



# 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
3.8	2025 年 02 月 14 日	<ol style="list-style-type: none"><li>1. 存储卷和缓存卷支持将数据从后端上传到 AWS S3。</li><li>2. 支持设置基础服务的数据存储目录。</li><li>3. Target 增加回收策略，支持无卷关联后自动删除。</li><li>4. 针对智算、虚拟化以及高可用敏感度等场景，支持一键调整系统参数。</li></ol> <p>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。</p>
3.7	2024 年 08 月 08 日	<ol style="list-style-type: none"><li>1. 支持设置集群拓扑。</li><li>2. 支持创建和管理多存储池。</li><li>3. 支持机房和机架级别故障域。</li><li>4. 支持设置卷的高速缓存池。</li><li>5. 支持基础服务迁移。</li></ol>
3.6	2024 年 06 月 03 日	<ol style="list-style-type: none"><li>1. 支持存储卷和缓存卷，将数据从后端上传到 OOS。</li><li>2. 硬件及 HBlock 监控数据支持对接到 Prometheus。</li><li>3. 支持 HBlock 告警信息对接到智能运维平台。</li><li>4. 优化读写性能。</li><li>5. 增加对龙芯服务器的支持。</li></ol>
3.5	2024 年 03 月 04 日	<ol style="list-style-type: none"><li>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。</li><li>2. 支持指定基础服务的安装节点。</li><li>3. 支持数据目录配额，设置 HBlock 可写入的数据量上限。</li><li>4. Target 可被多个客户端发现并连接。</li><li>5. 设置卷的最小写入副本数，提高数据写入安全性。</li><li>6. 扩大纠删码 EC N+M 支持范围，满足 <math>N+M \leq 128</math>。</li></ol>
3.4	2023 年 07 月 12 日	<ol style="list-style-type: none"><li>1. 卷连接支持一主多备，提高业务可用性。</li><li>2. 支持 IPv6 环境。</li><li>3. 控制台提供 Dashboard 一页式概览。</li></ol>

		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	<ol style="list-style-type: none"> <li>“产品概述”章节新增“基本概念”。</li> <li>监控项增加，扩大覆盖范围。</li> <li>增加对告警、日志管理的支持。</li> <li>事件中增加对系统事件的支持。</li> <li>“附录”章节新增：系统事件列表、HBlock 进程、监控指标、告警列表。</li> </ol>
3.1	2022 年 06 月 14 日	<ol style="list-style-type: none"> <li>单机版支持添加多个数据目录。</li> <li>集群版支持创建 Target 时指定对应的服务器，支持 Target 迁移。</li> <li>支持用户事件的记录和查询。</li> </ol>
3.0	2022 年 01 月 18 日	<ol style="list-style-type: none"> <li>命令行变更为非交互式</li> <li>支持 WEB、API 调用方式。</li> <li>卷操作：支持设置卷的高可用类型和卷的写策略。</li> </ol>
2.1	2021 年 08 月 27 日	<ol style="list-style-type: none"> <li>增加对 ARM 服务器的支持。</li> <li>软件许可证：查看许可证时，可以显示允许的容量。</li> <li>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。</li> </ol>
2.0	2021 年 05 月 28 日	<ol style="list-style-type: none"> <li>支持集群版部署。</li> <li>支持多副本和纠删码数据冗余。</li> </ol>

# 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	5
1.3.1 iSCSI	5
1.3.2 卷	5
1.3.3 iSCSI Target	5
1.3.4 iSCSI 目标门户	5
1.3.5 存储池	6
1.3.6 集群拓扑	6
1.3.7 故障域	6
1.3.8 数据服务	6
1.3.9 监控	7
1.3.10 事件	7
1.3.11 日志采集	7
1.3.12 告警	7
1.4 术语与缩略语	9
2 服务器部署要求	10
2.1 环境要求	10
2.2 HBlock 配置环境 - 单机版	12
2.3 HBlock 配置环境 - 集群版	14
3 安装访问	17
3.1 安装	17
3.2 访问地址	19
3.3 接口	20
3.3.1 URL 规则	20
3.3.2 通用请求头	21
3.3.3 签名方法	22
3.3.4 通用响应头	23
3.3.5 通用错误码	24
4 管理 API	29
4.1 初始化	29

4.1.1	初始化	29
4.1.2	查询初始化进度	51
4.1.3	错误码列表	53
4.2	软件许可证	62
4.2.1	导入软件许可证	62
4.2.2	批量查询软件许可证信息	67
4.2.3	查询指定软件许可证信息	73
4.2.4	错误码列表	78
4.3	卷管理	80
4.3.1	创建卷	80
4.3.2	启用卷	102
4.3.3	禁用卷	103
4.3.4	删除卷	105
4.3.5	扩容卷	108
4.3.6	修改单个卷	110
4.3.7	设置卷的主备优先级或自动切换	118
4.3.8	触发卷对应 Target 的主备切换（集群版适用）	123
4.3.9	还原卷（上云卷）	126
4.3.10	恢复还原中断的卷（上云卷适用）	138
4.3.11	批量查询卷	140
4.3.12	查询指定卷	154
4.3.13	错误码列表	167
4.4	iSCSI Target 管理	174
4.4.1	创建 iSCSI Target	174
4.4.2	修改 iSCSI Target 的 CHAP 认证	181
4.4.3	迁移 iSCSI Target（集群版适用）	184
4.4.4	修改 iSCSI Target 下每个 IQN 允许建立的最大会话数	189
4.4.5	修改 iSCSI Target 的回收策略	191
4.4.6	删除 iSCSI Target	193
4.4.7	删除 CHAP	195
4.4.8	批量查询 iSCSI Target	196
4.4.9	查询指定 iSCSI Target	208
4.4.10	查询 iSCSI Target 连接	214
4.4.11	删除 iSCSI Target 连接	217
4.4.12	错误码列表	219
4.5	存储池（集群版）	223
4.5.1	创建存储池	223
4.5.2	添加节点到存储池	225

4.5.3	修改存储池	228
4.5.4	移除存储池内的节点	230
4.5.5	删除非基础存储池	233
4.5.6	批量查询存储池	235
4.5.7	查询指定存储池	245
4.5.8	错误码列表	253
4.6	集群拓扑（集群版）	259
4.6.1	创建拓扑节点	259
4.6.2	修改拓扑节点信息	262
4.6.3	删除拓扑节点	264
4.6.4	查询整个拓扑	266
4.6.5	查询拓扑节点	275
4.6.6	错误码列表	282
4.7	服务器管理	285
4.7.1	添加服务器（集群版适用）	285
4.7.2	修改服务器属性	291
4.7.3	删除服务器属性	296
4.7.4	迁移服务器上的基础服务（集群版适用）	297
4.7.5	移除服务器（集群版适用）	300
4.7.6	重启服务器上的HBlock	304
4.7.7	批量查询服务器	306
4.7.8	查询指定服务器	319
4.7.9	添加数据目录	331
4.7.10	修改数据目录的容量配额	334
4.7.11	移除数据目录	337
4.7.12	错误码列表	340
4.8	查询HBlock信息	349
4.9	查询HBlock服务状态	352
4.10	监控	359
4.10.1	获取实时监控数据	359
4.10.2	获取历史性能数据	367
4.10.3	错误码列表	377
4.11	告警	378
4.11.1	批量查询HBlock告警信息	378
4.11.2	查询指定的HBlock告警	392
4.11.3	手动解除HBlock告警	397
4.11.4	静默告警/解除告警静默	399
4.11.5	导出告警	402

4. 11. 6 错误码列表 .....	404
4. 12 事件和日志管理 .....	405
4. 12. 1 查看事件 .....	405
4. 12. 2 导出事件 .....	410
4. 12. 3 发起 HBlock 日志采集 .....	413
4. 12. 4 批量查询采集的日志 .....	418
4. 12. 5 查询单个采集的日志 .....	423
4. 12. 6 下载采集的日志文件 .....	426
4. 12. 7 删除指定的采集日志 .....	428
4. 12. 8 删除所有的采集日志 .....	429
4. 12. 9 错误码列表 .....	430
4. 13 HBlock 系统设置 .....	432
4. 13. 1 修改管理员密码 .....	432
4. 13. 2 邮件设置 .....	434
4. 13. 3 远程协助 .....	445
4. 13. 4 pushgateway 监控配置 .....	453
4. 13. 5 智维平台推送告警配置 .....	477
4. 14 调整 HBlock 性能参数（集群版适用） .....	487
4. 14. 1 调整 HBlock 性能参数 .....	487
4. 14. 2 查看性能调优配置 .....	489
4. 14. 3 错误码 .....	491
4. 15 查询 HBlock 版本 .....	492
4. 16 升级 HBlock .....	494
4. 16. 1 升级 HBlock .....	494
4. 16. 2 查询升级进度 .....	497
4. 16. 3 错误码 .....	502
5 附录 .....	506
5. 1 转义字符表 .....	506
5. 2 HBlock 服务 .....	507
5. 3 用户事件列表 .....	508
5. 4 系统事件列表 .....	512
5. 5 监控指标 .....	516
5. 6 告警列表 .....	523
5. 7 OOS Endpoint 和 Region .....	528
5. 8 HBlock 可推送的操作系统监控指标 .....	529
5. 9 集群拓扑文件 .....	534

# 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中进行了充分的测试。

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可以与天翼云对象存储（经典版）I型（Object-Oriented Storage，简称OOS）和AWS S3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 利用了对象存储的原子操作，能够真正确保云上数据的一致性（即任何时候云上数据都是本地数据的一个快照），不会出现因云上数据的不一致而导致无法恢复整个业务的情况，从而保证数据安全。

## 1.2 应用场景

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

## 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 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。

### 1.3.4 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。

### 1.3.5 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。

### 1.3.6 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。

### 1.3.7 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。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。

### 1.3.8 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。

### 1.3.9 监控

监控是指对 HBlock 系统、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**。

### 1.3.10 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**。

### 1.3.11 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。

### 1.3.12 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。

告警列表详见附件告警列表。

## 1.4 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, CTyunOS 3。64 位操作系统。 ARM 架构的硬件环境下，推荐使用 PageSize 为 4k 的操作系统。
硬件	x86 服务器、ARM 服务器或者龙芯服务器。 最低配置：单核 CPU、2GB 内存。可根据实际业务需要增加配置。
带宽	<ul style="list-style-type: none"> <li>● 客户端到 HBlock 的带宽：读写带宽能力大于业务读写带宽。</li> <li>● 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。</li> <li>● 上云带宽大于业务写入带宽。</li> </ul>
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	<ul style="list-style-type: none"> <li>● 最小配置：5GB，可根据实际业务需要增加配置。</li> <li>● 根据存储容量和副本模式配置数据目录对应分区的容量。</li> </ul> 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	<ul style="list-style-type: none"> <li>● 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）。</li> <li>● 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。</li> </ul> 网络整体架构如下： <ol style="list-style-type: none"> <li>1. HBlock 内部各节点之间通过内网互联。</li> <li>2. HBlock 与上层应用之间通过内网或专线或公网互联。</li> </ol>

