassword": newPassword,
  "edition": edition,
  "iSCSIPort": iSCSIPort,
  "portRange": port1-port2,
  "faultDomain": faultDomain,
  "topology": topologyString,
  "ports": {
    "dataPort1": dataPort1,
    "managementPort1": managementPort1,
    "managementPort2": managementPort2,
    "managementPort3": managementPort3,
    "managementPort4": managementPort4,
    "managementPort5": managementPort5,
    "managementPort6": managementPort6,
    "metadataPort1": metadataPort1,
    "metadataPort2": metadataPort2,
    "metadataPort3": metadataPort3,
    "metadataPort4": metadataPort4,
    "metadataPort5": metadataPort5,
    "metadataPort6": metadataPort6,
    "metadataPort7": metadataPort7,
```

```
"metadataPort8": metadataPort8
},
"services": {
  "mdm": [
    {
      "ip": ip,
      "metaDir": path
    },
    {
      "ip": ip,
      "metaDir": path
    }
  ],
  "ls": [
    {
      "ip": ip,
      "metaDir": path
    },
    {
      "ip": ip,
      "metaDir": path
    },
    {
      "ip": ip,
      "metaDir": path
    }
  ],
  "cs": [
    {
      "ip": ip,
      "metaDir": path
    },
    {
      "ip": ip,
      "metaDir": path
    }
  ]
}
```

```

    },
    {
      "ip": ip,
      "metaDir": path
    }
  ]
},
"clusterNetwork": clusterNetwork,
"publicNetwork": publicNetwork
}

```

请求参数

参数	类型	描述	是否必须
storName	String	HBlock 名称。 取值：长度范围是 1~64，可以包含字母、数字、下划线（_）和短横线（-），字母区分大小写，且仅支持以字母或数字开头。	是
userName	String	HBlock 的管理员用户名。 取值：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 storuser。	否
newPassword	String	设置的新密码。初始化时必须修改密码。 取值：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）。	是
edition	String	指定版本类型。	否

		取值： ● Free: 免费版。 ● Commercial: 商业版。 默认值为 Commercial 。	
iSCSIPort	Integer	iSCSI 端口号。 取值: [1, 65535], 默认值为 3260。	否
portRange	String	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: [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。	否
faultDomain	String	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： ● room: 机房级别。 ● rack: 机架级别。 ● path: 数据目录级别。 ● server: 服务器级别。 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。	否
ports.dataPort1	Integer	数据端口 1（仅集群版支持）。 取值: [1, 65535]。	否
ports.managementPort1	Integer	管理服务端口 1。 取值: [1, 65535]。	否

ports.managementPort2	Integer	管理服务端口 2。 取值：[1, 65535]。	否
ports.managementPort3	Integer	管理服务端口 3。 取值：[1, 65535]。	否
ports.managementPort4	Integer	管理服务端口 4。 取值：[1, 65535]。	否
ports.managementPort5	Integer	管理服务端口 5（仅集群版支持）。 取值：[1, 65535]。	否
ports.managementPort6	Integer	管理服务端口 6。 取值：[1, 65535]。	否
ports.metadataPort1	Integer	元数据端口 1（仅集群版支持）。 取值：[1, 65535]。	否
ports.metadataPort2	Integer	元数据端口 2（仅集群版支持）。 取值：[1, 65535]。	否
ports.metadataPort3	Integer	元数据端口 3（仅集群版支持）。 取值：[1, 65535]。	否
ports.metadataPort4	Integer	元数据端口 4（仅集群版支持）。 取值：[1, 65535]。	否
ports.metadataPort5	Integer	元数据端口 5（仅集群版支持）。 取值：[1, 65535]。	否
ports.metadataPort6	Integer	元数据端口 6（仅集群版支持）。 取值：[1, 65535]。	否
ports.metadataPort7	Integer	元数据端口 7（仅集群版支持）。 取值：[1, 65535]。	否
ports.metadataPort8	Integer	元数据端口 8（仅集群版支持）。 取值：[1, 65535]。	否
topology	String	导入集群拓扑文件内容（仅集群版支持）。	否

		取值：拓扑文件需要采用 UTF-8 编码，且为 JSON 文件格式，然后转换成字符串格式复制给该参数。拓扑文件的构成详见集群拓扑文件。 注意：“topology”与“servers”必须选一种，且只能选一种。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。	
servers	Array of server	HBlock 服务器的属性集合，详见“表 1 请求参数 server 说明（仅集群版支持）”。 注意：“topology”与“servers”必须选一种，且只能选一种。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。	否
services	Array of service	服务地址的集合（仅集群版支持），详见“表 3 请求参数 service 说明（仅集群版支持）”。	否
clusterNetwork	String	集群网络（仅集群版支持），用于集群间的数据通信。 ● 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 0:0:0:0:0:0:1。 取值：如果指定地址范围，IP 地址后加掩码表示，如 192.163.1.5/24。	否
publicNetwork	String	业务网络，用于客户端和 HBlock 之间的数据传输式。	否

		● 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 0:0:0:0:0:0:1。 取值：如果指定地址范围，IP 地址后加掩码表示，如 192.163.1.5/24 列表。	

表 1 请求参数 servers 说明

参数	类型	描述	是否必须
nodeName	String	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。	否
ip	String	HBlock 的服务器 IP。 取值：IPv4 或[IPv6]地址。	是
apiPort	Integer	管理 API 端口号。 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
diskPaths	Array of diskPath	数据目录属性集合，详见“表 2 请求参数 diskPath 说明”。	否

表 2 请求参数 diskPath 说明

参数	类型	描述	是否必须
path	String	指定数据目录。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。 取值：只能包含字母、数字、汉字和特殊字符(~!@\$()_+-;.:)。 说明：对于单机版，第一个数据目录为默认数据目录。对于集群版，需要至少有一个数据目录。 注意：如果故障域是 server 和 path 级别，此项必填。	否
capacityQuota	Long	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：小于数据目录所在磁盘的总容量，单位是 bytes。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

表 3 请求参数 services 说明（仅集群版支持）

参数	类型	描述	是否必须
mdm	Array of mdm	元数据管理服务（仅集群版支持），详见“表 4 请求参数 mdm 说明（仅集群版支持）”。	否
ls	Array of ls	日志服务（仅集群版支持），详见“表 5 请求参数 ls 说明（仅集群版支持）”。	否
cs	Array of cs	协调服务（仅集群版支持），详见“表 6 请求参数 cs 说明”。	否

		(仅集群版支持) ”。	

表 4 请求参数 mdm 说明（仅集群版支持）

参数	类型	描述	是否必须
ip	String	元数据管理服务的地址（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。 取值：IPv4 或[IPv6]地址。	否
metaDir	String	元数据服务的数据目录（仅集群版支持），用于存储元数据服务的相关数据信息。 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分开。 取值： 只能包含字母、数字、汉字和特殊字符(~!@\$()_+ - ; . :)。 默认目录为安装目录。	否

表 5 请求参数 ls 说明（仅集群版支持）

参数	类型	描述	是否必须
ip	String	日志服务的地址（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。 取值：IPv4 或[IPv6]地址。	否
metaDir	String	日志服务的数据目录（仅集群版支持），用于存储日志服务的相关数据信息。 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分开。 取值： 只能包含字母、数字、汉字和特殊字符(~!@\$()_+ - ; . :)。	否

		默认目录为安装目录。	

表 6 请求参数 cs 说明（仅集群版支持）

参数	类型	描述	是否必须
ip	String	协调服务的地址（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。 取值：IPv4 或[IPv6]地址。	否
metaDir	String	协调服务的数据目录（仅集群版支持），用于存储协调服务的相关数据信息。 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分开。 取值：数据目录名不能含逗号。默认目录为安装目录。	否

请求示例 1

单机-商业版：初始化。

```
POST /rest/v1/system/setup HTTP/1.1
Date: Mon, 11 Dec 2023 06:06:47 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 543
Host: 192.168.0.32:1443
Authorization: HBlock storuser:signature

{
  "storName": "stor1",
  "newPassword": "*****",
  "portRange": "20000-20500",
  "iSCSIPort": 3260,
  "servers": [
    {
```

```
{
  "ip": "192.168.0.32",
  "apiPort": 1443,
  "diskPaths": [
    {
      "path": "/mnt/storage01",
      "capacityQuota": 96636764160
    },
    {
      "path": "/mnt/dskb",
      "capacityQuota": 96636764160
    }
  ]
}
```

响应示例 1

```
HTTP/1.1 202 Accepted
x-hblock-request-id: 6589a0abcbde4bc5a4db3a6b9315008f
Connection: keep-alive
Date: Mon, 11 Dec 2023 06:06:47 GMT
Content-Length: 0
Server: HBlock
```

请求示例 2

单机-免费版：初始化。

```
POST /rest/v1/system/setup HTTP/1.1
Date: Mon, 12 Jan 2026 09:45:13 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 431
Host: 192.168.0.66:1443
Authorization: HBlock storuser:signature

{
```

```
"storName": "stor1",
"newPassword": "*****",
"edition": "Free",
"portRange": "20000-20500",
"iSCSIPort": 3260,
"servers": [
  {
    "ip": "192.168.0.66",
    "apiPort": 1443,
    "diskPaths": [
      {
        "path": "/mnt/stor01",
        "capacityQuota": 16636764160
      }
    ]
  }
]
```

响应示例 2

```
HTTP/1.1 202 Accepted
x-hblock-request-id: 2bcc2a5cae4843f5bd08168128d63e7f
Connection: keep-alive
Date: Mon, 12 Jan 2026 09:45:13 GMT
Content-Length: 0
Server: HBlock
```

请求示例 3

集群-商业版：输入服务器 IP，初始化。

```
POST /rest/v1/system/setup HTTP/1.1
Date: Mon, 12 Jan 2026 09:36:02 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 1653
Host: 192.168.0.65:1443
```


Authorization: HBlock storuser:*signature*

```
{
  "storName": "stor1",
  "UserName": "storuser",
  "newPassword": "*****",
  "iSCSIPort": 3260,
  "portRange": "20000-20500",
  "faultDomain": "path",
  "servers": [
    {
      "ip": "192.168.0.64",
      "apiPort": 1443,
      "diskPaths": [
        {
          "path": "/mnt/stor02",
          "capacityQuota": 96636764160
        }
      ]
    },
    {
      "ip": "192.168.0.65",
      "apiPort": 1443,
      "diskPaths": [
        {
          "path": "/mnt/stor01",
          "capacityQuota": 96636764160
        }
      ]
    },
    {
      "ip": "192.168.0.67",
      "apiPort": 1443,
      "diskPaths": [
        {
```

```
        "path": "/mnt/stor01",
        "capacityQuota": 96636764160
      }
    ]
  },
],
"services": {
  "mdm": [
    {
      "ip": "192.168.0.64"
    },
    {
      "ip": "192.168.0.65"
    }
  ],
  "ls": [
    {
      "ip": "192.168.0.64"
    },
    {
      "ip": "192.168.0.65"
    },
    {
      "ip": "192.168.0.67"
    }
  ],
  "cs": [
    {
      "ip": "192.168.0.64"
    },
    {
      "ip": "192.168.0.65"
    },
    {
      "ip": "192.168.0.67"
    }
  ]
}
```

```
    }  
  ]  
}  
}
```

响应示例 3

```
HTTP/1.1 202 Accepted  
x-hblock-request-id: 900956191b34450eb1bc799bd69dd096  
Connection: keep-alive  
Date: Mon, 12 Jan 2026 09:36:02 GMT  
Content-Length: 0  
Server: HBlock
```

请求示例 4

集群-商业版：导入拓扑文件进行初始化。

```
POST /rest/v1/system/setup HTTP/1.1  
Date: Wed, 3 Jul 2024 09:47:16 GMT  
Content-Type: application/json; charset=utf-8  
Content-Length: 1951  
Host: 192.168.0.110:1443  
Authorization: HBlock storuser:signature  
  
{  
  "storName": "stor1",  
  "UserName": "storuser",  
  "newPassword": "*****",  
  "iSCSIPort": 3260,  
  "portRange": "20000-20500",  
  "faultDomain": "room",  
  "topology": "{\r\n    \"name\": \"default\", \r\n    \"childNodes\": [\r\n\r\n      {\r\n        \"name\": \"room1\", \r\n        \"type\": \"room\", \r\n        \"childNodes\": [\r\n          {\r\n            \"name\": \"server1\", \r\n            \"ip\": \"192.168.0.192\", \r\n            \"apiPort\": 1443, \r\n            \"childNodes\": [\r\n\r\n
```



```
        "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"
      }
    ]
  }
]
}
```

响应示例 4

HTTP/1.1 202 Accepted

```
x-hblock-request-id: 068e0a8e16884bf28104b0e84fdb50e
Connection: keep-alive
Date: Wed, 3 Jul 2024 09:47:16 GMT
Content-Length: 0
Server: HBlock
```

请求示例 5

集群-免费版：输入服务器 IP，初始化。

```
POST /rest/v1/system/setup HTTP/1.1
Date: Mon, 12 Jan 2026 09:19:36 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 1014
Host: 192.168.0.65:1443
Authorization: HBlock storuser:signature

{
  "storName": "stor1",
  "UserName": "storuser",
  "newPassword": "*****",
  "iSCSIPort": 3260,
  "portRange": "20000-20500",
  "edition": "Free",
  "faultDomain": "path",
  "servers": [
    {
      "ip": "192.168.0.64",
      "apiPort": 1443,
      "diskPaths": [
        {
          "path": "/mnt/stor02",
          "capacityQuota": 96636764160
        }
      ]
    }
  ],
}
```

```
{
  "ip": "192.168.0.65",
  "apiPort": 1443,
  "diskPaths": [
    {
      "path": "/mnt/stor01",
      "capacityQuota": 96636764160
    }
  ],
},
{
  "ip": "192.168.0.67",
  "apiPort": 1443,
  "diskPaths": [
    {
      "path": "/mnt/stor01",
      "capacityQuota": 96636764160
    }
  ]
}
]
```

响应示例 5

```
HTTP/1.1 202 Accepted
x-hblock-request-id: 60531d110f8a4b609402156f71c99ed2
Connection: keep-alive
Date: Mon, 12 Jan 2026 09:19:36 GMT
Content-Length: 0
Server: HBlock
```

4.1.2 查询初始化进度

此操作用来查询初始化进度。

请求语法

```
GET /rest/v1/system/setup HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

说明：本 API 请求不校验签名。

响应结果

名称	类型	描述
progress	Integer	初始化进度百分比，范围为 0~100。初始化成功或者失败，进度都为 100。可以通过返回的 status 来查看初始化是否成功。
status	String	初始化状态： ● Uninitialized: 未初始化。 ● Processing: 初始化中。 ● Succeeded: 初始化成功。 ● Failed: 初始化失败。

请求示例

查询初始化进度。

```
GET /rest/v1/system/setup HTTP/1.1
Date: Mon, 14 Mar 2022 10:21:44 GMT
Host: 192.168.0.121:1443
Authorization: HBlock userName:signature
```

响应示例


```
HTTP/1.1 200 OK
Date: Mon, 14 Mar 2022 10:21:44 GMT
Content-Type: application/json;charset=utf-8
Content-Length: 46
Connection: keep-alive
x-hblock-request-id: fa25dee228334c70bd05c2edf46393a6
Server: HBlock

{
  "data": {
    "progress": 25,
    "status": "Processing"
  }
}
```

4.1.3 错误码列表

HTTP status	错误码	错误信息	描述
400	BadUserName	Value ' <i>value</i> ' at ' <i>argument</i> ' failed to satisfy constraint: Argument can only contain letters and digits, and has length between 5 and 16.	用户名不合法。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。
400	CanNotConnectToIP	Can not connect to the IP <i>IP</i> .	无法连接指定的 IP。
400	CanNotConnectToServer	Can not connect to the server <i>IP[:port]</i> .	无法连接指定的服务器。
400	ClusterNetworkDoesNotMatch	All IPs on the server <i>serverIP</i> do not match the 'cluster network' <i>clusterNetwork</i> .	服务器的 IP 和 cluster network 不匹配。
400	DuplicateIP	The IP <i>IP</i> of ' <i>argument</i> ' is duplicated.	服务的 IP 地址重复。
400	InconsistentIPformat	The server IPs in the cluster should be unified as IPv4 or IPv6 format when cluster network is not specified.	如果不指定集群网络，集群内服务器 IP 应统一为 IPv4 或 IPv6 格式。
400	InsufficientPath	The base storage pool must have at least one disk path.	基础存储池至少要有一个数据目录。
400	InsufficientPorts	Insufficient available ports for server <i>server</i> . Expand the port range to support <i>N</i> additional ports.	服务器 X 端口范围中可用端口数量不足，请调整范围使之至少增加 N 个端

			口。
400	InvalidCIDR	Value <i>CIDR</i> at ' <i>networkType</i> network' failed to satisfy constraint: Argument does not match the IP of the ' <i>localhost</i> '/'127.0.0.1'/'0:0:0:0:0:0:0:1' server.	当 server ip 包含 localhost、127.0.0.1 或 0:0:0:0:0:0:0:1 时，用户指定的 ' <i>cluster/public</i> network 和服务器 IP 不匹配。
400	InvalidClusterNetwork	Value <i>clusterNetwork</i> at ' <i>cluster</i> network' failed to satisfy constraint: Argument must satisfy CIDR specifications.	集群网格式错误。
400	InvalidDiskPath	The disk paths do not exist: <i>serverIP:diskPaths</i> [<i>,diskPaths...</i>] [; <i>serverIP:diskPaths</i>[<i>,diskPaths...</i>]]. The available disk space must be greater than or equal to 1 GiB for each disk path: <i>serverIP:diskPaths</i> [<i>,diskPaths...</i>] [; <i>serverIP:diskPaths</i>[<i>,diskPaths...</i>]]. The disk paths can not access: <i>serverIP:diskPaths</i> [<i>,diskPaths...</i>] [;	初始化过程中，因数 据目录有问题而导致的 失败。

		<i>serverIP:diskPaths[,diskPaths...]</i>. The disk paths must be directory: <i>serverIP:diskPaths[,diskPaths...]</i> [; <i>serverIP:diskPaths[,diskPaths...]</i>. The disk paths include incompatible data: <i>serverIP:diskPaths[,diskPaths...]</i> [; <i>serverIP:diskPaths[,diskPaths...]</i>. The disk paths include unsupported symbols: <i>serverIP:diskPaths[,diskPaths...]</i> [; <i>serverIP:diskPaths[,diskPaths...]</i>].	
400	InvalidDiskPathCapacityQuota	The capacity quota for the following disk paths failed to satisfy constraint. It must be integer and not greater than the total capacity of disk path. <i>serverIP:diskPaths[,diskPaths...]</i> [; <i>serverIP:diskPaths[,diskPaths...]</i>.	容量配额参数错误。

		.].].	
400	InvalidEnumValue	Value <i>value</i> at ' <i>argument</i> ' failed to satisfy constraint: Argument must satisfy enum value set: [<i>value1</i> , <i>value2</i> ...].	枚举参数不合法。
400	InvalidIP	Value <i>IP</i> at ' <i>argument</i> ' failed to satisfy constraint: Argument must satisfy IPv4 or IPv6 specifications.	IP 格式错误。
400	InvalidLong	Value <i>value</i> at ' <i>argument</i> ' failed to satisfy constraint: Argument must be of type long.	参数填写错误，必须为长整型。
400	InvalidMetaDir	The meta directory <i>serverIP:metaDir</i> is invalid. <i>reason</i>	服务数据目录无效，请修改后重试。
400	InvalidNewPassword	Value ' <i>value</i> ' at 'new password' failed to satisfy constraint: Argument must have length between 8 and 16, and must contain 3 of the following: uppercase letters, lowercase letters, digits, special symbols (~!@#\$%^&*()_+[]{} ;:.,/<>?), and must not contain any of the following: 3 consecutive repeating characters,	新密码不符合规则。 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

		3 consecutive or in-reverse order of numbers or letters (case-insensitive) , 3 consecutive or in-reverse order of keyboard sequences (case-insensitive).	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）。
400	InvalidNodeName	Value <i>value</i> at 'node name' failed to satisfy constraint: Argument must contain only letters, digits, dots (.), underscores (_) or hyphens (-), and does not exceed 63 characters, must begin with a letter or digit.	节点名称不正确。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。
400	InvalidNodeType	Value <i>value</i> at 'node type' failed to satisfy constraint: the first node type must be root.	节点类型的包含关系错误，第一个节点必须是 root。
400	InvalidNodeType	Value <i>value</i> at 'node type' failed to satisfy constraint: childNodes of room are limited to [rack, server], childNodes of rack to [server], childNodes of server to [path] only.	节点类型的包含关系错误。room 的节点仅限于 rack、server，机架的子节点是 server，server 的子节点是 path。
400	InvalidPort	Value <i>value</i> at 'argument' failed to satisfy constraint: Argument must have value between 1 and 65535.	Port 的取值必须在[1, 65535]之间。
400	InvalidPortRange	Value <i>value</i> at 'port range' failed	参数必须是 “port1-

		to satisfy constraint: Argument must satisfy pattern 'port1-port2', where port1, port2 are positive integers between 1 and 65535, port1 is less than port2.	port2”格式，port1 和 port2 必须是介于 [1,65535]之间的正整数，且 port1 小于 port2。
400	InvalidPublicNetwork	Value <i>publicNetwork</i> at 'public network' failed to satisfy constraint: Argument must satisfy CIDR specifications.	业务网格式错误。
400	InvalidServers	The current server is not in the servers list.	当前服务器不在 servers 列表内。
400	InvalidServersCount	The number of servers for <i>xx</i> service shoule be <i>value</i> .	XX 服务的服务器数量应该为 <i>value</i> 台。
400	InvalidStorName	Value ' <i>value</i> ' at ' <i>argument</i> ' failed to satisfy constraint: Argument can only contain letters, digits, hyphens (-) or underscores (_), and does not exceed 64 characters, must begin with a letter or digit.	HBlock 名称不合法。
400	InvalidTopologyContent	The content of the topology is not compliant.	拓扑文件内容解析错误。
400	MissingClusterNetwork	Value null at 'cluster network' failed to satisfy constraint: Argument must not be null when 'server ip' contains 'localhost'/'127.0.0.1'/'0:0:0:0:0:0	当 server ip 包含 localhost、127.0.0.1 或 0:0:0:0:0:0:1 时，集群网不能为空。

		:0:1'.	
400	MissingPublicNetwork	Value null at 'public network' failed to satisfy constraint: Argument must not be null when 'server ip' contains 'localhost'/'127.0.0.1'/'0:0:0:0:0:0:0:1'.	当 server ip 包含 localhost、127.0.0.1 或 0:0:0:0:0:0:0:1 时，业务网不能为 空。
400	NotInterfaceIP	The IP <i>IP</i> is not an interface IP address, check and retry.	IP 地址不是服务器的 接口 IP，请修改并重 试。
400	ProductTypeDoesNotMatch	HBlock product type does not match.	产品类型不一致。
400	ServerAlreadyInitialized	The server has been initialized.	服务器已初始化。
400	StandaloneModeNotAllowed	' <i>operation</i> ' is not supported by standalone mode of HBlock.	单机版本 HBlock 不 支持该操作。
400	TooFewServers	The number of servers must be greater than or equal to 3.	服务器必须大于等于 3 台。
400	UnrecognizedServer	Value <i>IP</i> at ' <i>argument</i> ' failed to satisfy constraint: Argument must be in the setup server list.	服务器 IP 不在初始 化服务器列表内。
400	VersionDoesNotMatch	Please replace the HBlock version of server <i>serverIP</i> [, <i>serverIP</i> ...] to <i>clutserVersion</i> , then try again.	版本号不一致。
403	InvalidUserName	Invalid user name.	用户名不正确。
403	InvalidUserNameOrPassword	Invalid user name or password hash.	用户名或密码不正 确。

409	PortChanged	Initialization failed because the port on the server <i>serverIP</i> is currently used by another service during the initialization process. Please try again.	端口在初始化过程中被其他服务占用，请重试。
409	PortConflict	The following port is/ports are in use. <i>server_ID/IP/local server:</i> <i>port_name port[, port_name port...]</i> [<i>server_ID/IP/local server:</i> <i>port_name port[, port_name port...]</i> ...]	端口冲突。
500	InitializeServerFailed	Server <i>serverIP</i> initialization failed, please check the firewall, network, memory, diskpath, etc, or contact technical team for support.	服务器初始化失败，请检查防火墙、网络、内存、数据目录等设置，或联系技术团队进行支持。

4.2 软件许可证

4.2.1 加载软件许可证

此操作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。
- 可以通过**查询 HBlock 信息**获取 HBlock 序列号 serialId，将该该序列号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，仅部分功能可用，详见**软件许可证到期（试用期或订阅模式）后仍可用的功能**。
- 永久模式许可证的维保服务过期后，升级功能不可用，其余功能均可正常使用。

请求语法

```
POST /rest/v1/system/license HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "key": "key"
}
```

请求参数

参数	类型	描述	是否必须
key	String	软件许可证密钥。	是

响应结果

名称	类型	描述
licenseId	String	软件许可证 ID。
account	String	软件许可证所属的账号。
customerName	String	客户名称。 说明： 如果申请许可证时未填写，不显示此字段。
type	String	软件许可证的购买类型： ● Subscription: 订阅模式。 ● Perpetual: 永久许可模式。
status	String	软件许可证的状态： ● Effective: 已生效。 ● Expired: 已过期。 ● Invalid: 已失效。 如果软件许可证是 Perpetual 类型的，不存在过期状态。
maximumLocalCapacity	Long	当前时间允许的最大本地卷容量，单位 TiB。如果不限容量，不返回此项。
records	Array of record	软件许可证购买记录的集合，详见“ 表 1 响应参数 record 说明 ”。
usages	Array of usage	软件许可证的使用记录的集合，详见“ 表 2 响应参数 usage 说明 ”。

表 1 响应参数 record 说明

名称	类型	描述
purchaseTime	Long	记录购买的时间，unix 时间戳（UTC），精确到毫秒。
operationtype	String	软件许可证的购买操作记录： ● New: 首次购买。

		● Expand: 扩容。 ● Renew: 续订/续保。
localCapacity	Long	本次购买对应的本地卷容量，单位 TiB。如果不限容量，不返回此项。
subscribeEffectiveTime	Long	对于订阅模式的软件许可证，本次购买操作对应的软件许可证生效时间，unix 时间戳（UTC），精确到毫秒。
subscribeExpireTime	Long	对于订阅模式的软件许可证，本次购买操作对应的软件许可证的过期时间，unix 时间戳（UTC），精确到毫秒。
maintenanceEffectiveTime	Long	对于永久许可模式的软件许可证，本次购买操作对应的维保生效时间，unix 时间戳（UTC），精确到毫秒。
maintenanceExpireTime	Long	对于永久许可模式的软件许可证，本次购买操作对应的维保的过期时间，unix 时间戳（UTC），精确到毫秒。
substatus	String	本次购买操作的当前生效状态： ● NotStart: 未生效。 ● Effective: 已生效。 ● Expired: 已过期。

表 2 响应参数 usage 说明

名称	类型	描述
maximumLocalCapacity	Long	对应时间段内允许的最大本地卷容量，单位 TiB。如果不限容量，不返回此项。
subscribeEffectiveTime	Long	对于订阅模式的软件许可证，许可证对应的生效时间，unix 时间戳（UTC），精确到毫秒。

subscribeExpireTime	Long	对于订阅模式的软件许可证，许可证对应的过期时间，unix 时间戳（UTC），精确到毫秒。
maintenanceEffectiveTime	Long	对于永久许可模式的软件许可证，许可证对应的维保生效时间，unix 时间戳（UTC），精确到毫秒。
maintenanceExpireTime	Long	对于永久许可模式的软件许可证，许可证对应的维保过期时间，unix 时间戳（UTC），精确到毫秒。
substatus	String	对于订阅模式的软件许可证，表示对应时间段内许可证的状态；对于永久许可模式的软件许可证，表示对应时间段内许可证的维保状态： ● NotStart: 未生效。 ● Effective: 已生效。 ● Expired: 已过期。

请求示例

加载软件许可证。

```
POST /rest/v1/system/license HTTP/1.1
Date: Thu, 17 Mar 2022 08:26:28 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 876
Host: 192.168.0.121:1443
Authorization: HBlock userName:signature

{
  "key": "xxxxxxxxxxxxxxxx"
}
```

响应示例

```
HTTP/1.1 200 OK
Date: Thu, 17 Mar 2022 08:26:28 GMT
Content-Length: 658
Connection: keep-alive
Content-Type: application/json; charset=utf-8
x-hblock-request-id: 24f2b6f45baf49a69fa9d0bfad1d76f3
Server: HBlock

{
  "data": {
    "licenseId": "ehc2b6a9-f3fb-4098-a6b3-3652a5d76269",
    "account": "xxx@ctyun.cn",
    "status": "Effective",
    "type": "Subscription",
    "maximumLocalCapacity": 2048,
    "records": [
      {
        "purchaseTime": 1647505314201,
        "operationtype": "New",
        "localCapacity": 2048,
        "subscribeEffectiveTime": 1647505314201,
        "subscribeExpireTime": 1655281314201,
        "substatus": "Effective"
      },
      {
        "purchaseTime": 1650097314201,
        "operationtype": "Renew",
        "localCapacity": 2048,
        "subscribeEffectiveTime": 1655281314201,
        "subscribeExpireTime": 1663057314201,
        "substatus": "NotStart"
      }
    ],
    "usages": [
      {
```

```
    "maximumLocalCapacity": 2048,  
    "subscribeEffectiveTime": 1647505314201,  
    "subscribeExpireTime": 1663057314201,  
    "substatus": "Effective"  
  }  
]  
}  
}
```

4.2.2 批量查询软件许可证信息

批量查询软件许可证信息，信息包括软件许可证的 **record** 信息以及 **usage** 信息：

- **record** 信息：购买记录的信息。
- **usage** 信息：软件许可证在不同时间段内的对应的容量和状态信息。

请求语法

```
GET /rest/v1/system/license?filter=filter&range=i-j&show=show HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization:authorization
```

请求参数

名称	类型	描述	是否必须
filter	String	设置查询的过滤条件。 过滤条件由查询属性和属性值组成，其中属性和属性值之间用单个冒号分隔（key:value）表示模糊匹配，用两个冒号分隔（key::value）表示精确匹配。可以选择多个查询，如果是或的关系，使用“or”将查询条件分隔开；如果是与的关系，使用“and”将查询条件分隔开。 支持的查询类型包括： ● licenseId：软件许可证 ID。 ● status：软件许可证的状态： ■ Effective：已生效。 ■ Expired：已过期。 ■ Invalid：无效。	否

		● type: 软件许可证的购买类型: ■ Subscription: 订阅模式。 ■ Perpetual: 永久许可模式。	
range	String	查询范围，格式: i-j，其中 i,j 是正整数，i 小于 j，j-i<1000。 按照 license 的购买时间由新到旧降序排列，返回第 i 个到第 j 个软件许可证。 如果输入的查询范围超过软件许可证的总个数，那么返回空结果集。 如果不输入 range 参数，那么返回所有软件许可证。	否
show	String	查询的输出结果，支持的输出结果包括 record, usage。 如果不填写，就是默认两个信息都输出。	否

响应结果

名称	类型	描述
licenseId	String	软件许可证 ID。
account	String	软件许可证所属的账号。
customerName	String	客户名称。 说明: 如果申请许可证时未填写，不显示此字段。
type	String	软件许可证的购买类型: ● Subscription: 订阅模式。 ● Perpetual: 永久许可模式。
status	String	软件许可证的状态: ● Effective: 已生效。 ● Expired: 已过期。 ● Invalid: 已失效。 如果软件许可证是 Perpetual 类型的，不存在过期状态。

maximumLocalCapacity	Long	当前时间允许的最大本地卷容量，单位 TiB。如果不限容量，不返回此项。
records	Array of record	软件许可证购买记录的集合，详见“表 1 响应参数 record 说明”。
usages	Array of usage	软件许可证的使用记录的集合，详见“表 2 响应参数 usage 说明”。

表 1 响应参数 record 说明

名称	类型	描述
purchaseTime	Long	记录购买的时间，unix 时间戳（UTC），精确到毫秒。
operationtype	String	软件许可证的操作记录： ● New: 首次购买。 ● Expand: 扩容。 ● Renew: 续订/续保。
localCapacity	Long	本次购买对应的本地卷容量，单位 TiB。如果不限容量，不返回此项。
subscribeEffectiveTime	Long	对于订阅模式的软件许可证，本次购买操作对应的软件许可证生效时间，unix 时间戳（UTC），精确到毫秒。
subscribeExpireTime	Long	对于订阅模式的软件许可证，本次购买操作对应的软件许可证的过期时间，unix 时间戳（UTC），精确到毫秒。
maintenanceEffectiveTime	Long	对于永久许可模式的软件许可证，本次购买操作对应的维保生效时间，unix 时间戳（UTC），精确到毫秒。
maintenanceExpireTime	Long	对于永久许可模式的软件许可证，本次购买操作

		对应的维保的过期时间，unix 时间戳（UTC），精确到毫秒。
substatus	String	本次购买操作的当前生效状态： ● NotStart: 未生效。 ● Effective: 已生效。 ● Expired: 已过期。

表 2 响应参数 usage 说明

名称	类型	描述
maximumLocalCapacity	Long	对应时间段内允许的最大本地卷容量，单位 TiB。如果不限容量，不返回此项。
subscribeEffectiveTime	Long	对于订阅模式的软件许可证，许可证对应的生效时间，unix 时间戳（UTC），精确到毫秒。
subscribeExpireTime	Long	对于订阅模式的软件许可证，许可证对应的过期时间，unix 时间戳（UTC），精确到毫秒。
maintenanceEffectiveTime	Long	对于永久许可模式的软件许可证，许可证对应的维保生效时间，unix 时间戳（UTC），精确到毫秒。
maintenanceExpireTime	Long	对于永久许可模式的软件许可证，许可证对应的维保过期时间，unix 时间戳（UTC），精确到毫秒。
substatus	String	对于订阅模式的软件许可证，表示对应时间段内许可证的状态；对于永久许可模式的软件许可证，表示对应时间段内许可证的维保状态： ● NotStart: 未生效。 ● Effective: 已生效。 ● Expired: 已过期。

请求示例

批量查询软件许可证信息。

```
GET /rest/v1/system/license?filter=type:Subscription HTTP/1.1
Date: Tue, 05 Aug 2025 03:09:08 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.65:1443
```

响应示例

```
HTTP/1.1 200 OK
x-hblock-request-id: 649a886c536148418d05e717aa30769b
Connection: keep-alive
Content-Length: 700
Date: Tue, 05 Aug 2025 03:09:08 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "licenses": [
      {
        "licenseId": "ehc2b6a9-f3fb-4098-a6b3-3652a5d76269",
        "account": "test@ctyun.cn",
        "customerName": "Customer ABC",
        "status": "Invalid",
        "type": "Subscription",
        "maximumLocalCapacity": 2048,
        "records": [
          {
            "purchaseTime": 1647505314201,
            "operationtype": "New",
            "localCapacity": 2048,
            "subscribeEffectiveTime": 1647505314201,
```

```
        "subscribeExpireTime": 1655281314201,
        "substatus": "Effective"
    },
    {
        "purchaseTime": 1650097314201,
        "operationtype": "Renew",
        "localCapacity": 2048,
        "subscribeEffectiveTime": 1655281314201,
        "subscribeExpireTime": 1663057314201,
        "substatus": "NotStart"
    }
],
"usages": [
    {
        "maximumLocalCapacity": 2048,
        "subscribeEffectiveTime": 1647505314201,
        "subscribeExpireTime": 1663057314201,
        "substatus": "Effective"
    }
]
}
}
```

4.2.3 查询指定软件许可证信息

查询指定软件许可证的信息。信息包括软件许可证的 record 信息以及 usage 信息：

- record 信息：购买记录的信息。
- usage 信息：软件许可证在不同时间段内的对应的容量和状态信息。

请求语法

```
GET /rest/v1/system/license/LicenseId?show=show HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization
```

请求参数

名称	类型	描述	是否必须
licenseId	String	软件许可证的 ID。	否
show	String	查询的输出结果，record、usage。 如果不填写，就是默认两个信息都输出。	否

响应结果

名称	类型	描述
licenseId	String	软件许可证 ID。
account	String	软件许可证所属的账号。
customerName	String	客户名称。 说明： 如果申请许可证时未填写，不显示此字段。
type	String	软件许可证的购买类型：

		● Subscription: 订阅模式。 ● Perpetual: 永久许可模式。
status	String	软件许可证的状态： ● Effective: 已生效。 ● Expired: 已过期。 ● Invalid: 已失效。 如果软件许可证是 Perpetual 类型的，不存在过期状态。
maximumLocalCapacity	Long	当前时间允许的最大本地卷容量，单位 TiB。如果不限容量，不返回此项。
records	Array of record	软件许可证的购买记录的集合，详见“表 1 响应参数 record 说明”。
usages	Array of usage	软件许可证使用记录的集合，详见“表 2 响应参数 usage 说明”。

表 1 响应参数 **record** 说明

名称	类型	描述
purchaseTime	Long	记录购买的时间，unix 时间戳（UTC），精确到毫秒。
operationtype	String	软件许可证的操作记录： ● New: 首次购买。 ● Expand: 扩容。 ● Renew: 续订/续保。
localCapacity	Long	本次购买对应的本地卷容量，单位 TiB。如果不限容量，不返回此项。
subscribeEffectiveTime	Long	对于订阅模式的软件许可证，本次购买操作对应的软件许可证生效时间，unix 时间戳（UTC），

		精确到毫秒。
subscribeExpireTime	Long	对于订阅模式的软件许可证，本次购买操作对应的软件许可证的过期时间，unix 时间戳（UTC），精确到毫秒。
maintenanceEffectiveTime	Long	对于永久许可模式的软件许可证，本次购买操作对应的维保生效时间，unix 时间戳（UTC），精确到毫秒。
maintenanceExpireTime	Long	对于永久许可模式的软件许可证，本次购买操作对应的维保的过期时间，unix 时间戳（UTC），精确到毫秒。
substatus	String	本次购买操作的当前状态： ● NotStart: 未生效。 ● Effective: 已生效。 ● Expired: 已过期。

表 2 响应参数 usage 说明

名称	类型	描述
maximumLocalCapacity	Long	对应时间段内允许的最大本地卷容量，单位 TiB。如果不限容量，不返回此项。
subscribeEffectiveTime	Long	对于订阅模式的软件许可证，许可证对应的生效时间，unix 时间戳（UTC），精确到毫秒。
subscribeExpireTime	Long	对于订阅模式的软件许可证，许可证对应的过期时间，unix 时间戳（UTC），精确到毫秒。
maintenanceEffectiveTime	Long	对于永久许可模式的软件许可证，许可证对应的维保生效时间，unix 时间戳（UTC），精确到毫秒。
maintenanceExpireTime	Long	对于永久许可模式的软件许可证，许可证对应的维保过期时间，unix 时间戳（UTC），精确到毫秒。

substatus	String	对于订阅模式的软件许可证，表示对应时间段内许可证的状态；对于永久许可模式的软件许可证，表示对应时间段内许可证的维保状态： ● NotStart：未生效。 ● Effective：已生效。 ● Expired：已过期。

请求示例 1

订阅模式：查询软件许可证 ID 为 ehc2b6a9-f3fb-4098-a6b3-3652a5d71232 的软件许可证信息。

```
GET /rest/v1/system/license/ehc2b6a9-f3fb-4098-a6b3-3652a5d71232 HTTP/1.1
Date: Tue, 05 Aug 2025 02:40:01 GMT
Authorization: HBlock userName:signature
Host:192.168.0.66:1443
```

响应示例 1

```
HTTP/1.1 200 OK
x-hblock-request-id: e4a388cfebe24398a09ee6bdc5d22b2a
Connection: keep-alive
Content-Length: 518
Date: Tue, 05 Aug 2025 02:40:01 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "licenseId": "ehc2b6a9-f3fb-4098-a6b3-3652a5d71232",
    "account": "testaccount1@ctyun.com",
    "status": "Effective",
    "type": "Subscription",
    "maximumLocalCapacity": 1024,
    "records": [
```

```
{
  "purchaseTime": 1753857636859,
  "operationtype": "New",
  "localCapacity": 1024,
  "subscribeEffectiveTime": 1753857636859,
  "subscribeExpireTime": 1785393636859,
  "substatus": "Effective"
},
"usages": [
  {
    "maximumLocalCapacity": 1024,
    "subscribeEffectiveTime": 1753857636859,
    "subscribeExpireTime": 1785393636859,
    "substatus": "Effective"
  }
]
}
```

请求示例 2

永久许可模式：查询软件许可证 ID 为 `qws2b6a9-f3fb-4098-a6b3-3652a5a76530` 的软件许可证信息。

```
GET /rest/v1/system/license/qws2b6a9-f3fb-4098-a6b3-3652a5a76530 HTTP/1.1
Date: Tue, 05 Aug 2025 02:17:13 GMT
Authorization: HBlock userName:signature
Host:192.168.0.65:1443
```

响应示例 2

```
HTTP/1.1 200 OK
x-hblock-request-id: 7b4ee400d293497bba0c9182b256e678
Connection: keep-alive
Content-Length: 518
```

```
Date: Tue, 05 Aug 2025 02:17:13 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "licenseId": "qws2b6a9-f3fb-4098-a6b3-3652a5a76530",
    "account": "test@ctyun.cn",
    "customerName": "Customer ABC",
    "status": "Effective",
    "type": "Perpetual",
    "maximumLocalCapacity": 2048,
    "records": [
      {
        "purchaseTime": 1754357667366,
        "operationtype": "New",
        "localCapacity": 2048,
        "maintenanceEffectiveTime": 1754357667366,
        "maintenanceExpireTime": 1785893667366,
        "substatus": "Effective"
      }
    ],
    "usages": [
      {
        "maximumLocalCapacity": 2048,
        "maintenanceEffectiveTime": 1754357667366,
        "maintenanceExpireTime": 1785893667366,
        "substatus": "Effective"
      }
    ]
  }
}
```

4.2.4 错误码列表

HTTP status	错误码	错误信息	描述
400	ExceedLocalCapacityLimit	The total capacity of local mode LUN can not exceed the maximum capacity limit agreed in the software license.	本地卷容量不能超过软件许可证规定的容量。
400	InvalidFilterEnum	Value <i>value</i> at 'filter' failed to satisfy constraint: Argument can only contain: [<i>licenseId</i> , <i>status</i> , <i>type</i>].	filter 参数不正确。
400	InvalidLicenseKey	The license key is invalid.	软件许可证无效。
400	InvalidRange	Value <i>value</i> at 'range' failed to satisfy constraint: Argument must satisfy pattern 'i-j', where i, j are positive integers, i is less than j, and j-i<1000.	range 参数不正确。
400	InvalidShow	Value <i>value</i> at 'show' failed to satisfy constraint: Argument can only contain: [<i>record</i> , <i>usage</i>].	show 参数不正确。
400	LicenseKeyDoesNotMatch	This license key does not match the current HBlock. Please obtain the software	软件许可证密钥与当前 HBlock 不匹配。

		license again.	
400	ReimportLicenseKey	Do not reimport the license key that is in effect.	重复导入正在生效的软件许可证密钥。
400	SystemTimeTooEarly	System time is <i>time</i> , it cannot be less than the effective time <i>time</i> of the software license.	系统时钟不能早于软件许可证的生效时间。
400	SystemTimeTooLate	System time is <i>time</i> , it cannot be later than the expiration time <i>time</i> of the software license.	系统时钟不能晚于软件许可证的结束时间。
403	LicenseExpired	The software license has expired.	软件许可证过期。
403	LicenseMaintenanceExpired	The maintenance for software license has expired.	软件许可证过保。
403	LicenseNotEffective	The license is not yet effective.	许可证生效时间未到。
403	TrialNotEffective	The trial period has not yet begun.	试用期生效时间未到。
404	NoSuchLicenseID	The software license with ID <i>ID</i> does not exist.	软件许可证不存在。

4.3 卷管理

4.3.1 创建卷

此操作用来创建卷。创建卷时，如果指定的 `target` 名称不存在，则同时创建 `target`。

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：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）、Endpoint、上云签名认证版本、区域、AK/SK、是否压缩数据。

请求语法

- 本地卷：
 - 单机版

```
POST /rest/v1/block/lun HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "lunName": LunName,
  "targetName": targetName,
  "capacity": capacity,
  "storageMode": storageMode,
  "config": {
    "sectorSize": sectorSize,
    "writePolicy": writePolicy,
    "qosPolicy": {
      "qosName": qosName,
      "IOPS": IOPS,
```

```
    "readIOPS": readIOPS,
    "writeIOPS": writeIOPS,
    "Bps": Bps,
    "readBps": readBps,
    "writeBps": writeBps,
    "IOPSBurst": IOPSBurst,
    "readIOPSBurst": readIOPSBurst,
    "writeIOPSBurst": writeIOPSBurst,
    "BpsBurst": BpsBurst,
    "readBpsBurst": readBpsBurst,
    "writeBpsBurst": writeBpsBurst,
    "IOPSBurstSecs": IOPSBurstSecs,
    "readIOPSBurstSecs": readIOPSBurstSecs,
    "writeIOPSBurstSecs": writeIOPSBurstSecs,
    "BpsBurstSecs": BpsBurstSecs,
    "readBpsBurstSecs": readBpsBurstSecs,
    "writeBpsBurstSecs": writeBpsBurstSecs
  },
  "path": path
}
```

■ 集群版

POST /rest/v1/block/lun HTTP/1.1

Date: *date*

Content-Type: application/json; charset=utf-8

Content-Length: *Length*

Host: *ip:port*

Authorization: *authorization*

```
{
  "lunName": LunName,
  "targetName": targetName,
  "capacity": capacity,
  "storageMode": storageMode,
  "config": {
    "localStorageClass": LocalStorageClass,
```

```
"minReplica": minReplica,
"ECfragmentSize": ECfragmentSize,
"redundancyOverlap": redundancyOverlap,
"sectorSize": sectorSize,
"cachePool": poolName,
"pool": poolName,
"highAvailability": highAvailability,
"writePolicy": writePolicy,
"serverAffinity": {
  "autoFailback": autoFailback,
  "priority": [
    SeverID,ServerID...
  ]
},
"qosPolicy": {
  "qosName": qosName,
  "IOPS": IOPS,
  "readIOPS": readIOPS,
  "writeIOPS": writeIOPS,
  "Bps": Bps,
  "readBps": readBps,
  "writeBps": writeBps,
  "IOPSBurst": IOPSBurst,
  "readIOPSBurst": readIOPSBurst,
  "writeIOPSBurst": writeIOPSBurst,
  "BpsBurst": BpsBurst,
  "readBpsBurst": readBpsBurst,
  "writeBpsBurst": writeBpsBurst,
  "IOPSBurstSecs": IOPSBurstSecs,
  "readIOPSBurstSecs": readIOPSBurstSecs,
  "writeIOPSBurstSecs": writeIOPSBurstSecs,
  "BpsBurstSecs": BpsBurstSecs,
  "readBpsBurstSecs": readBpsBurstSecs,
  "writeBpsBurstSecs": writeBpsBurstSecs
}
}
```


- 上云卷:

- 单机版

```
POST /rest/v1/block/lun HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "lunName": LunName,
  "targetName": targetName,
  "capacity": capacity,
  "storageMode": storageMode,
  "config": {
    "sectorSize": sectorSize,
    "writePolicy": writePolicy,
    "qosPolicy": {
      "qosName": qosName,
      "IOPS": IOPS,
      "readIOPS": readIOPS,
      "writeIOPS": writeIOPS,
      "Bps": Bps,
      "readBps": readBps,
      "writeBps": writeBps,
      "IOPSBurst": IOPSBurst,
      "readIOPSBurst": readIOPSBurst,
      "writeIOPSBurst": writeIOPSBurst,
      "BpsBurst": BpsBurst,
      "readBpsBurst": readBpsBurst,
      "writeBpsBurst": writeBpsBurst,
      "IOPSBurstSecs": IOPSBurstSecs,
      "readIOPSBurstSecs": readIOPSBurstSecs,
      "writeIOPSBurstSecs": writeIOPSBurstSecs,
      "BpsBurstSecs": BpsBurstSecs,
      "readBpsBurstSecs": readBpsBurstSecs,
      "writeBpsBurstSecs": writeBpsBurstSecs
    }
  }
}
```

```
    },  
    "path": path  
  },  
  "cloud": {  
    "provider": provider,  
    "bucketName": bucketName,  
    "prefix": prefix,  
    "accessKey": accessKey,  
    "secretKey": secretKey,  
    "endpoint": endpoint,  
    "signVersion": signVersion,  
    "region": region,  
    "objectSize": objectSize,  
    "storageClass": storageClass,  
    "compression": compression  
  }  
}
```

■ 集群版

```
POST /rest/v1/block/lun HTTP/1.1  
Date:date  
Content-Type: application/json; charset=utf-8  
Content-Length: length  
Host: ip:port  
Authorization: authorization  
  
{  
  "lunName": LunName,  
  "targetName": targetName,  
  "capacity": capacity,  
  "storageMode": storageMode,  
  "config": {  
    "localStorageClass": localStorageClass,  
    "minReplica": minReplica,  
    "ECfragmentSize": ECfragmentSize,
```

```
"redundancyOverlap": redundancyOverlap,
"sectorSize": sectorSize,
"cachePool": poolName,
"pool": poolName,
"highAvailability": highAvailability,
"writePolicy": writePolicy,
"serverAffinity": {
  "autoFailback": autoFailback,
  "priority": [
    SeverID,ServerID...
  ]
},
"qosPolicy": {
  "qosName": qosName,
  "IOPS": IOPS,
  "readIOPS": readIOPS,
  "writeIOPS": writeIOPS,
  "Bps": Bps,
  "readBps": readBps,
  "writeBps": writeBps,
  "IOPSBurst": IOPSBurst,
  "readIOPSBurst": readIOPSBurst,
  "writeIOPSBurst": writeIOPSBurst,
  "BpsBurst": BpsBurst,
  "readBpsBurst": readBpsBurst,
  "writeBpsBurst": writeBpsBurst,
  "IOPSBurstSecs": IOPSBurstSecs,
  "readIOPSBurstSecs": readIOPSBurstSecs,
  "writeIOPSBurstSecs": writeIOPSBurstSecs,
  "BpsBurstSecs": BpsBurstSecs,
  "readBpsBurstSecs": readBpsBurstSecs,
  "writeBpsBurstSecs": writeBpsBurstSecs
}
},
"cloud": {
  "provider": provider,
  "bucketName": bucketName,
  "prefix": prefix,
```

```

    "accessKey": accessKey,
    "secretKey": secretKey,
    "endpoint": endpoint,
    "signVersion": signVersion,
    "region": region,
    "objectSize": objectSize,
    "storageClass": storageClass,
    "compression": compression
  }
}

```

请求参数

参数	类型	描述	是否必须
lunName	String	卷名称。 取值：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。	是
targetName	String	iSCSI target 名称。 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。 取值：长度范围是 1~16，可以由小写字母、数字、句点（.）和短横线（-）组成，且仅支持以字母或数字开头。	是
capacity	Integer	卷容量。 取值：[1,1048576]，单位是 GiB。	是
storageMode	String	卷的存储类型。	否

		取值： ● Local：本地模式，数据全部保留在本地。 ● Cache：缓存模式，本地保留部分热数据，全部数据异步存储到云端。 ● Storage：存储模式，本地保留全部数据，并异步存储到云端。 存储类型默认为 Local 。	
<code>config.localStorageClass</code>	String	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，$N \geq M$，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	否

		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。	
config.minReplica	Integer	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：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，	否

		取值范围是[N, N+M]，默认值为 N。	
<code>config.ECfragmentSize</code>	Integer	纠删码模式分片大小（仅集群版支持）。 <code>config.localStorageClass</code> 为 EC 模式时，此设置才生效，否则忽略。 取值：1、2、4、8、16、32、64、128、256、512、1024、2048、4096，单位是 KiB。默认值为 16。	否
<code>config.redundancyOverlap</code>	Integer	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 <code>path</code> 上。 注意：如果存储池故障域级别为 <code>path</code>，此参数不生效。 取值：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。	否
<code>config.sectorSize</code>	Integer	扇区大小。 取值：512、4096，单位是 bytes。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 4 KiB，则推荐选择 4096；单次 I/O 操作的数据大小接近 512 bytes，则推荐选择 512。如果对接 VMware 等虚拟化平台，则推荐选择 512。	否

config.cachePool	String	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。 取值：长度范围是 1~16，只能由字母、数字和短横线（-）、下划线（_）组成，字母区分大小写，且仅支持以字母和数字开头。	否
config.pool	String	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。 取值：长度范围是 1~16，只能由字母、数字和短横线（-）、下划线（_）组成，字母区分大小写，且仅支持以字母和数字开头。默认使用基础存储池。	否
config.highAvailability	String	是否开启高可用模式（仅集群版支持）。 取值： ● ActiveStandby：启用主备，该卷关联对应 iSCSI target 下的所有 IQN。 ● Disabled：不启用主备，该卷关联对应 iSCSI target 下的 1 个 IQN。 默认值为 ActiveStandby。	否
config.writePolicy	String	卷的写策略。 取值： ● 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。适用于对性能要求较高，稳定性要求不高的场景。	否

		● 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。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: 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。	
<code>config.serverAffinity</code>	Object	卷主备分布优先级设置（仅集群版支持），详见“表 1 请求参数 serverAffinity 说明（仅集群版支持）”。	否
<code>config.qosPolicy</code>	Object	卷关联的 QoS 策略信息，详见“表 2 请求参数 config.qosPolicy 说明”。	否
<code>config.path</code>	String	创建卷时，指定存储卷数据的数据目录（仅单机版支持）。 取值：只能包含字母、数字、汉字和特殊字符（~!@\$()_+-.:）。如果创建卷时不指定数据目录，使用服务器设置的默认数据目录。	否
<code>cloud.provider</code>	String	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。	否
<code>cloud.bucketName</code>	String	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。	上云 卷必 填
<code>cloud.prefix</code>	String	设置对象存储的前缀名称，设置前缀名称后，	否

		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：长度范围是 1~256。	
cloud.accessKey	String	对象存储服务的 Access Key。	上云卷必填
cloud.secretKey	String	对象存储服务的 Secret Access Key。	上云卷必填
cloud.endpoint	String	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。	上云卷必填
cloud.signVersion	String	指定上云签名认证的类型。 取值： - v2：V2 签名认证。 - v4：V4 签名认证。 默认值为 v2。	否
cloud.region	String	对象存储服务的 Endpoint 资源池所在区域。 OOS region 详见 OOS Endpoint 和 Region。 V4 签名时，此项必填。	否

cloud.objectSize	Integer	数据存储在对云端的大小。 取值：128、256、512、1024、2048、4096、8192，单位是 KiB。默认值为 1024。	否
cloud.storageClass	String	设置数据在云端的存储类型。 取值： ● STANDARD：标准存储。 ● STANDARD_IA：低频访问存储。 默认为 STANDARD。	否
cloud.compression	String	是否压缩数据上传至云端。 取值： ● Enabled：压缩数据上传至云端。 ● Disabled：不压缩数据上传至云端。 默认值为 Enabled。	否

表 1 请求参数 serverAffinity（仅集群版支持）说明

参数	类型	描述	是否必须
autoFailback	String	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。	否
priority	Array of string	指定卷主备分布优先级的服务器 ID（仅集群版支持），系统根据指定的服务器 ID 顺序来选择卷的主	否

		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。 取值：服务器 ID。	

表 2 请求参数 config.qosPolicy 说明

参数	类型	描述	是否必须
qosName	String	指定 QoS 策略名称。 说明：创建卷时，如果指定 QoS 策略名称不存在，那么同时创建 QoS 策略，新创建的 QoS 策略的回收策略默认为 Delete。 对于集群版，不指定 QoS 策略时： ● 如果卷的缓存存储池和存储池同时存在，卷的 QoS 策略为缓存存储池设置的“存储池内卷的默认 QoS 策略”；如果缓存存储池未设置“存储池内卷的默认 QoS 策略”，则卷无 QoS 策略。 ● 如果卷只有存储池，卷的 QoS 策略为存储池设置的“存储池内卷的默认 QoS 策略”；如果存储池未设置“存储池内卷的默认 QoS 策略”，则卷无 QoS 策略。 取值：长度范围 1~64，只能由字母、数字和短横线(-)组成，区分大小写，且仅支持以字母或数字开头。 注意：	否

		● 如果没有指定 qosName，但是设置了 IOPS/Bps 中的任何一个参数，则根据 IOPS/Bps 自动创建一个 QoS 策略，系统为策略名赋值：lun-lunname-qos-timestamp。 ● 如果指定了 qosName 以及 IOPS/Bps 中的任何一个参数，则 qosName 不能与已存在的 QoS 策略重名。 ● 如果指定了 qosName，没有设置 IOPS/Bps 中的任何一个参数，则 qosName 需要为已存在的 QoS 策略名称。	
IOPS	Long	每秒能够进行读写操作次数的最大值。 取值：取值范围为[-1, 999999999]，默认值为-1。-1 表示不限制。	否
readIOPS	Long	每秒能够进行读操作次数的最大值。 取值：取值范围为[-1, 999999999]，默认值为-1。-1 表示不限制。	否
writeIOPS	Long	每秒能够进行写操作次数的最大值。 取值：取值范围为[-1, 999999999]，默认值为-1。-1 表示不限制。	否
Bps	Long	每秒可传输数据量的最大值。 取值：取值范围为[-1, 4096000000000]，默认值为-1，单位是 B/s。-1 表示不限制。	否
readBps	Long	读带宽上限。 取值：取值范围为[-1, 4096000000000]，默认值为-1，单位是 B/s。-1 表示不限制。	否
writeBps	Long	写带宽上限。 取值：取值范围为[-1, 4096000000000]，默认值	否

		为-1，单位是 B/s。-1 表示不限制。	
IOPSBurst	Long	使用 Burst 功能时，每秒能够进行读写操作次数的最大值。 取值：只有当 IOPS 大于等于 1 时，此项设置为 -1 或(IOPS, 999999999]内的正整数方可生效。默认值为-1，表示不限制。	否
readIOPSBurst	Long	使用 Burst 功能时，每秒能够进行读操作次数的最大值。 取值：只有当 readIOPS 大于等于 1 时，此项设置为 -1 或(readIOPS, 999999999]内的正整数方可生效。默认值为-1，表示不限制。	否
writeIOPSBurst	Long	使用 Burst 功能时，每秒能够进行写操作次数的最大值。 取值：只有当 writeIOPS 大于等于 1 时，此项设置为 -1 或(writeIOPS, 999999999]内的正整数方可生效。默认值为-1，表示不限制。	否
BpsBurst	Long	使用 Burst 功能时，每秒可传输的数据量最大值。 取值：只有当 Bps 大于等于 1 时，此项设置为 -1 或(Bps, 4096000000000]内的正整数方可生效，默认值为-1，单位是 B/s。-1 表示不限制。	否
readBpsBurst	Long	使用 Burst 功能时，读带宽上限。 取值：只有当 readBps 大于等于 1 时，此项设置为 -1 或(readBps, 4096000000000]内的正整数方可生效，默认值为-1，单位是 B/s。-1 表示不限制。	否
writeBpsBurst	Long	使用 Burst 功能时，写带宽上限。	否

		取值：只有当 writeBps 大于等于 1 时，此项设置为-1 或(writeBps, 4096000000000]内的正整数方可生效，默认值为-1，单位是 B/s。-1 表示不限制。	
IOPSBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的能力进行读写操作所能持续的时间。 注意： 只有在 IOPS Burst 功能启用时，此配置才生效。 取值：取值范围为[1, 999999999]，默认值为 1，单位是秒。	否
readIOPSBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的能力进行读操作所能持续的时间。 注意： 只有在 read IOPS Burst 功能启用时，此配置才生效。 取值：取值范围为[1, 999999999]，默认值为 1，单位是秒。	否
writeIOPSBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的能力进行写操作所能持续的时间。 注意： 只有在 write IOPS Burst 功能启用时，此配置才生效。 取值：取值范围为[1, 999999999]，默认值为 1，单位是秒。	否
BpsBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的流量能力所能持续的时间。 注意： 只有在 Bps Burst 功能启用时，此配置才生效。 取值：取值范围为[1, 999999999]，默认值为	否

		1，单位是秒。	
readBpsBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的读流量能力所能持续的时间。 注意：只有在 read Bps Burst 功能启用时，此配置才生效。 取值：取值范围为[1, 999999999]，默认值为1，单位是秒。	否
writeBpsBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的写流量能力所能持续的时间。 注意：只有在 write Bps Burst 功能启用时，此配置才生效。 取值：取值范围为[1, 999999999]，默认值为1，单位是秒。	否

响应结果

名称	类型	描述
iSCSITargets	Array of iSCSITarget	iSCSI target 属性集合，详见“表 3 响应参数 iSCSITarget 说明”。
message	String	如果 storageMode 设置为 Local，但是请求体包含了上云参数，会返回此信息："message": "Ignore cloud property when LUN storage mode is local."。

表 3 响应参数 iSCSITarget 说明

名称	类型	描述
iqn	String	iSCSI target IQN。
status	String	卷对应的 iSCSI target 的状态： ● Active: 主 target。

		● Standby: 热备 target。 ● Unavailable: 冷备 target。
ips	Array of ip	iSCSI target 的 IP 属性集合，详见“表 4 响应参数 ip 说明”。
serverId	String	服务器 ID。

表 4 响应参数 ip 说明

名称	类型	描述
ip	String	iSCSI target 的 IP。
port	Integer	iSCSI target 的端口号。

请求示例 1

单机版本地卷：创建卷 lun8。

```
POST /rest/v1/block/lun HTTP/1.1
Date: Wed, 18 Oct 2023 02:56:09 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 210
Host:192.168.0.69:1443
```

```
{
  "lunName": "lun8",
  "targetName": "target8",
  "capacity": 500,
  "config": {
    "sectorSize": 512,
    "writePolicy": "WriteThrough",
    "path": "/mnt/storage01"
  }
}
```

响应示例 1

```
HTTP/1.1 200 OK
x-hblock-request-id: c79736f32fe249cc8c40af6d90d3fd68
Connection: keep-alive
Content-Length: 158
Date: Wed, 18 Oct 2023 02:56:09 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "iSCSITargets": [
      {
        "iqn": "iqn.2012-08.cn.ctyunapi.oos:target8.2",
        "status": "Active",
        "ips": [
          {
            "ip": "192.168.0.202",
            "port": 3260
          }
        ],
        "serverId": "hblock_1"
      }
    ]
  }
}
```

请求示例 2

集群本地卷：创建卷 lun08a。

```
POST /rest/v1/block/lun HTTP/1.1
Date: Thu, 5 Dec 2024 09:11:40 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 529
```

```
Host:192.168.0.192:1443

{
  "lunName": "lun08a",
  "targetName": "target08",
  "capacity": 500,
  "config": {
    "localStorageClass": "EC 2+1",
    "minReplica": 3,
    "ECfragmentSize": 1024,
    "sectorSize": 512,
    "cachePool": "default",
    "pool": "pool2",
    "highAvailability": "ActiveStandby",
    "writePolicy": "WriteThrough",
    "serverAffinity": {
      "autoFailback": "Enabled",
      "priority": [
        "hblock_1"
      ]
    }
  }
}
```

响应示例 2

```
HTTP/1.1 200 OK
x-hblock-request-id: 9430c50b4152442c8401d3fc8f4ced1b
Connection: keep-alive
Content-Length: 303
Date: Thu, 5 Dec 2024 09:11:40 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
```

```
{
  "iSCSITargets": [
    {
      "iqn": "iqn.2012-08.cn.ctyunapi.oos:target08.11",
      "status": "Unavailable",
      "ips": [
        {
          "ip": "192.168.0.192",
          "port": 3260
        }
      ],
      "serverId": "hblock_1"
    },
    {
      "iqn": "iqn.2012-08.cn.ctyunapi.oos:target08.12",
      "status": "Unavailable",
      "ips": [
        {
          "ip": "192.168.0.117",
          "port": 3260
        }
      ],
      "serverId": "hblock_2"
    }
  ]
}
```

请求示例 3

单机版上云卷：创建卷 **luna9**。

```
POST /rest/v1/block/lun HTTP/1.1
Date: Mon, 13 May 2024 09:50:20 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 617
```

```
Host:192.168.0.32:1443

{
  "lunName": "luna9",
  "targetName": "targeta",
  "storageMode": "Storage",
  "capacity": 500,
  "config": {
    "sectorSize": 512,
    "writePolicy": "WriteThrough"
  },
  "cloud": {
    "bucketName": "hblocktest3",
    "prefix": "hblock1",
    "accessKey": "cb22b08b1f9229f85874",
    "secretKey": "*****",
    "endpoint": "oos-cn.ctyunapi.cn",
    "signVersion": "v4",
    "region": "cn",
    "objectSize": 1024,
    "storageClass": "STANDARD",
    "compression": "Enabled"
  }
}
```

响应示例 3

```
HTTP/1.1 200 OK
x-hblock-request-id: 2019e40fcec64256b68983a9b03e844e
Connection: keep-alive
Content-Length: 157
Date: Mon, 13 May 2024 09:50:20 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
```

```
{
  "data": {
    "iSCSITargets": [
      {
        "iqn": "iqn.2012-08.cn.ctyunapi.oos:targeta.2",
        "status": "Active",
        "ips": [
          {
            "ip": "192.168.0.32",
            "port": 3260
          }
        ],
        "serverId": "hblock_1"
      }
    ]
  }
}
```

请求示例 4

集群版上云卷：创建卷 lun03a。

```
POST /rest/v1/block/lun HTTP/1.1
Date: Thu, 9 May 2024 07:15:36 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 749
Host:192.168.0.192:1443
```

```
{
  "lunName": "lun03a",
  "targetName": "target03",
  "capacity": 300,
  "storageMode": "Cache",
  "config": {
    "localStorageClass": "EC 2+1",
    "minReplica": 3,
```

```
    "ECfragmentSize": 1024,  
    "sectorSize": 512,  
    "highAvailability": "ActiveStandby",  
    "writePolicy": "WriteThrough"  
  },  
  "cloud": {  
    "bucketName": "hblocktest2",  
    "prefix": "hblock",  
    "accessKey": "d0434d3e9e8dd462d6e0",  
    "secretKey": "*****",  
    "endpoint": "oos-cn.ctyunapi.cn",  
    "signVersion": "v4",  
    "region": "cn",  
    "objectSize": 1024,  
    "storageClass": "STANDARD",  
    "compression": "Enabled"  
  }  
}
```

响应示例 4

```
HTTP/1.1 200 OK  
x-hblock-request-id: 3d4faf99cb9447bfaa337fd2a63187eb  
Connection: keep-alive  
Content-Length: 296  
Date: Thu, 9 May 2024 07:15:36 GMT  
Content-Type: application/json; charset=utf-8  
Server: HBlock  
  
{  
  "data": {  
    "iSCSITargets": [  
      {  
        "iqn": "iqn.2012-08.cn.ctyunapi.oos:target03.7",  
        "status": "Active",  
        "ips": [  

```

```
{
  {
    "ip": "192.168.0.192",
    "port": 3260
  }
],
"serverId": "hblock_2"
},
{
  "iqn": "iqn.2012-08.cn.ctyunapi.oos:target03.8",
  "status": "Unavailable",
  "ips": [
    {
      "ip": "192.168.0.102",
      "port": 3260
    }
  ],
  "serverId": "hblock_3"
}
]
}
}
```


4.3.2 创建克隆卷

此操作用来创建克隆卷。

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。

请求语法

- 单机版

```
POST /rest/v1/block/lun/clone HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "lunName": LunName,
  "snapshotName": snapshotName,
  "targetName": targetName,
  "capacity": capacity,
  "config": {
    "sectorSize": sectorSize,
    "writePolicy": writePolicy,
    "qosPolicy": {
      "qosName": qosName,
      "IOPS": IOPS,
      "readIOPS": readIOPS,
      "writeIOPS": writeIOPS,
      "Bps": Bps,
      "readBps": readBps,
      "writeBps": writeBps,
```

```
    "IOPSBurst": IOPSBurst,
    "readIOPSBurst": readIOPSBurst,
    "writeIOPSBurst": writeIOPSBurst,
    "BpsBurst": BpsBurst,
    "readBpsBurst": readBpsBurst,
    "writeBpsBurst": writeBpsBurst,
    "IOPSBurstSecs": IOPSBurstSecs,
    "readIOPSBurstSecs": readIOPSBurstSecs,
    "writeIOPSBurstSecs": writeIOPSBurstSecs,
    "BpsBurstSecs": BpsBurstSecs,
    "readBpsBurstSecs": readBpsBurstSecs,
    "writeBpsBurstSecs": writeBpsBurstSecs
  },
  "path": path
}
```

● 集群版

```
POST /rest/v1/block/lun/clone HTTP/1.1
Date:date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "lunName": LunName,
  "snapshotName": snapshotName,
  "targetName": targetName,
  "capacity": capacity,
  "config": {
    "localStorageClass": localStorageClass,
    "minReplica": minReplica,
    "ECfragmentSize": ECfragmentSize,
    "redundancyOverlap": redundancyOverlap,
    "sectorSize": sectorSize,
```

```
"cachePool": poolName,
"pool": poolName,
"highAvailability": highAvailability,
"writePolicy": writePolicy,
"serverAffinity": {
  "autoFailback": autoFailback,
  "priority": [
    SeverID,ServerID...
  ]
},
"qosPolicy": {
  "qosName": qosName,
  "IOPS": IOPS,
  "readIOPS": readIOPS,
  "writeIOPS": writeIOPS,
  "Bps": Bps,
  "readBps": readBps,
  "writeBps": writeBps,
  "IOPSBurst": IOPSBurst,
  "readIOPSBurst": readIOPSBurst,
  "writeIOPSBurst": writeIOPSBurst,
  "BpsBurst": BpsBurst,
  "readBpsBurst": readBpsBurst,
  "writeBpsBurst": writeBpsBurst,
  "IOPSBurstSecs": IOPSBurstSecs,
  "readIOPSBurstSecs": readIOPSBurstSecs,
  "writeIOPSBurstSecs": writeIOPSBurstSecs,
  "BpsBurstSecs": BpsBurstSecs,
  "readBpsBurstSecs": readBpsBurstSecs,
  "writeBpsBurstSecs": writeBpsBurstSecs
}
}
}
```

请求参数

参数	类型	描述	是否必须
lunName	String	克隆卷名称。 取值：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。	是
snapshotName	String	指定克隆卷关联的快照名称。 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。	是
targetName	String	指定克隆卷关联的 iSCSI target 名称，可以跟源卷的 iSCSI target 不同。 取值：长度范围是 1~16，可以由小写字母、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 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。	是
capacity	Integer	指定克隆卷的容量。 取值：[1,1048576]，单位是 GiB。 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。	否
config.localStorageClass	String	指定克隆卷的冗余模式（仅集群版支持）。 注意： 如果设置了克隆卷的 minReplica 或 redundancyOverlap，则必须同时指定该参	否

		数。 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，$N \geq M$，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。	
config.minReplica	Integer	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。 注意：如果指定该参数，则必须指定克隆卷的 <code>localStorageClass</code>。 取值： ● 如果没有指定克隆卷的 <code>localStorageClass</code>：默认值为源卷的最小副本数。 ● 如果指定了克隆卷的 <code>localStorageClass</code>：对于副本卷，取值范围是[1, N]，N 为副本模式卷的副本数，	否

		默认值为 1；对于 EC 卷，取值范围是[N, N+M]，默认值为 N。	
<code>config.ECfragmentSize</code>	Integer	指定克隆卷的纠删码模式分片大小（仅集群版支持）。 <code>config.localStorageClass</code> 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。	否
<code>config.redundancyOverlap</code>	Integer	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。如果指定该参数，则必须指定克隆卷的 <code>localStorageClass</code>。 取值： ● 如果没有指定克隆卷的 <code>localStorageClass</code>：默认值为源卷的折叠副本数。 ● 如果指定了克隆卷的 <code>localStorageClass</code>：对副本模式，取值范围是[1, 副本数]；对于 EC 卷，取值范围是[1, N+M]。默认值为 1。	否
<code>config.sectorSize</code>	Integer	指定克隆卷的扇区大小。	否

		取值：512、4096，单位是 bytes。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 4 KiB，则推荐选择 4096；单次 I/O 操作的数据大小接近 512 Bytes，则推荐选择 512。如果对接 VMware 等虚拟化平台，则推荐选择 512。	
config.cachePool	String	指定缓存存储池（仅集群版支持）。 取值：长度范围是 1~16，只能由字母、数字和短横线（-）、下划线（_）组成，字母区分大小写，且仅支持以字母和数字开头。默认值与源卷的配置一致。 注意： - pool 与 cachePool 不能设置为同一个存储池。 - 当克隆卷的 pool 和 cachePool 配置与源卷一致，无需额外设置。若单独设置了克隆卷的 pool 或 cachePool，则不再沿用源卷的 pool 和 cachePool，而是采用所设参数值。例如，源卷同时有 cachePool 和 pool，若克隆卷仅重新设置了 pool 而未设置 cachePool，则克隆卷使用新设置的 pool，无 cachePool；反之，若克隆卷设置了 cachePool，则必须同时设置 pool，或者让 pool 使用基础存储池。	否
config.pool	String	指定存储池（仅集群版支持）。	否

		取值：长度范围是 1~16，只能由字母、数字和短横线（-）、下划线（_）组成，字母区分大小写，且仅支持以字母和数字开头。默认值与源卷的配置一致。 注意： ● pool 与 cachePool 不能设置为同一个存储池。 ● 当克隆卷的 pool 和 cachePool 配置与源卷一致，无需额外设置。若单独设置了克隆卷的 pool 或 cachePool，则不再沿用源卷的 pool 和 cachePool，而是采用所设参数值。例如，源卷同时有 cachePool 和 pool，若克隆卷仅重新设置了 pool 而未设置 cachePool，则克隆卷使用新设置的 pool，无 cachePool；反之，若克隆卷设置了 cachePool，则必须同时设置 pool，或者让 pool 使用基础存储池。	
config.highAvailability	String	指定克隆卷的高可用类型（仅集群版支持）。 取值： ● ActiveStandby：启用主备，该卷关联对应 target 下的所有 IQN。 ● Disabled：不启用主备，该卷关联对应 target 下的 1 个 IQN。 默认值为源卷的高可用类型。	否
config.writePolicy	String	指定克隆卷的写策略。 取值： ● WriteBack：回写，指数据写入到内存后	否

		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: 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: 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。	
<code>config.serverAffinity</code>	Object	卷主备分布优先级设置，详见“表 1 请求参数 <code>config.serverAffinity</code> 说明”。	否
<code>config.qosPolicy</code>	Object	卷关联的 QoS 策略信息，详见“表 2 请求参数 <code>config.qosPolicy</code> 说明”。	否
<code>config.path</code>	String	指定存储克隆卷数据的数据目录（仅单机版支持）。 取值：只能包含字母、数字、汉字和特殊字符（~!@\$()_+-;.:）。如果创建卷时不指定数据目录，默认与源卷配置保持一致。	否

表 1 请求参数 `config.serverAffinity` 说明

参数	类型	描述	是否必须
<code>autoFailback</code>	String	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（仅集群版支持），即针对卷主备状态，当	否

		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。	
serverAffinity	Array of string	指定卷主备分布优先级的服务器 ID（仅集群版支持），系统根据指定的服务器 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。 取值：服务器 ID。	否

表 2 请求参数 **config.qosPolicy** 说明

参数	类型	描述	是否必须
qosName	String	指定 QoS 策略名称。 说明：创建卷时，如果指定 QoS 策略名称不存在，那么同时创建 QoS 策略，新创建的 QoS 策略的回收策略默认为 Delete。 对于集群版，不指定 QoS 策略时： ● 如果卷的缓存存储池和存储池同时存在，卷的 QoS 策略为缓存存储池设置的“存储池内卷的默认 QoS 策略”；如果缓存存储池未设置“存储池内卷的默认 QoS 策略”，则卷无 QoS 策略。	否

		- 如果卷只有存储池，卷的 QoS 策略为存储池设置的“存储池内卷的默认 QoS 策略”；如果存储池未设置“存储池内卷的默认 QoS 策略”，则卷无 QoS 策略。 取值：长度范围 1~64，只能由字母、数字和短横线(-)组成，区分大小写，且仅支持以字母或数字开头。 注意： - 如果没有指定 qosName，但是设置了 IOPS/Bps 中的任何一个参数，则根据 IOPS/Bps 自动创建一个 QoS 策略，系统为策略名赋值：lun-lunname-qos-timestamp。 - 如果指定了 qosName 以及 IOPS/Bps 中的任何一个参数，则 qosName 不能与已存在的 QoS 策略重名。 - 如果指定了 qosName，没有设置 IOPS/Bps 中的任何一个参数，则 qosName 需要为已存在的 QoS 策略名称。	
IOPS	Long	每秒能够进行读写操作次数的最大值。 取值：取值范围为[-1, 999999999]，默认值为-1。-1 表示不限制。	否
readIOPS	Long	每秒能够进行读操作次数的最大值。 取值：取值范围为[-1, 999999999]，默认值为-1。-1 表示不限制。	否
writeIOPS	Long	每秒能够进行写操作次数的最大值。 取值：取值范围为[-1, 999999999]，默认值为-1。-1 表示不限制。	否