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）。

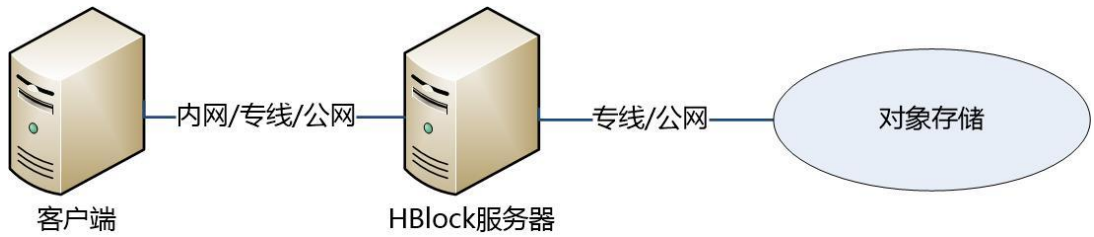


图1. 单机版网络拓扑图

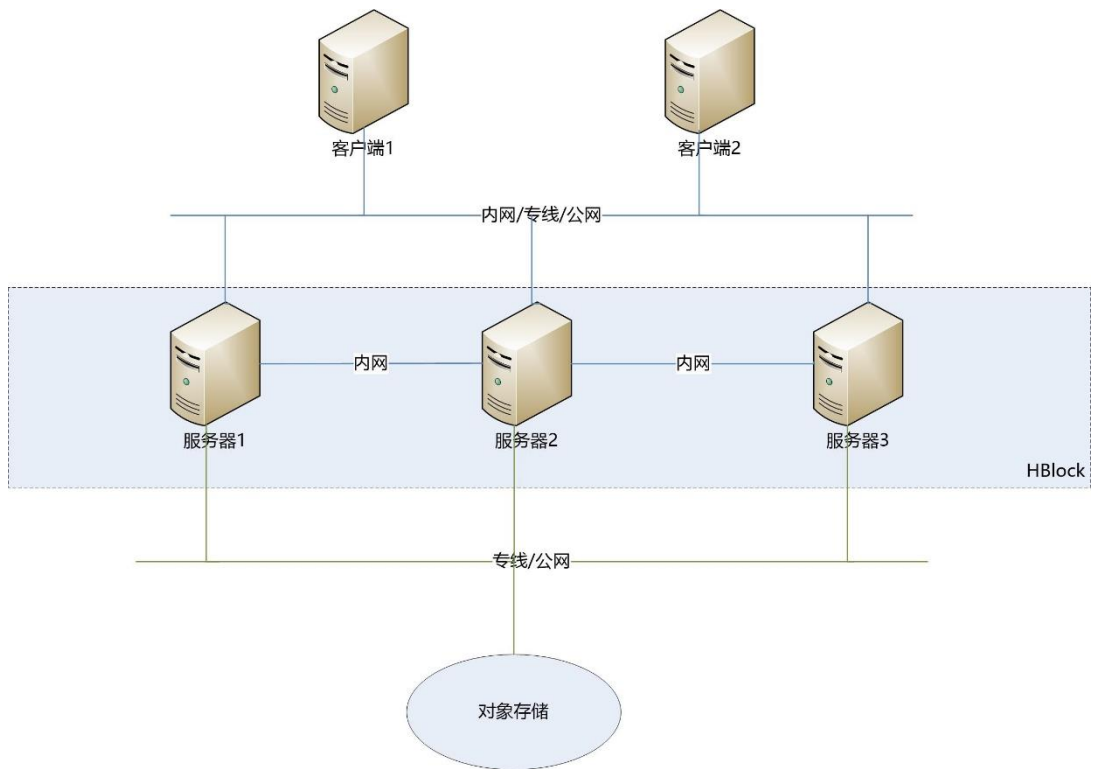


图2. HBlock 集群版网络拓扑图

**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。

(三) 关闭 selinux 和 swap 分区（建议）

(四) 防火墙设定

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

1. 开启 iSCSI 端口，如 iSCSI 端口为 3260 时：

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

## 2. 重新加载防火墙使配置生效:

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

### (五) 设置资源限制

修改配置文件/etc/security/limits.conf, 在配置文件中增加下列内容, 设置在 *domain* 中打开的最大文件数。

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

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

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。

```
* soft nofile 65536 # *为参数 domain 的取值
* hard nofile 65536 # *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。

```
root soft nofile 65536 # root 为参数 domain 的取值
root hard nofile 65536 # 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。

示例如下：

1. 开启 iSCSI 端口，如 iSCSI 端口为 3260 时：

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

2. 集群各服务器的 IP 添加白名单：

- 添加 IPv4 地址

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

- 添加 IPv6 地址

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

3. 重新加载防火墙使配置生效：

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

## (五) 设置资源限制

修改配置文件 `/etc/security/limits.conf`，在配置文件中增加下列内容，设置在 `domain` 中打开的最大文件数。

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

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

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。

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

```
root soft nfile 65536 # root 为参数 domain 的取值  
root hard nfile 65536 # 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_3.8.0_x64.zip
cd CTYUN_HBlock_Plus_3.8.0_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_3.8.0_x64.zip
.....
[root@hblockserver opt]# cd CTYUN_HBlock_Plus_3.8.0_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_3.8.0_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，目前版本号只能为 v1。	是
prefix	前缀，取值为： <ul style="list-style-type: none"> <li>● system: 系统相关接口。</li> <li>● block: 块操作相关接口。</li> </ul>	是
resourceName	资源名称。	是
name	具体实例的名称或 ID。	否

### 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： <ul style="list-style-type: none"> <li>● keep-alive: 持续连接。</li> <li>● close: 关闭连接。</li> </ul> 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： <i>EEE, dd MMM yyyy HH:mm:ss 'GMT'</i> 。	是
Content-Type	请求体MIME类型，取值为： <code>application/json; charset=utf-8</code> 。	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： <code>HBlock+ " "+userName+": "+signature</code> 。	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： <ul style="list-style-type: none"><li>● <b>keep-alive</b>: 持续连接。</li><li>● <b>close</b>: 关闭连接。</li></ul>
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。 <b>backoff 说明</b> : 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。



			<p>加载配置文件需要满足以下条件：</p> <ul style="list-style-type: none"> <li>● 需要基础服务器至少有 2 台满足 cs 和 ls 服务成功启动。</li> <li>● 需要基础服务器至少有 1 台满足 fc、mdm 和 ds 服务成功启动。</li> <li>● 需要保证待启动的服务器和主服务器（Master）网络通信正常。</li> </ul>
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.	系统中存在低冗余或错误数据，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 LUN status of ' <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 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,
"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,
  "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,
  "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	描述	是否必须
storName	HBlock 名称。 类型：字符串 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	HBlock 的管理员用户名。 类型：字符串 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	设置的新密码。初始化时必须修改密码。 类型：字符串 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
iSCSIPort	iSCSI 端口号。 类型：整型 取值：[1, 65535]，默认值为 3260。	否

<p>portRange</p>	<p>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。</p> <p>类型：整型</p> <p>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>&lt;<i>port2</i>。<i>port1</i> 默认取值为 20000，<i>port2</i> 默认取值为 20500。</p> <p>说明：建议指定的端口范围至少包含 500 个端口。</p>	<p>否</p>
<p>faultDomain</p>	<p>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。</p> <p>类型：枚举</p> <p>取值：</p> <ul style="list-style-type: none"> <li>● room: 机房级别。</li> <li>● rack: 机架级别。</li> <li>● path: 数据目录级别。</li> <li>● server: 服务器级别。</li> </ul> <p>默认值为 server。</p> <p>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。</p>	<p>否</p>
<p>ports.dataPort1</p>	<p>数据端口 1（仅集群版支持）。</p> <p>类型：整型</p> <p>取值：[1, 65535]。</p>	<p>否</p>
<p>ports.managementPort1</p>	<p>管理服务端口 1。</p> <p>类型：整型</p> <p>取值：[1, 65535]。</p>	<p>否</p>
<p>ports.managementPort2</p>	<p>管理服务端口 2。</p>	<p>否</p>

	类型：整型 取值：[1, 65535]。	
ports.managementPort3	管理服务端口 3。 类型：整型 取值：[1, 65535]。	否
ports.managementPort4	管理服务端口 4。 类型：整型 取值：[1, 65535]。	否
ports.managementPort5	管理服务端口 5（仅集群版支持）。 类型：整型 取值：[1, 65535]。	否
ports.managementPort6	管理服务端口 6。 类型：整型 取值：[1, 65535]。	否
ports.metadataPort1	元数据端口 1（仅集群版支持）。 类型：整型 取值：[1, 65535]。	否
ports.metadataPort2	元数据端口 2（仅集群版支持）。 类型：整型 取值：[1, 65535]。	否
ports.metadataPort3	元数据端口 3（仅集群版支持）。 类型：整型 取值：[1, 65535]。	否
ports.metadataPort4	元数据端口 4（仅集群版支持）。 类型：整型 取值：[1, 65535]。	否
ports.metadataPort5	元数据端口 5（仅集群版支持）。	否

	类型：整型 取值：[1, 65535]。	
ports.metadataPort6	元数据端口 6（仅集群版支持）。 类型：整型 取值：[1, 65535]。	否
ports.metadataPort7	元数据端口 7（仅集群版支持）。 类型：整型 取值：[1, 65535]。	否
ports.metadataPort8	元数据端口 8（仅集群版支持）。 类型：整型 取值：[1, 65535]。	否
topology	导入集群拓扑文件内容（仅集群版支持）。 类型：字符串 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 <b>集群拓扑文件</b> 。 <b>注意：</b> “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
servers	HBlock 服务器的属性集合，包含：nodeName（仅集群版支持）、ip、apiPort、diskPaths。 <b>注意：</b> “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。 类型：数组	否
nodeName	拓扑节点名称。 类型：字符串	否

	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。	
ip	<p>HBlock 的服务器 IP。</p> <p>取值：IPv4 或[IPv6]地址。</p>	是
apiPort	<p>管理 API 端口号。</p> <p>类型：整型</p> <p>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。</p>	否
diskPaths	<p>数据目录属性集合。包括 path、capacityQuota。</p> <p>类型：数组</p>	否
path	<p>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。数据目录中不能有逗号(,)。</p> <p><b>说明：</b>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。</p> <p><b>注意：</b>如果故障域是 server 和 path 级别，此项必填。</p> <p>类型：字符串列表</p>	否
capacityQuota	<p>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。</p> <p>类型：长整型</p>	否

	<p>取值：小于数据目录所在磁盘的总容量，单位是字节。负整数表示无限制写入，0 表示禁止写入。默认不限制写入。</p> <p><b>注意：</b>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。</p>	
services	<p>服务地址的集合（仅集群版支持）。包括 mdm、ls、cs。</p> <p>类型：数组</p>	否
mdm.ip	<p>元数据管理服务的地址（仅集群版支持）。如果填写，必须填写集群中的 2 个 IP 地址。如果不填写，元数据管理服务的地址默认为集群中的前 2 个 IP 地址。</p> <p>取值：IPv4 或[IPv6]地址。</p>	否
mdm.metaDir	<p>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。</p> <p><b>说明：</b>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。</p> <p>类型：字符串</p> <p>取值：数据目录名不能含逗号。默认目录为安装目录。</p>	否
ls.ip	<p>日志服务的地址（仅集群版支持）。如果填写，必须填写集群中的 3 个 IP 地址。日志服务的地址默认为集群中的前 3 个 IP 地址。</p> <p>取值：IPv4 或[IPv6]地址。</p>	否
ls.metaDir	<p>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。</p>	否



	<p><b>说明：</b> 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。</p> <p>类型：字符串</p> <p>取值：数据目录名不能含逗号。默认目录为安装目录。</p>	
cs.ip	<p>协调服务的地址（仅集群版支持）。如果填写，必须填写集群中的 3 个 IP 地址。协调服务的地址默认为集群中的前 3 个 IP 地址。</p> <p>取值：IPv4 或[IPv6]地址。</p>	否
cs.metaDir	<p>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。</p> <p><b>说明：</b> 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。</p> <p>类型：字符串</p> <p>取值：数据目录名不能含逗号。默认目录为安装目录。</p>	否
clusterNetwork	<p>集群网络（仅集群版支持），用于集群间的数据通信。</p> <ul style="list-style-type: none"> <li>● 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。</li> <li>● 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。</li> </ul> <p>类型：CIDR 格式</p>	否

	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。	
publicNetwork	<p>业务网络，用于客户端和 HBlock 之间的数据传输式。</p> <ul style="list-style-type: none"> <li>● 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。</li> <li>● 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。</li> </ul> <p>类型：CIDR 格式</p> <p>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。</p>	否

● 请求示例 1

单机版：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

集群版：输入服务器 IP，初始化。

```
POST /rest/v1/system/setup HTTP/1.1  
Date: Wed, 24 Jan 2024 06:59:48 GMT  
Content-Type: application/json; charset=utf-8  
Content-Length: 1344
```

```
Host: 192.168.0.110:1443
Authorization: HBlock storuser:signature

{
  "storName": "stor1",
  "userName": "storuser1",
  "newPassword": "*****",
  "iSCSIPort": 3260,
  "portRange": "20000-20500",
  "faultDomain": "path",
  "servers": [
    {
      "ip": "192.168.0.110",
      "apiPort": 1443,
      "diskPaths": [
        {
          "path": "/mnt/storage01",
          "capacityQuota": 96636764160
        }
      ]
    },
    {
      "ip": "192.168.0.192",
      "apiPort": 1443,
      "diskPaths": [
        {
          "path": "/mnt/stor",
          "capacityQuota": 96636764160
        }
      ]
    },
    {
      "ip": "192.168.0.102",
      "apiPort": 1443,
      "diskPaths": [
```

```
        {
            "path": "/mnt/stor",
            "capacityQuota": 96636764160
        }
    ]
}
],
"services": {
    "mdm": [
        "192.168.0.110",
        "192.168.0.192"
    ],
    "ls": [
        "192.168.0.110",
        "192.168.0.192",
        "192.168.0.102"
    ],
    "cs": [
        "192.168.0.110",
        "192.168.0.192",
        "192.168.0.102"
    ]
}
}
```

### ● 响应示例 2

```
HTTP/1.1 202 Accepted
x-hblock-request-id: 096e457a5c26465590ff8f312a493f2c
Connection: keep-alive
Date: Wed, 24 Jan 2024 06:59:48 GMT
Content-Length: 0
Server: HBlock
```

### ● 请求示例 3



```

\path"\r\n
    }\r\n
    ]\r\n
    }\r\n
    ]\r\n}",
    "services": {
      "mdm": [
        "192.168.0.110",
        "192.168.0.192"
      ],
      "ls": [
        "192.168.0.110",
        "192.168.0.192",
        "192.168.0.102"
      ],
      "cs": [
        "192.168.0.110",
        "192.168.0.192",
        "192.168.0.102"
      ]
    }
}
    
```

**说明：**拓扑文件内容为转换为字符串后的符合 UTF-8 编码格式的 JSON 文件。拓扑文件的源码如下。

```

{
  "name": "default",
  "childNodes": [
    {
      "name": "room1",
      "type": "room",
      "childNodes": [
        {
          "type": "server",
          "name": "server1",
          "ip": "192.168.0.192",
          "apiPort": 1443,
          "childNodes": [
            {
            
```

```
        "name": "/mnt/stor",
        "type": "path"
    },
    {
        "name": "/mnt/storage01",
        "type": "path"
    }
]
},
{
    "type": "server",
    "name": "server2",
    "ip": "192.168.0.110",
    "apiPort": 1443,
    "childNodes": [
        {
            "name": "/mnt/stor",
            "type": "path"
        }
    ]
},
{
    "type": "server",
    "name": "server3",
    "ip": "192.168.0.102",
    "apiPort": 1443,
    "childNodes": [
        {
            "name": "/mnt/stor",
            "type": "path"
        }
    ]
}
]
}
]
```

- 响应示例 3



```
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
```

## 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	描述
progress	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	初始化状态： <ul style="list-style-type: none"> <li>● Uninitialized: 未初始化。</li> <li>● Processing: 初始化中。</li> <li>● Succeeded: 初始化成功。</li> <li>● Failed: 初始化失败。</li> </ul>

### ● 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有一个数据目录。

400	InsufficientPorts	Insufficient available ports, need to increase the port range by $N$ ports.	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 network</i> ' 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 network</i> 和服务器 IP 不匹配。
400	InvalidClusterNetwork	Value <i>clusterNetwork</i> at ' <i>cluster network</i> ' 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>;</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>	初始化过程中，因数据目录有问题而导致的失败。

		<p><i>serverIP:diskPaths[,diskPaths...]</i>].</p> <p>The disk paths can not access:</p> <p><i>serverIP:diskPaths</i> <i>[,diskPaths...]</i> [;</p> <p><i>serverIP:diskPaths[,diskPaths...]</i>].</p> <p>The disk paths must be directory: <i>serverIP:diskPaths</i> <i>[,diskPaths...]</i> [;</p> <p><i>serverIP:diskPaths[,diskPaths...]</i>].</p> <p>The disk paths include incompatible data:</p> <p><i>serverIP:diskPaths</i> <i>[,diskPaths...]</i> [;</p> <p><i>serverIP:diskPaths[,diskPaths...]</i>].</p> <p>The disk paths can not contain commas:</p> <p><i>serverIP:diskPaths</i> <i>[,diskPaths...]</i> [;</p> <p><i>serverIP:diskPaths[,diskPaths...]</i>].</p>	
400	InvalidDiskPathCapacityQuota	<p>The capacity quota for the following disk paths failed to satisfy constraint. It must be</p>	<p>容量配额参数错误。</p>

		integer and not greater than the total capacity of disk path. <i>serverIP:diskPaths</i> [, <i>diskPaths</i> ...] [; <i>serverIP:diskPaths</i> [, <i>diskPaths</i> ...].]. ].	
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	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

		letters, digits, special symbols (~!@#\$%^&*()_+[]{} ;:./<>?), and must not contain any of the following: 3 consecutive repeating characters, 3 consecutive or in-reverse order of numbers or letters (case-insensitive), 3 consecutive or in-reverse order of keyboard sequences (case-insensitive)	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)。
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	节点类型的包含关系错误。room 的



		of room are limited to [rack, server], childNodes of rack to [server], childNodes of server to [path] only.	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 ' <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	InvalidPortRange	Value <i>value</i> at ' <i>port range</i> ' failed to satisfy constraint: Argument must satisfy pattern ' <i>port1-port2</i> ', where port1, port2 are positive integers between 1 and 65535, port1 is less than port2.	参数必须是“port1-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 ' <i>public network</i> ' 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 should 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	HBlock 名称不合法。

		can only contain letters, digits, hyphens(-) or underscores(_), and does not exceed 64 characters, must begin with a letter or digit.	
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'/'0:0:0:0:0:0:0:1'.	当 server ip 包含 localhost、127.0.0.1 或 0:0:0:0:0:0:0:1 时，集群网不能为空。
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	'operation' 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,	服务器初始化失败，请检查防火

		network, memory, diskpath, etc, or contact technical team for support.	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 导入软件许可证

此操作用来导入软件许可证。

#### ● 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	描述	是否必须
key	软件许可证密钥。	是

#### ● 响应结果

名称	描述
licenseId	软件许可证 ID。
account	软件许可证所属的账号。
type	软件许可证的购买类型： <ul style="list-style-type: none"> <li>● Subscription: 订阅模式。</li> <li>● Perpetual: 永久许可模式。</li> </ul>
status	软件许可证的状态：

	<ul style="list-style-type: none"> <li>● <b>Effective:</b> 已生效。</li> <li>● <b>Expired:</b> 已过期。</li> <li>● <b>Invalid:</b> 已失效。</li> </ul> <p>如果软件许可证是 <b>Perpetual</b> 类型的，不存在过期状态。</p>
<b>maximumLocalCapacity</b>	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。
<b>records</b>	<p>软件许可证购买记录的集合，包含：<b>purchaseTime</b>、<b>operationtype</b>、<b>localCapacity</b>、<b>subscribeEffectiveTime</b>、<b>subscribeExpireTime</b>、<b>maintenanceEffectiveTime</b>、<b>maintenanceExpireTime</b>、<b>substatus</b>。</p> <p>类型：数组</p>
<b>purchaseTime</b>	记录购买的时间，unix 时间戳（UTC），精确到毫秒。
<b>operationtype</b>	<p>软件许可证的购买操作记录：</p> <ul style="list-style-type: none"> <li>● <b>New:</b> 首次购买。</li> <li>● <b>Expand:</b> 扩容。</li> <li>● <b>Renew:</b> 续订/续保。</li> </ul>
<b>localCapacity</b>	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。
<b>subscribeEffectiveTime</b>	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。
<b>subscribeExpireTime</b>	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。
<b>maintenanceEffectiveTime</b>	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。
<b>maintenanceExpireTime</b>	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	本次购买操作的当前生效状态： <ul style="list-style-type: none"> <li>● NotStart: 未生效。</li> <li>● Effective: 已生效。</li> <li>● Expired: 已过期。</li> </ul>
usages	软件许可证的使用记录的集合，包含： maximumLocalCapacity、subscribeEffectiveTime、 subscribeExpireTime、maintenanceEffectiveTime、 maintenanceExpireTime、maintenanceExpireTime、 substatus。 类型：数组
maximumLocalCapacity	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	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	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	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	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	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： <ul style="list-style-type: none"> <li>● NotStart: 未生效。</li> <li>● Effective: 已生效。</li> <li>● Expired: 已过期。</li> </ul>

- 请求示例

导入软件许可证。

```
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	描述	是否必须
filter	设置查询的过滤条件。支持的查询类型包括： <ul style="list-style-type: none"> <li>● licenseId: 软件许可证 ID。</li> <li>● status: 软件许可证的状态：                             <ul style="list-style-type: none"> <li>➢ Effective: 已生效。</li> <li>➢ Expired: 已过期。</li> <li>➢ Invalid: 无效。</li> </ul> </li> <li>● type: 软件许可证的购买类型：                             <ul style="list-style-type: none"> <li>➢ Subscription: 订阅模式。</li> <li>➢ Perpetual: 永久许可模式。</li> </ul> </li> </ul>	否
range	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。	否

	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 <code>range</code> 参数，那么返回所有软件许可证。	
<code>show</code>	查询的输出结果，支持的输出结果包括 <code>record, usage</code> 。 如果不填写，就是默认两个信息都输出。	否

### ● 响应结果

名称	描述
<code>licenseId</code>	软件许可证 ID。
<code>account</code>	软件许可证所属的账号。
<code>type</code>	软件许可证的购买类型： <ul style="list-style-type: none"> <li>● <b>Subscription</b>: 订阅模式。</li> <li>● <b>Perpetual</b>: 永久许可模式。</li> </ul>
<code>status</code>	软件许可证的状态： <ul style="list-style-type: none"> <li>● <b>Effective</b>: 已生效。</li> <li>● <b>Expired</b>: 已过期。</li> <li>● <b>Invalid</b>: 已失效。</li> </ul> 如果软件许可证是 <b>Perpetual</b> 类型的，不存在过期状态。
<code>maximumLocalCapacity</code>	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。
<code>records</code>	软件许可证购买记录的集合，包含： <code>purchaseTime</code> 、 <code>operationtype</code> 、 <code>localCapacity</code> 、 <code>subscribeEffectiveTime</code> 、 <code>subscribeExpireTime</code> 、 <code>maintenanceEffectiveTime</code> 、 <code>maintenanceExpireTime</code> 、 <code>substatus</code> 。 类型：数组
<code>purchaseTime</code>	记录购买的时间，unix 时间戳（UTC），精确到毫秒。

operationtype	软件许可证的操作记录： <ul style="list-style-type: none"> <li>● <b>New</b>: 首次购买。</li> <li>● <b>Expand</b>: 扩容。</li> <li>● <b>Renew</b>: 续订/续保。</li> </ul>
localCapacity	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	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	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	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	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	本次购买操作的当前生效状态： <ul style="list-style-type: none"> <li>● <b>NotStart</b>: 未生效。</li> <li>● <b>Effective</b>: 已生效。</li> <li>● <b>Expired</b>: 已过期。</li> </ul>
usages	软件许可证的使用记录的集合，包含： maximumLocalCapacity、subscribeEffectiveTime、 subscribeExpireTime、maintenanceEffectiveTime、 maintenanceExpireTime、substatus。 类型：数组
maximumLocalCapacity	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	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	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	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	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	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： <ul style="list-style-type: none"> <li>● NotStart: 未生效。</li> <li>● Effective: 已生效。</li> <li>● Expired: 已过期。</li> </ul>

### ● 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: Thu, 17 Mar 2022 09:40:42 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.121:1443
```

### ● 响应示例