Bps	Long	每秒可传输数据量的最大值。 取值：取值范围为[-1, 4096000000000], 默认值为-1，单位是 B/s。-1 表示不限制。	否
readBps	Long	读带宽上限。 取值：取值范围为[-1, 4096000000000], 默认值为-1，单位是 B/s。-1 表示不限制。	否
writeBps	Long	写带宽上限。 取值：取值范围为[-1, 4096000000000], 默认值为-1，单位是 B/s。-1 表示不限制。	否
IOPSBurst	Long	使用 Burst 功能时，每秒能够进行读写操作次数的最大值。 取值：只有当 IOPS 大于等于 1 时，此项设置为 -1 或(IOPS, 999999999]内的正整数方可生效。默认值为-1，表示不限制。	否
readIOPSBurst	Long	使用 Burst 功能时，每秒能够进行读操作次数的最大值。 取值：只有当 readIOPS 大于等于 1 时，此项设置为-1 或(readIOPS, 999999999]内的正整数方可生效。默认值为-1，表示不限制。	否
writeIOPSBurst	Long	使用 Burst 功能时，每秒能够进行写操作次数的最大值。 取值：只有当 writeIOPS 大于等于 1 时，此项设置为-1 或(writeIOPS, 999999999]内的正整数方可生效。默认值为-1，表示不限制。	否
BpsBurst	Long	使用 Burst 功能时，每秒可传输的数据量最大值。 取值：只有当 Bps 大于等于 1 时，此项设置为-1	否

		或(<i>Bps</i> , 4096000000000]内的正整数方可生效，默认值为-1，单位是 B/s。-1 表示不限制。	
<code>readBpsBurst</code>	Long	使用 Burst 功能时，读带宽上限。 取值：只有当 <code>readBps</code> 大于等于 1 时，此项设置为-1 或(<i>readBps</i> , 4096000000000]内的正整数方可生效，默认值为-1，单位是 B/s。-1 表示不限制。	否
<code>writeBpsBurst</code>	Long	使用 Burst 功能时，写带宽上限。 取值：只有当 <code>writeBps</code> 大于等于 1 时，此项设置为-1 或(<i>writeBps</i> , 4096000000000]内的正整数方可生效，默认值为-1，单位是 B/s。-1 表示不限制。	否
<code>IOPSBurstSecs</code>	Long	使用 Burst 功能时，按照 Burst 上限的能力进行读写操作所能持续的时间。 注意： 只有在 IOPS Burst 功能启用时，此配置才生效。 取值：取值范围为[1, 999999999]，默认值为 1，单位是秒。	否
<code>readIOPSBurstSecs</code>	Long	使用 Burst 功能时，按照 Burst 上限的能力进行读操作所能持续的时间。 注意： 只有在 read IOPS Burst 功能启用时，此配置才生效。 取值：取值范围为[1, 999999999]，默认值为 1，单位是秒。	否
<code>writeIOPSBurstSecs</code>	Long	使用 Burst 功能时，按照 Burst 上限的能力进行写操作所能持续的时间。 注意： 只有在 write IOPS Burst 功能启用时，此	否

		配置才生效。 取值：取值范围为[1, 999999999]，默认值为1，单位是秒。	
BpsBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的流量能力所能持续的时间。 注意： 只有在 Bps Burst 功能启用时，此配置才生效。 取值：取值范围为[1, 999999999]，默认值为1，单位是秒。	否
readBpsBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的读流量能力所能持续的时间。 注意： 只有在 read Bps Burst 功能启用时，此配置才生效。 取值：取值范围为[1, 999999999]，默认值为1，单位是秒。	否
writeBpsBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的写流量能力所能持续的时间。 注意： 只有在 write Bps Burst 功能启用时，此配置才生效。 取值：取值范围为[1, 999999999]，默认值为1，单位是秒。	否

响应结果

名称	类型	描述
iSCSITargets	Array of iSCSItarget	iSCSI target 属性集合，详见“表 3 响应参数 iSCSITarget 说明”。
cloneDepth	Integer	克隆卷的深度。

表 3 响应参数 iSCSITarget 说明

名称	类型	描述
iqn	String	iSCSI target IQN。
status	String	克隆卷对应的 target 的状态： ● Active: 主 target。 ● Standby: 热备 target。 ● Unavailable: 冷备 target。
ips	Array of ip	iSCSI target 的 IP 属性集合，详见“表 4 响应参数 ip 说明”。
serverId	String	服务器 ID。

表 4 响应参数 ip 说明

名称	类型	描述
ip	String	iSCSI target 的 IP。
port	Integer	iSCSI target 的端口号。

请求示例 1

单机版：创建克隆卷 luna1-C2。

```
POST /rest/v1/block/lun/clone HTTP/1.1
Date: Tue, 11 Mar 2025 06:55:43 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Connection: keep-alive
Content-Length: 211
Host:192.168.0.66:1443

{
  "lunName": "luna1-C2",
  "snapshotName": "luna1-snap4",
  "targetName": "targeta",
```



```
"capacity": 500,  
"config": {  
  "sectorSize": 512,  
  "writePolicy": "WriteThrough"  
}  
}
```

响应示例 1

```
HTTP/1.1 200 OK  
x-hblock-request-id: 1e3c374dc4044830828d69d3657d996a  
Connection: keep-alive  
Content-Length: 172  
Date: Tue, 11 Mar 2025 06:55:43 GMT  
Content-Type: application/json; charset=utf-8  
Server: HBlock  
  
{  
  "data": {  
    "iSCSITargets": [  
      {  
        "iqn": "iqn.2012-08.cn.ctyunapi.oos:targeta.1",  
        "status": "Active",  
        "ips": [  
          {  
            "ip": "192.168.0.66",  
            "port": 3260  
          }  
        ],  
        "serverId": "hblock_1"  
      }  
    ],  
    "cloneDepth": 1  
  }  
}
```

请求示例 2

集群版：创建克隆卷 lun02-clone-6。

```
POST /rest/v1/block/lun/clone HTTP/1.1
Date: Tue, 11 Mar 2025 06:41:03 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Connection: keep-alive
Content-Length: 559
Host:192.168.0.64:1443
```

```
{
  "lunName": "lun02-clone-6",
  "snapshotName": "lun02-snap20250304170942",
  "targetName": "target02",
  "capacity": 500,
  "config": {
    "localStorageClass": "EC 2+1",
    "ECfragmentSize": 1024,
    "sectorSize": 512,
    "cachePool": "default",
    "pool": "pool2",
    "highAvailability": "ActiveStandby",
    "writePolicy": "WriteThrough",
    "serverAffinity": {
      "autoFailback": "Enabled",
      "priority": [
        "hblock_1"
      ]
    }
  }
}
```

响应示例 2

```
HTTP/1.1 200 OK
```

```
x-hblock-request-id: 379a264ec280457ea92a44ba3e0413eb
Connection: keep-alive
Content-Length: 314
Date: Tue, 11 Mar 2025 06:41:03 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "iSCSITargets": [
      {
        "iqn": "iqn.2012-08.cn.ctyunapi.oos:target02.3",
        "status": "Unavailable",
        "ips": [
          {
            "ip": "192.168.0.65",
            "port": 3260
          }
        ],
        "serverId": "hblock_1"
      },
      {
        "iqn": "iqn.2012-08.cn.ctyunapi.oos:target02.4",
        "status": "Unavailable",
        "ips": [
          {
            "ip": "192.168.0.64",
            "port": 3260
          }
        ],
        "serverId": "hblock_2"
      }
    ],
    "cloneDepth": 1
  }
}
```

```
}
```

4.3.3 断开克隆卷与快照的关系链

此操作用来断开克隆卷与快照的关系链。该操作会将快照中存储的数据复制到克隆中，断开关系链后该卷将成为一个独立的卷。

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。

请求语法

```
PUT /rest/v1/block/lun/LunName/flatten HTTP/1.1
```

Date: *date*

Host: *ip:port*

Authorization: *authorization*

请求参数

参数	类型	描述	是否必须
lunName	String	要断开的克隆卷名称。 取值：1~16 位字符串，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。	是

请求示例

断开克隆卷 **luna1-C2** 与快照的关系链。

```
PUT /rest/v1/block/lun/luna1-C2/flatten HTTP/1.1
```

Date: Tue, 11 Mar 2025 07:38:50 GMT

Authorization: HBlock *userName:signature*

```
Connection: keep-alive
Content-Length: 0
Host: 192.168.0.66:1443
```

响应示例

```
HTTP/1.1 204 No Content
x-hblock-request-id: 10263b8586af4cdd8dfc6863032d7d66
Connection: keep-alive
Date: Tue, 11 Mar 2025 07:38:50 GMT
Server: HBlock
```

4.3.4 删除卷

此操作用来删除卷。

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。

请求语法

● 本地卷

```
DELETE /rest/v1/block/lun/LunName?force=force HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

● 上云卷

```
DELETE /rest/v1/block/lun/LunName?force=force&deleteCloudData=deleteCloudData HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

请求参数

参数	类型	描述	是否必须
lunName	String	要删除的卷名称。	是

		取值：1~16 位字符串，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。	
force	Boolean	是否强制删除卷。 注意：强制删除卷，会产生数据残留风险，请谨慎操作。 取值： ● true：强制删除卷。 ● false：不强制删除卷。 默认值为 false。	否
deleteCloudData	Boolean	删除卷时，是否删除云上的数据。 取值： ● true：删除卷时，删除云上数据。 ● false：删除卷时，不删除云上数据。 默认值为 false。 说明： ● 如果删除 Local 模式的卷时携带此参数，会直接忽略。 ● 如果卷状态为 Suspended、Suspending 或 SuspendFailed，即使强制删除卷，也不能删除云上数据。	否

请求示例 1

本地卷：删除卷名称为 testLun1 的卷。

```
DELETE /rest/v1/block/lun/testLun1 HTTP/1.1
Date: Tue, 15 Mar 2022 02:40:17 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.121:1443
```


响应示例 1

```
HTTP/1.1 202 Accepted
x-hblock-request-id: 75dd09b8f97941d98b74ac15b4b5c855
Connection: keep-alive
Content-Length: 0
Date: Tue, 15 Mar 2022 02:40:17 GMT
Server: HBlock
```

请求示例 2

上云卷：删除卷名称为 **lun03a** 的卷，删除卷时删除云上数据。

```
DELETE /rest/v1/block/lun/lun03a?deleteCloudData=true HTTP/1.1
Date: Thu, 9 May 2024 08:26:41 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.192:1443
```

响应示例 2

```
HTTP/1.1 202 Accepted
x-hblock-request-id: 460b71d05c4c4b119811cb4b8f6ddc5b
Connection: keep-alive
Content-Length: 0
Date: Thu, 9 May 2024 08:26:41 GMT
Server: HBlock
```

4.3.5 扩容卷

此操作用来扩容卷。

注意：卷扩容，除了在服务器端进行扩容外，还需要在客户端进行操作，详细操作可以参考命令行手册“4.6.5 扩容卷”章节。

请求语法

```
PUT /rest/v1/block/lun/LunName/expand HTTP/1.1
Date: date
Host: ip:port
Authorization:authorization

{
  "capacity": capacity
}
```

请求参数

参数	类型	描述	是否必须
lunName	String	卷名称。 取值：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。	是
capacity	Integer	卷要扩容后的容量。 取值：[1,1048576]，单位是 GiB。	是

请求示例

将卷 `lunName1` 扩容至 200000 GiB。

```
PUT /rest/v1/block/lun/lunName1/expand HTTP/1.1
Date: Fri, 29 Jul 2022 06:43:11 GMT
Content-Type: application/json; charset=utf-8
```

```
Authorization: HBlock userName:signature
Content-Length: 26
Host: 192.168.0.121:1443

{
  "capacity":200000
}
```

响应示例

```
HTTP/1.1 204 No Content
x-hblock-request-id: 49ed1b92c7e74137ae8d5572cb1a0aa1
Connection: keep-alive
Date: Fri, 29 Jul 2022 06:43:11 GMT
Server: HBlock
```

4.3.6 修改单个卷

此操作用来修改单个卷的配置。

说明：卷处于“删除中”、“回滚中”，不能修改卷的配置。

请求语法

- 本地卷

```
PUT /rest/v1/block/lun/LunName HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization

{
  "config": {
    "minReplica": minReplica,
    "redundancyOverlap": redundancyOverlap,
    "writePolicy": writePolicy,
    "serverAffinity": {
      "autoFailback": autoFailback,
      "priority": [
        SeverID,ServerID...
      ]
    }
  }
}
```

- 上云卷

```
PUT /rest/v1/block/lun/LunName HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization

{
  "config": {
    "minReplica": minReplica,
```

```
{
  "redundancyOverlap": redundancyOverlap,
  "writePolicy": writePolicy,
  "serverAffinity": {
    "autoFailback": autoFailback,
    "priority": [
      SeverID,ServerID...
    ]
  },
},
"cloud": {
  "compression": compression,
  "accessKey": accessKey,
  "secretKey": secretKey,
  "endpoint": endpoint,
  "signVersion": signVersion,
  "region": region
}
}
```

请求参数

参数	类型	描述	是否必须
lunName	String	卷名称。 取值：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。	是
config.minReplica	Integer	最小副本数（仅集群版支持）。 对于副本模式的卷，假设卷副本数为 X，最小副本数为 Y（Y 必须 ≤X），该卷每次写入时，至少 Y 份	否

		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：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$[N, N+M]$。	
config.redundancyOverlap	Integer	修改卷的折叠副本数（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 <code>path</code> 上。 注意：如果存储池故障域级别为 <code>path</code>，此参数不生效。 取值：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$[1, N+M]$。	否
config.writePolicy	String	卷的写策略。 取值： ● 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。适用于对性能要求较高，稳定性	否

		要求不高的场景。 ● 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。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: 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。	
<code>config.serverAffinity</code>	Object	卷主备分布优先级设置，详见“表 1 请求参数 config.serverAffinity 说明”。	否
<code>cloud.compression</code>	String	是否压缩数据上传至云端。 取值： ● Enabled: 压缩数据上传至云端。 ● Disabled: 不压缩数据上传至云端。	否
<code>cloud.accessKey</code>	String	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。	否
<code>cloud.secretKey</code>	String	修改对象存储服务的 Secret Access	否

		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。	
cloud.endpoint	String	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。	否
cloud.signVersion	String	指定上云签名认证的类型。 取值： - v2：V2 签名认证。 - v4：V4 签名认证。	否
cloud.region	String	表示 Endpoint 资源池所在区域。 V4 签名时，此项必填。	否

表 1 请求参数 config.serverAffinity 说明

参数	类型	描述	是否必须
autoFailback	String	修改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。	否

		取值： ● Enabled: 自动进行主备切换。 ● Disabled: 不自动进行主备切换。	
priority	Array of priority	修改卷主备分布优先级的服务器 ID（仅集群版支持），系统根据指定的服务器 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： 如果为空数组，即[]，表示清空优先级设置。 取值：服务器 ID。	否

请求示例 1

修改卷 lun8 的写策略为 WriteThrough。

```
PUT /rest/v1/block/lun/lun8 HTTP/1.1
Date: Fri, 29 Jul 2022 07:29:17 GMT
Host: 192.168.0.121:1443
Content-Length: 53
Authorization: HBlock userName:signature

{
  "config":{
    "writePolicy":"WriteThrough"
  }
}
```

响应示例 1

```
HTTP/1.1 204 No Content
Date: Fri, 29 Jul 2022 07:29:18 GMT
Connection: keep-alive
x-hblock-request-id: 140a25c99a9a4efc87e561b44b595926
Server: HBlock
```

请求示例 2

修改卷的最小副本数为 3。

```
PUT /rest/v1/block/lun/lun01a HTTP/1.1
Date: Sat, 7 Oct 2023 07:58:35 GMT
Host: 192.168.0.117:1443
Content-Length: 53
Authorization: HBlock userName:signature

{
  "config": {
    "minReplica": 3
  }
}
```

响应示例 2

```
HTTP/1.1 204 No Content
Date: Sat, 7 Oct 2023 07:58:35 GMT
Connection: keep-alive
x-hblock-request-id: 437f52eff9fe44a6ad9aae796beda55e
Server: HBlock
```

请求示例 3

修改卷 lun02a 的上云信息。

```
PUT /rest/v1/block/lun/lun02a HTTP/1.1
Date: Thu, 9 May 2024 09:12:47 GMT
Host: 192.168.0.110:1443
Content-Length: 271
Authorization: HBlock userName:signature

{
  "cloud": {
    "accessKey": "d0434d3e9e8dd462d6e0",
    "secretKey": "*****",
    "endpoint": "oos-cn.ctyunapi.cn",
```

```
    "signVersion": "v4",  
    "region": "cn",  
    "compression": "Enabled"  
  }  
}
```

响应示例 3

```
HTTP/1.1 204 No Content  
Date: Thu, 9 May 2024 09:12:47 GMT  
Connection: keep-alive  
x-hblock-request-id: 01d8a27085444f6aaabbd3ebd16348db  
Server: HBlock
```

4.3.7 设置卷的主备优先级或自动切换（集群版适用）

此操作用来批量修改卷的配置。

说明：卷处于“删除中”、“回滚中”，不能修改卷的配置。

请求语法

```
PUT /rest/v1/block/lun HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization

{
  "action": action,
  "lunList": [
    LunName, LunName...
  ],
  "autoFailback": autoFailback,
  "priority": [
    SeverID, SeverID...
  ],
  "statusDriven": statusDriven
}
```

请求参数

参数	类型	描述	是否必须
action	String	指定卷的操作内容。 取值： ● setServerAffinity：对卷的主备优先级进行设置。	是

		● switch: 对卷进行主备切换。	
lunList	Array of lun	需要修改的卷列表。 取值: 卷名称。 说明: 如果此参数指定为空, 即"lunList": [], 或者不指定此参数, 表示批量修改所有的卷。	否
autoFailback	String	修改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。 注意: action 取值为 setServerAffinity 时生效。 取值: ● Enabled: 自动进行主备切换。 ● Disabled: 不自动进行主备切换。	否
priority	Array of priority	修改卷主备分布优先级的服务器 ID, 系统根据指定的服务器 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: action 取值为 setServerAffinity 时生效, 且 priority 和 statusDriven 互斥。如果为空数组, 即 [], 表示清空优先级设置。 取值: 如果卷列表中指定一个卷, priority 取值为服务器 ID 列表或者空数组; 如果卷列表中指定多个 (大于 1) 卷, 或者卷列表为空, 或者不指定卷列表, priority 取值只能为空数组, 或者不指定 priority。	否
statusDriven	Boolean	是否基于卷目前的主备状态进行优先级的设置, 即根据卷的当前主备 target (Active target、Standby target、Cold target) 所在的服务器, 设置卷主备分布优先级的服务器 ID。 注意: action 取值为 setServerAffinity 时生效。	否

		取值： ● true: 基于目前的主备状态进行优先级的设置。 ● false: 不做任何变化。	

请求示例 1

清空卷 lun01a、lun02a 的优先级设置。

```
PUT /rest/v1/block/lun HTTP/1.1
Date: Wed, 15 Jan 2025 09:27:11 GMT
Host: 192.168.0.192:1443
Content-Length: 123
Authorization: HBlock userName:signature

{
  "action": "setServerAffinity",
  "lunList": [
    "lun01a",
    "lun02a"
  ],
  "priority": []
}
```

响应示例 1

```
HTTP/1.1 202 Accepted
Date: Wed, 15 Jan 2025 09:27:11 GMT
Connection: keep-alive
x-hblock-request-id: bdb3c7eb8eac4342b161364f5034c3e9
Content-Length: 66
Server: HBlock

{
  "data": {
    "successLunList": [
      "lun01a",
```

```
        "lun02a"  
    ],  
    "failedLunList": []  
}  
}
```

请求示例 2

对 lun01a、lun02a、lun03a 进行主备切换。

```
PUT /rest/v1/block/lun HTTP/1.1  
Date: Wed, 15 Jan 2025 09:36:44 GMT  
Host: 192.168.0.192:1443  
Content-Length: 110  
Authorization: HBlock userName:signature  
  
{  
    "action": "switch",  
    "lunList": [  
        "lun01a",  
        "lun02a",  
        "lun03a"  
    ]  
}
```

响应示例 2

```
HTTP/1.1 202 Accepted  
Date: Wed, 15 Jan 2025 09:36:44 GMT  
Connection: keep-alive  
x-hblock-request-id: 4e48e27a0db4495ca60b673ffd2cb3d5  
Content-Length: 75  
Server: HBlock  
  
{  
    "data": {  
        "successLunList": [  

```

```
        "lun01a",  
        "lun02a"  
    ],  
    "failedLunList": []  
}  
}
```


4.3.8 触发卷对应 target 的主备切换（集群版适用）

此操作用来触发卷对应 target 的主备切换。

注意：执行此操作后，客户端不需要进行任何操作。

请求语法

```
PUT /rest/v1/block/lun/LunName/switch HTTP/1.1
```

Date: *date*

Host: *ip:port*

Authorization: *authorization*

请求参数

参数	类型	描述	是否必须
lunName	String	卷名称。 取值：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。	是

响应结果

名称	类型	描述
iSCSITargets	Array of iSCSITarget	iSCSI target 属性集合，详见“表 1 响应参数 iSCSITarget 说明”。

表 1 响应参数 iSCSITarget 说明

名称	类型	描述
iqn	String	target IQN。
status	String	卷对应的 target 的状态： ● Active: 主 target。

		● Standby: 热备 target。 ● Unavailable: 冷备 target。
ips	Array of ip	iSCSI target 的 IP 属性集合，详见“表 2 响应参数 ip 说明”。

表 2 响应参数 ip 说明

名称	类型	描述
ip	String	iSCSI target 的 IP。
port	Integer	iSCSI target 的端口号。

请求示例

对卷 lun01a 进行切换。

```
PUT /rest/v1/block/lun/lun01a/switch HTTP/1.1
Date: Fri, 10 Feb 2023 09:42:01 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.117:1443
```

响应示例

```
HTTP/1.1 200 OK
x-hblock-request-id: 10d4e743ef08473fbe41cc61cd933844
Connection: keep-alive
Content-Length: 296
Date: Fri, 10 Feb 2023 09:42:01 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "iSCSITargets": [
      {
        "iqn": "iqn.2012-08.cn.ctyunapi.oos:target01.3",
```

```
        "status": "Unavailable",
        "ips": [
            {
                "ip": "192.168.0.117",
                "port": 3260
            }
        ],
        "serverId": "hblock_1"
    },
    {
        "iqn": "iqn.2012-08.cn.ctyunapi.oos:target01.4",
        "status": "Active",
        "ips": [
            {
                "ip": "192.168.0.192",
                "port": 3260
            }
        ],
        "serverId": "hblock_3"
    }
]
}
```

4.3.9 还原卷（上云卷适用）

此操作用来还原 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）。
- 还原的卷名称在当前 HBlock 中不存在。
- 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。
- 在单机版还原卷之前，需确保其对应的数据目录中无该卷的残留数据。
- 卷处于 Recovering、RecoverFailed 状态时，不支持读写。

请求语法

- 单机版

```
POST /rest/v1/block/lun/recover HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "lunName": LunName,
  "uuid": Lunuuid,
  "targetName": targetName,
```

```
"storageMode": storageMode,
"config": {
  "sectorSize": sectorSize,
  "writePolicy": writePolicy,
  "qosPolicy": {
    "qosName": qosName,
    "IOPS": IOPS,
    "readIOPS": readIOPS,
    "writeIOPS": writeIOPS,
    "Bps": Bps,
    "readBps": readBps,
    "writeBps": writeBps,
    "IOPSBurst": IOPSBurst,
    "readIOPSBurst": readIOPSBurst,
    "writeIOPSBurst": writeIOPSBurst,
    "BpsBurst": BpsBurst,
    "readBpsBurst": readBpsBurst,
    "writeBpsBurst": writeBpsBurst,
    "IOPSBurstSecs": IOPSBurstSecs,
    "readIOPSBurstSecs": readIOPSBurstSecs,
    "writeIOPSBurstSecs": writeIOPSBurstSecs,
    "BpsBurstSecs": BpsBurstSecs,
    "readBpsBurstSecs": readBpsBurstSecs,
    "writeBpsBurstSecs": writeBpsBurstSecs
  },
  "path": path
},
"cloud": {
  "provider": provider,
  "bucketName": bucketName,
  "prefix": prefix,
  "accessKey": accessKey,
  "secretKey": secretKey,
  "endpoint": endpoint,
  "signVersion": signVersion,
  "region": region,
  "storageClass": storageClass,
  "compression": compression
```

```
}  
}
```

● 集群版

POST /rest/v1/block/lun/recover HTTP/1.1

Date: *date*

Content-Type: application/json; charset=utf-8

Content-Length: *length*

Host: *ip:port*

Authorization: *authorization*

```
{  
  "lunName": LunName,  
  "uuid": Lunuuid,  
  "targetName": targetName,  
  "storageMode": storageMode,  
  "config": {  
    "localStorageClass": LocalStorageClass,  
    "minReplica": minReplica,  
    "redundancyOverlap": redundancyOverlap,  
    "ECfragmentSize": ECfragmentSize,  
    "cachePool": poolName,  
    "pool": poolName,  
    "highAvailability": highAvailability,  
    "writePolicy": writePolicy,  
    "serverAffinity": {  
      "autoFailback": autoFailback,  
      "priority": [  
        SeverID,ServerID...  
      ]  
    },  
    "qosPolicy": {  
      "qosName": qosName,  
      "IOPS": IOPS,  
      "readIOPS": readIOPS,
```

```
        "writeIOPS": writeIOPS,
        "Bps": Bps,
        "readBps": readBps,
        "writeBps": writeBps,
        "IOPSBurst": IOPSBurst,
        "readIOPSBurst": readIOPSBurst,
        "writeIOPSBurst": writeIOPSBurst,
        "BpsBurst": BpsBurst,
        "readBpsBurst": readBpsBurst,
        "writeBpsBurst": writeBpsBurst,
        "IOPSBurstSecs": IOPSBurstSecs,
        "readIOPSBurstSecs": readIOPSBurstSecs,
        "writeIOPSBurstSecs": writeIOPSBurstSecs,
        "BpsBurstSecs": BpsBurstSecs,
        "readBpsBurstSecs": readBpsBurstSecs,
        "writeBpsBurstSecs": writeBpsBurstSecs
    }
},
"cloud": {
    "provider": provider,
    "bucketName": bucketName,
    "prefix": prefix,
    "accessKey": accessKey,
    "secretKey": secretKey,
    "endpoint": endpoint,
    "signVersion": signVersion,
    "region": region,
    "storageClass": storageClass,
    "compression": compression
}
}
```

请求参数

参数	类型	描述	是否必须
lunName	String	源卷的卷名称。	是

uuid	String	源卷的唯一标识码。	是
targetName	String	指定还原卷的 target 名称。 说明：还原卷时，如果指定的 target 名称不存在，那么同时创建 iSCSI target。 取值：长度范围是 1~16，可以由小写字母、数字、句点 (.) 和短横线 (-) 组成，且仅支持以字母或数字开头。	是
storageMode	String	指定还原卷的存储类型。 取值： ● Cache：缓存模式，本地保留部分热数据，全部数据异步存储到云端。 ● Storage：存储模式，本地保留全部数据，并异步存储到云端。 存储类型默认为源卷的存储类型。	否
config.localStorageClass	String	指定还原卷冗余模式（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，$N \geq M$，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：	否

		■ 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。	
<code>config.minReplica</code>	Integer	指定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：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。	否
<code>config.redundancyOverlap</code>	Integer	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 <code>path</code> 上。 注意：如果存储池故障域级别为 <code>path</code>，此	否

		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：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。	
<code>config.ECfragmentSize</code>	Integer	设置还原卷的纠删码模式分片大小（仅集群版支持）。 <code>config.localStorageClass</code> 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。	否
<code>config.cachePool</code>	String	指定还原卷的缓存存储池（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。 取值：长度范围是 1~16，只能由字母、数字和短横线（-）、下划线（_）组成，字母区分大小写，且仅支持以字母和数字开头。	否
<code>config.pool</code>	String	指定还原卷的存储池（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。	否

		取值：长度范围是 1~16，只能由字母、数字和短横线（-）、下划线（_）组成，字母区分大小写，且仅支持以字母和数字开头。 说明：不指定，默认使用基础存储池。	
config.highAvailability	String	指定还原卷的高可用类型（仅集群版支持）。 取值： ● ActiveStandby：启用主备，该卷关联对应 target 下的所有 IQN。 ● Disabled：不启用主备，该卷关联对应 target 下的 1 个 IQN。 默认值为源卷的高可用类型。	否
config.writePolicy	String	设置还原卷的写策略。 取值： ● 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。适用于对性能要求较高，稳定性要求不高的场景。 ● 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。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：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	否

		少的场景。 默认值为源卷的写策略。	
<code>config.serverAffinity</code>	Object	卷主备分布优先级设置（仅集群版支持），详见“表 1 请求参数 serverAffinity 说明（仅集群版支持）”。	否
<code>config.qosPolicy</code>	Object	卷关联的 QoS 策略信息，详见“表 2 请求参数 config.qosPolicy 说明”。	否
<code>config.path</code>	String	指定还原卷存储数据的数据目录（仅单机版支持）。 取值：只能包含字母、数字、汉字和特殊字符(~!@\$()_+-.;.:)。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。	否
<code>cloud.provider</code>	String	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。	否
<code>cloud.bucketName</code>	String	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。	是
<code>cloud.prefix</code>	String	源卷的前缀名称。如果源卷未指定前缀，则此处不设置。 取值：长度范围是 1~256。	否
<code>cloud.accessKey</code>	String	源卷的 Access Key。	是

cloud.secretKey	String	源卷的 Secret Access Key。	是
cloud.endpoint	String	源卷的 Endpoint。 注意： 如果仅输入域名将会使用 HTTPS 协议进行访问。	是
cloud.signVersion	String	指定还原卷上云签名认证的类型。 取值： ● v2：V2 签名认证。 ● v4：V4 签名认证。 默认值为 v2。	否
cloud.region	String	还原卷的 region，即 Endpoint 资源池所在区域。 V4 签名时，此项必填。	否
cloud.storageClass	String	设置还原卷上传数据至云端的存储类型。 取值： ● STANDARD：标准存储。 ● STANDARD_IA：低频访问存储。 默认为源卷上传数据至云端的存储类型。	否
cloud.compression	String	设置还原卷是否压缩数据上传至云端。 取值： ● Enabled：压缩数据上传至云端。 ● Disabled：不压缩数据上传至云端。 默认值为源卷的设置。	否

表 1 请求参数 serverAffinity（仅集群版支持）说明

参数	类型	描述	是否必须
autoFailback	String	是否根据指定的卷主备分布优先级自动进行主备切	否

		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。	
priority	Array of string	指定卷主备分布优先级的服务器 ID（仅集群版支持），系统根据指定的服务器 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。 取值：服务器 ID。	否

表 2 请求参数 config.qosPolicy 说明

参数	类型	描述	是否必须
qosName	String	指定 QoS 策略名称。 说明：创建卷时，如果指定 QoS 策略名称不存在，那么同时创建 QoS 策略，新创建的 QoS 策略的回收策略默认为 Delete。 对于集群版，不指定 QoS 策略时： ● 如果卷的缓存存储池和存储池同时存在，卷的 QoS 策略为缓存存储池设置的“存储池内卷的默认 QoS 策略”；如果缓存存储池未设置“存储池内卷的默认 QoS 策略”，则卷无	否

		QoS 策略。 ● 如果卷只有存储池，卷的 QoS 策略为存储池设置的“存储池内卷的默认 QoS 策略”；如果存储池未设置“存储池内卷的默认 QoS 策略”，则卷无 QoS 策略。 取值：长度范围 1~64，只能由字母、数字和短横线(-)组成，区分大小写，且仅支持以字母或数字开头。 注意： ● 如果没有指定 qosName，但是设置了 IOPS/Bps 中的任何一个参数，则根据 IOPS/Bps 自动创建一个 QoS 策略，系统为策略名赋值：lun-lunname-qos-timestamp。 ● 如果指定了 qosName 以及 IOPS/Bps 中的任何一个参数，则 qosName 不能与已存在的 QoS 策略重名。 ● 如果指定了 qosName，没有设置 IOPS/Bps 中的任何一个参数，则 qosName 需要为已存在的 QoS 策略名称。	
IOPS	Long	每秒能够进行读写操作次数的最大值。 取值：取值范围为[-1, 999999999]，默认值为-1。-1 表示不限制。	否
readIOPS	Long	每秒能够进行读操作次数的最大值。 取值：取值范围为[-1, 999999999]，默认值为-1。-1 表示不限制。	否
writeIOPS	Long	每秒能够进行写操作次数的最大值。 取值：取值范围为[-1, 999999999]，默认值为-	否

		1。-1 表示不限制。	
BPS	Long	每秒可传输数据量的最大值。 取值：取值范围为[-1, 4096000000000]，默认值为-1，单位是 B/s。-1 表示不限制。	否
readBps	Long	读带宽上限。 取值：取值范围为[-1, 4096000000000]，默认值为-1，单位是 B/s。-1 表示不限制。	否
writeBps	Long	写带宽上限。 取值：取值范围为[-1, 4096000000000]，默认值为-1，单位是 B/s。-1 表示不限制。	否
IOPSBurst	Long	使用 Burst 功能时，每秒能够进行读写操作次数的最大值。 取值：只有当 IOPS 大于等于 1 时，此项设置为 -1 或(IOPS, 999999999]内的正整数方可生效。默认值为-1，表示不限制。	否
readIOPSBurst	Long	使用 Burst 功能时，每秒能够进行读操作次数的最大值。 取值：只有当 readIOPS 大于等于 1 时，此项设置为-1 或(readIOPS, 999999999]内的正整数方可生效。默认值为-1，表示不限制。	否
writeIOPSBurst	Long	使用 Burst 功能时，每秒能够进行写操作次数的最大值。 取值：只有当 writeIOPS 大于等于 1 时，此项设置为-1 或(writeIOPS, 999999999]内的正整数方可生效。默认值为-1，表示不限制。	否
BpsBurst	Long	使用 Burst 功能时，每秒可传输的数据量最大值。	否

		取值：只有当 Bps 大于等于 1 时，此项设置为-1 或(<i>Bps</i> , 4096000000000]内的正整数方可生效，默认值为-1，单位是 B/s。-1 表示不限制。	
readBpsBurst	Long	使用 Burst 功能时，读带宽上限。 取值：只有当 readBps 大于等于 1 时，此项设置为-1 或(<i>readBps</i> , 4096000000000]内的正整数方可生效，默认值为-1，单位是 B/s。-1 表示不限制。	否
writeBpsBurst	Long	使用 Burst 功能时，写带宽上限。 取值：只有当 writeBps 大于等于 1 时，此项设置为-1 或(<i>writeBps</i> , 4096000000000]内的正整数方可生效，默认值为-1，单位是 B/s。-1 表示不限制。	否
IOPSBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的能力进行读写操作所能持续的时间。 注意： 只有在 IOPS Burst 功能启用时，此配置才生效。 取值：取值范围为[1, 999999999]，默认值为 1，单位是秒。	否
readIOPSBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的能力进行读操作所能持续的时间。 注意： 只有在 read IOPS Burst 功能启用时，此配置才生效。 取值：取值范围为[1, 999999999]，默认值为 1，单位是秒。	否
writeIOPSBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的能力进行写操作所能持续的时间。	否

		注意：只有在 write IOPS Burst 功能启用时，此配置才生效。 取值：取值范围为[1, 999999999]，默认值为1，单位是秒。	
BpsBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的流量能力所能持续的时间。 注意：只有在 Bps Burst 功能启用时，此配置才生效。 取值：取值范围为[1, 999999999]，默认值为1，单位是秒。	否
readBpsBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的读流量能力所能持续的时间。 注意：只有在 read Bps Burst 功能启用时，此配置才生效。 取值：取值范围为[1, 999999999]，默认值为1，单位是秒。	否
writeBpsBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的写流量能力所能持续的时间。 注意：只有在 write Bps Burst 功能启用时，此配置才生效。 取值：取值范围为[1, 999999999]，默认值为1，单位是秒。	否
reclaimPolicy	String	QoS 策略的回收策略。 取值： ● Delete: 解除 QoS 策略关联的所有对象或者删除 QoS 策略关联的所有对象，系统会触发删除 QoS 策略。	否

		● Retain: 解除 QoS 策略关联的所有对象或者删除 QoS 策略关联的所有对象，QoS 策略仍保留。 默认值为 Retain。	
description	String	QoS 策略的描述信息。 取值：1~256 位字符串。	否

请求示例 1

单机版：还原卷 luna9。

```
POST /rest/v1/block/lun HTTP/1.1
Date: Mon, 13 May 2024 10:46:42 GMT
Content-Type: application/json; charset=utf-8
Connection: keep-alive
Authorization: HBlock userName:signature
Content-Length: 516
Host:192.168.0.32:1443

{
  "lunName": "luna9",
  "uuid": "lun-uuid-df8308a9-bfb0-40a0-bf9d-7e741b1b6a63",
  "targetName": "targeta",
  "storageMode": "Storage",
  "config": {
    "writePolicy": "WriteThrough"
  },
  "cloud": {
    "bucketName": "hblocktest3",
    "prefix": "hblock1",
    "accessKey": "cb22b08b1f9229f85874",
    "secretKey": "*****",
    "endpoint": "oos-cn.ctyunapi.cn",
    "signVersion": "v4",
```

```
"region": "cn"
}
}
```

响应示例 1

```
HTTP/1.1 200 OK
x-hblock-request-id: 20b087c7711f44af9588cc013ede8654
Connection: keep-alive
Content-Length: 0
Date: Mon, 13 May 2024 10:46:42 GMT
Server: HBlock
```

请求示例 2

集群版：还原卷 lun08a。

```
POST /rest/v1/block/lun/recover HTTP/1.1
Date: Mon, 13 May 2024 09:33:12 GMT
Content-Type: application/json; charset=utf-8
Connection: keep-alive
Authorization: HBlock userName:signature
Content-Length: 629
Host:192.168.0.110:1443

{
  "lunName": "lun08a",
  "uuid": "lun-uuid-a74475f7-8d73-49b5-bd82-0ac0309e1fae",
  "targetName": "target08",
  "storageMode": "Storage",
  "config": {
    "localStorageClass": "EC 2+1",
    "minReplica": 2,
    "highAvailability": "ActiveStandby",
    "writePolicy": "WriteThrough"
```

```
    },  
    "cloud": {  
      "bucketName": "hblocktest3",  
      "prefix": "hblock",  
      "accessKey": "cb22b08b1f9229f85874",  
      "secretKey": "*****",  
      "endpoint": "oos-cn.ctyunapi.cn",  
      "signVersion": "v4",  
      "region": "cn"  
    }  
  }  
}
```

响应示例 2

```
HTTP/1.1 200 OK  
x-hblock-request-id: 251c5dcfab5247599aca4f60f5784b34  
Connection: keep-alive  
Content-Length: 0  
Date: Mon, 13 May 2024 09:33:12 GMT  
Server: HBlock
```

4.3.10 挂起卷（上云卷适用）

此操作用来挂起 HBlock 上云卷。

挂起卷适用于“多集群轮流写入同一上云卷”的场景，例如：A 集群把卷挂起后，B 集群立即原地还原并写入新数据；待 B 集群再次挂起，A 集群恢复卷即可直接看到 B 集群的最新写入，实现“一写一挂、交替接管”。

注意：

- 挂起后将立即冻结该卷的所有读写请求。请确保卷的所有数据已持久化，即如果卷已经被客户端挂载，需确保客户端的数据都已经同步到卷上。恢复前，依赖此卷的业务将不可用。
- 卷在挂起状态下，仅允许挂起、修改卷配置、修改上云配置、恢复、删除、查询。
- 卷处于 Suspended、Suspending、SuspendFailed 状态时，不支持读写。
- 挂起卷前须先 logout 断开客户端连接，否则在卷状态恢复 Normal 前将无法断开客户端连接。
- 仅卷处于 Normal、Suspended、Suspending、SuspendFailed 状态时，才可以执行此操作。

请求语法

```
PUT /rest/v1/block/lun/LunName/suspend HTTP/1.1
Date: date
Content-Length: length
Host: ip:port
Authorization:authorization

{
    "force": force
}
```

请求参数

参数	类型	描述	是否

			必须
lunName	String	指定需要挂起卷的名称。 取值：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。	是
force	Boolean	是否强制挂起卷。 注意： 强制挂起卷，会产生本地数据未被上传到云端的风险，请谨慎操作。 取值： ● true：强制挂起卷。 ● false：不强制挂起卷。 默认值为 false 。	否

请求示例

挂起卷 lunb1。

```
PUT /rest/v1/block/lun/lunb1/suspend HTTP/1.1
Date: Wed, 21 Jan 2026 06:01:23 GMT
Content-Length: 0
Connection: keep-alive
Authorization: HBlock userName:signature
Host: 192.168.0.64:1443
```

响应示例

```
HTTP/1.1 202 Accepted
x-hblock-request-id: 8804e05a56b5425087f83db16b85a021
Connection: keep-alive
Content-Length: 0
Date: Wed, 21 Jan 2026 06:01:23 GMT
Server: HBlock
```


4.3.11 恢复还原中断或挂起的卷（上云卷适用）

此操作用来恢复还原中断或挂起的 HBlock 上云卷。

注意：

- 仅卷处于 RecoverFailed、Suspended 状态下才可以执行此操作。
- 请确保本地数据目录中的卷数据文件未做任何修改，否则可能导致恢复后的卷数据不完整或卷无法正常使用。

请求语法

```
PUT /rest/v1/block/lun/LunName/resume-recovery HTTP/1.1
Date: date
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "syncMode": syncMode,
  "force": force
}
```

请求参数

参数	类型	描述	是否必须
lunName	String	指定需要继续还原或挂起的卷名称。 取值：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。	是
syncMode	String	指定缓存模式卷的恢复数据范围。 取值： ● MetaOnly：仅更新卷索引中与云端数据索引差异的部分。 ● Full：更新卷索引与云端差异部分，以及本地所存热数据	否

		与云端的差异部分。 默认值为 Full。 注意：此参数仅适用于缓存（cache）模式卷，如果本地没有数据，则不更新本地数据。	
force	Boolean	是否强制恢复卷。 注意：强制恢复，会产生部分本地数据丢失的风险，请谨慎操作。 取值： ● true：强制恢复卷。 ● false：不强制恢复卷。 默认值为 false。	否

请求示例

恢复卷 lun04a。

<pre>PUT /rest/v1/block/lun/lun04a/resume-recovery HTTP/1.1 Date: Wed, 31 Dec 2025 01:44:32 GMT Content-Length: 0 Connection: keep-alive Authorization: HBlock <i>userName:signature</i> Host: 192.168.0.192:1443</pre>

响应示例

<pre>HTTP/1.1 204 No Content x-hblock-request-id: f9c9a201db6a4104a57d9e3c27eea7e7 Connection: keep-alive Date: Wed, 31 Dec 2025 01:44:32 GMT Server: HBlock</pre>

4.3.12 清空卷（本地卷适用）

此操作用来清空卷。

注意：

- 如果克隆卷执行清空所有卷数据及卷对应的快照，会变成独立卷。
- 卷处于 Wiping、WipeFailed 状态时，不支持读写。
- 清空卷前须先 logout 断开客户端连接，否则在卷状态恢复 Normal 前将无法断开客户端连接。

请求语法

```
PUT /rest/v1/block/lun/LunName/wipe HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "scope": scope
}
```

请求参数

参数	类型	描述	是否必须
lunName	String	指定需要清空的卷的名称。 取值：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。	是
scope	String	指定清空卷的数据范围。 取值： ● All：清空所有卷数据及卷对应的快照。	否

		注意：如果卷存在关联的克隆卷，不能执行清空所有卷数据及卷对应快照的操作。 ● NoSnapshot：清空卷数据，保留卷对应的快照。后续可以使用卷快照进行卷回滚、或者导出备份。 注意：克隆卷不能执行仅清空卷数据保留对应快照的操作。 默认值为 All。	

请求示例 1

清空卷 lun01，包括卷数据及对应的快照。

```
PUT /rest/v1/block/lun/lun01/wipe HTTP/1.1
Date: Fri, 12 Dec 2025 03:31:37 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 25
Authorization: HBlock userName:signature
Host: 192.168.0.64:1443

{
  "scope": "All"
}
```

响应示例 1

```
HTTP/1.1 202 Accept
x-hblock-request-id: e6b663f013574507b542ce507528bde0
Connection: keep-alive
Date: Fri, 12 Dec 2025 03:31:37 GMT
Content-Length: 0
Server: HBlock
```

请求示例 2

清空卷 lun04b，清空卷数据，保留卷对应的快照。

```
PUT /rest/v1/block/lun/lun04b/wipe HTTP/1.1
Date: Fri, 12 Dec 2025 03:49:34 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 32
Authorization: HBlock userName:signature
Host: 192.168.0.64:1443

{
  "scope": "NoSnapshot"
}
```

响应示例 2

```
HTTP/1.1 202 Accept
x-hblock-request-id: 535e2361cb1c4dd1b7b8f72ba79db002
Connection: keep-alive
Date: Fri, 12 Dec 2025 03:49:34 GMT
Content-Length: 0
Server: HBlock
```

4.3.13 设置卷的扩展属性（本地卷适用）

此操作用来设置指定卷的扩展属性（Extended Attributes，简称 xattr）。

注意：

- 卷状态为 Deleting、DeleteFailed 状态时，不允许修改卷的 xattr。
- 卷无 Active IQN 时，不允许修改卷的 xattr。

请求语法

```
PUT /rest/v1/block/lun/LunName/xattr HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "action": "update",
  "xattrKey": xattrKey,
  "xattrValue": {
    xattrInnerKey1: xattrInnerKeyValue1,
    xattrInnerKey2: xattrInnerKeyValue2,
    .....
  }
}
```

请求参数

参数	类型	描述	是否必须
lunName	String	指定要设置扩展属性的卷名称。 取值：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。	是

action	String	操作类型。 取值：update：修改单个卷的扩展属性。即更新 xattrValue 中的 KV 对（innerKey 和 innerValue），不存在的 KV 对按照入参执行新增，存在的 KV 对按照入参执行更新。	是
xattrKey	String	指定卷的扩展属性的 key 值。 取值： ● OPENSTACK_META：定义卷的 Openstack 扩展属性。 ● CUSTOM：用户自定义卷的扩展属性。	是
xattrValue	Object	卷的扩展属性的 Value，为 json 对象格式，长度不超过 262144 bytes，详见“表 1 请求参数 xattrValue 说明”。	是

表 1 请求参数 xattrValue 说明

参数	类型	描述	是否必须
xattrInnerKey	String	xattrValue 中的 Key 值。 取值：长度范围是 1~500，只能由字母、数字、汉字和特殊字符(~!@\$()_+ - ; . :)组成，特殊符号需转义。	是
xattrInnerKeyValue	String	xattrValue 中 Key 的 Value 值。 取值：长度不超过 255 bytes 的字符串。	是

请求示例

设置卷 lun01a 的扩展属性。

```
PUT /rest/v1/block/lun/lun01a/xattr HTTP/1.1
```

```
Date: Fri, 19 Dec 2025 02:57:22 GMT
```

```
Content-Type: application/json; charset=utf-8
Content-Length: 279
Authorization: HBlock userName:signature
Host: 192.168.0.64:1443

{
  "action": "update",
  "xattrKey": "OPENSTACK_META",
  "xattrValue": {
    "5b157817-cccf-471d-bca3-5cb7598bb376": "{\"key1\":\"value1\",\"key2\":\"value2\"}",
    "27157817-cccf-471d-bca3-5cb7598bb389": "{\"key1\":\"value1\",\"key2\":\"value2\"}"
  }
}
```

响应示例 1

```
HTTP/1.1 204 No Content
x-hblock-request-id: e57fb247430d4711a711588cc5faa70d
Connection: keep-alive
Date: Fri, 19 Dec 2025 02:57:22 GMT
Server: HBlock
```


4.3.14 删除卷的指定扩展属性（本地卷适用）

此操作用来删除卷的指定扩展属性。

注意：

- 卷状态为 Deleting、DeleteFailed 状态时，不允许删除卷的 xattr。
- 卷无 Active IQN 时，不允许删除卷的 xattr。

请求语法

```
PUT /rest/v1/block/lun/LunName/xattr HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "action": "delete",
  "xattrKey": xattrKey,
  "xattrInnerKey": "xattrInnerKey1, xattrInnerKey2..."
}
```

请求参数

参数	类型	描述	是否必须
lunName	String	指定要删除扩展属性的卷名称。 取值：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。	是
action	String	操作类型。 取值：delete：删除单个卷指定的扩展属性。	是
xattrKey	String	指定卷的扩展属性的 key 值。	是

		取值： ● OPENSTACK_META: 定义卷的 Openstack 扩展属性。 ● CUSTOM: 用户自定义卷的扩展属性。	
xattrInnerKey	String	指定要删除 xattrValue 中的 Key 值，可以指定多个 Key 值，以英文逗号隔开。 说明 ：如果不指定此参数，则删除扩展属性 key 下的所有属性。 取值：长度范围是 1~500，只能由字母、数字、汉字和特殊字符(~!@\$()_+-.:;)组成，特殊符号需转义。	否

请求示例

删除卷 lun01a 中 xattrInnerKey 为 5b157817-cccf-471d-bca3-5cb7598bb376、27157817-cccf-471d-bca3-5cb7598bb389 扩展属性。

```
PUT /rest/v1/block/lun/lun01a/xattr HTTP/1.1
Date: Tue, 03 Mar 2026 02:17:39 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 162
Authorization: HBlock userName:signature
Host: 192.168.0.64:1443

{
  "action": "delete",
  "xattrKey": "OPENSTACK_META",
  "xattrInnerKey": "5b157817-cccf-471d-bca3-5cb7598bb376,27157817-cccf-471d-bca3-5cb7598bb389"
}
```

响应示例

```
HTTP/1.1 204 No Content
```

x-hblock-request-id: ff7f2e6f32c24ee68d1e838c263ac8c6

Connection: keep-alive

Date: Tue, 03 Mar 2026 02:17:39 GMT

Server: HBlock

4.3.15 删除卷的所有扩展属性（本地卷适用）

此操作用来删除指定卷的所有扩展属性。

请求语法

```
DELETE /rest/v1/block/lun/LunName/xattr HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

请求参数

参数	类型	描述	是否必须
lunName	String	要删除扩展属性的卷名称。 取值：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。	是

请求示例

删除卷 lun01a 的所有扩展属性。

```
DELETE /rest/v1/block/lun/lun01a/xattr HTTP/1.1
Date: Fri, 19 Dec 2025 08:52:07 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.64:1443
```

响应示例

```
HTTP/1.1 204 No Content
x-hblock-request-id: 6a0d6941ab09438a98c7918a22ae4195
Connection: keep-alive
Date: Fri, 19 Dec 2025 08:52:07 GMT
Server: HBlock
```

4.3.16 查询卷的扩展属性（本地卷适用）

此操作用来查询指定卷的扩展属性。

注意：如果返回信息内容整体过大（超过 2 MiB），仅返回 2 MiB 以内的数据。

请求语法

```
GET /rest/v1/block/lun/LunName/xattr?xattrKey=xattrKey&xattrInnerKey=xattrInnerKey HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

请求参数

参数	类型	描述	是否必须
lunName	String	卷名称。 取值：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。	是
xattrKey	String	指定卷的扩展属性的 key 值。 取值： ● OPENSTACK_META：定义卷的 Openstack 扩展属性。 ● CUSTOM：用户自定义卷的扩展属性。	否
xattrInnerKey	String	指定 xattrValue 中的 Key 值。可以指定多个 Key 值，以英文逗号隔开。如果不填写，则查询扩展属性 key 下的所有属性。 取值：长度范围是 1~500，只能由字母、数字、汉字和特殊字符(~!@\$()_+-;.:)组成，特殊符号需转义。	否

响应结果

名称	类型	描述
xattr	Object	卷的扩展属性，详见“表 1 响应参数 xattr 说明”。

表 1 响应参数 xattr 说明

名称	类型	描述
xattrKey	Object	卷的扩展属性的 key 值，详见“表 2 响应参数 xattrKey 说明”。

表 2 响应参数 xattrKey 说明

名称	类型	描述
xattrInnerKey	String	xattrValue 中的 Key 值。
xattrInnerKeyValue	String	xattrValue 中的 Key 的 Value 值。

请求示例 1

查询卷 lun01a 中 xattrKey 为 OPENSTACK_META 的扩展属性。

```
GET /rest/v1/block/lun/lun01a/xattr?xattrKey=OPENSTACK_META HTTP/1.1
Date: Tue, 3 Mar 2026 02:41:07 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.64:1443
```

响应示例 1

```
HTTP/1.1 200 OK
x-hblock-request-id: 440b18eb0f374be0830fe086ff307981
Connection: keep-alive
Content-Length: 371
Date: Tue, 3 Mar 2026 02:41:07 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock
```

```
{
  "data": {
    "xattr": {
      "OPENSTACK_META": {
        "5b157817-cccf-471d-bca3-5cb7598bb376": "{\"key1\":\"value1\",\"key2\":\"value2\"}",
        "27157817-cccf-471d-bca3-5cb7598bb389": "{\"key1\":\"value1\",\"key2\":\"value2\"}",
        "6b157817-cccf-471d-bca3-5cb7598bb376": "{\"key1\":\"value1\",\"key2\":\"value2\"}",
        "77157817-cccf-471d-bca3-5cb7598bb389": "{\"key1\":\"value1\",\"key2\":\"value2\"}"
      }
    }
  }
}
```

请求示例 2

查询卷 lun01a 的扩展属性，其中 `xattrKey` 为 `OPENSTACK_META`，且 `xattrInnerKey` 为 `5b157817-cccf-471d-bca3-5cb7598bb376` 和 `27157817-cccf-471d-bca3-5cb7598bb389`。

```
GET /rest/v1/block/lun/lun01a/xattr?xattrKey=OPENSTACK_META&xattrInnerKey=5b157817-cccf-471d-bca3-5cb7598bb376,27157817-cccf-471d-bca3-5cb7598bb389 HTTP/1.1
Date: Tue, 03 Mar 2026 02:37:36 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.64:1443
```

响应示例 2

```
HTTP/1.1 200 OK
x-hblock-request-id: a1e5b61f421844f9bbdafbd4a9d383f6
Connection: keep-alive
Content-Length: 205
Date: Tue, 03 Mar 2026 02:37:36 GMT
Content-Type: application/json; charset=utf-8
Server: HBlock
{
  "data": {
```

```
"xattr": {  
  "OPENSTACK_META": {  
    "5b157817-cccf-471d-bca3-5cb7598bb376": "{\"key1\":\"value1\",\"key2\":\"value2\"}",  
    "27157817-cccf-471d-bca3-5cb7598bb389": "{\"key1\":\"value1\",\"key2\":\"value2\"}"  
  }  
}  
}
```


4.3.17 批量查询卷

此操作用来批量查询卷。可以通过设置过滤条件和 `range` 范围进行查询。如果未设置过滤条件和 `range` 范围，则查询所有的卷。

请求语法

```
GET /rest/v2/block/lun?filter=filter&range=i-j HTTP/1.1
```

Date: *date*

Host: *ip:port*

Authorization: *authorization*

请求参数

参数	类型	描述	是否必须
filter	String	查询的过滤条件。 过滤条件由查询属性和属性值组成，其中属性和属性值之间用单个冒号分隔（<code>key:value</code>）表示模糊匹配，用两个冒号分隔（<code>key::value</code>）表示精确匹配。可以选择多个查询，如果是或的关系，使用“or”将查询条件分隔开；如果是与的关系，使用“and”将查询条件分隔开。 支持的过滤条件类型包括： ● <code>lunName</code>：卷名称。 ● <code>status</code>：卷的状态。 ● <code>capacity</code>：卷容量。 ● <code>iqn</code>：target 的识别。 ● <code>isClone</code>：是否是克隆卷，取值：<code>true</code>、<code>false</code>。 ● <code>pool</code>：存储池（仅集群版支持）。 ● <code>cachePool</code>：缓存存储池（仅集群版支持）。	否

		● storageMode: 卷的存储类型。 ● path: 数据目录（仅单机版支持）。	
range	String	查询卷的范围，格式为 i-j，i 和 j 为正整数，0<j-i<1000。按照卷名称进行排序，返回第 i 到第 j 的卷。 如果输入的查询范围超过卷的总个数，返回的结果为空，如果未指定 range，则返回所有符合条件的卷。	否

响应结果

名称	类型	描述
luns	Array of lun	卷信息集合，详见“表 1 响应参数 lun 说明”。

表 1 响应参数 lun 说明

名称	类型	描述
lunName	String	卷名称。
status	String	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。 ● Importing: 正在导入备份数据。 ● Wiping: 卷正在清空数据。 ● WipeFailed: 清空数据失败。 ● Suspended: 卷挂起。 ● Suspending: 卷正在挂起。 ● SuspendFailed: 卷挂起失败。
progress	Double	卷还原进度百分比。 仅卷状态为 Recovering 时显示。
recoverFailedReason	String	卷还原失败的原因。 仅卷状态为 RecoverFailed 时显示。
deleteFailedReason	String	卷删除失败的原因。 仅卷状态为 DeleteFailed 时显示。
wipeFailedReason	String	卷清空失败的原因。 仅卷状态为 WipeFailed 时显示。
waitUploadBytes	String	待上传数据量，单位是 bytes。 仅卷状态为 Suspending 时显示。

		示。
SuspendFailedReason	String	卷挂起失败的原因。 仅卷状态为 SuspendFailed 时显示。
storageMode	String	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云端。 ● Storage: 存储模式，本地保留全部数据，并异步存储到云端。
capacity	Integer	卷容量，单位是 GiB。
createTime	Long	卷创建时间。unix 时间戳（UTC），精确到毫秒。
iSCSITargets	Array of iSCSITarget	iSCSI target 属性集合，详见“ 表 2 响应参数说明 iSCSITarget ”。
cloud.bucketName	String	存储桶的名称（仅上云卷支持）。
cloud.endpoint	String	对象存储的 Endpoint（仅上云卷支持）。
cloud.signVersion	String	上云签名认证的类型（仅上云卷支持）： ● v2: V2 签名认证。 ● v4: V4 签名认证。
cloud.region	String	对象存储的 Endpoint 资源池所在

		区域（仅上云卷支持）。
<code>cloud.provider</code>	String	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。
<code>config.localStorageClass</code>	String	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。
<code>config.minReplica</code>	Integer	卷的最小副本数（仅集群版支持）。
<code>config.sectorSize</code>	Integer	扇区大小： ● 512。 ● 4096。
<code>config.highAvailability</code>	String	高可用模式类型（仅集群版支持）： ● ActiveStandby：主备，该卷关联对应 <code>target</code> 下的所有 IQN。 ● Disabled：禁用，该卷关联对应 <code>target</code> 下的 1 个 IQN。
<code>config.writePolicy</code>	String	卷的写策略： ● 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, 指数数据直接写到磁盘, 不写入内存。
<code>config.cachePool</code>	String	缓存存储池 (仅集群版支持)。
<code>config.pool</code>	String	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。
<code>config.redundancyOverlap</code>	Integer	卷的折叠副本数 (仅集群版支持)。
<code>config.reclaimPolicy</code>	String	卷回收策略: ● Delete: 当卷关联的快照全部删除后, 卷自动删除。 ● Retain: 当卷关联的快照全部删除后, 卷仍然保留。
<code>config.serverAffinity</code>	Object	卷主备分布优先级设置, 详见“表 4 响应参数 config.serverAffinity 说明”。
<code>config.qosPolicy</code>	Object	卷关联的 QoS 策略信息, 详见“表 5 响应参数 config.qosPolicy 说明”。
<code>config.qosPolicyforLUN</code>	Object	卷所在存储池中卷的默认 QoS 策略信息 (仅集群版支持), 详见“表 6 响应参数

		config.qosPolicyforLUN”。 说明： ● 如果卷的缓存存储池和存储池同时存在时，展示缓存存储池的“存储池内卷的默认 QoS 策略”；如果缓存存储池未设置“存储池内卷的默认 QoS 策略”，则不返回此项。 ● 如果卷只有存储池：展示存储池的“存储池内卷的默认 QoS 策略”；如果存储池未设置“存储池内卷的默认 QoS 策略”，则不返回此项。
config.ECfragmentSize	String	分片大小（仅集群版支持）。如果冗余模式为 EC $N+M$ ，则会返回此项。
config.path	String	数据目录（仅单机版支持）。
dataHealth.normal	Double	卷正常数据的百分比。
dataHealth.lowRedundancy	Double	卷降级数据的百分比。
dataHealth.error	Double	卷错误数据的百分比。
dataHealth.reconstructionProgress	Double	当卷存在降级状态的数据时，数据重建进度的百分比。
dataHealth.ETR	Long	当卷存在降级数据时，预计剩余的数据重建时间。单位为秒。如果返回值为负数，说明数据无法重建，需用户尽快修复。
number	String	卷编号。

isClone	Boolean	是否是克隆卷： ● true ：是克隆卷。 ● false ：不是克隆卷。
sourceSnapshotName	String	对应的快照名称（仅克隆卷支持）。
sourceLunName	String	克隆卷的源卷名称（仅克隆卷支持）。
snapshots.snapshotNumbers	Integer	卷快照的数量（仅本地卷支持）。
snapshots.snapshotSize	Long	卷关联的快照大小的总和（仅本地卷支持），即快照记录的数据量。单位是 bytes。 注意： 卷异常或上游快照删除等因素可能导致快照大小波动。
WWID	String	卷的唯一标识符。

表 2 响应参数 iSCSITarget 说明

名称	类型	描述
iqn	String	iSCSI target IQN。
status	String	卷对应的 target 的状态： ● Active ：主 target。 ● Standby ：热备 target。 ● Offline ：离线 target。 ● Unavailable ：冷备 target。
ips	Array of ip	iSCSI target 的 IP 属性集合，详见“表 3 响应参数 ip 说明”。
serverId	String	IQN 所在的服务器 ID。

表 3 响应参数 ip 说明

名称	类型	描述
ip	String	iSCSI target 的 IP。
port	Integer	iSCSI target 的端口号。

表 4 响应参数 config.serverAffinity 说明

名称	类型	描述
autoFailback	String	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（仅集群版支持）： ● Enabled: 自动进行主备切换。 ● Disabled: 不自动进行主备切换。
priority	Array of string	卷主备分布优先级的服务器 ID（仅集群版支持），系统根据指定的服务器 ID 顺序来选择卷的主备 IQN。

表 5 响应参数 config.qosPolicy

名称	类型	描述
qosName	String	QoS 策略名称。

表 6 响应参数 config.qosPolicyforLUN

名称	类型	描述
qosName	String	QoS 策略名称。
poolName	String	存储池名称。

请求示例 1

单机版：查询卷名称包含 luna 或 lunb 的卷。

```
GET /rest/v2/block/lun?filter=lunName:luna or lunName:lunb HTTP/1.1
```

```
Date: Wed, 31 Dec 2025 07:47:17 GMT
Authorization: userName:signature
Host: 192.168.0.66:1443
```

响应示例 1

```
HTTP/1.1 200 OK
x-hblock-request-id: a9700efafc624b89b529c534283b777f
Connection: keep-alive
Content-Length: 2875
Date: Wed, 31 Dec 2025 07:47:17 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "luns": [
      {
        "lunName": "luna1",
        "status": "Normal",
        "capacity": 100,
        "storageMode": "Local",
        "createTime": 1765260238771,
        "iSCSITargets": [
          {
            "iqn": "iqn.2012-08.cn.ctyunapi.oos:targeta.1",
            "status": "Active",
            "ips": [
              {
                "ip": "192.168.0.66",
                "port": 3260
              }
            ],
            "serverId": "hblock_1"
          }
        ]
      }
    ],
  },
}
```

```
"config": {
  "sectorSize": 4096,
  "writePolicy": "WriteBack",
  "path": "/mnt/stor01",
  "reclaimPolicy": "Retain",
  "qosPolicy": {
    "qosName": "QoS1"
  }
},
"dataHealth": {
  "normal": 100,
  "lowRedundancy": 0,
  "error": 0
},
"number": 0,
"isClone": false,
"snapshots": {
  "snapshotNumbers": 3,
  "snapshotSize": 494063616
},
"WWID": "33000000027321825"
},
{
  "lunName": "luna1-C1",
  "status": "Normal",
  "capacity": 200,
  "storageMode": "Local",
  "createTime": 1766387401903,
  "iSCSITargets": [
    {
      "iqn": "iqn.2012-08.cn.ctyunapi.oos:targeta1.4",
      "status": "Active",
      "ips": [
        {
          "ip": "192.168.0.66",
```

```
        "port": 3260
      }
    ],
    "serverId": "hblock_1"
  }
],
"config": {
  "sectorSize": 4096,
  "writePolicy": "WriteBack",
  "path": "/mnt/stor01",
  "reclaimPolicy": "Retain"
},
"dataHealth": {
  "normal": 100,
  "lowRedundancy": 0,
  "error": 0
},
"number": 0,
"isClone": true,
"sourceSnapshotName": "luna1-snapshot",
"sourceLunName": "luna1",
"snapshots": {
  "snapshotNumbers": 1,
  "snapshotSize": 0
},
"WWID": "33000000077524d87"
},
{
  "lunName": "luna2",
  "status": "Normal",
  "capacity": 1,
  "storageMode": "Local",
  "createTime": 1766022566790,
  "iSCSITargets": [
    {
```

```
        "iqn": "iqn.2012-08.cn.ctyunapi.oos:targeta.1",
        "status": "Active",
        "ips": [
            {
                "ip": "192.168.0.66",
                "port": 3260
            }
        ],
        "serverId": "hblock_1"
    }
],
"config": {
    "sectorSize": 4096,
    "writePolicy": "WriteBack",
    "path": "/mnt/stor01",
    "reclaimPolicy": "Retain",
    "qosPolicy": {
        "qosName": "QosPolicy20251218094923"
    }
},
"dataHealth": {
    "normal": 100,
    "lowRedundancy": 0,
    "error": 0
},
"number": 1,
"isClone": false,
"snapshots": {
    "snapshotNumbers": 1,
    "snapshotSize": 0
},
"WWID": "33ffffffffc93cbdeb"
},
{
    "lunName": "luna2-c1",
```

```
"status": "Normal",
"capacity": 1,
"storageMode": "Local",
"createTime": 1766371846053,
"iSCSITargets": [
  {
    "iqn": "iqn.2012-08.cn.ctyunapi.oos:targeta.1",
    "status": "Active",
    "ips": [
      {
        "ip": "192.168.0.66",
        "port": 3260
      }
    ],
    "serverId": "hblock_1"
  }
],
"config": {
  "sectorSize": 4096,
  "writePolicy": "WriteBack",
  "path": "/mnt/stor01",
  "reclaimPolicy": "Retain"
},
"dataHealth": {
  "normal": 100,
  "lowRedundancy": 0,
  "error": 0
},
"number": 2,
"isClone": true,
"sourceSnapshotName": "luna2-snap20251218095041",
"sourceLunName": "luna2",
"snapshots": {
  "snapshotNumbers": 0,
  "snapshotSize": 0
}
```

```
    },
    "WWID": "33ffffffff893d7109"
  },
  {
    "lunName": "lunb1",
    "status": "Normal",
    "capacity": 220,
    "storageMode": "Cache",
    "createTime": 1765260305925,
    "iSCSITargets": [
      {
        "iqn": "iqn.2012-08.cn.ctyunapi.oos:targetb.2",
        "status": "Active",
        "ips": [
          {
            "ip": "192.168.0.66",
            "port": 3260
          }
        ],
        "serverId": "hblock_1"
      }
    ],
    "cloud": {
      "bucketName": "hblocktest3",
      "endpoint": "https://oos-cn.ctyunapi.cn",
      "signVersion": "v2",
      "provider": "OOS"
    },
    "config": {
      "sectorSize": 4096,
      "writePolicy": "WriteBack",
      "path": "/mnt/stor01",
      "reclaimPolicy": "Retain",
      "qosPolicy": {
        "qosName": "QoS1"
      }
    }
  }
]
```

```
    }
  },
  "dataHealth": {
    "normal": 100,
    "lowRedundancy": 0,
    "error": 0
  },
  "number": 0,
  "WWID": "3300000003ea215ef"
}
]
}
```

请求示例 2

集群版：查询卷名称包含 **lun04** 或 **lun01a** 的卷，并且只列出第 1 个到第 5 个查询结果。

```
GET /rest/v2/block/lun?filter=lunName:lun04 or lunName:lun01a &range=1-5 HTTP/1.1
Date: Wed, 31 Dec 2025 07:56:50 GMT
Authorization: HBlock userName:signature
Connection: keep-alive
Host: 192.168.0.65:1443
```

响应示例 2

```
HTTP/1.1 200 OK
x-hblock-request-id: 0aa67f8aec6443689a4b09e6b2d878cb
Connection: keep-alive
Content-Length: 4643
Date: Wed, 31 Dec 2025 07:56:50 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "luns": [
```



```
{
  "lunName": "lun01a",
  "status": "Normal",
  "capacity": 110,
  "storageMode": "Local",
  "createTime": 1766021937501,
  "iSCSITargets": [
    {
      "iqn": "iqn.2012-08.cn.ctyunapi.oos:target01.1",
      "status": "Active",
      "ips": [
        {
          "ip": "192.168.0.67",
          "port": 3260
        }
      ],
      "serverId": "hblock_3"
    },
    {
      "iqn": "iqn.2012-08.cn.ctyunapi.oos:target01.2",
      "status": "Standby",
      "ips": [
        {
          "ip": "192.168.0.64",
          "port": 3260
        }
      ],
      "serverId": "hblock_2"
    }
  ],
  "config": {
    "localStorageClass": "EC 2+1",
    "minReplica": 2,
    "sectorSize": 4096,
    "highAvailability": "ActiveStandby",
```

```
        "writePolicy": "WriteBack",
        "pool": "default",
        "redundancyOverlap": 1,
        "reclaimPolicy": "Retain",
        "serverAffinity": {
            "autoFailback": "Enabled"
        },
        "qosPolicy": {
            "qosName": "QoS1"
        },
        "qosPolicyforLUN": {
            "qosName": "QoS1",
            "poolName": "default"
        },
        "ECfragmentSize": 16
    },
    "dataHealth": {
        "normal": 100,
        "lowRedundancy": 0,
        "error": 0
    },
    "number": 0,
    "isClone": false,
    "snapshots": {
        "snapshotNumbers": 2,
        "snapshotSize": 494260224
    },
    "WWID": "33ffffffffc9c6296b"
},
{
    "lunName": "lun01a-C1",
    "status": "Normal",
    "capacity": 110,
    "storageMode": "Local",
    "createTime": 1766022784343,
```

```
"iSCSITargets": [  
  {  
    "iqn": "iqn.2012-08.cn.ctyunapi.oos:target01.1",  
    "status": "Active",  
    "ips": [  
      {  
        "ip": "192.168.0.67",  
        "port": 3260  
      }  
    ],  
    "serverId": "hblock_3"  
  },  
  {  
    "iqn": "iqn.2012-08.cn.ctyunapi.oos:target01.2",  
    "status": "Standby",  
    "ips": [  
      {  
        "ip": "192.168.0.64",  
        "port": 3260  
      }  
    ],  
    "serverId": "hblock_2"  
  }  
],  
"config": {  
  "localStorageClass": "EC 2+1",  
  "minReplica": 2,  
  "sectorSize": 4096,  
  "highAvailability": "ActiveStandby",  
  "writePolicy": "WriteBack",  
  "pool": "default",  
  "redundancyOverlap": 1,  
  "reclaimPolicy": "Retain",  
  "serverAffinity": {  
    "autoFailback": "Enabled"  
  }  
}
```

```
    },
    "qosPolicy": {
      "qosName": "QoS1"
    },
    "qosPolicyforLUN": {
      "qosName": "QoS1",
      "poolName": "default"
    },
    "ECfragmentSize": 16
  },
  "dataHealth": {
    "normal": 100,
    "lowRedundancy": 0,
    "error": 0
  },
  "number": 2,
  "isClone": true,
  "sourceSnapshotName": "lun01a-s1",
  "sourceLunName": "lun01a",
  "snapshots": {
    "snapshotNumbers": 1,
    "snapshotSize": 4096
  },
  "WWID": "330000000142740b3"
},
{
  "lunName": "lun04a",
  "status": "Suspended",
  "capacity": 420,
  "storageMode": "Storage",
  "createTime": 1766382960537,
  "iSCSITargets": [
    {
      "iqn": "iqn.2012-08.cn.ctyunapi.oos:target04.8",
      "status": "Active",
```

```
        "ips": [
          {
            "ip": "192.168.0.67",
            "port": 3260
          }
        ],
        "serverId": "hblock_3"
      },
      {
        "iqn": "iqn.2012-08.cn.ctyunapi.oos:target04.7",
        "status": "Standby",
        "ips": [
          {
            "ip": "192.168.0.65",
            "port": 3260
          }
        ],
        "serverId": "hblock_1"
      }
    ],
    "cloud": {
      "bucketName": "hblocktest3",
      "endpoint": "https://oos-cn.ctyunapi.cn",
      "signVersion": "v2",
      "provider": "OOS"
    },
    "config": {
      "localStorageClass": "EC 2+1",
      "minReplica": 2,
      "sectorSize": 4096,
      "highAvailability": "ActiveStandby",
      "writePolicy": "WriteBack",
      "pool": "default",
      "redundancyOverlap": 1,
      "reclaimPolicy": "Retain",
```

```
    "serverAffinity": {
      "autoFailback": "Enabled"
    },
    "qosPolicyforLUN": {
      "qosName": "QoS1",
      "poolName": "default"
    },
    "ECfragmentSize": 16
  },
  "dataHealth": {
    "normal": 100,
    "lowRedundancy": 0,
    "error": 0
  },
  "number": 0,
  "WWID": "33ffffffff94c72aed"
},
{
  "lunName": "lun04c",
  "status": "Recovering",
  "progress": 0,
  "capacity": 480,
  "storageMode": "Cache",
  "createTime": 1767160398843,
  "iSCSITargets": [
    {
      "iqn": "iqn.2012-08.cn.ctyunapi.oos:target04.7",
      "status": "Unavailable",
      "ips": [
        {
          "ip": "192.168.0.65",
          "port": 3260
        }
      ]
    }
  ],
  "serverId": "hblock_1"
```

```
    },
    {
      "iqn": "iqn.2012-08.cn.ctyunapi.oos:target04.8",
      "status": "Unavailable",
      "ips": [
        {
          "ip": "192.168.0.67",
          "port": 3260
        }
      ],
      "serverId": "hblock_3"
    }
  ],
  "cloud": {
    "bucketName": "hblocktest3",
    "endpoint": "https://oos-cn.ctyunapi.cn",
    "signVersion": "v4",
    "region": "cn",
    "provider": "OOS"
  },
  "config": {
    "localStorageClass": "EC 2+1",
    "minReplica": 2,
    "sectorSize": 4096,
    "highAvailability": "ActiveStandby",
    "writePolicy": "WriteBack",
    "pool": "default",
    "redundancyOverlap": 1,
    "reclaimPolicy": "Retain",
    "serverAffinity": {
      "autoFailback": "Enabled"
    },
    "qosPolicy": {
      "qosName": "QoS1"
    }
  },
```

```
        "qosPolicyforLUN": {
            "qosName": "QoS1",
            "poolName": "default"
        },
        "ECfragmentSize": 16
    },
    "dataHealth": {
        "normal": 100,
        "lowRedundancy": 0,
        "error": 0
    },
    "number": 2,
    "WWID": "33000000006405e91"
},
{
    "lunName": "lun040012",
    "status": "Normal",
    "capacity": 101,
    "storageMode": "Local",
    "createTime": 1766728790846,
    "iSCSITargets": [
        {
            "iqn": "iqn.2012-08.cn.ctyunapi.oos:target04.7",
            "status": "Active",
            "ips": [
                {
                    "ip": "192.168.0.65",
                    "port": 3260
                }
            ],
            "serverId": "hblock_1"
        },
        {
            "iqn": "iqn.2012-08.cn.ctyunapi.oos:target04.8",
            "status": "Standby",
```



```
        "ips": [  
            {  
                "ip": "192.168.0.67",  
                "port": 3260  
            }  
        ],  
        "serverId": "hblock_3"  
    }  
],  
"config": {  
    "localStorageClass": "EC 2+1",  
    "minReplica": 2,  
    "sectorSize": 4096,  
    "highAvailability": "ActiveStandby",  
    "writePolicy": "WriteBack",  
    "pool": "default",  
    "redundancyOverlap": 1,  
    "reclaimPolicy": "Retain",  
    "serverAffinity": {  
        "autoFailback": "Enabled"  
    },  
    "qosPolicyforLUN": {  
        "qosName": "QoS1",  
        "poolName": "default"  
    },  
    "ECfragmentSize": 16  
},  
"dataHealth": {  
    "normal": 100,  
    "lowRedundancy": 0,  
    "error": 0  
},  
"number": 1,  
"isClone": false,  
"snapshots": {
```

```
        "snapshotNumbers": 0,  
        "snapshotSize": 0  
      },  
      "WWID": "33ffffffffcc028cda"  
    }  
  ]  
}  
}
```

4.3.18 查询指定卷

此操作用来查询指定卷。

请求语法

```
GET /rest/v2/block/lun/LunName HTTP/1.1
```

Date: *date*

Host: *ip:port*

Authorization: *authorization*

请求参数

参数	类型	描述	是否必须
lunName	String	卷名称。 说明： 如果不填写此项，会返回所有卷信息。 取值： 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。	否