```
HTTP/1.1 200 OK
x-hblock-request-id: 453c0e4d909b4cb4801759b89f490bbc
Connection: keep-alive
Content-Length: 671
Date: Thu, 17 Mar 2022 09:40:42 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
```

```
"data": {
  "licenses": [
    {
      "licenseId": "ehc2b6a9-f3fb-4098-a6b3-3652a5d76269",
      "account": "test@ctyun.cn",
      "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	描述	是否必须
licenseId	软件许可证的 ID。	否
show	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	描述
licenseId	软件许可证 ID。
account	软件许可证所属的账号。
type	软件许可证的购买类型： <ul style="list-style-type: none"> <li>● Subscription: 订阅模式。</li> <li>● Perpetual: 永久许可模式。</li> </ul>
status	软件许可证的状态：



	<ul style="list-style-type: none"> <li>● <b>Effective:</b> 已生效。</li> <li>● <b>Expired:</b> 已过期。</li> <li>● <b>Invalid:</b> 已失效。</li> </ul> 如果软件许可证是 <b>Perpetual</b> 类型的，不存在过期状态。
<b>maximumLocalCapacity</b>	当前时间允许的最大本地卷容量，单位 <b>TiB</b> 。如果不限容量，不返回此项。
<b>records</b>	软件许可证的购买记录的集合，包含： <b>purchaseTime</b> 、 <b>operationtype</b> 、 <b>localCapacity</b> 、 <b>subscribeEffectiveTime</b> 、 <b>subscribeExpireTime</b> 、 <b>maintenanceEffectiveTime</b> 、 <b>maintenanceExpireTime</b> 、 <b>substatus</b> 。 类型：数组
<b>purchaseTime</b>	记录购买的时间， <b>unix</b> 时间戳（ <b>UTC</b> ），精确到毫秒。
<b>operationtype</b>	软件许可证的操作记录： <ul style="list-style-type: none"> <li>● <b>New:</b> 首次购买。</li> <li>● <b>Expand:</b> 扩容。</li> <li>● <b>Renew:</b> 续订/续保。</li> </ul>
<b>localCapacity</b>	本次购买对应的本地卷容量，单位 <b>TiB</b> 。如果不限容量，不返回此项。
<b>subscribeEffectiveTime</b>	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， <b>unix</b> 时间戳（ <b>UTC</b> ），精确到毫秒。
<b>subscribeExpireTime</b>	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， <b>unix</b> 时间戳（ <b>UTC</b> ），精确到毫秒。
<b>maintenanceEffectiveTime</b>	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， <b>unix</b> 时间戳（ <b>UTC</b> ），精确到毫秒。
<b>maintenanceExpireTime</b>	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， <b>unix</b> 时间戳（ <b>UTC</b> ），精确到毫秒。

substatus	本次购买操作的当前状态： <ul style="list-style-type: none"> <li>● <b>NotStart</b>: 未生效。</li> <li>● <b>Effective</b>: 已生效。</li> <li>● <b>Expired</b>: 已过期。</li> </ul>
usages	软件许可证使用记录的集合，包含： maximumLocalCapacity、subscribeEffectiveTime、 subscribeExpireTime、 maintenanceEffectiveTime、 maintenanceExpireTime、substatus。 类型：数组
maximumLocalCapacity	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	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	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	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	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	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： <ul style="list-style-type: none"> <li>● <b>NotStart</b>: 未生效。</li> <li>● <b>Effective</b>: 已生效。</li> <li>● <b>Expired</b>: 已过期。</li> </ul>

- 请求示例

查询软件许可证 ID 为 ehc2b6a9-f3fb-4098-a6b3-3652a5d76269 的软件许可证信息。

```
GET /rest/v1/system/license/ehc2b6a9-f3fb-4098-a6b3-3652a5d76269 HTTP/1.1
Date: Thu, 17 Mar 2022 09:43:54 GMT
Authorization: HBlock userName:signature
Host:192.168.0.121:1443
```

- 响应示例

```
HTTP/1.1 200 OK
x-hblock-request-id: a63115d3bc704ef7a64a400c747a6b2d
Connection: keep-alive
Content-Length: 656
Date: Thu, 17 Mar 2022 09:43:54 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "licenseId": "ehc2b6a9-f3fb-4098-a6b3-3652a5d76269",
    "account": "test@ctyun.cn ",
    "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"  
        }  
    ]  
}
```

#### 4.2.4 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.	软件许可证密钥与当前 HBlock 不匹配。

		Please obtain the software 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。
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）、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,
    "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...
      ]
    }
  }
}
```

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,
    "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...
      ]
    }
  },
  "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	描述	是否必须
lunName	卷名称。 类型：字符串 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	Target 名称。 <b>说明：</b> 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。 类型：字符串 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	卷容量。 类型：整型	是

	取值：[1,1048576]，单位是 GiB。	
storageMode	<p>卷的存储类型。</p> <p>类型：枚举</p> <p>取值：</p> <ul style="list-style-type: none"> <li>● <b>Local</b>：本地模式，数据全部保留在本地。</li> <li>● <b>Cache</b>：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。</li> <li>● <b>Storage</b>：存储模式，本地保留全部数据，并异步存储到对象存储中。</li> </ul> <p>存储类型默认为 <b>Local</b>。</p>	否
config.localStorageClass	<p>卷冗余模式（仅集群版支持）。</p> <p>类型：枚举</p> <p>取值：</p> <ul style="list-style-type: none"> <li>● <b>single-copy</b>：单副本。</li> <li>● <b>2-copy</b>：两副本。</li> <li>● <b>3-copy</b>：三副本。</li> <li>● <b>EC N+M</b>：纠删码模式。其中 N、M 为正整数，<math>N &gt; M</math>，且 <math>N + M \leq 128</math>。表示将数据分割成 N 个片段，并生成 M 个校验数据。</li> </ul> <p>默认值为 EC 2+1。</p> <p><b>说明</b>（以下场景均为集群可用的前提下）：</p> <ul style="list-style-type: none"> <li>● 创建 EC N+M 的卷时：</li> </ul>	否

	<ul style="list-style-type: none"> <li>■ 服务器级别故障域：集群中可用节点台数不足 <math>N+M</math> 时，卷可以创建成功，但是卷数据将处于降级状态，允许故障的服务器台数不足 <math>M</math>，建议尽快添加服务器或修复故障服务器。 可用故障域不足 <math>N</math> 时，可以创建卷，但无法写入数据，且系统会产生告警。</li> <li>■ 数据目录级别故障域：集群中可用数据目录数量不足 <math>N+M</math> 时，卷可以创建成功，但是卷数据将处于降级状态，允许故障的数据目录个数不足 <math>M</math>，建议尽快添加数据目录或修复故障数据目录。可用故障域不足 <math>N</math> 时，可以创建卷，但无法写入数据，且系统会产生告警。</li> <li>● 创建副本模式的卷时：可用故障域数量大于等于 1，即可创建两副本、三副本的卷，卷数据将处于降级状态，建议尽快添加服务器或修复故障服务器。</li> </ul>	
<code>config.minReplica</code>	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	否

	<p>功，才视为本次写入成功。对于 EC <math>N+M</math> 模式的卷，假设该卷最小副本数设置为 <math>Y</math>（必须满足 <math>N \leq Y \leq N+M</math>），必须满足总和至少为 <math>Y</math> 的数据块和校验块写入成功，才视为本次写入成功。</p> <p>类型：整型</p> <p>取值：对于副本卷，取值范围是 <math>[1, N]</math>，<math>N</math> 为副本模式卷的副本数，默认值为 1。对于 EC 卷，取值范围是 <math>[N, N+M]</math>，默认值为 <math>N</math>。</p>	
<code>config.ECfragmentSize</code>	<p>纠删码模式分片大小（仅集群版支持）。</p> <p><code>config.localStorageClass</code> 为 EC 模式时，此设置才生效，否则忽略。</p> <p>类型：枚举</p> <p>取值：1、2、4、8、16、32、64、128、256、512、1024、2048、4096，单位是 KiB。默认值为 16。</p>	否
<code>config.redundancyOverlap</code>	<p>指定卷的折叠副本数（仅集群版支持）。在数据冗余模式下，同一份数据的不同副本/分片默认分布在不同的故障域，当故障域损坏时，允许根据卷的冗余折叠原则，将多份数据副本放在同一个故障域中，但是分布在不同的 path 上。</p> <p><b>注意：</b>如果存储池故障域级别为 path，此参数不生效。</p>	否

	<p>类型：整型</p> <p>取值：对副本模式，取值范围是[1, 副本数]，默认值为 1；对于 EC 模式，取值范围是[1, M+N]，默认值为 1。</p>	
config.sectorSize	<p>扇区大小。</p> <p>类型：枚举</p> <p>取值：512、4096，单位为字节。默认值为 4096。</p> <p><b>说明：</b>扇区大小的选取：根据自身业务场景，一般情况下，单次 I/O 操作的数据大小大于或接近 4KiB，则推荐选择 4096；单次 I/O 操作的数据大小接近 512Bytes，则推荐选择 512。如果对接 VMware 等虚拟化平台，则推荐选择 512。</p>	否
config.cachePool	<p>指定缓存存储池（仅集群版支持）。如果指定了缓存存储池，卷数据首先写入缓存存储池，然后再存入存储池。</p> <p><b>注意：</b>存储池与缓存存储池不能是同一个存储池。</p> <p>类型：字符串</p> <p>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。</p>	否

<p><code>config.pool</code></p>	<p>指定存储池（仅集群版支持），表示最终存储池，卷数据最终落在该存储池内。</p> <p>类型：字符串</p> <p>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。</p>	<p>否</p>
<p><code>config.highAvailability</code></p>	<p>是否开启高可用模式（仅集群版支持）。</p> <p>类型：枚举</p> <p>取值：</p> <ul style="list-style-type: none"> <li>● <b>ActiveStandby</b>: 启用主备，该卷关联对应 Target 下的所有 IQN。</li> <li>● <b>Disabled</b>: 不启用主备，该卷关联对应 Target 下的 1 个 IQN。</li> </ul> <p>默认值为 <b>ActiveStandby</b>。</p>	<p>否</p>
<p><code>config.writePolicy</code></p>	<p>卷的写策略。</p> <p>类型：枚举</p> <p>取值：</p> <ul style="list-style-type: none"> <li>● <b>WriteBack</b>: 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。</li> <li>● <b>WriteThrough</b>: 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</li> </ul>	<p>否</p>



	<p>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。</p> <ul style="list-style-type: none"> <li>● <b>WriteAround</b>: 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。</li> </ul> <p>默认值为 <b>WriteBack</b>。</p>	
<code>config.serverAffinity.autoFailback</code>	<p>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。</p> <p>取值：</p> <ul style="list-style-type: none"> <li>● <b>Enabled</b>: 自动进行主备切换。</li> <li>● <b>Disabled</b>: 不自动进行主备切换。</li> </ul> <p>默认值为 <b>Enabled</b>。</p>	否
<code>config.serverAffinity.priority</code>	<p>指定卷主备分布优先级的服务器 ID（仅集群版支持），根据系统指定的服务器 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。</p> <p><b>前置条件</b>: 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。</p> <p>类型：数组</p> <p>取值：服务器 ID。</p>	否

<code>config.path</code>	<p>创建卷时，指定存储卷数据的数据目录（仅单机版支持）。</p> <p>如果创建卷时不指定数据目录，使用服务器设置的默认数据目录。</p>	否
<code>cloud.provider</code>	<p>对象存储服务名称：</p> <ul style="list-style-type: none"> <li>● <b>OOS</b>：天翼云对象存储经典版 I 型。</li> <li>● <b>S3</b>：AWS 的 S3。</li> </ul> <p>默认值为 OOS。</p>	否
<code>cloud.bucketName</code>	<p>已存在的存储桶的名称。</p> <p><b>注意：</b>请勿开启 Bucket 的生命周期设定和合规保留。</p> <p>类型：字符串</p>	上云卷必填
<code>cloud.prefix</code>	<p>设置对象存储的前缀名称，设置前缀名称后，卷数据会存在存储桶以前缀命名的类文件夹中。如果未指定前缀，则直接存储在以卷名称命名的类文件夹中。</p> <p>类型：字符串</p> <p>取值：长度范围是 1~256。</p>	否
<code>cloud.accessKey</code>	<p>对象存储服务的 Access Key。</p> <p>类型：字符串</p>	上云卷必填
<code>cloud.secretKey</code>	<p>对象存储服务的 Secret Access Key。</p> <p>类型：字符串</p>	上云卷必填

cloud.endpoint	设置对象存储服务的 Endpoint。 OOS Endpoint 详见 <b>OOS Endpoint 和 Region</b> 。 类型：字符串	上云卷必填
cloud.signVersion	指定上云签名认证的类型。 类型：枚举 取值： <ul style="list-style-type: none"> <li>● v2: V2 签名认证。</li> <li>● v4: V4 签名认证。</li> </ul> 默认值为 v2。	否
cloud.region	对象存储服务的 Endpoint 资源池所在区域。 OOS region 详见 <b>OOS Endpoint 和 Region</b> 。 V4 签名时，此项必填。 类型：字符串	否
cloud.objectSize	数据存储在对象存储的大小。 类型：枚举 取值：128、256、512、1024、2048、4096、8192，单位是 KiB。默认值为 1024。	否
cloud.storageClass	设置数据在对象存储的存储类型。 类型：枚举 取值： <ul style="list-style-type: none"> <li>● STANDARD: 标准存储。</li> <li>● STANDARD_IA: 低频访问存储。</li> </ul> 默认为 STANDARD。	否

cloud.compression	是否压缩数据上传至对象存储。 类型：枚举 取值： <ul style="list-style-type: none"> <li>● <b>Enabled</b>: 压缩数据上传至对象存储。</li> <li>● <b>Disabled</b>: 不压缩数据上传至对象存储。</li> </ul> 默认值为 Enabled。	否
-------------------	--	---

● 响应结果

名称	描述
iSCSITargets	iSCSI Target 属性集合。包含：iqn、status、ips。 类型：数组
iqn	iSCSI Target IQN。
status	卷对应的 Target 的状态： <ul style="list-style-type: none"> <li>● <b>Active</b>: 主 Target。</li> <li>● <b>Standby</b>: 热备 Target。</li> <li>● <b>Unavailable</b>: 冷备 Target。</li> </ul>
ips	iSCSI Target 的 IP 属性集合，包含：ip、port。 类型：数组
ip	iSCSI Target 的 IP。
port	iSCSI Target 的端口号。
serverId	服务器 ID。
message	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."。

- 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 启用卷

此操作用来启用卷。

### ● 请求语法

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

### ● 请求参数

参数	描述	是否必须
lunName	要启用的卷名称。 类型：字符串 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

启用卷 lun02a。