响应结果

名称	类型	描述
lunName	String	卷名称。
targetName	String	target 名称。
status	String	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。 ● Importing: 正在导入备份数据。 ● Wiping: 卷正在清空数据。 ● WipeFailed: 清空数据失败。 ● Suspended: 卷挂起。 ● Suspending: 卷正在挂起。 ● SuspendFailed: 卷挂起失败。
progress	Double	卷还原进度百分比。 仅卷状态为 Recovering 时显示。
recoverFailedReason	String	卷还原失败的原因。 仅卷状态为 RecoverFailed 时显示。
deleteFailedReason	String	卷删除失败的原因。 仅卷状态为 DeleteFailed 时显示。
wipeFailedReason	String	卷清空失败的原因。 仅卷状态为 WipeFailed 时显示。
waitUploadBytes	String	待上传数据量，单位是 bytes 。 仅卷状态为 Suspending 时显示。
SuspendFailedReason	String	卷挂起失败的原因。 仅卷状态为 SuspendFailed 时显示。
capacity	Integer	卷容量，单位是 GiB 。
storageMode	String	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云端。 ● Storage: 存储模式，本地保留全部数据，并异步存储到云端。
createTime	Long	卷创建时间。unix 时间戳（UTC），精确到毫秒。
iSCSITargets	Array of iSCSITarget	iSCSI target 属性集合，详见“表 1 响应参数 iSCSITarget 说明”。
cloud.bucketName	String	存储桶的名称（仅上云卷支持）。
cloud.prefix	String	对象存储中的前缀名称（仅上云卷支持）。 如果未设置，则不返回此项。
cloud.accessKey	String	对象存储服务的 Access Key（仅上云卷支持）。
cloud.endpoint	String	对象存储 Endpoint（仅上云卷支持）。
cloud.objectSize	Integer	数据存储在云端的大小（仅上云卷支持）。
cloud.storageClass	String	数据在云端的存储类型（仅上云卷支持）： ● STANDARD: 标准存储。 ● STANDARD_IA: 低频访问存储。
cloud.compression	String	是否压缩数据上传至云端（仅上云卷支持）： ● Enabled: 压缩数据上传至云端。 ● Disabled: 不压缩数据上传至云端。
cloud.signVersion	String	上云签名认证的类型（仅上云卷支持）： ● v2: V2 签名认证。

		● v4: V4 签名认证。
cloud.region	String	Endpoint 资源池所在区域（仅上云卷支持）。
cloud.provider	String	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。
config.localStorageClass	String	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 $N+M$:
config.minReplica	Integer	卷的最小副本数。
config.ECfragmentSize	Integer	分片大小（仅集群版支持）。如果冗余模式为 EC $N+M$ ，则会返回此项。
config.sectorSize	Long	扇区大小，单位是 bytes。
config.highAvailability	String	高可用模式类型（仅集群版支持）： ● ActiveStandby: 主备，该卷关联对应 target 下的所有 IQN。 ● Disabled: 禁用，该卷关联对应 target 下的 1 个 IQN。
config.writePolicy	String	卷的写策略： ● 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。
<code>config.cachePool</code>	String	缓存存储池（仅集群版支持）。
<code>config.pool</code>	String	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。
<code>config.redundancyOverlap</code>	Integer	卷的折叠副本数（仅集群版支持）。
<code>config.reclaimPolicy</code>	String	卷回收策略： ● Delete: 当卷关联的快照全部删除后，卷自动删除。 ● Retain: 当卷关联的快照全部删除后，卷仍然保留。
<code>config.serverAffinity</code>	Object	卷主备分布优先级设置，详见“表 5 响应参数 <code>config.serverAffinity</code> 说明”。
<code>config.qosPolicy</code>	Object	卷关联的 QoS 策略信息，详见表“6 响应参数 <code>config.qosPolicy</code> 说明”。
<code>config.qosPolicyforLUN</code>	Object	卷所在存储池中卷的默认 QoS 策略信息（仅集群版支持），详见“表 7 响应参数 <code>config.qosPolicyforLUN</code> ”。
		说明： ● 如果卷的缓存存储池和存储池同时存在时，展示缓存存储池的“存储池内卷的默认 QoS 策略”；如果缓存存储池未设置“存储池内卷的默认 QoS 策略”，则不返回此项。 ● 如果卷只有存储池：展示存储池的“存储池内卷的默认 QoS 策略”；如果存储池未设置“存储池内卷的默认 QoS 策

		略”，则不返回此项。
config.path	String	数据目录（仅单机版支持）。
dataHealth.normal	Double	卷正常数据的百分比。
dataHealth.lowRedundancy	Double	卷降级数据的百分比。
dataHealth.error	Double	卷错误数据的百分比。
dataHealth.reconstructionProgress	Double	如果卷存在降级状态的数据，数据重建进度的百分比。
dataHealth.ETR	Long	卷如果存在降级数据，预计剩余的数据重建时间。单位为秒。如果返回值为负数，说明数据无法重建，需用户尽快修复。
number	Long	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。
uuid	String	卷的唯一识别码。
isClone	Boolean	是否是克隆卷： ● true: 是克隆卷。 ● false: 不是克隆卷。
sourceSnapshotName	String	对应的快照名称（仅克隆卷支持）。
sourceLunName	String	克隆卷的源卷名称（仅克隆卷支持）。
snapshots.snapshotNumbers	Integer	卷的快照个数（仅本地卷支持）。
snapshots.basisofCurData	Boolean	当前的读写是否是基于卷的初始状态： ● true: 是。 ● false: 否。
snapshots.lunSnapshots	Array of lunSnapshot	卷的快照信息集合（仅本地卷支持），详见“表 8 响应参数 lunSnapshot/Children 说明”。

snapshots.snapshotSize	Long	卷关联的快照大小的总和（仅本地卷支持），即快照记录的数据量。单位是 bytes。 注意： 卷异常或上游快照删除等因素可能导致快照大小波动。
WWID	String	卷的唯一标识符。

表 1 响应参数 iSCSITarget 说明

名称	类型	描述
iqn	String	iSCSI target IQN。
status	String	卷对应的 target 的状态： ● Active: 主 target。 ● Standby: 热备 target。 ● Offline: 离线 target。 ● Unavailable: 冷备 target。
ips	Array of ip	iSCSI target 的 IP 属性集合，详见“表 2 响应参数 ip 说明”。
targetPortalIP	Array of targetPortalIP	iSCSI 目标门户，详见“表 3 响应参数 targetPortalIP 说明”。
serverId	String	IQN 所在的服务器 ID。

表 2 响应参数 ip 说明

名称	类型	描述
ip	String	iSCSI target 的 IP。
port	Integer	iSCSI target 的端口号。

表 3 响应参数 targetPortalIP 说明

名称	类型	描述
ips	Array of ip	iSCSI 目标门户的 IP 属性集合，详见“表 4 响应参数 ip（iSCSI 目标门户的 IP）说明”。
status	String	iSCSI 目标门户的状态： ● Enabled: 启用状态。 ● Disabled: 禁用状态。

表 4 响应参数 ip（iSCSI 目标门户的 IP）说明

名称	类型	描述
ip	String	iSCSI 目标门户的 IP 地址。
port	Integer	iSCSI 目标门户的端口号。

表 5 响应参数 config.serverAffinity 说明

名称	类型	描述
autoFailback	String	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（仅集群版支持）： ● Enabled: 自动进行主备切换。 ● Disabled: 不自动进行主备切换。
priority	Array of string	卷主备分布优先级的服务器 ID（仅集群版支持），系统根据指定的服务器 ID 顺序来选择卷的主备 IQN。

表 6 响应参数 config.qosPolicy

名称	类型	描述
qosName	String	QoS 策略名称。

表 7 响应参数 config.qosPolicyforLUN

名称	类型	描述

qosName	String	QoS 策略名称。
poolName	String	存储池名称。

表 8 响应参数 lunSnapshot/Children 说明

名称	类型	描述
snapshotName	String	快照名称。
description	String	快照描述信息。
lunCapacity	Integer	创建快照时刻，源卷的容量。
consistencySnapshotName	String	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	String	快照的状态： ● Normal: 正常。 ● Error: 异常。 ● Pending: 创建中。 ● Deleting: 删除中。
reclaimPolicy	String	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。
createTime	Long	快照的创建时间。
basisofCurData	Boolean	卷当前数据是否基于该快照： ● true: 基于该快照。 ● false: 不基于该快照。
snapshotSize	Long	快照大小，具体快照记录的数据量。单位是

		bytes。 注意： 卷异常或上游快照删除等因素可能导致快照大小波动。
children	Array of children	基于该快照创建的快照信息集合，参数解释详见本表。

请求示例 1

单机版本地卷（非克隆卷）：查询卷 **luna1**。

```
GET /rest/v2/block/lun/luna1 HTTP/1.1
Date: Wed, 31 Dec 2025 08:03:56 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.66:1443
```

响应示例 1

```
HTTP/1.1 200 OK
x-hblock-request-id: 929380badbc446278733a8c5230c128f
Connection: keep-alive
Content-Length: 1245
Date: Wed, 31 Dec 2025 08:03:56 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "lunName": "luna1",
    "targetName": "targeta",
    "status": "Normal",
    "capacity": 100,
    "storageMode": "Local",
    "createTime": 1765260238771,
    "iSCSITargets": [
      {
```

```
    "iqn": "iqn.2012-08.cn.ctyunapi.oos:targeta.1",
    "status": "Active",
    "ips": [
      {
        "ip": "192.168.0.66",
        "port": 3260
      }
    ],
    "serverId": "hblock_1"
  }
],
"config": {
  "sectorSize": 4096,
  "writePolicy": "WriteBack",
  "path": "/mnt/stor01",
  "reclaimPolicy": "Retain",
  "qosPolicy": {
    "qosName": "QoS1"
  }
},
"dataHealth": {
  "normal": 100,
  "lowRedundancy": 0,
  "error": 0
},
"number": 0,
"uuid": "lun-uuid-b5e99a16-1ec0-415b-88d1-f14c4ac8e455",
"isClone": false,
"snapshots": {
  "snapshotNumbers": 3,
  "basisofCurData": false,
  "lunSnapshots": [
    {
      "snapshotName": "luna1-s1",
      "lunCapacity": 100,
```

```
    "status": "Normal",
    "reclaimPolicy": "Retain",
    "createTime": 1766022586905,
    "basisofCurData": false,
    "children": [
      {
        "snapshotName": "luna1-snap20251218095041",
        "lunCapacity": 100,
        "consistencySnapshotName": "cons1",
        "status": "Normal",
        "reclaimPolicy": "Retain",
        "createTime": 1766022641722,
        "basisofCurData": false,
        "children": [
          {
            "snapshotName": "luna1-snapshot",
            "lunCapacity": 100,
            "status": "Normal",
            "reclaimPolicy": "Retain",
            "createTime": 1766387371152,
            "basisofCurData": true,
            "snapshotSize": 45056
          }
        ],
        "snapshotSize": 200536064
      }
    ],
    "snapshotSize": 293482496
  },
  "snapshotSize": 494063616
},
"WWID": "33000000027321825"
}
```

请求示例 2

单机本地卷（克隆卷）：查询卷 luna1-C1。

```
GET /rest/v2/block/lun/luna1-C1 HTTP/1.1
Date: Wed, 31 Dec 2025 08:19:51 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.66:1443
```

响应示例 2

```
HTTP/1.1 200 OK
x-hblock-request-id: ff7ef330b99a4632825e1b10f648d6df
Connection: keep-alive
Content-Length: 858
Date: Wed, 31 Dec 2025 08:19:51 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "lunName": "luna1-C1",
    "targetName": "targeta1",
    "status": "Normal",
    "capacity": 200,
    "storageMode": "Local",
    "createTime": 1766387401903,
    "iSCSITargets": [
      {
        "iqn": "iqn.2012-08.cn.ctyunapi.oos:targeta1.4",
        "status": "Active",
        "ips": [
          {
            "ip": "192.168.0.66",
            "port": 3260
          }
        ]
      }
    ]
  }
}
```

```
    ],
    "serverId": "hblock_1"
  }
],
"config": {
  "sectorSize": 4096,
  "writePolicy": "WriteBack",
  "path": "/mnt/stor01",
  "reclaimPolicy": "Retain"
},
"dataHealth": {
  "normal": 100,
  "lowRedundancy": 0,
  "error": 0
},
"number": 0,
"uuid": "lun-uuid-e1db792b-7171-48c0-b5ae-75f2351a5356",
"isClone": true,
"sourceSnapshotName": "luna1-snapshot",
"sourceLunName": "luna1",
"snapshots": {
  "snapshotNumbers": 1,
  "basisofCurData": false,
  "lunSnapshots": [
    {
      "snapshotName": "a1-C1-s1",
      "lunCapacity": 200,
      "status": "Normal",
      "reclaimPolicy": "Retain",
      "createTime": 1767166277470,
      "basisofCurData": true,
      "snapshotSize": 0
    }
  ]
},
"snapshotSize": 0
```



```
    },  
    "WWID": "33000000077524d87"  
  }  
}
```

请求示例 3

单机版上云卷：查询卷 lunb1。

```
GET /rest/v2/block/lun/lunb1 HTTP/1.1  
Date: Wed, 31 Dec 2025 08:24:53 GMT  
Authorization: HBlock userName:signature  
Host: 192.168.0.66:1443
```

● 响应示例 3

```
HTTP/1.1 200 OK  
x-hblock-request-id: 284bb95a14f84f7dbdc5ae501ab09ab9  
Connection: keep-alive  
Content-Length: 777  
Date: Wed, 31 Dec 2025 08:24:53 GMT  
Content-Type: application/json;charset=utf-8  
Server: HBlock  
  
{  
  "data": {  
    "lunName": "lunb1",  
    "targetName": "targetb",  
    "status": "Normal",  
    "capacity": 220,  
    "storageMode": "Cache",  
    "createTime": 1765260305925,  
    "iSCSITargets": [  
      {  
        "iqn": "iqn.2012-08.cn.ctyunapi.oos:targetb.2",  
        "status": "Active",  
        "ips": [  

```

```
{
  {
    "ip": "192.168.0.66",
    "port": 3260
  }
],
  "serverId": "hblock_1"
}
],
"cloud": {
  "bucketName": "hblocktest3",
  "accessKey": "8f129a5529f202811fd0",
  "endpoint": "https://oos-cn.ctyunapi.cn",
  "objectSize": 1024,
  "storageClass": "STANDARD",
  "compression": "Enabled",
  "signVersion": "v2",
  "provider": "OOS"
},
"config": {
  "sectorSize": 4096,
  "writePolicy": "WriteBack",
  "path": "/mnt/stor01",
  "reclaimPolicy": "Retain",
  "qosPolicy": {
    "qosName": "QoS1"
  }
},
"dataHealth": {
  "normal": 100,
  "lowRedundancy": 0,
  "error": 0
},
"number": 0,
"uuid": "lun-uuid-6d66d545-e76b-489d-a9af-b918871b7626",
"WWID": "3300000003ea215ef"
```

```
}  
}
```

请求示例 4

集群本地卷（非克隆卷）：查询卷 **lun01a**。

```
GET /rest/v2/block/lun/lun01a HTTP/1.1  
Date: Wed, 31 Dec 2025 08:26:50 GMT  
Authorization: HBlock userName:signature  
Host: 192.168.0.64:1443
```

响应示例 4

```
HTTP/1.1 200 OK  
x-hblock-request-id: 46d64571da7e4c2b9056b728cde33a49  
Connection: keep-alive  
Content-Length: 1414  
Date: Wed, 31 Dec 2025 08:26:50 GMT  
Content-Type: application/json;charset=utf-8  
Server: HBlock  
  
{  
  "data": {  
    "lunName": "lun01a",  
    "targetName": "target01",  
    "status": "Normal",  
    "capacity": 110,  
    "storageMode": "Local",  
    "createTime": 1766021937501,  
    "iSCSITargets": [  
      {  
        "iqn": "iqn.2012-08.cn.ctyunapi.oos:target01.1",  
        "status": "Active",  
        "ips": [  
          {  
            "ip": "192.168.0.67",
```

```
        "port": 3260
      }
    ],
    "serverId": "hblock_3"
  },
  {
    "iqn": "iqn.2012-08.cn.ctyunapi.oos:target01.2",
    "status": "Standby",
    "ips": [
      {
        "ip": "192.168.0.64",
        "port": 3260
      }
    ],
    "serverId": "hblock_2"
  }
],
"config": {
  "localStorageClass": "EC 2+1",
  "minReplica": 2,
  "sectorSize": 4096,
  "highAvailability": "ActiveStandby",
  "writePolicy": "WriteBack",
  "pool": "default",
  "redundancyOverlap": 1,
  "reclaimPolicy": "Retain",
  "serverAffinity": {
    "autoFailback": "Enabled"
  },
  "qosPolicy": {
    "qosName": "QoS1"
  },
  "qosPolicyforLUN": {
    "qosName": "QoS1",
    "poolName": "default"
```

```
    },
    "ECfragmentSize": 16
  },
  "dataHealth": {
    "normal": 100,
    "lowRedundancy": 0,
    "error": 0
  },
  "number": 0,
  "uuid": "lun-uuid-efa1d8b2-92ee-487e-8f5d-bf0c9d5f3d8d",
  "isClone": false,
  "snapshots": {
    "snapshotNumbers": 2,
    "basisofCurData": true,
    "lunSnapshots": [
      {
        "snapshotName": "lun01a-s1",
        "lunCapacity": 110,
        "status": "Normal",
        "reclaimPolicy": "Retain",
        "createTime": 1766022696553,
        "basisofCurData": false,
        "children": [
          {
            "snapshotName": "lun01a-snap20251218095146",
            "lunCapacity": 110,
            "consistencySnapshotName": "cons1",
            "status": "Normal",
            "reclaimPolicy": "Retain",
            "createTime": 1766022706147,
            "basisofCurData": false,
            "snapshotSize": 0
          }
        ]
      },
      {
        "snapshotName": "lun01a-snap20251218095146",
        "lunCapacity": 110,
        "consistencySnapshotName": "cons1",
        "status": "Normal",
        "reclaimPolicy": "Retain",
        "createTime": 1766022706147,
        "basisofCurData": false,
        "snapshotSize": 0
      }
    ],
    "snapshotSize": 494260224
  }
}
```

```
    }  
  ],  
  "snapshotSize": 494260224  
},  
"WWID": "33ffffffffc9c6296b"  
}  
}
```

请求示例 5

集群本地卷（克隆卷）：查询卷 **lun01a-C1**。

```
GET /rest/v2/block/lun/lun01a-C1 HTTP/1.1  
Date: Wed, 31 Dec 2025 08:29:57 GMT  
Authorization: HBlock userName:signature  
Host: 192.168.0.64:1443
```

响应示例 5

```
HTTP/1.1 200 OK  
x-hblock-request-id: 3673ff6f3068432f983cf3fc08d91880  
Connection: keep-alive  
Content-Length: 1297  
Date: Wed, 31 Dec 2025 08:29:57 GMT  
Content-Type: application/json; charset=utf-8  
Server: HBlock  
  
{  
  "data": {  
    "lunName": "lun01a-C1",  
    "targetName": "target01",  
    "status": "Normal",  
    "capacity": 110,  
    "storageMode": "Local",  
    "createTime": 1766022784343,  
    "iSCSITargets": [  
      {
```

```
    "iqn": "iqn.2012-08.cn.ctyunapi.oos:target01.1",
    "status": "Active",
    "ips": [
      {
        "ip": "192.168.0.67",
        "port": 3260
      }
    ],
    "serverId": "hblock_3"
  },
  {
    "iqn": "iqn.2012-08.cn.ctyunapi.oos:target01.2",
    "status": "Standby",
    "ips": [
      {
        "ip": "192.168.0.64",
        "port": 3260
      }
    ],
    "serverId": "hblock_2"
  }
],
"config": {
  "localStorageClass": "EC 2+1",
  "minReplica": 2,
  "sectorSize": 4096,
  "highAvailability": "ActiveStandby",
  "writePolicy": "WriteBack",
  "pool": "default",
  "redundancyOverlap": 1,
  "reclaimPolicy": "Retain",
  "serverAffinity": {
    "autoFailback": "Enabled"
  },
  "qosPolicy": {
```

```
        "qosName": "QoS1"
      },
      "qosPolicyforLUN": {
        "qosName": "QoS1",
        "poolName": "default"
      },
      "ECfragmentSize": 16
    },
    "dataHealth": {
      "normal": 100,
      "lowRedundancy": 0,
      "error": 0
    },
    "number": 2,
    "uuid": "lun-uuid-28273c97-1517-4146-8aaa-8bdb5b29b51f",
    "isClone": true,
    "sourceSnapshotName": "lun01a-s1",
    "sourceLunName": "lun01a",
    "snapshots": {
      "snapshotNumbers": 1,
      "basisofCurData": false,
      "lunSnapshots": [
        {
          "snapshotName": "lun01a-C1-snap20251218104054",
          "lunCapacity": 110,
          "consistencySnapshotName": "cons2",
          "status": "Normal",
          "reclaimPolicy": "Retain",
          "createTime": 1766025654740,
          "basisofCurData": true,
          "snapshotSize": 4096
        }
      ],
      "snapshotSize": 4096
    },
  },
```



```
"WWID": "330000000142740b3"

}
```

请求示例 6

集群版上云卷：查询卷 lun04c。

```
GET /rest/v2/block/lun/lun04c HTTP/1.1
Date: Wed, 31 Dec 2025 08:32:41 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.64:1443
```

● 响应示例 6

```
HTTP/1.1 200 OK
x-hblock-request-id: ed453e40a5e745c5961074ad300d97ae
Connection: keep-alive
Content-Length: 1145
Date: Wed, 31 Dec 2025 08:32:41 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "lunName": "lun04c",
    "targetName": "target04",
    "status": "Normal",
    "capacity": 480,
    "storageMode": "Cache",
    "createTime": 1767160398843,
    "iSCSITargets": [
      {
        "iqn": "iqn.2012-08.cn.ctyunapi.oos:target04.8",
        "status": "Active",
        "ips": [
          {
```

```
        "ip": "192.168.0.67",
        "port": 3260
      }
    ],
    "serverId": "hblock_3"
  },
  {
    "iqn": "iqn.2012-08.cn.ctyunapi.oos:target04.7",
    "status": "Standby",
    "ips": [
      {
        "ip": "192.168.0.65",
        "port": 3260
      }
    ],
    "serverId": "hblock_1"
  }
],
"cloud": {
  "bucketName": "hblocktest3",
  "accessKey": "8f129a5529f202811fd0",
  "endpoint": "https://oos-cn.ctyunapi.cn",
  "objectSize": 1024,
  "storageClass": "STANDARD",
  "compression": "Enabled",
  "signVersion": "v4",
  "region": "cn",
  "provider": "OOS"
},
"config": {
  "localStorageClass": "EC 2+1",
  "minReplica": 2,
  "sectorSize": 4096,
  "highAvailability": "ActiveStandby",
  "writePolicy": "WriteBack",
```

```
    "pool": "default",
    "redundancyOverlap": 1,
    "reclaimPolicy": "Retain",
    "serverAffinity": {
      "autoFailback": "Enabled"
    },
    "qosPolicy": {
      "qosName": "QoS1"
    },
    "qosPolicyforLUN": {
      "qosName": "QoS1",
      "poolName": "default"
    },
    "ECfragmentSize": 16
  },
  "dataHealth": {
    "normal": 100,
    "lowRedundancy": 0,
    "error": 0
  },
  "number": 2,
  "uuid": "lun-uuid-faad3b10-cdbe-4e36-a849-2ba57987c41b",
  "WWID": "33000000006405e91"
}
```

4.3.19 查询卷关联的 QoS

此操作用来查询卷关联的 QoS 信息。

请求语法

```
GET /rest/v1/block/lun/LunName/qos HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

请求参数

参数	类型	描述	是否必须
lunName	String	卷名称。	是

响应结果

名称	类型	描述
qosPolicy	Object of qosPolicy	QoS 策略信息集合，详见“表 1 响应参数 qosPolicy/qosPolicyforLUN 说明”。 说明：卷关联了具体 QoS 策略时才会输出此项。
qosPolicyforLUN	Object of qosPolicyforLUN	卷所在存储池中卷的默认 QoS 策略信息（仅集群版支持），详见“表 1 响应参数 qosPolicy/qosPolicyforLUN 说明”。 说明：卷所在的存储池已设置了存储池内卷的默认 QoS 策略时才会输出此项。

表 1 响应参数 qosPolicy/qosPolicyforLUN

名称	类型	描述
qosName	String	QoS 策略名称。

IOPS	Long	每秒能够进行读写操作次数的最大值。 说明： -1 表示不限制。
readIOPS	Long	每秒能够进行读操作次数的最大值。 说明： -1 表示不限制。
writeIOPS	Long	每秒能够进行写操作次数的最大值。 说明： -1 表示不限制。
Bps	Long	每秒可传输数据量的最大值。单位是 B/s。 说明： -1 表示不限制。
readBps	Long	读带宽上限。单位是 B/s。 说明： -1 表示不限制。
writeBps	Long	写带宽上限。单位是 B/s。 说明： -1 表示不限制。
IOPSBurst	Long	使用 Burst 功能时，每秒能够进行读写操作次数的最大值。 说明： -1 表示不限制。
readIOPSBurst	Long	使用 Burst 功能时，每秒能够进行读操作次数的最大值。 说明： -1 表示不限制。
writeIOPSBurst	Long	使用 Burst 功能时，每秒能够进行写操作次数的最大值。 说明： -1 表示不限制。
BpsBurst	Long	使用 Burst 功能时，每秒可传输的数据量最大值。单位是 B/s。 说明： -1 表示不限制。
readBpsBurst	Long	使用 Burst 功能时，读带宽上限。单位是 B/s。 说明： -1 表示不限制。
writeBpsBurst	Long	使用 Burst 功能时，写带宽上限。单位是 B/s。

		说明：-1 表示不限制。
IOPSBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的能力进行读写操作所能持续的时间。单位是秒。
readIOPSBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的能力进行读操作所能持续的时间。单位是秒。
writeIOPSBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的能力进行写操作所能持续的时间。单位是秒。
BpsBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的流量能力所能持续的时间。单位是秒。
readBpsBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的读流量能力所能持续的时间。单位是秒。
writeBpsBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的写流量能力所能持续的时间。单位是秒。
createTime	Long	QoS 策略创建的时间。
reclaimPolicy	String	QoS 策略的回收策略： ● Delete: 解除 QoS 策略关联的所有对象或者删除 QoS 策略关联的所有对象，系统会触发删除 QoS 策略。 ● Retain: 解除 QoS 策略关联的所有对象或者删除 QoS 策略关联的所有对象，QoS 策略仍保留。
poolName	String	卷所属的存储池名称（仅集群版支持）。只存在 qosPolicyforLUN{ } 中。
description	String	QoS 策略的描述信息。

请求示例 1

单机版：查询卷 luna1 的 QoS 策略。

```
GET /rest/v1/block/lun/luna1/qos HTTP/1.1
```

```
Date: Thu, 21 Aug 2025 03:09:15 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.66:1443
```

响应示例 1

```
HTTP/1.1 200 OK
x-hblock-request-id: 824579904057466ebbc9e2e0aadb0afd
Connection: keep-alive
Content-Length: 484
Date: Thu, 21 Aug 2025 03:09:15 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "qosPolicy": {
      "qosName": "QoS1",
      "readIOPS": 1000,
      "writeIOPS": 1000,
      "readBps": 3670016000,
      "writeBps": 3670016000,
      "readIOPSBurst": 1000,
      "writeIOPSBurst": 1500,
      "readBpsBurst": 3670016000,
      "writeBpsBurst": 3670016000,
      "readIOPSBurstSecs": 1,
      "writeIOPSBurstSecs": 1,
      "readBpsBurstSecs": 1,
      "writeBpsBurstSecs": 1,
      "createTime": 1754375362963,
      "reclaimPolicy": "Retain",
      "description": "It is QoS1.",
      "IOPS": 2500,
      "Bps": 5242880000,
      "IOPSBurst": 2000,
```

```
        "BpsBurst": 5242880000,  
        "IOPSBurstSecs": 1,  
        "BpsBurstSecs": 1  
    }  
}  
}
```

请求示例 2

集群版：查询卷 lun01a 的 QoS 策略。

```
GET /rest/v1/block/lun/lun01a/qos HTTP/1.1  
Date: Thu, 21 Aug 2025 02:55:16 GMT  
Authorization: HBlock userName:signature  
Host: 192.168.0.64:1443
```

响应示例 2

```
HTTP/1.1 200 OK  
x-hblock-request-id: 694c5550cf95444597e49f9d7321abb3  
Connection: keep-alive  
Content-Length: 753  
Date: Thu, 21 Aug 2025 02:55:16 GMT  
Content-Type: application/json; charset=utf-8  
Server: HBlock  
  
{  
  "data": {  
    "qosPolicy": {  
      "qosName": "QoS6",  
      "readIOPS": 600,  
      "writeIOPS": 440,  
      "readBps": 1048576000,  
      "writeBps": 1572864000,  
      "readIOPSBurst": 6000,  
      "writeIOPSBurst": 4000,  
      "readBpsBurst": 2097152000,  
    }  
  }  
}
```



```
        "writeBpsBurst": 2097152000,  
        "readIOPSBurstSecs": 1,  
        "writeIOPSBurstSecs": 1,  
        "readBpsBurstSecs": 1,  
        "writeBpsBurstSecs": 1,  
        "createTime": 1753950531353,  
        "reclaimPolicy": "Retain",  
        "description": "It is QoS6.",  
        "IOPS": 1000,  
        "Bps": 2097152000,  
        "IOPSBurst": 10000,  
        "BpsBurst": 2202009600,  
        "IOPSBurstSecs": 1,  
        "BpsBurstSecs": 1  
    },  
    "qosPolicyforLUN": {  
        "qosName": "QoS-Test",  
        "readIOPSBurstSecs": 1,  
        "writeIOPSBurstSecs": 1,  
        "readBpsBurstSecs": 1,  
        "writeBpsBurstSecs": 1,  
        "createTime": 1754470952232,  
        "reclaimPolicy": "Retain",  
        "poolName": "defaultpool",  
        "IOPS": 8000,  
        "Bps": 8388608000,  
        "IOPSBurstSecs": 1,  
        "BpsBurstSecs": 1  
    }  
}
```

4.3.20 错误码列表

HTTP status	错误码	错误信息	描述
400	AccessCloudFailed	Failed to access cloud. Please check your network or contact your software vendor.	无法连接至云，请检查网络后重试或联系软件供应商。
400	BucketEnableLifecycle	Can not use this bucket because the lifecycle was enabled. Please use another bucket.	不能使用该 Bucket，因为该 Bucket 开启了生命周期。
400	BucketEnableObjectLock	Can not use this bucket because the object lock configuration was enabled. Please use another bucket.	不能使用该 Bucket，因为该 Bucket 开启了合规保留。
400	CanNotDeleteCloudData	The cloud data for LUN lunName can not be deleted. The LUN has been recovered.	无法删除云上数据，该卷已经被其他集群还原。
400	CanNotModifyEndpoint	Can not modify endpoint for LUN lunname.	无法修改卷的 Endpoint。
400	ExceedMaxCapacity	Value <i>value</i> at 'capacity' failed to satisfy constraint: Argument must be less than <i>value</i> .	存储容量非法。
400	ExceedThreshold	The number of <i>operation</i> cannot exceed <i>value</i> .	超出了系统允许的最大个数，无法执行该操作。
400	FailedOperateLUN	Failed to <i>operation</i> LUN, <i>reason</i>	卷操作执行失败。
400	HighAvailabilityNotAllowed	The 'high availability' is invalid for standalone mode of HBlock.	单机版本 HBlock 不支持设置卷的高可用

			模式。
400	InvalidBucketName	The bucket name <i>bucketName</i> is invalid.	Bucket 名称无效。
400	InvalidDiskPath	The disk paths do not exist: <i>serverIP:diskPaths</i> [,<i>diskPaths</i>...] [; <i>serverIP:diskPaths</i>[,<i>diskPaths</i>...]]. The available disk space must be greater than or equal to 1 GiB for each disk path: <i>serverIP:diskPaths</i> [,<i>diskPaths</i>...] [; <i>serverIP:diskPaths</i>[,<i>diskPaths</i>...]]. The disk paths can not access: <i>serverIP:diskPaths</i> [,<i>diskPaths</i>...] [; <i>serverIP:diskPaths</i>[,<i>diskPaths</i>...]]. The disk paths must be directory: <i>serverIP:diskPaths</i> [,<i>diskPaths</i>...] [; <i>serverIP:diskPaths</i>[,<i>diskPaths</i>...]]. The disk paths include incompatible data: <i>serverIP:diskPaths</i> [,<i>diskPaths</i>...] [; <i>serverIP:diskPaths</i>[,<i>diskPaths</i>...]]. The disk paths include unsupported symbols: <i>serverIP:diskPaths</i> [,<i>diskPaths</i>...] [; <i>serverIP:diskPaths</i>[,<i>diskPaths</i>...]]. The disk paths is already on the server and the status is removing: <i>serverIP:diskPaths</i> [,<i>diskPaths</i>...]	因数据目录有问题导致操作失败。
400	InvalidEndpoint	The endpoint <i>endpoint</i> is invalid.	Endpoint 无效。

400	InvalidEnumValue	Value <i>value</i> at ' <i>argument</i> ' failed to satisfy constraint: Argument must satisfy enum value set: [<i>value1</i> , <i>value2</i> ...]	枚举参数不合法。
400	InvalidFilterEnum	Value <i>value</i> at ' <i>filter</i> ' failed to satisfy constraint: Argument can only contain: [<i>lunName</i> , <i>status</i> , <i>capacity</i> , <i>targetName</i>].	filter 参数不正确。
400	InvalidHighAvailability	Value <i>value</i> at ' <i>high availability</i> ' failed to satisfy constraint: Argument must satisfy enum value set: [Disabled, ActiveStandby].	高可用模式取值无效。
400	InvalidLUNName	Value <i>value</i> at ' <i>lun name</i> ' failed to satisfy constraint: Argument must contain only letters, digits, or hyphens (-), and does not exceed 16 characters, must begin with a letter or digit.	卷名称不正确。
400	InvalidMinReplicaNumber	Argument ' <i>min replica</i> ' failed to satisfy constraint: argument range is [1, copy number] for replica mode and [N, N+M] for EC N+M mode.	最小副本数取值范围为：副本卷为[1, 副本数]，EC 卷为[N, N+M]。
400	InvalidPathStatus	The path <i>path</i> status is ' <i>status</i> ', the request is invalid.	数据目录状态不正确，无法执行操作。
400	InvalidPositiveInteger	Value <i>value</i> at ' <i>number</i> ' failed to satisfy constraint: Argument must be positive integer.	值必须为正整数。
400	InvalidPrefix	Value <i>value</i> at ' <i>prefix</i> ' failed to satisfy constraint: Argument must not exceed 256 characters.	prefix 无效。

400	InvalidRange	Value <i>value</i> at 'range' failed to satisfy constraint: Argument must satisfy pattern 'i-j', where i, j are positive integers, i is less than j, and j-i<1000.	range 参数不正确。
400	InvalidRecoverLUN	The LUN with name <i>lunName</i> and uuid <i>uuid</i> can not be recovered.	指定的卷无法用于还原操作，请重新输入。
400	InvalidSectorSize	Value <i>value</i> at 'sector size' failed to satisfy constraint: Argument must satisfy enum value set: [512, 4096].	卷的块大小不在枚举范围内。
400	InvalidStorageClass	Value <i>value</i> at 'local storage class' failed to satisfy constraint: Argument must satisfy enum value set: [single-copy, 2-copy, 3-copy, EC $N+M$], N and M are positive integers, $N \geq M$ and $N + M \leq 128$.	副本模式与当前实际不匹配，服务器个数不够。
400	InvalidStorageMode	Value <i>value</i> at 'storage mode' failed to satisfy constraint: Argument must satisfy enum value set: [Local, Cache, Storage].	存储模式无效。
400	InvalidTargetName	Value <i>value</i> at 'target name' failed to satisfy constraint: Argument must contain only lowercase letters, digits, dots (.) or hyphens (-), and does not exceed 16 characters, must begin with a letter or digit.	target 名称不正确。
400	InvalidWritePolicy	Value <i>value</i> at 'write policy' failed to satisfy constraint: Argument must satisfy	写策略非法。

		enum value set: [WriteBack, WriteThrough, WriteAround].	
400	InvalidXattrInnerKey	Value ' <i>value</i> ' at 'xattr inner key' failed to satisfy constraint: Argument include unsupported symbols.	“xattr inner key” 的值不满足约束条件：参数包含不支持的符号。
400	LessThanCurrentCapacity	Value <i>value</i> at 'capacity' failed to satisfy constraint: Argument must be greater than <i>value</i> .	扩容卷时新容量必须大于当前容量。
400	LessThanMinCapacity	Value <i>value</i> at 'capacity' failed to satisfy constraint: Argument must be greater than or equal to 1 GiB.	容量不能小于 1 GiB。
400	LUNDuplicatedCatalog	Duplicated with the existing LUN catalog.	与已有卷的数据目录重复。
400	MissingRelevantArgument	Value null at ' <i>argument1</i> ' failed to satisfy constraint: Argument must not be null when ' <i>argument2</i> ' has value.	参数 1 和参数 2 必须同时设置。
400	NoSuchStoragePool	The storage pool with name <i>poolName</i> does not exist.	存储池不存在。
400	OnlyHALUNAllowed	' <i>operation</i> ' is not supported for LUN luname which is not configured for high availability.	单控卷不支持该操作。
400	SamePool	Cannot use the same storage pool <i>poolName</i> .	不能使用名字相同的存储池。
400	ServerNotMatchTarget	The server <i>serverid</i> does not match the location of target <i>targetname</i> .	服务器 ID 和 target 名称不对应。
400	SingleLUNforAffinityPriority	Only one LUN is supported for allocation	仅支持针对单个

		when a specific server list is designated as the affinity priority.	LUN 设置主备优先级列表。
400	SignVersionNotAllowed	The sign version ' <i>signVersion</i> ' is not supported by provider ' <i>provider</i> '.	该对象存储服务不支持这种类型的签名，请重新选择。
400	StandaloneModeNotAllowed	' <i>operation</i> ' is not supported by standalone mode of HBlock.	单机版本 HBlock 不支持该操作。
400	StorageClassNotAllowed	The 'local storage class' is invalid for standalone mode of HBlock.	单机版 HBlock 不支持设置副本模式。
400	SwitchLUNNotAllowed	Failed to switch LUN. The LUN does not support high availability.	单控卷不支持卷的主备切换。
400	TooManyLUNs	The number of LUNs in a target cannot exceed 256.	单个 target 中的卷不能超过 256 个。
403	CloudAccessDenied	Failed to authenticate because of the privilege of access key and secret key. Please use another one.	密钥权限不足，请重新输入 AccesskeyID 和 SecretAccessKey。
403	CloudAccountAbnormal	Your cloud account has been frozen due to arrears or violations, please resolve it and try again.	云账户状态异常，可能由于欠费或违规被冻结，请解决后重试。
403	CloudRequestTimeTooSkewed	The difference between the server's time and cloud time is too large.	服务器时间和云的时间相差较大。请同步 HBlock 服务器与云上的系统时间。
403	GetLifecycleDenied	User is not authorized to perform: GetLifecycleConfiguration on	无权限获得存储桶的生命周期配置，请更

		resource: <i>bucketName</i> .	新 AK/SK 后重试。
403	GetObjectLockDenied	User is not authorized to perform: GetBucketObjectLockConfiguration on resource: <i>bucketName</i> .	无权限获得存储桶的 合规保留配置, 请更 新 AK/SK 后重试。
404	NoSuchBucket	The bucket with name <i>bucketName</i> does not exist.	Bucket 不存在。
404	NoSuchLicense	The HBlock license does not exist.	没有软件许可证。
404	NoSuchLUN	The LUN with name <i>lunName</i> does not exist.	卷不存在。
404	NoSuchLUNUuid	The LUN with name <i>lunName</i> and uuid <i>uuid</i> does not exist.	卷标识不存在, 无法 执行操作。
404	NoSuchLUNUuidConfiguration	Configuration file of the LUN with name <i>lunName</i> and uuid <i>uuid</i> does not exist.	指定的卷标识下缺失 配置文件, 无法执行 操作。
404	NoSuchPath	The path <i>path</i> does not exist in the HBlock.	路径不存在。
409	CloneLUNExist	There are clone LUNs referencing <i>Name</i> . Please delete or flatten all these clone LUNs and try again.	存在克隆卷, 无法执 行该操作, 请删除克 隆卷或者断开链接后 重试。
409	InvalidLUNMode	The storage mode of LUN <i>lunName</i> is <i>storageMode</i> , cannot operate.	该操作不支持此类型 的卷。
409	InvalidSnapshotStatus	The status of snapshot <i>name</i> is ' <i>status</i> ', the request is invalid.	快照的状态不对, 请 求无效。
409	InvalidTargetIQNsStatus	The request is invalid. Ensure the number of target IQNs whose status is neither	操作失败, target IQNs 状态异常, 请

		unavailable nor offline is greater than or equal to 2.	检查后重试。
409	LUNAlreadyExists	The LUN with name <i>lunName</i> already exists.	卷已经存在。
409	LUNDataResidue	Data residue is detected for LUN <i>lunname</i> on path <i>path</i> . Please clean up promptly and try again.	卷数据残留，请及时清理。
409	LUNRollbacking	The LUN <i>lunname</i> is using snapshot <i>snapshotname</i> for rollback, the request is invalid.	正在使用快回滚，无法执行操作。
409	SnapshotExists	The LUN <i>lunname</i> has snapshot or consistency snapshot. You can use force delete. When using this option, all snapshots of the LUN, including LUN snapshots within consistency snapshots, will be deleted together.	卷存在快照，只能使用强制删除。请注意，强制删除会一同删除快照和一致性快照中的卷快照。
500	CloudInternalError	Cloud encountered an internal error. Please try again.	云内部错误，请重试。

4.4 快照管理

4.4.1 创建快照

此操作用来创建单卷快照。

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：如果客户端支持 `sync -f`（可以通过 `sync --help` 命令查看是否支持），在客户端执行 `sync -f` 命令；否则在客户端执行 `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。
- 源卷需为处于 Normal 状态的本地卷。

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。

请求语法

```
POST /rest/v1/block/snapshot HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "snapshotName": snapshotName,
  "lunName": LunName,
  "reclaimPolicy": reclaimPolicy,
```

```
"description": description
}
```

请求参数

参数	类型	描述	是否必须
snapshotName	String	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。	是
lunName	String	指定快照对应的源卷。 注意： 仅支持本地模式（Local）的卷。	是
reclaimPolicy	String	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。 说明：如果快照回收策略为 Delete，下列操作会触发快照删除：回滚一致性快照、回滚快照、克隆卷解链操作、克隆卷删除操作、清空卷操作、删除快照。 ● 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 。	否
description	String	指定快照的描述信息。	否

		取值：1~256 位字符串。	

响应参数

名称	类型	描述
snapshotName	String	快照的名称。
lunName	String	快照对应的源卷。

请求示例

创建卷 lun01a 的快照 snapshot-lun01a-2。

<pre>POST /rest/v1/block/snapshot HTTP/1.1 Date: Wed, 05 Mar 2025 06:26:39 GMT Content-Type: application/json; charset=utf-8 Authorization: HBlock <i>userName:signature</i> Host: 192.168.0.66:1443 Connection: keep-alive Content-Length: 157 { "snapshotName": "snapshot-lun01a-2", "lunName": "lun01a", "reclaimPolicy": "Delete", "description": "The second snapshot of lun01a." }</pre>

响应示例

<pre>HTTP/1.1 202 Accepted x-hblock-request-id: 1657c5585b25408f883cb76e5d3ad7ef Date: Wed, 05 Mar 2025 06:26:39 GMT Connection: keep-alive Content-Type: application/json; charset=utf-8 Content-Length: 64 Server: HBlock</pre>

```
{  
  "data": {  
    "snapshotName": "snapshot-lun01a-2",  
    "lunName": "lun01a"  
  }  
}
```

4.4.2 修改快照

此操作用来修改单卷快照（包括一致性快照中的卷快照）信息。

请求语法

```
PUT /rest/v1/block/snapshot/snapshotName HTTP/1.1
```

Date: *date*

Content-Type: application/json; charset=utf-8

Content-Length: *length*

Host: *ip:port*

Authorization: *authorization*

```
{
  "newSnapshotName": newSnapshotName,
  "reclaimPolicy": reclaimPolicy,
  "description": description
}
```

请求参数

参数	类型	描述	是否必须
snapshotName	String	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。	是
newSnapshotName	String	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	否

		快照名称必须唯一，不可相互重复。	
reclaimPolicy	String	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。	否
description	String	修改快照的描述信息。 取值：1~256 位字符串。	否

请求示例

修改快照 snapshot-lun02-2 的名称、回收策略、描述信息。

```
PUT /rest/v1/block/snapshot/snapshot-lun02-2 HTTP/1.1
Date: Wed, 05 Mar 2025 08:17:48 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Host: 192.168.0.65:1443
Connection: keep-alive
Content-Length: 131

{
  "newSnapshotName": "snapshot-lun02-3",
  "reclaimPolicy": "Retain",
  "description": "The third snapshot of lun02."
}
```

响应示例

```
HTTP/1.1 204 No Content
x-hblock-request-id: 2bd30e44011d412ea3182d8e39b3b76b
Date: Wed, 05 Mar 2025 08:17:48 GMT
```

Connection: keep-alive

Server: HBlock

4.4.3 回滚快照

此操作用来回滚单卷快照（包括一致性快照中的卷快照）。

前提条件：回滚的快照和源卷都必须为 Normal 状态。

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：

- 不能读写。
- 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。

请求语法

```
PUT /rest/v1/block/snapshot/snapshotName/rollback HTTP/1.1
```

Date: *date*

Content-Length: *length*

Host: *ip:port*

Authorization: *authorization*

请求参数

参数	类型	描述	是否必须
snapshotName	String	指定要回滚的快照名称。 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。	是

请求示例

回滚快照 snapshot-lun02-3。

```
PUT /rest/v1/block/snapshot/snapshot-lun02-3/rollback HTTP/1.1
```

Date: Wed, 05 Mar 2025 08:37:48 GMT

Authorization: HBlock *userName:signature*

Host: 192.168.0.65:1443

Connection: keep-alive

Content-Length: 0

响应示例

```
HTTP/1.1 202 Accepted
```

x-hblock-request-id: ea5952634d5a4054b62fa73354216237

Date: Wed, 05 Mar 2025 08:37:48 GMT

Connection: keep-alive

Content-Length: 0

Server: HBlock

4.4.4 删除快照

此操作用来删除单卷快照（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。

请求语法

```
DELETE /rest/v1/block/snapshot/snapshotName HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

请求参数

参数	类型	描述	是否必须
snapshotName	String	指定要删除的快照名称。 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。	是

请求示例

删除快照 snapshot-lun01-1。

```
DELETE /rest/v1/block/snapshot/snapshot-lun01-1 HTTP/1.1
Date: Wed, 05 Mar 2025 09:33:42 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.65:1443
```

```
Connection: keep-alive
```

响应示例

```
HTTP/1.1 202 Accepted
x-hblock-request-id: 659acfea0d49484a89078cbec4b6f2aa
Date: Wed, 05 Mar 2025 09:33:42 GMT
Connection: keep-alive
Content-Length: 0
Server: HBlock
```

4.4.5 批量查询快照

此操作用来批量查询快照。

请求语法

```
GET /rest/v1/block/snapshot?filter=filter&range=i-j HTTP/1.1
```

Date: *date*

Host: *ip:port*

Authorization: *authorization*

请求参数

参数	类型	描述	是否必须
filter	String	查询的过滤条件。 过滤条件由查询属性和属性值组成，其中属性和属性值之间用单个冒号分隔（key:value）表示模糊匹配，用两个冒号分隔（key::value）表示精确匹配。可以选择多个查询，如果是或的关系，使用“or”将查询条件分隔开；如果是与的关系，使用“and”将查询条件分隔开。 支持的过滤条件类型包括： ● snapshotName: 快照名称。 ● lunName: 源卷名称。 ● status: 快照的状态。	否
range	String	查询快照的范围，格式为 i-j，i 和 j 为正整数，0<j-i<1000。 按照快照名称进行排序，返回第 i 到第 j 的快照。 如果输入的查询范围超过快照的总个数，返回的结果为空，如果未指定 range，则返回所有符合条件的快照。	否

响应参数

名称	类型	描述
snapshots	Array of snapshot	快照信息集合，详见“表 1 响应参数 snapshot 说明”。

表 1 响应参数 snapshot 说明

名称	类型	描述
snapshotName	String	快照名称。
description	String	快照描述信息。
lunName	String	源卷的名称。
lunCapacity	Integer	创建快照时刻，源卷的容量。
consistencySnapshotName	String	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	String	快照的状态： ● Normal：正常。 ● Error：异常。 ● Pending：创建中。 ● Deleting：删除中。
reclaimPolicy	String	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。
createTime	Long	快照的创建时间。
cloneNumbers	Integer	克隆卷的个数。
snapshotSize	Long	快照大小，即快照记录的数据量。单位是 bytes。 注意： 卷异常或上游快照删除等因素可能导致快

		照大小波动。

请求示例

批量查询快照信息：源卷名称中包含 **lun**，快照名称中包 **luna1**，输出第 1-3 条的显示信息。

```
GET /rest/v1/block/snapshot?filter= lunName:lun and snapshotName:luna1&range=1-3
HTTP/1.1
Date: Mon, 28 Jul 2025 08:24:31 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.66:1443
Connection: keep-alive
```

响应示例

```
HTTP/1.1 200 OK
x-hblock-request-id: c3a93a73e6844b159f369659b9c841aa
Date: Mon, 28 Jul 2025 08:24:31 GMT
Connection: keep-alive
Content-Type: application/json;charset=utf-8
Content-Length: 602
Server: HBlock

{
  "data": {
    "snapshots": [
      {
        "snapshotName": "luna1-clone-snap20250728153126",
        "lunName": "luna1-clone",
        "lunCapacity": 100,
        "consistencySnapshotName": "consnap2",
        "status": "Normal",
        "reclaimPolicy": "Retain",
        "createTime": 1753687886870,
        "cloneNumbers": 0,
```

```
        "snapshotSize": 4096
      },
      {
        "snapshotName": "luna1-snap3",
        "lunName": "luna1",
        "lunCapacity": 100,
        "status": "Normal",
        "reclaimPolicy": "Retain",
        "createTime": 1753687901979,
        "cloneNumbers": 1,
        "snapshotSize": 0
      },
      {
        "snapshotName": "luna1-snap4",
        "lunName": "luna1",
        "lunCapacity": 100,
        "status": "Normal",
        "reclaimPolicy": "Retain",
        "createTime": 1753687946159,
        "cloneNumbers": 1,
        "snapshotSize": 0
      }
    ]
  }
}
```


4.4.6 查询指定快照

此操作用查询指定快照。

请求语法

```
GET /rest/v1/block/snapshot/snapshotName HTTP/1.1
```

Date: *date*

Host: *ip:port*

Authorization: *authorization*

请求参数

参数	类型	描述	是否必须
snapshotName	String	指定要查询的快照名称。	是

响应参数

名称	类型	描述
snapshotName	String	快照名称。
description	String	快照描述信息。
lunName	String	源卷的名称。
lunCapacity	Integer	创建快照时刻，源卷的容量。
consistencySnapshotName	String	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	String	快照的状态： ● Normal：正常。 ● Error：异常。 ● Pending：创建中。 ● Deleting：删除中。

reclaimPolicy	String	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。
createTime	Long	快照的创建时间。
clones	Array of clone	克隆卷信息，详见“表 1 响应参数 clone 说明”。
snapshotSize	Long	快照大小，即快照记录的数据量。单位是 bytes。 注意 ：卷异常或上游快照删除等因素可能导致快照大小波动。

表 1 响应参数 clone 说明

名称	类型	描述
lunName	String	克隆卷名称。
capacity	Integer	克隆卷容量，单位是 GiB。
status	String	克隆卷的状态： ● Normal : 正常。 ● Deleting : 卷正在删除中。 ● DeleteFailed : 卷删除失败。 ● 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。

		● Importing: 正在导入备份数据。 ● Wiping: 卷正在清空数据。 ● WipeFailed: 清空数据失败。

请求示例 1

查询快照 luna1-snap20250728153126 的信息。

```
GET /rest/v1/block/snapshot/luna1-snap20250728153126 HTTP/1.1
Date: Mon, 28 Jul 2025 08:42:00 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.66:1443
Connection: keep-alive
```

响应示例 1

```
HTTP/1.1 200 OK
x-hblock-request-id: 213f54cf9c1b4c4d993dee36f3c36664
Date: Mon, 28 Jul 2025 08:42:00 GMT
Connection: keep-alive
Content-Type: application/json;charset=utf-8
Content-Length: 333
Server: HBlock

{
  "data": {
    "snapshotName": "luna1-snap20250728153126",
    "description": "This is the snapshot of luna1.",
    "lunName": "luna1",
    "lunCapacity": 100,
    "consistencySnapshotName": "consnap2",
    "status": "Normal",
    "reclaimPolicy": "Retain",
    "createTime": 1753687886869,
    "clones": [
      {
```

```
        "lunName": "clone-1",
        "capacity": 100,
        "status": "Normal"
    }
],
    "snapshotSize": 401063936
}
```

请求示例 2

查询快照 `luna1-snap6` 的信息。

```
GET /rest/v1/block/snapshot/luna1-snap6 HTTP/1.1
Date: Mon, 28 Jul 2025 09:03:56 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.66:1443
Connection: keep-alive
```

响应示例 2

```
HTTP/1.1 200 OK
x-hblock-request-id: aa3de918f8c447d6b45565bf4fb0af00
Date: Mon, 28 Jul 2025 09:03:56 GMT
Connection: keep-alive
Content-Type: application/json;charset=utf-8
Content-Length: 271
Server: HBlock

{
  "data": {
    "snapshotName": "luna1-snap6",
    "description": "The snapshot of luna1.",
    "lunName": "luna1",
    "lunCapacity": 100,
    "status": "Normal",
    "reclaimPolicy": "Retain",
```

```
{
  "createTime": 1753692886817,
  "clones": [
    {
      "lunName": "clone-6",
      "capacity": 100,
      "status": "Normal"
    }
  ],
  "snapshotSize": 98304
}
```

4.4.7 错误码列表

HTTP status	错误码	错误信息	描述
400	ExceedThreshold	The <i>item</i> of <i>operation</i> cannot exceed <i>value</i> .	超出了系统允许的最大个数，无法执行该操作。
400	InvalidArgumentLength	Value <i>value</i> at ' <i>argument</i> ' failed to satisfy constraint: Argument must not exceed <i>value</i> characters.	参数长度错误。
400	InvalidFilter	Value <i>value</i> at ' <i>filter</i> ' failed to satisfy constraint: Argument must use a single colon (':' means fuzzy match) or two colons '::' means exact match) to separate the attribute key and value. If you query multiple attributes, you can use 'and' or 'or' keywords to separate them, and there must be spaces before and after 'and' or 'or'.	filter 参数不合法。
400	InvalidFilterEnum	Value <i>value</i> at ' <i>filter</i> ' failed to satisfy constraint: Argument can only contain: [snapshotName, runningStatus, lunName].	filter 参数不正确。
400	InvalidRange	Value <i>value</i> at ' <i>range</i> ' failed to satisfy constraint: Argument must satisfy pattern ' <i>i-j</i> ', where <i>i</i> , <i>j</i> are	range 参数不正确。

		positive integers, i is less than j , and $j-i < 1000$.	
400	InvalidSnapshotName	Value <i>value</i> at 'snapshot name' failed to satisfy constraint: Argument can contain letters, digits, underscores (<code>_</code>) or hyphens (<code>-</code>), and the length is no more than 256, must begin with a letter or digit.	快照名称不符合要求：字符串形式，长度范围是 1~256，只能由字母、数字、短横线(<code>-</code>)、下划线(<code>_</code>)组成，字母区分大小写，且仅支持以字母或数字开头。
400	MissingArgument	Value null at ' <i>argument</i> ' failed to satisfy constraint: Argument must not be null.	请求中必填元素未填写。
400	SnapshotReferenced	The snapshot <i>snapshotname</i> cannot be deleted because it is the basis of multiple snapshots <i>XX</i> [, <i>XX</i>].	快照是 2 个及 2 个以上子节点的快照，无法删除。
404	NoSuchLUN	The LUN with name <i>lunName</i> does not exist.	卷不存在。
404	NoSuchSnapshot	The snapshot with name <i>SnapshotName</i> does not exist.	快照不存在。
409	CloneLUNExist	There are clone LUNs referencing <i>Name</i> . Please delete or flatten all these clone LUNs and try again.	存在克隆卷，无法执行该操作，请删除克隆卷或者断开链接后重试。
409	InvalidLUNMode	The storage mode of LUN <i>lunName</i> is <i>storageMode</i> , cannot operate.	cache 卷和 storage 卷不支持快照相关操作。

409	InvalidLUNStatus	The status of LUN ' <i>lunname</i> ' is ' <i>status</i> ', the request is invalid.	LUN 当前的状态不正确，请求无效。
409	InvalidSnapshotStatus	The status of snapshot <i>name</i> is ' <i>status</i> ', the request is invalid.	快照的状态不对，请求无效。
409	LUNRollbacking	The LUN <i>lunname</i> is using snapshot <i>snapshotname</i> for rollback, the request is invalid.	正在使用快回滚，无法执行操作。
409	SnapshotAlreadyExists	The snapshot with name <i>snapshotName</i> already exists.	快照已存在，请重新输入。
500	InternalError	HBlock encountered an internal error. Please try again or contact the software vendor.	请重试或者联系软件供应商。

4.5 一致性快照管理

4.5.1 创建一致性快照

此操作用来创建一致性快照。

一致性快照是指在某一时刻，对选中的所有卷同时创建快照，以确保数据一致性。

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**操作修改快照名称。

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：如果客户端支持 `sync -f`（可以通过 `sync --help` 命令查看是否支持），在客户端执行 `sync -f` 命令；否则在客户端执行 `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，源卷需为处于 Normal 状态的本地卷。

请求语法

```
POST /rest/v1/block/conssnap HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "luns": [
    {
```

```

        "lunName": LunName1,
        "description": description
    },
    {
        "lunName": LunName2,
        "description": description
    },
    .....
],
"consistencySnapshotName": consistencySnapshotName,
"reclaimPolicy": reclaimPolicy,
"description": description
}

```

请求参数

参数	类型	描述	是否必须
luns	Array of lun	一致性快照的源卷信息，详见“ 表 1 请求参数 lun 说明 ”。 说明： 一次最多可以添加 512 个源卷。	是
consistencySnapshotName	String	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。	是
reclaimPolicy	String	一致性快照内的单卷快照的回收策略。 取值：	否

		● 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。	
description	String	指定一致性快照的描述信息。 取值：1~256 位字符串。	否

表 1 请求参数 lun 说明

参数	类型	描述	是否必须
lunName	String	指定一致性快照对应的源卷。 注意： 仅支持本地模式（Local）的卷。最多可以添加 512 个卷。	是
description	String	指定一致性快照中卷快照的描述信息。 取值：1~256 位字符串。	否

响应参数

名称	类型	描述
consistencySnapshotName	String	一致性快照的名称。
lunSnapshots	Array of lunSnapshot	一致性快照中单卷快照的信息集合，详见“表 2 响应参数 lunSnapshot 说明”。

表 2 响应参数 lunSnapshot 说明

名称	类型	描述

snapshotName	String	单卷快照的名称。
lunName	String	单卷快照的源卷名称。

请求示例

创建卷 lun001、lun01a 的一致性快照 consistencysnapshot3。

```
POST /rest/v1/block/conssnap HTTP/1.1
Date: Wed, 26 Mar 2025 05:35:41 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Host: 192.168.0.65:1443
Connection: keep-alive
Content-Length: 429

{
  "luns": [
    {
      "lunName": "lun001",
      "description": "The consistency snapshot of lun001."
    },
    {
      "lunName": "lun01a",
      "description": "The consistency snapshot of lun01a."
    }
  ],
  "consistencySnapshotName": "consistencysnapshot3",
  "reclaimPolicy": "Retain",
  "description": "The consistency snapshot of lun001 and lun01a."
}
```

响应示例

```
HTTP/1.1 202 Accepted
x-hblock-request-id: f53ec5817a1a48a2bf25fc5ed3546784
```

```
Date: Wed, 26 Mar 2025 05:35:41 GMT
Connection: keep-alive
Content-Type: application/json;charset=utf-8
Content-Length: 204
Server: HBlock

{
  "data": {
    "consistencySnapshotName": "consistencysnapshot3",
    "lunSnapshots": [
      {
        "snapshotName": "lun001-snap20250326133541",
        "lunName": "lun001"
      },
      {
        "snapshotName": "lun01a-snap20250326133541",
        "lunName": "lun01a"
      }
    ]
  }
}
```

4.5.2 修改一致性快照

此操作用来修改一致性快照信息。

请求语法

```
PUT /rest/v1/block/conssnap/consistencySnapshotName HTTP/1.1
```

Date: *date*

Content-Type: application/json; charset=utf-8

Content-Length: *length*

Host: *ip:port*

Authorization: *authorization*

```
{
  "newSnapshotName": newSnapshotName,
  "reclaimPolicy": reclaimPolicy,
  "description": description
}
```

请求参数

参数	类型	描述	是否必须
consistencySnapshotName	String	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。	是
newSnapshotName	String	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	否

		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。	
reclaimPolicy	String	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。	否
description	String	修改一致性快照的描述信息。 取值：1~256 位字符串。	否

请求示例

修改一致性快照 `consistencysnapshot2`：名称为 `consistencysnapshot-lun001-lun01a`，回收策略为 `Retain`，描述信息为 “The consistency snapshot of lun001 and lun01a.”。

```
PUT /rest/v1/block/conssnap/consistencysnapshot2 HTTP/1.1
Date: Thu, 06 Mar 2025 10:10:36 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Host: 192.168.0.65:1443
Connection: keep-alive
Content-Length: 166

{
  "newSnapshotName": "consistencysnapshot-lun001-lun01a",
```

```
"reclaimPolicy": "Retain",  
"description": "The consistency snapshot of lun001 and lun01a."  
}
```

响应示例

```
HTTP/1.1 204 No Content  
x-hblock-request-id: f9372193ce35405fb7ff0a7b78e47907  
Date: Thu, 06 Mar 2025 10:10:36 GMT  
Connection: keep-alive  
Server: HBlock
```


4.5.3 回滚一致性快照

此操作用来回滚一致性快照。

前提条件：回滚的一致性快照和源卷都必须为 Normal 状态。

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。

请求语法

```
PUT /rest/v1/block/conssnap/consistencySnapshotName/rollback HTTP/1.1
Date: date
Content-Length: Length
Host: ip:port
Authorization: authorization
```

请求参数

参数	类型	描述	是否必须
consistencySnapshotName	String	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。	是

响应结果

名称	类型	描述
lunList	Array of LUN	执行回滚的卷列表。

请求示例

回滚一致性快照 consistencysnapshot4。

```
PUT /rest/v1/block/conssnap/consistencysnapshot4/rollback HTTP/1.1
Date: Mon, 28 Jul 2025 03:01:21 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.65:1443
Connection: keep-alive
Content-Length: 0
```

响应示例

```
HTTP/1.1 202 Accepted
x-hblock-request-id: 1fe1d1894e9d47e0acc2b3590c6c885c
Date: Mon, 28 Jul 2025 03:01:21 GMT
Connection: keep-alive
Content-Type: application/json;charset=utf-8
Content-Length: 58
```

Server: HBlock

```
{
  "data": {
    "lunList": [
      "lun01a",
      "lun01b",
      "lun01c",
      "lun02a"
    ]
  }
}
```

4.5.4 删除一致性快照

此操作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。

请求语法

```
DELETE /rest/v1/block/conssnap/consistencySnapshotName HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

请求参数

参数	类型	描述	是否必须
consistencySnapshotName	String	指定要删除的一致性快照名称。	是

请求示例

删除一致性快照 consistencysnapshot-lun001-lun01a。

```
DELETE /rest/v1/block/conssnap/consistencysnapshot-lun001-lun01a HTTP/1.1
Date: Fri, 07 Mar 2025 01:45:41 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.65:1443
Connection: keep-alive
```

响应示例

```
HTTP/1.1 202 Accepted
x-hblock-request-id: da16dc4a8eb2421d8c7af6c04bc919a5
```

```
Date: Fri, 07 Mar 2025 01:45:41 GMT
Connection: keep-alive
Content-Length: 0
Server: HBlock
```

4.5.5 批量查询一致性快照

此操作用来批量查询一致性快照。

请求语法

```
GET /rest/v1/block/conssnap?filter=filter&range=i-j HTTP/1.1  
Date: date  
Host: ip:port  
Authorization: authorization
```

请求参数

参数	类型	描述	是否必须
filter	String	查询的过滤条件。 过滤条件由查询属性和属性值组成，其中属性和属性值之间用单个冒号分隔（key:value）表示模糊匹配，用两个冒号分隔（key::value）表示精确匹配。可以选择多个查询，如果是或的关系，使用“or”将查询条件分隔开；如果是与的关系，使用“and”将查询条件分隔开。 支持的过滤条件类型包括： ● consistencySnapshotName: 一致性快照的名称。	否
range	String	查询一致性快照的范围，格式为 i-j，i 和 j 为正整数，0<j-i<1000。按照一致性快照名称进行排序，返回第 i 到第 j 的快照。 如果输入的查询范围超过一致性快照的总个数，返回的结果为空，如果未指定 range，则返回所有符合条件的一致性快照。	否

响应参数

名称	类型	描述
consistencySnapshot	Array of consistencySnapshot	一致性快照信息集合，详见“表 1 响应参数 consistencySnapshot 说明”。

表 1 响应参数 consistencySnapshot 说明

名称	类型	描述
consistencySnapshotName	String	一致性快照名称。
description	String	一致性快照描述信息。
lunSnapshotNumbers	Integer	一致性快照的卷快照数。
status	String	一致性快照的状态： ● Normal：正常。 ● Pending：创建中。 ● Deleting：删除中。 ● Error：异常。
createTime	Long	一致性快照的创建时间。
lunSnapshots	Array of lunSnapshot	一致性快照中卷快照信息集合，详见“表 2 响应参数 lunSnapshot 说明”。

表 2 响应参数 lunSnapshot 说明

名称	类型	描述
snapshotName	String	一致性快照中卷快照名称。
cloneNumbers	Integer	卷快照关联的链接克隆卷的个数。

请求示例

批量查询一致性快照信息：一致性快照名称中包含 consistencysnapshot。

```
GET /rest/v1/block/conssnap?filter=consistencySnapshotName:consistencysnapshot HTTP/1.1
Date: Tue, 11 Mar 2025 02:22:20 GMT
```

```
Authorization: HBlock userName:signature  
Host: 192.168.0.66:1443  
Connection: keep-alive
```

响应示例

```
HTTP/1.1 200 OK  
x-hblock-request-id: fcd556c82e19480ab050c200c631e89f  
Date: Tue, 11 Mar 2025 02:22:20 GMT  
Connection: keep-alive  
Content-Type: application/json;charset=utf-8  
Content-Length: 719  
Server: HBlock  
  
{  
  "data": {  
    "consistencySnapshot": [  
      {  
        "consistencySnapshotName": "consistencysnapshot1",  
        "description": "consistency snapshot",  
        "lunSnapshotNumbers": 1,  
        "status": "Normal",  
        "createTime": 1741076297083,  
        "lunSnapshots": [  
          {  
            "snapshotName": "lun01a-snap20250304161817",  
            "cloneNumbers": 1  
          }  
        ]  
      },  
      {  
        "consistencySnapshotName": "consistencysnapshot4",  
        "description": "The consistencysnapshot of lun01,lun01a,lun02,lun03.",  
        "lunSnapshotNumbers": 4,  
        "status": "Normal",  
        "createTime": 1741079382519,
```



```
    "lunSnapshots": [  
      {  
        "snapshotName": "lun01-snap20250304170942",  
        "cloneNumbers": 1  
      },  
      {  
        "snapshotName": "lun02-snap20250304170942",  
        "cloneNumbers": 0  
      },  
      {  
        "snapshotName": "lun01a-snap20250304170942",  
        "cloneNumbers": 2  
      },  
      {  
        "snapshotName": "lun03-snap20250304170942",  
        "cloneNumbers": 0  
      }  
    ]  
  }  
]  
}
```

4.5.6 查询指定一致性快照

此操作用查询指定一致性快照。

请求语法

```
GET /rest/v1/block/conssnap/consistencySnapshotName HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

请求参数

参数	类型	描述	是否必须
consistencySnapshotName	String	指定要查询的一致性快照名称。	是

响应参数

名称	类型	描述
consistencySnapshotName	String	一致性快照名称。
description	String	一致性快照描述信息。
status	String	一致性快照的状态： ● Normal: 正常。 ● Pending: 创建中。 ● Deleting: 删除中。 ● Error: 异常。
createTime	Long	一致性快照的创建时间。
lunSnapshotNumbers	Integer	一致性快照的卷快照数。
lunSnapshots	Array of lunSnapshot	一致性快照中卷快照信息集合，详见“表 1 响应参数 lunSnapshot 说明”。

表 1 响应参数 lunSnapshot 说明

名称	类型	描述
snapshotName	String	一致性快照中卷快照名称。
description	String	卷快照的描述信息。
lunName	String	源卷名称。
lunCapacity	Integer	创建快照时刻，源卷的容量，单位是 GiB。
status	String	卷快照的状态： ● Normal: 正常。 ● Error: 异常。 ● Pending: 创建中。 ● Deleting: 删除中。
cloneNumbers	Integer	卷快照关联的链接克隆卷的个数。
reclaimPolicy	String	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。

请求示例

查询一致性快照 consistencysnapshot4。

```
GET /rest/v1/block/conssnap/consistencysnapshot4 HTTP/1.1
Date: Thu, 21 Aug 2025 05:46:01 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.65:1443
Connection: keep-alive
```

响应示例

```
HTTP/1.1 200 OK
x-hblock-request-id: d43aa43c9efe42b2b3f943569e255658
Date: Thu, 21 Aug 2025 05:46:01 GMT
Connection: keep-alive
Content-Type: application/json;charset=utf-8
Content-Length: 789
Server: HBlock

{
  "data": {
    "consistencySnapshotName": "consistencysnapshot4",
    "description": "There are 4 luns.",
    "status": "Normal",
    "createTime": 1755755068817,
    "lunSnapshotNumbers": 4,
    "lunSnapshots": [
      {
        "snapshotName": "lun001-c001-snap20250821134428",
        "lunName": "lun001-c001",
        "lunCapacity": 101,
        "status": "Normal",
        "cloneNumbers": 0,
        "reclaimPolicy": "Retain"
      },
      {
        "snapshotName": "lun001-snap20250821134428",
        "lunName": "lun001",
        "lunCapacity": 101,
        "status": "Normal",
        "cloneNumbers": 0,
        "reclaimPolicy": "Retain"
      },
      {
        "snapshotName": "lun01a-snap20250821134428",
```

```
        "description": "lun01a's snapshot.",
        "lunName": "lun01a",
        "lunCapacity": 200,
        "status": "Normal",
        "cloneNumbers": 0,
        "reclaimPolicy": "Retain"
    },
    {
        "snapshotName": "lun04-snap20250821134428",
        "lunName": "lun04",
        "lunCapacity": 404,
        "status": "Normal",
        "cloneNumbers": 0,
        "reclaimPolicy": "Retain"
    }
]
}
```

4.5.7 错误码列表

HTTP status	错误码	错误信息	描述
400	ExceedThreshold	The number of <i>operation</i> cannot exceed value.	超出了系统允许的最大个数，无法执行该操作。
400	InvalidArgumentLength	Value <i>value</i> at ' <i>argument</i> ' failed to satisfy constraint: Argument must not exceed <i>value</i> characters.	参数长度错误。
400	InvalidFilter	Value <i>value</i> at ' <i>filter</i> ' failed to satisfy constraint: Argument must use a single colon ':' means fuzzy match) or two colons '::' means exact match) to separate the attribute key and value. If you query multiple attributes, you can use 'and' or 'or' keywords to separate them, and there must be spaces before and after 'and' or 'or'.	filter 参数不合法。
400	InvalidFilterEnum	Value <i>value</i> at ' <i>filter</i> ' failed to satisfy constraint: Argument can only contain:	filter 参数不正确。

		[snapshotName, runningStatus, lunName].	
400	InvalidRange	Value <i>value</i> at 'range' failed to satisfy constraint: Argument must satisfy pattern 'i-j', where i, j are positive integers, i is less than j, and j-i<1000.	range 参数不正确。
400	InvalidSnapshotName	Value <i>value</i> at 'consistency snapshot name' failed to satisfy constraint: Argument can contain letters, digits, underscores (_) or hyphens (-), and the length is no more than 256, must begin with a letter or digit.	一致性快照名称不符合要求：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。
400	MissingArgument	Value null at ' <i>argument</i> ' failed to satisfy constraint: Argument must not be null.	请求中必填元素未填写。
404	NoSuchLUN	The LUN with name <i>lunName</i> does not exist.	卷不存在。
404	NoSuchConsistencySnapshot	The consistency snapshot with name <i>ConsistencySnapshotName</i>	一致性快照不存在。

		does not exist.	
409	CloneLUNExist	There are clone LUNs referencing <i>Name</i> . Please delete or flatten all these clone LUNs and try again.	存在克隆卷，无法执行该操作，请删除克隆卷或者断开链接后重试。
409	ConsistencySnapshotAlreadyExists	The consistency snapshot with name <i>consistencysnapshotName</i> already exists.	一致性快照已存在，请重新输入。
409	InvalidConsistencySnapshotStatus	The status of consistency snapshot <i>snapshotname</i> is ' <i>status</i> ', the request is invalid.	一致性快照的状态不对，请求无效。
409	InvalidLUNMode	The storage mode of LUN <i>lunName</i> is <i>storageMode</i> , cannot operate.	cache 卷和 storage 卷不支持快照相关操作。
409	InvalidLUNStatus	The status of LUN ' <i>lunname</i> ' is ' <i>status</i> ', the request is invalid.	LUN 当前的状态不正确，请求无效。
409	InvalidSnapshotStatus	The status of snapshot <i>name</i> is ' <i>status</i> ', the request is invalid.	快照的状态不对，请求无效。
409	LUNRollbacking	The LUN <i>lunname</i> is using snapshot <i>snapshotname</i> for rollback, the request is invalid.	正在使用快回滚，无法执行操作。

500	InternalError	HBlock encountered an internal error. Please try again or contact the software vendor.	请重试或者联系软件供应商。

4.6 备份管理（本地卷）

4.6.1 导出备份

此操作用来导出备份。

备份分为全量备份和增量备份：

- **全量备份**：导出从卷创建到快照时刻数据到备份文件中。
- **增量备份**：导出同一个卷的两个快照之间的差异数据到备份文件中，如创建 snap1-snap2 的增量备份，即导出从快照 snap1 时刻到快照 snap2 时刻的卷数据。

注意：

- 只有在快照均为 Normal 状态下才可以导出备份。
- 只有卷在 Normal、Flattening 和 Rollbacking 状态下，才可以导出备份。
- 导出备份时，不能删除相关的快照和卷。
- 每个服务器最多同时导出 4 个备份，此处的服务器为快照源卷 Activer IQN 所在的服务器。

请求语法 1

```
POST /rest/v1/block/backup/export HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "snapshot": snapshot,
  "fromSnap": snapshot,
  "breakpoint": breakpoint,
  "compression": compression
}
```

请求语法 2

GET
/rest/v1/block/backup/export?snapshot= <i>snapshotName</i> &checkFlag= <i>checkFlag</i> &fromSnap= <i>snapshotName</i> &breakpoint= <i>breakpoint</i> &compression= <i>compression</i> HTTP/1.1
Date: <i>date</i>
Content-Type: application/json; charset=utf-8
Content-Length: <i>length</i>
Host: <i>ip:port</i>
Authorization: <i>authorization</i>

请求参数

参数	类型	描述	是否必须
snapshot	String	快照名称，表示将快照时刻的数据进行导出。	是
fromSnap	String	快照名称，表示导出此快照时刻之后的数据。 说明： 导出增量备份时必须设置此参数，表示增量备份的起始快照名称。不设置此参数，表示从卷的起始位置开始备份。	否
breakpoint	Long	断点位置，本次备份任务导出任务的起始位置。如果上次导出时中断，本次导出任务需从中断位置进行导出。 取值：268435456 的倍数，单位是 bytes。不输入或者 0 表示导出全部数据。如果超过了备份文件的大小，则以空数据生成一个空备份文件。 说明： 针对非压缩模式的备份文件，由固定	否

		大小的分片（256MiB）组成。在计算导出任务的中断位置时，根据已经导出的备份文件大小，以及固定分片大小，计算出已经成功导出的分片文件大小，即中断位置。 示例： - 第一次中断时，已写入 257 MiB，向下取整到 256 MiB，故中断位置 = 268435456 bytes（1 个完整 256 MiB 分片）。 - 第二次中断时，累计写入 256 + 530 = 786 MiB，向下取整到 768 MiB（3 个完整 256 MiB 分片），故中断位置 = 805306368 bytes。 后续中断依此规则类推。	
compression	String	是否对备份文件压缩。 取值： ● Enabled：压缩。 ● Disabled：不压缩。 如果 breakpoint 取值不为 0，默认值为 Disabled；如果 breakpoint 取值为 0，或者没有赋值，默认值为 Enabled。 注意： compression=Enabled 和 breakpoint=!0 不能同时存在。	否
checkFlag	Boolean	是否仅校验参数。 取值： ● true：仅校验参数。参数无问题返回 200；参数有问题返回具体错误。	否

		● false: 校验参数并返回文件流。参数无问题直接返回文件流；参数有问题返回具体错误信息。 默认值为 false 。	

请求示例 1

导出全量备份。

```
POST /rest/v1/block/backup/export HTTP/1.1
Date: Wed, 30 Jul 2025 06:51:04 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Host: 192.168.0.66:1443
Connection: keep-alive
Content-Length: 35

{
  "snapshot": "luna1-snap1"
}
```

响应示例 1

```
HTTP/1.1 200 OK
x-hblock-request-id: 7548f1bb2ae048c59925ab8f15c8dcc5
Date: Wed, 05 Mar 2025 06:26:39 GMT
Connection: keep-alive
Content-Type: application/octet-stream; charset=utf-8
Server: HBlock
Content-Disposition: attachment; filename=[luna1]_[luna1-snap1]_20250730145104
```

请求示例 2

导出增量备份。

```
GET /rest/v1/block/backup/export?snapshot=luna1-snap3&checkFlag=false&fromSnap=luna1-
```

```
snap1 HTTP/1.1
Date: Wed, 30 Jul 2025 07:07:55
Authorization: HBlock userName:signature
Host: 192.168.0.66:1443
Connection: keep-alive
```

响应示例 2

```
HTTP/1.1 200 OK
x-hblock-request-id: d16ed90970f3443daab104b6b9cd5441
Date: Wed, 30 Jul 2025 07:07:57
Connection: keep-alive
Content-Type: application/octet-stream;charset=utf-8
Server: HBlock
Content-Disposition: attachment;filename=[luna1]_[luna1-snap1]_[luna1-snap3]_20250730150757
```

4.6.2 导入备份

此操作用来将备份文件的数据导入卷。

导入全量备份时，建议使用新卷，或者卷原有数据已经清除。可以使用清空卷的功能实现卷数据的清除，根据实际情况决定是否保留卷的快照数据。

注意：

- 每次只能导入一个备份文件。如果涉及多个备份文件，请按照快照生成的先后顺序依次导入对应的备份文件。
- 被导入备份的卷，导入的时候卷不能有读写操作。
- 导入备份文件时，尤其是涉及多个全量和增量备份导入时，在没有把多个备份文件导入完成之前，不能将目标卷挂载到客户端，此时如果挂载到客户端可能会引起文件系统损坏。需将全部需要导入的备份导入完成后再进行挂载，且挂载后不再导入备份文件。如果卷已经挂载到客户端，需先取消挂载，执行清空卷后再导入备份文件。
- 如果用户先将完整备份文件自行拆分为多个分片，再执行导入操作，需要确保前一个分片文件与后一个分片文件存在数据重叠，并且每个分片的起始位置需为 256MiB 的倍数。例如第一个分片为 0 - 515MiB，第二个分片为 512MiB - 1024MiB。

请求语法

```
PUT /rest/v1/block/backup/importfile/LunName HTTP/1.1
Date: date
Content-Type: application/octet-stream
Content-Length: Length
Host: ip:port
Authorization: authorization

backupFileContent
```

请求参数

参数	类型	描述	是否必须

lunName	String	要导入备份的卷名称。 注意： 卷大小需要大于备份文件对应快照在快照时刻的源卷大小。	是
backupFileContent	无	要导入的备份文件。	是

请求示例

导入备份文件到 luncl。

```
PUT /rest/v1/block/backup/importfile/lunc1 HTTP/1.1
Date: Wed, 30 Jul 2025 08:48:10 GMT
Content-Type: application/octet-stream
Authorization: HBlock userName:signature
Host: 192.168.0.66:1443
Connection: keep-alive
Content-Length: 300190151

file
```

响应示例

```
HTTP/1.1 204 No Content
x-hblock-request-id: e89fd1bc1a264baaa26e4c7e50c63585
Date: Wed, 30 Jul 2025 08:48:10 GMT
Connection: keep-alive
Server: HBlock
```


4.6.3 错误码列表

HTTP status	错误码	错误信息	描述
400	BackupFailed	Failed to execute backup task. Reason: <i>reason</i>	备份任务执行失败。
400	InvalidBackupFile	Invalid or conflicting data in backup file. Please get the correct file.	您的备份文件错误，请更换正确的备份文件后重试。
400	InvalidLong	Value <i>value</i> at ' <i>argument</i> ' failed to satisfy constraint: Argument must be of type long.	参数填写错误，必须为长整型。
400	InvalidMultipleValue	Value <i>value</i> at ' <i>argument</i> ' failed to satisfy constraint: Argument must be the multiples of <i>value2</i> .	<i>value</i> 需要是某个数(<i>value2</i>)的倍数值。
400	InvalidSnapBranch	<i>fromsnapshotA</i> and <i>tosnapshotB</i> must be in the same branch.	增量备份对应的 2 个快照不能分叉。
400	InvalidSnapSequence	Snapshot sequence error: Operation requires from <i>fromsnapshotA</i> to <i>tosnapshotB</i> .	两个快照顺序写反了。
400	InsufficientLUNCapacity	The capacity of LUN lunname needs to be extended to <i>XXX</i> GiB.	卷容量不足，需要扩容到某个值才可以。
400	LUNNotMatch	<i>snapshotA</i> and <i>snapshotB</i> do not belong to the same LUN.	起始快照和目标快照不属于同一

			个卷。
400	SameSnapshot	Cannot use the same snapshot <i>snapshotName</i> .	不能使用同一个快照名称。
409	BackupExist	The backup is in progress. Please wait for it to finish and try again.	备份任务正在进行，无法执行该操作，请等待备份任务完成后重试。
409	InvalidLUNMode	The storage mode of LUN <i>lunName</i> is <i>storageMode</i> , cannot operate.	该操作不支持此类型的卷。
409	InvalidLUNStatus	The status of LUN ' <i>lunname</i> ' is ' <i>status</i> ', the request is invalid.	LUN 当前的状态不正确，请求无效。

4.7 iSCSI target 管理

4.7.1 创建 iSCSI target

此操作用来创建 iSCSI target。

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。

请求语法

● 单机版

```
POST /rest/v1/block/target HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization
```

```
{
  "targetName": targetName,
  "maxSessions": value,
  "reclaimPolicy": reclaimPolicy,
  "chap": {
    "name": name,
    "password": password,
    "status": status
  },
  "allowlist": {
    "initiator": [
      {
        "IPs": [ip, ip, ...],
        "names": [name, name, ...]
      },
      {
        "IPs": [ip, ip, ...],
```

```
        "names": [name,name,...]
    },
    .....
],
"target": [
    {
        "IPs": [ip,ip,...],
        "NICs": [NIC,NIC,...]
    },
    {
        "IPs": [ip,ip,...],
        "NICs": [NIC,NIC,...]
    },
    .....
]
}
```

● 集群版

```
POST /rest/v1/block/target HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
    "targetName": targetName,
    "maxSessions": value,
    "reclaimPolicy": reclaimPolicy,
    "chap":{
        "name": name,
        "password": password,
        "status": status
    },
    "serverNumbers": value,
```

```
"server":["server_ID",...],
"allowlist": {
  "initiator": [
    {
      "IPs": [ip,ip,...],
      "names": [name,name,...]
    },
    {
      "IPs": [ip,ip,...],
      "names": [name,name,...]
    },
    .....
  ],
  "target": [
    {
      "IPs": [ip,ip,...],
      "NICs": [NIC,NIC,...]
    },
    {
      "IPs": [ip,ip,...],
      "NICs": [NIC,NIC,...]
    },
    .....
  ]
}
```

请求参数

参数	类型	描述	是否必须
targetName	String	iSCSI target 名称。 取值：长度范围是 1~16，可以由小写字母、数字、句点（.）和短横线（-）组成，且仅支持以	是

		字母或数字开头。	
maxSessions	Integer	iSCSI target 下每个 IQN 允许建立的最大会话数。 取值：取值范围是[0, 1024]，默认值为 1。0 表示客户端无法发现该 target。 注意：如果多个客户端连接同一 target IQN，客户端可以同时读，但不能同时写。	否
reclaimPolicy	String	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。	否
chap.name	String	CHAP 名称。 取值：长度范围是 3~64，可以由字母、数字、句点 (.)、短横线 (-)、下划线 (_) 和冒号 (:) 组成，字母区分大小写，且仅支持以字母或数字开头。	否
chap.password	String	CHAP 密码。 取值：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。	否
chap.status	String	是否启用 CHAP 认证。 取值：	否

		● Enabled: 启用 CHAP 认证。 ● Disabled: 禁用 CHAP 认证。 注意: 如果启用 CHAP 认证, name 和 password 必须设置。	
serverNumbers	Integer	target 所在的服务器数量 (仅集群版支持)。 取值: $[2, n]$, n 为集群内服务器的数量。默认值是 2。	否
server	Array of string	指定 iSCSI target 对应的服务器 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。	否
allowlist.initiator	Array of initiator	对 target 下所有 IQN 指定 iSCSI 发起方 (initiator) 允许访问列表。可以设置多组 initiator 允许访问列表, 各组允许访问列表之间为“或”的关系。每组允许访问列表可以同时指定 IP 和 initiator 名称, 二者为“与”的关系。详见“表 1 参数 initiator 说明”。 说明: 如果不设置, 则表示允许所有访问。	否
allowlist.target	Array of target	对 target 下所有 IQN 指定目标端 (target) 的允许访问列表。可以设置多组 target 允许访问列表, 各组允许访问列表之间为“或”的关系。每组允许访问列表可以同时指定 IP 和 NIC 名称, 二者为“与”的关系。详见“表 2 参数 target 说明”。 说明: 如果不设置, 则表示允许所有访问。	否

表 1 参数 initiator 说明

参数	类型	描述	是否必须
IPs	Array of ip	根据 initiator IP 地址设置 iSCSI 发起方的允许访问列表。 取值：IPv4、IPv6、CIDR 子网，可以设置多个，以英文逗号分隔。如果设置"IPs":[], 表示禁止所有发起端进行访问。 注意：IP 不能为 localhost。	否
names	Array of name	根据 initiator 名称设置 iSCSI 发起方的允许访问列表。 取值：字符串形式，长度范围是 1~223，只能由字母、数字、句点(.)、短横线(-)、冒号(:)组成，字母区分大小写。支持通配符*和?。可以设置多个，以英文逗号分隔。如果设置"names":[], 表示禁止所有发起端进行访问。	否

表 2 参数 target 说明

参数	类型	描述	是否必须
IPs	Array of ip	通过 target 端 IP 进行设置 target 允许访问列表，表示只允许通过 target 端的指定 IP 访问 target。 取值：IPv4、IPv6、CIDR 子网，可以设置多个，以英文逗号分隔。如果设置"IPs":[], 表示禁止通过 target 端进行访问。 注意：IP 不能为 localhost。	否
NICs	Array of NIC	通过 target 端的网卡名称设置 target 允许访问列表，表示只允许通过 target 端的指定网卡访问 target。 取值：target 端的网卡名称，可包含字母、数字、句点	否

		(.)、短横线(-)和下划线(_), 最多 100 个字符。支持通配符*和?。可以设置多个, 以英文逗号分隔。如果设置 "NICs": [], 表示禁止通过 target 端进行访问。	

响应结果

名称	类型	描述
iSCSITargets	Array of iSCSITarget	iSCSI target 属性集合, 详见“表 3 响应参数 iSCSITarget 说明”。

表 3 响应参数 iSCSITarget 说明

名称	类型	描述
iqn	String	iSCSI target IQN。
ips	Array of ip	iSCSI target 的 IP 属性集合, 详见“表 4 响应参数 ip 说明”。
allowlist.initiator	Array of initiator	iSCSI 发起方 (initiator) 允许访问列表。详见“表 5 响应参数 initiator 说明”。
allowlist.target	Array of target	目标端 (target) 的允许访问列表。详见“表 6 响应参数 target 说明”。

表 4 响应参数 ip 说明

名称	类型	描述
ip	String	iSCSI target 的 IP。
port	Integer	iSCSI target 的端口号。

表 5 响应参数 initiator 说明

名称	类型	描述
IPs	Array of ip	根据 initiator IP 地址设置的 iSCSI 发起方允许访问列表。

names	Array of name	根据 initiator 名称设置的 iSCSI 发起方允许访问列表。

表 6 响应参数 target 说明

名称	类型	描述
IPs	Array of ip	通过 target 端 IP 进行设置的 target 允许访问列表，表示只允许通过 target 端的指定 IP 访问 target。
NICs	Array of NIC	通过 target 端的网卡名称设置的 target 允许访问列表，表示只允许通过 target 端的指定网卡访问 target。

请求示例 1

单机版：创建一个名称为 **targete** 的 target。

```
POST /rest/v1/block/target HTTP/1.1
Date: Tue, 14 Feb 2023 07:58:44 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 155
Host: 192.168.0.32:1443

{
  "targetName": "targete",
  "chap": {
    "name": "test",
    "password": "12345678901a",
    "status": "Enabled"
  }
}
```

响应示例 1

```
HTTP/1.1 200 OK
x-hblock-request-id: 8c2a6566e6d84b5e82ac26387c6a50ac
Connection: keep-alive
```

```
Content-Length: 217
Date: Tue, 14 Feb 2023 07:58:44 GMT
Content-Type: application/json; charset=utf-8
Server: HBlock

{
  "data": {
    "iSCSITargets": [
      {
        "iqn": "iqn.2012-08.cn.ctyunapi.oos:targete.6",
        "ips": [
          {
            "ip": "192.168.0.32",
            "port": 3260
          }
        ]
      }
    ]
  }
}
```

请求示例 2

单机版：创建一个名称为 **targetf** 的 target。

```
POST /rest/v1/block/target HTTP/1.1
Date: Mon, 18 Aug 2025 02:25:34 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 571
Host: 192.168.0.66:1443

{
  "targetName": "targetf",
  "allowlist": {
    "initiator": [
```

```
{
  {
    "IPs": [
      "192.168.0.70"
    ],
    "names": [
      "iqn.1991-05.com.microsoft:songt-0001"
    ]
  },
  {
    "IPs": [
      "192.168.0.67"
    ]
  }
],
"target": [
  {
    "IPs": [
      "192.168.0.66"
    ]
  }
]
}
```

响应示例 2

```
HTTP/1.1 200 OK
x-hblock-request-id: ca3aa28f0807489fa4af85833c4de8a7
Connection: keep-alive
Content-Length: 280
Date: Mon, 18 Aug 2025 02:25:34 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
```

```
"iSCSITargets": [  
  {  
    "iqn": "iqn.2012-08.cn.ctyunapi.oos:targetf.4",  
    "ips": [  
      {  
        "ip": "192.168.0.66",  
        "port": 3260  
      }  
    ],  
    "allowlist": {  
      "initiator": [  
        {  
          "names": [  
            "iqn.1991-05.com.microsoft:songt-0001"  
          ],  
          "IPs": [  
            "192.168.0.70"  
          ]  
        },  
        {  
          "IPs": [  
            "192.168.0.67"  
          ]  
        }  
      ],  
      "target": [  
        {  
          "IPs": [  
            "192.168.0.66"  
          ]  
        }  
      ]  
    }  
  }  
]
```

```
}  
}
```

请求示例 3

集群版：创建一个名称为 **target09** 的 target。

```
POST /rest/v1/block/target HTTP/1.1  
Date: Mon, 18 Aug 2025 02:19:06 GMT  
Content-Type: application/json; charset=utf-8  
Authorization: HBlock userName:signature  
Content-Length: 910  
Host: 192.168.0.64:1443  
  
{  
  "targetName": "target09",  
  "maxSessions": 6,  
  "serverNumbers": 3,  
  "server": [  
    "hblock_1",  
    "hblock_3"  
  ],  
  "allowlist": {  
    "initiator": [  
      {  
        "IPs": [  
          "192.168.0.70"  
        ],  
        "names": [  
          "iqn.1991-05.com.microsoft:songt-0001"  
        ]  
      },  
      {  
        "IPs": [  
          "192.168.0.66"  
        ]  
      }  
    ]  
  }  
}
```

```
    }
  ],
  "target": [
    {
      "IPs": [
        "192.168.0.64",
        "192.168.0.67"
      ],
    },
    {
      "IPs": [
        "192.168.0.65"
      ],
      "NICs": [
        "eth0"
      ]
    }
  ]
}
```

响应示例 3

```
HTTP/1.1 200 OK
x-hblock-request-id: 1f7fe9f8cd054d629a88f618fac0a878
Connection: keep-alive
Content-Length: 960
Date: Mon, 18 Aug 2025 02:19:06 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "iSCSITargets": [
      {
        "iqn": "iqn.2012-08.cn.ctyunapi.oos:target09.17",
```

```
"ips": [
  {
    "ip": "192.168.0.64",
    "port": 3260
  }
],
"allowlist": {
  "initiator": [
    {
      "names": [
        "iqn.1991-05.com.microsoft:songt-0001"
      ],
      "IPs": [
        "192.168.0.70"
      ]
    },
    {
      "IPs": [
        "192.168.0.66"
      ]
    }
  ],
  "target": [
    {
      "IPs": [
        "192.168.0.64",
        "192.168.0.67"
      ]
    },
    {
      "IPs": [
        "192.168.0.65"
      ],
      "NICs": [
        "eth0"
      ]
    }
  ]
}
```



```
        ]
      }
    ]
  }
},
{
  "iqn": "iqn.2012-08.cn.ctyunapi.oos:target09.18",
  "ips": [
    {
      "ip": "192.168.0.65",
      "port": 3260
    }
  ],
  "allowlist": {
    "initiator": [
      {
        "names": [
          "iqn.1991-05.com.microsoft:songt-0001"
        ],
        "IPs": [
          "192.168.0.70"
        ]
      },
      {
        "IPs": [
          "192.168.0.66"
        ]
      }
    ],
    "target": [
      {
        "IPs": [
          "192.168.0.64",
          "192.168.0.67"
        ]
      }
    ]
  }
}
```

```
        },
        {
            "IPs": [
                "192.168.0.65"
            ],
            "NICs": [
                "eth0"
            ]
        }
    ]
},
{
    "iqn": "iqn.2012-08.cn.ctyunapi.oos:target09.19",
    "ips": [
        {
            "ip": "192.168.0.67",
            "port": 3260
        }
    ],
    "allowlist": {
        "initiator": [
            {
                "names": [
                    "iqn.1991-05.com.microsoft:songt-0001"
                ],
                "IPs": [
                    "192.168.0.70"
                ]
            },
            {
                "IPs": [
                    "192.168.0.66"
                ]
            }
        ]
    }
}
```

```
    ],  
    "target": [  
      {  
        "IPs": [  
          "192.168.0.64",  
          "192.168.0.67"  
        ],  
      },  
      {  
        "IPs": [  
          "192.168.0.65"  
        ],  
        "NICs": [  
          "eth0"  
        ],  
      }  
    ],  
  }  
}  
]  
}
```

4.7.2 修改 iSCSI target 的 CHAP 认证

此操作用来修改 iSCSI target 的 CHAP 认证。

请求语法

```
PUT /rest/v1/block/target/targetName HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "chap": {
    "name": name,
    "password": password,
    "status": status
  }
}
```

请求参数

参数	类型	描述	是否必须
targetName	String	iSCSI target 名称。 取值：长度范围是 1~16，可以由小写字母、数字、句点（.）和短横线（-）组成，且仅支持以字母或数字开头。	是
chap.name	String	CHAP 名称。 取值：长度范围是 3~64，可以由字母、数字、句点（.）、短横线（-）、下划线（_）和冒号（:）组成，字母区分大小写，且仅支持以字母或数字开头。	否
chap.password	String	CHAP 密码。	否

		取值：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。	
chap.status	String	是否启用 CHAP 认证。 取值： ● Enabled: 启用 CHAP 认证。 ● Disabled: 禁用 CHAP 认证。 注意：如果是首次启用 CHAP 认证，name 和 password 必须设置。	否

请求示例 1

修改 target2：启用 CHAP 认证，并设置 CHAP 认证的名称和密码。

```
PUT /rest/v1/block/target/target2 HTTP/1.1
Date: Fri, 29 Jul 2022 06:17:27 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 85
Host: 192.168.0.121:1443

{
  "chap":{
    "name":"test",
    "password":"*****",
    "status":"Enabled"
  }
}
```

响应示例 1

```
HTTP/1.1 204 No Content
x-hblock-request-id: b18c4215a42a460789fb54879acfa3e3
Connection: keep-alive
```

```
Date: Fri, 29 Jul 2022 06:17:27 GMT
```

```
Server: HBlock
```

请求示例 2

修改 **target2**: 禁止 CHAP 认证，但不删除先前 CHAP 认证的名称和密码。

```
PUT /rest/v1/block/target/target2 HTTP/1.1
Date: Fri, 29 Jul 2022 06:17:53 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 60
Host: 192.168.0.121:1443
{
  "chap":{
    "status":"Disabled"
  }
}
```

响应示例 2

```
HTTP/1.1 204 No Content
x-hblock-request-id: 850b187b2d844b928998a190b8a034f3
Connection: keep-alive
Date: Fri, 29 Jul 2022 06:17:53 GMT
Server: HBlock
```

4.7.3 迁移 iSCSI target（集群版适用）

此操作用来迁移 iSCSI target，修改 iSCSI target 对应的服务器。

说明：一次只允许迁移 iSCSI target 对应的一个服务器。

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 可能会造成数据丢失。
- 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。
- 迁移完成后，请检查并调整 target 允许访问列表配置，确保符合访问控制要求。

请求语法

```
PUT /rest/v1/block/target/targetName/migrate?force=force HTTP/1.1
```

Date: *date*

Content-Type: application/json; charset=utf-8

Content-Length: *length*

Host: *ip:port*

Authorization: *authorization*

```
{
  "server": [
    {
      "source": source_server_ID,
      "dest": dest_server_ID
    }
  ]
}
```

请求参数

参数	类型	描述	是否必须
targetName	String	iSCSI target 名称。 取值：长度范围是 1~16，可以由小写字母、数字、句点（.）和短横线（-）组成，且仅支持以字母或数字开头。	是
force	Boolean	是否强制迁移 iSCSI target。 注意： 目前仅支持强制迁移 iSCSI target，强制迁移 iSCSI target 可能会造成数据丢失。 取值： true ：强制迁移 iSCSI target。	是
server	Array of server	迁移的源和目的服务器集合，详见“表 1 请求参数 server 说明”。	是

表 1 请求参数 server 说明

参数	类型	描述	是否必须
source	String	iSCSI target 的源服务器 ID。	是
dest	String	iSCSI target 的目标服务器 ID。	是

响应结果

名称	类型	描述
luns	Array of lun	卷的属性集合，详见“表 2 响应参数 lun 说明”。

表 2 响应参数 lun 说明

名称	类型	描述
lunName	String	关联卷的名称。

iSCSITargets	Array of iSCSITarget	iSCSI target 属性集合，详见“表 3 响应参数 iSCSITarget 说明”。

表 3 响应参数 iSCSITarget 说明

名称	类型	描述
iqn	String	iSCSI target IQN。
status	String	卷对应的 target 的状态： ● Active: 主 target。 ● Standby: 热备 target。 ● Offline: 离线 target。 ● Unavailable: 冷备 target。
ips	Array of ip	iSCSI target 的 IP 属性集合，详见“表 4 响应参数 ip 说明”。
targetPortalIP	Object	iSCSI 目标门户，详见“表 5 响应参数 targetPortalIP 说明”。

表 4 响应参数 ip 说明

名称	类型	描述
ip	String	iSCSI target 的 IP。
port	Integer	iSCSI target 的端口号。

表 5 响应参数 targetPortalIP 说明

名称	类型	描述
ips	Array of ip	iSCSI 目标门户的 IP 属性集合，详见“表 6 响应参数 targetPortalIP.ip 说明”。
status	String	iSCSI 目标门户的状态： ● Enabled: 启用状态。

		● Disabled: 禁用状态。

表 6 响应参数 `targetPortalIP.ip` 说明

名称	类型	描述
ip	String	iSCSI 目标门户的 IP 地址。
port	Integer	iSCSI 目标门户的端口号。

请求示例

迁移 `target01`。

<pre>PUT /rest/v1/block/target/target01/migrate?force=true HTTP/1.1 Date: Thu, 4 Jul 2024 03:23:39 GMT Content-Type: application/json; charset=utf-8 Authorization: HBlock <i>userName:signature</i> Content-Length: 95 Host: 192.168.0.192:1443 { "server": [{ "source": "hblock_1", "dest": "hblock_2" }] }</pre>

响应示例

<pre>HTTP/1.1 200 OK x-hblock-request-id: 4e88a225b1114c4ca3f2611e336ac431 Connection: keep-alive Content-Type: application/json; charset=utf-8 Content-Length: 282 Date: Thu, 4 Jul 2024 03:23:39 GMT</pre>

Server: HBlock

```
{
  "data": {
    "luns": [
      {
        "lunName": "lun01a",
        "iSCSITargets": [
          {
            "iqn": "iqn.2012-08.cn.ctyunapi.oos:target01.1",
            "status": "Active",
            "ips": [
              {
                "ip": "192.168.0.102",
                "port": 3260
              }
            ]
          },
          {
            "iqn": "iqn.2012-08.cn.ctyunapi.oos:target01.2",
            "status": "Unavailable",
            "ips": [
              {
                "ip": "192.168.0.192",
                "port": 3260
              }
            ]
          }
        ]
      }
    ]
  }
}
```

说明：卷关联的 iSCSI target 处于 **Unavailable** 状态，表示 iSCSI 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 iSCSI target 处于 **Unavailable** 状态。迁移完成后，iSCSI target 将变为 **Active** 或者 **Standby** 状态。

4.7.4 修改 iSCSI target 下每个 IQN 允许建立的最大会话数

此操作用来修改 iSCSI target 下每个 IQN 允许建立的最大会话数。

请求语法

```
PUT /rest/v1/block/target/targetName HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "maxSessions": value
}
```

请求参数

参数	类型	描述	是否必须
targetName	String	iSCSI target 名称。 取值：长度范围是 1~16，可以由小写字母、数字、句点（.）和短横线（-）组成，且仅支持以字母或数字开头。	是
maxSessions	Integer	iSCSI target 下每个 IQN 允许建立的最大会话数。 取值：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 target IQN，客户端可以同时读，但不能同时写。	否

请求示例

将 target06 下每个 IQN 允许建立的最大会话数修改为 11。

```
PUT /rest/v1/block/target/target06 HTTP/1.1
Date: Thu, 28 Sep 2023 07:53:46 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 28
Host: 192.168.0.117:1443

{
  "maxSessions": 11
}
```

响应示例

```
HTTP/1.1 204 No Content
x-hblock-request-id: a0a69bafb19a4371a978fc4bcfee48ac
Connection: keep-alive
Date: Thu, 28 Sep 2023 07:53:46 GMT
Server: HBlock
```

4.7.5 修改 iSCSI target 的回收策略

此操作用来修改 iSCSI target 的回收策略。

请求语法

```
PUT /rest/v1/block/target/targetName HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "reclaimPolicy": reclaimPolicy
}
```

请求参数

参数	类型	描述	是否必须
targetName	String	iSCSI target 名称。 取值：长度范围是 1~16，可以由小写字母、数字、句点（.）和短横线（-）组成，且仅支持以字母或数字开头。	是
reclaimPolicy	String	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。	否

请求示例

将 target01 的回收策略修改为 Delete。

```
PUT /rest/v1/block/target/target01 HTTP/1.1
Date: Thu, 28 Nov 2024 09:09:51 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 34
Host: 192.168.0.117:1443

{
  "reclaimPolicy": "Delete"
}
```

响应示例

```
HTTP/1.1 204 No Content
x-hblock-request-id: 304414616c994deabd10e3a39cc3802b
Connection: keep-alive
Date: Thu, 28 Nov 2024 09:09:51 GMT
Server: HBlock
```

4.7.6 设置 iSCSI target 的允许访问列表

此操作用来设置 iSCSI target 的允许访问列表。

请求语法

```
PUT /rest/v1/block/target/targetName/allowlist HTTP/1.1
```

Date: *date*

Content-Type: application/json; charset=utf-8

Content-Length: *length*

Host: *ip:port*

Authorization: *authorization*

```
{
  "iqns": [iqn,iqn,...],
  "action": action,
  "initiator": [
    {
      "IPs": [ip,ip,...],
      "names": [name,name,...]
    },
    {
      "IPs": [ip,ip,...],
      "names": [name,name,...]
    },
    .....
  ],
  "target": [
    {
      "IPs": [ip,ip,...],
      "NICs": [NIC,NIC,...]
    },
    {
      "IPs": [ip,ip,...],
      "NICs": [NIC,NIC,...]
    }
  ]
}
```



```

    },
    .....
  ]
}

```

请求参数

参数	类型	描述	是否必须
targetName	String	iSCSI target 名称。 取值：长度范围是 1~16，可以由小写字母、数字、句点 (.) 和短横线 (-) 组成，且仅支持以字母或数字开头。	是
iqns	Array of string	指定 target IQN 名称。 如果指定了 target IQN 名称，则代表对 target 下指定的 IQN 置访问权限；如果未指定 target IQN 名称，则代表对 target 下所有 IQN 进行设置访问权限。	否
action	String	设置 iSCSI target 的访问权限。 取值： ● add: 添加一组或多组允许访问列表。 ● delete: 删除一组或多组允许访问列表。 ● replace: 替换现有允许访问列表。	是
initiator	Array of initiator	指定 iSCSI 发起方 (initiator) 允许访问列表。可以设置多组 initiator 允许访问列表，各组允许访问列表之间为“或”的关系。每组允许访问列表可以同时指定 IP 和 initiator 名称，二者为“与”的关系。详见“表 1 参数 initiator 说明”。	否

		注意： ● 设置"initiator":[]，表示不能被任何客户端发现。 ● initiator 和 target 至少填一个。	
target	Array of target	指定目标端（target）的允许访问列表。可以设置多组 target 允许访问列表，各组允许访问列表之间为“或”的关系。每组允许访问列表可以同时指定 IP 和 NIC 名称，二者为“与”的关系。详见“表 2 参数 target 说明”。 注意： ● 设置"target":[]，表示不能被任何客户端发现。 ● initiator 和 target 至少填一个。	否

表 1 参数 initiator 说明

参数	类型	描述	是否必须
IPs	Array of string	根据 initiator IP 地址设置 iSCSI 发起方的允许访问列表。 取值：IPv4、IPv6、CIDR 子网，可以设置多个，以英文逗号分隔。如果设置"IPs":[]，表示禁止所有发起端进行访问。 注意：IP 不能为 localhost。	否
names	Array of string	根据 initiator 名称设置 iSCSI 发起方的允许访问列表。 取值：字符串形式，长度范围是 1~223，只能由字母、数字、句点(.)、短横线(-)、冒号(:)组成，字母区分大小写。支持通配符*和?。可以设置多个，以英文逗号	否

		分隔。如果设置"names":[], 表示禁止所有发起端进行访问。	

表 2 参数 target 说明

参数	类型	描述	是否必须
IPs	Array of string	通过 target 端 IP 进行设置 target 允许访问列表，表示只允许通过 target 端的指定 IP 访问 target。 取值：IPv4、IPv6、CIDR 子网，可以设置多个，以英文逗号分隔。如果设置"IPs":[], 表示禁止通过 target 端进行访问。 注意： IP 不能为 localhost。	否
NICs	Array of string	通过 target 端的网卡名称设置 target 允许访问列表，表示只允许通过 target 端的指定网卡访问 target。 取值：target 端的网卡名称，可包含字母、数字、句点(.)、短横线(-)和下划线(_)，最多 100 个字符。支持通配符*和?。可以设置多个，以英文逗号分隔。如果设置"NICs":[], 表示禁止通过 target 端进行访问。	否

请求示例

为 target07 增加允许访问列表。

```
PUT /rest/v1/block/target/target07/allowlist HTTP/1.1
Date: Mon, 18 Aug 2025 03:03:52 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 636
Host: 192.168.0.64:1443

{
```

```
{
  "action": "add",
  "initiator": [
    {
      "IPs": [
        "192.168.0.70"
      ],
      "names": [
        "iqn.1991-05.com.microsoft:songt-0001"
      ]
    },
    {
      "IPs": [
        "192.168.0.66"
      ]
    }
  ],
  "target": [
    {
      "IPs": [
        "192.168.0.64",
        "192.168.0.67"
      ]
    },
    {
      "IPs": [
        "192.168.0.65"
      ],
      "NICs": [
        "eth0"
      ]
    }
  ]
}
```

响应示例

```
HTTP/1.1 204 No Content
x-hblock-request-id: db9d71cdd4c342ea895cd71efe66482a
Connection: keep-alive
Date: Mon, 18 Aug 2025 03:03:52 GMT
Server: HBlock
```

4.7.7 删除 iSCSI target 的允许访问列表

此操作用来删除 iSCSI target 的允许访问列表。

请求语法

```
DELETE /rest/v1/block/target/targetName/allowlist HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "iqns": [iqn,iqn,...]
}
```

请求参数

参数	类型	描述	是否必须
targetName	String	iSCSI target 名称。 取值：长度范围是 1~16，可以由小写字母、数字、句点（.）和短横线（-）组成，且仅支持以字母或数字开头。	是
iqns	Array of string	指定 target IQN 名称。 如果指定了 target IQN 名称，则代表删除 target 下指定 IQN 的允许访问列表；如果未指定 target IQN 名称，则代表删除 target 下所有 IQN 的允许访问列表。	否

请求示例

删除 target07 下 iqn.2012-08.cn.ctyunapi.oos:target07.13 的允许访问列表。

```
DELETE /rest/v1/block/target/target07/allowlist HTTP/1.1
```

```
Date: Thu, 24 Jul 2025 08:09:58 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 77
Host: 192.168.0.64:1443
{
  "iqns": [
    "iqn.2012-08.cn.ctyunapi.oos:target07.13"
  ]
}
```

响应示例

```
HTTP/1.1 204 No Content
x-hblock-request-id: 273048935a3e4f10a1a3a07a1e54bbdb
Connection: keep-alive
Date: Thu, 24 Jul 2025 08:09:58 GMT
Server: HBlock
```

4.7.8 删除 iSCSI target

此操作用来删除 iSCSI target。

注意：只有未关联任何卷的 iSCSI target 才能被删除。

请求语法

```
DELETE /rest/v1/block/target/targetName HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

请求参数

参数	类型	描述	是否必须
targetName	String	iSCSI target 名称。 取值：长度范围是 1~16，可以由小写字母、数字、句点（.）和短横线（-）组成，且仅支持以字母或数字开头。	是

请求示例

删除名为 target2 的 target。

```
DELETE /rest/v1/block/target/target2 HTTP/1.1
Date: Wed, 16 Mar 2022 01:40:22 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.121:1443
```

响应示例

```
HTTP/1.1 204 No Content
x-hblock-request-id: c9ca51dddbbd4cafa077fa86180a8e51
Connection: keep-alive
Date: Wed, 16 Mar 2022 01:40:22 GMT
```


Server: HBlock

4.7.9 删除 CHAP

此操作用来删除 CHAP 名称和密码。

请求语法

```
DELETE /rest/v1/block/target/targetName/chap HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

请求参数

参数	类型	描述	是否必须
targetName	String	iSCSI target 名称。 取值：长度范围是 1~16，可以由小写字母、数字、句点（.）和短横线（-）组成，且仅支持以字母或数字开头。	是

请求示例

删除 target2 的 CHAP 名称和密码。

```
DELETE /rest/v1/block/target/target2/chap HTTP/1.1
Date: Wed, 16 Mar 2022 01:25:42 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.121:1443
```

响应示例

```
HTTP/1.1 204 No Content
x-hblock-request-id: aa3fa33cdeba42ffa888392f92500772
Connection: keep-alive
Date: Wed, 16 Mar 2022 01:25:43 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock
```

4.7.10 批量查询 iSCSI target

此操作用来批量查询 iSCSI target。

请求语法

```
GET /rest/v1/block/target?filter=filter&range=i-j HTTP/1.1
```

Date: *date*

Host: *ip:port*

Authorization: *authorization*

请求参数

参数	类型	描述	是否必须
filter	String	查询的过滤条件。 过滤条件由查询属性和属性值组成，其中属性和属性值之间用单个冒号分隔（key:value）表示模糊匹配，用两个冒号分隔（key::value）表示精确匹配。可以选择多个查询，如果是或的关系，使用“or”将查询条件分隔开；如果是与的关系，使用“and”将查询条件分隔开。 支持的过滤条件类型包括： ● lunName: iSCSI target 关联的卷名称。 ● targetName: iSCSI target 名称。 ● iqn: iSCSI target IQN。	否
range	String	查询 iSCSI target 的范围，格式为 i-j，i 和 j 为正整数，0<j-i<1000。按照 target 名称进行排序，返回第 i 到第 j 的 target。 如果输入的查询范围超过 iSCSI target 的总个数，返回的结果为空，如果未指定 range，则返回所有符合条件的 iSCSI target。	否

响应结果

名称	类型	描述
targets	Array of target	iSCSI target 信息集合，详见“表 1 响应参数 target 说明”。

表 1 响应参数 target 说明

名称	类型	描述
targetName	String	iSCSI target 名称。
maxSessions	Integer	iSCSI target 下每个 IQN 允许建立的最大会话数。
iSCSITargets	Array of iSCSITarget	iSCSI target 属性集合，详见“表 2 响应参数 iSCSITarget 说明”。
chap.name	String	CHAP 认证名称。
chap.status	String	CHAP 认证状态： ● Enabled: 启用 CHAP 认证。 ● Disabled: 禁用 CHAP 认证。
createTime	Long	创建 iSCSI target 的时间，unix 时间戳（UTC），精确到毫秒。
serverIds	Array of string	iSCSI target 对应的服务器 ID（仅集群版支持）。
num	Integer	target 所在的服务器数量（仅集群版支持）。
reclaimPolicy	String	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。
status	String	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。

表 2 响应参数 iSCSItarget 说明

名称	类型	描述
iqn	String	iSCSI target IQN。
ips	Array of ip	iSCSI target 的 IP 属性集合，详见“表 3 响应参数 ip 说明”。
targetPortalIP	Object	iSCSI 目标门户，详见“表 4 响应参数 targetPortalIP 说明”。
serverId	String	iSCSI target 的 IQN 所在的服务器 ID。
allowlist.initiator	Array of initiator	iSCSI 发起方（initiator）允许访问列表。详见“表 6 响应参数 initiator 说明”。
allowlist.target	Array of target	目标端（target）的允许访问列表。详见“表 7 响应参数 target 说明”。

表 3 响应参数 ip 说明

名称	类型	描述
ip	String	iSCSI target 的 IP。
port	Integer	iSCSI target 的端口号。

表 4 响应参数 targetPortalIP 说明

名称	类型	描述
ips	Array of ip	iSCSI 目标门户的 IP 属性集合，详见“表 5 响应参数 targetPortalIP.ip 说明”。
status	String	iSCSI 目标门户的状态： ● Enabled: 启用状态。 ● Disabled: 禁用状态。 说明：如果未配置，不返回该项。

表 5 响应参数 **targetPortalIP.ip** 说明

名称	类型	描述
ip	String	iSCSI 目标门户的 IP 地址。
port	Integer	iSCSI 目标门户的端口号。

表 6 响应参数 **initiator** 说明

参数	类型	描述
IPs	Array of ip	根据 initiator IP 地址设置的 iSCSI 发起方允许访问列表。
names	Array of name	根据 initiator 名称设置的 iSCSI 发起方允许访问列表。

表 7 响应参数 **target** 说明

参数	类型	描述
IPs	Array of ip	通过 target 端 IP 进行设置的 target 允许访问列表，表示只允许通过 target 端的指定 IP 访问 target。
NICs	Array of NIC	通过 target 端的网卡名称设置的 target 允许访问列表，表示只允许通过 target 端的指定网卡访问 target。

请求示例 1

单机版：查询 iSCSI target 名称中包含 **target**，且关联的卷名称包含 **lun** 的 iSCSI target，且只列出第 1 个到第 3 个的查询结果。

```
GET /rest/v1/block/target?filter=targetName:target and lunName:lun&range=1-3 HTTP/1.1
Date: Thu, 24 Jul 2025 08:16:12 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.66:1443
```

响应示例 1

```
HTTP/1.1 200 OK
```

x-hblock-request-id: c5271562a56f4805a9ce2f39214953ab

Connection: keep-alive

Content-Length: 874

Date: Thu, 24 Jul 2025 08:16:12 GMT

Content-Type: application/json; charset=utf-8

Server: HBlock

```
{
  "data": {
    "targets": [
      {
        "targetName": "targeta",
        "maxSessions": 3,
        "iSCSITargets": [
          {
            "iqn": "iqn.2012-08.cn.ctyunapi.oos:targeta.1",
            "ips": [
              {
                "ip": "192.168.0.66",
                "port": 3260
              }
            ],
            "serverId": "hblock_1",
            "allowlist": {
              "initiator": [
                {
                  "IPs": [
                    "192.168.0.68"
                  ]
                },
                {
                  "IPs": [
                    "192.168.0.70"
                  ]
                }
              ]
            }
          }
        ]
      }
    ]
  }
}
```

```
        ],
        "target": [
            {
                "IPs": [
                    "192.168.0.66"
                ]
            }
        ]
    }
},
"chap": {
    "name": "chanptest",
    "status": "Disabled"
},
"createTime": 1753172337443,
"num": 1,
"reclaimPolicy": "Delete"
},
{
    "targetName": "targetb",
    "maxSessions": 1,
    "iSCSITargets": [
        {
            "iqn": "iqn.2012-08.cn.ctyunapi.oos:targetb.2",
            "ips": [
                {
                    "ip": "192.168.0.66",
                    "port": 3260
                }
            ],
            "serverId": "hbblock_1"
        }
    ],
    "createTime": 1753237554317,
```



```
        "num": 1,
        "reclaimPolicy": "Delete"
    },
    {
        "targetName": "targete",
        "maxSessions": 1,
        "iSCSITargets": [
            {
                "iqn": "iqn.2012-08.cn.ctyunapi.oos:targete.3",
                "ips": [
                    {
                        "ip": "192.168.0.66",
                        "port": 3260
                    }
                ],
                "serverId": "hblock_1"
            }
        ],
        "createTime": 1753324422341,
        "num": 1,
        "reclaimPolicy": "Retain"
    }
]
}
```

请求示例 2

集群版：查询 iSCSI target 名称中包含 **target0**，且关联的卷名称包含 **lun0** 的 iSCSI target，且只列出第 1 个到第 3 个的查询结果。

```
GET /rest/v1/block/target?filter=targetName:target0 and lunName:lun0&range=1-3 HTTP/1.1
Date: Thu, 24 Jul 2025 08:24:26 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.64:1443
```

响应示例 2

```
HTTP/1.1 200 OK
x-hblock-request-id: d820500731e04a03a2477fb09e2fd3fe
Connection: keep-alive
Content-Length: 1942
Date: Thu, 24 Jul 2025 08:24:26 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "targets": [
      {
        "targetName": "target0",
        "maxSessions": 1,
        "iSCSITargets": [
          {
            "iqn": "iqn.2012-08.cn.ctyunapi.oos:target0.1",
            "ips": [
              {
                "ip": "192.168.0.67",
                "port": 3260
              }
            ],
            "serverId": "hblock_3",
            "allowlist": {
              "initiator": [
                {
                  "IPs": [
                    "1.1.1.1"
                  ]
                }
              ]
            }
          }
        ]
      }
    ],
  },
}
```

```
{
  {
    "iqn": "iqn.2012-08.cn.ctyunapi.oos:target0.2",
    "ips": [
      {
        "ip": "192.168.0.64",
        "port": 3260
      }
    ],
    "serverId": "hblock_2",
    "allowlist": {
      "initiator": [
        {
          "IPs": [
            "1.1.1.1"
          ]
        }
      ]
    }
  },
  "createTime": 1752111608672,
  "serverIds": [
    "hblock_2",
    "hblock_3"
  ],
  "num": 2,
  "reclaimPolicy": "Delete"
},
{
  "targetName": "target01",
  "maxSessions": 1,
  "iSCSITargets": [
    {
      "iqn": "iqn.2012-08.cn.ctyunapi.oos:target01.3",
      "ips": [
```

```
        {
            "ip": "192.168.0.65",
            "port": 3260
        }
    ],
    "serverId": "hblock_1"
},
{
    "iqn": "iqn.2012-08.cn.ctyunapi.oos:target01.4",
    "ips": [
        {
            "ip": "192.168.0.67",
            "port": 3260
        }
    ],
    "serverId": "hblock_3",
    "allowlist": {
        "target": [
            {
                "IPs": [
                    "192.168.0.67"
                ]
            }
        ]
    }
}
],
"createTime": 1752111636790,
"serverIds": [
    "hblock_1",
    "hblock_3"
],
"num": 2,
"reclaimPolicy": "Delete"
},
```

```
{
  "targetName": "target03",
  "maxSessions": 2,
  "iSCSITargets": [
    {
      "iqn": "iqn.2012-08.cn.ctyunapi.oos:target03.5",
      "ips": [
        {
          "ip": "192.168.0.67",
          "port": 3260
        }
      ],
      "serverId": "hbblock_3",
      "allowlist": {
        "initiator": [
          {
            "names": [
              "iqn.1991-05.com.microsoft:songt-0001"
            ],
            "IPs": [
              "192.168.0.70"
            ]
          },
          {
            "IPs": [
              "192.168.0.66"
            ]
          }
        ],
        "target": [
          {
            "IPs": [
              "192.168.0.64"
            ]
          },
        ],
      }
    }
  ]
}
```

```
        {
            "IPs": [
                "192.168.0.65"
            ]
        },
        {
            "IPs": [
                "192.168.0.67"
            ]
        },
        {
            "IPs": [
                "192.168.0.64",
                "192.168.0.65",
                "192.168.0.67"
            ]
        },
        {
            "IPs": [
                "192.168.0.71"
            ]
        }
    ]
},
{
    "iqn": "iqn.2012-08.cn.ctyunapi.oos:target03.6",
    "ips": [
        {
            "ip": "192.168.0.65",
            "port": 3260
        }
    ],
    "serverId": "hblock_1",
    "allowlist": {
```

```
"initiator": [  
  {  
    "names": [  
      "iqn.1991-05.com.microsoft:songt-0001"  
    ],  
    "IPs": [  
      "192.168.0.70"  
    ]  
  },  
  {  
    "IPs": [  
      "192.168.0.66"  
    ]  
  }  
],  
"target": [  
  {  
    "IPs": [  
      "192.168.0.64"  
    ]  
  },  
  {  
    "IPs": [  
      "192.168.0.65"  
    ]  
  },  
  {  
    "IPs": [  
      "192.168.0.67"  
    ]  
  },  
  {  
    "IPs": [  
      "192.168.0.64",  
      "192.168.0.65",  
      "192.168.0.66",  
      "192.168.0.67"  
    ]  
  }  
]
```

```
        "192.168.0.67"
      ]
    },
    {
      "IPs": [
        "192.168.0.71"
      ]
    }
  ]
}
},
],
"chap": {
  "name": "chap-test",
  "status": "Disabled"
},
"createTime": 1752111720420,
"serverIds": [
  "hblock_1",
  "hblock_3"
],
"num": 2,
"reclaimPolicy": "Delete"
}
]
}
}
```


4.7.11 查询指定 iSCSI target

此操作用来查询指定的 iSCSI target。

请求语法

```
GET /rest/v1/block/target/targetName HTTP/1.1  
Date: date  
Host: ip:port  
Authorization: authorization
```

请求参数

参数	类型	描述	是否必须
targetName	String	iSCSI target 名称。 说明：如果不填写此项，会返回所有 target 信息。 取值：长度范围是 1~16，可以由小写字母、数字、句点（.）和短横线（-）组成，且仅支持以字母或数字开头。	否

响应结果

名称	类型	描述
targetName	String	iSCSI target 名称。
maxSessions	Integer	iSCSI target 下每个 IQN 允许建立的最大会话数。
iSCSITargets	Array of iSCSItarget	iSCSI target 属性集合，详见“表 1 响应参数 iSCSItarget 说明”。
luns	Array of lun	关联卷的属性，详见“表 7 响应参数 lun 说明”。
chap.name	String	CHAP 认证名称。
chap.status	String	CHAP 认证状态：

		● Enabled: 启用 CHAP 认证。 ● Disabled: 禁用 CHAP 认证。
createTime	Long	创建 iSCSI target 的时间，unix 时间戳（UTC），精确到毫秒。
serverIds	Array of string	iSCSI target 对应的服务器 ID（仅集群版支持）。
num	Integer	target 所在的服务器数量（仅集群版支持）。
reclaimPolicy	String	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。
status	String	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。

表 1 响应参数 iSCSITarget 说明

名称	类型	描述
iqn	String	iSCSI target IQN。
ips	Array of ip	iSCSI target 的 IP 属性集合，详见“表 2 响应参数 ip 说明”。
targetPortalIP	Object	iSCSI 目标门户，详见“表 3 响应参数 targetPortalIP 说明”。
serverId	String	iSCSI target 的 IQN 所在的服务器 ID。
allowlist.initiator	Array of initiator	iSCSI 发起方（initiator）允许访问列表。详见“表 5 响应参数 initiator 说明”。
allowlist.target	Array of	目标端（target）的允许访问列表。详见“表 6 响

	target	应参数 target 说明”。

表 2 响应参数 **ip** 说明

名称	类型	描述
ip	String	iSCSI target 的 IP。
port	Integer	iSCSI target 的端口号。

表 3 响应参数 **targetPortalIP** 说明

名称	类型	描述
ips	Array of ip	iSCSI 目标门户的 IP 属性集合，详见“表 4 响应参数 targetPortalIP.ip 说明”。
status	String	iSCSI 目标门户的状态： ● Enabled: 启用状态。 ● Disabled: 禁用状态。 说明：如果未配置，不返回该项。

表 4 响应参数 **ip** (iSCSI 目标门户的 IP) 说明

名称	类型	描述
ip	String	iSCSI 目标门户的 IP 地址。
port	Integer	iSCSI 目标门户的端口号。

表 5 响应参数 **initiator** 说明

名称	类型	描述
IPs	Array of ip	根据 initiator IP 地址设置的 iSCSI 发起方允许访问列表。
names	Array of name	根据 initiator 名称设置的 iSCSI 发起方允许访问列表。

表 6 响应参数 **target** 说明

名称	类型	描述
IPs	Array of ip	通过 target 端 IP 进行设置的 target 允许访问列表，表示只允许通过 target 端的指定 IP 访问 target。
NICs	Array of NIC	通过 target 端的网卡名称设置的 target 允许访问列表，表示只允许通过 target 端的指定网卡访问 target。

表 7 响应参数 lun 说明

名称	类型	描述
lunName	String	关联卷的名称。
number	Long	卷编号。

请求示例 1

单机版：查询 iSCSI target 名称为 **targeta** 的信息。

```
GET /rest/v1/block/target/targeta HTTP/1.1
Date: Mon, 08 Sep 2025 01:29:15 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.32:1443
Connection: keep-alive
```

响应示例 1

```
HTTP/1.1 200 OK
x-hblock-request-id: e66f2a16825240009d3de70b158ad161
Connection: keep-alive
Content-Length: 478
Date: Mon, 08 Sep 2025 01:29:15 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "targetName": "targeta",
    "maxSessions": 3,
```

```
"iSCSITargets": [  
  {  
    "iqn": "iqn.2012-08.cn.ctyunapi.oos:targeta.1",  
    "ips": [  
      {  
        "ip": "192.168.0.66",  
        "port": 3260  
      }  
    ],  
    "serverId": "hblock_1",  
    "allowlist": {  
      "initiator": [  
        {  
          "names": [  
            "iqn.1994-05.com.redhat:c73bf5e45bd4"  
          ],  
          "IPs": [  
            "192.168.0.68"  
          ]  
        },  
        {  
          "IPs": [  
            "192.168.0.70"  
          ]  
        }  
      ],  
      "target": [  
        {  
          "IPs": [  
            "192.168.0.66"  
          ]  
        }  
      ]  
    }  
  }  
],  
"luns": [  
  {  
    "lunName": "luna1",  
    "number": 0  
  },  
]
```

```
{
  {
    "lunName": "luna1-clone1",
    "number": 1
  }
],
"createTime": 1754375366254,
"num": 1,
"reclaimPolicy": "Delete"
}
```

请求示例 2

集群版：查询 iSCSI target 名称为 **target03** 的信息。

```
GET /rest/v1/block/target/target03 HTTP/1.1
Date: Thu, 24 Jul 2025 08:36:15 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.64:1443
Connection: keep-alive
```

响应示例 2

```
HTTP/1.1 200 OK
x-hblock-request-id: b859187a66164328b538bc421c2861da
Connection: keep-alive
Content-Length: 1096
Date: Thu, 24 Jul 2025 08:36:15 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "targetName": "target03",
    "maxSessions": 2,
    "iSCSITargets": [
      {
        "iqn": "iqn.2012-08.cn.ctyunapi.oos:target03.5",
```

```
"ips": [  
  {  
    "ip": "192.168.0.67",  
    "port": 3260  
  }  
],  
"serverId": "hblock_3",  
"allowlist": {  
  "initiator": [  
    {  
      "names": [  
        "iqn.1991-05.com.microsoft:songt-0001"  
      ],  
      "IPs": [  
        "192.168.0.70"  
      ]  
    },  
    {  
      "IPs": [  
        "192.168.0.66"  
      ]  
    }  
  ],  
  "target": [  
    {  
      "IPs": [  
        "192.168.0.64"  
      ]  
    },  
    {  
      "IPs": [  
        "192.168.0.65"  
      ]  
    },  
    {  
      "IPs": [  
        "192.168.0.67"  
      ]  
    }  
  ]  
}
```

```
        "IPs": [
            "192.168.0.67"
        ],
    },
    {
        "IPs": [
            "192.168.0.64",
            "192.168.0.65",
            "192.168.0.67"
        ],
    },
    {
        "IPs": [
            "192.168.0.71"
        ],
    }
]
},
{
    "iqn": "iqn.2012-08.cn.ctyunapi.oos:target03.6",
    "ips": [
        {
            "ip": "192.168.0.65",
            "port": 3260
        }
    ],
    "serverId": "hblock_1",
    "allowlist": {
        "initiator": [
            {
                "names": [
                    "iqn.1991-05.com.microsoft:songt-0001"
                ],
                "IPs": [
```



```
        "192.168.0.70"
      ]
    },
    {
      "IPs": [
        "192.168.0.66"
      ]
    }
  ],
  "target": [
    {
      "IPs": [
        "192.168.0.64"
      ]
    },
    {
      "IPs": [
        "192.168.0.65"
      ]
    },
    {
      "IPs": [
        "192.168.0.67"
      ]
    },
    {
      "IPs": [
        "192.168.0.64",
        "192.168.0.65",
        "192.168.0.67"
      ]
    },
    {
      "IPs": [
        "192.168.0.71"
```

```
        ]
      }
    ]
  }
},
"luns": [
  {
    "lunName": "lun03a",
    "number": 0
  },
  {
    "lunName": "lun03b",
    "number": 1
  }
],
"chap": {
  "name": "chap-test",
  "status": "Disabled"
},
"createTime": 1752111720420,
"serverIds": [
  "hblock_1",
  "hblock_3"
],
"num": 2,
"reclaimPolicy": "Delete"
}
}
```

4.7.12 查询 iSCSI target 连接

此操作用来查询 iSCSI target 的连接信息。

请求语法

```
GET /rest/v1/block/connection/?filter=filter&range=i-j HTTP/1.1
```

Date: *date*

Host: *ip:port*

Authorization: *authorization*

请求参数

参数	类型	描述	是否必须
filter	String	查询的过滤条件。 过滤条件由查询属性和属性值组成，其中属性和属性值之间用单个冒号分隔（key:value）表示模糊匹配，用两个冒号分隔（key::value）表示精确匹配。可以选择多个查询，如果是或的关系，使用“or”将查询条件分隔开；如果是与的关系，使用“and”将查询条件分隔开。 支持的过滤条件类型包括： ● targetName: iSCSI target 名称。	否
range	String	查询 iSCSI target 连接的范围，格式为 i-j，i 和 j 为正整数，0<j-i<1000。按照 iSCSI target 名称进行排序，返回第 i 到第 j 的 iSCSI target 连接信息。 如果输入的查询范围超过 iSCSI target 连接的总个数，返回的结果为空，如果未指定 range，则返回所有符合条件的 iSCSI target 连接。	否

响应结果

名称	类型	描述
connections	Array of connection	连接信息集合，详见“表 1 响应参数 connection 说明”。

表 1 响应参数 connection 说明

名称	类型	描述
targetName	String	iSCSI target 名称。
targetConnections	Array of targetConnection	iSCSI target 连接信息集合，详见“表 2 响应参数 targetConnection 说明”。

表 2 响应参数 targetConnection 说明

名称	类型	描述
iqn	String	IQN。
targetIP	String	服务器端 IP。
establishedSessions	Integer	会话数。
sessionDetails	Array of sessionDetail	会话详细信息集合，详见“表 3 响应参数 sessionDetail 说明”。

表 3 响应参数 sessionDetail 说明

名称	类型	描述
sessionId	String	会话 ID。
clientIP	String	客户端的 IP。
clientPort	Integer	客户端的端口号。
initiatorName	String	连接所属的 initiator 名称。

请求示例

查询 iSCSI target 名称为 **target01** 的连接。

```
GET /rest/v1/block/connection/?filter=targetName::target01 HTTP/1.1
Date: Fri, 27 Mar 2026 08:57:09 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Host:192.168.0.65:1443
```

响应示例

```
HTTP/1.1 200 OK
x-hblock-request-id: 631a52fa43544b2bbdbba088f9d5f645
Connection: keep-alive
Content-Length: 678
Date: Fri, 27 Mar 2026 08:57:09 GMT
Content-Type: application/json; charset=utf-8
Server: HBlock

{
  "data": {
    "connections": [
      {
        "targetName": "target01",
        "targetConnections": [
          {
            "iqn": "iqn.2012-08.cn.ctyunapi.oos:target01.5",
            "targetIP": "192.168.0.65",
            "establishedSessions": 1,
            "sessionDetails": [
              {
                "sessionId": "0xf159",
                "clientIP": "192.168.0.69",
                "clientPort": 59169,
                "initiatorName": "iqn.1991-05.com.microsoft:songt-0008"
              }
            ]
          }
        ]
      }
    ]
  }
}
```

```
]
},
{
  "iqn": "iqn.2012-08.cn.ctyunapi.oos:target01.6",
  "targetIP": "192.168.0.64",
  "establishedSessions": 2,
  "sessionDetails": [
    {
      "sessionId": "0x12f",
      "clientIP": "192.168.0.70",
      "clientPort": 54870,
      "initiatorName": "iqn.1991-05.com.microsoft:songt-0001"
    },
    {
      "sessionId": "0xffe9",
      "clientIP": "192.168.0.69",
      "clientPort": 59170,
      "initiatorName": "iqn.1991-05.com.microsoft:songt-0008"
    }
  ]
}
]
}
]
}
}
```

4.7.13 删除 iSCSI target 连接

此操作用来删除 iSCSI target 连接。

请求语法

```
DELETE
/rest/v1/block/connection/targetName?initiatorName=initiatorName&targetIP=targetIP
HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

请求参数

参数	类型	描述	是否必须
targetName	String	删除的连接所属的 iSCSI target 名称。 取值：长度范围是 1~16，可以由小写字母、数字、句点（.）和短横线（-）组成，且仅支持以字母或数字开头。	是
initiatorName	String	删除的连接所属的 initiator 名称。 取值： - 若使用 Windows iSCSI 发起程序，initiatorName 为“发起程序名称”。 - 若使用 Linux iSCSI 发起程序，initiatorName 为使用 cat /etc/iscsi/initiatorname.iscsi 指令取得“InitiatorName”。	否
targetIP	String	要删除连接所对应的 iSCSI target IP。	否

请求示例

删除 iSCSI target 名称为 **target02**、发起程序名称为 **iqn.1991-05.com.microsoft:ecs-28f3**、targetIP 为 **192.168.0.72** 的连接。

```
DELETE /rest/v1/block/connection/target02?initiatorName=iqn.1991-05.com.microsoft:ecs-28f3&targetIP=192.168.0.72 HTTP/1.1
Date: Wed, 16 Mar 2022 03:24:59 GMT
Host: 192.168.0.121:1443
Authorization: HBlock userName:signature
```

响应示例

```
HTTP/1.1 204 No Content
Date: Wed, 16 Mar 2022 03:24:59 GMT
Connection: keep-alive
Content-Type: application/json; charset=utf-8
x-hblock-request-id: acec0eaf2909485992e90032ce587604
Server: HBlock
```


4.7.14 错误码列表

HTTP status	错误码	错误信息	描述
400	CanNotMigrateTarget	The target is not available to migrate.	该 target 无法进行迁移。
400	ExceedThreshold	The number of <i>operation</i> cannot exceed <i>value</i> .	操作的数量超阈值。
400	InsufficientServers	This method is not allowed because the number of available servers in the cluster is not enough.	集群中可用服务器数量不足，无法操作，请解决后重试。
400	InvalidArgumentConsist	Value <i>value</i> at <i>argument</i> failed to satisfy the constraint: Argument must contain <i>Format</i> .	参数格式不符合要求。
400	InvalidArgumentFormat	Value <i>value</i> at ' <i>argument</i> ' failed to satisfy constraint: Argument must conform to <i>Format</i> format.	参数格式不正确。
400	InvalidCHAPName	Value <i>value</i> at 'chap name' failed to satisfy constraint: Argument can contain letters, digits, dots (.), colons (:), underscores (_) or hyphens (-), and the length is between 3 and 64, must begin with a letter or digit.	CHAP 名字不合法。
400	InvalidCHAPPassword	Value <i>value</i> at 'password' failed to satisfy constraint: Argument	密码不合法：字符串形式，长度范围

		must contain at least 2 of the following: uppercase letters, lowercase letters, digits, underscores (<code>_</code>), and the length is between 12 and 16.	是 12~16，必须包含大写字母、小写字母、数字、下划线(<code>_</code>)中的至少两种字符，字母区分大小写。
400	InvalidFile	Failed to parse file <i>filename</i> .	文件格式不符合要求。
400	InvalidString	Value " at ' <i>argument</i> ' failed to satisfy constraint: Argument must not be an empty string.	参数不能是空字符串。
400	InvalidTargetDest	Invalid target destination server <i>server_ID</i> .	目标服务器目前无法创建该 target，请重新选择。
400	InvalidTargetName	Value <i>value</i> at 'target name' failed to satisfy constraint: Argument must contain only lowercase letters, digits, dots (<code>.</code>) or hyphens (<code>-</code>), and does not exceed 16 characters, must begin with a letter or digit.	target 名称不正确。
400	InvalidTargetSource	Invalid target source server <i>server_ID</i> .	target 的源服务器目前不能进行迁移操作，请稍后再试。
400	LessThanMinValue	Value <i>value</i> at 'number' failed to satisfy constraint: Argument must be greater than or equal to <i>value</i> .	该参数必须大于或等于另一个参数。

400	MigrateTargetForcibly	Force migrate should be used when migrating a target. When using this option, there may be a risk of data loss.	迁移 target 仅能使用强制迁移。请注意，强制移除，会产生数据丢失风险，请谨慎操作。
400	MissingAtLeastOneArgument	Missing argument, at least one argument in ' <i>argument1</i> , <i>argument2</i> , <i>argument3</i> ...' is needed.	至少选择一个参数。
400	MissingCHAPName	Value null at 'chap name' failed to satisfy constraint: Argument must not be null when CHAP is enabled.	启用 CHAP 时，缺少 CHAP 名称。
400	MissingCHAPPassword	Value null at 'password' failed to satisfy constraint: Argument must not be null when CHAP is enabled.	启用 CHAP 时，缺少密码。
400	MissingTargetSourceOrDest	Target source and destination server ID must not be null.	迁移 target 时，源和目的服务器 ID 必须存在。
400	StandaloneModeNotAllowed	' <i>operation</i> ' is not supported by standalone mode of HBlock.	单机版本 HBlock 不支持该操作。
400	TooManyServers	The number of servers cannot exceed <i>value</i> .	target 对应的服务器数量不能超过 <i>value</i> 个。
400	TooManyTargets	The number of target IQNs cannot exceed 32766.	target IQN 数量超过上限。

400	TooManyTargets	The number of target IQNs cannot exceed 1.	target 数量超过上限。一次只能迁移 1 个 target IQN。
404	NoSuchCHAP	The target with name <i>name</i> has no CHAP.	target 下没有 CHAP。
404	NoSuchConnection	The connection does not exist.	连接不存在。
404	NoSuchIQN	The IQN with name <i>IQNName</i> does not exist.	指定的 iqn 不存在。
404	NoSuchTarget	The target with name <i>targetName</i> does not exist.	target 不存在。
404	NoSuchTargetIP	The target with IP <i>targetIP</i> does not exist.	IP 为该值的 target 不存在。
409	ConflictWithTargetMigrating	Failed to <i>operation</i> because the target <i>targetName</i> is migrating.	target 正在迁移，操作失败。
409	ConnectionExists	There are connections on target <i>targetName</i> . Please disconnect the connections and try again.	target 上有连接，请先断开所有连接后重试。
409	InvalidServerStatus	The server <i>serverId</i> status is ' <i>status</i> ', the request is invalid.	服务器当前的状态不正确，请求无效。
409	InvalidStorStatus	The HBlock status is ' <i>status</i> ', the request is invalid.	HBlock 当前的状态不正确，请求无效。
409	InvalidTargetStatus	The target status is ' <i>status</i> ', the request is invalid.	target 当前的状态不正确，请求无效。
409	LUNExists	There are LUNs in this target. Please delete all LUNs and try	target 有关联的卷。

		again.	
409	TargetAlreadyExists	The target with name <i>targetName</i> already exists.	target 已经存在

4.8 存储池（集群版）

4.8.1 创建存储池

此操作用来创建存储池。

说明：HBlock 集群中最多可以创建 32768 个存储池。

请求语法

```
POST /rest/v1/system/storagepool HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "poolName": poolName,
  "faultDomain": faultDomain,
  "description": description,
  "qosPolicy": {
    "qosName": qosName,
    "IOPS": IOPS,
    "readIOPS": readIOPS,
    "writeIOPS": writeIOPS,
    "Bps": Bps,
    "readBps": readBps,
    "writeBps": writeBps,
    "IOPSBurst": IOPSBurst,
    "readIOPSBurst": readIOPSBurst,
    "writeIOPSBurst": writeIOPSBurst,
    "BpsBurst": BpsBurst,
    "readBpsBurst": readBpsBurst,
    "writeBpsBurst": writeBpsBurst,
    "IOPSBurstSecs": IOPSBurstSecs,
    "readIOPSBurstSecs": readIOPSBurstSecs,
    "writeIOPSBurstSecs": writeIOPSBurstSecs,
    "BpsBurstSecs": BpsBurstSecs,
    "readBpsBurstSecs": readBpsBurstSecs,
```

```
"writeBpsBurstSecs": writeBpsBurstSecs
}
}
```

请求参数

参数	类型	描述	是否必须
poolName	String	存储池名称。 取值：长度范围是 1~16，只能由字母、数字和短横线（-）、下划线（_）组成，字母区分大小写，且仅支持以字母和数字开头。	是
faultDomain	String	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。	否
description	String	存储池的描述信息。 取值：1~50 位字符串。	否
qosPolicy	Object	卷关联的 QoS 策略信息，详见“表 1 请求参数 config.qosPolicy 说明”。	否

表 1 请求参数 config.qosPolicy 说明

参数	类型	描述	是否必须
qosName	String	指定 QoS 策略名称。 说明：创存储池时，如果指定 QoS 策略名称不	否

		存在，那么同时创建 QoS 策略，新创建的 QoS 策略的回收策略默认为 Delete。 取值：长度范围 1~64，只能由字母、数字和短横线(-)组成，区分大小写，且仅支持以字母或数字开头。 注意： ● 如果没有指定 qosName，但是设置了 IOPS/Bps 中的任何一个参数，则根据 IOPS/Bps 自动创建一个 QoS 策略，系统为策略名赋值：pool-poolname-qos-timestamp。 ● 如果指定了 qosName 以及 IOPS/Bps 中的任何一个参数，则 qosName 不能与已存在的 QoS 策略重名。 ● 如果指定了 qosName，没有设置 IOPS/Bps 中的任何一个参数，则 qosName 需要为已存在的 QoS 策略名称。	
IOPS	Long	每秒能够进行读写操作次数的最大值。 取值：取值范围为[-1, 999999999]，默认值为-1。-1 表示不限制。	否
readIOPS	Long	每秒能够进行读操作次数的最大值。 取值：取值范围为[-1, 999999999]，默认值为-1。-1 表示不限制。	否
writeIOPS	Long	每秒能够进行写操作次数的最大值。 取值：取值范围为[-1, 999999999]，默认值为-1。-1 表示不限制。	否
Bps	Long	每秒可传输数据量的最大值。	否

		取值：取值范围为[-1, 4096000000000]，默认值为-1，单位是 B/s。-1 表示不限制。	
readBps	Long	读带宽上限。 取值：取值范围为[-1, 4096000000000]，默认值为-1，单位是 B/s。-1 表示不限制。	否
writeBps	Long	写带宽上限。 取值：取值范围为[-1, 4096000000000]，默认值为-1，单位是 B/s。-1 表示不限制。	否
IOPSBurst	Long	使用 Burst 功能时，每秒能够进行读写操作次数的最大值。 取值：只有当 IOPS 大于等于 1 时，此项设置为 -1 或(IOPS, 999999999]内的正整数方可生效。默认值为-1，表示不限制。	否
readIOPSBurst	Long	使用 Burst 功能时，每秒能够进行读操作次数的最大值。 取值：只有当 readIOPS 大于等于 1 时，此项设置为-1 或(readIOPS, 999999999]内的正整数方可生效。默认值为-1，表示不限制。	否
writeIOPSBurst	Long	使用 Burst 功能时，每秒能够进行写操作次数的最大值。 取值：只有当 writeIOPS 大于等于 1 时，此项设置为-1 或(writeIOPS, 999999999]内的正整数方可生效。默认值为-1，表示不限制。	否
BpsBurst	Long	使用 Burst 功能时，每秒可传输的数据量最大值。 取值：只有当 Bps 大于等于 1 时，此项设置为-1 或(Bps, 4096000000000]内的正整数方可生效，	否

		默认值为-1，单位是 B/s。-1 表示不限制。	
readBpsBurst	Long	使用 Burst 功能时，读带宽上限。 取值：只有当 readBps 大于等于 1 时，此项设置为-1 或(readBps, 4096000000000]内的正整数方可生效，默认值为-1，单位是 B/s。-1 表示不限制。	否
writeBpsBurst	Long	使用 Burst 功能时，写带宽上限。 取值：只有当 writeBps 大于等于 1 时，此项设置为-1 或(writeBps, 4096000000000]内的正整数方可生效，默认值为-1，单位是 B/s。-1 表示不限制。	否
IOPSBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的能力进行读写操作所能持续的时间。 注意：只有在 IOPS Burst 功能启用时，此配置才生效。 取值：取值范围为[1, 999999999]，默认值为 1，单位是秒。	否
readIOPSBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的能力进行读操作所能持续的时间。 注意：只有在 read IOPS Burst 功能启用时，此配置才生效。 取值：取值范围为[1, 999999999]，默认值为 1，单位是秒。	否
writeIOPSBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的能力进行写操作所能持续的时间。 注意：只有在 write IOPS Burst 功能启用时，此配置才生效。	否

		取值：取值范围为[1, 999999999]，默认值为1，单位是秒。	
BpsBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的流量能力所能持续的时间。 注意： 只有在 Bps Burst 功能启用时，此配置才生效。 取值：取值范围为[1, 999999999]，默认值为1，单位是秒。	否
readBpsBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的读流量能力所能持续的时间。 注意： 只有在 read Bps Burst 功能启用时，此配置才生效。 取值：取值范围为[1, 999999999]，默认值为1，单位是秒。	否
writeBpsBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的写流量能力所能持续的时间。 注意： 只有在 write Bps Burst 功能启用时，此配置才生效。 取值：取值范围为[1, 999999999]，默认值为1，单位是秒。	否

请求示例

创建存储池 pool4。

```
POST /rest/v1/system/storagepool HTTP/1.1
Date: Thu, 21 Aug 2025 01:53:15 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 159
Host: 192.168.0.65:1443
```

```
{
  "poolName": "pool4",
  "faultDomain": "path",
  "description": "The pool is path level.",
  "qosPolicy": {
    "qosName": "QoS6"
  }
}
```

响应示例

```
HTTP/1.1 200 OK
x-hblock-request-id: 2a6ccdba8d284dbd8d8aa7aaadc3f050
Connection: keep-alive
Content-Length: 0
Date: Thu, 21 Aug 2025 01:53:15 GMT
Server: HBlock
```

4.8.2 添加节点到存储池

此操作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。

请求语法

```
PUT /rest/v1/system/storagepool/poolName/node HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "nodes": [
    node1,
    node2,
    ...
  ]
}
```

请求参数

参数	类型	描述	是否必须
poolName	String	存储池名称。 取值：长度范围是 1~16，只能由字母、数字和短横线（-）、下划线（_）组成，字母区分大小写，且仅支持以字母和数字开头。	是
nodes	Array of String	指定加入存储池的节点，该节点必须属于集群拓扑节点。 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 <i>default:room4:hblock_4</i>、<i>room4:hblock_4</i>、<i>hblock_4</i> 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。	是

请求示例

为存储池 pool4 添加存储节点。

```
PUT /rest/v1/system/storagepool/pool4/node HTTP/1.1
Date: Tue, 18 Jun 2024 10:06:09 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 104
Host: 192.168.0.110:1443

{
  "nodes": [
```

```
"hblock_1:/mnt/storage01",  
  "rack1:hblock_2:/mnt/storage01"  
]  
}
```

响应示例

```
HTTP/1.1 204 No Content  
x-hblock-request-id: bcd355abb8d481fb17bd2c40affb5bc  
Connection: keep-alive  
Date: Tue, 18 Jun 2024 08:41:36 GMT  
Server: HBlock
```

4.8.3 修改存储池

此操作用来修改存储池信息。

请求语法

```
PUT /rest/v1/system/storagepool/poolName HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "newName": poolName,
  "description": description
}
```

请求参数

参数	类型	描述	是否必须
poolName	String	存储池名称。	是
newName	String	存储池的新名称。 取值：长度范围是 1~16，只能由字母、数字和短横线（-）、下划线（_）组成，字母区分大小写，且仅支持以字母和数字开头。	否
description	String	存储池的描述信息。 取值：1~50 位字符串。	否

请求示例

修改存储池 pool14 的名称为 storagepool-4，并修改描述信息。

```
PUT /rest/v1/system/storagepool/pool14 HTTP/1.1
```



```
Date: Wed, 19 Jun 2024 06:02:44 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 90
Host: 192.168.0.110:1443

{
  "newName": "storagepool-4",
  "description": "The storage pool level is path"
}
```

响应示例

```
HTTP/1.1 200 OK
x-hblock-request-id: 2a6ccdba8d284dbd8d8aa7aaadc3f050
Connection: keep-alive
Date: Wed, 19 Jun 2024 06:02:44 GMT
Server: HBlock
```

4.8.4 移除存储池内的节点

此操作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。

请求语法

```
DELETE /rest/v1/system/storagepool/poolName/node HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "nodes": [
    node1,
    node2,
    ...
  ],
  "force": force,
  "deleteLocalData": deleteLocalData
}
```

请求参数

参数	类型	描述	是否必须
poolName	String	存储池名称。 取值：长度范围是 1~16，只能由字母、数字和短横线（-）、下划线（_）组成，字母区分大小写，且仅支持以字母和数字开头。	是
nodes	Array of	要移除的节点。一次可以移除多个同一故障域的节	是

	String	点。一次不能移除涉及多个故障域的节点。	
force	Boolean	是否强制移除存储池内的节点。 注意：强制将节点从存储池中移除，可能造成数据丢失。 取值： ● true：强制移除存储池内的节点。 ● false：不强制移除存储池内的节点。 默认值为 false。	否
deleteLocalData	Boolean	将节点从存储池移除，是否删除存储在该节点上的 HBlock 数据。 取值： ● true：删除存储在该节点上的 HBlock 数据。 ● false：不删除存储在该节点上的 HBlock 数据。 默认值为 false。	否

请求示例

移除存储池 `storagepool-4` 内的节点。

```
DELETE /rest/v1/system/storagepool/storagepool-4/node HTTP/1.1
Date: Wed, 19 Jun 2024 08:06:59 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 111
Host: 192.168.0.110:1443

{
  "nodes": [
    "default:hblock_3:/mnt/stor",
    "default:hblock_3:/mnt/storage01"
  ]
}
```

```
}
```

响应示例

```
HTTP/1.1 202 Accepted
x-hblock-request-id: 81f0984208a54271a4a5fdfeed0091e0
Connection: keep-alive
Content-Length: 0
Date: Wed, 19 Jun 2024 08:06:59 GMT
Server: HBlock
```

4.8.5 删除非基础存储池

此操作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。

请求语法

```
DELETE /rest/v1/system/storagepool/poolName HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization
```

请求参数

参数	类型	描述	是否必须
poolName	String	存储池名称。 取值：长度范围是 1~16，只能由字母、数字和短横线（-）、下划线（_）组成，字母区分大小写，且仅支持以字母和数字开头。	是

请求示例

删除存储池 storagepool-4。

```
DELETE /rest/v1/system/storagepool/storagepool-4 HTTP/1.1
Date: Wed, 19 Jun 2024 08:23:16 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Host: 192.168.0.110:1443
```

响应示例

```
HTTP/1.1 202 Accepted
x-hblock-request-id: Wed, 19 Jun 2024 08:23:16 GMT
Connection: keep-alive
Content-Length: 0
Date: Wed, 19 Jun 2024 08:23:16 GMT
Server: HBlock
```

4.8.6 批量查询存储池

此操作用来批量查询存储池。

请求语法