```
PUT /rest/v1/block/lun/lun02a/enable HTTP/1.1
Date: Fri, 29 Jul 2022 06:51:31 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.121:1443
Content-Length: 0
```

### ● 响应示例

```
HTTP/1.1 204 No Content
x-hblock-request-id: d38ccce65daf426eb04e294c86beb5a1
Connection: keep-alive
Date: Fri, 29 Jul 2022 06:51:31 GMT
Server: HBlock
```

### 4.3.3 禁用卷

此操作用来禁用卷。

**注意：**如果禁用卷的编号是 0（卷的编号可以通过[查询指定卷](#)获取），禁用后：

- 如果客户端保持与该卷所属 iSCSI Target 的连接，则客户端可以继续正常访问该 iSCSI Target 下的其他卷。
- 如果客户端与该卷所属 iSCSI Target 断开连接后再重新连接，可能会导致无法再正常访问该 iSCSI Target 下的其他卷，如出现此情况，将该卷启用后即可重新正常访问。

#### ● 请求语法

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

#### ● 请求参数

参数	描述	是否必须
lunName	要禁用卷的卷名称。 类型：字符串 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 1

禁用卷 lun02a，卷编号为 0。

```
PUT /rest/v1/block/lun/lun02a/disable HTTP/1.1
Date: Fri, 29 Jul 2022 07:12:46 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.121:1443
```

```
Content-Length: 0
```

- 响应示例 1

```
HTTP/1.1 200 OK
x-hblock-request-id: 79c24ffcf1f84c0c8a6a2250cc8ffb96
Connection: keep-alive
Content-Type: application/json;charset=utf-8
Content-Length: 151
Date: Fri, 29 Jul 2022 07:12:46 GMT
Server: HBlock

{
  "data": {
    "message": "Disable LUN lun02a successfully. This is LUN 0, if the client
reconnects, it may not be able to access other LUNs of the target."
  }
}
```

- 请求示例 2

禁用卷 lun02b。

```
PUT /rest/v1/block/lun/lun02b/disable HTTP/1.1
Date: Fri, 29 Jul 2022 07:18:22 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.121:1443
Content-Length: 0
```

- 响应示例 2

```
HTTP/1.1 204 No Content
x-hblock-request-id: cda3a9d5ba424235803438a448294571
Connection: keep-alive
Date: Fri, 29 Jul 2022 07:18:23 GMT
Server: HBlock
```

### 4.3.4 删除卷

此操作用来删除卷。

**说明：**卷禁用后，才能被删除。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。

**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。

- 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	描述	是否必须
lunName	要删除的卷名称。 类型：字符串 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
force	是否强制删除卷。 <b>注意：</b> 强制删除卷，会产生数据残留风险，请谨慎操作。 类型：布尔	否



	取值： <ul style="list-style-type: none"> <li>● <b>true</b>: 强制删除卷。</li> <li>● <b>false</b>: 不强制删除卷。</li> </ul> 默认值为 <b>false</b> 。	
deleteCloudData	删除卷时，是否删除云上的数据。 类型：布尔 取值： <ul style="list-style-type: none"> <li>● <b>true</b>: 删除卷时，删除云上数据。</li> <li>● <b>false</b>: 删除卷时，不删除云上数据。</li> </ul> 默认值为 <b>false</b> 。 如果删除 Local 模式的卷时携带此参数，会直接忽略。	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	描述	是否必须
lunName	卷名称。 类型：字符串 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	卷要扩容后的容量。 类型：整型 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：卷处于启用或禁用状态时可以修改；卷处于“删除中”，不能修改。

- 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, SeverID...
      ]
    }
  }
}
```

- 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	描述	是否必须
lunName	卷名称。 类型：字符串 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	最小副本数（仅集群版支持）。	否

	<p>对于副本模式的卷，假设卷副本数为 <math>X</math>，最小副本数为 <math>Y</math> (<math>Y</math> 必须 <math>\leq X</math>)，该卷每次写入时，至少 <math>Y</math> 份数据写入成功，才视为本次写入成功。对于 EC <math>N+M</math> 模式的卷，假设该卷最小副本数设置为 <math>Y</math> (必须满足 <math>N \leq Y \leq N+M</math>)，必须满足总和至少为 <math>Y</math> 的数据块和校验块写入成功，才视为本次写入成功。</p> <p>类型：整型</p> <p>取值：对于副本卷，取值范围是 <math>[1, N]</math>，<math>N</math> 为副本模式卷的副本数，默认值为 1。对于 EC 卷，取值范围是 <math>[N, N+M]</math>，默认值为 <math>N</math>。</p>	
<p><code>config.redundancyOverlap</code></p>	<p>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。</p> <p><b>注意：</b>如果存储池故障域级别为 <code>path</code>，此参数不生效。</p> <p>类型：整型</p> <p>取值：对副本模式，取值范围是 <math>[1, \text{副本数}]</math>；对于 EC 模式，取值范围是 <math>[1, M+N]</math>。</p>	<p>否</p>

<p><code>config.writePolicy</code></p>	<p>卷的写策略。</p> <p>类型：枚举</p> <p>取值：</p> <ul style="list-style-type: none"> <li>● <b>WriteBack</b>: 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。</li> <li>● <b>WriteThrough</b>: 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。</li> <li>● <b>WriteAround</b>: 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。</li> </ul>	<p>否</p>
<p><code>config.serverAffinity.autoFailback</code></p>	<p>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。</p> <p>取值：</p> <ul style="list-style-type: none"> <li>● <b>Enabled</b>: 自动进行主备切换。</li> <li>● <b>Disabled</b>: 不自动进行主备切换。</li> </ul>	<p>否</p>



<code>config.serverAffinity.priority</code>	修改卷主备分布优先级的服务器 ID（仅集群版支持），根据系统指定的服务器 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。 <b>注意：</b> 如果为空数组，即[ ]，表示清空优先级设置。 类型：数组 取值：服务器 ID。	否
<code>cloud.compression</code>	是否压缩数据上传至对象存储。 类型：枚举 取值： <ul style="list-style-type: none"> <li>● <b>Enabled:</b> 压缩数据上传至对象存储。</li> <li>● <b>Disabled:</b> 不压缩数据上传至对象存储。</li> </ul>	否
<code>cloud.accessKey</code>	修改对象存储服务的 Access Key。 <b>注意：</b> 如果要修改对象存储服务的 Access Key、Secret Access Key，必须同时修改。 类型：字符串	否
<code>cloud.secretKey</code>	修改对象存储服务的 Secret Access Key。 <b>注意：</b> 如果要修改对象存储服务的 Access Key、Secret Access Key，必须同时修改。 类型：字符串	否

cloud.endpoint	设置对象存储服务的 Endpoint。 类型：字符串	否
cloud.signVersion	指定上云签名认证的类型。 类型：枚举 取值： ● v2: V2 签名认证。 ● v4: V4 签名认证。	否
cloud.region	表示 Endpoint 资源池所在区域。 V4 签名时，此项必填。 类型：字符串	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

此操作用来批量修改卷的配置。

说明：卷处于启用或禁用状态时可以修改；卷处于“删除中”，不能修改。

#### ● 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": [
    ServerID, ServerID...
  ],
  "statusDriven": statusDriven
}
```

#### ● 请求参数

参数	描述	是否必须
action	指定卷的操作内容。 类型：枚举值： <ul style="list-style-type: none"> <li>● <b>setServerAffinity</b>：对卷的主备优先级进行设置。</li> <li>● <b>switch</b>：对卷进行主备切换。</li> </ul>	是

<b>lunList</b>	需要修改的卷列表。 类型：数组 取值：卷名称。 <b>说明：</b> 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
<b>autoFailback</b>	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。 <b>注意：</b> action 取值为 setServerAffinity 时生效。 类型：字符串 取值： <ul style="list-style-type: none"> <li>● <b>Enabled:</b> 自动进行主备切换。</li> <li>● <b>Disabled:</b> 不自动进行主备切换。</li> </ul>	否
<b>priority</b>	修改卷主备分布优先级的服务器 ID（仅集群版支持），根据系统指定的服务器 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。 <b>注意：</b> 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。 类型：数组 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
<b>statusDriven</b>	是否基于卷目前的主备状态进行优先级的设置。 <b>注意：</b> action 取值为 setServerAffinity 时生效。 类型：布尔型	否

	取值： <ul style="list-style-type: none"> <li>● <b>true</b>：基于目前的主备状态进行优先级的设置。</li> <li>● <b>false</b>：不做任何变化。</li> </ul>	
--	--	--

### ● 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	描述	是否必须
lunName	卷名称。 类型：字符串 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	描述
iSCSITargets	iSCSI Target 属性集合。包含：iqn、status、ips。 类型：数组
iqn	Target IQN。
status	卷对应的 Target 的状态： <ul style="list-style-type: none"> <li>● Active: 主 Target。</li> <li>● Standby: 热备 Target。</li> <li>● Unavailable: 冷备 Target。</li> </ul>
ips	iSCSI Target 的 IP 属性集合，包含：ip、port。

	类型：数组
ip	iSCSI Target 的 IP。
port	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）

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：**

- 在执行还原操作的时候，请确保原 HBlock 和云上的连接已经断开，不再向该卷写入新的数据。
- 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。

- **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,
"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
    },
    "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	描述	是否必须
lunName	源卷的卷名称。 类型：字符串	是
uuid	源卷的唯一标识码。 类型：字符串	是
targetName	指定还原卷的 Target 名称。 <b>说明：</b> 还原卷时，如果指定的 Target 名称不存在，那么同时创建 iSCSI Target。 类型：字符串	是

	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。	
<code>storageMode</code>	<p>指定还原卷的存储类型。</p> <p>类型：枚举</p> <p>取值：</p> <ul style="list-style-type: none"> <li>● <b>Cache</b>：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。</li> <li>● <b>Storage</b>：存储模式，本地保留全部数据，并异步存储到对象存储中。</li> </ul> <p>存储类型默认为源卷的存储类型。</p>	否
<code>config.localStorageClass</code>	<p>指定还原卷冗余模式（仅集群版支持）。</p> <p>类型：枚举</p> <p>取值：</p> <ul style="list-style-type: none"> <li>● <b>single-copy</b>：单副本。</li> <li>● <b>2-copy</b>：两副本。</li> <li>● <b>3-copy</b>：三副本。</li> <li>● <b>EC N+M</b>：纠删码模式。其中 N、M 为正整数，<math>N &gt; M</math>，且 <math>N + M \leq 128</math>。表示将数据分割成 N 个片段，并生成 M 个校验数据。</li> </ul> <p>默认值为源卷的冗余模式。</p> <p><b>说明</b>（以下场景均为集群可用的前提下）：</p> <ul style="list-style-type: none"> <li>● 还原 EC <math>N+M</math> 的卷时： <ul style="list-style-type: none"> <li>■ <b>服务器级别故障域</b>：集群中可用节点台数不足 <math>N+M</math> 时，卷可以还原成功，但是卷数据将处于降级状态，允许故障的服务器台数不足 M，建议尽快添加服务</li> </ul> </li> </ul>	否



	<p>器或修复故障服务器。可用故障域不足 <math>N</math> 时，可以执行还原卷，但是不会还原成功，建议尽快添加服务器或修复故障服务器。</p> <ul style="list-style-type: none"> <li>■ 数据目录级别故障域：集群中可用数据目录数量不足 <math>N+M</math> 时，卷可以还原成功，但是卷数据将处于降级状态，允许故障的数据目录个数不足 <math>M</math>，建议尽快添加数据目录或修复故障数据目录。可用故障域不足 <math>N</math> 时，可以执行还原卷，但是不会还原成功，建议尽快添加数据目录或修复故障数据目录。</li> <li>● 还原副本模式的卷时：可用故障域数量大于等于 1，可以还原单副本、两副本、三副本的卷，但数量小于卷副本要求数量时，卷数据将处于降级状态，建议尽快添加或修复故障域。</li> </ul>	
<p><code>config.minReplica</code></p>	<p>指定还原卷的最小副本数（仅集群版支持）。</p> <p>对于副本模式的卷，假设卷副本数为 <math>X</math>，最小副本数为 <math>Y</math>（<math>Y</math> 必须 <math>\leq X</math>），该卷每次写入时，至少 <math>Y</math> 份数据写入成功，才视为本次写入成功。</p> <p>对于 EC <math>N+M</math> 模式的卷，假设该卷最小副本数设置为 <math>Y</math>（必须满足 <math>N \leq Y \leq N+M</math>），必须满足总和至少为 <math>Y</math> 的数据块和校验块写入成功，才视为本次写入成功。</p> <p><b>注意：</b>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。</p>	<p>否</p>

	<p>类型：整型</p> <p>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，默认值为源卷的最小副本数。对于 EC 卷，取值范围是[N, N+M]，默认值为源卷的最小副本数。</p>	
config.redundancyOverlap	<p>指定还原卷的折叠副本数（仅集群版支持）。在数据冗余模式下，同一份数据的不同副本/分片默认分布在不同的故障域，当故障域损坏时，允许根据卷的冗余折叠原则，将多份数据副本放在同一个故障域中，但是分布在不同的 path 上。</p> <p><b>注意：</b>如果存储池故障域级别为 path，此参数不生效。如果指定了还原卷的折叠副本数，必须指定还原卷的冗余模式。</p> <p>类型：整型</p> <p>取值：对副本模式，取值范围是[1, 副本数]；对于 EC 模式，取值范围是[1, M+N]。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。</p>	否
config.ECfragmentSize	<p>设置还原卷的纠删码模式分片大小（仅集群版支持）。</p> <p><b>config.localStorageClass</b> 为 EC 模式时，此设置才生效，否则忽略。</p> <p>类型：枚举</p> <p>取值：1、2、4、8、16、32、64、128、256、512、1024、2048、4096，单位是 KiB。默认值为源卷的纠删码模式分片大小。</p>	否

<p><code>config.cachePool</code></p>	<p>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。</p> <p><b>注意：</b>存储池与缓存存储池不能是同一个存储池。</p> <p>类型：字符串</p> <p>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。</p>	<p>否</p>
<p><code>config.pool</code></p>	<p>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。</p> <p>类型：字符串</p> <p>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。</p> <p><b>说明：</b>不指定，默认使用基础存储池。</p>	<p>否</p>
<p><code>config.highAvailability</code></p>	<p>指定还原卷的高可用类型（仅集群版支持）。</p> <p>类型：枚举</p> <p>取值：</p> <ul style="list-style-type: none"> <li>● <b>ActiveStandby:</b> 启用主备，该卷关联对应 Target 下的所有 IQN。</li> <li>● <b>Disabled:</b> 不启用主备，该卷关联对应 Target 下的 1 个 IQN。</li> </ul> <p>默认值为源卷的高可用类型。</p>	<p>否</p>
<p><code>config.writePolicy</code></p>	<p>设置还原卷的写策略。</p> <p>类型：枚举</p> <p>取值：</p>	<p>否</p>

	<ul style="list-style-type: none"> <li>● <b>WriteBack:</b> 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。</li> <li>● <b>WriteThrough:</b> 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。</li> <li>● <b>WriteAround:</b> 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。</li> </ul> <p>默认值为源卷的写策略。</p>	
<code>config.path</code>	<p>指定还原卷存储数据的数据目录（仅单机版支持）。</p> <p>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。</p> <p>类型：字符串</p>	否
<code>cloud.provider</code>	<p>指定还原卷的对象存储服务名称：</p> <ul style="list-style-type: none"> <li>● <b>OOS:</b> 天翼云对象存储经典版 I 型。</li> <li>● <b>S3:</b> AWS 的 S3。</li> </ul> <p>默认值为 OOS。</p> <p><b>说明：</b>源卷和还原卷必须使用相同的对象存储服务。</p>	否
<code>cloud.bucketName</code>	<p>源卷所在存储桶的名称。</p> <p><b>注意：</b>请勿开启 Bucket 的生命周期设定和合规保留。</p>	是

	类型：字符串	
cloud.prefix	源卷的前缀名称。如果源卷未指定前缀，则此处不设置。 类型：字符串 取值：长度范围是 1~256。	否
cloud.accessKey	源卷的 Access Key。 类型：字符串	是
cloud.secretKey	源卷的 Secret Access Key。 类型：字符串	是
cloud.endpoint	源卷的 Endpoint。 类型：字符串	是
cloud.signVersion	指定还原卷上云签名认证的类型。 类型：枚举 取值： <ul style="list-style-type: none"> <li>● v2: V2 签名认证。</li> <li>● v4: V4 签名认证。</li> </ul> 默认值为 v2。	否
cloud.region	还原卷的 region，即 Endpoint 资源池所在区域。 V4 签名时，此项必填。 类型：字符串	否
cloud.storageClass	设置还原卷上传数据至对象存储的存储类型。 类型：枚举 取值： <ul style="list-style-type: none"> <li>● STANDARD: 标准存储。</li> <li>● STANDARD_IA: 低频访问存储。</li> </ul> 默认为源卷上传数据至对象存储的存储类型。	否
cloud.compression	设置还原卷是否压缩数据上传至对象存储。	否

	类型：枚举 取值： <ul style="list-style-type: none"> <li>● <b>Enabled</b>：压缩数据上传至对象存储。</li> <li>● <b>Disabled</b>：不压缩数据上传至对象存储。</li> </ul> 默认值为源卷的设置。	
--	--	--

● 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 恢复还原中断的卷（上云卷适用）

此操作用来恢复还原中断的 HBlock 上云卷。

**前置条件：** 还原的卷因为某些原因中断，且没有对还原卷数据做任何修改。

- 请求语法

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

- 请求参数

参数	描述	是否必须
lunName	指定需要继续还原的卷的名称。 类型：字符串 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

恢复还原中断的卷 lun04a。

```
PUT /rest/v1/block/lun/lun04a/resume-recovery HTTP/1.1
Date: Tue, 17 Dec 2024 09:54:53 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Host: 192.168.0.192:1443
```

- 响应示例

```
HTTP/1.1 204 No Content
x-hblock-request-id: 6235aabf21d345a48aca2031e447b44d
Connection: keep-alive
```

Date: Tue, 17 Dec 2024 09:54:53 GMT

Server: HBlock

### 4.3.11 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/v1/block/lun?filter=filter&range=i-j HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

#### ● 请求参数

参数	描述	是否必须
filter	查询的过滤条件。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： <ul style="list-style-type: none"> <li>● lunName: 卷名称。</li> <li>● status: 卷的状态。</li> <li>● capacity: 卷容量。</li> <li>● iqn: Target 的识别。</li> </ul>	否
range	查询卷的范围，格式为 <code>i-j</code> ， <code>i</code> 和 <code>j</code> 为正整数， $0 < j-i < 1000$ 。按照卷名称进行排序，返回第 <code>i</code> 到第 <code>j</code> 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 <code>range</code> ，则返回所有符合条件的卷。	否

#### ● 响应结果

名称	描述
----	----

lunName	卷名称。
status	卷的状态： <ul style="list-style-type: none"> <li>● <b>Enabled</b>: 卷处于启用状态。</li> <li>● <b>Disabled</b>: 卷处于禁用状态。</li> <li>● <b>Deleting</b>: 卷正在删除中。</li> <li>● <b>DeleteFailed</b>: 卷删除失败。</li> <li>● <b>Recovering</b>: 卷正在还原中。</li> <li>● <b>RecoverFailed</b>: 卷还原失败。</li> </ul>
recoverFailedReason	卷还原失败的原因。 仅卷状态为 <b>RecoverFailed</b> 时显示。
deleteFailedReason	卷云上数据删除失败的原因。 仅卷状态为 <b>DeleteFailed</b> 时显示。
storageMode	卷的存储类型： <ul style="list-style-type: none"> <li>● <b>Local</b>: 本地模式，数据全部保留在本地。</li> <li>● <b>Cache</b>: 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。</li> <li>● <b>Storage</b>: 存储模式，本地保留全部数据，并异步存储到对象存储中。</li> </ul>
capacity	卷容量，单位是 GiB。
createTime	卷创建时间。unix 时间戳（UTC），精确到毫秒。
iSCSITargets	iSCSI Target 属性集合。包含： <b>iqn</b> 、 <b>status</b> 、 <b>ips</b> 、 <b>serverId</b> 。 类型：数组
iqn	iSCSI Target IQN。
iSCSITargets.status	卷对应的 Target 的状态：

	<ul style="list-style-type: none"> <li>● <b>Active:</b> 主 Target。</li> <li>● <b>Standby:</b> 热备 Target。</li> <li>● <b>Offline:</b> 离线 Target。</li> <li>● <b>Unavailable:</b> 冷备 Target。</li> </ul>
<code>ips</code>	iSCSI Target 的 IP 属性集合，包含： <code>ip</code> 、 <code>port</code> 。 类型：数组
<code>ip</code>	iSCSI Target 的 IP。
<code>port</code>	iSCSI Target 的端口号。
<code>serverId</code>	IQN 所在的服务器 ID。
<code>cloud.bucketName</code>	存储桶的名称（仅上云卷支持）。
<code>cloud.endpoint</code>	对象存储的 Endpoint（仅上云卷支持）。
<code>cloud.signVersion</code>	上云签名认证的类型（仅上云卷支持）： <ul style="list-style-type: none"> <li>● <b>v2:</b> V2 签名认证。</li> <li>● <b>v4:</b> V4 签名认证。</li> </ul>
<code>cloud.region</code>	对象存储的 Endpoint 资源池所在区域（仅上云卷支持）。
<code>cloud.provider</code>	对象存储服务名称： <ul style="list-style-type: none"> <li>● <b>OOS:</b> 天翼云对象存储经典版 I 型。</li> <li>● <b>S3:</b> AWS 的 S3。</li> </ul>
<code>config.localStorageClass</code>	卷冗余模式（仅集群版支持）： <ul style="list-style-type: none"> <li>● <b>single-copy:</b> 单副本。</li> <li>● <b>2-copy:</b> 两副本。</li> <li>● <b>3-copy:</b> 三副本。</li> <li>● <b>EC N+M.</b></li> </ul>
<code>config.minReplica</code>	卷的最小副本数（仅集群版支持）。

<code>config.ECfragmentSize</code>	分片大小。如果冗余模式为 EC $N+M$ ，则会返回此项。
<code>config.sectorSize</code>	扇区大小： <ul style="list-style-type: none"> <li>● 512。</li> <li>● 4096。</li> </ul>
<code>config.highAvailability</code>	高可用模式类型（仅集群版支持）： <ul style="list-style-type: none"> <li>● <b>ActiveStandby</b>: 主备，该卷关联对应 Target 下的所有 IQN。</li> <li>● <b>Disabled</b>: 禁用，该卷关联对应 Target 下的 1 个 IQN。</li> </ul>
<code>config.writePolicy</code>	卷的写策略： <ul style="list-style-type: none"> <li>● <b>WriteBack</b>: 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。</li> <li>● <b>WriteThrough</b>: 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。</li> <li>● <b>WriteAround</b>: 绕写，指数据直接写到磁盘，不写入内存。</li> </ul>
<code>config.cachePool</code>	缓存存储池（仅集群版支持）。
<code>config.pool</code>	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>	卷的折叠副本数（仅集群版支持）。
<code>config.serverAffinity.autoFailback</code>	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）： <ul style="list-style-type: none"> <li>● <b>Enabled</b>: 自动进行主备切换。</li> </ul>

	<ul style="list-style-type: none"> <li>● <b>Disabled:</b> 不自动进行主备切换。</li> </ul>
config.serverAffinity.priority	卷主备分布优先级的服务器 ID（仅集群版支持），根据系统指定的服务器 ID 顺序来选择卷的主备 IQN。
config.path	数据目录（仅单机版支持）。
number	卷编号。
WWID	卷的唯一标识符。

### ● 请求示例 1

单机版：查询卷名称包含 lunb 或 lunc 的卷，且状态为 Enabled 的卷。