```
GET /rest/v1/system/storagepool?filter=filter HTTP/1.1
```

Date: *date*

Content-Type: application/json; charset=utf-8

Host: *ip:port*

Authorization: *authorization*

请求参数

参数	类型	描述	是否必须
filter	String	查询的过滤条件。 过滤条件由查询属性和属性值组成，其中属性和属性值之间用单个冒号分隔（key:value）表示模糊匹配，用两个冒号分隔（key::value）表示精确匹配。可以选择多个查询，如果是或的关系，使用“or”将查询条件分隔开；如果是与的关系，使用“and”将查询条件分隔开。 支持的过滤条件类型包括： ● poolName: 存储池的名称。	否

响应结果

名称	类型	描述
storagePool	Array of storagePool	存储池信息集合，详见“表 1 响应参数 storagePool 说明”。

表 1 响应参数 storagePool 说明

名称	类型	描述
name	String	存储池名称。
id	Short	存储池序号。
isBasePool	Boolean	是否是基础存储池： ● true: 是基础存储池。 ● false: 不是基础存储池。
status	String	存储池状态： ● Normal: 正常。 ● Deleting: 删除中。
faultDomain	String	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。
description	String	存储池描述信息。
totalCapacity	Long	存储池总容量，单位是 bytes。
usedCapacity	Long	存储池已用容量，单位是 bytes。
usedCapacityQuota	Long	存储池已用容量配额，单位是 bytes。
capacityQuota	Long	存储池容量配额，单位

		是 bytes。
createTime	Long	存储池创建时间。
luns.lunNamesForCachePool	Array of lunNamesForCachePool	缓存存储池卷列表。
luns.lunNamesForPool	Array of lunNamesForPool	存储池卷列表。
luns.status	Object	卷的状态信息，详见 “表 2 响应参数 luns.status 说明”。
topology	Array of topology	存储池拓扑信息集合， 详见“表 3 响应参数 topology/childNode 说明”。
qosPolicy.qosName	String	存储池关联的 QoS 策略 名称。
qosPolicyforLUN.qosName	String	存储池内卷的默认 QoS 策略名称。
dataHealth.normal	Double	该存储池所有卷正常数 据的百分比。
dataHealth.lowRedundancy	Double	该存储池所有卷降级数 据的百分比。
dataHealth.error	Double	该存储池所有卷错误数 据的百分比。
dataHealth.reconstructionProgress	Double	如果该存储池卷存在降 级状态的数据，数据重 建进度的百分比。
dataHealth.ETR	Long	如果该存储池卷存在降

		级数据，预计剩余的数据重建时间。单位为秒。如果返回值为负数，说明数据无法重建，需用户尽快修复。

表 2 响应参数 `luns.status` 说明

名称	类型	描述
number	Integer	存储池关联卷的个数。
cacheNumber	Integer	缓存模式卷的个数。
storageNumber	Integer	存储模式卷的个数。
localNumber	Integer	本地模式卷的个数。
cacheCapacity	Long	缓存模式卷容量，单位是 bytes。
storageCapacity	Long	存储模式卷容量，单位是 bytes。
localCapacity	Long	本地模式卷容量，单位是 bytes。

表 3 响应参数 `topology/childNode` 说明

名称	类型	描述
name	String	拓扑节点或者子节点名称。
fullname	String	拓扑节点或者子节点的全路径名称。
type	String	拓扑节点或者子节点的类型： ● room: 机房。 ● rack: 机架。 ● server: 服务器。 ● path: 数据目录。
description	String	拓扑节点或子节点的描述信

			息。 如果未添加描述信息，或者节点类型是 path，不返回此项。
healthStatus		String	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。
childNodes		Array of childNode	拓扑图中子节点。childNode 和 topology 基本一致，类型不同，包含的内容不同，详见本表其他字段。
如果 topology 或者 childNode 的类型是 server	clusterAddress	String	拓扑图集群网络 IP。
	publicAddress	String	拓扑图业务网络 IP。
	allDiskPaths.usedCapacity	Long	拓扑图中所有数据目录的已用容量。
	allDiskPaths.totalCapacity	Long	拓扑图中所有数据目录的总容量。
	serverStatus	String	拓扑图的服务器状态： ● Connected: 已连接。 ● Disconnected: 未连接。 ● Removing: 移除中。
	serverId	String	拓扑图的服务器 ID。
	childNodes	Array of	拓扑图中子节点，childNode

		childNodes	和 topology 基本一致，类型不同，包含的内容不同，详见本表其他字段。
如果 topology 或者 childNodes 的类型是 path	usedCapacity	Long	HBlock 数据目录对应分区的已用容量，单位是 bytes。
	totalCapacity	Long	HBlock 数据目录对应分区的总容量，单位是 bytes。
	usedCapacityQuota	Long	数据目录已用容量配额，单位是 bytes。
	capacityQuota	Long	数据目录容量配额，单位是 bytes。
	managementStatus	String	数据目录的管理状态： ● Added: 已添加。 ● Removing: 正在移除。
	healthStatus	String	数据目录健康状态： ● 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；HBlock 对

			这个目录停写；数据目录配额使用率超过阈值（系统默认值为 95%）；数据目录配额为 0。 ● 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。
	healthDetail	Array of string	数据目录健康状态详情： ● 如果健康状态为 Healthy，此字段为空。 ● 如果健康状态为 Warning 或 Error，显示警告或错误的详细信息。
	serverId	String	数据目录所属的服务器 ID。

请求示例

查询存储池中包含 pool 的存储池。

```
GET /rest/v1/system/storagepool?filter=poolName:pool HTTP/1.1
Date: Wed, 31 Dec 2025 09:23:34 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Host: 192.168.0.64:1443
```

响应示例

```
HTTP/1.1 200 OK
x-hblock-request-id: e35107c99bf0466eb46362c4858cbc3b
Connection: keep-alive
```

```
Content-Length: 2727
Date: Wed, 31 Dec 2025 09:23:34 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "storagePool": [
      {
        "name": "pool1",
        "id": 0,
        "isBasePool": true,
        "status": "Normal",
        "faultDomain": "server",
        "totalCapacity": 299111325696,
        "usedCapacity": 4151316480,
        "usedCapacityQuota": 3298676875,
        "capacityQuota": 299111325696,
        "createTime": 1766021671041,
        "luns": {
          "lunNamesForCachePool": [],
          "lunNamesForPool": [
            "lun04c",
            "lun01a-C1",
            "lun03a",
            "lun040012",
            "lun04a",
            "luncloud",
            "lun01a",
            "lun01b"
          ],
          "status": {
            "number": 8,
            "cacheNumber": 1,
            "storageNumber": 3,
            "localNumber": 4,
            "cacheCapacity": 515396075520,
            "storageCapacity": 784905273344,
            "localCapacity": 505732399104
          }
        }
      }
    ],
  },
}
```

```
"topology": [  
  {  
    "name": "hblock_1",  
    "fullname": "default:hblock_1",  
    "type": "server",  
    "healthStatus": "Healthy",  
    "clusterAddress": "192.168.0.65",  
    "publicAddress": "192.168.0.65",  
    "allDiskPaths": {  
      "usedCapacity": 1815838720,  
      "totalCapacity": 99703775232  
    },  
    "serverStatus": "Connected",  
    "serverId": "hblock_1",  
    "childNodes": [  
      {  
        "name": "/mnt/stor01",  
        "fullname": "default:hblock_1:/mnt/stor01",  
        "type": "path",  
        "usedCapacity": 1815838720,  
        "totalCapacity": 99703775232,  
        "usedCapacityQuota": 1100965008,  
        "capacityQuota": 99703775232,  
        "managementStatus": "Added",  
        "healthStatus": "Healthy",  
        "healthDetail": [],  
        "serverId": "hblock_1"  
      }  
    ]  
  },  
  {  
    "name": "hblock_2",  
    "fullname": "default:hblock_2",  
    "type": "server",  
    "healthStatus": "Healthy",  
    "clusterAddress": "192.168.0.64",  
    "publicAddress": "192.168.0.64",  
    "allDiskPaths": {  
      "usedCapacity": 1134886912,  
      "totalCapacity": 99703775232  
    },  
  },  
]
```

```
"serverStatus": "Connected",
"serverId": "hblock_2",
"childNodes": [
  {
    "name": "/mnt/stor02",
    "fullname": "default:hblock_2:/mnt/stor02",
    "type": "path",
    "usedCapacity": 1134886912,
    "totalCapacity": 99703775232,
    "usedCapacityQuota": 1100928144,
    "capacityQuota": 99703775232,
    "managementStatus": "Added",
    "healthStatus": "Healthy",
    "healthDetail": [],
    "serverId": "hblock_2"
  }
]
},
{
  "name": "hblock_3",
  "fullname": "default:hblock_3",
  "type": "server",
  "healthStatus": "Healthy",
  "clusterAddress": "192.168.0.67",
  "publicAddress": "192.168.0.67",
  "allDiskPaths": {
    "usedCapacity": 1200590848,
    "totalCapacity": 99703775232
  },
  "serverStatus": "Connected",
  "serverId": "hblock_3",
  "childNodes": [
    {
      "name": "/mnt/stor01",
      "fullname": "default:hblock_3:/mnt/stor01",
      "type": "path",
      "usedCapacity": 1200590848,
      "totalCapacity": 99703775232,
      "usedCapacityQuota": 1096783723,
      "capacityQuota": 99703775232,
      "managementStatus": "Added",
```



```
        "healthStatus": "Healthy",
        "healthDetail": [],
        "serverId": "hblock_3"
      }
    ]
  },
  ],
  "qosPolicy": {
    "qosName": "QoS1"
  },
  "qosPolicyforLUN": {
    "qosName": "QoS1"
  },
  "dataHealth": {
    "normal": 100,
    "lowRedundancy": 0,
    "error": 0
  }
},
{
  "name": "pool2",
  "id": 1,
  "isBasePool": false,
  "status": "Normal",
  "faultDomain": "server",
  "totalCapacity": 0,
  "usedCapacity": 0,
  "usedCapacityQuota": 0,
  "capacityQuota": 0,
  "createTime": 1766384595246,
  "luns": {
    "lunNamesForCachePool": [],
    "lunNamesForPool": [],
    "status": {}
  },
  "topology": [],
  "dataHealth": {
    "normal": 100,
    "lowRedundancy": 0,
    "error": 0
  }
}
```

```
}  
  ]  
}  
{
```

4.8.7 查询指定存储池

此操作用来查询指定存储池。

请求语法

```
GET /rest/v1/system/storagepool/poolName HTTP/1.1
```

Date: *date*

Content-Type: application/json; charset=utf-8

Host: *ip:port*

Authorization: *authorization*

请求参数

参数	类型	描述	是否必须
poolName	String	存储池名称。 取值：长度范围是 1~16，只能由字母、数字和短横线（-）、下划线（_）组成，字母区分大小写，且仅支持以字母和数字开头。	是

响应结果

名称	类型	描述
name	String	存储池名称。
id	Short	存储池序号。
isBasePool	Boolean	是否是基础存储池： ● true：是基础存储池。 ● false：不是基础存储池。

status	String	存储池状态： ● Normal: 正常。 ● Deleting: 删除中。
faultDomain	String	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。
description	String	存储池描述信息。
totalCapacity	Long	存储池总容量，单位是 bytes。
usedCapacity	Long	存储池已用容量，单位是 bytes。
usedCapacityQuota	Long	存储池已用容量配额，单位是 bytes。
capacityQuota	Long	存储池容量配额，单位是 bytes。
createTime	Long	存储池创建时间。
luns.lunNamesForCachePool	Array of lunNamesForCachePool	缓存存储池卷列表。
luns.lunNamesForPool	Array of lunNamesForPool	存储池卷列表。
luns.status	Object	卷的状态信息，详见“表 1 响应参数”

		luns.status 说明”。
topology	Array of topology	存储池拓扑信息集合，详见“表 2 响应参数 topology/childNode 说明”。
qosPolicy.qosName	String	存储池关联的 QoS 策略名称。
qosPolicyforLUN.qosName	String	存储池内卷的默认 QoS 策略名称。
dataHealth.normal	Double	该存储池所有卷正常数据的百分比。
dataHealth.lowRedundancy	Double	该存储池所有卷降级数据的百分比。
dataHealth.error	Double	该存储池所有卷错误数据的百分比。
dataHealth.reconstructionProgress	Double	如果该存储池卷存在降级状态的数据，数据重建进度的百分比。
dataHealth.ETR	Long	如果该存储池卷存在降级数据，预计剩余的数据重建时间。单位为秒。如果返回值为负数，说明数据无法重建，需用户尽快修复。

表 1 响应参数 **luns.status** 说明

名称	类型	描述

number	Integer	存储池关联卷的个数。
cacheNumber	Integer	缓存模式卷的个数。
storageNumber	Integer	存储模式卷的个数。
localNumber	Integer	本地模式卷的个数。
cacheCapacity	Long	缓存模式卷容量，单位是 bytes。
storageCapacity	Long	存储模式卷容量，单位是 bytes。
localCapacity	Long	本地模式卷容量，单位是 bytes。

表 2 响应参数 topology/childNode 说明

名称	类型	描述
name	String	拓扑节点或者子节点名称。
fullname	String	拓扑节点或者子节点的全路径名称。
type	String	拓扑节点或者子节点的类型： ● room: 机房。 ● rack: 机架。 ● server: 服务器。 ● path: 数据目录。
description	String	拓扑节点或子节点的描述信息。 如果未添加描述信息，或者节点类型是 path，不返回此项。
healthStatus	String	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。
childNodes		Array of childNode	拓扑图中子节点， childNode 和 topology 基本一致，类型不同，包含的内容不同，详见本表其他字段。
如果 topology 或者 childNode 的类型是 server	clusterAddress	String	拓扑图集群网络 IP。
	publicAddress	String	拓扑图业务网络 IP。
	allDiskPaths.usedCapacity	Long	拓扑图中所有数据目录的已用容量。
	allDiskPaths.totalCapacity	Long	拓扑图中所有数据目录的总容量。
	serverStatus	String	拓扑图的服务器状态： ● Connected: 已连接。 ● Disconnected: 未连接。 ● Removing: 移除中。
	serverId	String	拓扑图的服务器 ID。
	childNodes	Array of childNodes	拓扑图中子节点， childNode 和 topology 基本一致，类型不同，包含的内容不同，详见本表其他字段。
如果 topology 或者 childNode 的类型是	usedCapacity	Long	HBlock 数据目录对应分区的已用容量，单位是 bytes。
	totalCapacity	Long	HBlock 数据目录对应分区的总容量，单位是 bytes。
	usedCapacityQuota	Long	数据目录已用容量配额，单位

path			是 bytes。
	capacityQuota	Long	数据目录容量配额，单位是 bytes。
	managementStatus	String	数据目录的管理状态： ● Added: 已添加。 ● Removing: 正在移除。
	healthStatus	String	数据目录健康状态： ● 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；HBlock 对这个目录停写；数据目录配额使用率超过阈值（系统默认值为 95%）；数据目录配额为 0。 ● 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。
	healthDetail	Array of string	数据目录健康状态详情： ● 如果健康状态为 Healthy，此字段为空。 ● 如果健康状态为 Warning 或 Error，显示警告或错误的详细信息。
	serverId	String	数据目录所属的服务器 ID。

请求示例

查询存储池 **pool1**。

```
GET /rest/v1/system/storagepool/pool1 HTTP/1.1
Date: Wed, 31 Dec 2025 09:26:21 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Host: 192.168.0.64:1443
```

响应示例

```
HTTP/1.1 200 OK
x-hblock-request-id: c68b4918d8814c908c6a5e10c25c163d
Connection: keep-alive
Content-Length: 2385
Date: Wed, 31 Dec 2025 09:26:21 GMT
Content-Type: application/json; charset=utf-8
Server: HBlock

{
  "data": {
    "name": "pool1",
    "id": 0,
    "isBasePool": true,
    "status": "Normal",
    "faultDomain": "server",
```

```
"totalCapacity": 299111325696,
"usedCapacity": 4151316480,
"usedCapacityQuota": 3298676875,
"capacityQuota": 299111325696,
"createTime": 1766021671041,
"luns": {
  "lunNamesForCachePool": [],
  "lunNamesForPool": [
    "lun04c",
    "lun01a-C1",
    "lun03a",
    "lun040012",
    "lun04a",
    "luncloud",
    "lun01a",
    "lun01b"
  ],
  "status": {
    "number": 8,
    "cacheNumber": 1,
    "storageNumber": 3,
    "localNumber": 4,
    "cacheCapacity": 515396075520,
    "storageCapacity": 784905273344,
    "localCapacity": 505732399104
  }
},
"topology": [
  {
    "name": "hblock_1",
    "fullname": "default:hblock_1",
    "type": "server",
    "healthStatus": "Healthy",
    "clusterAddress": "192.168.0.65",
    "publicAddress": "192.168.0.65",
    "allDiskPaths": {
      "usedCapacity": 1815838720,
      "totalCapacity": 99703775232
    },
    "serverStatus": "Connected",
    "serverId": "hblock_1",
```

```
    "childNodes": [
      {
        "name": "/mnt/stor01",
        "fullname": "default:hblock_1:/mnt/stor01",
        "type": "path",
        "usedCapacity": 1815838720,
        "totalCapacity": 99703775232,
        "usedCapacityQuota": 1100965008,
        "capacityQuota": 99703775232,
        "managementStatus": "Added",
        "healthStatus": "Healthy",
        "healthDetail": [],
        "serverId": "hblock_1"
      }
    ]
  },
  {
    "name": "hblock_2",
    "fullname": "default:hblock_2",
    "type": "server",
    "healthStatus": "Healthy",
    "clusterAddress": "192.168.0.64",
    "publicAddress": "192.168.0.64",
    "allDiskPaths": {
      "usedCapacity": 1134886912,
      "totalCapacity": 99703775232
    },
    "serverStatus": "Connected",
    "serverId": "hblock_2",
    "childNodes": [
      {
        "name": "/mnt/stor02",
        "fullname": "default:hblock_2:/mnt/stor02",
        "type": "path",
        "usedCapacity": 1134886912,
        "totalCapacity": 99703775232,
        "usedCapacityQuota": 1100928144,
        "capacityQuota": 99703775232,
        "managementStatus": "Added",
        "healthStatus": "Healthy",
        "healthDetail": [],
```

```
        "serverId": "hblock_2"
      }
    ]
  },
  {
    "name": "hblock_3",
    "fullname": "default:hblock_3",
    "type": "server",
    "healthStatus": "Healthy",
    "clusterAddress": "192.168.0.67",
    "publicAddress": "192.168.0.67",
    "allDiskPaths": {
      "usedCapacity": 1200590848,
      "totalCapacity": 99703775232
    },
    "serverStatus": "Connected",
    "serverId": "hblock_3",
    "childNodes": [
      {
        "name": "/mnt/stor01",
        "fullname": "default:hblock_3:/mnt/stor01",
        "type": "path",
        "usedCapacity": 1200590848,
        "totalCapacity": 99703775232,
        "usedCapacityQuota": 1096783723,
        "capacityQuota": 99703775232,
        "managementStatus": "Added",
        "healthStatus": "Healthy",
        "healthDetail": [],
        "serverId": "hblock_3"
      }
    ]
  }
],
"qosPolicy": {
  "qosName": "QoS1"
},
"qosPolicyforLUN": {
  "qosName": "QoS1"
},
"dataHealth": {
```

```
    "normal": 100,  
    "lowRedundancy": 0,  
    "error": 0  
  }  
}  
}
```

4.8.8 查询存储池关联的 QoS

此操作用来查询存储池关联的 QoS 信息。

请求语法

```
GET /rest/v1/system/storagepool/poolName/qos HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

请求参数

参数	类型	描述	是否必须
poolName	String	存储池名称。	是

响应结果

名称	类型	描述
qosPolicy	Object of qosPolicy	QoS 策略信息集合，详见“表 1 响应参数 qosPolicy/qosPolicyforLUN 说明”。
qosPolicyforLUN	Object of qosPolicyforLUN	存储池中卷的默认 QoS 策略信息，详见“表 1 响应参数 qosPolicy/qosPolicyforLUN 说明”。

表 1 响应参数 qosPolicy/qosPolicyforLUN

名称	类型	描述
qosName	String	QoS 策略名称。
IOPS	Long	每秒能够进行读写操作次数的最大值。 说明：-1 表示不限制。
readIOPS	Long	每秒能够进行读操作次数的最大值。

		说明：-1 表示不限制。
writeIOPS	Long	每秒能够进行写操作次数的最大值。 说明：-1 表示不限制。
Bps	Long	每秒可传输数据量的最大值，单位是 B/s。 说明：-1 表示不限制。
readBps	Long	读带宽上限，单位是 B/s。 说明：-1 表示不限制。
writeBps	Long	写带宽上限，单位是 B/s。 说明：-1 表示不限制。
IOPSBurst	Long	使用 Burst 功能时，每秒能够进行读写操作次数的最大值。 说明：-1 表示不限制。
readIOPSBurst	Long	使用 Burst 功能时，每秒能够进行读操作次数的最大值。 说明：-1 表示不限制。
writeIOPSBurst	Long	使用 Burst 功能时，每秒能够进行写操作次数的最大值。 说明：-1 表示不限制。
BpsBurst	Long	使用 Burst 功能时，每秒可传输的数据量最大值，单位是 B/s。 说明：-1 表示不限制。
readBpsBurst	Long	使用 Burst 功能时，读带宽上限，单位是 B/s。 说明：-1 表示不限制。
writeBpsBurst	Long	使用 Burst 功能时，写带宽上限，单位是 B/s。 说明：-1 表示不限制。
IOPSBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的能力进行读写操作所能持续的时间。单位是秒。

readIOPSBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的能力进行读操作所能持续的时间。单位是秒。
writeIOPSBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的能力进行写操作所能持续的时间。单位是秒。
BpsBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的流量能力所能持续的时间。单位是秒。
readBpsBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的读流量能力所能持续的时间。单位是秒。
writeBpsBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的写流量能力所能持续的时间。单位是秒。
createTime	Long	QoS 策略创建的时间。
reclaimPolicy	String	QoS 策略的回收策略： ● Delete: 解除 QoS 策略关联的所有对象或者删除 QoS 策略关联的所有对象，系统会触发删除 QoS 策略。 ● Retain: 解除 QoS 策略关联的所有对象或者删除 QoS 策略关联的所有对象，QoS 策略仍保留。
description	String	QoS 策略的描述信息。

请求示例

查询存储池 defaultpool 的 QoS 策略。

```
GET /rest/v1/system/storagepool/defaultpool/qos HTTP/1.1
Date: Thu, 21 Aug 2025 03:36:22 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.64:1443
```

响应示例

```
HTTP/1.1 200 OK
```


x-hblock-request-id: 8fda965846fa4ce4b0e1e96e40494786

Connection: keep-alive

Content-Length: 728

Date: Thu, 21 Aug 2025 03:36:22 GMT

Content-Type: application/json;charset=utf-8

Server: HBlock

```
{
  "data": {
    "qosPolicy": {
      "qosName": "QoS6",
      "readIOPS": 600,
      "writeIOPS": 440,
      "readBps": 1048576000,
      "writeBps": 1572864000,
      "readIOPSBurst": 6000,
      "writeIOPSBurst": 4000,
      "readBpsBurst": 2097152000,
      "writeBpsBurst": 2097152000,
      "readIOPSBurstSecs": 1,
      "writeIOPSBurstSecs": 1,
      "readBpsBurstSecs": 1,
      "writeBpsBurstSecs": 1,
      "createTime": 1753950531353,
      "reclaimPolicy": "Retain",
      "description": "It is QoS6.",
      "IOPS": 1000,
      "Bps": 2097152000,
      "IOPSBurst": 10000,
      "BpsBurst": 2202009600,
      "IOPSBurstSecs": 1,
      "BpsBurstSecs": 1
    },
    "qosPolicyforLUN": {
      "qosName": "QoS-Test",
```

```
    "readIOPSBurstSecs": 1,  
    "writeIOPSBurstSecs": 1,  
    "readBpsBurstSecs": 1,  
    "writeBpsBurstSecs": 1,  
    "createTime": 1754470952232,  
    "reclaimPolicy": "Retain",  
    "IOPS": 8000,  
    "Bps": 8388608000,  
    "IOPSBurstSecs": 1,  
    "BpsBurstSecs": 1  
  }  
}  
}
```

4.8.9 错误码列表

HTTP status	错误码	错误信息	描述
400	CanNotAddNodeToPool	The node <i>nodeName</i> has no available paths added to storage pool <i>poolname</i> .	该节点没有可用的 path 节点添加到存储池。
400	CanNotDeleteBasePool	The storage pool with name <i>poolName</i> can not be deleted. It is a base pool.	无法删除基础存储池。
400	CanNotDeleteNode	The node <i>nodeName</i> can not be deleted. Please remove its child nodes and try again.	该节点不能移除，需要先移除它的子节点后才能移除该节点。
400	CanNotDeleteRootNode	The type of node <i>nodeName</i> is root, it cannot be deleted.	不能删除根节点。
400	CanNotMoveNode	The node <i>nodeName</i> can not be moved. Please make sure the node type is not lower than the fault domain level of storage pool which its leaf node belongs to.	节点不满足重新映射条件，请修改后重试。
400	DuplicatedNode	The node with name <i>nodeName</i> is not unique in the topology.	节点名称在集群中不唯一。
400	ExceedThreshold	The number of <i>operation</i> cannot exceed <i>value</i> .	超出了系统允许的最大个数，无法执行该操作。

400	InvalidEnumValue	Value <i>value</i> at 'node type' failed to satisfy constraint: Argument must satisfy enum value set: [<i>value1</i> , <i>value2</i> ...]	节点类型错误，不符合枚举值。
400	InvalidFile	Failed to parse file <i>filename</i> .	文件格式不正确。
400	InvalidNodeName	Value <i>value</i> at 'node name' failed to satisfy constraint: Argument must contain only letters, digits, dots (.), underscores (_) or hyphens (-), and does not exceed 63 characters, must begin with a letter or digit.	节点名称不正确。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。
400	InvalidNodeType	Value <i>value</i> at 'node type' failed to satisfy constraint: the first node type must be root.	节点类型的包含关系错误，第一个节点必须是 root。
400	InvalidNodeType	Value <i>value</i> at 'node type' failed to satisfy constraint: childNodes of room are limited to [rack, server], childNodes of rack to [server], childNodes of server to [path] only.	节点类型的包含关系错误。room 的节点仅限于 rack、server，机架的子节点是 server，server 的子节点是 path。
400	InvalidPathStatus	The path <i>path</i> status is ' <i>status</i> ', the request is	数据目录状态不正确，无法执行操作。

		invalid.	
400	InvalidPoolName	Value <i>value</i> at 'pool name' failed to satisfy constraint: Argument must contain only letters, digits, underscores (_) or hyphens (-), and does not exceed 16 characters, must begin with a letter or digit.	存储池名称不正确。
400	InvalidTopology	Failed to parse topology.	拓扑格式不符合要求。
400	InvalidTopologyContent	The content of the topology is not compliant.	拓扑文件内容解析错误。
400	InsufficientPath	The base storage pool must have at least one disk path.	基础存储池至少要有 一个数据目录。
400	LUNCorruptedRisk	There is a LUN with redundancy mode, which has the minimum requirement for the number of fault domains. LUN: <i>lunName1</i> [, <i>lunName2</i>] Removing the node may cause data loss. You can use force remove. When using this option, there may be a risk of data loss.	节点移除后会造成卷数据损毁： <i>lunName1</i> [, <i>lunName2</i>]。 移除节点可能会引起数据丢失。只能使用强制移除。请注意，强制移除，会产生数据丢失风险，请谨慎操作。

400	LUNExists	There are LUNs in this storage pool Please delete all LUNs and try again.	存储池关联的卷不存在。
400	MissingRequiredAncestorNode	The node <i>nodeName</i> has no ancestor node with type <i>nodetype</i> .	没有对应的父节点类型。
400	MissingRequiredDescendantNode	The node <i>nodeName</i> has no descendant node with type <i>nodetype</i> .	没有对应的子节点类型。
400	NodeAlreadyExists	The node with name <i>nodeName</i> has already exists in the topology.	节点名称已经在拓扑图中存在。
400	NodeNotAllowed	The type of node <i>nodeName</i> is <i>type</i> , the request is invalid.	节点名称和节点类型不匹配。
400	NoSuchFile	The file with name <i>filename</i> does not exist.	文件不存在。
400	NoSuchNode	The node with name <i>nodeName</i> does not exist.	节点不存在。
400	NoSuchStoragePool	The storage pool with name <i>poolName</i> does not exist.	存储池不存在。
400	NodeTypeNotMatch	Node type does not match. Please check parent node type.	节点类型不匹配，请检查父节点。
400	PathConflict	Can not use path <i>pathName</i> because it belongs to another storage pool.	不能添加该节点，因为该节点已经属于其他存储池。

400	PermissionDenied	Failed to <i>action</i> because user has no permission.	用户的权限不足，无法执行操作。
400	PoolAlreadyExists	The storage pool with name <i>poolName</i> already exists.	存储池名字已经存在。
400	RemoveFaultDomainRisk	Can not remove multiple fault domains at the same time. You can use force remove. When using this option, there may be a risk of data loss.	无法同时移除的多个故障域。
400	RemoveNodeRisk	The node <i>nodeName</i> or its child nodes are being used by a storage pool. Removing the node may cause data loss. You can use force remove. When using this option, there may be a risk of data loss.	移除节点可能会引起数据丢失。只能使用强制移除。请注意，强制移除，会产生数据丢失风险，请谨慎操作。
400	SamePool	Cannot use the same storage pool <i>poolName</i> .	不能使用名字相同的存储池。
400	StandaloneModeNotAllowed	' <i>operation</i> ' is not supported by standalone mode of HBlock.	单机版本 HBlock 不支持该操作。
400	SystemCrashRisk	The node <i>nodename</i> can not be removed because there is only one available fault domain left in the base	无法移除节点，存在以下情况之一，移除节点可能引发系统崩溃：基础存储池仅剩

		storage pool, or the node is related to multiple fault domains in the base pool, removing node may result in a system crash.	一个可用故障域时，无法移除故障域内的任何节点；节点涉及基础存储池的多个故障域。
409	InvalidPoolStatus	The status of storage pool <i>poolName</i> is ' <i>status</i> ', the request is invalid.	存储池当前的状态不正确，请求无效。

4.9 QoS 策略

4.9.1 创建 QoS 策略

此操作用来创建 QoS 策略。

请求语法

```
POST /rest/v1/system/qos HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "qosName": qosName,
  "IOPS": IOPS,
  "readIOPS": readIOPS,
  "writeIOPS": writeIOPS,
  "Bps": Bps,
  "readBps": readBps,
  "writeBps": writeBps,
  "IOPSBurst": IOPSBurst,
  "readIOPSBurst": readIOPSBurst,
  "writeIOPSBurst": writeIOPSBurst,
  "BpsBurst": BpsBurst,
  "readBpsBurst": readBpsBurst,
  "writeBpsBurst": writeBpsBurst,
  "IOPSBurstSecs": IOPSBurstSecs,
  "readIOPSBurstSecs": readIOPSBurstSecs,
  "writeIOPSBurstSecs": writeIOPSBurstSecs,
  "BpsBurstSecs": BpsBurstSecs,
  "readBpsBurstSecs": readBpsBurstSecs,
  "writeBpsBurstSecs": writeBpsBurstSecs,
  "reclaimPolicy": reclaimPolicy,
  "description": description
}
```

请求参数

参数	类型	描述	是否必须
qosName	String	指定 QoS 策略名称。 取值：长度范围 1~64，只能由字母、数字和短横线(-)组成，区分大小写，且仅支持以字母或数字开头。	是
IOPS	Long	每秒能够进行读写操作次数的最大值。 取值：取值范围为[-1, 999999999]，默认值为-1。-1 表示不限制。	否
readIOPS	Long	每秒能够进行读操作次数的最大值。 取值：取值范围为[-1, 999999999]，默认值为-1。-1 表示不限制。	否
writeIOPS	Long	每秒能够进行写操作次数的最大值。 取值：取值范围为[-1, 999999999]，默认值为-1。-1 表示不限制。	否
Bps	Long	每秒可传输数据量的最大值。 取值：取值范围为[-1, 4096000000000]，默认值为-1，单位是 B/s。-1 表示不限制。	否
readBps	Long	读带宽上限。 取值：取值范围为[-1, 4096000000000]，默认值为-1，单位是 B/s。-1 表示不限制。	否
writeBps	Long	写带宽上限。 取值：取值范围为[-1, 4096000000000]，默认值为-1，单位是 B/s。-1 表示不限制。	否
IOPSBurst	Long	使用 Burst 功能时，每秒能够进行读写操作次数的最大值。	否

		取值：只有当 IOPS 大于等于 1 时，此项设置为 -1 或(<i>IOPS</i> , 999999999]内的正整数方可生效。默认值为-1，表示不限制。	
readIOPSBurst	Long	使用 Burst 功能时，每秒能够进行读操作次数的最大值。 取值：只有当 readIOPS 大于等于 1 时，此项设置为-1 或(<i>readIOPS</i> , 999999999]内的正整数方可生效。默认值为-1，表示不限制。	否
writeIOPSBurst	Long	使用 Burst 功能时，每秒能够进行写操作次数的最大值。 取值：只有当 writeIOPS 大于等于 1 时，此项设置为-1 或(<i>writeIOPS</i> , 999999999]内的正整数方可生效。默认值为-1，表示不限制。	否
BpsBurst	Long	使用 Burst 功能时，每秒可传输的数据量最大值。 取值：只有当 Bps 大于等于 1 时，此项设置为-1 或(<i>Bps</i> , 4096000000000]内的正整数方可生效，默认值为-1，单位是 B/s。-1 表示不限制。	否
readBpsBurst	Long	使用 Burst 功能时，读带宽上限。 取值：只有当 readBps 大于等于 1 时，此项设置为-1 或(<i>readBps</i> , 4096000000000]内的正整数方可生效，默认值为-1，单位是 B/s。-1 表示不限制。	否
writeBpsBurst	Long	使用 Burst 功能时，写带宽上限。 取值：只有当 writeBps 大于等于 1 时，此项设置为-1 或(<i>writeBps</i> , 4096000000000]内的正整数方可生效，默认值为-1，单位是 B/s。-1 表示不	否

		限制。	
IOPSBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的能力进行读写操作所能持续的时间。 注意：只有在 IOPS Burst 功能启用时，此配置才生效。 取值：取值范围为[1, 999999999]，默认值为1，单位是秒。	否
readIOPSBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的能力进行读操作所能持续的时间。 注意：只有在 read IOPS Burst 功能启用时，此配置才生效。 取值：取值范围为[1, 999999999]，默认值为1，单位是秒。	否
writeIOPSBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的能力进行写操作所能持续的时间。 注意：只有在 write IOPS Burst 功能启用时，此配置才生效。 取值：取值范围为[1, 999999999]，默认值为1，单位是秒。	否
BpsBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的流量能力所能持续的时间。 注意：只有在 Bps Burst 功能启用时，此配置才生效。 取值：取值范围为[1, 999999999]，默认值为1，单位是秒。	否
readBpsBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的读流量能力所能持续的时间，单位是秒。	否

		注意： 只有在 read Bps Burst 功能启用时，此配置才生效。 取值：取值范围为[1, 999999999]，默认值为1，单位是秒。	
writeBpsBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的写流量能力所能持续的时间。 注意： 只有在 write Bps Burst 功能启用时，此配置才生效。 取值：取值范围为[1, 999999999]，默认值为1，单位是秒。	否
reclaimPolicy	String	QoS 策略的回收策略。 取值： ● Delete: 解除 QoS 策略关联的所有对象或者删除 QoS 策略关联的所有对象，系统会触发删除 QoS 策略。 ● Retain: 解除 QoS 策略关联的所有对象或者删除 QoS 策略关联的所有对象，QoS 策略仍保留。 默认值为 Retain 。	否
description	String	QoS 策略的描述信息。 取值：1~256 位字符串。	否

请求示例

创建 QoS 策略 QoStest6。

<pre> POST /rest/v1/system/qos HTTP/1.1 Date: Tue, 01 Jul 2025 07:57:43 GMT Content-Type: application/json; charset=utf-8 Authorization: HBlock <i>userName:signature</i> Content-Length: 650 </pre>

Host: 192.168.0.64:1443

```
{
  "qosName": "QoStest6",
  "IOPS": 5000000,
  "readIOPS": 5000000,
  "writeIOPS": 6000000,
  "Bps": 4096000000,
  "readBps": 4096000000,
  "writeBps": 4096000000,
  "IOPSBurst": 99999999,
  "readIOPSBurst": 99999999,
  "writeIOPSBurst": 99999999,
  "BpsBurst": -1,
  "readBpsBurst": -1,
  "writeBpsBurst": -1,
  "IOPSBurstSecs": 99999999,
  "readIOPSBurstSecs": 99999999,
  "writeIOPSBurstSecs": 99999999,
  "BpsBurstSecs": 99999999,
  "readBpsBurstSecs": 99999999,
  "writeBpsBurstSecs": 99999999,
  "reclaimPolicy": "Delete",
  "description": "It is a test Qos policy"
}
```

响应示例

```
HTTP/1.1 204 No Content
x-hblock-request-id: 9f8da64f4a1947cf837a1eca46ce08da
Connection: keep-alive
Date: Tue, 01 Jul 2025 07:57:43 GMT
Server: HBlock
```

4.9.2 修改 QoS 策略

此操作用来修改 QoS 策略。

请求语法

```
PUT /rest/v1/system/qos/qosName HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "newName": qosName,
  "description": description,
  "IOPS": IOPS,
  "readIOPS": readIOPS,
  "writeIOPS": writeIOPS,
  "Bps": Bps,
  "readBps": readBps,
  "writeBps": writeBps,
  "IOPSBurst": IOPSBurst,
  "readIOPSBurst": readIOPSBurst,
  "writeIOPSBurst": writeIOPSBurst,
  "BpsBurst": BpsBurst,
  "readBpsBurst": readBpsBurst,
  "writeBpsBurst": writeBpsBurst,
  "IOPSBurstSecs": IOPSBurstSecs,
  "readIOPSBurstSecs": readIOPSBurstSecs,
  "writeIOPSBurstSecs": writeIOPSBurstSecs,
  "BpsBurstSecs": BpsBurstSecs,
  "readBpsBurstSecs": readBpsBurstSecs,
  "writeBpsBurstSecs": writeBpsBurstSecs,
  "reclaimPolicy": reclaimPolicy
}
```

请求参数

参数	类型	描述	是否必须
qosName	String	指定要修改的 QoS 策略名称。	是
newName	String	指定 QoS 策略的新名称。 取值：长度范围 1~64，只能由字母、数字和短横线(-)组成，区分大小写，且仅支持以字母或数字开头。	否
description	String	QoS 策略的描述信息。 取值：1~256 位字符串。	否
IOPS	Long	每秒能够进行读写操作次数的最大值。 取值：取值范围为[-1, 999999999]。-1 表示不限制。	否
readIOPS	Long	每秒能够进行读操作次数的最大值。 取值：取值范围为[-1, 999999999]。-1 表示不限制。	否
writeIOPS	Long	每秒能够进行写操作次数的最大值。 取值：取值范围为[-1, 999999999]。-1 表示不限制。	否
Bps	Long	每秒可传输数据量的最大值。 取值：取值范围为[-1, 4096000000000]，单位是 B/s。-1 表示不限制。	否
readBps	Long	读带宽上限。 取值：取值范围为[-1, 4096000000000]，单位是 B/s。-1 表示不限制。	否
writeBps	Long	写带宽上限。 取值：取值范围为[-1, 4096000000000]，单位是	否

		B/s。-1 表示不限制。	
IOPSBurst	Long	使用 Burst 功能时，每秒能够进行读写操作次数的最大值。 取值：只有当 IOPS 大于等于 1 时，此项设置为 -1 或(IOPS, 999999999]内的正整数方可生效。-1 表示不限制。	否
readIOPSBurst	Long	使用 Burst 功能时，每秒能够进行读操作次数的最大值。 取值：只有当 readIOPS 大于等于 1 时，此项设置为 -1 或(readIOPS, 999999999]内的正整数方可生效。-1 表示不限制。	否
writeIOPSBurst	Long	使用 Burst 功能时，每秒能够进行写操作次数的最大值。 取值：只有当 writeIOPS 大于等于 1 时，此项设置为 -1 或(writeIOPS, 999999999]内的正整数方可生效。-1 表示不限制。	否
BpsBurst	Long	使用 Burst 功能时，每秒可传输的数据量最大值。 取值：只有当 Bps 大于等于 1 时，此项设置为 -1 或(Bps, 4096000000000]内的正整数方可生效，单位是 B/s。-1 表示不限制。	否
readBpsBurst	Long	使用 Burst 功能时，读带宽上限。 取值：只有当 readBps 大于等于 1 时，此项设置为 -1 或(readBps, 4096000000000]内的正整数方可生效，单位是 B/s。-1 表示不限制。	否
writeBpsBurst	Long	使用 Burst 功能时，写带宽上限。 取值：只有当 writeBps 大于等于 1 时，此项设	否

		置为-1 或(<i>writeBps</i> , 4096000000000]内的正整数方可生效, 单位是 B/s。-1 表示不限制。	
IOPSBurstSecs	Long	使用 Burst 功能时, 按照 Burst 上限的能力进行读写操作所能持续的时间。 注意: 只有在 IOPS Burst 功能启用时, 此配置才生效。 取值: 取值范围为[1, 999999999], 单位是秒。	否
readIOPSBurstSecs	Long	使用 Burst 功能时, 按照 Burst 上限的能力进行读操作所能持续的时间。 注意: 只有在 read IOPS Burst 功能启用时, 此配置才生效。 取值: 取值范围为[1, 999999999], 单位是秒。	否
writeIOPSBurstSecs	Long	使用 Burst 功能时, 按照 Burst 上限的能力进行写操作所能持续的时间。 注意: 只有在 write IOPS Burst 功能启用时, 此配置才生效。 取值: 取值范围为[1, 999999999], 单位是秒。	否
BpsBurstSecs	Long	使用 Burst 功能时, 按照 Burst 上限的流量能力所能持续的时间。 注意: 只有在 Bps Burst 功能启用时, 此配置才生效。 取值: 取值范围为[1, 999999999], 单位是秒。	否
readBpsBurstSecs	Long	使用 Burst 功能时, 按照 Burst 上限的读流量能力所能持续的时间。 注意: 只有在 read Bps Burst 功能启用时, 此配置才生效。 取值: 取值范围为[1, 999999999], 单位是秒。	否

writeBpsBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的写流量能力所能持续的时间。 注意：只有在 write Bps Burst 功能启用时，此配置才生效。 取值：取值范围为[1, 999999999]，单位是秒。	否
reclaimPolicy	String	QoS 策略的回收策略。 取值： ● Delete: 解除 QoS 策略关联的所有对象或者删除 QoS 策略关联的所有对象，系统会触发删除 QoS 策略。 ● Retain: 解除 QoS 策略关联的所有对象或者删除 QoS 策略关联的所有对象，QoS 策略仍保留。	否

请求示例

修改 QoS 策略 QoSTest6。

<pre> PUT /rest/v1/system/qos/QoSTest6 HTTP/1.1 Date: Tue, 01 Jul 2025 10:11:31 GMT Content-Type: application/json; charset=utf-8 Authorization: HBlock <i>userName:signature</i> Content-Length: 612 Host: 192.168.0.64:1443 { "newName": "QoS6", "description": "It is QoS6", "IOPS": 5000000, "readIOPS": 5000000, "writeIOPS": 6000000, "Bps": 40960000000, "readBps": 40960000000, "writeBps": 40960000000, "IOPSBurst": -1, </pre>

```
"readIOPSBurst": -1,  
"writeIOPSBurst": -1,  
"BpsBurst": -1,  
"readBpsBurst": -1,  
"writeBpsBurst": -1,  
"IOPSBurstSecs": 999999999,  
"readIOPSBurstSecs": 999999999,  
"writeIOPSBurstSecs": 999999999,  
"BpsBurstSecs": 999999999,  
"readBpsBurstSecs": 999999999,  
"writeBpsBurstSecs": 999999999,  
"reclaimPolicy": "Delete"  
}
```

响应示例

```
HTTP/1.1 204 No Content  
x-hblock-request-id: 5416bb143c504e95986f9489c1d86272  
Connection: keep-alive  
Date: Tue, 01 Jul 2025 10:11:31 GMT  
Server: HBlock
```

4.9.3 设置卷关联的 QoS 策略

此操作用来设置卷关联的 QoS 策略。

注意： 关联的卷状态不能为 Deleting、Deletefailed。

说明：

- 当同一个卷设置多次 QoS 策略时，以最后一次操作为准。
- 对于集群版，QoS 策略的生效规则如下：
 - 若卷已关联了 QoS 策略，则以该策略为准。
 - 若卷未关联 QoS 策略：
 - ◆ 当卷同时存在缓存存储池和存储池时，采用缓存存储池设置的“存储池内卷的默认 QoS 策略”；若缓存存储池未设置“存储池内卷的默认 QoS 策略”，则卷无 QoS 策略。
 - ◆ 当卷仅存在存储池时，采用存储池设置的“存储池内卷的默认 QoS 策略”；若存储池未设置“存储池内卷的默认 QoS 策略”，则卷无 QoS 策略。

请求语法

```
PUT /rest/v1/system/qos/qosName/lun/assoc HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "list": [ LunName1, LunName2... ],
}
```

请求参数

参数	类型	描述	是否必须
qosName	String	指定 QoS 策略名称。	是

list	Array of string	指定关联 QoS 策略的卷名称。	是

请求示例

将设置卷的 QoS 策略。

```
PUT /rest/v1/system/qos/QoS6/lun/assoc HTTP/1.1
Date: Wed, 02 Jul 2025 03:22:28 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 43
Host: 192.168.0.64:1443

{
  "list": ["luna1","lunb1","lunb2"]
}
```

响应示例

```
HTTP/1.1 204 No Content
x-hblock-request-id: 3f0246e386ca42988afe8abc672bb789
Connection: keep-alive
Date: Wed, 02 Jul 2025 03:22:28 GMT
Server: HBlock
```

4.9.4 解除卷关联的 QoS 策略

此操作用来解除卷关联的 QoS 策略。

请求语法

```
PUT /rest/v1/system/qos/qosName/lun/disass HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "list": [ LunName1, LunName2... ],
}
```

请求参数

参数	类型	描述	是否必须
qosName	String	指定 QoS 策略名称。	是
list	Array of string	指定解除 QoS 策略关联的卷名称。	是

请求示例

解除卷关联的 QoS 策略。

```
PUT /rest/v1/system/qos/QoS6/lun/disass HTTP/1.1
Date: Wed, 02 Jul 2025 05:47:59 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 43
Host: 192.168.0.64:1443

{
  "list": ["luna1", "lunb1", "lunb2"]
}
```

```
}
```

响应示例

```
HTTP/1.1 204 No Content
x-hblock-request-id: 42dbc503bc2548448ad4dcf62b610d95
Connection: keep-alive
Date: Wed, 02 Jul 2025 05:47:59 GMT
Server: HBlock
```


4.9.5 设置存储池关联的 QoS 策略（集群版）

此操作用来设置存储池关联的 QoS 策略。

注意：存储池的状态不能为 Deleting。

说明：当同一个存储池设置多次 QoS 策略时，以最后一次操作为准。

请求语法

```
PUT /rest/v1/system/qos/qosName/storagepool/assoc HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "list": [ storagepool1,storagepool2... ],
}
```

请求参数

参数	类型	描述	是否必须
qosName	String	指定 QoS 策略名称。	是
list	Array of string	指定关联 QoS 策略的存储池名称。	是

请求示例

将 QoS 策略 QoS6 关联存储池。

```
PUT /rest/v1/system/qos/QoS6/storagepool/assoc HTTP/1.1
Date: Wed, 02 Jul 2025 03:36:45 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 37
Host: 192.168.0.64:1443
```

```
{  
  "list": ["default","pool1"]  
}
```

响应示例

```
HTTP/1.1 204 No Content  
x-hblock-request-id: 8c918e90ad1e4aae982eb19817481a24  
Connection: keep-alive  
Date: Wed, 02 Jul 2025 03:36:45 GMT  
Server: HBlock
```

4.9.6 解除存储池关联的 QoS 策略（集群版）

此操作用来解除存储池关联的 QoS 策略。

请求语法

```
PUT /rest/v1/system/qos/qosName/storagepool/disass HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "list": [ storagepool1,storagepool2... ],
}
```

请求参数

参数	类型	描述	是否必须
qosName	String	指定 QoS 策略名称。	是
list	Array of string	指定解除 QoS 策略关联的存储池名称。	是

请求示例

解除存储池关联的 QoS 策略。

```
PUT /rest/v1/system/qos/QoS6/storagepool/disass HTTP/1.1
Date: Wed, 02 Jul 2025 05:59:09 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 37
Host: 192.168.0.64:1443

{
  "list": ["default","pool1"]
}
```

```
}
```

响应示例

```
HTTP/1.1 204 No Content
x-hblock-request-id: 163fc68155bc4d40bb40f0bb2f6d9a92
Connection: keep-alive
Date: Wed, 02 Jul 2025 05:59:09 GMT
Server: HBlock
```

4.9.7 设置存储池内卷的默认 QoS 策略（集群版）

此操作用来设置存储池内卷的默认 QoS 策略。

注意：存储池的状态不能为 Deleting。

说明：当多次设置存储池内卷的默认 QoS 策略，以最后一次操作为准。

请求语法

```
PUT /rest/v1/system/qos/qosName/storagepoolforlun/assoc HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "list": [ storagepool1,storagepool2... ],
}
```

请求参数

参数	类型	描述	是否必须
qosName	String	指定 QoS 策略名称。	是
list	Array of string	指定存储池名称，设置该存储池内卷的默认 QoS 策略。	是

请求示例

设置存储池内卷的默认 QoS 策略。

```
PUT /rest/v1/system/qos/QoS6/storagepoolforlun/assoc HTTP/1.1
Date: Wed, 02 Jul 2025 04:01:21 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 37
Host: 192.168.0.64:1443
```

```
{  
  "list": ["default","pool1"]  
}
```

响应示例

```
HTTP/1.1 204 No Content  
x-hblock-request-id: e4cec28689ef42bc9cd8caac91134fdc  
Connection: keep-alive  
Date: Wed, 02 Jul 2025 04:01:21 GMT  
Server: HBlock
```

4.9.8 解除存储池内卷的默认 QoS 策略（集群版）

此操作用来解除存储池内卷的默认 QoS 策略。

请求语法

```
PUT /rest/v1/system/qos/qosName/storagepoolforlun/disass HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "list": [ storagepool1,storagepool2... ],
}
```

请求参数

参数	类型	描述	是否必须
qosName	String	指定 QoS 策略名称。	是
list	Array of string	指定存储池名称，解除该存储池内卷的默认 QoS 策略。	是

请求示例

解除存储池内卷的默认 QoS 策略。

```
PUT /rest/v1/system/qos/QoS6/storagepoolforlun/disass HTTP/1.1
Date: Wed, 02 Jul 2025 06:32:25 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 37
Host: 192.168.0.64:1443

{
  "list": ["default","pool1"]
}
```

```
}
```

响应示例

```
HTTP/1.1 204 No Content
x-hblock-request-id: 6293d0a0eef54948b6c78705a95fba9b
Connection: keep-alive
Date: Wed, 02 Jul 2025 06:32:25 GMT
Server: HBlock
```


4.9.9 删除 QoS 策略

此操作用来删除 QoS 策略。

注意：QoS 策略没有关联任何对象的时候，才可以删除。

请求语法

```
DELETE /rest/v1/system/qos/qosName HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

请求参数

参数	类型	描述	是否必须
qosName	String	指定要删除的 QoS 策略名称。	是

请求示例

删除 QoS 策略 qos2。

```
DELETE /rest/v1/system/qos/qos2 HTTP/1.1
Date: Wed, 02 Jul 2025 08:51:22 GMT
Connection: keep-alive
Authorization: HBlock userName:signature
Host: 192.168.0.64:1443
```

响应示例

```
HTTP/1.1 204 No Content
x-hblock-request-id: 7869c880cb014b6f9002e467ec6bf1b5
Connection: keep-alive
Date: Wed, 02 Jul 2025 08:51:22 GMT
Server: HBlock
```

4.9.10 批量查询 QoS 策略

此操作用来批量 QoS 策略。

请求语法

```
GET /rest/v1/system/qos?filter=filter&range=i-j HTTP/1.1
```

Date: *date*

Content-Type: application/json; charset=utf-8

Host: *ip:port*

Authorization: *authorization*

请求参数

参数	类型	描述	是否必须
filter	String	查询的过滤条件。 过滤条件由查询属性和属性值组成，其中属性和属性值之间用单个冒号分隔（key:value）表示模糊匹配，用两个冒号分隔（key::value）表示精确匹配。可以选择多个查询，如果是或的关系，使用“or”将查询条件分隔开；如果是与的关系，使用“and”将查询条件分隔开。 支持的过滤条件类型包括： ● qosName: 策略名称。	否
range	String	查询 QoS 策略的范围，格式为 i-j，i 和 j 为正整数，0<j-i<1000。按照 QoS 策略名称进行排序，返回第 i 到第 j 的 target。 如果输入的查询范围超过 QoS 策略的总个数，返回的结果为空，如果未指定 range，则返回所有符合条件的 QoS 策略。	否

响应结果

名称	类型	描述
qosPolicies	Array of qosPolicy	QoS 策略信息集合，详见“表 1 响应参数 qosPolicy 说明”。

表 1 响应参数 qosPolicy 说明

名称	类型	描述
qosName	String	QoS 策略名称。
IOPS	Long	每秒能够进行读写操作次数的最大值。 说明：-1 表示不限制。
readIOPS	Long	每秒能够进行读操作次数的最大值。 说明：-1 表示不限制。
writeIOPS	Long	每秒能够进行写操作次数的最大值。 说明：-1 表示不限制。
Bps	Long	每秒可传输数据量的最大值，单位是 B/s。 说明：-1 表示不限制。
readBps	Long	读带宽上限，单位是 B/s。 说明：-1 表示不限制。
writeBps	Long	写带宽上限，单位是 B/s。 说明：-1 表示不限制。
IOPSBurst	Long	使用 Burst 功能时，每秒能够进行读写操作次数的最大值。 说明：-1 表示不限制。
readIOPSBurst	Long	使用 Burst 功能时，每秒能够进行读操作次数的最大值。 说明：-1 表示不限制。
writeIOPSBurst	Long	使用 Burst 功能时，每秒能够进行写操作次数的最大

		值。 说明： -1 表示不限制。
BpsBurst	Long	使用 Burst 功能时，每秒可传输的数据量最大值，单位是 B/s。 说明： -1 表示不限制。
readBpsBurst	Long	使用 Burst 功能时，读带宽上限，单位是 B/s。 说明： -1 表示不限制。
writeBpsBurst	Long	使用 Burst 功能时，写带宽上限，单位是 B/s。 说明： -1 表示不限制。
IOPSBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的能力进行读写操作所能持续的时间。单位是秒。
readIOPSBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的能力进行读操作所能持续的时间。单位是秒。
writeIOPSBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的能力进行写操作所能持续的时间。单位是秒。
BpsBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的流量能力所能持续的时间。单位是秒。
readBpsBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的读流量能力所能持续的时间。单位是秒。
writeBpsBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的写流量能力所能持续的时间。单位是秒。
createTime	Long	QoS 策略创建的时间。
reclaimPolicy	String	QoS 策略的回收策略： ● Delete: 解除 QoS 策略关联的所有对象或者删除 QoS 策略关联的所有对象，系统会触发删除 QoS 策略。 ● Retain: 解除 QoS 策略关联的所有对象或者删除

		QoS 策略关联的所有对象，QoS 策略仍保留。
description	String	QoS 策略的描述信息。

请求示例

查询所有 QoS 策略。

```
GET /rest/v1/system/qos HTTP/1.1
Date: Thu, 03 Jul 2025 08:11:07 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.64:1443
```

响应示例

```
HTTP/1.1 200 OK
x-hblock-request-id: 81a1353ea5da4425b5e27a7d665651e3
Connection: keep-alive
Content-Length: 1586
Date: Thu, 03 Jul 2025 08:11:07 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "qosPolicies": [
      {
        "qosName": "QoS-test1",
        "readIOPSBurstSecs": 1,
        "writeIOPSBurstSecs": 1,
        "readBpsBurstSecs": 1,
        "writeBpsBurstSecs": 1,
        "createTime": 1751275484300,
        "reclaimPolicy": "Retain",
        "IOPS": 0,
        "IOPSBurstSecs": 1,
        "BpsBurstSecs": 1
      },
      {
```

```
    "qosName": "QoS6",
    "readIOPS": 5000000,
    "writeIOPS": 6000000,
    "readBps": 1073741824,
    "writeBps": 1073741824,
    "readIOPSBurst": 5000000,
    "writeIOPSBurst": 5000000,
    "readBpsBurst": 2147483648,
    "writeBpsBurst": 2147483648,
    "readIOPSBurstSecs": 10,
    "writeIOPSBurstSecs": 10,
    "readBpsBurstSecs": 100,
    "writeBpsBurstSecs": 100,
    "createTime": 1751437708609,
    "reclaimPolicy": "Retain",
    "description": "It is QoS6",
    "IOPS": 5000000,
    "Bps": 1073741824,
    "IOPSBurst": 5000000,
    "BpsBurst": 2147483648,
    "IOPSBurstSecs": 10,
    "BpsBurstSecs": 100
  },
  {
    "qosName": "QosPolicy20250701103509",
    "readIOPSBurst": 0,
    "readIOPSBurstSecs": 120,
    "writeIOPSBurstSecs": 1,
    "readBpsBurstSecs": 1,
    "writeBpsBurstSecs": 1,
    "createTime": 1751337350718,
    "reclaimPolicy": "Retain",
    "IOPSBurstSecs": 1,
    "BpsBurstSecs": 1
  },
  {
    "qosName": "qos11",
    "readIOPSBurstSecs": 1,
    "writeIOPSBurstSecs": 1,
    "readBpsBurstSecs": 1,
    "writeBpsBurstSecs": 1,
```

```
        "createTime": 1751443786042,
        "reclaimPolicy": "Retain",
        "IOPSBurstSecs": 1,
        "BpsBurstSecs": 1
    },
    {
        "qosName": "qostest2",
        "readIOPSBurstSecs": 1,
        "writeIOPSBurstSecs": 1,
        "readBpsBurstSecs": 1,
        "writeBpsBurstSecs": 1,
        "createTime": 1751342299027,
        "reclaimPolicy": "Retain",
        "Bps": 4398046511104,
        "IOPSBurstSecs": 1,
        "BpsBurstSecs": 1
    },
    {
        "qosName": "qostest3",
        "readIOPSBurstSecs": 1,
        "writeIOPSBurstSecs": 1,
        "readBpsBurstSecs": 1,
        "writeBpsBurstSecs": 1,
        "createTime": 1751342317324,
        "reclaimPolicy": "Retain",
        "Bps": 5497558138880,
        "IOPSBurstSecs": 1,
        "BpsBurstSecs": 1
    }
]
}
```

4.9.11 查询指定 QoS 策略

此操作用来查询指定 QoS 策略。

请求语法

```
GET /rest/v1/system/qos/qosName HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Host: ip:port
Authorization: authorization
```

请求参数

参数	类型	描述	是否必须
qosName	String	指定要查询的 QoS 策略名称。	是

响应结果

名称	类型	描述
qosName	String	QoS 策略名称。
IOPS	Long	每秒能够进行读写操作次数的最大值。 说明：-1 表示不限制。
readIOPS	Long	每秒能够进行读操作次数的最大值。 说明：-1 表示不限制。
writeIOPS	Long	每秒能够进行写操作次数的最大值。 说明：-1 表示不限制。
Bps	Long	每秒可传输数据量的最大值，单位是 B/s。 说明：-1 表示不限制。
readBps	Long	读带宽上限，单位是 B/s。

		说明：-1 表示不限制。
writeBps	Long	写带宽上限，单位是 B/s。 说明：-1 表示不限制。
IOPSBurst	Long	使用 Burst 功能时，每秒能够进行读写操作次数的最大值。 说明：-1 表示不限制。
readIOPSBurst	Long	使用 Burst 功能时，每秒能够进行读操作次数的最大值。 说明：-1 表示不限制。
writeIOPSBurst	Long	使用 Burst 功能时，每秒能够进行写操作次数的最大值。 说明：-1 表示不限制。
BpsBurst	Long	使用 Burst 功能时，每秒可传输的数据量最大值，单位是 B/s。 说明：-1 表示不限制。
readBpsBurst	Long	使用 Burst 功能时，读带宽上限，单位是 B/s。 说明：-1 表示不限制。
writeBpsBurst	Long	使用 Burst 功能时，写带宽上限，单位是 B/s。 说明：-1 表示不限制。
IOPSBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的能力进行读写操作所能持续的时间。单位是秒。
readIOPSBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的能力进行读操作所能持续的时间。单位是秒。
writeIOPSBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的能力进行写操作所能持续的时间。单位是秒。
BpsBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的流量能力所能持续的时间。单位是秒。

readBpsBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的读流量能力所能持续的时间。单位是秒。
writeBpsBurstSecs	Long	使用 Burst 功能时，按照 Burst 上限的写流量能力所能持续的时间。单位是秒。
createTime	Long	QoS 策略创建的时间。
reclaimPolicy	String	QoS 策略的回收策略： ● Delete: 解除 QoS 策略关联的所有对象或者删除 QoS 策略关联的所有对象，系统会触发删除 QoS 策略。 ● Retain: 解除 QoS 策略关联的所有对象或者删除 QoS 策略关联的所有对象，QoS 策略仍保留。
description	String	QoS 策略的描述信息。

请求示例

查询 QoS 策略 QoS6。

```
GET /rest/v1/system/qos/QoS6 HTTP/1.1
Date: Thu, 03 Jul 2025 08:44:58 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.64:1443
```

响应示例

```
HTTP/1.1 200 OK
x-hblock-request-id: bc5613e65a3544e996002a403674e331
Connection: keep-alive
Content-Length: 496
Date: Thu, 03 Jul 2025 08:44:58 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
```

```
"qosName": "QoS6",  
"readIOPS": 5000000,  
"writeIOPS": 6000000,  
"readBps": 1073741824,  
"writeBps": 1073741824,  
"readIOPSBurst": 5000000,  
"writeIOPSBurst": 5000000,  
"readBpsBurst": 2147483648,  
"writeBpsBurst": 2147483648,  
"readIOPSBurstSecs": 10,  
"writeIOPSBurstSecs": 10,  
"readBpsBurstSecs": 100,  
"writeBpsBurstSecs": 100,  
"createTime": 1751437708609,  
"reclaimPolicy": "Retain",  
"description": "It is QoS6",  
"IOPS": 5000000,  
"Bps": 1073741824,  
"IOPSBurst": 5000000,  
"BpsBurst": 2147483648,  
"IOPSBurstSecs": 10,  
"BpsBurstSecs": 100
```

```
}
```

```
}
```

4.9.12 查询 QoS 策略关联/可关联的对象信息

此操作用来查询 QoS 策略关联/可关联的对象信息。

请求语法

```
GET /rest/v1/system/qos/qosName?object=object&status=status HTTP/1.1  
Date: date  
Content-Type: application/json; charset=utf-8  
Host: ip:port  
Authorization: authorization
```

请求参数

参数	类型	描述	是否必须
qosName	String	指定要查询的 QoS 策略名称。	是
object	String	指定关联的对象类型。 取值： ● lun: 卷。 ● storagepool（仅集群版支持）：存储池。 ● storagepoolforlun（仅集群版支持）：存储池内卷的默认 QoS 策略。	是
status	String	对象与 QoS 策略的关联状态。 取值： ● associated: 已关联。 ● available: 可关联。 默认值为 associated。	否

响应结果

名称	类型	描述
totalNum	Integer	对象总个数。
luns	Array of lun	卷信息集合。详见“表 1 响应参数 lun 说明”。 说明：如果查询的对象是 lun，才输出此项。
storagepools	Array of storagepool	存储池信息集合。详见“表 2 响应参数 storagepool 说明”。 说明：如果查询的对象是 storagepool、storagepoolforlun，才输出此项。

表 1 响应参数 lun 说明

名称	类型	描述
lunName	String	卷名称。
status	String	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。 ● Importing：正在导入备份数据。 ● Wiping：卷正在清空数据。 ● WipeFailed：清空数据失败。

capacity	Integer	卷容量，单位是 GiB。
storageMode	String	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云端。 ● Storage: 存储模式，本地保留全部数据，并异步存储到云端。
config.localStorageClass	String	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 $N+M$。
config.ECfragmentSize	Integer	分片大小（仅集群版支持）。如果冗余模式为 EC $N+M$ ，则会返回此项。

表 2 响应参数 **storagepool** 说明

名称	类型	描述
name	String	存储池名称。
isBasePool	Boolean	是否是基础存储池： ● true: 是基础存储池。 ● false: 不是基础存储池。
status	String	存储池状态： ● Normal: 正常。 ● Deleting: 删除中。
faultDomain	String	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。
totalCapacity	Long	存储池总容量, 单位是 bytes。
usedCapacity	Long	存储池已用容量, 单位是 bytes。

请求示例 1

单机版: 查询所有 QoS 策略 QoS1 关联的卷。

```
GET /rest/v1/system/qos/QoS1?object=lun HTTP/1.1
Date: Fri, 15 Aug 2025 07:48:10 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.66:1443
```

响应示例 1

```
HTTP/1.1 200 OK
x-hblock-request-id: 3be0bb06fb144b25a3d77beba4bf24f7
Connection: keep-alive
Content-Length: 182
Date: Fri, 15 Aug 2025 07:48:10 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "totalNum": 2,
    "luns": [
      {
        "lunName": "luna1",
        "status": "Normal",
        "capacity": 100,
        "storageMode": "Local"
      },
      {
        "lunName": "lunc3",
        "status": "Normal",
```

```
        "capacity": 390,  
        "storageMode": "Cache"  
    }  
]  
}  
}
```

请求示例 2

集群版：查询所有 QoS 策略 QoS6 关联的卷。

```
GET /rest/v1/system/qos/QoS6?object=lun HTTP/1.1
```

```
Date: Tue, 05 Aug 2025 07:59:50 GMT
```

```
Authorization: HBlock userName:signature
```

```
Host: 192.168.0.64:1443
```

响应示例 2

```
HTTP/1.1 200 OK
```

```
x-hblock-request-id: 731afda1b93c4b3186ba1c9d5ab181ed
```

```
Connection: keep-alive
```

```
Content-Length: 447
```

```
Date: Tue, 05 Aug 2025 07:59:50 GMT
```

```
Content-Type: application/json;charset=utf-8
```

```
Server: HBlock
```

```
{  
  "data": {  
    "totalNum": 3,  
    "luns": [  
      {  
        "lunName": "lun01a",  
        "status": "Normal",  
        "capacity": 200,  
        "storageMode": "Local",  
        "config": {  
          "localStorageClass": "EC 2+1",  
          "ECfragmentSize": 16  
        }  
      },  
    ],  
  },  
}
```



```
{
  {
    "lunName": "lunn01a-clone1",
    "status": "Normal",
    "capacity": 200,
    "storageMode": "Local",
    "config": {
      "localStorageClass": "EC 2+1",
      "ECfragmentSize": 16
    }
  },
  {
    "lunName": "lun04",
    "status": "Normal",
    "capacity": 404,
    "storageMode": "Local",
    "config": {
      "localStorageClass": "EC 2+1",
      "ECfragmentSize": 16
    }
  }
]
}
```

请求示例 3

集群版：查询所有 QoS 策略 QoS6 关联的存储池。

```
GET /rest/v1/system/qos/QoS6?object=storagepool HTTP/1.1
Date: Fri, 04 Jul 2025 03:27:32 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.64:1443
```

响应示例 3

```
HTTP/1.1 200 OK
x-hblock-request-id: 18732cfc7cb74914a89c79ca1f324316
Connection: keep-alive
Content-Length: 286
Date: Fri, 04 Jul 2025 03:27:32 GMT
Content-Type: application/json;charset=utf-8
```

```
Server: HBlock

{
  "data": {
    "totalNum": 2,
    "storagepools": [
      {
        "name": "default",
        "isBasePool": true,
        "status": "Normal",
        "faultDomain": "server",
        "totalCapacity": 299111325696,
        "usedCapacity": 70197202944
      },
      {
        "name": "pool1",
        "isBasePool": false,
        "status": "Normal",
        "faultDomain": "server",
        "totalCapacity": 0,
        "usedCapacity": 0
      }
    ]
  }
}
```

请求示例 4

集群版：查询设置存储池内卷的默认 QoS 策略为 QoS6 的存储池。

```
GET /rest/v1/system/qos/QoS6?object=storagepoolforlun HTTP/1.1
Date: Fri, 04 Jul 2025 03:31:06 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.64:1443
```

响应示例 4

```
HTTP/1.1 200 OK
x-hblock-request-id: e943ad21499844549fd553cc4ef86b57
Connection: keep-alive
Content-Length: 286
```

```
Date: Fri, 04 Jul 2025 03:31:06 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "totalNum": 2,
    "storagepools": [
      {
        "name": "pool1",
        "isBasePool": false,
        "status": "Normal",
        "faultDomain": "server",
        "totalCapacity": 0,
        "usedCapacity": 0
      },
      {
        "name": "default",
        "isBasePool": true,
        "status": "Normal",
        "faultDomain": "server",
        "totalCapacity": 299111325696,
        "usedCapacity": 70197215232
      }
    ]
  }
}
```

4.9.13 错误码列表

HTTP status	错误码	错误信息	描述
400	CanNotDeleteQoS	The QoS policy <i>Name</i> cannot be deleted. Please disassociate all the dependencies and try again.	QoS 有关联的对象，不能删除。
400	InvalidQoSName	Value <i>value</i> at 'qos name' failed to satisfy the constraint: Argument must contain only letters, digits, or hyphens (-) and does not exceed 64 characters, must begin with a letter or digit.	QoS 策略名称不合法，请重新输入。
400	MissingRelevantArgument	Value null at ' <i>argument1</i> ' failed to satisfy constraint: Argument must not be null when ' <i>argument2</i> ' has value.	参数 1 和参数 2 必须同时设置。
404	NoSuchQoS	The QoS policy with name <i>Name</i> does not exist.	指定的 QoS 策略不存在。
409	QoSAlreadyExists	The QoS policy with name <i>Name</i> already exists.	QoS 策略已存在，请重新输入。

4.10 集群拓扑（集群版）

4.10.1 创建拓扑节点

此操作用来创建拓扑节点。

请求语法

```
POST /rest/v1/system/topology HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "nodeName": nodeName,
  "nodeType": nodeType,
  "description": description,
  "parentName": nodeName
}
```

请求参数

参数	类型	描述	是否必须
nodeName	String	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。	是
nodeType	String	拓扑节点的类型。 取值： ● rack：机架类型。 ● room：机房类型。	是

description	String	节点描述信息。 取值：1~50 位字符串。	否
parentName	String	父节点名称。 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。 默认为根节点。	否

请求示例

创建节点 rack3。

```
POST /rest/v1/system/topology HTTP/1.1
Date: Mon, 24 Jun 2024 09:34:29 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 119
Host: 192.168.0.110:1443

{
  "nodeName": "rack3",
  "nodeType": "rack",
  "description": "The new rack.",
  "parentName": "room-3"
}
```

响应示例

```
HTTP/1.1 200 OK
x-hblock-request-id: e59fb36c5d3e4a72a93d8eb8caa3ee7e
Connection: keep-alive
Content-Length: 0
Date: Mon, 24 Jun 2024 09:34:29 GMT
Server: HBlock
```

4.10.2 修改拓扑节点信息

此操作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。

请求语法

```
PUT /rest/v1/system/topology/nodeName HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "newName": nodeName,
  "description": description,
  "parentName": nodeName
}
```

请求参数

参数	类型	描述	是否必须
nodeName	String	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。	是
newName	String	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。	否
description	String	节点描述信息。	否

		取值：1~50 位字符串。	
parentName	String	父节点名称。 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。 默认为根节点。	否

请求示例

修改节点 rack3。

```
PUT /rest/v1/system/topology/rack3 HTTP/1.1
Date: Mon, 24 Jun 2024 10:10:38 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 94
Host: 192.168.0.110:1443

{
  "newName": "rack-3",
  "description": "The new rack.",
  "parentName": "room-3"
}
```

响应示例

```
HTTP/1.1 204 No Content
x-hblock-request-id: 9182545a7fb2455da83cebb3177cf8a0
Connection: keep-alive
Date: Mon, 24 Jun 2024 10:10:38 GMT
Server: HBlock
```


4.10.3 删除拓扑节点

此操作用来删除拓扑节点。

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，nodeName 可以带上父节点的名称，否则可能找不到对应的节点。
- 此操作不支持删除 path 节点，如果需要删除 path 节点，请使用[移除数据目录](#)。

请求语法

```
DELETE /rest/v1/system/topology/nodeName?force=force&deleteLocalData=deleteLocalData HTTP/1.1
Date: date
Connection: keep-alive
Host: ip:port
Authorization: authorization
```

请求参数

参数	类型	描述	是否必须
nodeName	String	要删除的拓扑节点。	是
force	Boolean	是否强制删除拓扑节点。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。 取值： ● true：是。	否

		● false: 否。 默认值为 false 。	
<code>deleteLocalData</code>	Boolean	删除 server 类型的拓扑节点时，是否删除拓扑节点上 HBlock 数据目录中的数据。仅删除 server 类型的节点时支持此参数。 取值： ● true: 删除拓扑节点上 HBlock 数据。 ● false: 不删除拓扑节点上 HBlock 数据。 默认值为 false 。	否

请求示例

删除节点 rack-3。

```
DELETE /rest/v1/system/topology/rack-3 HTTP/1.1
Date: Mon, 24 Jun 2024 10:21:24 GMT
Connection: keep-alive
Authorization: HBlock userName:signature
Host: 192.168.0.110:1443
```

响应示例

```
HTTP/1.1 202 Accepted
x-hblock-request-id: b779af1c40d942b6b2f70819a42c1679
Connection: keep-alive
Content-Length: 0
Date: Mon, 24 Jun 2024 10:21:24 GMT
Server: HBlock
```

4.10.4 查询整个拓扑

此操作用来查询整个拓扑。

请求语法

```
GET /rest/v1/system/topology HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Host: ip:port
Authorization: authorization
```

响应结果

名称	类型	描述
name	String	拓扑节点名称。
fullname	String	节点在集群中的全路径名称，从根节点开始，使用 <i>name:name:name</i> 格式来唯一标识该节点的名称。
id	String	拓扑节点 ID。
type	String	节点的类型： root ：根节点。
description	String	节点描述。
childNodes	Array of childNodes	子节点信息集合，根据子节点类型不同，展示的信息不同，详见“表 1 响应参数 ChildNode 说明”。

表 1 响应参数 ChildNode 说明

名称		类型	描述
子节点类型 为 room	name	String	子节点名称。
	fullname	String	节点在集群中的全路径名称，从根节点开始，使用 <i>name:name</i> 格式来唯一标识该节点的名称。

	id	String	节点 ID。
	type	String	节点类型。
	description	String	节点描述。
子节点类型为 rack	name	String	子节点名称。
	fullname	String	节点在集群中的全路径名称，从根节点开始，使用 <i>name:name:name</i> 格式来唯一标识该节点的名称。
	id	String	节点 ID。
	type	String	节点类型。
	description	String	节点描述。
子节点类型为 server	name	String	子节点名称。
	fullname	String	节点在集群中的全路径名称，从根节点开始，使用 <i>name:name:name</i> 格式来唯一标识该节点的名称。
	id	String	节点 ID。
	serverId	String	服务器 ID。
	type	String	节点类型。
	description	String	节点描述。
	clusterAddress	String	拓扑图集群网络 IP。
	publicAddress	String	拓扑图业务网络 IP。
	allDiskPaths.usedCapacity	Long	服务器上 HBlock 数据目录对应分区的已用容量，单位是 bytes。
	allDiskPaths.totalCapacity	Long	服务器上 HBlock 数据目录对应分区的总容量，单位是 bytes。
	serverStatus	String	服务器状态： ● Connected: 已连接。 ● Disconnected: 未连接。

			● Removing: 移除中。
子节点类型为 path	name	String	子节点名称。
	fullname	String	节点在集群中的全路径名称，从根节点开始，使用 <i>name:name:name</i> 格式来唯一标识该节点的名称。
	id	String	节点 ID。
	serverId	String	服务器 ID。
	type	String	节点类型。
	totalCapacity	Long	HBlock 数据目录对应分区的总容量，单位是 bytes。
	usedCapacity	Long	HBlock 数据目录对应分区的已用容量，单位是 bytes。
	capacityQuota	Long	数据目录容量配额。 负整数表示无限制写入。
	usedCapacityQuota	Long	数据目录已用容量配额。
	storagePool	String	所属存储池。
	healthStatus	String	数据目录健康状态： ● 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；HBlock 对这个目录停写；数据目录配额使用率超过阈值（系统默认值为 95%）；数据目录配额为 0。

			● 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。
	healthDetail	Array of string	数据目录健康状态详情: ● 如果健康状态为 Healthy, 此字段为空。 ● 如果健康状态为 Warning 或 Error, 显示警告或错误的详细信息。

请求示例

查询整个拓扑。

```
GET /rest/v1/system/topology HTTP/1.1
Date: Fri, 2 Aug 2024 10:10:18 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Host: 192.168.0.110:1443
```

响应示例

```
HTTP/1.1 200 OK
x-hblock-request-id: ea0f7997eeeb415087c269c61681e4bc
Connection: keep-alive
Content-Length: 4234
Date: Fri, 2 Aug 2024 10:10:18 GMT
Content-Type: application/json; charset=utf-8
Server: HBlock

{
  "data": {
    "name": "default",
    "fullname": "default",
    "id": "e69eb32c-9311-43fd-9b9e-01af53bf32c1",
    "type": "root",
    "childNodes": [
      {
        "name": "rack1",
```

```
"fullname": "default:rack1",
"id": "568156af-0c15-4984-ac5f-ea877db752e5",
"type": "rack",
"description": "rack1",
"childNodes": [
  {
    "name": "server4",
    "fullname": "default:rack1:server4",
    "id": "2257dc92-fafe-4e91-9000-2c127ae13f68",
    "serverId": "hblock_4",
    "type": "server",
    "clusterAddress": "192.168.0.117",
    "publicAddress": "192.168.0.117",
    "allDiskPaths": {
      "usedCapacity": 29337317376,
      "totalCapacity": 131883315200
    },
    "serverStatus": "Connected",
    "childNodes": [
      {
        "name": "/mnt/storage02",
        "fullname": "default:rack1:server4:/mnt/storage02",
        "id": "fddcbc2b-808f-4ed0-b9f4-1fa61914011c",
        "serverId": "hblock_4",
        "type": "path",
        "totalCapacity": 56028450816,
        "usedCapacity": 57221120,
        "capacityQuota": -1,
        "usedCapacityQuota": 69632,
        "storagePool": "pool2",
        "healthStatus": "Healthy",
        "healthDetail": []
      },
      {
        "name": "/mnt/storage03",
        "fullname": "default:rack1:server4:/mnt/storage03",
        "id": "cec39c97-8b67-414a-adc7-4e9275d091f7",
        "serverId": "hblock_4",
        "type": "path",
        "totalCapacity": 3927900160,
        "usedCapacity": 16855040,
```