```
GET /rest/v1/block/lun?filter=lunName:lunb or lunName:lunc and status:Enabled HTTP/1.1
Date: Fri, 03 Jan 2025 03:22:03 GMT
Authorization: userName:signature
Host: 192.168.0.32:1443
```

### ● 响应示例 1

```
HTTP/1.1 200 OK
x-hblock-request-id: 04521019519844689f4051424b170aaf
Connection: keep-alive
Content-Length: 1593
Date: Fri, 03 Jan 2025 03:22:03 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "luns": [
      {
        "lunName": "lunb1",
        "status": "Enabled",
        "capacity": 230,
```

```
"storageMode": "Cache",
"createTime": 1734407368558,
"iSCSITargets": [
  {
    "iqn": "iqn.2012-08.cn.ctyunapi.oos:targetb.2",
    "status": "Active",
    "ips": [
      {
        "ip": "192.168.0.32",
        "port": 3260
      }
    ],
    "serverId": "hblock_1"
  }
],
"cloud": {
  "bucketName": "hblocktest3",
  "endpoint": "https://oos-cn.ctyunapi.cn",
  "signVersion": "v4",
  "region": "cn",
  "provider": "OOS"
},
"config": {
  "sectorSize": 4096,
  "writePolicy": "WriteBack",
  "path": "/mnt/stor"
},
"number": 0,
"WWID": "330000002f3e892d"
},
{
  "lunName": "lunb3",
  "status": "Enabled",
  "capacity": 299,
  "storageMode": "Local",
```



```
"createTime": 1735176509888,
  "iSCSITargets": [
    {
      "iqn": "iqn.2012-08.cn.ctyunapi.oos:targetb.2",
      "status": "Active",
      "ips": [
        {
          "ip": "192.168.0.32",
          "port": 3260
        }
      ],
      "serverId": "hblock_1"
    }
  ],
  "config": {
    "sectorSize": 4096,
    "writePolicy": "WriteBack",
    "path": "/mnt/stor"
  },
  "number": 2,
  "WWID": "3300000006b0fed64"
},
{
  "lunName": "lunb4",
  "status": "Enabled",
  "capacity": 333,
  "storageMode": "Local",
  "createTime": 1735176585689,
  "iSCSITargets": [
    {
      "iqn": "iqn.2012-08.cn.ctyunapi.oos:targetb.2",
      "status": "Active",
      "ips": [
        {
          "ip": "192.168.0.32",
```

```
        "port": 3260
      }
    ],
    "serverId": "hblock_1"
  }
],
"config": {
  "sectorSize": 4096,
  "writePolicy": "WriteBack",
  "path": "/mnt/stor"
},
"number": 3,
"WWID": "3300000006346a3f7"
},
{
  "lunName": "lunc2",
  "status": "Enabled",
  "capacity": 30,
  "storageMode": "Local",
  "createTime": 1735176602182,
  "iSCSITargets": [
    {
      "iqn": "iqn.2012-08.cn.ctyunapi.oos:targetc.3",
      "status": "Active",
      "ips": [
        {
          "ip": "192.168.0.32",
          "port": 3260
        }
      ],
      "serverId": "hblock_1"
    }
  ],
  "config": {
    "sectorSize": 4096,
```

```

        "writePolicy": "WriteBack",
        "path": "/mnt/stor"
    },
    "number": 1,
    "WWID": "33ffffffff1b6f1c9"
}
]
}
}

```

### ● 请求示例 2

集群版：查询卷名称包含 lun 的卷，并且只列出第 1 个到第 3 个查询结果。

```

GET /rest/v1/block/lun?filter=lunName:lun&range=1-3 HTTP/1.1
Date: Fri, 03 Jan 2025 06:24:53 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.192:1443

```

### ● 响应示例 2

```

HTTP/1.1 200 OK
x-hblock-request-id: ff61c214073f47e6a1dde46272677fea
Connection: keep-alive
Content-Length: 2268
Date: Fri, 03 Jan 2025 06:24:53 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "luns": [
      {
        "lunName": "lun01a",
        "status": "Enabled",
        "capacity": 100,
        "storageMode": "Local",

```

```
"createTime": 1735875169673,
"iSCSITargets": [
  {
    "iqn": "iqn.2012-08.cn.ctyunapi.oos:target01.2",
    "status": "Active",
    "ips": [
      {
        "ip": "192.168.0.202",
        "port": 3260
      }
    ],
    "serverId": "hblock_2"
  },
  {
    "iqn": "iqn.2012-08.cn.ctyunapi.oos:target01.1",
    "status": "Standby",
    "ips": [
      {
        "ip": "192.168.0.102",
        "port": 3260
      }
    ],
    "serverId": "hblock_3"
  }
],
"config": {
  "localStorageClass": "EC 2+1",
  "minReplica": 2,
  "sectorSize": 4096,
  "highAvailability": "ActiveStandby",
  "writePolicy": "WriteBack",
  "pool": "default",
  "redundancyOverlap": 1,
  "serverAffinity": {
    "autoFailback": "Enabled",
```

```
        "priority": [
            "hblock_2"
        ]
    },
    "ECfragmentSize": 16
},
"number": 0,
"WWID": "3300000000aa369a3"
},
{
    "lunName": "lun02a",
    "status": "Enabled",
    "capacity": 200,
    "storageMode": "Cache",
    "createTime": 1735875228560,
    "iSCSITargets": [
        {
            "iqn": "iqn.2012-08.cn.ctyunapi.oos:target02.3",
            "status": "Active",
            "ips": [
                {
                    "ip": "192.168.0.192",
                    "port": 3260
                }
            ],
            "serverId": "hblock_1"
        },
        {
            "iqn": "iqn.2012-08.cn.ctyunapi.oos:target02.4",
            "status": "Standby",
            "ips": [
                {
                    "ip": "192.168.0.102",
                    "port": 3260
                }
            ]
        }
    ]
}
```

```
    ],
    "serverId": "hblock_3"
  }
],
"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,
  "serverAffinity": {
    "autoFailback": "Enabled"
  },
  "ECfragmentSize": 16
},
"number": 0,
"WWID": "33ffffffff9000b000"
},
{
  "lunName": "lun03a",
  "status": "Enabled",
  "capacity": 300,
  "storageMode": "Storage",
  "createTime": 1735875276924,
  "iSCSITargets": [
    {
      "iqn": "iqn.2012-08.cn.ctyunapi.oos:target03.5",
```