```
        "capacityQuota": -1
      },
      {
        "name": "/mnt/storage04",
        "fullname": "default:rack1:server4:/mnt/storage04",
        "id": "22eb96ff-ea92-4e22-b1f0-1ed42f2409aa",
        "serverId": "hblock_4",
        "type": "path",
        "totalCapacity": 35963482112,
        "usedCapacity": 14631620608,
        "capacityQuota": -1,
        "usedCapacityQuota": 69632,
        "storagePool": "pool2",
        "healthStatus": "Healthy",
        "healthDetail": []
      },
      {
        "name": "/mnt/storage05",
        "fullname": "default:rack1:server4:/mnt/storage05",
        "id": "d7e3dd03-e127-4e3e-b619-73fcfd4956fd",
        "serverId": "hblock_4",
        "type": "path",
        "totalCapacity": 35963482112,
        "usedCapacity": 14631620608,
        "capacityQuota": -1,
        "usedCapacityQuota": 69632,
        "storagePool": "pool2",
        "healthStatus": "Healthy",
        "healthDetail": []
      }
    ]
  },
  {
    "name": "room1",
    "fullname": "default:room1",
    "id": "00457811-af8e-4f39-ba6c-f856cdc173f8",
    "type": "room",
    "childNodes": [
      {
```



```
"name": "rack2",
"fullname": "default:room1:rack2",
"id": "3f591741-a387-44d2-8ae8-36c4ba546bd2",
"type": "rack",
"childNodes": [
  {
    "name": "server1",
    "fullname": "default:room1:rack2:server1",
    "id": "36a55e61-47e5-43f0-9d73-357a86d3ff53",
    "serverId": "hblock_1",
    "type": "server",
    "clusterAddress": "192.168.0.192",
    "publicAddress": "192.168.0.192",
    "allDiskPaths": {
      "usedCapacity": 18890235904,
      "totalCapacity": 136132825088
    },
    "serverStatus": "Connected",
    "childNodes": [
      {
        "name": "/mnt/stor",
        "fullname": "default:room1:rack2:server1:/mnt/stor",
        "id": "d14f3938-a6de-4950-87bc-e461d743284a",
        "serverId": "hblock_1",
        "type": "path",
        "totalCapacity": 100169342976,
        "usedCapacity": 11510775808,
        "capacityQuota": -1,
        "usedCapacityQuota": 172032,
        "storagePool": "default",
        "healthStatus": "Healthy",
        "healthDetail": []
      },
      {
        "name": "/mnt/storage01",
        "fullname":
"default:room1:rack2:server1:/mnt/storage01",
        "id": "9999d96e-1032-404c-85de-4c76f8100e12",
        "serverId": "hblock_1",
        "type": "path",
        "totalCapacity": 35963482112,
```

```
        "usedCapacity": 7379460096,
        "capacityQuota": -1,
        "usedCapacityQuota": 106496,
        "storagePool": "default",
        "healthStatus": "Healthy",
        "healthDetail": []
      }
    ]
  },
  {
    "name": "server2",
    "fullname": "default:room1:server2",
    "id": "444dd9a4-88db-4df5-9959-88d7b82edad8",
    "serverId": "hblock_2",
    "type": "server",
    "clusterAddress": "192.168.0.110",
    "publicAddress": "192.168.0.110",
    "allDiskPaths": {
      "usedCapacity": 8582782976,
      "totalCapacity": 100169342976
    },
    "serverStatus": "Connected",
    "childNodes": [
      {
        "name": "/mnt/stor",
        "fullname": "default:room1:server2:/mnt/stor",
        "id": "1451c75b-2e6f-4cbe-82ab-962019247f35",
        "serverId": "hblock_2",
        "type": "path",
        "totalCapacity": 100169342976,
        "usedCapacity": 8582782976,
        "capacityQuota": -1,
        "usedCapacityQuota": 229376,
        "storagePool": "default",
        "healthStatus": "Healthy",
        "healthDetail": []
      }
    ]
  },
}
```

```
{
  {
    "name": "server3",
    "fullname": "default:room1:server3",
    "id": "f89a94bc-2cc3-4ac9-b5dc-13d227b40cf4",
    "serverId": "hblock_3",
    "type": "server",
    "clusterAddress": "192.168.0.102",
    "publicAddress": "192.168.0.102",
    "allDiskPaths": {
      "usedCapacity": 11067289600,
      "totalCapacity": 100169342976
    },
    "serverStatus": "Connected",
    "childNodes": [
      {
        "name": "/mnt/stor",
        "fullname": "default:room1:server3:/mnt/stor",
        "id": "effad142-9e34-43f4-8390-ccb5c211f60f",
        "serverId": "hblock_3",
        "type": "path",
        "totalCapacity": 100169342976,
        "usedCapacity": 11067289600,
        "capacityQuota": -1,
        "usedCapacityQuota": 229376,
        "storagePool": "default",
        "healthStatus": "Healthy",
        "healthDetail": []
      }
    ]
  }
}
```

4.10.5 查询拓扑节点

此操作用来查询指定拓扑节点。

请求语法

```
GET /rest/v1/system/topology/nodeName HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Host: ip:port
Authorization: authorization
```

请求参数

参数	类型	描述	是否必须
nodeName	String	要查询的拓扑节点。 不支持查询 path 级别的节点信息。	是

响应结果

名称	类型	描述
name	String	拓扑节点名称。
fullname	String	节点在集群中的全路径名称，从根节点开始，使用 <i>name:name:name</i> 格式来唯一标识该节点的名称。
id	String	拓扑节点 ID。
type	String	节点的类型： ● server：服务器类型。 ● rack：机架类型。 ● room：机房类型。 ● root：根节点。

description	String	节点描述。
childNodes	Array of childNode	子节点信息集合，根据子节点类型不同，展示的信息不同，详见“表 1 响应参数 childNode 说明”。

表 1 响应参数 childNode 说明

名称		类型	描述
子节点 类型为 room	name	String	子节点名称。
	fullname	String	节点在集群中的全路径名称，从根节点开始，使用 <i>name:name</i> 格式来唯一标识该节点的名称。
	id	String	节点 ID。
	type	String	节点类型。
	description	String	节点描述。
子节点 类型为 rack	name	String	子节点名称。
	fullname	String	节点在集群中的全路径名称，从根节点开始，使用 <i>name:name:name</i> 格式来唯一标识该节点的名称。
	id	String	节点 ID。
	type	String	节点类型。
	description	String	节点描述。
子节点 类型为 server	name	String	子节点名称。
	fullname	String	节点在集群中的全路径名称，从根节点开始，使用 <i>name:name:name</i> 格式来唯一标识该节点的名称。
	id	String	节点 ID。
	serverId	String	服务器 ID。
	type	String	节点类型。

	description	String	节点描述。
	clusterAddress	String	拓扑图集群网络 IP。
	publicAddress	String	拓扑图业务网络 IP。
	allDiskPaths.usedCapacity	Long	服务器上 HBlock 数据目录对应分区的已用容量，单位是 bytes。
	allDiskPaths.totalCapacity	Long	服务器上 HBlock 数据目录对应分区的总容量，单位是 bytes。
	serverStatus	String	服务器状态： ● Connected: 已连接。 ● Disconnected: 未连接。 ● Removing: 移除中。
子节点 类型为 path	name	String	子节点名称。
	fullname	String	节点在集群中的全路径名称，从根节点开始，使用 <i>name:name:name</i> 格式来唯一标识该节点的名称。
	id	String	节点 ID。
	serverId	String	服务器 ID。
	type	String	节点类型。
	totalCapacity	Long	HBlock 数据目录对应分区的总容量，单位是 bytes。
	usedCapacity	Long	HBlock 数据目录对应分区的已用容量，单位是 bytes。
	capacityQuota	Long	数据目录容量配额。 负整数表示无限制写入。
	usedCapacityQuota	Long	数据目录已用容量配额。
	storagePool	String	所属存储池。
	healthStatus	String	数据目录健康状态：

			● 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；HBlock 对这个目录停写；数据目录配额使用率超过阈值（系统默认值为 95%）；数据目录配额为 0。 ● 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。
	healthDetail	Array of string	数据目录健康状态详情： ● 如果健康状态为 Healthy，此字段为空。 ● 如果健康状态为 Warning 或 Error，显示警告或错误的详细信息。

请求示例

查询节点 room1。

```
GET /rest/v1/system/topology/room1 HTTP/1.1
Date: Fri, 2 Aug 2024 10:12:05 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Host: 192.168.0.110:1443
```

响应示例

```
HTTP/1.1 200 OK
x-hblock-request-id: a04f0b22a3844518afb8ab61e5bc3757
Connection: keep-alive
Content-Length: 2480
Date: Fri, 2 Aug 2024 10:12:05 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock
```

```
{
  "data": {
    "name": "room1",
    "fullname": "default:room1",
    "id": "00457811-af8e-4f39-ba6c-f856cdc173f8",
    "type": "room",
    "childNodes": [
      {
        "name": "rack2",
        "fullname": "default:room1:rack2",
        "id": "3f591741-a387-44d2-8ae8-36c4ba546bd2",
        "type": "rack",
        "childNodes": [
          {
            "name": "server1",
            "fullname": "default:room1:rack2:server1",
            "id": "36a55e61-47e5-43f0-9d73-357a86d3ff53",
            "serverId": "hblock_1",
            "type": "server",
            "clusterAddress": "192.168.0.192",
            "publicAddress": "192.168.0.192",
            "allDiskPaths": {
              "usedCapacity": 18887569408,
              "totalCapacity": 136132825088
            },
            "serverStatus": "Connected",
            "childNodes": [
              {
                "name": "/mnt/stor",
                "fullname": "default:room1:rack2:server1:/mnt/stor",
                "id": "d14f3938-a6de-4950-87bc-e461d743284a",
                "serverId": "hblock_1",
                "type": "path",
```



```
        "totalCapacity": 100169342976,
        "usedCapacity": 11510775808,
        "capacityQuota": -1,
        "usedCapacityQuota": 172032,
        "storagePool": "default",
        "healthStatus": "Healthy",
        "healthDetail": []
      },
      {
        "name": "/mnt/storage01",
        "fullname": "default:room1:rack2:server1:/mnt/storage01",
        "id": "9999d96e-1032-404c-85de-4c76f8100e12",
        "serverId": "hblock_1",
        "type": "path",
        "totalCapacity": 35963482112,
        "usedCapacity": 7376793600,
        "capacityQuota": -1,
        "usedCapacityQuota": 106496,
        "storagePool": "default",
        "healthStatus": "Healthy",
        "healthDetail": []
      }
    ]
  },
  {
    "name": "server2",
    "fullname": "default:room1:server2",
    "id": "444dd9a4-88db-4df5-9959-88d7b82edad8",
    "serverId": "hblock_2",
    "type": "server",
    "clusterAddress": "192.168.0.110",
    "publicAddress": "192.168.0.110",
    "allDiskPaths": {
      "usedCapacity": 8582782976,
      "totalCapacity": 100169342976
    },
    "serverStatus": "Connected",
    "childNodes": [
      {
```

```
        "name": "/mnt/stor",
        "fullname": "default:room1:server2:/mnt/stor",
        "id": "1451c75b-2e6f-4cbe-82ab-962019247f35",
        "serverId": "hblock_2",
        "type": "path",
        "totalCapacity": 100169342976,
        "usedCapacity": 8582782976,
        "capacityQuota": -1,
        "usedCapacityQuota": 229376,
        "storagePool": "default",
        "healthStatus": "Healthy",
        "healthDetail": []
      }
    ]
  },
  {
    "name": "server3",
    "fullname": "default:room1:server3",
    "id": "f89a94bc-2cc3-4ac9-b5dc-13d227b40cf4",
    "serverId": "hblock_3",
    "type": "server",
    "clusterAddress": "192.168.0.102",
    "publicAddress": "192.168.0.102",
    "allDiskPaths": {
      "usedCapacity": 11067289600,
      "totalCapacity": 100169342976
    },
    "serverStatus": "Connected",
    "childNodes": [
      {
        "name": "/mnt/stor",
        "fullname": "default:room1:server3:/mnt/stor",
        "id": "effad142-9e34-43f4-8390-ccb5c211f60f",
        "serverId": "hblock_3",
        "type": "path",
        "totalCapacity": 100169342976,
        "usedCapacity": 11067289600,
        "capacityQuota": -1,
        "usedCapacityQuota": 229376,
        "storagePool": "default",
        "healthStatus": "Healthy",
```

```
    "healthDetail": []  
  }  
]  
}  
]
```

```
}
```

4.10.6 错误码列表

HTTP status	错误码	错误信息	描述
400	CanNotDeleteNode	The node <i>nodeName</i> can not be deleted. Please remove its child nodes and try again.	该节点不能移除，需要先移除它的子节点后才能移除该节点。
400	CanNotMoveNode	The node <i>nodeName</i> can not be moved. Please make sure the node type is not lower than the fault domain level of storage pool which its leaf node belongs to.	节点不满足重新映射条件，请修改后重试。
400	DuplicatedNode	The node with name <i>nodeName</i> is not unique in the topology.	节点名称在集群中不唯一。
400	InvalidEnumValue	Value <i>value</i> at 'node type' failed to satisfy constraint: Argument must satisfy enum value set: [<i>value1</i> , <i>value2</i> ...]	节点类型错误，不符合枚举值。
400	InvalidFile	Failed to parse file <i>filename</i> .	文件格式不正确。
400	InvalidNodeName	Value <i>value</i> at 'node name' failed to satisfy constraint: Argument must contain only letters, digits, dots (.), underscores (<code>_</code>) or hyphens (<code>-</code>)	节点名称不正确。长度范围 1~63，只能由字母、数字、句点(.)、下划线(<code>_</code>)和短横线(<code>-</code>)组

		), and does not exceed 63 characters, must begin with a letter or digit.	成，字母区分大小写，且仅支持以字母或数字开头。
400	NodeAlreadyExists	The node with name <i>nodeName</i> has already exists in the topology.	节点名称已经在拓扑图中存在。
400	NodeNotAllowed	The type of node <i>nodeName</i> is <i>type</i> , the request is invalid.	节点名称和节点类型不匹配。
400	NoSuchNode	The node with name <i>nodeName</i> does not exist.	节点不存在。
400	NodeTypeNotMatch	Node type does not match. Please check parent node type.	节点类型不匹配，请检查父节点。
400	RemoveNodeRisk	The node <i>nodeName</i> or its child nodes are being used by a storage pool. Removing the node may cause data loss. You can use force remove. When using this option, there may be a risk of data loss.	移除节点可能会引起数据丢失。只能使用强制移除。请注意，强制移除，会产生数据丢失风险，请谨慎操作。
400	StandaloneModeNotAllowed	' <i>operation</i> ' is not supported by standalone mode of HBlock.	单机版本 HBlock 不支持该操作。
400	SystemCrashRisk	The node <i>nodename</i> can not be removed because there is only one available fault	无法移除节点，存在以下情况之一，移除节点可能引发

		domain left in the base storage pool, or the node is related to multiple fault domains in the base pool, removing node may result in a system crash.	系统崩溃：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；节点涉及基础存储池的多个故障域。

4.11 服务器管理

4.11.1 添加服务器（集群版适用）

此操作用来向集群中添加新的服务器。

请先在待添加的服务器上执行下列步骤：

- 1) 拷贝安装包至安装路径。
- 2) 解压安装包。
- 3) 执行 `./stor install` 命令安装。

然后在集群中的任意一台服务器上，执行添加服务器的操作。

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（portRange）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。

请求语法

```
POST /rest/v1/system/server HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "nodeName": nodeName,
  "parentName": nodeName,
  "description": description,
  "ip": ip,
```

```
"apiPort": apiPort,
"iSCSIPort": iSCSIPort,
"portRange": port1-port2,
"diskPaths": [
  { "path":path,
    "capacityQuota": capacityvalue
  },
  { "path":path,
    "capacityQuota": capacityvalue
  },
  .....
],
"ports":{
  "dataPort1": dataPort1,
  "managementPort1": managementPort1,
  "managementPort2": managementPort2,
  "managementPort3": managementPort3,
  "managementPort4": managementPort4,
  "managementPort5": managementPort5,
  "managementPort6": managementPort6
}
}
```

请求参数

参数	类型	描述	是否必须
nodeName	String	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。	否

parentName	String	父节点名称。 默认映射给根节点。	否
description	String	服务器的描述信息。 取值：1~50 位字符串。	否
ip	String	要添加的服务器 IP。	是
apiPort	Integer	指定管理 API 端口号。 取值：[1, 65535]，默认值为 1443。	否
iSCSIPort	Integer	指定 iSCSI 端口号。 取值：[1, 65535]，默认值为 3260。	否
portRange	String	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：[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。	否
diskPaths	Array of diskPath	数据目录属性集合。包括 path、capacityQuota，详见“表 1 请求参数 diskPath 说明”。	否
ports.dataPort1	Integer	数据端口 1。 取值：[1, 65535]。	否
ports.managementPort1	Integer	管理服务端口 1。 取值：[1, 65535]。	否
ports.managementPort2	Integer	管理服务端口 2。 取值：[1, 65535]。	否
ports.managementPort3	Integer	管理服务端口 3。	否

		取值：[1, 65535]。	
ports.managementPort4	Integer	管理服务端口 4。 取值：[1, 65535]。	否
ports.managementPort5	Integer	管理服务端口 5。 取值：[1, 65535]。	否
ports.managementPort6	Integer	管理服务端口 6。 取值：[1, 65535]。	否

表 1 请求参数 diskPath 说明

参数	类型	描述	是否必须
path	String	指定数据目录。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。 取值：只能包含字母、数字、汉字和特殊字符(~!@\$()_+ - ; . :)。	否
capacityQuota	Long	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：小于数据目录所在磁盘的总容量，单位是 bytes。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

请求示例

添加 IP 为 192.168.0.202 的服务器至集群，数据目录为/mnt/storage01 和 /mnt/storage02。

```
POST /rest/v1/system/server HTTP/1.1
Date: Mon, 7 Apr 2025 10:07:39 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 180
Host: 192.168.0.110:1443

{
  "ip": "192.168.0.202",
  "diskPaths": [
    {
      "path": "/mnt/storage01"
    },
    {
      "path": "/mnt/storage02"
    }
  ]
}
```

响应示例

```
HTTP/1.1 200 OK
x-hblock-request-id: 169b96ef760644759afb69470f903fc0
Connection: keep-alive
Date: Mon, 7 Apr 2025 10:07:39 GMT
Content-Length: 11
Content-Type: application/json; charset=utf-8
Server: HBlock

{
  "data": {}
}
```

4.11.2 修改指定服务器属性

此操作用来修改指定服务器属性。

注意：修改端口范围（portRange）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。

请求语法

```
PUT /rest/v1/system/server/serverId HTTP/1.1
Date:date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization:authorization

{
  "targetPortalIP": {
    "ips": [
      {
        "ip": ip,
        "port": port,
      }
    ],
    "status": status
  },
  "defaultPath": defaultPath,
  "portRange": port1-port2
}
```

请求参数

参数	类型	描述	是否必须
serverId	String	要修改的服务器 ID。	是
targetPortalIP.ip	Array of ip	iSCSI 目标门户 IP 属性集合，详见“表 1 请求参数 targetPortalIP.ip 说明”。 说明：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。 注意：指定目标门户时，建议仅指定一个服务器 ID。	否
targetPortalIP.status	String	是否启用 iSCSI 目标门户 IP。 取值： ● Enabled: 启用。 ● Disabled: 禁用。	是
defaultPath	String	设置默认的数据目录（仅单机版本支持）。 数据目录必须是已经添加到 HBlock 系统中，并且状态为 Normal 的数据目录。	否
portRange	String	指定端口范围，用于相关服务。 取值：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。	否

表 1 请求参数 targetPortalIP.ip 说明

参数	类型	描述	是否必须
ip	String	iSCSI 目标门户 IP。 说明：ip 和 port 必须同时存在，或者同时不存在。	否
port	Integer	iSCSI 目标门户端口号。 说明：ip 和 port 必须同时存在，或者同时不存在。 取值：[1, 65535]。	否

请求示例 1

单机版：修改服务器 hblock_1 的 iSCSI 目标门户 IP 和端口修改为：10.1.0.1 和 101，设置默认数据目录为：/mnt/storage01。

```
PUT /rest/v1/system/server/hblock_1 HTTP/1.1
Date: Fri, 29 Jul 2022 09:12:25 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 270
Host: 192.168.0.69:1443

{
  "targetPortalIP": {
    "ips": [
      {
        "ip": "10.1.0.1",
        "port": "101"
      }
    ],
    "status": "Enabled"
  },
  "defaultPath": "/mnt/storage01"
}
```

响应示例 1

```
HTTP/1.1 204 No Content
x-hblock-request-id: facb844a44034035bd9254c7d901cfac
Connection: keep-alive
Date: Fri, 29 Jul 2022 09:12:25 GMT
Server: HBlock
```

请求示例 2

集群版：修改服务器 `hblock_1` 的 iSCSI 目标门户 IP 和端口修改为：10.1.0.1 和 101。

```
PUT /rest/v1/system/server/hblock_1 HTTP/1.1
Date: Fri, 29 Jul 2022 09:00:33 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 221
Host: 192.168.0.121:1443

{
  "targetPortalIP": {
    "ips":[
      {
        "ip":"10.1.0.1",
        "port":"101"
      }
    ],
    "status":"Enabled"
  }
}
```

响应示例 2

```
HTTP/1.1 204 No Content
x-hblock-request-id: cbe33ebab9ad46df9fe691b71585e230
Connection: keep-alive
Date: Fri, 29 Jul 2022 09:00:33 GMT
```

Server: HBlock

4.11.3 批量修改服务器属性

此操作用来批量修改服务器属性。

说明：后续新增的数据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（portRange）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。
- 下列情况，修改 HBlock 可使用内存（maxMemoryRatio 或 maxMemorySize）后，必须重启对应服务器上的 HBlock 服务 cs、ps 才能生效；HBlock 可使用的内存由小于 8 GiB 调整为大于等于 8 GiB，或反之，还需重启对应服务器上的 HBlock 服务 ms：
 - 若修改指定服务器的 HBlock 可使用内存，重启该服务器的对应服务。
 - 若修改所有服务器（all）的 HBlock 可使用的内存，重启所有服务器上的对应服务。

请求语法

```
PUT /rest/v1/system/server HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "action": "setProperties",
  "serverIds": [serverId1, serverId2, ... ],
  "targetPortalIP": {
    "ips": [
      {
        "ip": ip,
```

```

        "port": port,
    }
],
    "status": status
},
    "defaultPath": defaultPath,
    "portRange": port1-port2,
    "maxMemoryRatio": maxMemoryRatio,
    "maxMemorySize": maxMemorySize
}

```

请求参数

参数	类型	描述	是否必须
action	String	操作动作。 取值： ● setProperties ：服务器配置。	是
serverIds	Array of string	指定需要设置的服务器。 取值： ● SERVER_ID ：需要修改的服务器 ID。 ● all ：指定修改所有 HBlock 服务器配置。 ● default ：指定后续加入集群节点的 HBlock 默认占用内存和端口范围。 默认修改所有服务器的配置。 说明：一次可用指定多个参数。	否
targetPortalIP.ips	Array of ip	iSCSI 目标门户 IP 属性集合，详见“ 表 1 请求参数 targetPortalIP.ip 说明 ”。 说明：若服务器与客户端不在同一网段（如	否

		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。 注意：指定目标门户时，建议仅指定一个服务器 ID。	
targetPortalIP.status	String	是否启用 iSCSI 目标门户 IP。 取值： ● Enabled: 启用。 ● Disabled: 禁用。	是
defaultPath	String	设置默认的数据目录（仅单机版本支持）。 数据目录必须是已经添加到 HBlock 系统中，并且状态为 Normal 的数据目录。	否
portRange	String	指定端口范围，用于相关服务。 取值：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。	否
maxMemoryRatio	Double	指定 HBlock 可使用的内存占用系统总内存的比例上限。 取值：取值范围是[0.01, 1]，支持小数点后 2 位。 说明： ● 对于集群版，HBlock 实际占用的内存与加入存储池的磁盘数量有关，可能会突破	否

		设置的比例上限，一旦超限将触发系统事件“内存不足（InsufficientMemory）”。 ● maxMemoryRatio 和 maxMemorySize 同时设置时，取二者的最小值。	
maxMemorySize	Long	指定 HBlock 可使用的内存最大值。 取值：整数形式，取值范围[2147483648, 9223372036854775807]，单位是 bytes。 说明： ● 对于集群版，HBlock 实际占用的内存与加入存储池的磁盘数量有关，可能会突破设置的比例上限，一旦超限将触发系统事件“内存不足（InsufficientMemory）”。 ● maxMemoryRatio 和 maxMemorySize 同时设置时，取二者的最小值。	否

表 1 请求参数 targetPortalIP.ip 说明

参数	类型	描述	是否必须
ip	String	iSCSI 目标门户 IP。 说明：ip 和 port 必须同时存在，或者同时不存在。	否
port	Integer	iSCSI 目标门户端口号。 说明：ip 和 port 必须同时存在，或者同时不存在。 取值：[1, 65535]。	否

请求示例 1

单机版：修改服务器的 iSCSI 目标门户 IP 和端口修改为：**10.1.0.1** 和 **101**，默认数据目录为：**/mnt/stor01**，端口范围为 19500-21500，HBlock 可使用的内存占用系统总内存的比例上限为 0.8。

```
PUT /rest/v1/system/server HTTP/1.1
Date: Fri, 09 Jan 2026 01:56:58 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Connection: keep-alive
Content-Length: 319
Host: 192.168.0.66:1443
```

```
{
  "action": "setProperties",
  "targetPortalIP": {
    "ips": [
      {
        "ip": "10.1.0.1",
        "port": "101"
      }
    ],
    "status": "Enabled"
  },
  "defaultPath": "/mnt/stor01",
  "portRange": "19500-21500",
  "maxMemoryRatio": 0.8
}
```

响应示例 1

```
HTTP/1.1 204 No Content
x-hblock-request-id: c1983e31153d4ad6b89ef4d7d3b2b2e3
Connection: keep-alive
Date: Fri, 09 Jan 2026 01:56:58 GMT
Content-Type: application/json; charset=utf-8
Server: HBlock
```

请求示例 2

集群版：修改服务器 hblock_1 的 iSCSI 目标门户 IP 和端口修改为：10.1.0.3 和 105，端口范围为 19500-22515，HBlock 可使用的内存占用系统总内存的比例上限为 0.95。修改后续加入集群节点的 HBlock 默认占用内存的比例上限为 0.95。

```
PUT /rest/v1/system/server HTTP/1.1
Date: Fri, 09 Jan 2026 03:09:48 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Connection: keep-alive
Content-Length: 353
Host: 192.168.0.65:1443
```

```
{
  "action": "setProperties",
  "serverIds": [
    "hblock_1",
    "default"
  ],
  "targetPortalIP": {
    "ips": [
      {
        "ip": "10.1.0.3",
        "port": "105"
      }
    ],
    "status": "Enabled"
  },
  "portRange": "19500-22515",
  "maxMemoryRatio": 0.95
}
```

响应示例 2

```
HTTP/1.1 204 No Content
x-hblock-request-id: 4c66e360cee64e4d8bdb33a479e224a0
Connection: keep-alive
Date: Fri, 09 Jan 2026 03:09:48 GMT
Content-Type: application/json; charset=utf-8
Server: HBlock
```

4.11.4 删除服务器属性

此操作用来删除服务器 targetPortalIP 配置。

请求语法

```
DELETE /rest/v1/system/server/serverId/targetportalIP HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

请求参数

参数	类型	描述	是否必须
serverId	String	要删除 targetPortalIP 配置的服务器 ID。	是

请求示例

删除 hblock_1 的 iSCSI 目标门户 IP。

```
DELETE /rest/v1/system/server/hblock_1/targetportalIP HTTP/1.1
Date: Thu, 17 Mar 2022 01:01:08 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.121:1443
```

响应示例

```
HTTP/1.1 204 No Content
x-hblock-request-id: 821c5921163f47d292aa281887c551f3
Connection: keep-alive
Date: Thu, 17 Mar 2022 01:01:09 GMT
Server: HBlock
```


4.11.5 迁移服务器上的基础服务（集群版适用）

此操作用来迁移 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**查询指定服务器**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获取。

请求语法

```
PUT /rest/v1/system/server/serverId/migrate HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "destServerId": serverId,
  "service": service,
```

```
"metaDir": path
}
```

请求参数

参数	类型	描述	是否必须
serverId	String	源 HBlock 服务器的 ID。	是
destServerId	String	目的服务器 ID。	是
service	String	需要迁移的基础服务。 取值： ● mdm：元数据管理服务。 ● ls：日志服务。 ● cs：协调服务。	是
metaDir	String	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： 只能包含字母、数字、汉字和特殊字符(~!@\$()_+-;.:)。默认目录为安装目录。	否

请求示例

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

```
PUT /rest/v1/system/server/hblock_1/migrate HTTP/1.1
Date: Thu, 4 Jul 2024 05:50:03 GMT
Content-Type: application/json
Authorization: HBlock userName:signature
Connection: keep-alive
Content-Length: 59
Host: 192.168.0.110:1443
```

```
{  
  "destServerId": "hblock_3",  
  "service": "mdm"  
}
```

响应示例

```
HTTP/1.1 202 Accepted  
x-hblock-request-id: 81b9f9c90114466695f1a114a524c191  
Content-Length: 0  
Date: Thu, 4 Jul 2024 05:50:03 GMT  
Connection: keep-alive  
Server: HBlock
```

4.11.6 移除服务器（集群版适用）

此操作用来从 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 deleteLocalData 为 true，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。
- 如果服务器上有基础服务，不允许移除。

请求语法

```
DELETE /rest/v1/system/server/serverId?force=force&deleteLocalData=deleteLocalData
HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

请求参数

参数	类型	描述	是否必须
serverId	String	要移除服务器的服务器 ID。	是
force	Boolean	是否强制移除服务器。 注意： 强制移除服务器，可能造成数据丢失。 取值： ● true：是。 ● false：否。 默认值为 false 。	否
deleteLocalData	Boolean	移除服务器时，是否删除服务器上 HBlock 数据目录中的数据。 取值： ● true：删除服务器上 HBlock 数据目录中的数据。 ● false：不删除服务器上 HBlock 数据目录中的数据。 默认值为 false 。	否

响应结果

名称	类型	描述
luns	Array of lun	被移除服务器上卷的属性信息集合，包含：lunName、iSCSITargets。详见“表 1 响应参数 lun 说明”。

表 1 响应参数 lun 说明

名称	类型	描述

lunName	String	被移除服务器上的卷名称。
iSCSITargets	Array of iSCSITarget	被移除服务器上卷对应的 iSCSI target 属性集合。包含：iqn、status、ips，详见“表 2 响应参数 iSCSITarget 说明”。

2 响应参数 iSCSITarget 说明

名称	类型	描述
iqn	String	被移除服务器上的卷对应的新 target IQN。
status	String	被移除服务器上的卷对应的 target 的状态： ● Active: 主 target。 ● Standby: 热备 target。 ● Offline: 离线 target。 ● Unavailable: 冷备 target。
ips	Array of ip	被移除服务器上卷对应 iSCSI target 的 IP 属性集合，详见“表 3 响应参数 ip 说明”。

表 3 响应参数 ip 说明

名称	类型	描述
ip	String	被移除服务器上的卷对应的新 iSCSI target 的 IP。
port	Integer	被移除服务器上的卷对应的新的 iSCSI target 的端口号。

请求示例

移除服务器 hblock_4。

DELETE /rest/v1/system/server/hblock_4?force=false&deleteLocalData=false HTTP/1.1 Date: Thu, 17 Mar 2022 02:16:39 GMT Authorization: HBlock <i>userName:signature</i> Host: 192.168.0.121:1443

响应示例

```
HTTP/1.1 202 Accepted
x-hblock-request-id: a1546d76c10a4758ae4b19f7fe140a02
Connection: keep-alive
Content-Length: 276
Date: Thu, 17 Mar 2022 02:16:42 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "luns": [
      {
        "lunName": "lun4",
        "iSCSITargets": [
          {
            "iqn": "iqn.2012-08.cn.ctyunapi.oos:target4.15",
            "status": "Offline",
            "ips": [
              {
                "ip": "192.168.0.72",
                "port": 3260
              }
            ]
          },
          {
            "iqn": "iqn.2012-08.cn.ctyunapi.oos:target4.16",
            "status": "Standby",
            "ips": [
              {
                "ip": "192.168.0.121",
                "port": 3260
              }
            ]
          }
        ]
      }
    ]
  }
}
```

```
}  
  }  
    ]  
      }  
        ]  
          }
```


4.11.7 重启服务器上的 HBlock 服务

此操作用来重启服务器上的 HBlock 服务。

请求语法

```
PUT /rest/v1/system/server/serverId/restart HTTP/1.1
```

Date: *date*

Content-Type: application/json; charset=utf-8

Content-Length: *length*

Host: *ip:port*

Authorization: *authorization*

```
{  
  "scope": [service1, service2,.....],  
  "force": force  
}
```

请求参数

参数	类型	描述	是否必须
serverId	String	要重启的服务器 ID。	是
scope	Array of string	重启的 HBlock 服务范围。一次可以指定多个服务，以英文逗号隔开。 取值： ● All: 重启服务器上的所有服务。 ● Min: 仅重启服务器上的 ms 和 ps 服务。 ● mdm: 元数据管理服务（仅集群版支持）。 ● fc: 故障转移控制服务（仅集群版支持）。 ● ls: 日志服务（仅集群版支持）。 ● ds: 数据服务（仅集群版支持）。	否

		● ds-x: 磁盘的数据服务（仅集群版支持）。 ● cs: 协调服务（仅集群版支持）。 ● ms: 管理服务。 ● ws: 监控服务。 ● ps: 协议解析服务。 ● ag: 数据采集服务。 ● ua: 升级监听服务。 默认值为 Min。 说明: 各服务具体作用详见 HBlock 服务。	
force	Boolean	是否强制重启。 注意: 强制重启服务器，可能造成数据丢失。 取值: ● true: 强制重启。 ● false: 不强制重启。 默认值为 false。	否

请求示例

重启服务器 hblock_3 上的 HBlock ag、ua 服务。

```
PUT /rest/v1/system/server/hblock_3/restart HTTP/1.1
```

```
Date: Wed, 07 Jan 2026 02:11:33 GMT
```

```
Content-Length: 56
```

```
Authorization: HBlock userName:signature
```

```
Host: 192.168.0.65:1443
```

```
{
  "scope": [
    "ag",
    "ua"
  ]
}
```

响应示例

```
HTTP/1.1 202 Accepted
x-hblock-request-id: 900aef1278ed40429f204f7ff6d052a8
Connection: keep-alive
Date: Wed, 07 Jan 2026 02:11:33 GMT
Content-Length: 0
Server: HBlock
```

4.11.8 批量查询服务器

此操作用来批量查询服务器。

请求语法

```
GET /rest/v1/system/server?filter=filter&serverId=serverId&detail=Boolean&range=i-j HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

请求参数

参数	类型	描述	是否必须
filter	String	查询的过滤条件。 过滤条件由查询属性和属性值组成，其中属性和属性值之间用单个冒号分隔（key:value）表示模糊匹配，用两个冒号分隔（key::value）表示精确匹配。可以选择多个查询，如果是或的关系，使用“or”将查询条件分隔开；如果是与的关系，使用“and”将查询条件分隔开。 支持的查询条件为： ● serverName: 服务器名称。 ● status: 服务器状态（Connected: 已连接；Disconnected: 未连接；Removing: 移除中）。	否
serverId	String	服务器 ID。 取值： ● 具体服务器 ID: 查询指定的服务器信息。 ● all: 输出所有的服务器信息。 ● default: 查询集群中后续加入节点的 HBlock 默认参	否

		数。 默认值为 all。 说明：可以同时指定多个参数，以逗号隔开，如 serverId=hblock_1,hblock_2,default。	
detail	Boolean	是否展示服务器的详细信息。 取值： ● true：展示服务器详细信息。 ● false：展示服务器基本信息。 默认值为 false。	否
range	String	查询服务器的范围，格式为 i-j，i 和 j 为正整数，$0 < j - i < 1000$。按照服务器 ID 进行排序，返回第 i 到第 j 的服务器。 如果输入的查询范围超过服务器的总个数，返回的结果为空，如果未指定 range，则返回所有符合条件的服务器。	否

响应结果

名称	类型	描述
servers	Array of server	服务器 HBlock 属性集合，详见“ 表 1 响应参数 server 说明 ”。
default	List of default	集群中后续加入节点的 HBlock 默认参数集合，详见“ 表 7 响应参数 default 说明 ”。

表 1 响应参数 server 说明

名称	类型	描述
serverName	String	服务器名称。
serverId	String	服务器 ID。
status	String	服务器状态：

		● Connected: 已连接。 ● Disconnected: 未连接。 ● Removing: 移除中。
version	String	服务器上的 HBlock 版本。
isMasterServer	Boolean	该服务器是否为 Master 节点（仅集群版支持）： ● true: 是。 ● false: 否。
isBaseServer	Boolean	该服务器是否为基础节点（仅集群版支持）： ● true: 是。 ● false: 否。
publicAddress.ip	String	业务网的 IP。
publicAddress.port	Integer	业务网端口号。
clusterAddress.ip	String	集群网 IP 地址。
clusterAddress.port	Integer	集群网端口号。
allDiskPaths.usedCapacity	Long	当前服务器所有 HBlock 数据目录对应分区的已用容量，单位是 bytes。 说明： 查询参数 detail 取值为 false 时展示。
allDiskPaths.totalCapacity	Long	当前服务器所有 HBlock 数据目录对应分区的总容量，单位是 bytes。 说明： 查询参数 detail 取值为 false 时展示。
targetPortalIP ips	Array of targetPortalIP.ip	iSCSI 目标门户 IP 属性集合，详见“ 表 2 响应参数 targetPortalIP.ip 说明 ”。 说明： 查询参数 detail 取值为 true 时展示。
targetPortalIP.status	String	iSCSI 目标门户 IP 状态： ● Enabled: 启用状态。 ● Disabled: 停用状态。 说明： 查询参数 detail 取值为 true 时展示。

diskPaths	Array of diskPath	HBlock 数据目录信息集合，详见“表 3 响应参数 diskPath 说明”。 说明：查询参数 detail 取值为 true 时展示。
defaultPath	String	默认的数据目录（仅单机版支持）。
recentStartTime	Long	HBlock 服务在该节点上最近一次成功启动的时间。 服务器未连接时，不返回此项。
ports	Array of port	HBlock 端口的集合，详见“表 4 响应参数 port 说明”。
portRange	String	端口范围。
iSCSIPort	Integer	iSCSI 端口号。
storagePorts	Array of storagePort	存储服务端口集合（仅集群版支持），详见“表 5 响应参数 storagePort 说明”。
nodeName	String	节点名称（仅集群版支持）。
parentName	String	父节点名称（仅集群版支持）。
description	String	节点描述（仅集群版支持）。
baseServices	Array of baseService	基础服务属性的集合（仅集群版基础服务器支持），详见“表 6 响应参数 baseService 说明”。
maxMemoryRatio	Double	HBlock 可使用的内存占用系统总内存的比例上限。
maxMemorySize	Long	HBlock 可使用的内存的最大值，单位是 bytes。

表 2 响应参数 **targetPortalIP.ip** 说明

名称	类型	描述
ip	String	iSCSI 目标门户 IP。
port	Integer	iSCSI 目标门户端口号。

表 3 响应参数 diskPath 说明

名称	类型	描述
path	String	HBlock 数据目录。
storagePool	String	数据目录所属存储池（仅集群版支持）。
totalCapacity	Long	当前 HBlock 数据目录对应分区的总容量，单位是 bytes。
usedCapacity	Long	服务器上 HBlock 数据目录对应分区的已用容量，单位是 bytes。
usedCapacityQuota	Long	HBlock 在该数据目录的已写入量，即已用容量配额。单位是 bytes。
capacityQuota	Long	数据目录的容量配额。单位是 bytes。 0 表示禁止写入。不输出，表示没有限制。
managementStatus	String	数据目录管理状态： - Added: 数据目录已添加至系统中。 - Removing: 数据目录正在移除中。
subStatus.stage	String	移除数据目录所处的阶段： - Reconfiguration: 重置中。 - CheckingData: 检查数据。在该阶段，会同时显示下列字段的信息 totalFD、healthyFD、warningFD、errorFD、safeData、reconstructionWaitData、availFDWaitData、singleCopyLUNData、corruptedData、具体告警信息（availFDWaitWarning、singleCopyLUNWarning 或者 corruptedWarning）。

		● Executing: 执行移除数据目录
subStatus.totalFD	Integer	总的故障域个数。
subStatus.healthyFD	Integer	状态为 Healthy 的故障域个数。
subStatus.warningFD	Integer	状态为 Warning 的故障域个数。
subStatus.errorFD	Integer	状态为 Error 的故障域个数。
subStatus.safeData	Double	安全数据百分比，单位是%。
subStatus.reconstructionWaitData	Double	需要重建的数据百分比，单位是%。
subStatus.availFDWaitData	Double	移除磁盘或者服务器时，数据重建需要的故障域个数不足，因此引发的无法重建的数据百分比。
subStatus.singleCopyLUNData	Double	单副本数据百分比，单位是%。
subStatus.corruptedData	Double	已经损坏的数据百分比，单位是%。
subStatus.availFDWaitWarning	String	告警信息。 根据状态不同，显示的告警信息不同： ● availFDWaitWarning: 移除磁盘或者服务器时，数据重建需要的故障域个数不足。 ● singleCopyLUNWarning: 单副本卷数据所在磁盘或者服务器被移除。 ● corruptedWarning: 故障域异常导致数据损坏。
subStatus.singleCopyLUNWarning	String	
subStatus.corruptedWarning	String	
healthStatus	String	数据目录健康状态： ● 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；HBlock 对

		这个目录停写；数据目录配额使用率超过阈值（系统默认值为 95%）；数据目录配额为 0。 ● 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。
healthDetail	Array of string	数据目录健康状态详情： ● 如果健康状态为 Healthy，此字段为空。 ● 如果健康状态为 Warning 或 Error，显示警告或错误的详细信息。
dsId	String	数据目录对应的 ds 进程（仅集群版支持）。 说明：数据目录加入存储池后才会展示 ds 进程。

表 4 响应参数 port 说明

名称	类型	描述
portName	String	HBlock 端口名称。
port	Integer	HBlock 在当前服务器上使用的端口号。

表 5 响应参数 storagePort 说明

名称	类型	描述
portN	Integer	存储服务端口。
diskpath	String	存储服务的数据目录。
pathId	Integer	数据目录 ID。

表 6 响应参数 baseService 说明

名称	类型	描述
name	String	基础服务名称： ● mdm：元数据管理服务。

		● ls: 日志服务。 ● cs: 协调服务。
status	String	基础服务的状态: ● Up: 可用。 ● Down: 不可用。 ● Migrating: 正在迁移。 ● Unkown: 未知。
metaDir	String	存储基础服务数据的数据目录。

表 7 响应参数 **default** 说明

名称	类型	描述
portRange	String	后续加入集群节点的端口范围。
maxMemoryRatio	Double	后续加入集群节点的 HBlock 可使用的内存占用系统总内存的比例上限。
maxMemorySize	Long	后续加入集群节点的 HBlock 可使用的内存的最大值，单位是 bytes。

请求示例 1

单机版：查询服务器的详细信息。

```
GET /rest/v1/system/server?detail=true HTTP/1.1
Date: Fri, 09 Jan 2026 07:55:19 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.66:1443
```

响应示例 1

```
HTTP/1.1 200 OK
Date: Fri, 09 Jan 2026 07:55:19 GMT
x-hblock-request-id: a779ec83337e4fee814f10cc2955e976
Connection: keep-alive
```

```
Content-Length: 941
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "servers": [
      {
        "serverName": "songt-005",
        "serverId": "hblock_1",
        "status": "Connected",
        "version": "4.0.0",
        "publicAddress": {
          "ip": "192.168.0.66",
          "port": 3260
        },
        "clusterAddress": {
          "ip": "192.168.0.66",
          "port": 20001
        },
        "targetPortalIP": {
          "ips": [
            {
              "ip": "10.1.0.1",
              "port": 101
            }
          ],
          "status": "Disabled"
        },
        "diskPaths": [
          {
            "path": "/mnt/stor01",
            "usedCapacity": 2528911360,
            "totalCapacity": 100169342976,
            "usedCapacityQuota": 1097037922,
```

```
        "managementStatus": "Added",
        "healthStatus": "Healthy",
        "healthDetail": []
    }
],
"defaultPath": "/mnt/stor01",
"recentStartTime": 1767919098886,
"ports": [
    {
        "portName": "apiPort",
        "port": 1443
    },
    {
        "portName": "webPort",
        "port": 2443
    },
    {
        "portName": "managementPort1",
        "port": 20001
    },
    {
        "portName": "managementPort2",
        "port": 20003
    },
    {
        "portName": "managementPort3",
        "port": 20000
    },
    {
        "portName": "managementPort4",
        "port": 20004
    },
    {
        "portName": "managementPort6",
        "port": 20002
    }
]
```

```
        }
      ],
      "portRange": "19500-21500",
      "iSCSIPort": 3260,
      "maxMemoryRatio": 0.8,
      "maxMemorySize": 9223372036854775807
    }
  ]
}
```

请求示例 2

集群版：查询第 2 和第 3 个服务器。

```
GET /rest/v1/system/server?range=2-3 HTTP/1.1
Date: Sun, 04 Jan 2026 05:39:36 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.65:1443
```

响应示例 2

```
HTTP/1.1 200 OK
Date: Sun, 04 Jan 2026 05:39:36 GMT
x-hblock-request-id: 35482534f144466fb71a88c585796879
Connection: keep-alive
Content-Length: 3302
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "servers": [
      {
        "serverName": "k8s-master",
        "serverId": "hblock_2",
        "status": "Connected",
```

```
"version": "4.0.0",
"isMasterServer": false,
"isBaseServer": true,
"publicAddress": {
  "ip": "192.168.0.64",
  "port": 3260
},
"clusterAddress": {
  "ip": "192.168.0.64",
  "port": 20015
},
"allDiskPaths": {
  "usedCapacity": 19339370496,
  "totalCapacity": 142631432192
},
"recentStartTime": 1767489601645,
"ports": [
  {
    "portName": "apiPort",
    "port": 1443
  },
  {
    "portName": "webPort",
    "port": 2443
  },
  {
    "portName": "managementPort1",
    "port": 20015
  },
  {
    "portName": "managementPort2",
    "port": 20013
  },
  {
    "portName": "managementPort3",
```

```
        "port": 20016
      },
      {
        "portName": "managementPort4",
        "port": 20000
      },
      {
        "portName": "managementPort5",
        "port": 20003
      },
      {
        "portName": "managementPort6",
        "port": 20010
      },
      {
        "portName": "managementPort7",
        "port": 20014
      },
      {
        "portName": "dataPort1",
        "port": 20007
      },
      {
        "portName": "metadataPort1",
        "port": 20011
      },
      {
        "portName": "metadataPort2",
        "port": 20004
      },
      {
        "portName": "metadataPort3",
        "port": 20005
      },
      {
```



```
        "portName": "metadataPort4",
        "port": 20009
    },
    {
        "portName": "metadataPort5",
        "port": 20006
    },
    {
        "portName": "metadataPort6",
        "port": 20008
    },
    {
        "portName": "metadataPort7",
        "port": 20002
    },
    {
        "portName": "metadataPort8",
        "port": 20001
    },
    {
        "portName": "metadataPort9",
        "port": 20012
    }
],
"portRange": "20000-20500",
"iSCSIPort": 3260,
"storagePorts": [
    {
        "port1": 20017,
        "port2": 20019,
        "port3": 20018,
        "diskpath": "/mnt/stor02",
        "pathId": "1"
    },
    {
```

```
        "port1": 20020,
        "port2": 20022,
        "port3": 20021,
        "diskpath": "/mnt/stor01",
        "pathId": "2"
    }
],
"nodeName": "default:hblock_2",
"parentName": "default",
"baseServices": [
    {
        "name": "mdm",
        "status": "Up",
        "metaDir": "/mnt/storage01/CTYUN_HBlock_Plus_4.0.0_x64"
    },
    {
        "name": "ls",
        "status": "Up",
        "metaDir": "/mnt/storage01/CTYUN_HBlock_Plus_4.0.0_x64"
    },
    {
        "name": "cs",
        "status": "Up",
        "metaDir": "/mnt/storage01/CTYUN_HBlock_Plus_4.0.0_x64"
    }
],
"maxMemoryRatio": 0.9,
"maxMemorySize": 9223372036854775807
},
{
    "serverName": "songt-0006",
    "serverId": "hblock_3",
    "status": "Connected",
    "version": "4.0.0",
    "isMasterServer": false,
```

```
"isBaseServer": true,
"publicAddress": {
  "ip": "192.168.0.67",
  "port": 3260
},
"clusterAddress": {
  "ip": "192.168.0.67",
  "port": 20015
},
"allDiskPaths": {
  "usedCapacity": 1816420352,
  "totalCapacity": 199407550464
},
"recentStartTime": 1767489617411,
"ports": [
  {
    "portName": "apiPort",
    "port": 1443
  },
  {
    "portName": "webPort",
    "port": 2443
  },
  {
    "portName": "managementPort1",
    "port": 20015
  },
  {
    "portName": "managementPort2",
    "port": 20013
  },
  {
    "portName": "managementPort3",
    "port": 20016
  },
]
```

```
{
  "portName": "managementPort4",
  "port": 20000
},
{
  "portName": "managementPort5",
  "port": 20003
},
{
  "portName": "managementPort6",
  "port": 20010
},
{
  "portName": "managementPort7",
  "port": 20014
},
{
  "portName": "dataPort1",
  "port": 20007
},
{
  "portName": "metadataPort4",
  "port": 20009
},
{
  "portName": "metadataPort5",
  "port": 20006
},
{
  "portName": "metadataPort6",
  "port": 20008
},
{
  "portName": "metadataPort7",
  "port": 20002
}
```

```
    },
    {
      "portName": "metadataPort8",
      "port": 20001
    },
    {
      "portName": "metadataPort9",
      "port": 20012
    }
  ],
  "portRange": "20000-20500",
  "iSCSIPort": 3260,
  "storagePorts": [
    {
      "port1": 20004,
      "port2": 20011,
      "port3": 20005,
      "diskpath": "/mnt/stor01",
      "pathId": "1"
    },
    {
      "port1": 20017,
      "port2": 20019,
      "port3": 20018,
      "diskpath": "/mnt/stor02",
      "pathId": "2"
    }
  ],
  "nodeName": "default:hblock_3",
  "parentName": "default",
  "baseServices": [
    {
      "name": "ls",
      "status": "Up",
      "metaDir": "/mnt/storage01/CTYUN_HBlock_Plus_4.0.0_x64"
```

```
    },
    {
      "name": "cs",
      "status": "Up",
      "metaDir": "/mnt/storage01/CTYUN_HBlock_Plus_4.0.0_x64"
    }
  ],
  "maxMemoryRatio": 0.9,
  "maxMemorySize": 9223372036854775807
}
]
```

请求示例 3

集群版：查询 hblock_1、hblock_2 服务器详细信息及后续加入节点的默认信息。

```
GET /rest/v1/system/server?serverId=hblock_1,hblock_2,default&detail=true HTTP/1.1
Date: Thu, 05 Mar 2026 01:21:11 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.65:1443
```

响应示例 3

```
HTTP/1.1 200 OK
Date: Thu, 05 Mar 2026 01:21:11 GMT
x-hblock-request-id: 0894b4c100524575ad220028f046d8ce
Connection: keep-alive
Content-Length: 3836
Content-Type: application/json;charset=utf-8
Server: HBlock

{"data": {
  "servers": [
    {
```

```
"serverName": "songt-0004",
"serverId": "hblock_1",
"status": "Connected",
"version": "4.0.0",
"isMasterServer": true,
"isBaseServer": true,
"publicAddress": {
  "ip": "192.168.0.65",
  "port": 3260
},
"clusterAddress": {
  "ip": "192.168.0.65",
  "port": 20005
},
"targetPortalIP": {
  "ips": [
    {
      "ip": "10.0.0.3",
      "port": 3261
    }
  ],
  "status": "Enabled"
},
"diskPaths": [
  {
    "path": "/mnt/stor01",
    "storagePool": "default",
    "usedCapacity": 43635916800,
    "totalCapacity": 99703775232,
    "usedCapacityQuota": 43619094956,
    "managementStatus": "Added",
    "healthStatus": "Healthy",
    "healthDetail": [],
    "dsId": "ds-1"
  }
]
```

```
],  
"recentStartTime": 1772608577246,  
"ports": [  
  {  
    "portName": "apiPort",  
    "port": 1443  
  },  
  {  
    "portName": "webPort",  
    "port": 2443  
  },  
  {  
    "portName": "managementPort1",  
    "port": 20005  
  },  
  {  
    "portName": "managementPort2",  
    "port": 20013  
  },  
  {  
    "portName": "managementPort3",  
    "port": 20002  
  },  
  {  
    "portName": "managementPort4",  
    "port": 20016  
  },  
  {  
    "portName": "managementPort5",  
    "port": 20003  
  },  
  {  
    "portName": "managementPort6",  
    "port": 20006  
  },  
]
```



```
{
  "portName": "managementPort7",
  "port": 20007
},
{
  "portName": "dataPort1",
  "port": 20014
},
{
  "portName": "metadataPort1",
  "port": 20008
},
{
  "portName": "metadataPort2",
  "port": 20004
},
{
  "portName": "metadataPort3",
  "port": 20011
},
{
  "portName": "metadataPort4",
  "port": 20012
},
{
  "portName": "metadataPort5",
  "port": 20000
},
{
  "portName": "metadataPort6",
  "port": 20015
},
{
  "portName": "metadataPort7",
  "port": 20009
}
```

```
    },
    {
      "portName": "metadataPort8",
      "port": 20001
    },
    {
      "portName": "metadataPort9",
      "port": 20010
    }
  ],
  "portRange": "20000-20500",
  "iSCSIPort": 3260,
  "storagePorts": [
    {
      "port1": 20017,
      "port2": 20019,
      "port3": 20018,
      "diskpath": "/mnt/stor01",
      "pathId": "1"
    }
  ],
  "nodeName": "default:hblock_1",
  "parentName": "default",
  "baseServices": [
    {
      "name": "mdm",
      "status": "Up",
      "metaDir": "/mnt/storage01/CTYUN_HBlock_Plus_4.0.0_x64"
    },
    {
      "name": "ls",
      "status": "Up",
      "metaDir": "/mnt/storage01/CTYUN_HBlock_Plus_4.0.0_x64"
    }
  ]
}
```

```
        "name": "cs",
        "status": "Up",
        "metaDir": "/mnt/storage01/CTYUN_HBlock_Plus_4.0.0_x64"
    }
],
"maxMemoryRatio": 0.95,
"maxMemorySize": 9223372036854775807
},
{
    "serverName": "k8s-master",
    "serverId": "hblock_2",
    "status": "Connected",
    "version": "4.0.0",
    "isMasterServer": false,
    "isBaseServer": true,
    "publicAddress": {
        "ip": "192.168.0.64",
        "port": 3260
    },
    "clusterAddress": {
        "ip": "192.168.0.64",
        "port": 20005
    },
    "diskPaths": [
        {
            "path": "/mnt/stor02",
            "storagePool": "default",
            "usedCapacity": 43636088832,
            "totalCapacity": 99703775232,
            "usedCapacityQuota": 43636043776,
            "managementStatus": "Added",
            "healthStatus": "Healthy",
            "healthDetail": [],
            "dsId": "ds-1"
        }
    ]
}
```

```
],  
"recentStartTime": 1772614410202,  
"ports": [  
  {  
    "portName": "apiPort",  
    "port": 1443  
  },  
  {  
    "portName": "webPort",  
    "port": 2443  
  },  
  {  
    "portName": "managementPort1",  
    "port": 20005  
  },  
  {  
    "portName": "managementPort2",  
    "port": 20013  
  },  
  {  
    "portName": "managementPort3",  
    "port": 20002  
  },  
  {  
    "portName": "managementPort4",  
    "port": 20016  
  },  
  {  
    "portName": "managementPort5",  
    "port": 20003  
  },  
  {  
    "portName": "managementPort6",  
    "port": 20006  
  },  
]
```

```
{
  "portName": "managementPort7",
  "port": 20007
},
{
  "portName": "dataPort1",
  "port": 20014
},
{
  "portName": "metadataPort1",
  "port": 20008
},
{
  "portName": "metadataPort2",
  "port": 20004
},
{
  "portName": "metadataPort3",
  "port": 20011
},
{
  "portName": "metadataPort4",
  "port": 20012
},
{
  "portName": "metadataPort5",
  "port": 20000
},
{
  "portName": "metadataPort6",
  "port": 20015
},
{
  "portName": "metadataPort7",
  "port": 20009
}
```

```
    },
    {
      "portName": "metadataPort8",
      "port": 20001
    },
    {
      "portName": "metadataPort9",
      "port": 20010
    }
  ],
  "portRange": "20000-20500",
  "iSCSIPort": 3260,
  "storagePorts": [
    {
      "port1": 20017,
      "port2": 20019,
      "port3": 20018,
      "diskpath": "/mnt/stor02",
      "pathId": "1"
    }
  ],
  "nodeName": "default:hblock_2",
  "parentName": "default",
  "baseServices": [
    {
      "name": "mdm",
      "status": "Up",
      "metaDir": "/mnt/storage01/CTYUN_HBlock_Plus_4.0.0_x64"
    },
    {
      "name": "ls",
      "status": "Up",
      "metaDir": "/mnt/storage01/CTYUN_HBlock_Plus_4.0.0_x64"
    }
  ]
}
```

```
        "name": "cs",
        "status": "Up",
        "metaDir": "/mnt/storage01/CTYUN_HBlock_Plus_4.0.0_x64"
      }
    ],
    "maxMemoryRatio": 0.95,
    "maxMemorySize": 9223372036854775807
  }
],
"default": {
  "portRange": "20000-20500",
  "maxMemoryRatio": 0.9,
  "maxMemorySize": 9223372036854775807
}
}
```

4.11.9 查询指定服务器

此操作用来查询指定服务器。

请求语法

```
GET /rest/v1/system/server/serverId HTTP/1.1
```

Date: *date*

Host: *ip:port*

Authorization: *authorization*

请求参数

参数	类型	描述	是否必须
serverId	String	要查询的服务器 ID。 如果未指定服务器 ID，则查询所有服务器。	否

响应结果

名称	类型	描述
serverName	String	服务器名称。
serverId	String	服务器 ID。
status	String	服务器状态： ● Connected: 已连接。 ● Disconnected: 未连接。 ● Removing: 移除中。
version	String	服务器上的 HBlock 版本。
isMasterServer	Boolean	该服务器是否为 Master 节点（仅集群版支持）： ● true: 是。 ● false: 否。

isBaseServer	Boolean	该服务器是否为基础节点（仅集群版支持）： ● true: 是。 ● false: 否。
publicAddress.ip	String	业务网的 IP。
publicAddress.port	Integer	业务网端口号。
clusterAddress.ip	String	集群网 IP 地址。
clusterAddress.port	Integer	集群网端口号。
targetPortalIP.ips	Array of targetPortalIP.ip	iSCSI 目标门户 IP 属性集合，详见“表 1 响应参数 targetPortalIP.ip 说明”。
targetPortalIP.status	String	iSCSI 目标门户 IP 状态： ● Enabled: 启用状态。 ● Disabled: 停用状态。
diskPaths	Array of diskPath	HBlock 数据目录信息集合，详见“表 2 响应参数 diskPath 说明”。
defaultPath	String	默认的数据目录（仅单机版支持）。
recentStartTime	Long	HBlock 服务在该节点上最近一次成功启动的时间。 服务器未连接时，不返回此项。
ports	Array of port	HBlock 端口的集合，详见“表 3 响应参数 port 说明”。
portRange	String	端口范围。
iSCSIPort	Integer	iSCSI 端口号。
storagePorts	Array of storagePort	存储服务端口集合（仅集群版支持），详见“表 4 响应参数 storagePort 说明”。
nodeName	String	节点名称（仅集群版支持）。
parentName	String	父节点名称（仅集群版支持）。
description	String	节点描述（仅集群版支持）。

baseServices	Array of baseService	基础服务属性的集合（仅集群版基础服务器支持），详见“表 5 响应参数 baseService 说明”。
maxMemoryRatio	Double	HBlock 可使用的内存占用系统总内存的比例上限。
maxMemorySize	Long	HBlock 可使用的内存的最大值，单位是 bytes。

表 1 响应参数 targetPortalIP.ip 说明

名称	类型	描述
ip	String	iSCSI 目标门户 IP。
port	Integer	iSCSI 目标门户端口号。

表 2 响应参数 diskPath 说明

名称	类型	描述
path	String	HBlock 数据目录。
storagePool	String	数据目录所属存储池（仅集群版支持）。
totalCapacity	Long	当前 HBlock 数据目录对应分区的总容量，单位是 bytes。
usedCapacity	Long	服务器上 HBlock 数据目录对应分区的已用容量，单位是 bytes。
usedCapacityQuota	Long	HBlock 在该数据目录的已写入量，即已用容量配额。单位是 bytes。
capacityQuota	Long	数据目录的容量配额。单位是 bytes。 0 表示禁止写入。不输出，表示没有限制。
managementStatus	String	数据目录管理状态： - Added: 数据目录已添加至系统中。 - Removing: 数据目录正在移除中。
subStatus.stage	String	移除数据目录所处的阶段：

		● Reconfiguration: 重置中。 ● CheckingData: 检查数据。在该阶段，会同时显示下列字段的信息 totalFD、healthyFD、warningFD、errorFD、safeData、reconstructionWaitData、availFDWaitData、singleCopyLUNData、corruptedData、具体告警信息（availFDWaitWarning、singleCopyLUNWarning 或者 corruptedWarning）。 ● Executing: 执行移除数据目录
subStatus.totalFD	Integer	总的故障域个数。
subStatus.healthyFD	Integer	状态为 Healthy 的故障域个数。
subStatus.warningFD	Integer	状态为 Warning 的故障域个数。
subStatus.errorFD	Integer	状态为 Error 的故障域个数。
subStatus.safeData	Double	安全数据百分比，单位是%。
subStatus.reconstructionWaitData	Double	需要重建的数据百分比，单位是%。
subStatus.availFDWaitData	Double	移除磁盘或者服务器时，数据重建需要的故障域个数不足，因此引发的无法重建的数据百分比。
subStatus.singleCopyLUNData	Double	单副本数据百分比，单位是%。
subStatus.corruptedData	Double	已经损坏的数据百分比，单位是%。
subStatus.availFDWaitWarning	String	告警信息。
subStatus.singleCopyLUNWarning	String	根据状态不同，显示的告警信息不同：
subStatus.corruptedWarning	String	● availFDWaitWarning: 移除磁盘或者服务器时，数据重建需要的故障域个数不足。 ● singleCopyLUNWarning: 单副本卷数据所在磁盘或者服务器被移除。

		● corruptedWarning: 故障域异常导致数据损坏。
healthStatus	String	数据目录健康状态: ● 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; HBlock 对这个目录停写; 数据目录配额使用率超过阈值 (系统默认值为 95%); 数据目录配额为 0。 ● 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。
healthDetail	Array of string	数据目录健康状态详情: ● 如果健康状态为 Healthy, 此字段为空。 ● 如果健康状态为 Warning 或 Error, 显示警告或错误的详细信息。
dsId	String	数据目录对应的 ds 进程 (仅集群版支持)。 说明: 数据目录加入存储池后才会展示 ds 进程。

表 3 响应参数 port 说明

名称	类型	描述
portName	String	HBlock 端口名称。
port	Integer	HBlock 在当前服务器上使用的端口号。

表 4 响应参数 storagePort 说明

名称	类型	描述
portN	Integer	存储服务端口。
diskpath	String	存储服务的数据目录。
pathId	Integer	数据目录 ID。

表 5 响应参数 baseService 说明

名称	类型	描述
name	String	基础服务名称： ● mdm: 元数据管理服务。 ● ls: 日志服务。 ● cs: 协调服务。
status	String	基础服务的状态： ● Up: 可用。 ● Down: 不可用。 ● Migrating: 正在迁移。 ● Unkown: 未知。
metaDir	String	存储基础服务数据的数据目录。

请求示例 1

单机版：查询服务器 ID 为 hblock_1 的信息。

```
GET /rest/v1/system/server/hblock_1 HTTP/1.1
Date: Sun, 04 Jan 2026 03:32:34 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.66:1443
```

响应示例 1

```
HTTP/1.1 200 OK
```

x-hblock-request-id: 5a8e4a8d99f742379829435b56692a09

Connection: keep-alive

Content-Length: 851

Date: Sun, 04 Jan 2026 03:32:34 GMT

Content-Type: application/json; charset=utf-8

Server: HBlock

```
{
  "data": {
    "serverName": "songt-005",
    "serverId": "hblock_1",
    "status": "Connected",
    "version": "4.0.0",
    "publicAddress": {
      "ip": "192.168.0.66",
      "port": 3260
    },
    "clusterAddress": {
      "ip": "192.168.0.66",
      "port": 20001
    },
    "diskPaths": [
      {
        "path": "/mnt/stor01",
        "usedCapacity": 2528108544,
        "totalCapacity": 100169342976,
        "usedCapacityQuota": 1096309995,
        "managementStatus": "Added",
        "healthStatus": "Healthy",
        "healthDetail": []
      }
    ],
    "defaultPath": "/mnt/stor01",
    "recentStartTime": 1767492674311,
    "ports": [
```

```
{
  {
    "portName": "apiPort",
    "port": 1443
  },
  {
    "portName": "webPort",
    "port": 2443
  },
  {
    "portName": "managementPort1",
    "port": 20001
  },
  {
    "portName": "managementPort2",
    "port": 20003
  },
  {
    "portName": "managementPort3",
    "port": 20000
  },
  {
    "portName": "managementPort4",
    "port": 20004
  },
  {
    "portName": "managementPort6",
    "port": 20002
  }
],
"portRange": "20000-20500",
"iSCSIPort": 3260,
"maxMemoryRatio": 0.9,
"maxMemorySize": 9223372036854775807
}
```

请求示例 2

集群版：查询服务器 ID 为 hblock_1 的信息。

```
GET /rest/v1/system/server/hblock_1 HTTP/1.1
Date: Thu, 05 Mar 2026 01:51:44 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.65:1443
```

响应示例 2

```
HTTP/1.1 200 OK
x-hblock-request-id: 942ae9373c644438b176b3b8a2376134
Connection: keep-alive
Content-Length: 1905
Date: Thu, 05 Mar 2026 01:51:44 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "serverName": "songt-0004",
    "serverId": "hblock_1",
    "status": "Connected",
    "version": "4.0.0",
    "isMasterServer": true,
    "isBaseServer": true,
    "publicAddress": {
      "ip": "192.168.0.65",
      "port": 3260
    },
    "clusterAddress": {
      "ip": "192.168.0.65",
      "port": 20005
    },
    "targetPortalIP": {
```



```
"ips": [  
  {  
    "ip": "10.0.0.3",  
    "port": 3261  
  }  
],  
"status": "Enabled"  
},  
"diskPaths": [  
  {  
    "path": "/mnt/stor01",  
    "storagePool": "default",  
    "usedCapacity": 43635916800,  
    "totalCapacity": 99703775232,  
    "usedCapacityQuota": 43627483478,  
    "managementStatus": "Added",  
    "healthStatus": "Healthy",  
    "healthDetail": [],  
    "dsId": "ds-1"  
  }  
],  
"recentStartTime": 1772608577246,  
"ports": [  
  {  
    "portName": "apiPort",  
    "port": 1443  
  },  
  {  
    "portName": "webPort",  
    "port": 2443  
  },  
  {  
    "portName": "managementPort1",  
    "port": 20005  
  },  
]
```

```
{
  "portName": "managementPort2",
  "port": 20013
},
{
  "portName": "managementPort3",
  "port": 20002
},
{
  "portName": "managementPort4",
  "port": 20016
},
{
  "portName": "managementPort5",
  "port": 20003
},
{
  "portName": "managementPort6",
  "port": 20006
},
{
  "portName": "managementPort7",
  "port": 20007
},
{
  "portName": "dataPort1",
  "port": 20014
},
{
  "portName": "metadataPort1",
  "port": 20008
},
{
  "portName": "metadataPort2",
  "port": 20004
}
```

```
    },  
    {  
      "portName": "metadataPort3",  
      "port": 20011  
    },  
    {  
      "portName": "metadataPort4",  
      "port": 20012  
    },  
    {  
      "portName": "metadataPort5",  
      "port": 20000  
    },  
    {  
      "portName": "metadataPort6",  
      "port": 20015  
    },  
    {  
      "portName": "metadataPort7",  
      "port": 20009  
    },  
    {  
      "portName": "metadataPort8",  
      "port": 20001  
    },  
    {  
      "portName": "metadataPort9",  
      "port": 20010  
    }  
  ],  
  "portRange": "20000-20500",  
  "iSCSIPort": 3260,  
  "storagePorts": [  
    {  
      "port1": 20017,
```

```
        "port2": 20019,  
        "port3": 20018,  
        "diskpath": "/mnt/stor01",  
        "pathId": "1"  
    },  
    ],  
    "nodeName": "default:hblock_1",  
    "parentName": "default",  
    "baseServices": [  
        {  
            "name": "mdm",  
            "status": "Up",  
            "metaDir": "/mnt/storage01/CTYUN_HBlock_Plus_4.0.0_x64"  
        },  
        {  
            "name": "ls",  
            "status": "Up",  
            "metaDir": "/mnt/storage01/CTYUN_HBlock_Plus_4.0.0_x64"  
        },  
        {  
            "name": "cs",  
            "status": "Up",  
            "metaDir": "/mnt/storage01/CTYUN_HBlock_Plus_4.0.0_x64"  
        }  
    ],  
    "maxMemoryRatio": 0.95,  
    "maxMemorySize": 9223372036854775807  
}
```

4.11.10 添加数据目录

此操作用来添加指定服务器的数据目录至 HBlock。一次可以添加多个数据目录。

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。

请求语法

```
POST /rest/v1/system/server/serverId/diskpaths HTTP/1.1
```

Date: *date*

Content-Type: application/json; charset=utf-8

Content-Length: *length*

Host: *ip:port*

Authorization: *authorization*

```
{
  "diskPaths": [
    { "path":path,
      "capacityQuota": capacityvalue
    },
    { "path":path,
      "capacityQuota": capacityvalue
    },
    .....
  ]
}
```

请求参数

参数	类型	描述	是否必须
serverId	String	要添加的数据目录所属服务器 ID。	是
diskPaths	Array of diskPath	数据目录属性集合，详见“表 1 请求参数 diskPath	是

		说明”。	

表 1 请求参数 **diskPath** 说明

参数	类型	描述	是否必须
path	String	指定添加的数据目录。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。 取值：只能包含字母、数字、汉字和特殊字符(~!@\$()_+-;.:)。	是
capacityQuota	Long	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：小于数据目录所在磁盘的总容量，单位是 bytes。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

请求示例

添加服务器 **hblock_1** 的数据目录。

```
POST /rest/v1/system/server/hblock_1/diskpaths HTTP/1.1
Date: Mon, 7 Apr 2025 08:24:32 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 134
Host: 192.168.0.65:1443

{
```

```
"diskPaths":[
  { "path": "/mnt/storage01",
    "capacityQuota": 1024000
  },
  { "path": "/mnt/storage02"}
]
```

响应示例

```
HTTP/1.1 200 OK
x-hblock-request-id: 18243552b8744a2990f33ff8d2d37bd7
Connection: keep-alive
Date: Mon, 7 Apr 2025 08:24:32 GMT
Content-Length: 11
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {}
}
```

4.11.11 修改数据目录的容量配额

此操作用来修改指定服务器数据目录的容量配额。

请求语法

```
PUT /rest/v1/system/server/serverId/diskpaths HTTP/1.1
```

Date: *date*

Content-Type: application/json; charset=utf-8

Content-Length: *length*

Host: *ip:port*

Authorization: *authorization*

```
{
  "diskPaths": [
    { "path":path,
      "capacityQuota": capacityvalue
    },
    { "path":path,
      "capacityQuota": capacityvalue
    },
    .....
  ]
}
```

请求参数

参数	类型	描述	是否必须
serverId	String	要修改的数据目录所属服务器 ID。	是
diskPaths	Array of diskPath	数据目录属性集合，详见“表 1 请求参数 diskPath 说明”。	是

表 1 请求参数 `diskPath` 说明

参数	类型	描述	是否必须
<code>path</code>	String	指定要修改容量配额的数据目录。	是
<code>capacityQuota</code>	Long	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：小于数据目录所在磁盘的总容量，单位是 bytes。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。	是

请求示例

修改服务器 `hblock_2` 数据目录 `/mnt/stor` 的容量配额。

```
PUT /rest/v1/system/server/hblock_2/diskpaths HTTP/1.1
Date: Wed, 11 Oct 2023 08:06:57 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 128
Host: 192.168.0.117:1443

{
  "diskPaths": [
    {
      "path": "/mnt/stor",
      "capacityQuota": 2048000000
    }
  ]
}
```

```
}
```

响应示例

```
HTTP/1.1 204 No Content
x-hblock-request-id: 45f9f3b6359b48a8a663bb053d2d2731
Connection: keep-alive
Date: Wed, 11 Oct 2023 08:06:57 GMT
Content-Length: 0
Server: HBlock
```

4.11.12 移除数据目录

此操作用来移除数据目录。

注意：

- 强制移除数据目录，存在数据丢失风险。
- 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，才可以移除该目录。
- 如果执行日志采集后，产生的日志保存在 HBlock 的数据目录内，移除该数据目录时 `deleteLocalData` 为 `true`，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。

请求语法

```
DELETE /rest/v1/system/server/serverId/diskpaths HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "diskPaths": [
    path,
    path
  ],
  "force": force,
  "deleteLocalData": deleteLocalData
}
```

请求参数

参数	类型	描述	是否必须
serverId	String	要移除数据目录的服务器 ID。	是
diskPaths	Array of string	要移除的 HBlock 数据目录，一次可以移除多个数据目录。	是
force	Boolean	是否强制移除数据目录。 注意： 强制移除数据目录，可能造成数据丢失。 取值： ● true：强制移除数据目录。 ● false：不强制移除数据目录。 默认值为 false 。	否
deleteLocalData	Boolean	是否删除数据目录上的数据。 取值： ● true：删除数据目录上的数据。 ● false：不删除数据目录上的数据。 默认值为 false 。	否

请求示例

移除服务器 ID 为 hblock_2 的数据目录/mnt/storage2。

```
DELETE /rest/v1/system/server/hblock_2/diskpaths HTTP/1.1
Date: Fri, 01 Apr 2022 01:53:09 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 124
Host: 192.168.0.121:1443

{
  "diskPaths": [
```

```
    "/mnt/storage02"  
  ],  
  "force": "false",  
  "deleteLocalData": "false"  
}
```

响应示例

```
HTTP/1.1 202 Accepted  
Date: Fri, 1 Apr 2022 01:53:12 GMT  
x-hblock-request-id: 5f1bfda64dc24879b7ed62d4b366f07d  
Connection: keep-alive  
Content-Length: 0  
Server: HBlock
```

4.11.13 错误码列表

HTTP status	错误码	错误信息	描述
400	CanNotDeleteServer	The server can not be deleted. Please stop the server, prepare a new server with same IP, and recover it.	服务器不能被移除。
400	CanNotRemoveDefaultPath	This path <i>path</i> is the default disk path, please specify a new default disk path first.	无法移除默认数据目录。
400	CanNotRemoveServer	The base server <i>serverId</i> can not be removed. Please migrate the base service on it and retry.	基础服务器不能删除，请先迁移服务器上的基础服务，然后重试。
400	DeleteServerForcibly	Force delete should be used when deleting a server. When using this option, there may be a risk of data loss.	移除服务器时，必须使用强制移除，且有丢数据的风险。
400	ExceedDecimalPlaces	Value <i>value</i> at 'max memory ratio' failed to satisfy constraint: Argument must have no more than 2 decimal places.	参数“max memory ratio”只支持小数点后 2 位。
400	ExceedMaxPathCount	You can only add 100 local storage paths.	每台服务器最多只能添加 100 个数据目录。

400	InsufficientPath	The disk path can not be deleted because the server must have at least one disk path.	数据目录不能被移除，因为服务器至少要有有一个数据目录。
400	InsufficientPorts	Insufficient available ports for server <i>server</i> . Expand the port range to support <i>N</i> additional ports.	服务器 X 端口范围中可用端口数量不足，请调整范围使之至少增加 N 个端口。
400	InsufficientSpace	This method is not allowed because there is no more space to store data.	不能执行此操作，因为存储空间不足。
400	InvalidDiskPath	The disk paths do not exist: <i>serverIP:diskPaths</i> <i>[,diskPaths...]</i> [<i>;</i> <i>serverIP:diskPaths[,diskPaths..]</i>]. The available disk space must be greater than or equal to 1 GiB for each disk path: <i>serverIP:diskPaths</i> <i>[,diskPaths...]</i> [<i>;</i> <i>serverIP:diskPaths[,diskPaths..]</i>]. The disk paths can not access: <i>serverIP:diskPaths</i> <i>[,diskPaths...]</i> [<i>;</i> <i>serverIP:diskPaths[,diskPaths..]</i>].	因数据目录有问题导致操作失败。

		..]]. The disk paths must be directory: <i>serverIP:diskPaths</i> [<i>,diskPaths...</i>] [; <i>serverIP:diskPaths</i> [<i>,diskPaths.</i>..]]. The disk paths include incompatible data: <i>serverIP:diskPaths</i> [<i>,diskPaths...</i>] [; <i>serverIP:diskPaths</i> [<i>,diskPaths.</i>..]]. The disk paths include unsupported symbols: <i>serverIP:diskPaths</i> [<i>,diskPaths...</i>] [; <i>serverIP:diskPaths</i> [<i>,diskPaths..</i>..]]. The disk paths is already on the server and the status is removing: <i>serverIP:diskPaths</i> [<i>,diskPaths...</i>]	
400	InvalidDiskPathCapacityQuota	The capacity quota for the following disk paths failed to satisfy constraint. It must be integer and not greater than the	容量配额参数错误。

		total capacity of disk path. <i>serverIP:diskPaths</i> [, <i>diskPaths</i> ...] [; <i>serverIP:diskPaths</i> [, <i>diskPaths</i> ...]]..].	
400	InvalidLong	Value <i>value</i> at ' <i>argument</i> ' failed to satisfy constraint: Argument must be of type long.	参数填写错误，取值必须为长整型。
400	InvalidMetaDir	The meta directory <i>metaDir</i> is invalid. <i>reason</i>	服务数据目录无效，请修改后重试。
400	InvalidPathStatus	The path <i>path</i> status is ' <i>status</i> ', the request is invalid.	数据目录状态不正确，无法执行操作。
400	InvalidPort	Value <i>value</i> at ' <i>argument</i> ' failed to satisfy constraint: Argument must have value between 1 and 65535.	Port 的取值必须在[1, 65535]之间。
400	InvalidtargetPortalIP	Value at 'ips' failed to satisfy constraint: only one target portal IP is supported.	iSCSI 目标门户 IP、Port 只允许输入一组。
400	MissingtargetPortalIP	Value null at 'ip' failed to satisfy constraint: Argument must not be null when target portal IP is enabled.	iSCSI 目标门户 IP 不能为空。
400	MissingtargetPortalIPOrPort	Target portal IP and port must exist at the same time.	iSCSI 目标门户 IP 和 Port 要同时存在。

400	MissingtargetPortalPort	Value null at 'port' failed to satisfy constraint: Argument must not be null when target portal IP is enabled.	iSCSI 目标门户 Port 不能为空。
400	NoSuchService	The <i>serviceName</i> service does not exist on server <i>serverid</i> .	待迁移的基础服务在源服务器不存在。
400	NotInterfaceIP	The IP <i>IP</i> is not an interface IP address, check and retry.	IP 地址不是服务器的接口 IP，请修改并重试。
400	PathUsed	The path <i>path</i> is being used by LUN <i>lunName</i> , <i>lunName</i> ..., please delete the LUN first.	路径正在被卷使用，请先删除卷后再重试。
400	RemoveFaultDomainRisk	Can not remove multiple fault domains at the same time. You can use force remove. When using this option, there may be a risk of data loss.	无法同时移除的多个故障域。
400	RemoveNodeRisk	The node <i>nodeName</i> or its child nodes are being used by a storage pool. Removing the node may cause data loss. You can use force remove. When using this option, there may be a risk of data loss.	移除节点可能会引起数据丢失。只能使用强制移除。请注意，强制移除，会产生数据丢失风险，请谨慎操作。
400	RemovePathRisk	The path <i>path</i> status is ' <i>status</i> '. Removing the path may cause	数据目录状态不正常，移除数据目录可

		data loss. You can use force remove. When using this option, there may be a risk of data loss.	能会引起数据丢失。 只能使用强制移除。 请注意，强制移除，会产生数据丢失风险，请谨慎操作。
400	RemovePathRisk	There is LUN with redundancy mode, which has the minimum requirement for the number of paths. LUN: <i>lunName1</i>[, <i>lunName2</i>....] Removing the path may cause data loss. You can use force remove. When using this option, there may be a risk of data loss.	移除数据目录可能会引起数据丢失。只能使用强制移除。请注意，强制移除，会产生数据丢失风险，请谨慎操作。 数据目录移除后会造成卷数据损毁： <i>lunName1</i>[, <i>lunName2</i>....]。
400	RemoveServerRisk	There is LUN with high availability disabled, and the LUN is connected with the client. LUN: <i>lunName1</i>[, <i>lunName2</i>....] There is LUN with redundancy mode, which has the minimum requirement for the number of servers. LUN: <i>lunName1</i>[,	有单控卷，且处于连接状态，移除服务器会导致卷冗余度失败。 移除服务器会导致卷冗余度失败。 移除服务器之后，该服务器上面部分的 IQN 无法在其他机器

		<i>lunName2....]</i> There is target IQN which has no available server to migrate, removing server will cause the IQN deleted. target: <i>targetIQN1_[, targetIQN2....]</i> Removing the server may cause data loss. You can use force remove. When using this option, there may be a risk of data loss.	上重建。 移除服务器时，有可能丢数据。如果必须移除，请使用强制移除。
400	RestartServerRisk	There is no more space to store data or HBlock encountered an internal error. You can use force restart. When using this option, there may be a risk of data loss.	重启服务器失败。可以强制重启，但有丢失数据风险。
400	SetConfigFailed	Failed to set the config '<i>argument1</i>', '<i>argument2</i>'...: <i>serverid1</i>, <i>serverid2</i>.... Failed to set the configuration on one or more servers: <i>serverid1</i>: <i>class1</i>, <i>class2</i> <i>serverid2</i>: <i>class2</i>, <i>class4</i>	配置失败。
400	StandaloneModeNotAllowed	'<i>operation</i>' is not supported by	单机版本 HBlock 不支

		standalone mode of HBlock.	持该操作。
400	SystemCrashRisk	The node <i>nodename</i> can not be removed because there is only one available fault domain left in the base storage pool, or the node is related to multiple fault domains in the base pool, removing node may result in a system crash.	无法移除节点，存在以下情况之一，移除节点可能引发系统崩溃：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；节点涉及基础存储池的多个故障域。
404	NoSuchPath	The path <i>path</i> does not exist in the HBlock.	路径不存在。
404	NoSuchServer	The server with ID <i>serverId</i> does not exist.	服务器不存在。
404	NotargetPortalIP	Target portal IP is not configured.	没有配置 target portal IP。
409	ConflictWithServiceMigrating	Failed to <i>action</i> because the service <i>serviceName</i> on <i>serverId</i> is migrating to <i>serverid</i> .	基础服务正在迁移，无法执行操作。
409	InvalidServiceStatus	The status of service <i>serviceName</i> on <i>serverId</i> is ' <i>status</i> ', the request is invalid.	某节点上的服务状态异常，不运行执行操作。
409	InvalidServerStatus	The server <i>serverId</i> status is ' <i>status</i> ', the request is invalid.	服务器当前的状态不正确，请求无效。
409	PortConflict	Error: the following port is/ports are in use.	端口冲突。

		<i>server_ID/IP/local server:</i> <i>port_name port[, port_name port...].</i>	
409	RemovingAnotherPath	There is another path being removed. You can use force remove. When using this option, there may be a risk of data loss.	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。
409	RemovingAnotherServer	There is another server being removed. You can use force remove. When using this option, there may be a risk of data loss.	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。
409	ServiceAlreadyExists	The service <i>serviceName</i> already exists on server <i>serverid</i> .	待迁移的服务在目标服务器上已存在。
409	ServerAlreadyExists	The server with IP <i>IP</i> already exists.	服务器 IP 已经存在。
500	InsufficientResource	Operation failed due to insufficient resources.	内存或其他资源不足，操作失败。

4.12 查询 HBlock 信息

此操作用来查询 HBlock 信息，包括 HBlock ID、HBlock 名称、HBlock 序列号、服务器数量、故障域（集群版）、初始化时间、试用期信息及 HBlock 的管理员用户名。

请求语法

```
GET /rest/v1/system/info HTTP/1.1
```

Date: *date*

Host: *ip:port*

Authorization: *authorization*

响应结果

名称	类型	描述
storId	String	HBlock ID。
storName	String	HBlock 的名称。
serialId	String	HBlock 序列号。
serverNumbers	Integer	HBlock 服务器数量。
basePoolName	String	基础存储池的名字（仅集群版）。
faultDomain	String	故障域类型（仅集群版）： ● rack: 机架级别的故障域。 ● room: 机房级别的故障域。 ● server: 服务器级别的故障域。 ● path: 数据目录级别的故障域。
setupTime	Long	系统初始化时间。unix 时间戳（UTC），精确到毫秒。
trialStatus	String	试用期标识： ● 对于商业版 ■ Trial: 处于试用期内，未导入软件许可

		证。 ■ TrialExpired: 试用期已过，未导入软件许可证。 ■ NonTrialVersion: 非试用版本，已导入软件许可证。 ● 对于免费版： ■ FreeEdition: 未导入软件许可证，未开启高级功能试用。 ■ ProTrialEnabled: 未导入软件许可证，开启了高级功能试用。 ■ ProTrialExpired: 未导入软件许可证，高级功能试用已经过期。
trialExpireTime	Long	试用期到期时间（商业版）。unix 时间戳（UTC），精确到毫秒。
userName	String	HBlock 的管理员用户名。
licenseId	String	最后一次成功导入的软件许可证 ID（商业版）。
type	String	软件许可证订购类型（商业版）： ● Subscription: 订阅模式。 ● Perpetual: 永久许可模式。
status	String	软件许可证状态（商业版）： ● 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。
maintenanceExpireTime	Long	对于永久许可模式软件许可证，软件许可证维保过期时间。unix 时间戳（UTC），精确到毫秒。
subscribeExpireTime	Long	对于订阅模式软件许可证，软件许可证过期时间。unix 时间戳（UTC），精确到毫秒。

proTrialStartTime	Long	免费版的高级功能试用开启时间，unix 时间戳（UTC），精确到毫秒。高级功能试用开启之后才返回，不开启不返回。
proTrialExpireTime	Long	免费版的高级功能试用结束时间，unix 时间戳（UTC），精确到毫秒。高级功能试用开启之后才返回，不开启不返回。
freeUpgradeExpireTime	Long	免费版的升级功能过期时间，unix 时间戳（UTC），精确到毫秒。

请求示例 1

单机-商业版：查询 HBlock 信息。

```
GET /rest/v1/system/info HTTP/1.1
Date: Fri, 26 Dec 2025 07:50:09 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.66:1443
```

响应示例 1

```
HTTP/1.1 200 OK
x-hblock-request-id: 55cd9d5a2a0d4fafb873d337971848c7
Connection: keep-alive
Content-Length: 401
Date: Fri, 26 Dec 2025 07:50:09 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "storId": "0A769770-1528-49B8-A82D-D1CB71333FFB",
    "storName": "storuser",
    "serialId": "0A769770-1528-49B8-A82D-D1CB71333FFB-0201-040000",
    "serverNumbers": 1,
```

```
"setupTime": 1765259023707,
"trialStatus": "NonTrialVersion",
"trialExpireTime": 1767851023707,
"userName": "storuser",
"licenseId": "qws2b6a9-f3fb-4098-a6b3-3652a5a76530",
"type": "Perpetual",
"status": "Effective",
"maintenanceExpireTime": 1798255505624
}
}
```

请求示例 2

单机-免费版（已开启高级功能试用）：查询 HBlock 信息。

```
GET /rest/v1/system/info HTTP/1.1
Date: Mon, 12 Jan 2026 03:56:23 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.66:1443
```

响应示例 2

```
HTTP/1.1 200 OK
x-hblock-request-id: 5b8713084ded4ebdb7b84f6d68d831b6
Connection: keep-alive
Content-Length: 344
Date: Mon, 12 Jan 2026 03:56:23 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "storId": "C3238159-40D5-4E6E-A719-1A5205BB688D",
    "storName": "stor1",
    "serialId": "C3238159-40D5-4E6E-A719-1A5205BB688D-0201-040000",
    "serverNumbers": 1,
    "setupTime": 1768185567944,
```

```
    "trialStatus": "ProTrialEnabled",
    "userName": "storuser",
    "proTrialStartTime": 1768188583000,
    "proTrialExpireTime": 1770780583000,
    "freeUpgradeExpireTime": 1831257567944
  }
}
```

请求示例 3

集群-商业版：查询 HBlock 信息。

```
GET /rest/v1/system/info HTTP/1.1
Date: Fri, 26 Dec 2025 07:20:37 GMT
Authorization: HBlock userName:signature
Host:192.168.0.65:1443
```

响应示例 3

```
HTTP/1.1 200 OK
x-hblock-request-id: 0e653a2596334cc18aa7517a1250129f
Connection: keep-alive
Content-Length: 447
Date: Fri, 26 Dec 2025 07:20:37 GMT
Content-Type: application/json; charset=utf-8
Server: HBlock

{
  "data": {
    "storId": "1E46239F-BD8B-4CB3-A151-C3A58892AEF9",
    "storName": "stor1",
    "serialId": "1E46239F-BD8B-4CB3-A151-C3A58892AEF9-0202-040000",
    "serverNumbers": 3,
    "basePoolName": "default",
    "faultDomain": "server",
    "setupTime": 1766021632008,
    "trialStatus": "NonTrialVersion",
```

```
{
  "trialExpireTime": 1768613632008,
  "userName": "storuser",
  "licenseId": "ehc2b6a9-f3fb-4098-a6b3-3652a5d71232",
  "type": "Subscription",
  "status": "Effective",
  "subscribeExpireTime": 1798255470002
}
```

请求示例 4

集群-免费版（未开启高级功能试用）：查询 HBlock 信息。

```
GET /rest/v1/system/info HTTP/1.1
Date: Mon, 12 Jan 2026 03:58:18 GMT
Authorization: HBlock userName:signature
Host:192.168.0.65:1443
```

响应示例 4

```
HTTP/1.1 200 OK
x-hblock-request-id: 74a51ff96dc444d9b4107293fce19a9b
Connection: keep-alive
Content-Length: 319
Date: Mon, 12 Jan 2026 03:58:18 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "storId": "10266612-3B09-4111-968A-CBB4D271F358",
    "storName": "stor2",
    "serialId": "10266612-3B09-4111-968A-CBB4D271F358-0202-040000",
    "serverNumbers": 3,
    "basePoolName": "default",
```

```
    "faultDomain": "server",  
    "setupTime": 1768187650175,  
    "trialStatus": "FreeEdition",  
    "userName": "storuser",  
    "freeUpgradeExpireTime": 1831259650175  
  }  
}
```

4.13 查询 HBlock 服务状态

此操作用来查询 HBlock 状态，包括系统、存储池（集群版）、服务器、数据目录（单机版）、卷的状态。

请求语法

```
GET /rest/v2/system/status HTTP/1.1
```

Date: *date*

Host: *ip:port*

Authorization: *authorization*

响应结果

名称	类型	描述
system.status	String	HBlock 服务状态： ● Upgrading: 升级中。 ● Uninstalling: 卸载中。 ● Working: 运行中。
system.licenseStatus	String	软件许可证状态： ● Expired: 软件许可证已过期。 ● Effective: 软件许可证已生效。 ● Invalid: 软件许可证无效。 ● None: 还未加载软件许可证。
server.status	Object	服务器的状态，详见“表 1 响应参数 server.status 说明”。
disk.status	Object	磁盘的状态（仅单机版支持），详见“表 2 响应参数 disk.status 说明（仅单机版支持）”。
disk.usage	Object	HBlock 数据目录使用情况，详见“表 3 响应

		参数 disk.usage 说明（仅单机版支持）”。
lun.status	Object	卷状态，详见“表 4 响应参数 lun.status 说明”。
lun.data	Object	卷数据信息，详见“表 5 响应参数 lun.data 说明”。
storagePool.number	Integer	存储池个数（仅集群版支持）。
storagePool.basePool	String	基础存储池名称（仅集群版支持）。
storagePool.basePoolDetail	Object	基础存储池详情（仅集群版支持），详见“表 6 响应参数 storagePool.basePoolDetail 说明（仅集群版支持）”。

表 1 响应参数 **server.status** 说明

名称	类型	描述
connected	Integer	状态为连接的服务器个数。
disconnected	Integer	状态为断开连接的服务器个数。
removing	Integer	状态为正在移除的服务器个数。

表 2 响应参数 **disk.status** 说明（仅单机版支持）

名称	类型	描述
healthy	Integer	状态为健康的磁盘个数。
warning	Integer	状态为告警的磁盘个数。
error	Integer	状态为错误的磁盘个数。

表 3 响应参数 `disk.usage` 说明（仅单机版支持）

名称	类型	描述
<code>used</code>	Long	HBlock 数据目录对应分区的已用容量，单位是 bytes。
<code>total</code>	Long	HBlock 数据目录对应分区的总容量，单位是 bytes。

表 4 响应参数 `lun.status` 说明

名称	类型	描述
<code>number</code>	Integer	卷的总个数。
<code>cacheNumber</code>	Integer	缓存模式卷的个数。
<code>storageNumber</code>	Integer	存储模式卷的个数。
<code>localNumber</code>	Integer	本地模式卷的个数。
<code>cacheCapacity</code>	Long	缓存模式卷的容量，单位是 bytes。
<code>storageCapacity</code>	Long	存储模式卷的容量，单位是 bytes。
<code>localCapacity</code>	Long	本地模式卷的容量，单位是 bytes。

表 5 响应参数 `lun.data` 说明

名称	类型	描述
<code>normal</code>	Double	卷数据中，正常状态的量百分比。
<code>lowRedundancy</code>	Double	卷数据中，降级状态的百分比。
<code>error</code>	Double	卷数据中，错误状态的百分比。
<code>reconstructionProgress</code>	Double	如果存在降级状态的数据，数据重建进度的百分比。
<code>ETR</code>	Long	如果存在降级数据，预计剩余的数据重建时间。单位为秒。 如果返回值为负数，说明数据无法重建，需用户尽快修复。

表 6 响应参数 `storagePool.basePoolDetail` 说明（仅集群版支持）

名称	类型	描述
<code>faultDomain.level</code>	String	基础存储池的故障域级别。

faultDomain.status.healthy	Integer	状态为健康的故障域个数。
faultDomain.status.warning	Integer	状态为告警的故障域个数。
faultDomain.status.error	Integer	状态为错误的故障域个数。
disk.status.healthy	Integer	状态为健康的磁盘个数。
disk.status.warning	Integer	状态为告警的磁盘个数。
disk.status.error	Integer	状态为错误的磁盘个数。
disk.usage.used	Long	基础存储池中 HBlock 数据目录对应分区的已用容量，单位是 bytes。
disk.usage.total	Long	基础存储池中 HBlock 数据目录对应分区的总容量，单位是 bytes。
dataHealth.normal	Double	该存储池所有正常数据的百分比。
dataHealth.lowRedundancy	Double	该存储池所有降级数据的百分比。
dataHealth.error	Double	该存储池所有错误数据的百分比。
dataHealth.reconstructionProgress	Double	如果该存储池存在降级状态的数据，数据重建进度的百分比。
dataHealth.ETR	Long	如果该存储池卷存在降级数据，预计剩余的数据重建时间。单位为秒。如果返回值为负数，说明数据无法重建，需用户尽快修复。

请求示例 1

单机版：查询 HBlock 状态。

```
GET /rest/v2/system/status HTTP/1.1
Date: Tue, 09 Dec 2025 06:11:13 GMT
Authorization: HBlock userName:signature
Host:192.168.0.66:1443
```

响应示例 1

```
HTTP/1.1 200 OK
```

x-hblock-request-id: b9a47583b571487697f327cfd70723fe

Connection: keep-alive

Content-Length: 440

Date: Tue, 09 Dec 2025 06:11:13 GMT

Content-Type: application/json; charset=utf-8

Server: HBlock

```
{
  "data": {
    "system": {
      "status": "Working",
      "licenseStatus": "None"
    },
    "server": {
      "status": {
        "connected": 1,
        "disconnected": 0,
        "removing": 0
      }
    },
    "disk": {
      "status": {
        "healthy": 1,
        "warning": 0,
        "error": 0
      },
      "usage": {
        "used": 1431740416,
        "total": 100169342976
      }
    },
    "lun": {
      "status": {
        "number": 2,
        "cacheNumber": 1,
```

```
        "storageNumber": 0,
        "localNumber": 1,
        "cacheCapacity": 236223201280,
        "storageCapacity": 0,
        "localCapacity": 107374182400
    },
    "data": {
        "normal": 100,
        "lowRedundancy": 0,
        "error": 0
    }
}
```

请求示例 2

集群版：查询 HBlock 状态。

```
GET /rest/v2/system/status HTTP/1.1
Date: Tue, 09 Dec 2025 02:42:02 GMT
Authorization: HBlock userName:signature
Host:192.168.0.64:1443
```

响应示例 2

```
HTTP/1.1 200 OK
x-hblock-request-id: 34e45fb7348647a6afbfb1dcb5b602cf
Connection: keep-alive
Content-Length: 660
Date: Tue, 09 Dec 2025 02:42:02 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
    "data": {
        "system": {
```

```
    "status": "Working",
    "licenseStatus": "Effective"
  },
  "server": {
    "status": {
      "connected": 3,
      "disconnected": 0,
      "removing": 0
    }
  },
  "storagePool": {
    "number": 2,
    "basePool": "default",
    "basePoolDetail": {
      "faultDomain": {
        "level": "server",
        "status": {
          "healthy": 3,
          "warning": 0,
          "error": 0
        }
      },
      "disk": {
        "status": {
          "healthy": 4,
          "warning": 0,
          "error": 0
        },
        "usage": {
          "used": 18785857536,
          "total": 342038982656
        }
      },
      "dataHealth": {
        "normal": 100,
```

```
        "lowRedundancy": 0,  
        "error": 0  
      }  
    },  
    },  
    "lun": {  
      "status": {  
        "number": 15,  
        "cacheNumber": 1,  
        "storageNumber": 2,  
        "localNumber": 12,  
        "cacheCapacity": 225485783040,  
        "storageCapacity": 323196289024,  
        "localCapacity": 928786677760  
      },  
      "data": {  
        "normal": 100,  
        "lowRedundancy": 0,  
        "error": 0  
      }  
    }  
  }  
}
```

4.14 监控

4.14.1 获取实时监控数据

此操作用来获取 HBlock 的实时性能数据。

请求语法

```
GET
/rest/v1/system/monitor/current?dimension=dimension&instanceId=instanceId&metric=metric
HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

请求参数

参数	类型	描述	是否必须
dimension	String	监控对象。 取值： ● system: 系统。 ● pool: 存储池（仅集群版支持）。 ● server: 服务器。 ● disk: 数据目录所在磁盘分区。 ● LUN: 卷。 单机版默认值为 system，集群版默认值为 pool。	否
instanceId	String	监控对象实例的唯一标识，可以填写多个，以英文逗号隔开。 取值： ● 监控对象为 system，没有实例。	否

		● 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。 ● 监控对象为 disk，实例取值为 “serverId+路径名称”，格式为 serverId:/diskpath。 ● 监控对象为 LUN，实例取值为卷名称。 如果不填写，默认查看监控对象下的所有实例性能数据。	
metric	String	指标名称，可以填写多个，以英文逗号隔开。指标具体描述详见监控指标。 取值： ● 监控对象为 system: ■ IOPS ■ R_IOPS ■ W_IOPS ■ Bandwidth ■ R_Bandwidth ■ W_Bandwidth ■ Latency ■ W_Latency ■ R_Latency ■ Path_Cap ■ Path_Used ■ Path_Rate ■ Path_Cap_Quota ■ Path_Cap_Quota_Used ■ Path_Cap_Quota_Rate ■ Cloud_Bandwidth ■ Cloud_U_Bandwidth	否

		■ Cloud_D_Bandwidth ● 监控对象为 pool: ■ IOPS ■ R_IOPS ■ W_IOPS ■ Bandwidth ■ R_Bandwidth ■ W_Bandwidth ■ Latency ■ W_Latency ■ R_Latency ■ Path_Cap ■ Path_Used ■ Path_Rate ■ Path_Cap_Quota ■ Path_Cap_Quota_Used ■ Path_Cap_Quota_Rate ● 监控对象为 server: ■ CPU_Rate ■ Mem_Rate ■ Mem_Total ■ Mem_Used ■ IOPS ■ R_IOPS ■ W_IOPS ■ Bandwidth ■ R_Bandwidth	

		■ W_Bandwidth ■ Latency ■ W_Latency ■ R_Latency ■ Path_Cap ■ Path_Used ■ Path_Rate ■ Path_Cap_Quota ■ Path_Cap_Quota_Used ■ Path_Cap_Quota_Rate ■ Cloud_Bandwidth ■ Cloud_U_Bandwidth ■ Cloud_D_Bandwidth ● 监控对象为 disk: ■ Path_Cap ■ Path_Used ■ Path_Rate ■ Path_Cap_Quota ■ Path_Cap_Quota_Used ■ Path_Cap_Quota_Rate ● 监控对象为 LUN: ■ IOPS ■ R_IOPS ■ W_IOPS ■ Bandwidth ■ R_Bandwidth ■ W_Bandwidth	

		■ Latency ■ W_Latency ■ R_Latency ■ Cloud_Bandwidth ■ Cloud_U_Bandwidth ■ Cloud_D_Bandwidth ■ Wait_Upload 不填写，默认查询监控对象的全部指标。	

响应结果

名称	类型	描述
monitoring	Array of monitoring	监控信息集合，详见“表 1 响应参数 monitoring 说明”。

表 1 响应参数 monitoring 说明

名称	类型	描述
dimension	String	监控对象： ● system：系统。 ● pool：存储池（仅集群版支持）。 ● server：服务器。 ● disk：数据目录所在磁盘分区。 ● LUN：卷。
instanceId	String	监控对象实例： ● 监控对象为 system，显示 HBlock 名称。 ● 监控对象为 pool，显示存储池名称（仅集群版支持）。 ● 监控对象为 server，显示为服务器 ID。

		● 监控对象为 disk，显示为数据目录。 ● 监控对象为 LUN，显示为卷名称。
metric	String	指标名称。具体详见 监控指标 。
avgValueSeries	Array of string	监控数据的[时间,数值]序列。 时间为 unix 时间戳，精确到毫秒。

请求示例

查询服务器 hblock_1、hblock_2 的 CPU 使用率和内存使用率。

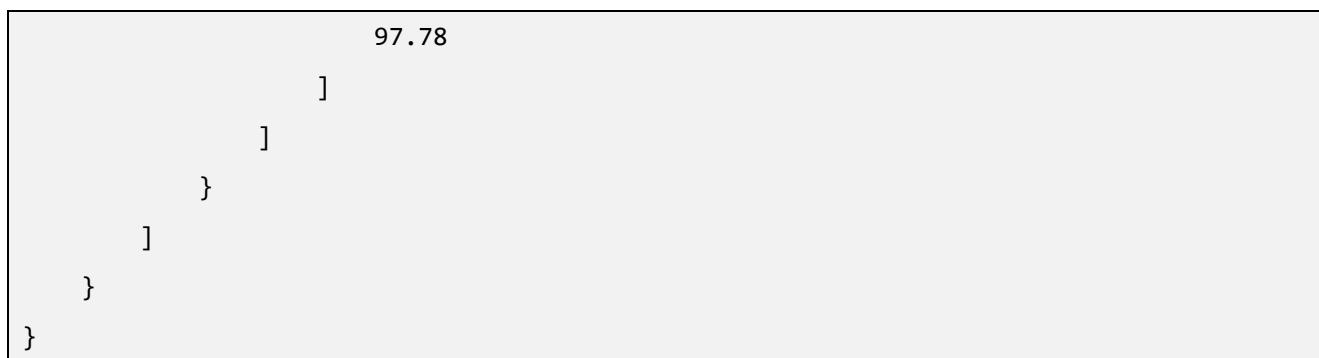
```
GET
/rest/v1/system/monitor/current?dimension=server&instanceId=hblock_1,hblock_2&metric=CPU_Rate,Mem_Rate HTTP/1.1
Date: Fri, 12 Aug 2022 07:48:25 GMT
Authorization: HBlock userName:signature
Host:192.168.0.121:1443
```

响应示例

```
HTTP/1.1 200 OK
x-hblock-request-id: 2e40eab6db8440b5b5ec68f2dd5f8cf1
Connection: keep-alive
Content-Length: 455
Date: Fri, 12 Aug 2022 07:48:25 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "monitoring": [
      {
        "dimension": "server",
        "instanceId": "hblock_1",
        "metric": "CPU_Rate",
        "avgValueSeries": [
```

```
[
    [
        1660290480000,
        1.35
    ]
],
{
    "dimension": "server",
    "instanceId": "hblock_1",
    "metric": "Mem_Rate",
    "avgValueSeries": [
        [
            1660290480000,
            98.15
        ]
    ]
},
{
    "dimension": "server",
    "instanceId": "hblock_2",
    "metric": "CPU_Rate",
    "avgValueSeries": [
        [
            1660290480000,
            0.94
        ]
    ]
},
{
    "dimension": "server",
    "instanceId": "hblock_2",
    "metric": "Mem_Rate",
    "avgValueSeries": [
        [
            1660290480000,
```



4.14.2 获取历史性能数据

此操作用来获取 HBlock 的历史性能数据。

请求语法

```
GET
/rest/v1/system/monitor?startTime=startTime&endTime=endTime&dimension=dimension&instanceId=instanceId1&metric=metric1,metric2 HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

请求参数

参数	类型	描述	是否必须
startTime	String	查询起始时间。 取值：unix 时间戳（UTC），精确到毫秒。默认值为结束时间 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。	否
endTime	String	查询结束时间。 取值：unix 时间戳（UTC），精确到毫秒。默认值为当前时间。 注意：起始时间必须早于结束时间。	否
dimension	String	监控对象。 取值： ● system: 系统。 ● pool: 存储池（仅集群版支持）。 ● server: 服务器。	否

		● disk: 数据目录所在磁盘分区。 ● LUN: 卷。 单机版默认值为 system，集群版默认值为 pool。	
instanceId	String	监控对象实例的唯一标识。 取值: ● 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（仅集群版支持）。 ● 监控对象为 server，必填，实例取值为服务器 ID。 ● 监控对象为 disk，必填，实例取值为 “<i>serverId+路径名称</i>”，格式为 <i>serverId:/diskpath</i>。 ● 监控对象为 LUN，必填，实例取值为卷名称。	否
metric	String	指标名称，可以填写多个，以英文逗号隔开。指标具体描述详见监控指标。 取值: ● 监控对象为 system: ■ IOPS ■ R_IOPS ■ W_IOPS ■ Bandwidth ■ R_Bandwidth ■ W_Bandwidth ■ Latency ■ W_Latency ■ R_Latency	否

		■ Path_Cap ■ Path_Used ■ Path_Rate ■ Path_Cap_Quota ■ Path_Cap_Quota_Used ■ Path_Cap_Quota_Rate ■ Cloud_Bandwidth ■ Cloud_U_Bandwidth ■ Cloud_D_Bandwidth ● 监控对象为 pool: ■ IOPS ■ R_IOPS ■ W_IOPS ■ Bandwidth ■ R_Bandwidth ■ W_Bandwidth ■ Latency ■ W_Latency ■ R_Latency ■ Path_Cap ■ Path_Used ■ Path_Rate ■ Path_Cap_Quota ■ Path_Cap_Quota_Used ■ Path_Cap_Quota_Rate ● 监控对象为 server: ■ CPU_Rate	

		■ Mem_Rate ■ Mem_Total ■ Mem_Used ■ IOPS ■ R_IOPS ■ W_IOPS ■ Bandwidth ■ R_Bandwidth ■ W_Bandwidth ■ Latency ■ W_Latency ■ R_Latency ■ Path_Cap ■ Path_Used ■ Path_Rate ■ Path_Cap_Quota ■ Path_Cap_Quota_Used ■ Path_Cap_Quota_Rate ■ Cloud_Bandwidth ■ Cloud_U_Bandwidth ■ Cloud_D_Bandwidth ● 监控对象为 disk: ■ Path_Cap ■ Path_Used ■ Path_Rate ■ Path_Cap_Quota ■ Path_Cap_Quota_Used	

		■ Path_Cap_Quota_Rate ● 监控对象为 LUN: ■ IOPS ■ R_IOPS ■ W_IOPS ■ Bandwidth ■ R_Bandwidth ■ W_Bandwidth ■ Latency ■ W_Latency ■ R_Latency ■ Cloud_Bandwidth ■ Cloud_U_Bandwidth ■ Cloud_D_Bandwidth ■ Wait_Upload 不填写，默认获取监控对象的全部指标。	

响应结果

名称	类型	描述
monitoring	Array of monitoring	监控信息集合，详见“表 1 响应参数 monitoring 说明”。

表 1 响应参数 monitoring 说明

名称	类型	描述
dimension	String	监控对象: ● system: 系统。 ● pool: 存储池（仅集群版支持）。

		● server: 服务器。 ● disk: 数据目录所在磁盘分区。 ● LUN: 卷。
instanceId	String	监控对象实例： ● 监控对象为 system，显示 HBlock 名称。 ● 监控对象为 pool，显示存储池名称（仅集群版支持） ● 监控对象为 server，显示为服务器 ID。 ● 监控对象为 disk，显示为 “<i>serverId</i>+路径名称”，格式为 <i>serverId:/path/dir</i>。 ● 监控对象为 LUN，显示为卷名称。
metric	String	指标名称。具体详见 监控指标 。
avgValueSeries	Array of string	监控数据的[时间,数值]序列。 时间为 unix 时间戳，精确到毫秒。

请求示例

查询卷 lun01a 2022-08-12 13:43:57 至 2022-08-12 14:03:57 的 IOPS 和带宽。

```
GET
/rest/v1/system/monitor?startTime=1660283037000&endTime=1660284237000&dimension=LUN&instanceId=lun01a&metric=IOPS,Bandwidth HTTP/1.1
Date: Fri, 12 Aug 2022 06:07:39 GMT
Authorization: HBlock userName:signature
Host:192.168.0.121:1443
```

响应示例

```
HTTP/1.1 200 OK
x-hblock-request-id: bdbfd372ebf143ae8c08c79b09909fc9
Connection: keep-alive
Content-Length: 2464
Date: Fri, 12 Aug 2022 06:07:39 GMT
Content-Type: application/json;charset=utf-8
```

Server: HBlock

```
{
  "data": {
    "monitoring": [
      {
        "dimension": "LUN",
        "instanceId": "lun01a",
        "metric": "IOPS",
        "avgValueSeries": [
          [
            1660283020000,
            0
          ],
          [
            1660283040000,
            0
          ],
          [
            1660283060000,
            0
          ],
          .....
          [
            1660284120000,
            0
          ],
          [
            1660284140000,
            0
          ],
          [
            1660284160000,
```

```
        0.45
      ],
      [
        1660284180000,
        304.75
      ],
      [
        1660284200000,
        108
      ],
      [
        1660284220000,
        141.9
      ],
      [
        1660284240000,
        84.8
      ]
    ]
  },
  {
    "dimension": "LUN",
    "instanceId": "lun01a",
    "metric": "Bandwidth",
    "avgValueSeries": [
      [
        1660283020000,
        0
      ],
      [
        1660283040000,
        0
      ],
      [
        1660283060000,
```

```
0
],
.....

[
    1660284080000,
    0
],
[
    1660284100000,
    0
],
[
    1660284120000,
    0
],
[
    1660284140000,
    0
],
[
    1660284160000,
    1843
],
[
    1660284180000,
    79475506
],
[
    1660284200000,
    28099583
],
[
    1660284220000,
```

```
    37126348
    ],
    [
    1660284240000,
    22083788
    ]
  ]
}
]
```

4.14.3 错误码列表

HTTP status	错误码	错误信息	描述
400	ArgumentNotAllowed	The argument ' <i>argument</i> ' is not allowed when value at ' <i>argument2</i> ' is <i>value2</i> .	参数值不正确。
400	ExceedQueryRange	Value <i>value</i> at ' <i>argument</i> ' failed to satisfy constraint: Argument must be <i>message</i> .	超出限定范围，无法执行该操作。
400	InvalidEnumValue	Value <i>value</i> at ' <i>argument</i> ' failed to satisfy constraint: Argument must satisfy enum value set: [<i>value1</i> , <i>value2</i> ...].	枚举参数不合法。
400	InvalidTime	The end time must be later than the start time.	开始时间必须早于结束时间
400	InvalidTimeFormat	Value <i>value</i> at ' <i>argument</i> ' failed to satisfy constraint: Argument must conform to <i>Format</i> format.	时间格式不合法。

4.15 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。

4.15.1 批量查询 HBlock 告警信息

此操作用来查询 HBlock 告警信息。

请求语法

```
GET /rest/v1/system/alarm?alarmStatus=alarmStatus&filter=filter&range=i-j HTTP/1.1  
Date: date  
Host: ip:port  
Authorization: authorization
```

请求参数

名称	类型	描述	是否必须
alarmStatus	String	告警状态： ● Unresolved: 告警中。 ● Resolved: 告警已解除。 ● Expired: 告警已失效。 默认值为 Unresolved。 说明： 已解除、已失效数据仅支持查询最近 10000 条。告警中的数据超出 10000 条告警，功能将不可用。	否
filter	String	设置查询的过滤条件。 过滤条件由查询属性和属性值组成，其中属性和属性值之间用单个冒号分隔（key:value）表示模糊匹配，用两个冒号分隔（key::value）表示精确匹配。可以选择多个查询，如果是或的关系，使用“or”将查询条件分隔开；如果是与的关系，使	否

		用“and”将查询条件分隔开。 支持的查询类型包括： ● instanceSnapshot: 告警实例的详情。 ● alarmRule: 告警规则编码名称，详见告警列表。 ● severity: 告警级别： ■ Warning: 警告。 ■ Major: 重要。 ■ Critical: 严重。 ● muteStatus: 静默状态： ■ Normal: 正常。 ■ Muted: 静默。 ● alarmTime: 告警发生时间，通过>和<支持范围查询。 unix 时间戳（UTC），精确到毫秒。 ● resolveTime: 解除时间，通过>和<支持范围查询。unix 时间戳（UTC），精确到毫秒。 ● resolveType: 告警解除方式： ■ Manual: 手动解除警告。 ■ Auto: 自动解除告警。 ● expireTime: 告警失效时间，通过>和<支持范围查询。 unix 时间戳（UTC），精确到毫秒。	
range	String	查询范围，格式：i-j，其中 i,j 是正整数，i 小于 j，j-i<1000。 按照 alarmTime 进行逆序排序，返回第 i 条到第 j 条的告警。 如果输入的查询范围超过告警的总个数，那么返回空结果集。 如果不输入 range 参数，那么返回所有符合条件的告警。	否

响应结果

名称	类型	描述
totalCount	Integer	符合查询条件的告警条数。
alarmList	Array of alarmList	告警列表，详见“表 1 响应参数 alarmList 说明”。

表 1 响应参数 alarmList 说明

名称	类型	描述
alarmId	String	告警 ID。
instanceId	String	告警实例 ID。
instanceSnapshot	String	告警实例快照，即告警发生时告警实例的详细信息。
alarmRule	String	告警规则。
severity	Enum	告警级别： ● Warning: 警告。 ● Major: 重要。 ● Critical: 严重。
alarmStatus	Enum	告警状态： ● Unresolved: 告警中。 ● Resolved: 告警已解除。 ● Expired: 告警已失效。
duration	Long	告警持续时长。unix 时间戳（UTC），精确到毫秒。
alarmTime	Long	告警发生时间。unix 时间戳（UTC），精确到毫秒。
muteStatus	Enum	静默状态： ● Muted: 静默。 ● Normal: 正常。
alarmValue	String	告警时数值。仅有数值型指标对应的告警会有返回此项，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，显示告警中数据容量使用率，百分比，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，显示许可证可用容量使用率，百分比，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。
currentValue	String	当前数值。仅有数值型指标对应的告警会有返回此项, 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，显示告警中数据容量使用率，百分比，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，显示许可证可用容量使用率，百分比，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。
resolveTime	Long	告警解除时间。unix 时间戳（UTC），精确到毫秒。 Resolved 状态的告警会返回此字段。

resolveValue	String	解除告警时的数值。仅有数值型指标对应的告警会有返回此项： ● AlarmNumberApproachingLimit: 告警中的告警条数接近上限，显示告警中数据容量使用率，百分比，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，显示许可证可用容量使用率，百分比，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。

resolveType	Enum	告警解除方式： ● Auto: 自动解除。 ● Manual: 手动解除。
expireTime	Long	告警失效时间。 Expired 状态的告警会返回此字段。
reason	String	告警解除原因或告警失效原因。
muteDueTime	Long	静默截止时间，unix 时间戳（UTC），精确到毫秒。 如果多次静默，返回最后一次操作的静默截止时间。
muteOperations	Array of muteOperation	静默操作，详见“表 2 响应参数 muteOperation 说明”。

表 2 响应参数 muteOperation 说明

名称	类型	描述
operTime	Long	静默操作时间。
operType	Enum	静默操作类型： ● Mute: 静默。 ● AutoUnmute: 自动解除静默。 ● ManualUnmute: 手动解除静默。
reason	String	静默/解除静默的原因。
dueTime	Long	静默截止时间。

请求示例 1

查询告警中的告警信息。

```
GET /rest/v1/system/alarm?alarmStatus=Unresolved HTTP/1.1
Date: Mon, 8 Jan 2024 03:26:58 GMT
Host: 192.168.0.110:1443
Authorization: HBlock userName:signature
```

响应示例 1

```
HTTP/1.1 200 OK
Date: Mon, 8 Jan 2024 03:26:58 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 1448
Connection: keep-alive
x-hblock-request-id: 0ba4372189a8470ca43fb8bea7959bf5
Server: HBlock

{
  "data": {
    "totalCount": 5,
    "alarmList": [
      {
        "alarmId": "4jlSnwld",
        "instanceId": "hblock_3:/mnt/stor",
        "instanceSnapshot": "hblock_3:/mnt/stor",
        "alarmRule": "CapacityQuotaUsageExceedsThreshold",
        "severity": "Warning",
        "alarmStatus": "Unresolved",
        "duration": 346407,
        "alarmTime": 1704684071663,
        "alarmValue": "93.69",
        "currentValue": "92.63",
        "muteStatus": "Normal"
      },
      {
        "alarmId": "4jlSnJ9h",
        "instanceId": "hblock_3",
        "instanceSnapshot": "hblock_3",
        "alarmRule": "CapacityQuotaUsageExceedsThreshold",
        "severity": "Warning",
        "alarmStatus": "Unresolved",
        "duration": 346861,
        "alarmTime": 1704684071209,
```



```
        "alarmValue": "93.69",
        "currentValue": "92.63",
        "muteStatus": "Normal"
    },
    {
        "alarmId": "4jlSmXuZ",
        "instanceId": "hblock_3:/mnt/stor",
        "instanceSnapshot": "hblock_3,ecs-9689-0915140,192.168.0.102:/mnt/stor",
        "alarmRule": "FaultDomainWarning",
        "severity": "Warning",
        "alarmStatus": "Unresolved",
        "duration": 347413,
        "alarmTime": 1704684070657,
        "muteStatus": "Normal"
    },
    {
        "alarmId": "4jlSmPPo",
        "instanceId": "hblock_3:/mnt/stor",
        "instanceSnapshot": "hblock_3,ecs-9689-0915140,192.168.0.102:/mnt/stor",
        "alarmRule": "DataServiceHealthStatusWarning",
        "severity": "Warning",
        "alarmStatus": "Unresolved",
        "duration": 347608,
        "alarmTime": 1704684070462,
        "muteStatus": "Normal"
    },
    {
        "alarmId": "4jlhtUel",
        "instanceId": "hblock_4",
        "instanceSnapshot": "hblock_4,ecs-9689-0915141,192.168.0.202",
        "alarmRule": "ProtocolServiceAbnormal",
        "severity": "Major",
        "alarmStatus": "Unresolved",
```

```
        "duration": 2634530,  
        "alarmTime": 1704681783540,  
        "muteStatus": "Normal"  
    }  
]  
}  
}
```

请求示例 2

查询已解除的告警信息。

```
GET /rest/v1/system/alarm?alarmStatus=Resolved HTTP/1.1  
Date: Mon, 8 Jan 2024 05:34:23 GMT  
Host: 192.168.0.110:1443  
Authorization: HBlock userName:signature
```

响应示例 2

```
HTTP/1.1 200 OK  
Date: Mon, 8 Jan 2024 05:34:23 GMT  
Content-Type: application/json; charset=utf-8  
Content-Length: 3268  
Connection: keep-alive  
x-hblock-request-id: b47f7f2aae794363bc577eb304425122  
Server: HBlock  
  
{  
  "data": {  
    "totalCount": 10,  
    "alarmList": [  
      {  
        "alarmId": "4jlSnwld",  
        "instanceId": "hblock_3:/mnt/stor",  
        "instanceSnapshot": "hblock_3:/mnt/stor",  
        "alarmRule": "CapacityQuotaUsageExceedsThreshold",  
        "severity": "Warning",  
      }  
    ]  
  }  
}
```

```
    "alarmStatus": "Resolved",
    "duration": 603073,
    "alarmTime": 1704684071663,
    "alarmValue": "93.69",
    "resolveTime": 1704684674736,
    "resolveValue": "5.06",
    "resolveType": "Auto",
    "muteStatus": "Normal"
  },
  {
    "alarmId": "4jlSnJ9h",
    "instanceId": "hblock_3",
    "instanceSnapshot": "hblock_3",
    "alarmRule": "CapacityQuotaUsageExceedsThreshold",
    "severity": "Warning",
    "alarmStatus": "Resolved",
    "duration": 603023,
    "alarmTime": 1704684071209,
    "alarmValue": "93.69",
    "resolveTime": 1704684674232,
    "resolveValue": "5.06",
    "resolveType": "Auto",
    "muteStatus": "Normal"
  },
  {
    "alarmId": "4jlSmXuZ",
    "instanceId": "hblock_3:/mnt/stor",
    "instanceSnapshot": "hblock_3,ecs-9689-0915140,192.168.0.102:/mnt/stor",
    "alarmRule": "FaultDomainWarning",
    "severity": "Warning",
    "alarmStatus": "Resolved",
    "duration": 603430,
    "alarmTime": 1704684070657,
    "resolveTime": 1704684674087,
    "resolveType": "Auto",
```

```
    "muteStatus": "Normal"
  },
  {
    "alarmId": "4jlSmPPo",
    "instanceId": "hblock_3:/mnt/stor",
    "instanceSnapshot": "hblock_3,ecs-9689-0915140,192.168.0.102:/mnt/stor",
    "alarmRule": "DataServiceHealthStatusWarning",
    "severity": "Warning",
    "alarmStatus": "Resolved",
    "duration": 602999,
    "alarmTime": 1704684070462,
    "resolveTime": 1704684673461,
    "resolveType": "Auto",
    "muteStatus": "Normal"
  },
  {
    "alarmId": "4ilZH7wX",
    "instanceId": "hblock_1",
    "instanceSnapshot": "hblock_1,hblockserver,192.168.0.110",
    "alarmRule": "ProtocolServiceAbnormal",
    "severity": "Major",
    "alarmStatus": "Resolved",
    "duration": 230649424,
    "alarmTime": 1704446197387,
    "resolveTime": 1704676846811,
    "resolveType": "Auto",
    "muteStatus": "Normal"
  },
  {
    "alarmId": "4hDwJnaH",
    "instanceId": "hblock_3",
    "instanceSnapshot": "hblock_3,ecs-9689-0915140,192.168.0.102",
    "alarmRule": "ProtocolServiceAbnormal",
    "severity": "Major",
    "alarmStatus": "Resolved",
```

```
        "duration": 361874,
        "alarmTime": 1704275265260,
        "resolveTime": 1704275627134,
        "resolveType": "Auto",
        "muteStatus": "Normal"
    },
    {
        "alarmId": "4hD1BMWS",
        "instanceId": "hblock_4",
        "instanceSnapshot": "hblock_4,ecs-9689-0915141,192.168.0.202",
        "alarmRule": "ProtocolServiceAbnormal",
        "severity": "Major",
        "alarmStatus": "Resolved",
        "duration": 2291572,
        "alarmTime": 1704273335042,
        "resolveTime": 1704275626614,
        "resolveType": "Auto",
        "muteStatus": "Normal"
    },
    {
        "alarmId": "4hD2AgSt",
        "instanceId": "hblock_1:/mnt/stor",
        "instanceSnapshot": "hblock_1,hblockserver,192.168.0.110:/mnt/stor",
        "alarmRule": "FaultDomainError",
        "severity": "Major",
        "alarmStatus": "Resolved",
        "duration": 61433,
        "alarmTime": 1704273396083,
        "resolveTime": 1704273457516,
        "resolveType": "Auto",
        "muteStatus": "Normal"
    },
    {
        "alarmId": "4hD2zESY",
        "instanceId": "hblock_1:/mnt/stor",
```

```
        "instanceSnapshot": "hblock_1,hblockserver,192.168.0.110:/mnt/stor",
        "alarmRule": "DataServiceHealthStatusError",
        "severity": "Major",
        "alarmStatus": "Resolved",
        "duration": 61164,
        "alarmTime": 1704273395799,
        "resolveTime": 1704273456963,
        "resolveType": "Auto",
        "muteStatus": "Normal"
    },
    {
        "alarmId": "4hCtBCK5",
        "instanceId": "hblock_4:/mnt/storage01",
        "instanceSnapshot": "hblock_4,ecs-9689-0915141,192.168.0.202:/mnt/storage01",
        "alarmRule": "DataServiceHealthStatusError",
        "severity": "Major",
        "alarmStatus": "Resolved",
        "duration": 60395,
        "alarmTime": 1704271227603,
        "resolveTime": 1704271287998,
        "resolveType": "Auto",
        "muteStatus": "Normal"
    }
]
}
```

请求示例 3

查询告警中、告警级别为 Major 的告警信息。

```
GET /rest/v1/system/alarm?alarmStatus=Unresolved&filter=severity:Major HTTP/1.1
Date: Mon, 8 Jan 2024 05:38:29 GMT
Host: 192.168.0.1110:1443
Authorization: HBlock userName:signature
```

响应示例 3

```
HTTP/1.1 200 OK
Date: Mon, 8 Jan 2024 05:38:29 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 299
Connection: keep-alive
x-hblock-request-id: 988b145267ac41c886954b6f81598ec4
Server: HBlock

{
  "data": {
    "totalCount": 1,
    "alarmList": [
      {
        "alarmId": "4j1htUe1",
        "instanceId": "hblock_4",
        "instanceSnapshot": "hblock_4,ecs-9689-0915141,192.168.0.202",
        "alarmRule": "ProtocolServiceAbnormal",
        "severity": "Major",
        "alarmStatus": "Unresolved",
        "duration": 10526372,
        "alarmTime": 1704681783540,
        "muteStatus": "Normal"
      }
    ]
  }
}
```

4.15.2 查询指定的 HBlock 告警

此操作用来查询指定的 HBlock 告警信息。

请求语法

```
GET /rest/v1/system/alarm/alarmId HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

请求参数

名称	类型	描述	是否必须
alarmId	String	告警 ID。 仅 Unresolved 状态的告警支持单条查询。	是

响应结果

名称	类型	描述
alarmId	String	告警 ID。
instanceId	String	告警实例 ID。
instanceSnapshot	String	告警实例快照，即告警发生时告警实例的详细信息。
alarmRule	String	告警规则。
severity	Enum	告警级别： ● Warning: 警告。 ● Major: 重要。 ● Critical: 严重。
alarmStatus	Enum	告警状态： ● Unresolved: 告警中。

		● Resolved: 告警已解除。 ● Expired: 告警已失效。
duration	Long	告警持续时长。unix 时间戳（UTC），精确到毫秒。
alarmTime	Long	告警发生时间。unix 时间戳（UTC），精确到毫秒。
alarmValue	String	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，显示告警中数据容量使用率，百分比，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，显示许可证可用容量使用率，百分比，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。
currentValue	String	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，显示告警中数据容量使用率，百分比，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，显示许可证可用容量使用率，百分比，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。
muteStatus	Enum	静默状态： ● Muted: 静默。 ● Normal: 正常。
muteDueTime	Long	静默截止时间，unix 时间戳（UTC），精确到毫秒。 如果多次静默，返回最后一次操作的静默截止时间。
muteOperations	Array of muteOperation	静默操作，详见“表 1 响应参数 muteOperation 说明”。

表 1 响应参数 muteOperation 说明

名称	类型	描述
operTime	Long	静默操作时间。
operType	Enum	静默操作类型： ● Mute: 静默。 ● AutoUnmute: 自动解除静默。 ● ManualUnmute: 手动解除静默。
reason	String	静默/解除静默的原因。
dueTime	Long	静默截止时间。

请求示例

查询告警 ID 为 4jlhtUe1 的告警信息。

```
GET /rest/v1/system/alarm/4jlhtUe1 HTTP/1.1
```

```
Date: Mon, 8 Jan 2024 05:42:22 GMT
Host: 192.168.0.110:1443
Authorization: HBlock userName:signature
```

响应示例

```
HTTP/1.1 200 OK
Date: Mon, 8 Jan 2024 05:42:22 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 268
Connection: keep-alive
x-hblock-request-id: dea2f8c199624d33aed3c4daa4654c2a
Server: HBlock

{
  "data": {
    "alarmId": "4jlhtUe1",
    "instanceId": "hblock_4",
    "instanceSnapshot": "hblock_4,ecs-9689-0915141,192.168.0.202",
    "alarmRule": "ProtocolServiceAbnormal",
    "severity": "Major",
    "alarmStatus": "Unresolved",
    "duration": 10718615,
    "alarmTime": 1704681783540,
    "muteStatus": "Normal"
  }
}
```

4.15.3 手动解除 HBlock 告警

此操作用来手动解除指定的 HBlock 告警。

请求语法

```
PUT /rest/v1/system/alarm/alarmId/resolve HTTP/1.1
Date: date
Host: ip:port
Content-Type: text/plain
Content-Length: length
Authorization: authorization

{
  "reason": reason
}
```

请求参数

名称	类型	描述	是否必须
alarmId	String	告警 ID。	是
reason	String	解除告警的原因。 取值：1~50 位字符串。	是

请求示例

手动解除告警 1kyvM3zJ。

```
PUT /rest/v1/system/alarm/1kyvM3zJ/resolve HTTP/1.1
Date: Wed, 17 Aug 2022 06:29:20 GMT
Host: 192.168.0.121:1443
Content-Type: text/plain
Content-Length: 41
Authorization: HBlock userName:signature
```

```
{  
  "reason": "已申请新 license"  
}
```

响应示例

```
HTTP/1.1 204 No Content  
Date: Wed, 17 Aug 2022 06:29:21 GMT  
Connection: keep-alive  
x-hblock-request-id: 9fe8d7733bd34c40a8895d674162c072  
Server: HBlock
```

4.15.4 静默告警/解除告警静默

此操作用来静默警或者解除告警静默。

请求语法

```
PUT /rest/v1/system/alarm/alarmId/mute HTTP/1.1
```

Date: *date*

Host: *ip:port*

Content-Type: text/plain

Content-Length: *length*

Authorization: *authorization*

```
{
  "operType": operType,
  "dueTime": dueTime,
  "reason": reason
}
```

请求参数

名称	类型	描述	是否必须
alarmId	String	告警 ID。	是
operType	Enum	静默操作。 取值： ● Mute: 静默告警。 ● ManualUnmute: 解除告警静默。	是
dueTime	Long	静默截止时间。 取值: unix 时间戳（UTC），精确到毫秒。如果静默操作为 Mute ，此项必填。	否
reason	String	静默/解除静默的原因。	是

		取值：1~50 位字符串。	

请求示例 1

静默告警 1kUyXPFy。

```
PUT /rest/v1/system/alarm/1kUyXPFy/mute HTTP/1.1
```

```
Date: Thu, 18 Aug 2022 08:25:06 GMT
```

```
Host: 192.168.0.121:1443
```

```
Content-Type: text/plain
```

```
Content-Length: 90
```

```
Authorization: HBlock userName:signature
```

```
{  
  "operType": "Mute",  
  "dueTime": 1660903200000,  
  "reason": "renew license"  
}
```

响应示例 1

```
HTTP/1.1 204 No Content
```

```
Date: Thu, 18 Aug 2022 08:25:06 GMT
```

```
Connection: keep-alive
```

```
x-hblock-request-id: 40d4c09de42d4d4cbf2f63e42d929208
```

```
Server: HBlock
```

请求示例 2

解除告警 1kUyXPFy 的静默。

```
PUT /rest/v1/system/alarm/1kUyXPFy/mute HTTP/1.1
```

```
Date: Thu, 18 Aug 2022 08:35:55 GMT
```

```
Host: 192.168.0.121:1443
```

```
Content-Type: text/plain
```

```
Content-Length: 81
```

```
Authorization: HBlock userName:signature
```



```
{  
  "operType": "ManualUnmute",  
  "reason": "license is about to expire."  
}
```

响应示例 2

```
HTTP/1.1 204 No Content  
Date: Thu, 18 Aug 2022 08:35:55 GMT  
Connection: keep-alive  
x-hblock-request-id: 40d4c09de42d4d4cbf2f63e42d929208  
Server: HBlock
```

4.15.5 导出告警

此操作用来导出告警信息。

请求语法

```
GET /rest/v1/system/alarm/file?alarmStatus=alarmStatus HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

请求参数

名称	类型	描述	是否必须
alarmStatus	String	告警状态。 取值： ● Unresolved: 告警中。 ● Resolved: 告警已解除。 ● Expired: 告警已失效。 默认值为 Unresolved。	否

请求示例

导出告警中的告警。

```
GET /rest/v1/system/alarm/file HTTP/1.1
Date: Mon, 08 Jan 2024 05:47:24 GMT
Host: 192.168.0.110:1443
Authorization: HBlock userName:signature
```

响应示例

```
HTTP/1.1 204 No Content
Date: Mon, 08 Jan 2024 05:47:24 GMT
```

```
Connection: keep-alive
x-hblock-request-id: 1c631d8b1b3549d7b4959d2bfae5f13b
Content-Disposition: attachment;filename=alarm_Unresolved_20220817160233.csv
Transfer-Encoding: chunked
Server: HBlock

alarmId,instanceId,instanceSnapshot,alarmRule,severity,alarmStatus,alarmTime,alarmValue
,muteStatus,muteOperations
4jlhtUe1,hblock_4,"hblock_4,ecs-9689-
0915141,192.168.0.202",ProtocolServiceAbnormal,Major,Unresolved,2024-01-08
10:43:03,,Normal,
```

4.15.6 错误码列表

HTTP status	错误码	错误信息	描述
400	InvalidArgumentLength	Value at 'reason' failed to satisfy constraint: Argument must not exceed 50 characters.	参数 reason 的长度不可超过 50 字符。
400	InvalidEnumValue	Value <i>value</i> at 'argument' failed to satisfy constraint: Argument must satisfy enum value set: [<i>value1</i> , <i>value2</i> ...]	枚举参数不合法。
400	InvalidFilterEnum	Value <i>value</i> at 'filter' failed to satisfy constraint: Argument can only contain: [instanceSnapshot, alarmRule, severity, muteStatus, alarmTime, resolveTime, resolveType, expireTime].	filter 参数不正确。
400	MissingArgument	Value null at 'reason' failed to satisfy constraint: Argument must not be null.	静默与手动解除静默时，reason 字段必填。
400	MissingArgument	Value null at 'dueTime' failed to satisfy constraint: Argument must not be null.	静默操作时，dueTime 字段必填。
400	InvalidTimeFormat	Value <i>value</i> at 'argument' failed to satisfy constraint: Argument must conform to <i>Format</i> format.	时间格式不合法。
400	NoSuchUnresolvedAlarm	The alarm with ID <i>ID</i> does not exist in 'Unresolved' alarm records.	指定告警 ID 的告警在告警记录中不存在。

4.16 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。

4.16.1 查看事件

此操作用来查看 HBlock 事件信息。

请求语法

```
GET /rest/v1/system/event?number=number&type=type HTTP/1.1  
Date: date  
Host: ip:port  
Authorization: authorization
```

请求参数

参数	类型	描述	是否必须
number	Integer	指定最近事件的查询个数。 取值：[1, 1000]，默认值为 1000。	否
type	Enum	指定事件的类型。 取值： ● user：用户事件。 ● system：系统事件。 默认值为 user 。	否

响应结果

名称	类型	描述
events	Array of event	事件信息集合，详见“表 1 响应参数 event 说明”。

表 1 响应参数 event 说明

名称	类型	描述
eventId	String	事件 ID。
module	String	事件所属模块。
eventTime	Long	对于用户事件：HBlock 接收到事件请求的时间，unix 时间戳（UTC），精确到毫秒。 对于系统事件：事件发生时间，unix 时间戳（UTC），精确到毫秒。
name	String	事件名称。 说明： 事件名称描述详见用户事件列表和系统事件列表。
requestId	String	用户事件请求 ID。
requesterIP	String	用户事件中，发起请求的源 IP 地址。
statusCode	Integer	用户事件响应状态码。
errorCode	String	用户事件错误码。 -：表示无错误码。
errorMessage	String	用户事件错误信息。 -：表示无错误信息。
detail.method	String	用户事件的请求方法。
detail.url	String	用户事件的 URL。
detail.body	String	事件的请求体。
instanceId	String	系统事件针对的实例 ID。 如果没有实例 ID，则不返回此项。

请求示例 1

查询 HBlock 用户事件。

```
GET /rest/v1/system/event?number=2 HTTP/1.1
```

```
Date: Thu, 04 Aug 2022 06:12:33 GMT
```

```
Authorization: HBlock userName:signature  
Host:192.168.0.121:1443
```

响应示例 1

```
HTTP/1.1 200 OK  
x-hblock-request-id: d701da5a23c94eef8a8efe0907b12751  
Connection: keep-alive  
Content-Length: 853  
Date: Thu, 04 Aug 2022 06:12:33 GMT  
Content-Type: application/json; charset=utf-8  
Server: HBlock  
  
{  
  "data": {  
    "events": [  
      {  
        "eventId": "78f85f5e-e8ca-42b9-889d-834a167283cc",  
        "module": "System",  
        "eventTime": 1659429529758,  
        "name": "StartLogCollect",  
        "requestId": "f630a474bc7e4d9eba75f3b9d28e5d51",  
        "requesterIP": "36.111.88.33",  
        "statusCode": 202,  
        "errorCode": "-",  
        "errorMessage": "-",  
        "detail": {  
          "method": "POST",  
          "url": "/rest/v1/system/logcollect",  
          "body":  
            "{ \"startTime\":1659422322714, \"endTime\":1659429522714, \"servers\":[\"hblock_1\", \"hblock_2\", \"hblock_3\"], \"logTypes\":[\"Config\", \"System\", \"Data\", \"Coordination\"] }"  
        }  
      },  
      {  
        "eventId": "37fd582e-6d36-4174-a9bd-94c825316f8a",
```

```
{
  "module": "System",
  "eventTime": 1659429493016,
  "name": "Login",
  "requestId": "ab3c00578efb4372b1fd622d827ad97c",
  "requesterIP": "36.111.88.33",
  "statusCode": 200,
  "errorCode": "-",
  "errorMessage": "-",
  "detail": {
    "method": "POST",
    "url": "/internal/v1/system/user/login",
    "body": "{\"userName\":\"storuser\"}"
  }
}
```

请求示例 2

查询 HBlock 系统事件。

```
GET /rest/v1/system/event?number=2&type=system HTTP/1.1
Date: Thu, 4 Aug 2022 06:27:30 GMT
Authorization: HBlock userName:signature
Host:192.168.0.121:1443
```

响应示例 2

```
HTTP/1.1 200 OK
x-hblock-request-id: 9199a40fdf4c48c58ec5a4f7a1acda95
Connection: keep-alive
Content-Length: 454
Date: Thu, 4 Aug 2022 06:27:30 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock
```



```
{
  "data": {
    "events": [
      {
        "eventId": "19cb57dd-5210-45af-a930-3d4cc70fe299",
        "module": "System",
        "eventTime": 1659343497897,
        "name": "DataResumed"
      },
      {
        "eventId": "1843075f-1281-4abc-af22-05a95c3a00ac",
        "module": "System",
        "eventTime": 1659343497897,
        "name": "DataLowRedundancy",
        "detail": {
          "body": "{\"percent\": \"%\"}"
        }
      },
      {
        "eventId": "6b6d9ec4-4a20-44d2-b00c-f3f154373a11",
        "module": "Disk",
        "eventTime": 1659343495978,
        "name": "PathRemoved",
        "instanceId": "hblock_2:/mnt/storage02"
      }
    ]
  }
}
```

4.16.2 导出事件

此操作用来导出 HBlock 事件信息。

请求语法

```
GET /rest/v1/system/event/file?number=numebr&type=type HTTP/1.1
```

Date: *date*

Host: *ip:port*

Authorization: *authorization*

请求参数

参数	类型	描述	是否必须
number	Integer	指定导出最近 HBlock 事件的个数。 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 HBlock 事件信息。 取值： [1, 10000]。	否
type	Enum	指定事件的类型。 取值： ● user： 用户事件。 ● system： 系统事件。 默认值为 user 。	否

响应头

名称	类型	描述
Content-Disposition	String	响应内容的显示形式。
Transfer-Encoding	String	实体传输给用户的编码形式。

请求示例 1

导出 HBlock 用户事件。

```
GET /rest/v1/system/event/file?number=3 HTTP/1.1
Date: Mon, 21 Mar 2022 01:53:48 GMT
Authorization: HBlock userName:signature
Host:192.168.0.121:1443
```

响应示例 1

```
HTTP/1.1 200 OK
x-hblock-request-id: 9fd643aa73bd4b60ab59a29e2904ab49
Connection: keep-alive
Date: Mon, 21 Mar 2022 01:53:48 GMT
Content-Disposition: attachment;filename=events_20220321095348.csv
Transfer-Encoding: chunked
Server: HBlock

eventId,eventTime,module,name,requestId,requesterIP,method,URL,body,statusCode,errorCod
e,errorMessage
a3830830-7f8a-40b2-a764-8ebfa6083f33,2022-03-21
09:53:28,LUN,ExpandLUN,5929bcbd0cf54e7f8a50b41f8bdd549b,111.198.231.68,PUT,http://218.7
8.41.185:1443/rest/v1/block/lun/lunName1/expand,{"
  "capacity":200000
}
",400,LessThanCurrentCapacity,Value 200000 at 'capacity' failed to satisfy constraint:
Argument must be greater than 200000.
79a4c7b3-5b5c-46da-9725-f14237d462be,2022-03-21
09:53:21,LUN,SwitchLUN,a9be97449f434964960dbc8b48c05818,111.198.231.68,PUT,http://218.7
8.41.185:1443/rest/v1/block/lun/Lun1/switch,,200,-,-
9481763e-da71-4354-8a77-bff1c2d3effd,2022-03-21
09:53:09,LUN,SetLUN,9718a304ab1f4f03b9530ebeeef3b51fc,111.198.231.68,PUT,http://218.78.4
1.185:1443/rest/v1/block/lun/Lun1,{"
  "config":{
    "writePolicy":"WriteThrough"
  }
}
```

```
}  
",200,-,-
```

请求示例 2

导出 HBlock 系统事件。

```
GET /rest/v1/system/event/file?number=3&type=system HTTP/1.1  
Date: Thu, 4 Aug 2022 09:28:28 GMT  
Authorization: HBlock userName:signature  
Host:192.168.0.121:1443
```

响应示例 2

```
HTTP/1.1 200 OK  
x-hblock-request-id: 0de4cbbd38104d8aaeae7700e6af9bae  
Connection: keep-alive  
Date: Thu, 4 Aug 2022 09:28:28 GMT  
Content-Disposition: attachment;filename=events_20220321095348.csv  
Transfer-Encoding: chunked  
Server: HBlock  
  
eventId,eventTime,module,name,instanceId,details  
19cb57dd-5210-45af-a930-3d4cc70fe299,2022-08-01 16:44:57,System,DataResumed,null,{}  
1843075f-1281-4abc-af22-05a95c3a00ac,2022-08-01  
16:44:57,System,DataLowRedundancy,null,{"percent": "0%"}  
6b6d9ec4-4a20-44d2-b00c-f3f154373a11,2022-08-01  
16:44:55,Disk,PathRemoved,hblock_2:/mnt/storage02,{}  

```

4.16.3 发起 HBlock 日志采集

此操作用来发起 HBlock 日志采集。

注意：日志采集的进程不能超过 10 个。

请求语法

```
POST /rest/v1/system/logcollect HTTP/1.1
```

Date: *date*

Host: *ip:port*

Content-Length: *length*

Authorization: *authorization*

```
{
  "startTime": startTime,
  "endTime": endTime,
  "servers": [ server_ID ],
  "logTypes": [ logTypes ],
  "outputDirectory": outputDirectory
}
```

请求参数

参数	类型	描述	是否必须
startTime	String	HBlock 日志采集的起始时间。 取值：unix 时间戳（UTC），精确到毫秒。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 HBlocks 日志采集的结束时间。	否

		● 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。	
endTime	String	HBlock 日志采集的结束时间。 取值：unix 时间戳（UTC），精确到毫秒。默认值为当前时间。 注意： ● HBlock 日志采集的起始时间必须早于 HBlocks 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。	否
servers	Array of String	要采集 HBlock 日志的服务器 ID。 取值：可以填写多个服务器 ID，以英文逗号（,）隔开。默认采集所有服务器。	否
logTypes	Array of logType	采集的日志类型 取值： ● 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）。	否

		默认采集所有类型的日志。	
outputDirectory	String	HBlock 日志采集后存放的目录，为绝对路径。 取值：如果不指定，默认存放在被请求服务器的 HBlock 安装目录下。以 collected_logs/hblock_logs_id_yyyyMMddHHmmss_yyyyMMddHHmmss.zip 命名，其中： ● 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。	否

响应结果

名称	类型	描述
id	String	日志 ID。

请求示例 1

采集服务器 ID 为 hblock_1 和 hblock_2、时间为 2022-08-07 10:00:00 到 2022-08-07 22:00:00、配置和系统的相关的日志，并将日志文件存放在/mnt/storage01 下。

<pre> POST /rest/v1/system/logcollect HTTP/1.1 Date: Mon, 08 Aug 2022 05:57:49 GMT Authorization: HBlock userName:signature Content-Length: 192 Content-Type: text/plain Host:192.168.0.121:1443 { "startTime": "1659837600000", "endTime": "1659880800000", "servers": ["hblock_1","hblock_2"], "logTypes": ["Config","System"], "outputDirectory": "/mnt/storage01" </pre>

```
}
```

响应示例 1

```
HTTP/1.1 202 Accepted
x-hblock-request-id: 68a53a147d504d5087841192adef799e
Connection: keep-alive
Content-Length: 50
Date: Mon, 8 Aug 2022 05:57:52 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "id": "7889764ce31444d2b06f8c2b8201c836"
  }
}
```

请求示例 2

采集 HBlock 最近两个小时所有服务器上所有类型的日志，并将日志文件存放在 /mnt/storage01 下。

```
POST /rest/v1/system/logcollect HTTP/1.1
Date: Mon, 08 Aug 2022 06:18:12 GMT
Authorization: HBlock userName:signature
Content-Length: 45
Content-Type: text/plain
Host:192.168.0.121:1443

{
  "outputDirectory": "/mnt/storage01"
}
```

响应示例 2

```
HTTP/1.1 202 Accepted
```


x-hblock-request-id: f9cbe1e2e7564f00becc8903aaa7c9e6

Connection: keep-alive

Content-Length: 50

Date: Mon, 8 Aug 2022 06:18:14 GMT

Content-Type: application/json; charset=utf-8

Server: HBlock

```
{
  "data": {
    "id": "8c042a768cc74b15b74669d7ebec683c"
  }
}
```

4.16.4 批量查询采集的日志

此操作用来批量查询采集的日志。

请求语法

```
GET /rest/v1/system/logcollect?filter=filter HTTP/1.1
```

Date: *date*

Host: *ip:port*

Authorization: *authorization*

请求参数

参数	类型	描述	是否必须
filter	String	查询的过滤条件。 过滤条件由查询属性和属性值组成，其中属性和属性值之间用单个冒号分隔（key:value）表示模糊匹配，用两个冒号分隔（key::value）表示精确匹配。可以选择多个查询，如果是或的关系，使用“or”将查询条件分隔开；如果是与的关系，使用“and”将查询条件分隔开。 支持的查询条件为： ● 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。 ● logTypes: 采集日志的类型 ■ Config: 配置相关的日志。 ■ System: 系统相关的日志。	否

		■ 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）。	

响应结果

名称	类型	描述
logs	Array of log	日志列表信息，详见“表 1 响应参数 log 说明”。
message	String	收集失败或者部分收集成功的原因，如果收集成功，则不会返回此项。

表 1 响应参数 log 说明

名称	类型	描述
id	String	日志 ID。
startTime	Long	HBlock 日志采集的起始时间。
endTime	Long	HBlock 日志采集的结束时间。
servers	Array of String	HBlock 日志采集的服务器 ID。
logTypes	Array of logType	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。 ● Coordination: 内部协调服务相关的日志（仅集群版支持）。
file	String	日志文件存储路径及名称。
status	Enum	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。
size	Long	采集到的日志文件大小，单位是 bytes。
createTime	Long	日志采集请求的创建时间。

请求示例 1

查询所有的采集日志。

```
GET /rest/v1/system/logcollect HTTP/1.1
Date: Wed, 7 Aug 2024 05:48:56 GMT
Authorization: HBlock userName:signature
Host:192.168.0.121:1443
```

响应示例 1

```
HTTP/1.1 200 OK
x-hblock-request-id: ae3b8ec3785e4792bf8aeddd1ed3ab26
Connection: keep-alive
Content-Length: 772
Date: Wed, 7 Aug 2024 05:48:56 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "logs": [
      {
        "id": "ea35db6a939d40b9825c470a9f8826bb",
        "startTime": 1723002728561,
        "endTime": 1723009928561,
        "servers": [
          "hblock_1"
        ],
        "logTypes": [
          "System"
        ],
      }
    ]
  }
}
```

```
      "file":
        "192.168.0.192:/mnt/storage01/CTYUN_HBlock_Plus_3.10.0_x64/collected_logs/hblock_logs_ea35db6a939d40b9825c470a9f8826bb_20240807035208_20240807055208.zip",
        "status": "Succeeded",
        "size": 27607851,
        "createTime": 1723009904028
      },
      {
        "id": "54dfd39212fc476dba61f761c08d8a8a",
        "startTime": 1722915993000,
        "endTime": 1723009593754,
        "servers": [
          "hblock_1",
          "hblock_2",
          "hblock_3",
          "hblock_4"
        ],
        "logTypes": [
          "Config"
        ],
        "file":
          "192.168.0.192:/mnt/storage01/CTYUN_HBlock_Plus_3.5.0_x64/collected_logs/hblock_logs_54dfd39212fc476dba61f761c08d8a8a_20240806034633_20240807054633.zip",
          "status": "Succeeded",
          "size": 104534,
          "createTime": 1723009566777
        }
      ]
    }
  }
}
```

请求示例 2

查询日志类型为 Config 的日志。

```
GET /rest/v1/system/logcollect?filter=logTypes:Config HTTP/1.1
```

```
Date: Wed, 7 Aug 2024 05:52:13 GMT
Authorization: HBlock userName:signature
Host:192.168.0.121:1443
```

响应示例 2

```
HTTP/1.1 200 OK
x-hblock-request-id: 3906705dd9e946ffb640c1b97649bee8
Connection: keep-alive
Content-Length: 411
Date: Wed, 7 Aug 2024 05:52:13 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "logs": [
      {
        "id": "54dfd39212fc476dba61f761c08d8a8a",
        "startTime": 1722915993000,
        "endTime": 1723009593754,
        "servers": [
          "hblock_1",
          "hblock_2",
          "hblock_3",
          "hblock_4"
        ],
        "logTypes": [
          "Config"
        ],
        "file":
"192.168.0.192:/mnt/storage01/CTYUN_HBlock_Plus_3.5.0_x64/collected_logs/hblock_logs_54
dfd39212fc476dba61f761c08d8a8a_20240806034633_20240807054633.zip",
        "status": "Succeeded",
        "size": 104534,
        "createTime": 1723009566777
      }
    ]
  }
}
```

```
}  
  ]  
}  
{
```

4.16.5 查询单个采集的日志

此操作用来查询单个采集的日志。

请求语法

```
GET /rest/v1/system/logcollect/id HTTP/1.1
```

Date: *date*

Host: *ip:port*

Authorization: *authorization*

请求参数

参数	类型	描述	是否必须
id	String	日志 ID。 如果不填写，查询所有的采集日志。	否

响应结果

名称	类型	描述
id	String	日志 ID。
startTime	Long	HBlock 日志采集的起始时间。
endTime	Long	HBlock 日志采集的结束时间。
servers	Array of string	日志采集的服务器 ID。
logTypes	Array of logType	采集的日志类型： ● 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）。

file	String	日志文件存储路径及名称。
status	Enum	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。
size	Long	采集到的日志文件大小，单位是 bytes。
createTime	Long	日志采集请求的创建时间。
message	String	收集失败或者部分收集成功的原因，如果收集成功，则不会返回此项。

请求示例

查询 ID 为 54dfd39212fc476dba61f761c08d8a8a 的日志。

```
GET /rest/v1/system/logcollect/5f84a6ea8edf4f9ea1debe601ce15c22 HTTP/1.1
Date: Wed, 07 Aug 2024 05:57:41 GMT
Authorization: HBlock userName:signature
Host:192.168.0.121:1443
```

响应示例

```
HTTP/1.1 200 OK
x-hblock-request-id: 83e8981108e948719fb64f7a977fede5
Connection: keep-alive
Content-Length: 400
Date: Wed, 07 Aug 2024 05:57:41 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "id": "54dfd39212fc476dba61f761c08d8a8a",
    "startTime": 1722915993000,
```

```
    "endTime": 1723009593754,
    "servers": [
      "hblock_1",
      "hblock_2",
      "hblock_3",
      "hblock_4"
    ],
    "logTypes": [
      "Config"
    ],
    "file":
"192.168.0.192:/mnt/storage01/CTYUN_HBlock_Plus_3.9.0_x64/collected_logs/hblock_logs_54
dfd39212fc476dba61f761c08d8a8a_20240806034633_20240807054633.zip",
    "status": "Succeeded",
    "size": 104534,
    "createTime": 1723009566777
  }
}
```

4.16.6 下载采集的日志文件

此操作用来下载采集的日志文件。

注意：只有采集成功的日志和部分采集成功的日志才能被下载。

请求语法

```
GET /rest/v1/system/logcollect/id/file HTTP/1.1
```

Date: *date*

Host: *ip:port*

Authorization: *authorization*

请求参数

参数	类型	描述	是否必须
id	String	日志 ID。	是

请求示例

下载 ID 为 903fec5908534445b847a2b13bd205e2 的日志。

```
GET /rest/v1/system/logcollect/903fec5908534445b847a2b13bd205e2/file HTTP/1.1
```

Date: Mon, 05 Sep 2022 08:26:41 GMT

Authorization: HBlock *userName:signature*

Host:192.168.0.121:1443

响应示例

```
HTTP/1.1 200 OK
```

x-hblock-request-id: c0958bc7287c4ca4a515e9ac5e32863d

Connection: keep-alive

Content-Length: 28950

Date: Mon, 5 Sep 2022 08:26:42 GMT

Content-Disposition:

attachment;filename=hblock_logs_903fec5908534445b847a2b13bd205e2_20220905082042_20220905082242.zip

Content-Type: application/zip;charset=utf-8

Server: HBlock

4.16.7 删除指定的采集日志

此操作用来删除指定的采集日志。

请求语法

```
DELETE /rest/v1/system/logcollect/id HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

请求参数

参数	类型	描述	是否必须
id	String	日志 ID。 如果不填写，删除所有采集日志。	否

请求示例

删除 ID 为 5f84a6ea8edf4f9ea1debe601ce15c22 的日志。

```
DELETE /rest/v1/system/logcollect/5f84a6ea8edf4f9ea1debe601ce15c22 HTTP/1.1
Date: Mon, 08 Aug 2022 08:55:18 GMT
Authorization: HBlock userName:signature
Host:192.168.0.121:1443
```

响应示例

```
HTTP/1.1 204 No Content
x-hblock-request-id: 9bed087398184a77b7ef63eabb0670d3
Connection: keep-alive
Date: Mon, 8 Aug 2022 08:55:20 GMT
Server: HBlock
```

4.16.8 删除所有的采集日志

此操作用来删除所有的采集日志。

请求语法

```
DELETE /rest/v1/system/logcollect HTTP/1.1  
Date: date  
Host: ip:port  
Authorization: authorization
```

请求示例

删除所有的采集日志。

```
DELETE /rest/v1/system/logcollect HTTP/1.1  
Date: Mon, 08 Aug 2022 09:01:57 GMT  
Authorization: HBlock userName:signature  
Host:192.168.0.121:1443
```

响应示例

```
HTTP/1.1 204 No Content  
x-hblock-request-id: d5e2973621064ed7b2e846b3874535d3  
Connection: keep-alive  
Date: Mon, 8 Aug 2022 09:02:00 GMT  
Server: HBlock
```

4.16.9 错误码列表

HTTP status	错误码	错误信息	描述
400	CanNotConnectToServer	Can not connect to the server <i>IP[:port]</i> .	无法连接服务器。
400	ExceedMaxValue	Value <i>value</i> at 'number' failed to satisfy constraint: Argument must be less than or equal to <i>value</i> .	超过最大值。
400	ExceedThreshold	The number of <i>operation</i> cannot exceed value.	超出了系统允许的最大个数，无法执行该操作。
400	InsufficientServerSpace	This method is not allowed because the free space of <i>directory</i> must be greater than or equal to <i>value</i> for server <i>serverId[, serverId...]</i> .	存储空间不足。
400	InvalidDirectory	The <i>argument</i> must be a directory specified with an absolute path.	输出路径必须是带有绝对路径的目录。
400	InvalidEnumValue	Value <i>value</i> at ' <i>argument</i> ' failed to satisfy constraint: Argument must satisfy enum value set: [<i>value1</i> , <i>value2</i> ...]	枚举参数不合法。
400	InvalidPositiveInteger	Value <i>value</i> at ' <i>number</i> ' failed to satisfy constraint: Argument	取值必须为正整数。

		must be positive integer.	
400	InvalidTime	The end time must be later than the start time.	结束时间必须晚于开始时间。
400	InvalidTimeFormat	Value <i>value</i> at ' <i>argument</i> ' failed to satisfy constraint: Argument must conform to <i>Format</i> format.	时间格式不合法。
400	StandaloneModeNotAllowed	' <i>operation</i> ' is not supported by standalone mode of HBlock.	单机版本 HBlock 不支持该操作。
400	TooLargeFile	The file is too large. It is recommended to narrow the query range.	日志文件过大，建议缩小查询范围。
403	PermissionDenied	Failed to action because user has no permission.	权限不足，无法执行该操作。
404	NoSuchLog	The log with ID <i>ID</i> does not exist.	日志文件不存在。
409	InvalidLogStatus	The status of log collection task with ID <i>ID</i> is ' <i>status</i> ', the request is invalid.	日志状态冲突，无法执行该操作。

4.17 HBlock 系统设置

4.17.1 修改管理员密码

此操作用来修改管理员密码。

请求语法

```
PUT /rest/v1/system/user/password HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "userName": userName,
  "newPassword": newPassword
}
```

请求参数

名称	类型	描述	是否必须
userName	String	HBlock 的管理员用户名。 取值：与初始化设置的用户名保持一致。	是
newPassword	String	用户设置的新密码的明文。 取值：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）。	是

请求示例

修改管理员密码。

```
PUT /rest/v1/system/user/password HTTP/1.1
Date: Wed, 24 Jan 2024 07:54:58 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 67
Host:192.168.0.110:1443
```

```
{
  "userName": "storuser",
  "newPassword": "*****"
}
```

响应示例

```
HTTP/1.1 204 No Content
x-hblock-request-id: 914ac22b6fb74f5ba5dc085e35718294
Connection: keep-alive
Date: Wed, 24 Jan 2024 07:54:58 GMT
Server: HBlock
```

4.17.2 邮件设置

4.17.2.1 设置邮件

此操作用来设置或者修改邮件配置信息。

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。

请求语法

```
PUT /rest/v1/system/config/notification HTTP/1.1
```

Date: *date*

Content-Type: application/json; charset=utf-8

Content-Length: *length*

Host: *ip:port*

Authorization: *authorization*

```
{
  "status": status,
  "smtpHost": smtpHost,
  "smtpPort": smtpPort,
  "SSL": ssl,
  "senderEmail": senderEmail,
  "password": password,
  "receiverEmail": receiverEmail
}
```

请求参数

参数	类型	描述	是否必须
status	String	是否启用邮件通知功能。 取值： ● Enabled: 启用。	是

		● Disabled: 禁用。	
smtpHost	String	SMTP 服务器。 取值：SMTP 服务器域名或 IP。 注意： 如果启用邮件通知时，未在服务端保存过邮件配置，此项必填。	否
smtpPort	Integer	SMTP 端口号。 取值：[1,65535]。 说明： ● 如果启用了 SSL，默认端口为 465。 ● 如果禁用 SSL，默认端口为 25。	否
SSL	String	是否启用 SSL。 取值： ● Enabled: 启用 SSL。 ● Disabled: 禁用 SSL。 默认值为 Disabled 。	否
senderEmail	String	发件箱。 邮箱格式 <i>local-part@domain</i> : ● local-part: 字符串形式，长度 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 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。	否

		■ 顶级域名不能是纯数字。 ■ 短横线 (-) 不能作为首尾字符。 注意：如果启用邮件通知时，未在服务端保存过邮件配置，此项必填。	
password	String	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。 注意：如果启用邮件通知时，未在服务端保存过邮件配置，此项必填。	否
receiverEmail	String	收件箱。可以填写多个收件箱。 邮箱格式 <i>local-part@domain</i>： ● <i>local-part</i>：字符串形式，长度 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。 ● <i>domain</i>：以句点 (.) 分隔的字符串形式，长度 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。 注意：如果启用邮件通知时，未在服务端保存过邮件配置，此项必填。	否

请求示例 1

设置邮件配置信息，并启用邮件通知功能。

```
PUT /rest/v1/system/config/notification HTTP/1.1
```

```
Date: Mon, 1 Aug 2022 01:53:36 GMT
```

```
Content-Type: application/json; charset=utf-8
```

```
Authorization: HBlock userName:signature
```

```
Content-Length: 243
```

```
Host: 192.168.0.121:1443
```

```
{  
  "status": "Enabled",  
  "smtpHost": "smtp.ctyun.cn",  
  "smtpPort": 475,  
  "SSL": "Enabled",  
  "senderEmail": "account1@ctyun.cn",  
  "password": "*****",  
  "receiverEmail": "account2@ctyun.cn"  
}
```

响应示例 1

```
HTTP/1.1 204 No Content
```

```
x-hblock-request-id: 24c17819e3ed4c648f37d13f9af05537
```

```
Connection: keep-alive
```

```
Date: Mon, 1 Aug 2022 01:53:36 GMT
```

```
Server: HBlock
```

请求示例 2

关闭邮件通知功能，但是不删除已经配置的 smtpHost 等信息。

```
PUT /rest/v1/system/config/notification HTTP/1.1
```

```
Date: Mon, 01 Aug 2022 01:56:05 GMT
```

```
Content-Type: application/json; charset=utf-8
```

```
Content-Length: 30
```

```
Host: 192.168.0.121:1443
```

```
Authorization: HBlock userName:signature
```

```
{  
  "status": "Disabled"  
}
```

响应示例 2

```
HTTP/1.1 204 No Content  
x-hblock-request-id: 9022434a208a4c2ebf89da97f3f72976  
Connection: keep-alive  
Date: Mon, 01 Aug 2022 01:56:05 GMT  
Server: HBlock
```

4.17.2.2 发送测试邮件

此操作用来发送测试邮件。

请求语法

```
POST /rest/v1/system/config/notification/testEmail HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

请求示例

发送测试邮件。

```
POST /rest/v1/system/config/notification/testEmail HTTP/1.1
Date: Thu, 17 Mar 2022 05:39:56 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 0
Host: 192.168.0.121:1443
```

响应示例

```
HTTP/1.1 200 OK
x-hblock-request-id: 3de55df515314e4cb9b7b61ade66937f
Connection: keep-alive
Date: Thu, 17 Mar 2022 05:40:01 GMT
Content-Length: 0
Server: HBlock
```


4.17.2.3 查询邮件配置信息

此操作用来查询邮件配置信息。

请求语法

```
GET /rest/v1/system/config/notification HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

响应结果

名称	类型	描述
status	String	是否启用邮件通知功能： ● Enabled: 启用。 ● Disabled: 禁用。
smtpHost	String	SMTP 服务器。
smtpPort	Integer	SMTP 服务器端口号。
SSL	String	是否启用 SSL： ● Enabled: 启用。 ● Disabled: 禁用。
senderEmail	String	发件箱。
receiverEmail	String	收件箱。

请求示例

查询邮件配置信息。

```
GET /rest/v1/system/config/notification HTTP/1.1
Date: Thu, 17 Mar 2022 05:46:57 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.121:1443
```

响应示例

```
HTTP/1.1 200 OK
x-hblock-request-id: 4b5e0623e73a4dcd8754fd2ed504b9c1
Connection: keep-alive
Content-Length: 199
Date: Thu, 17 Mar 2022 05:46:57 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "status": "Disabled",
    "smtpHost": "smtp.ctyun.cn",
    "smtpPort": 475,
    "SSL": "Enabled",
    "senderEmail": "account1@ctyun.cn",
    "receiverEmail": "account2@ctyun.cn"
  }
}
```

4.17.2.4 删除邮件配置

此操作用来删除邮件配置。

请求语法

```
DELETE /rest/v1/system/config/notification HTTP/1.1  
Date: date  
Host: ip:port  
Authorization: authorization
```

请求示例

删除邮件配置。

```
DELETE /rest/v1/system/config/notification HTTP/1.1  
Date: Thu, 17 Mar 2022 05:56:29 GMT  
Authorization: HBlock userName:signature  
Host: 192.168.0.121:1443
```

响应示例

```
HTTP/1.1 204 No Content  
x-hblock-request-id: 9639c0c89dca4fcb83719458a50452ac  
Connection: keep-alive  
Date: Thu, 17 Mar 2022 05:56:29 GMT  
Server: HBlock
```

4.17.2.5 错误码列表

HTTP status	错误码	错误信息	描述
400	InvalidReceiverEmail	Value <i>value</i> at 'receiver email' failed to satisfy constraint: Argument must conform to email format, and be separated by comma if there are multiple receivers.	接收者邮箱不合法。
400	InvalidSenderEmail	Value <i>value</i> at 'sender email' failed to satisfy constraint: Argument must conform to email format, and only one sender can be set.	发送者邮箱不合法。
400	MissingReceiverEmail	Value null at 'receiver email' failed to satisfy constraint: Argument must not be null when enable email configuration.	启用邮件设置时，receiverEmail 不能为空。
400	MissingSenderEmail	Value null at 'sender email' failed to satisfy constraint: Argument must not be null when enable email configuration.	启用邮件设置时，senderEmail 不能为空。
400	MissingSenderPassword	Value null at 'password' failed to satisfy constraint: Argument must not be null when enable email configuration.	启用邮件设置时，password 不能为空。
400	MissingSMTPHost	Value null at 'smtp host' failed to	启用邮件设置

		satisfy constraint: Argument must not be null when enable email configuration.	时，SMTP 服务器不能为空。
400	SendEmailFailed	Test email failed to send out.	发送测试邮件失败。
404	NoSuchNotificationConfiguration	Notification is not configured.	没有配置邮件通知功能。

4.17.3 远程协助

4.17.3.1 设置远程协助

此操作用来设置远程协助。

请求语法

```
PUT /rest/v1/system/config/remoteAccess HTTP/1.1
```

Date: *date*

Content-Type: application/json; charset=utf-8

Content-Length: *length*

Host: *ip:port*

Authorization: *authorization*

```
{
  "serverId": serverId,
  "status": status,
  "host": host,
  "port": port
}
```

请求参数

名称	类型	描述	是否必须
serverId	String	服务器 ID。 如果不指定，默认设置当前被请求的服务器。	否
status	String	是否启用远程协助。 取值： ● Enabled: 启用。 ● Disabled: 禁用。	是
host	String	远程协助的服务端 IP 或域名。 说明：请联系软件供应商获取 host 信息。	否

		除下列情况 host 必填外，其他情况选填： ● 如果启用远程协助时，此项在服务端没有保存过。 ● 如果设置远程协助时，输入了具体的端口号。	
port	Integer	远程协助的端口号。 说明： 请联系软件供应商获取端口号。 取值： [1,65535]，默认 18100。	否

响应结果

名称	类型	描述
code	Integer	远程协助码，6 位数字。
host	String	远程协助的服务端 IP。
port	Integer	远程协助端口号。

请求示例

设置远程协助。

<pre> PUT /rest/v1/system/config/remoteAccess HTTP/1.1 Date: Thu, 17 Mar 2022 09:20:16 GMT Content-Type: application/json; charset=utf-8 Content-Length: 113 Host: 192.168.0.121:1443 Authorization: HBlock <i>userName:signature</i> { "serverId": "hblock_1", "status": "Enabled", "host": "182.151.21.174", "port": 18100 } </pre>

响应示例

```
HTTP/1.1 200 OK
Date: Thu, 17 Mar 2022 09:20:16 GMT
Content-Length: 63
Connection:keep-alive
Content-Type: application/json; charset=utf-8
x-hblock-request-id: a839f9f8e76a4f18969f7e2637c957eb
Server: HBlock

{
  "data": {
    "code": "999342",
    "host": "182.151.21.174",
    "port": 18100
  }
}
```


4.17.3.2 查询远程协助

此操作用来查询服务器的远程协助配置信息。

注意：

- 如果 HBlock 还没有初始化，那么只显示本机的信息。
- 仅支持查询有远程协助配置的服务器。

请求语法

```
GET /rest/v1/system/config/remoteAccess?filter=filter&range=i-j HTTP/1.1  
Date: date  
Host: ip:port  
Authorization: authorization
```

请求参数

参数	类型	描述	是否必须
filter	String	查询的过滤条件。 支持的查询条件为：serverId（服务器 ID）。	否
range	String	查询服务器的范围，格式为 i-j，i 和 j 为正整数，0<j-i<1000。 按照服务器 ID 进行排序，返回第 i 到第 j 的服务器。 如果输入的查询范围超过服务器的总个数，返回的结果为空，如果未指定 range，则返回所有符合条件的服务器。	否

响应结果

名称	类型	描述
serverId	String	服务器 ID，若还没有初始化，则不显示此项。
status	String	远程协助是否启用。 ● Enabled：启用。 ● Disabled：禁用。

host	String	远程协助的服务器 Host。
port	Integer	远程协助的端口号。
code	String	远程协助码。

请求示例

查询远程协助。

```
GET /rest/v1/system/config/remoteAccess HTTP/1.1
Date: Thu, 17 Mar 2022 09:34:04 GMT
Host: 192.168.0.121:1443
Authorization: HBlock userName:signature
```

响应示例

```
HTTP/1.1 200 OK
Date: Thu, 17 Mar 2022 09:34:04 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 214
Connection: keep-alive
x-hblock-request-id: 0cda25588df44123ae5113876bd91ca0
Server: HBlock

{
  "data": {
    "servers": [
      {
        "serverId": "hblock_1",
        "status": "Enabled",
        "host": "182.151.21.174",
        "port": 18100,
        "code": "999342"
      },
      {
        "serverId": "hblock_2",
        "status": "Enabled",
```

```
    "host": "182.151.21.174",  
    "port": 18100,  
    "code": "258699"  
  }  
]  
}  
}
```

4.17.3.3 删除远程协助

此操作用来删除远程协助配置。

说明：如果不填写 `serverId`，表示删除当前被请求的服务器的远程协助配置。

请求语法

```
DELETE /rest/v1/system/config/remoteAccess/serverId HTTP/1.1  
Date: date  
Host: ip:port  
Authorization: authorization
```

请求示例

删除远程协助配置。

```
DELETE /rest/v1/system/config/remoteAccess/hblock_1 HTTP/1.1  
Date: Thu, 17 Mar 2022 09:37:30 GMT  
Authorization: HBlock userName:signature  
Host: 192.168.0.121:1443
```

响应示例

```
HTTP/1.1 204 No Content  
x-hblock-request-id: a4e3bc5f38a24992ab70da684388d6f6  
Connection: keep-alive  
Date: Thu, 17 Mar 2022 09:37:30 GMT  
Server: HBlock
```

4.17.3.4 错误码列表

HTTP status	错误码	错误信息	描述
400	CanNotConnectToServer	Can not connect to the server <i>IP[:port]</i> .	无法连接指定的服务器。
400	InvalidHost	The host is invalid.	host 无效。
400	MissingRemoteAccessHost	Value null at 'host' failed to satisfy constraint: Argument must not be null.	启用远程协助时，host 不能为空。
404	NoSuchRemoteAccessConfiguration	Remote access is not configured.	没有配置远程协助。
409	RemoteAccessAlreadyEnabled	Remote access to the <i>Host:Port</i> has been enabled, and the remote access code is: <i>CODE</i> . If you want to change the Host and Port, please disable the remote access first.	已经启用了远程协助，再次启用时 Host、Port 和第一次填写的不一致。

4.17.4 Pushgateway 监控配置

4.17.4.1 添加 Pushgateway 监控配置

此操作用来添加 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。

请求语法

```
POST /rest/v1/system/config/monitor?serverId=serverid1,serverid2,serveridN HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "pushgateway":address:port,
  "pushgatewayLabels": {
    "key1": "value1",
    "key2": "value2",
    "key3": "value3",
    ...
  }
}
```

请求参数

参数	类型	描述	是否必须
serverId	String	指定添加 Pushgateway 监控配置的 HBlock 服	否

		务器 ID。一次可以指定多个 HBlock 服务器的 ID，以英文逗号 (,) 分开。如果不填写，默认为所有 HBlock 服务器添加 Pushgateway 监控配置。	
pushgateway	String	指定 Pushgateway 的地址和接口。 取值：格式为 <i>IPv4:port</i> 、 <i>[IPv6]:port</i> 或者 <i>domain-name:port</i> 。	是
pushgatewayLabels	Object	指定 Pushgateway 对应的标签值。 取值：格式为 <i>"key": "value"</i> ， <i>key</i> 和 <i>value</i> 的类型都为字符串。	否

请求示例 1

为服务器 hblock_2、hblock_3 添加 Pushgateway 监控配置。

```
POST /rest/v1/system/config/monitor?serverId=hblock_3,hblock_2 HTTP/1.1
Date: Fri, 24 May 2024 02:24:01 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 138
Host: 192.168.0.110:1443

{
  "pushgateway": "192.168.0.1:9091",
  "pushgatewayLabels": {
    "agent": "agentname",
    "idc": "idcname"
  }
}
```

响应示例 1

```
HTTP/1.1 200 OK
x-hblock-request-id: 65ba83698f7742de82c2a26fcd57e6e1
Connection: keep-alive
Date: Fri, 24 May 2024 02:24:01 GMT
Server: HBlock
```

请求示例 2

为所有服务器添加 Pushgateway 监控配置。

```
POST /rest/v1/system/config/monitor HTTP/1.1
Date: Fri, 24 May 2024 02:43:57 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 130
Host: 192.168.0.110:1443

{
  "pushgateway": "192.168.0.1:9091",
  "pushgatewayLabels": {
    "agent": "ctyun",
    "idc": "idc8"
  }
}
```

响应示例 2

```
HTTP/1.1 200 OK
x-hblock-request-id: 9949f15915b24eada38615013e93ea53
Connection: keep-alive
Date: Fri, 24 May 2024 02:43:57 GMT
Server: HBlock
```


4.17.4.2 修改 Pushgateway 监控配置

此操作用来修改 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。

请求语法

```
PUT /rest/v1/system/config/monitor?serverId=serverid1,serverid2,serveridN HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "pushgateway":address:port,
  "pushgatewayTimeout":pushgatewayTimeout,
  "pushgatewayLabels": {
    "key1": "value1",
    "key2": "value2",
    "key3": "value3",
    ...
  },
  "changeAll": changeAll,
  "collectInterval": collectInterval,
  "collectMetric": name,
  "collectMetricItems": [
    "item1",
    "item2",
    "item3",
    ...
  ],
}
```

请求参数

参数	类型	描述	是否必须
serverId	String	指定要修改 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。如果不填写，默认为所有 HBlock 服务器修改 Pushgateway 监控配置。	否
pushgateway	String	指定 Pushgateway 的地址和接口。 取值：格式为 <i>IPv4:port</i> 、 <i>[IPv6]:port</i> 或者 <i>domain-name:port</i> 。	是
pushgatewayTimeout	Long	向 Pushgateway 发送数据的超时时间。 取值：[1000, 1000000]，单位为毫秒。	否
pushgatewayLabels	Object	指定 Pushgateway 对应的标签值。 取值：格式为 <i>"key": "value"</i> 。	否
changeAll	Boolean	修改配置时，以覆盖方式更新，还是以添加方式更新。主要针对 <i>pushgatewayLabels</i> 和 <i>collectMetricItems</i> 参数的设置。 注意： 如果未指定 <i>collectMetricItems</i> ，也未指定 <i>pushgatewayLabels</i> ，则此参数不起作用。 取值：	否

		● true: 配置以覆盖方式更新。 ● false: 配置以添加方式更新。 默认值为 false 。	
collectInterval	Long	指定采集时间间隔。 取值：负整数或[2000, 1000000]范围的，单位为毫秒。 说明： ● 如果与 collectMetric 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 collectMetric 时，表示设置所有监控指标的采集间隔，取值只能是[2000, 1000000]范围的整数。	否
collectMetric	String	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。	否
collectMetricItems	Array of collectMetricItem	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。	否

		● 如果未指定 <code>collectMetric</code>，不能指定此参数。	

请求示例 1

为服务器 hblock_2、hblock_3 修改 Pushgateway 监控配置。

```
PUT /rest/v1/system/config/monitor?serverId=hblock_3,hblock_2 HTTP/1.1
Date: Fri, 24 May 2024 02:59:53 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 317
Host: 192.168.0.110:1443

{
  "pushgateway": "192.168.0.1:9091",
  "pushgatewayTimeout": 3000,
  "pushgatewayLabels": {
    "agent": "ctyunoos",
    "idc": "2"
  },
  "changeAll": true,
  "collectInterval": 5000,
  "collectMetric": "disk",
  "collectMetricItems": [
    "tmpfs",
    "cgroup"
  ]
}
```

响应示例 1

```
HTTP/1.1 204 No Content
x-hblock-request-id: a097237d448f4371a8838df64e753f3e
Connection: keep-alive
Date: Fri, 24 May 2024 02:59:53 GMT
```

Server: HBlock

请求示例 2

为所有服务器修改 Pushgateway 监控配置。

```
PUT /rest/v1/system/config/monitor HTTP/1.1
Date: Fri, 24 May 2024 05:55:52 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 318
Host: 192.168.0.110:1443
```

```
{
  "pushgateway": "192.168.0.1:9091",
  "pushgatewayTimeout": 3000,
  "pushgatewayLabels": {
    "agent": "ctyun",
    "idc": "3"
  },
  "changeAll": false,
  "collectInterval": 5000,
  "collectMetric": "disk",
  "collectMetricItems": [
    "pstore",
    "devpts"
  ]
}
```

响应示例 2

```
HTTP/1.1 204 No Content
x-hblock-request-id: 860024bc04814789b55f9d662914922a
Connection: keep-alive
Date: Fri, 24 May 2024 05:55:52 GMT
Server: HBlock
```

4.17.4.3 删除 Pushgateway 监控配置

此操作用来删除 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。

请求语法

```
DELETE /rest/v1/system/config/monitor?serverId=serverid1,serverid2,serveridN HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "pushgateway":address:port,
  "pushgatewayLabels": [
    "key1",
    "key2",
    "key3",
    ...
  ]
  "collectMetric": name,
  "collectMetricItems": [
    "item1",
    "item2",
    "item3",
    ...
  ],
}
```

请求参数

参数	类型	描述	是否必须
serverId	String	指定要删除 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。如果不填写，默认为所有 HBlock 服务器删除指定的 Pushgateway 监控配置。	否
pushgateway	String	指定 Pushgateway 的地址和接口。 取值：格式为 <i>IPv4:port</i> 、 <i>[IPv6]:port</i> 或者 <i>domain-name:port</i> 。	是
pushgatewayLabels	Array of pushgatewayLabel	指定 Pushgateway 对应的标签值。 取值：label 项。	否
collectMetric	String	指定采集的监控指标。 取值为：server、fileSystem、interface、load、disk、tcp、os。 默认删除的上述所有监控指标。	否
collectMetricItems	Array of collectMetricItem	指定监控指标下的配置项。	否

请求示例 1

为服务器 hblock_1、hblock_2 删除相关的 Pushgateway 监控配置：标签为 agent、idc，监控指标为 disk，监控指标配置项为 pstore、devpts。

```
DELETE /rest/v1/system/config/monitor?serverId=hblock_1,hblock_2 HTTP/1.1
```

```
Date: Fri, 24 May 2024 07:01:39 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 216
Host: 192.168.0.110:1443
```

```
{
  "pushgateway": "192.168.0.1:9091",
  "pushgatewayLabels": [
    "agent",
    "idc"
  ],
  "collectMetric": "disk",
  "collectMetricItems": [
    "pstore",
    "devpts"
  ]
}
```

响应示例 1

```
HTTP/1.1 200 OK
x-hblock-request-id: 4b7487d492754feab5f572be3ae532f8
Connection: keep-alive
Date: Fri, 24 May 2024 07:01:39 GMT
Server: HBlock
```

请求示例 2

为所有服务器删除相关的 Pushgateway 监控配置：lable 为 agent、idc。

```
DELETE /rest/v1/system/config/monitor HTTP/1.1
Date: Fri, 24 May 2024 08:30:08 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 112
Host: 192.168.0.110:1443
```



```
{  
  "pushgateway": "192.168.0.1:9091",  
  "pushgatewayLabels": [  
    "agent",  
    "idc"  
  ]  
}
```

响应示例 2

```
HTTP/1.1 200 OK  
x-hblock-request-id: ba61c4c8ecf54ad89ca872faa02c8433  
Connection: keep-alive  
Date: Fri, 24 May 2024 08:30:08 GMT  
Server: HBlock
```

4.17.4.4 查询 Pushgateway 监控配置

此操作用来查询 Pushgateway 监控配置信息。

请求语法

```
GET /rest/v1/system/config/monitor?serverId=serverid1,serverid2,serveridN HTTP/1.1
HTTP/1.1
Date:date
Host: ip:port
Authorization:authorization
```

请求参数

参数	类型	描述	是否必须
serverId	String	指定查询 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。如果不填写，默认查询所有 HBlock 服务器的 Pushgateway 监控配置。	否

响应结果

名称	类型	描述
monitors	Array of monitor	监控信息，详见“表 1 响应参数 monitor 说明”。

表 1 响应参数 monitor 说明

名称	类型	描述
serverId	String	服务器 ID。
send.pushgatewayTimeout	Long	向所有 Pushgateway 推送数据的超时时长，超时即认为推送失败。
send.pushgateways	Array of	Pushgateway 信息，详见“表 2 响应参数

	pushgateway	Pushgateway 说明” 。
collect.interval	Long	硬件指标采集的采集间隔。
collect.metrics	Array of metric	硬件采集指标信息，详见“ 表 3 响应参数 metric 说明 ”。

表 2 响应参数 Pushgateway 说明

名称	类型	描述
address	String	Pushgateway 的 IP 或域名。
port	Integer	Pushgateway 的端口号。
labels.agent	String	Pushgateway 的客户端。
labels.idc	Integer	数据所属的数据中心。
labels.hostname	String	主机名称。
labels.serverIp	String	服务器 IP。
labels.clusterId	String	集群 ID。
labels.clusterName	String	集群名称。
labels.job	String	Pushgateway 推送数据的来源。

表 3 响应参数 metric 说明

名称	类型	描述
name	String	采集间隔或采集对象的硬件指标类别名称。
interval	Long	硬件指标类别单独设置的采集间隔。
items	String	硬件指标类别单独设置的采集对象。

请求示例 1

查询服务器 hblock_2、hblock_3 的 Pushgateway 监控配置。

```
GET /rest/v1/system/config/monitor?serverId=hblock_2,hblock_3 HTTP/1.1
Date: Fri, 24 May 2024 08:58:12 GMT
```

```
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Host: 192.168.0.110:1443
```

响应示例 1

```
HTTP/1.1 200 OK
x-hblock-request-id: 4b5e0623e73a4dcd8754fd2ed504b9c1
Connection: keep-alive
Content-Length: 1115
Date: Fri, 24 May 2024 08:58:12 GMT
Content-Type: application/json; charset=utf-8
Server: HBlock

{
  "data": {
    "monitors": [
      {
        "serverId": "hblock_2",
        "send": {
          "interval": 20000,
          "timeout": 2000,
          "pushgatewayTimeout": 3000,
          "pushgateways": [
            {
              "address": "192.168.0.1",
              "port": 9091,
              "labels": {
                "agent": "ctyun",
                "idc": "3",
                "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"
          ]
        },
        {
          "name": "server",
          "interval": 5000,
          "items": [
            "cpu"
          ]
        },
        {
          "name": "disk",
          "interval": 5000,
          "items": [
            "tmpfs",
            "devpts",
            "pstore",
            "cgroup"
          ]
        }
      ]
    }
  },
  {
```

```
"serverId": "hblock_3",
"send": {
  "interval": 20000,
  "timeout": 2000,
  "pushgatewayTimeout": 3000,
  "pushgateways": [
    {
      "address": "192.168.0.1",
      "port": 9091,
      "labels": {
        "agent": "ctyun",
        "idc": "3",
        "hostname": "ecs-9689-0915140",
        "serverIp": "192.168.0.102",
        "clusterId": "B8BFE310-8DF6-429E-8C8A-ACD22310AC08",
        "clusterName": "stor2",
        "job": "hblock"
      }
    }
  ]
},
"collect": {
  "interval": 14000,
  "metrics": [
    {
      "name": "storFs",
      "interval": 20000,
      "items": [
        "ds-1"
      ]
    },
    {
      "name": "server",
      "interval": 5000,
      "items": [
```

```
        "cpu"
      ],
    },
    {
      "name": "disk",
      "interval": 5000,
      "items": [
        "tmpfs",
        "devpts",
        "pstore",
        "cgroup"
      ]
    }
  ]
}
```

请求示例 2

查询所有服务器的 Pushgateway 监控配置。

```
GET /rest/v1/system/config/monitor HTTP/1.1
Date: Fri, 24 May 2024 09:06:56 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Host: 192.168.0.110:1443
```

响应示例 2

```
HTTP/1.1 200 OK
x-hblock-request-id: 4b5e0623e73a4dcd8754fd2ed504b9c1
Connection: keep-alive
Content-Length: 1115
Date: Fri, 24 May 2024 09:06:56 GMT
```

Content-Type: application/json;charset=utf-8

Server: HBlock

```
{
  "data": {
    "monitors": [
      {
        "serverId": "hblock_1",
        "send": {
          "interval": 20000,
          "timeout": 2000,
          "pushgatewayTimeout": 3000,
          "pushgateways": [
            {
              "address": "192.168.0.1",
              "port": 9091,
              "labels": {
                "hostname": "hblockserver",
                "serverIp": "192.168.0.110",
                "clusterId": "B8BFE310-8DF6-429E-8C8A-ACD22310AC08",
                "clusterName": "stor2",
                "job": "hblock",
                "agent": "ctyun",
                "idc": "3"
              }
            }
          ]
        },
        "collect": {
          "interval": 14000,
          "metrics": [
            {
              "name": "storFs",
              "interval": 20000,
              "items": [
```



```
        "ds-1",
        "ds-2",
        "mdm"
    ]
},
{
    "name": "disk",
    "interval": 5000,
    "items": [
        "tmpfs",
        "devpts",
        "pstore",
        "cgroup"
    ]
}
]
}
},
{
    "serverId": "hblock_3",
    "send": {
        "interval": 20000,
        "timeout": 2000,
        "pushgatewayTimeout": 3000,
        "pushgateways": [
            {
                "address": "192.168.0.1",
                "port": 9091,
                "labels": {
                    "agent": "ctyun",
                    "idc": "3",
                    "hostname": "ecs-9689-0915140",
                    "serverIp": "192.168.0.102",
                    "clusterId": "B8BFE310-8DF6-429E-8C8A-ACD22310AC08",
                    "clusterName": "stor2",
```

```
        "job": "hblock"
      }
    }
  ]
},
"collect": {
  "interval": 14000,
  "metrics": [
    {
      "name": "storFs",
      "interval": 20000,
      "items": [
        "ds-1"
      ]
    },
    {
      "name": "server",
      "interval": 5000,
      "items": [
        "cpu"
      ]
    },
    {
      "name": "disk",
      "interval": 5000,
      "items": [
        "tmpfs",
        "devpts",
        "pstore",
        "cgroup"
      ]
    }
  ]
},
```

```
{
  "serverId": "hblock_2",
  "send": {
    "interval": 20000,
    "timeout": 2000,
    "pushgatewayTimeout": 3000,
    "pushgateways": [
      {
        "address": "192.168.0.1",
        "port": 9091,
        "labels": {
          "agent": "ctyun",
          "idc": "3",
          "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"
        ]
      },
      {
        "name": "server",
```

```
        "interval": 5000,  
        "items": [  
            "cpu"  
        ]  
    },  
    {  
        "name": "disk",  
        "interval": 5000,  
        "items": [  
            "tmpfs",  
            "devpts",  
            "pstore",  
            "cgroup"  
        ]  
    }  
]  
}  
]  
}  
]  
}  
}
```

4.17.4.5 错误码列表

HTTP status	错误码	错误信息	描述
400	AddConfigFailed	Servers cannot be connected: <i>serverid1, serverid2....</i>	添加 Pushgateway 监控配置的服务器无法连接。
400	DeleteConfigFailed	Servers cannot be connected: <i>serverid1, serverid2....</i>	要删除 Pushgateway 监控配置的服务器无法连接。
400	ExceedMaxValue	Value <i>value</i> at 'number' failed to satisfy constraint: Argument must be less than or equal to <i>value</i> .	超过最大值。
400	InvalidEnumValue	Value <i>value</i> at 'argument' failed to satisfy constraint: Argument must satisfy enum value set: [<i>value1, value2... </i>]	枚举参数不合法。
400	InvalidKeyInPushgatewayLabel	The key in the pushgateway label can not be the same as built-in keywords, which include type, cpu, device, fstype, mountpoint, lu, path, file, mode, address and target.	不可使用系统内置关键字作为标签中的键名称，内置关键字有：type, cpu, device, fstype, mountpoint, lu, path, file, mode,

			address, target。
400	InvalidPort	Value <i>value</i> at 'port of pushgateway' failed to satisfy constraint: Argument must have value between 1 and 65535.	Port 的取值必须在[1, 65535]之间。
400	InvalidString	Value " at ' <i>argument</i> ' failed to satisfy constraint: Argument must not be an empty string.	参数不能为空字符串。
400	LessThanMinValue	Value <i>value</i> at 'number' failed to satisfy constraint: Argument must be greater than or equal to <i>value</i> .	该参数必须大于或等于另一个参数。
400	MissingAtLeastOneArgument	Missing argument, at least one argument in ' <i>argument1</i> , <i>argument2</i> , <i>argument3</i> ...' is needed.	至少输入一个参数值。
400	SetConfigFailed	Servers cannot be connected: <i>serverid1</i> , <i>serverid2</i> The pushgateway with address <i>address</i> and port <i>port</i> does not exist in <i>serverid1</i> , <i>serverid2</i>	设置服务器 Pushgateway 监控的服务器无法连接。 指定的 Pushgateway 不存在。

4.17.5 智维平台推送告警配置

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。

4.17.5.1 添加向智维平台推送告警的配置

此操作用来添加向智维平台推送告警的配置。

请求语法

```
POST /rest/v1/system/config/ctyuncms HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "url":url,
  "labels": {
    "key1": "value1",
    "key2": "value2",
    "key3": "value3",
    ...
  }
}
```

请求参数

参数	类型	描述	是否必须
url	String	智维平台第三方告警的推送地址。	否
labels	Object	指定推送至智维平台的告警数据中携带的标签信息。	是

		取值: "key":"value"格式。 注意: 标签 idc 和 mid 必须添加。	

请求示例

```
POST /rest/v1/system/config/ctyuncms HTTP/1.1
Date: Mon, 27 May 2024 00:56:03 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 162
Host: 192.168.0.110:1443

{
  "url": "https://ctyuncms:9527/blackhole/api/v1/issue",
  "labels": {
    "mid": "65de96adf2faca1f3a75b33a",
    "idc": "neimeng"
  }
}
```

响应示例

```
HTTP/1.1 200 OK
x-hblock-request-id: 10787692b440453ba0e17c50a8c0cbcb
Connection: keep-alive
Date: Mon, 27 May 2024 00:56:03 GMT
Server: HBlock
```


4.17.5.2 修改向智维平台推送告警的配置

此操作用来修改向智维平台推送告警的配置。

请求语法

```
PUT /rest/v1/system/config/ctyuncms HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "url": url,
  "labels": {
    "key1": "value1",
    "key2": "value2",
    "key3": "value3",
    ...
  },
  "all": changeAll,
}
```

请求参数

参数	类型	描述	是否必须
url	String	智维平台第三方告警的推送地址。	是
labels	Object	指定推送至智维平台的告警数据中携带的标签信息。 取值: " <i>key</i> ": " <i>value</i> "格式。	是
all	Boolean	修改配置时, 以覆盖方式更新, 还是以添加方式更新。 取值: ● true: 配置以覆盖方式更新。	否

		● false: 配置以添加方式更新。 默认值为 false 。	

请求示例

```
PUT /rest/v1/system/config/ctyuncms HTTP/1.1
Date: Mon, 27 May 2024 01:13:58 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 177
Host: 192.168.0.110:1443

{
  "url": "https://ctyuncms:9527/blackhole/api/v1/issue",
  "labels": {
    "mid": "65de96adf2faca1f3a75b33a",
    "idc": "xinjiang"
  },
  "all": true
}
```

响应示例

```
HTTP/1.1 204 No Content
x-hblock-request-id: 0bbdce9b065545b891a4433664a9c81e
Connection: keep-alive
Date: Mon, 27 May 2024 01:13:58 GMT
Server: HBlock
```

4.17.5.3 删除向智维平台推送告警的配置

此操作用来删除向智维平台推送告警的配置。

请求语法

```
DELETE /rest/v1/system/config/ctyuncms HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "url":url,
  "labels": [
    "key1",
    "key2",
    "key3",
    ...
  ]
}
```

请求参数

参数	类型	描述	是否必须
url	String	智维平台第三方告警的推送地址。	是
labels	Array of string	指定要删除的标签信息。 注意： idc 和 mid 不能删除。 说明： 如果未指定，说明删除指定智维平台的所有推送告警配置信息。 取值： <i>"key"</i> 。	否

请求示例 1

删除智维平台 <https://ctyuncms:9527/blackhole/api/v1/issue> 的告警配置项：

oos、ctyun。

```
DELETE /rest/v1/system/config/ctyuncms HTTP/1.1
Date: Mon, 27 May 2024 01:50:30 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 122
Host: 192.168.0.110:1443

{
  "url": "https://ctyuncms:9527/blackhole/api/v1/issue",
  "labels": [
    "oos",
    "ctyun"
  ]
}
```

响应示例 1

```
HTTP/1.1 200 OK
x-hblock-request-id: 7751e86299ef429494747eaad551b2ed
Connection: keep-alive
Date: Mon, 27 May 2024 01:50:30 GMT
Server: HBlock
```

请求示例 2

删除智维平台 <https://ctyuncms:9527/blackhole/api/v1/issue> 的所有告警的配置。

```
DELETE /rest/v1/system/config/ctyuncms HTTP/1.1
Date: Mon, 27 May 2024 01:58:10 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 63
Host: 192.168.0.110:1443
```

```
{  
  "url": "https://ctyuncms:9527/blackhole/api/v1/issue"  
}
```

响应示例 2

```
HTTP/1.1 200 OK  
x-hblock-request-id: 38024a15cbb243aa9c5b38a7c3238685  
Connection: keep-alive  
Date: Mon, 27 May 2024 01:58:10 GMT  
Server: HBlock
```

4.17.5.4 查询向智维平台推送告警的配置

此操作用来查询向智维平台推送告警的配置信息。

请求语法

```
GET /rest/v1/system/config/ctyuncms HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

响应结果

名称	类型	描述
ctyunCMSInfos	Array of ctyunCMSInfo	智维平台第三方告警的推送信息，详见“表 1 响应参数 ctyunCMSInfo 说明”。

表 1 响应参数 ctyunCMSInfo 说明

名称	类型	描述
url	String	智维平台第三方告警的推送地址。
labels	Array of label	推送至智维平台的告警数据中携带的标签信息。

请求示例

```
GET /rest/v1/system/config/ctyuncms HTTP/1.1
Date: Mon, 27 May 2024 02:10:19 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Host: 192.168.0.110:1443
```

响应示例

```
HTTP/1.1 200 OK
x-hblock-request-id: 4bfaf79117c04fac8dab84b3b2a5f4da
Connection: keep-alive
Content-Length: 187
Date: Mon, 27 May 2024 02:10:19 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "ctyunCMSInfos": [
      {
        "url": "https://ctyuncms:9527/blackhole/api/v1/issue",
        "labels": {
          "mid": "65de96adf2faca1f3a75b33a",
          "idc": "neimeng",
          "hblockname": "stor1",
          "clusterID": "cluster1"
        }
      }
    ]
  }
}
```

4.17.5.5 错误码列表

HTTP status	错误码	错误信息	描述
400	CtyunCMSAlreadyExists	The ctyuncms ' <i>url</i> ' already exists.	此 URL 配置已经存在。
400	InvalidString	Value " at ' <i>argument</i> ' failed to satisfy constraint: Argument must not be an empty string.	参数不能为空。
400	KeyMustExistInCtyunCMSLabel	These keys must exist in the ctyuncms labels: mid, idc.	标签中必须包含 idc 和 mid 两个 key。
400	NoSuchCtyunCMS	The ctyuncms ' <i>url</i> ' does not exist.	此 URL 不存在。

4.17.6 鉴权方式

4.17.6.1 设置鉴权方式

此操作用来设置 HBlock 鉴权方式。

请求语法

```
PUT /rest/v1/system/config/parameter HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "authMode": authMode
}
```

请求参数

参数	类型	描述	是否必须
authMode	String	设置 HBlock 鉴权方式。 取值： ● strict：表示使用最新的鉴权方式。 ● compatible：表示兼容历史的鉴权方式。 从 3.8 以下的版本升级上来的集群，鉴权方式为 compatible；3.8 及以上的版本，安装之后鉴权方式默认为 strict。 说明： ● 3.8 版本之前的鉴权方式为：SHA256(Date 请求头的值 + "\n" + SHA256(password))。	否

		● 3.8 版本及后续版本鉴权方式为：HMAC-SHA256(SHA256(password), Date 请求头的值 + "\n" + SHA256(password))。	

请求示例

设置 HBlock 使用最新的鉴权方式。

```
PUT /rest/v1/system/config/parameter HTTP/1.1
Date: Fri, 09 Jan 2026 08:29:37 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 30
Connection: keep-alive
Host: 192.168.0.65:1443

{
  "authMode": "strict"
}
```

响应示例

```
HTTP/1.1 200 OK
x-hblock-request-id: f50be9c8d72e4f0ea2eb8850e69b7fcd
Connection: keep-alive
Date: Fri, 09 Jan 2026 08:29:37 GMT
Server: HBlock
```

4.17.6.2 查询鉴权方式

此操作用来查询 HBlock 鉴权方式。

请求语法

```
GET /rest/v1/system/config/parameter HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

响应结果

名称	类型	描述
authMode	String	HBlock 鉴权方式： ● strict: 表示使用最新的鉴权方式。 ● compatible: 表示兼容历史的鉴权方式。

请求示例

查询 HBlock 鉴权方式。

```
GET /rest/v1/system/config/parameter HTTP/1.1
Date: Fri, 09 Jan 2026 08:29:37 GMT
Authorization: HBlock userName:signature
Connection: keep-alive
Host: 192.168.0.65:1443
```

响应示例

```
HTTP/1.1 200 OK
x-hblock-request-id: 4b5e0623e73a4dcd8754fd2ed504b9c1
Date: Fri, 09 Jan 2026 08:35:13 GMT
Connection: keep-alive
Content-Type: application/json;charset=utf-8
Content-Length: 30
Server: HBlock
```

```
{  
  "data": {  
    "authMode": "strict"  
  }  
}
```

4.17.6.3 错误码列表

HTTP status	错误码	错误信息	描述
400	SetConfigFailed	Failed to set config.	配置失败。

4.17.7 开启高级功能试用（免费版）

4.17.7.1 开启高级功能试用

此操作用来开启免费版的高级功能试用。

高级功能开启后，可以试用商业版的一些高级功能，商业版与免费版的区别详见[商业版与免费版](#)。

注意：

- 免费版享有 2 年的升级服务，到期后不再支持升级。若用户从未开启过高级功能试用，2 年后仍可正常开启试用。
- 对于免费版，高级功能试用只能开启 1 次。开启高级功能试用之后，可以试用高级功能 1 个月，1 个月之后，高级功能将被禁用。

请求语法

```
PUT /rest/v1/system/config/protrial HTTP/1.1
Date: date
Connection: keep-alive
Content-Length: length
Host: ip:port
Authorization: authorization
```

响应结果

名称	类型	描述
proTrialStartTime	Long	免费版的高级功能试用开启时间，unix 时间戳（UTC），精确到毫秒。
proTrialExpireTime	Long	免费版的高级功能试用结束时间，unix 时间戳（UTC），精确到毫秒。

请求示例

开启免费版的高级功能试用。

```
PUT /rest/v1/system/config/protrial HTTP/1.1
Date: Mon, 12 Jan 2026 07:28:32 GMT
Authorization: HBlock userName:signature
Content-Length: 0
Connection: keep-alive
Host: 192.168.0.65:1443
```

响应示例

```
HTTP/1.1 200 OK
x-hblock-request-id: 392b32b685ca4556965878adef3d4093
Connection: keep-alive
Content-Type: application/json;charset=utf-8
Content-Length: 79
Date: Mon, 12 Jan 2026 07:28:32 GMT
Server: HBlock

{
  "data": {
    "proTrialStartTime": 1768202912282,
    "proTrialExpireTime": 1770794912282
  }
}
```

4.17.7.2 错误码列表

HTTP status	错误码	错误信息	描述
400	CommercialNotAllowed	The request is invalid in the commercial edition.	商业版不支持该操作。
400	ProFeatureDisabled	This feature is only available in the commercial edition. Please contact your software vendor to get a license to unlock more capabilities.	此功能仅在商业版中提供，请联系软件供应商获取商业版软件许可证，解锁更多功能。
400	ProFeatureTrialNotify	This feature is only available in the commercial edition. You can start a Pro trial or contact your software vendor to obtain a license to unlock more capabilities.	此功能仅在商业版中提供，请开启高级功能试用权限，或联系软件供应商获取商业版软件许可证，解锁更多功能。
409	ProFeatureEnabled	The Pro trial (started on YYYY-MM-DD XX:XX:XX) ends on YYYY-MM-DD XX:XX:XX. This is your one and only trial opportunity.	高级功能试用权限已于 YYYY-MM-DD XX:XX:XX 开启，截止时间为 YYYY-MM-DD XX:XX:XX，此后不再提供任何形式的高级功能试用权限。

4.18 调整 HBlock 性能参数（集群版适用）

4.18.1 调整 HBlock 性能参数

根据硬件环境和业务场景，可以使用此操作进行性能调优。

请求语法

```
PUT /rest/v1/system/tuning HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "perfScen": perfScen,
  "haSens": haSens
}
```

请求参数

参数	类型	描述	是否必须
perfScen	String	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。	否
haSens	String	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。	否

		● Low: 低敏感型场景。 默认值为 Middle 。	

请求示例

HBlock 性能调优。

```
PUT /rest/v1/system/tuning HTTP/1.1
Date: Thu, 19 Dec 2024 08:39:13 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 50
Host: 192.168.0.192:1443

{
  "perfScen": "ICv1",
  "haSens": "Low"
}
```

响应示例

```
HTTP/1.1 204 No Content
Date: Thu, 19 Dec 2024 08:39:13 GMT
Connection: keep-alive
x-hblock-request-id: e2216c93aade40e3b7dec96f6b0ef15a
Server: HBlock
```

4.18.2 查看性能调优配置

此操作用来查询性能调优配置。

请求语法

```
GET /rest/v1/system/tuning HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

响应结果

名称	类型	描述
perfScen	String	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。
haSens	String	性能调优针对的场景敏感型： ● High: 高敏感型场景。 ● Middle: 中等敏感型场景。 ● Low: 低敏感型场景。 ● custom: 用户设置的场景。 ● Unknown: 场景未知。

请求示例

查看性能调优配置。

```
GET /rest/v1/system/tuning HTTP/1.1
Date: Thu, 19 Dec 2024 08:58:46 GMT
Host: 192.168.0.192:1443
Authorization: HBlock userName:signature
```

响应示例

```
HTTP/1.1 200 OK
Date: Thu, 19 Dec 2024 08:58:46 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 46
Connection:keep-alive
x-hblock-request-id: ee4731a7add542f9af6d0955e3fbf4b1
Server: HBlock

{
  "data": {
    "perfScen": "ICv1",
    "haSens": "Middle"
  }
}
```

4.18.3 错误码

HTTP status	错误码	错误信息	描述
400	StandaloneModeNotAllowed	'operation' is not supported by standalone mode of HBlock.	单机版本 HBlock 不支持该操作。
404	TuningNotComplete	The previous tuning operation failed to execute on <i>serverid1,serverid2.....</i> , the system is automatically retrying. In the meantime, please check and ensure network connection is stable. If the issue persists, please contact software vendor.	上一次的性能调优操作未结束。

4.19 查询 HBlock 版本

此操作用来查询 HBlock 版本信息。

请求语法

```
GET /rest/v1/system/version HTTP/1.1
```

Date: *date*

Host: *ip:port*

Authorization: *authorization*

响应结果

名称	类型	描述
version	String	HBlock 的版本号。
type	String	HBlock 的版本类型： ● Stand-alone: 单机版。 ● Cluster: 集群版本。

请求示例

查询 HBlock 的版本信息。

```
GET /rest/v1/system/version HTTP/1.1
```

Date: Wed, 21 Jan 2026 08:51:02 GMT

Authorization: HBlock *userName:signature*

Host:192.168.0.121:1443

响应示例

```
HTTP/1.1 200 OK
```

x-hblock-request-id: ac2989a30d49439190da2f8ac9a3e520

Connection: keep-alive

Content-Length: 45

```
Date: Wed, 21 Jan 2026 08:51:02 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock
```

```
{
  "data": {
    "version": "4.0.0",
    "type": "Cluster"
  }
}
```

4.20 升级 HBlock

4.20.1 升级 HBlock

此操作用来升级 HBlock。

说明：可以通过查询升级进度接口查询升级服务是否完成。

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。
 - 如果软件许可证是订阅模式，需在有效期内；如果软件许可证是永久许可模式，需在维保期限内；如果处于试用期，需试用期未结束。
- 集群版：如果卷是高可用类型，至少保证卷的主备连接正常。
- 没有处于失败或者任务进行中状态的卷。
- 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。
- 集群版：数据服务（`stor:ds-x`）正常。
- 集群版：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`）。

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。
- 如果集群由不同架构服务器组成，请添加所有架构的升级文件，并保持版本一致。

请求语法

● 同架构升级：

```
PUT /rest/v1/system/upgrade HTTP/1.1
Date: date
Content-Type: multipart/form-data; boundary=value
Content-Length: length
x-update-parameter: key1=value1;key2=value2
Host: ip:port
Authorization: authorization

<upgradeFile>
```

● 混合部署升级（仅集群版支持）：

```
POST /rest/v1/system/upgrade HTTP/1.1
Date: date
Content-Type: multipart/form-data; boundary=value
Content-Length: length
Content-MD5: MD5
x-update-parameter: key1=value1;key2=value2
Host: ip:port
Authorization: authorization

upgradeFile1name:upgradeFile1
upgradeFile2name:upgradeFile2
.....
```

请求头

参数	类型	描述	是否必须
Content-Type	String	请求体 MIME 类型。 取值： ● 如果同架构升级，取值为 application/zip 或 multipart/form-data;	否

		<code>boundary=value</code>。 如果混合部署升级，取值为 <code>multipart/form-data</code>； <code>boundary=value</code>。 <code>boundary</code> 是一个分隔符，用于分隔表单中的不同字段，它的值由浏览器或客户端生成，通常是随机的。	
<code>x-update-parameter</code>	String	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，按 <code>key1=value1;key2=value2....</code> 的形式添加多个即可。 <code>key</code> 为参数名，<code>value</code> 为参数值，以“;”分割。	否

请求参数

参数	类型	描述	是否必须
<code>upgradeFile</code>	无	升级安装包的文件流。 如果集群使用的混合部署，需要上传集群中所有架构的升级安装包。	是

请求示例

升级 HBlock 服务。

```
PUT /rest/v1/system/upgrade HTTP/1.1
Date: Wed, 11 Mar 2026 06:04:03 GMT
Content-Type: multipart/form-data; boundary=-----068179374118126185601175
Authorization: HBlock userName:signature
Content-Length: 209060900
```

Host: 192.168.0.65:1443

upgradeFile: *upgradeFile*

响应示例

```
HTTP/1.1 202 Accepted
Date: Wed, 11 Mar 2026 06:04:03 GMT
Content-Length: 0
Connection: keep-alive
x-hblock-request-id: 6d028531f21940a1a20129eb49505676
Server: HBlock
```

4.20.2 查询升级进度

此操作用来升级进度。

请求语法

```
GET /rest/v1/system/upgrade HTTP/1.1
```

Date: *date*

Host: *ip:port*

Authorization: *authorization*

响应结果

名称	类型	描述
status	String	HBlock 服务升级状态： ● NotUpgraded: 未进行过升级。 ● Preparation: 正在准备升级。 ● Processing: 正在升级。 ● Succeeded: 升级成功。 ● Failed: 升级失败。 ● Interrupted: 升级中断。 ● Unknown: 未知。
progress	String	升级进程详细信息。
targetVersion	String	目标升级版本。
startTime	Long	本次升级开始时间，unix 时间戳（UTC），精确到毫秒。
lastEndTime	Long	上次升级结束时间，unix 时间戳（UTC），精确到毫秒。

请求示例

查看 HBlock 服务升级进度。

```
GET /rest/v1/system/upgrade HTTP/1.1
Date: Wed, 11 Mar 2026 06:07:50 GMT
Host: 192.168.0.65:1443
Authorization: HBlock userName:signature
```

响应示例

```
HTTP/1.1 200 OK
x-hblock-request-id: 1f4eb8ed32764e08b8cbc00252385fa4
Date: Wed, 11 Mar 2026 06:07:50 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 1326
Connection: keep-alive
Server: HBlock

{
  "data": {
    "status": "Processing",
    "progress": "Server info:\nhblock_1, 192.168.0.65, songt-0004\nhblock_2,
192.168.0.64, k8s-master\nhblock_3, 192.168.0.67, songt-0006\n\n2026-03-11 14:04:45
CST+0800 [Step 1/5] Checking for system status...\n2026-03-11 14:04:45 CST+0800
Checking system info\n2026-03-11 14:04:46 CST+0800 Checking data status\n\n2026-03-11
14:04:46 CST+0800 [Step 2/5] Uploading update files...\n\r2026-03-11 14:04:49 CST+0800
Upload [100%] for hblock_1 ###[server 1/3]\n\r2026-03-11 14:04:52 CST+0800 Upload
[100%] for hblock_2 ###[server 2/3]\n\r2026-03-11 14:04:54 CST+0800 Upload [100%] for
hblock_3 ###[server 3/3]\n\n2026-03-11 14:04:54 CST+0800 [Step 3/5] Upgrade
preparing...\n\n2026-03-11 14:05:31 CST+0800 [Step 4/5] Installing updates...\n\r2026-
03-11 14:05:31 CST+0800 Upgrading config - Update[√] ### [module 1/12]\n\r2026-03-11
14:05:40 CST+0800 Upgrading cs for hblock_1 - Check[√], Stop[√], Update[√], Start[√]
### [server 1/3] [module 2/12]\n\r2026-03-11 14:06:25 CST+0800 Upgrading cs for
hblock_2 - Check[√], Stop[√], Update[√], Start[√] ### [server 2/3] [module
2/12]\n\r2026-03-11 14:07:09 CST+0800 Upgrading cs for hblock_3 - Check[√], Stop[√],
```

```
Update[√], Start[&gt;] ### [server 3/3] [module 2/12]",  
    "targetVersion": "4.0.0",  
    "startTime": 1773209085788,  
    "lastEndTime": 0  
}  
}
```

4.20.3 错误码

HTTP status	错误码	错误信息	描述
400	CanNotConnectToIP	Can not connect to the IP <i>IP</i> .	无法连接指定的 IP。
400	CanNotConnectToServer	Can not connect to the server <i>IP[:port]</i> .	无法连接指定的服务器。
400	CanNotConnectToService	Can not connect to ' <i>serviceName</i> ' service on server <i>serverIp[:port]</i> .	服务调用不通。
400	DuplicatePackage	The name of the upgrade package must be unique. Please check and try again.	上传的安装包重名，请检查并重试。
400	ExceedThreshold	The number of <i>operation</i> cannot exceed <i>value</i> .	升级操作中的文件个数超阈值。
400	GetUpgradeStatusFailed	Failed to get upgrade status. Please check the network, service, etc, then try again.	无法获取到升级状态，请检查网络连接或者服务状态后重试。
400	InitUpgradeFailed	The upgrade operation failed to start. <i>reason</i>	升级操作启动失败。
400	InsufficientFDForLUNToWrite	Available fault domains and healthy disk paths are insufficient for the LUNs to write: <i>lunname1</i> , <i>lunname2</i> , <i>lunname3</i> Make sure	卷的可用故障域和健康数据目录数量不足，请确保可用故障域和健康数据目录的数量大于卷

		available fault domains and healthy disk paths are greater than the minimum replica number of the LUNs.	要求的最小副本数。
400	InsufficientAvailableServices	The number of available ' <i>serviceName</i> ' service node in the cluster must be greater than or equal to <i>value</i> .	集群中的某服务的可用节点数必须大于或等于某值才可以。
400	InvalidDouble	Value <i>value</i> at ' <i>argument</i> ' failed to satisfy constraint: Argument must be of type double.	double 类型不合法。
400	InvalidLong	Value <i>value</i> at ' <i>argument</i> ' failed to satisfy constraint: Argument must be of type long.	参数填写错误，必须为长整型。
400	InvalidPackage	Invalid upgrade package. Please get the correct package.	安装包无效。
400	InvalidProductType	Cannot run upgrade, product type mismatch. Please get the correct package.	安装包不匹配。
400	InvalidStorStatus	The HBlock status is ' <i>status</i> ', the request is invalid.	HBlock 当前的状态不正确，请求无效。
400	InvalidVersion	No need to upgrade because your current version is	不需要升级，当前运行的版本更新。

		newer.	
400	LoadConfigFailed	Failed to load configuration files.	获取配置文件失败。
400	LoadPackageFailed	Failed to load upgrade package. Reason: <i>Reason</i>	获取升级包失败。
400	LowVersion	The upgrade package requires the current system version to be <i>minimumVersion</i> or later.	当前版本太旧，不符合升级包需求。
400	MissingArchPackage	Missing <i>architecture</i> upgrade package. Please get the correct package.	缺少必要的升级包，请更换正确的安装包后重试。
400	MissingArgument	Value null at ' <i>argument</i> ' failed to satisfy constraint: Argument must not be null.	Json 参数不能为空。
400	NoUpgradeOperation	No upgrade operation.	没有升级操作。
400	SameVersion	No need to upgrade because the versions are same.	不需要升级，当前版本和目标升级版本相同。
400	ServiceWaitTimeout	' <i>serviceName</i> ' service wait timeout.	服务等待超时。
400	SetConfigFailed	Failed to set config.	配置失败。
400	SystemDataDegraded	There is low redundancy or error data in the system, operation is not allowed. Please repair fault domains and disk paths.	系统中存在降级或错误数据，不允许操作。请修复故障域和磁盘路径。

400	UpgradeExpired	Your upgrade function has expired on xxxx-xx-xx xx:xx:xx . To ensure business continuity and access the full product experience, please contact your software vendor to get a license.	您的免费版升级服务已于 xxxx-xx-xx xx:xx:xx 到期，现已停止更新支持。为确保您的业务连续性并获得完整的产品体验，请联系软件供应商获取商业版软件许可证。
400	UploadPackageFaied	Failed to upload or unzip package. Reason: <i>reason</i> .	上传或者解压缩安装包失败。
409	InvalidTargetIQNsStatus	The request is invalid. Ensure the number of target IQNs whose status is neither unavailable nor offline is greater than or equal to 2.	操作失败，target IQNs 状态异常，请检查后重试。
409	UpgradeInProgress	The upgrade task is in progress, the request is invalid.	系统升级正在进行中，无法执行操作。

5 附录

5.1 转义字符表

对于下列特殊字符，在响应体中会进行转义。

转义前	转义后
<	<
>	>
(	(
)	)
&	&

5.2 HBlock 服务

服务	服务名称	作用
stor:mdm	元数据管理服务（仅集群版支持）	管理整个系统的元数据。
stor:fc	故障转移控制服务（仅集群版支持）	进行系统健康检测，实现故障转移控制。
stor:ls	日志服务（仅集群版支持）	提供基于日志的数据同步功能。
stor:ds-x	数据服务（仅集群版支持）	管理用户的文件数据块。
stor:cs	协调服务（仅集群版支持）	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。

5.3 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
DeletetargetAllowlist	删除 target 允许访问列表

事件	描述
SettargetAllowlist	修改 target 允许访问列表

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
WipeLUN	清空卷
DeleteLUNXattr	删除卷扩展属性
SetLUNXattr	修改卷扩展属性
SuspendLUN	挂起卷
ResumeLUN	恢复卷

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

事件	描述
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
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

事件	描述
SetConsistencySnapshot	修改一致性快照
RollbackConsistencySnapshot	回滚一致性快照
DeleteConsistencySnapshot	删除一致性快照

QoS 策略

事件	描述
CreateQoSPolicy	创建 QoS 策略
SetQoSPolicy	修改 QoS 策略
DeleteQoSPolicy	删除 QoS 策略
AssociateQoSPolicy	关联 QoS 策略
DisassociateQoSPolicy	解除 QoS 策略关联

备份

事件	描述
ExportBackup	导出备份文件
ImportBackup	导入备份文件

5.4 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
InsufficientMemory	内存不足
ServiceRestartRequired	服务需重启

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
LUNResumed	卷恢复
LUNDataLowRedundancy	卷数据降级
LUNDataRedundancyResumed	卷数据降级恢复
LUNDataAccessResumed	卷数据访问恢复
LUNDataAccessFailed	卷数据无法访问

目标（iSCSI 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
DataLowRedundancy	数据降级
DataRedundancyResumed	数据降级恢复
DataResumed	数据恢复
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
DataAccessResumed	数据访问恢复
ProTrialActive	开启高级功能试用

故障域模块（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
PoolDataLowRedundancy	存储池数据降级
PoolDataRedundancyResumed	存储池数据降级恢复
PoolDataAccessResumed	存储池数据访问恢复
PoolDataAccessFailed	存储池数据无法访问

5.5 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 20 秒采集一次实时数据，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。	B/s	精细
	R_Bandwidth	客户端从 HBlock 读取数据的带宽。	B/s	精细
	W_Bandwidth	客户端向 HBlock 写入数据的带宽。	B/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	系统中所有数据目录所在磁盘总容量。	bytes	粗糙
	Path_Used	系统中所有数据目录所在磁盘已用	bytes	粗糙

		容量。		
	Path_Rate	系统的数据目录所在磁盘平均使用率，即 $\text{Path\_Used}/\text{Path\_Cap}$ 。	%	粗糙
	Path_Cap_Quota	系统中 HBlock 可用空间大小，即用户给 HBlock 分配的所有数据目录容量配额的总和。	bytes	粗糙
	Path_Cap_Quota_Used	系统中所有数据目录所在磁盘中，HBlock 数据占用的空间大小。	bytes	粗糙
	Path_Cap_Quota_Rate	系统的数据目录容量配额平均使用率，即 $\text{Path\_Cap\_Quota\_Used}/\text{Path\_Cap\_Quota}$ 。	%	粗糙
	Cloud_Bandwidth	HBlock 与云之间的总带宽。	B/s	精细
	Cloud_U_Bandwidth	HBlock 向云上传数据的带宽。	B/s	精细
server	Cloud_D_Bandwidth	HBlock 从云下载数据的带宽。	B/s	精细
	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。	bytes	精细
	Mem_Used	服务器内存使用量。	bytes	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。	B/s	精细
	R_Bandwidth	客户端从 HBlock 读取数据的带宽。	B/s	精细
	W_Bandwidth	客户端向 HBlock 写入数据的带宽。	B/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	HBlock 服务器的所有数据目录所在磁盘总容量。	bytes	粗糙
	Path_Used	HBlock 服务器的所有数据目录所在磁盘已用容量。	bytes	粗糙
	Path_Rate	HBlock 服务器的数据目录所在磁盘平均使用率，即 $\text{Path\_Used}/\text{Path\_Cap}$ 。	%	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。	bytes	粗糙
	Path_Cap_Quota_Used	HBlock 服务器所有数据目录所在磁盘中，HBlock 数据占用的空间大小。	bytes	粗糙
	Path_Cap_Quota_Rate	HBlock 服务器的数据目录容量配额平均使用率，即 $\text{Path\_Cap\_Quota\_Used}/\text{Path\_Cap\_Quota}$ 。	%	粗糙
	Cloud_Bandwidth	HBlock 服务器与云之间的总带宽。	B/s	精细

	Cloud_U_Bandwidth	HBlock 服务器向云上传数据的带宽。	B/s	精细
	Cloud_D_Bandwidth	HBlock 服务器从云下载数据的带宽。	B/s	精细
disk	Path_Cap	数据目录所在磁盘总容量。	bytes	粗糙
	Path_Used	数据目录所在磁盘已用容量。	bytes	粗糙
	Path_Rate	数据目录所在磁盘使用率。	%	粗糙
	Path_Cap_Quota	HBlock 可用空间大小，即用户给 HBlock 分配的容量配额。	bytes	粗糙
	Path_Cap_Quota_Used	数据目录所在磁盘，HBlock 数据占用的空间大小。	bytes	粗糙
	Path_Cap_Quota_Rate	数据目录容量配额使用率，即 $\text{Path\_Cap\_Quota\_Used} / \text{Path\_Cap\_Quota}$ 。	%	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。	B/s	精细
	R_Bandwidth	客户端从 HBlock 卷读取数据的带宽。	B/s	精细
	W_Bandwidth	客户端向 HBlock 卷写入数据的带宽。	B/s	精细
	Latency	客户端与 HBlock 之间的总时延，卷在一个采集周期内，读写操作平均	ms	精细

		时延，反映 HBlock 单卷处理读写请求的时长。		
	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。	B/s	精细
	Cloud_U_Bandwidth	HBlock 卷向云上传数据的带宽。	B/s	精细
	Cloud_D_Bandwidth	HBlock 卷从云下载数据的带宽。	B/s	精细
	Wait_Upload	待从 HBlock 卷上传云的数据量。	bytes	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。	B/s	精细
	R_Bandwidth	客户端从 HBlock 读取数据的带宽。	B/s	精细
	W_Bandwidth	客户端向 HBlock 写入数据的带宽。	B/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，	ms	精细

		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。		
	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的总容量。	bytes	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的已用容量。	bytes	粗糙
	Path_Rate	存储池的数据目录平均使用率，即 $\text{Path\_Used}/\text{Path\_Cap}$ 。	%	粗糙
	Path_Cap_Quota	HBlock 可用空间大小，即存储池中所有数据目录所在磁盘中，用户给 HBlock 分配的容量配额的总和。	bytes	粗糙
	Path_Cap_Quota_Used	存储池中所有数据目录所在磁盘中，HBlock 数据占用的空间大小。	bytes	粗糙
	Path_Cap_Quota_Rate	存储池的数据目录容量配额平均使用率，即 $\text{Path\_Cap\_Quota\_Used}/\text{Path\_Cap\_Quota}$ 。	%	粗糙

5.6 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 - 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 - 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 - 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	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	- 许可证过期 - 导入新的许可证（不同 id 的许可证）	是	每天 1 次

		许可证的维保到期时间 ≤ 15 天且 > 0 天	的维保到期时间 > 15 天			
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 $\geq$ 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 - 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 $>$	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 $<$	- 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

		=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。	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。			
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使用率<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	- 卷被删除 - 卷删除失败 - 卷还原失败 - 卷挂起中 - 卷已挂起 - 卷挂起失败 - 存储池名称变更	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 - 数据目录从存储池中移除 - 数据目录健康状态变为	是	每天 1 次

				“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 - 故障域中的数据目录被全部移除	是	每天 1 次

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 ● 卷删除失败 ● 卷还原失败 ● 卷挂起失败 ● 卷已挂起 ● 服务器被移除， ● target 发生迁移	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 ● 卷删除失败 ● 卷还原失败 ● 卷挂起失败 ● 卷已挂起	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	● 卷被删除或 ● 卷删除失败 ● 卷还原失败 ● 卷挂起失败 ● 卷已挂起	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 ● 卷删除失败 ● 卷还原失败 ● 卷挂起失败 ● 卷已挂起	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	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	是	每天 1 次

		障，导致卷数据无法同步删除 ● 云端数据残留：删除云上数据时，由于云端数据存储位置无法访问（包括网络连接失败、账号异常等），导致数据无法同步删除		解除，清理完云端残留数据后，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 ≤4GiB	安装目录所在磁盘的文件系统剩余空间 >5GiB	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	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	告警机器上的基础服务迁移完成	是	每天 1 次

		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）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	● 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
卷数据健康度降级 LUNDataLowRedundancy	警告	卷数据降级的百分比（含缓存池和最终池数据）>10%，持续超过 10 分钟（10 分钟内降级百分比一直低于或等于 10%）	卷数据降级的百分比（含缓存池和最终池数据）=0%	● 卷被删除 ● 卷删除失败	是	每天 1 次
卷数据损坏 LUNDataCorrupted	重要	卷数据 Error 的百分比（含缓存池和最终池数据）>0%	卷数据 Error 的百分比（含缓存池和最终池数据）=0%	● 卷被删除 ● 卷删除失败	是	每天 1 次
存储池数据健康度降级 PoolDataLowRedundancy	重要	基础存储池数据降级的百分比 > 10%，持续超过 10 分钟（10 分钟内降级百分比一直低于或等于 10%）	基础存储池数据降级的百分比= 0%	存储池名称修改	是	每天 1 次
存储池数据健康度降级	警告	普通存储池数据降级	普通存储池数据降级	● 存储池名称修	是	每天 1 次

PoolDataLowRedundancy	告	的百分比 > 10%，持续超过 10 分钟（10 分钟内降级百分比一直低于或等于 10%）	的百分比 = 0%	改 ● 存储池被删除		次
存储池数据损坏 PoolDataCorrupted	严重	基础存储池数据 Error 的百分比 > 0%	基础存储池数据 Error 的百分比 = 0%	存储池名称修改	是	每天 1 次
存储池数据损坏 PoolDataCorrupted	重要	普通存储池数据 Error 的百分比 > 0%	普通存储池数据 Error 的百分比 = 0%	● 存储池名称修改 ● 存储池被删除	是	每天 1 次

5.7 OOS Endpoint 和 Region

对象存储网络、香港节点、其他区域的 Endpoint 不同。

5.7.1 对象存储网络

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。

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
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

5.7.2 对象存储网络 2

对象存储网络 2 的 OOS Endpoint: oos-cn2.ctyunapi.cn，region 为 cn2。

说明：对于对象存储网络 2 中的 OOS API，如果您的数据存储在某个资源池，建议您直接使用该资源池的 Endpoint。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
内蒙古 1	oos-nm1.ctyunapi.cn	nm1
杭州 1	oos-hz1.ctyunapi.cn	hz1

5.7.3 香港节点

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-cnkhk-hqnet.ctyunapi.cn，region 为 cnhk-hqnet；香港普通网 OOS Endpoint: oos-cnkhk-nqnet.ctyunapi.cn，region 为 cnhk-nqnet。

5.7.4 其他地域

说明：其他地域只支持 V2 签名，不支持 V4 签名。

其他地域的 Endpoint 列表如下。

地区	OOS Endpoint
北京 2	oos-bj2.ctyunapi.cn
内蒙 2	oos-nm2.ctyunapi.cn
杭州	oos-hz.ctyunapi.cn
江苏	oos-js.ctyunapi.cn
北京	oos-hq-bj.ctyunapi.cn
上海	oos-hq-sh.ctyunapi.cn

5.8 HBlock 可推送的操作系统监控指标

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_sec	IO 读失败累积耗时

	onds	
	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
status	hblock_status_system	系统整体状态： ● 0：表示 Working。 ● 1：表示 Upgrading。 ● 2：表示 Uninstalling。 ● 3：表示 Unknown。
	hblock_status_data	系统整体数据健康度，单位是%。
	hblock_status_license	许可证状态： ● 0：表示订阅有效。 ● 1：表示维保有效。 ● 2：表示未导入。 ● 3：表示维保过期。 ● 4：表示订阅过期。
	hblock_status_licenseExpiredTime	许可证过期时间，unix 时间戳。
	hblock_status_poolDomain	存储池中每个故障域状态： ● 0：表示 Healthy。

		● 1: 表示 Warning。 ● 2: 表示 Error。 ● 3: 表示 Unknown。
	hblock_status_baseService	基础类服务运行状态： ● 0 表示服务正常。 ● 1: 表示服务异常。 ● 2: 表示状态未知。
	hblock_status_dataService	数据服务运行状态： ● 0: 表示 Healthy。 ● 1: 表示 Warning。 ● 2: 表示 Error。 ● 3: 表示 Unknown。

5.9 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	类型	描述	是否必须
name	String	根节点名称。 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。 默认为 default。	否
type	String	根节点的类型。 取值：root：根节点。 默认为 root。	否
childNodes	Array of childNode	子节点信息集合，根据子节点类型不同，需要的参数不同，详见“表 1 请求参数 childNode 说明”。	是

表 1 请求参数 `childNodes` 说明

参数		类型	描述	是否必须
name		String	根节点名称。 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。 默认为 default。	否
type		String	根节点的类型。 取值：root：根节点。 默认为 root。	否
childNodes		Array of childNode	子节点信息集合，根据子节点类型不同，需要的参数不同，详见本表其他子节点的描述。	是
子节点类型为 room	name	String	子节点名称。 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。	是
	type	String	子节点类型。 取值：room：机房类型。	是
	description	String	子节点描述信息。 取值：1~50 位字符串。	否
	childNodes	Array of	子节点信息集合，根据子节点类型不	否

		childNodes	同，需要的参数不同，子节点可以为 rack 或者 server。	
子节点类型为 rack	name	String	子节点名称。 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。	是
	type	String	子节点类型。 取值：rack：机架类型。	是
	description	String	子节点描述信息。 取值：1~50 位字符串。	否
	childNodes	Array of childNode	子节点信息集合，子节点只能为 server，详见“子节点类型为 server”的描述。	否
子节点类型为 server	name	String	子节点名称。 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。 默认为服务器 ID。	否
	type	String	子节点类型。 取值：server：服务器类型。	是
	description	String	子节点描述信息。 取值：1~50 位字符串。	否
	ip	String	HBlock 的服务器 IP。	是

			取值：IPv4 或[IPv6]地址。	
	apiPort	Integer	管理 API 端口号。 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	Array of childNode	子节点信息集合。子节点的类型为 path，详见“子节点类型为 path”的描述。	是
子节点类型为 path	name	String	子节点名称。 取值：数据目录具体路径。	是
	type	String	子节点类型。 取值：path：数据目录类型。	是
	capacityQuota	Long	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。 取值：小于数据目录所在磁盘的总容量，单位是 bytes。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"  
  }  
]  
}  
]  
}  
]  
}
```