```
        "status": "Active",
        "ips": [
            {
                "ip": "192.168.0.202",
                "port": 3260
            }
        ],
        "serverId": "hblock_2"
    },
    {
        "iqn": "iqn.2012-08.cn.ctyunapi.oos:target03.6",
        "status": "Standby",
        "ips": [
            {
                "ip": "192.168.0.192",
                "port": 3260
            }
        ],
        "serverId": "hblock_1"
    }
],
"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,  
        "serverAffinity": {  
            "autoFailback": "Enabled"  
        },  
        "ECfragmentSize": 16  
    },  
    "number": 0,  
    "WWID": "3300000002d28e800"  
}  
]  
}  
}
```



### 4.3.12 查询指定卷

此操作用来查询指定卷。

#### ● 请求语法

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

#### ● 请求参数

参数	描述	是否必须
lunName	卷名称。 <b>说明：</b> 如果不填写此项，会返回所有卷信息。 类型：字符串 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	描述
lunName	卷名称。
targetName	Target 名称。
status	卷的状态： <ul style="list-style-type: none"> <li>● <b>Enabled:</b> 卷处于启用状态。</li> <li>● <b>Disabled:</b> 卷处于禁用状态。</li> <li>● <b>Deleting:</b> 卷正在删除中。</li> <li>● <b>DeleteFailed:</b> 卷删除失败。</li> <li>● <b>Recovering:</b> 卷正在还原中。</li> </ul>

	<ul style="list-style-type: none"> <li>● <b>RecoverFailed</b>: 卷还原失败。</li> </ul>
recoverFailedReason	卷还原失败的原因。 仅卷状态为 <b>RecoverFailed</b> 时显示。
deleteFailedReason	卷云上数据删除失败的原因。 仅卷状态为 <b>DeleteFailed</b> 时显示。
capacity	卷容量，单位是 GiB。
storageMode	卷的存储类型： <ul style="list-style-type: none"> <li>● <b>Local</b>: 本地模式，数据全部保留在本地。</li> <li>● <b>Cache</b>: 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。</li> <li>● <b>Storage</b>: 存储模式，本地保留全部数据，并异步存储到对象存储中。</li> </ul>
createTime	卷创建时间。unix 时间戳（UTC），精确到毫秒。
iSCSITargets	iSCSI Target 属性集合。包含：iqn、status、ips、targetPortalIP、serverId。 类型：数组
iqn	Target IQN。
iSCSITargets.status	卷对应的 Target 的状态： <ul style="list-style-type: none"> <li>● <b>Active</b>: 主 Target。</li> <li>● <b>Standby</b>: 热备 Target。</li> <li>● <b>Offline</b>: 离线 Target。</li> <li>● <b>Unavailable</b>: 冷备 Target。</li> </ul>
ips	iSCSI Target 的 IP 属性集合，包含：ip、port。 类型：数组
ip	iSCSI Target 的 IP。

port	iSCSI Target 的端口号。
targetPortalIP.ips	iSCSI 目标门户的属性集合，包括： <b>ip</b> 、 <b>port</b> 。 类型：数组
ip	iSCSI 目标门户的 IP 地址。
port	iSCSI 目标门户的端口号。
targetPortalIP.status	iSCSI 目标门户的状态： <ul style="list-style-type: none"> <li>● <b>Enabled</b>: 启用状态。</li> <li>● <b>Disabled</b>: 禁用状态。</li> </ul>
serverId	IQN 所在的服务器 ID。
cloud.bucketName	存储桶的名称（仅上云卷支持）。
cloud.prefix	对象存储中的前缀名称（仅上云卷支持）。 如果未设置，则不显示此段。
cloud.accessKey	对象存储服务的 Access Key（仅上云卷支持）。
cloud.endpoint	对象存储 Endpoint（仅上云卷支持）。
cloud.objectSize	数据存储在对对象存储的大小（仅上云卷支持）。
cloud.storageClass	数据在对对象存储的存储类型（仅上云卷支持）： <ul style="list-style-type: none"> <li>● <b>STANDARD</b>: 标准存储。</li> <li>● <b>STANDARD_IA</b>: 低频访问存储。</li> </ul>
cloud.compression	是否压缩数据上传至对象存储（仅上云卷支持）： <ul style="list-style-type: none"> <li>● <b>Enabled</b>: 压缩数据上传至对象存储。</li> <li>● <b>Disabled</b>: 不压缩数据上传至对象存储。</li> </ul>
cloud.signVersion	上云签名认证的类型（仅上云卷支持）：

	<ul style="list-style-type: none"> <li>● v2: V2 签名认证。</li> <li>● v4: V4 签名认证。</li> </ul>
cloud.region	Endpoint 资源池所在区域（仅上云卷支持）。
cloud.provider	对象存储服务名称： <ul style="list-style-type: none"> <li>● OOS: 天翼云对象存储经典版 I 型。</li> <li>● S3: AWS 的 S3。</li> </ul>
config.localStorageClass	卷冗余模式（仅集群版支持）： <ul style="list-style-type: none"> <li>● single-copy: 单副本。</li> <li>● 2-copy: 两副本。</li> <li>● 3-copy: 三副本。</li> <li>● EC <math>N+M</math>。</li> </ul>
config.minReplica	卷的最小副本数（仅集群版支持）。
config.ECfragmentSize	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	扇区大小，单位为字节。
config.highAvailability	高可用模式类型（仅集群版支持）： <ul style="list-style-type: none"> <li>● ActiveStandby: 主备，该卷关联对应 Target 下的所有 IQN。</li> <li>● Disabled: 禁用，该卷关联对应 Target 下的 1 个 IQN。</li> </ul>
config.writePolicy	卷的写策略： <ul style="list-style-type: none"> <li>● 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。</li> <li>● 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。</li> </ul>

	<ul style="list-style-type: none"> <li>● <b>WriteAround:</b> 绕写，指数据直接写到磁盘，不写入内存。</li> </ul>
<code>config.cachePool</code>	缓存存储池（仅集群版支持）。
<code>config.pool</code>	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>	卷的折叠副本数（仅集群版支持）。
<code>config.serverAffinity.autoFailback</code>	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）： <ul style="list-style-type: none"> <li>● <b>Enabled:</b> 自动进行主备切换。</li> <li>● <b>Disabled:</b> 不自动进行主备切换。</li> </ul>
<code>config.serverAffinity.priority</code>	卷主备分布优先级的服务器 ID（仅集群版支持），根据系统指定的服务器 ID 顺序来选择卷的主备 IQN。
<code>config.path</code>	数据目录（仅单机版支持）。
<code>number</code>	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。
<code>uuid</code>	卷的唯一识别码。
<code>WWID</code>	卷的唯一标识符。

- 请求示例 1

单机版本地卷：查询卷 lun02。

<pre>GET /rest/v1/block/lun/lun02 HTTP/1.1 Date: Tue, 17 Dec 2024 03:54:16 GMT Authorization: HBlock <i>userName:signature</i> Host: 192.168.0.32:1443</pre>
--

## ● 响应示例 1

```
HTTP/1.1 200 OK
x-hblock-request-id: 94da6070904c49d486ec5b6c1244543c
Connection: keep-alive
Content-Length: 449
Date: Tue, 17 Dec 2024 03:54:16 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "lunName": "lun02",
    "targetName": "target02",
    "status": "Enabled",
    "capacity": 20,
    "storageMode": "Local",
    "createTime": 1734407307599,
    "iSCSITargets": [
      {
        "iqn": "iqn.2012-08.cn.ctyunapi.oos:target02.1",
        "status": "Active",
        "ips": [
          {
            "ip": "192.168.0.32",
            "port": 3260
          }
        ],
        "serverId": "hblock_1"
      }
    ],
    "config": {
      "sectorSize": 4096,
      "writePolicy": "WriteBack",
      "path": "/mnt/stor"
    }
  }
}
```

```
    },  
    "number": 0,  
    "uuid": "lun-uuid-c7ea21b5-81bf-494c-b093-2a26076ee404",  
    "WWID": "3300000002b6bda37"  
  }  
}
```

### ● 请求示例 2

单机版上云卷：查询卷 lunb1。

```
GET /rest/v1/block/lun/lunb1 HTTP/1.1  
Date: Fri, 03 Jan 2025 06:31:20 GMT  
Authorization: HBlock userName:signature  
Host: 192.168.0.32:1443
```

### ● 响应示例 2

```
HTTP/1.1 200 OK  
x-hblock-request-id: 9097f8ff916740b3893bbb45d063e01b  
Connection: keep-alive  
Content-Length: 694  
Date: Fri, 03 Jan 2025 06:31:20 GMT  
Content-Type: application/json;charset=utf-8  
Server: HBlock  
  
{  
  "data": {  
    "lunName": "lunb1",  
    "targetName": "targetb",  
    "status": "Enabled",  
    "capacity": 230,  
    "storageMode": "Cache",  
    "createTime": 1734407368558,  
    "iSCSITargets": [  
      {  
        "iqn": "iqn.2012-08.cn.ctyunapi.oos:targetb.2",
```

```
        "status": "Active",
        "ips": [
            {
                "ip": "192.168.0.32",
                "port": 3260
            }
        ],
        "serverId": "hblock_1"
    }
],
"cloud": {
    "bucketName": "hblocktest3",
    "prefix": "lunb",
    "accessKey": "8f129a5529f202811fd0",
    "endpoint": "https://oos-cn.ctyunapi.cn",
    "objectSize": 1024,
    "storageClass": "STANDARD",
    "compression": "Enabled",
    "signVersion": "v4",
    "region": "cn",
    "provider": "OOS"
},
"config": {
    "sectorSize": 4096,
    "writePolicy": "WriteBack",
    "path": "/mnt/stor"
},
"number": 0,
"uuid": "lun-uuid-27f8fa0e-86b4-4c3d-b8d9-8f3dabeadb85",
"WWID": "3300000002f3e892d"
}
}
```

- 请求示例 3

集群版本地卷：查询卷 lun01a。



```
GET /rest/v1/block/lun/lun01a HTTP/1.1
Date: Fri, 3 Jan 2025 16:32:55 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.192:1443
```

### ● 响应示例 3

```
HTTP/1.1 200 OK
x-hblock-request-id: 94bfd0aed30d41209bebee5281fbafb8
Connection: keep-alive
Content-Length: 772
Date: Fri, 3 Jan 2025 16:32:55 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "lunName": "lun01a",
    "targetName": "target01",
    "status": "Enabled",
    "capacity": 100,
    "storageMode": "Local",
    "createTime": 1735875169673,
    "iSCSITargets": [
      {
        "iqn": "iqn.2012-08.cn.ctyunapi.oos:target01.2",
        "status": "Active",
        "ips": [
          {
            "ip": "192.168.0.202",
            "port": 3260
          }
        ],
        "serverId": "hblock_2"
      }
    ],
  },
}
```

```

        "iqn": "iqn.2012-08.cn.ctyunapi.oos:target01.1",
        "status": "Standby",
        "ips": [
            {
                "ip": "192.168.0.102",
                "port": 3260
            }
        ],
        "serverId": "hblock_3"
    }
],
"config": {
    "localStorageClass": "EC 2+1",
    "minReplica": 2,
    "sectorSize": 4096,
    "highAvailability": "ActiveStandby",
    "writePolicy": "WriteBack",
    "pool": "default",
    "redundancyOverlap": 1,
    "serverAffinity": {
        "autoFailback": "Enabled",
        "priority": [
            "hblock_2"
        ]
    },
    "ECfragmentSize": 16
},
"number": 0,
"uuid": "lun-uuid-5816ffd2-c232-4e5a-8b53-2834bc9303b9",
"WWID": "3300000000aa369a3"
}
}

```

- 请求示例 4

集群版上云卷：查询卷 lun02a。

```
GET /rest/v1/block/lun/lun02a HTTP/1.1
Date: Fri, 3 Jan 2025 16:36:23 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.192:1443
```

#### ● 响应示例 4

```
HTTP/1.1 200 OK
x-hblock-request-id: b850de90c4c945f4a639a39f6a8efd5a
Connection: keep-alive
Content-Length: 982
Date: Fri, 3 Jan 2025 16:36:23 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "lunName": "lun02a",
    "targetName": "target02",
    "status": "Enabled",
    "capacity": 200,
    "storageMode": "Cache",
    "createTime": 1735875228560,
    "iSCSITargets": [
      {
        "iqn": "iqn.2012-08.cn.ctyunapi.oos:target02.3",
        "status": "Active",
        "ips": [
          {
            "ip": "192.168.0.192",
            "port": 3260
          }
        ]
      }
    ],
    "serverId": "hblock_1"
  },
  {
```

```
        "iqn": "iqn.2012-08.cn.ctyunapi.oos:target02.4",
        "status": "Standby",
        "ips": [
            {
                "ip": "192.168.0.102",
                "port": 3260
            }
        ],
        "serverId": "hblock_3"
    }
],
"cloud": {
    "bucketName": "hblocktest3",
    "prefix": "lun02a",
    "accessKey": "8f129a5529f202811fd0",
    "endpoint": "https://oos-cn.ctyunapi.cn",
    "objectSize": 1024,
    "storageClass": "STANDARD",
    "compression": "Enabled",
    "signVersion": "v2",
    "provider": "OOS"
},
"config": {
    "localStorageClass": "EC 2+1",
    "minReplica": 2,
    "sectorSize": 4096,
    "highAvailability": "ActiveStandby",
    "writePolicy": "WriteBack",
    "pool": "default",
    "redundancyOverlap": 1,
    "serverAffinity": {
        "autoFailback": "Enabled"
    },
    "ECfragmentSize": 16
},
```

```
"number": 0,  
"uuid": "lun-uuid-4b44d601-0f88-43e8-868e-6a7cb0fe55ee",  
"WWID": "33ffffffff9000b000"  
}  
}
```

### 4.3.13 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 life cycle 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 recoverd.	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 <i>lunname</i> .	无法修改卷的 Endpoint。
400	DisableDeletingLUN	Failed to disable LUN because the LUN <i>lunName</i> is deleting.	禁用卷失败，因为该卷正在删除。
400	EnableDeletingLUN	Failed to enable LUN because the LUN <i>lunName</i> is deleting.	启用卷失败，因为该卷正在删除。
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 value.	超出了系统允许的最大个数，无法执行该操作。
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	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 'filter' 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 'high availability' 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 'lun name' 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 'min replica' failed to satisfy constraint: argument range is [1, copy	最小副本数取值范围为：副本卷为[1, 副

		number] for replica mode and [N, N+M] for EC N+M mode.	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 ' <i>range</i> ' failed to satisfy constraint: Argument must satisfy pattern ' <i>i-j</i> ', where <i>i</i> , <i>j</i> are positive integers, <i>i</i> is less than <i>j</i> , 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 ' <i>sector size</i> ' 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 ' <i>local storage class</i> ' failed to satisfy constraint: Argument must satisfy enum value set: [single-copy, 2-copy, 3-copy, EC $N+M$ ], <i>N</i> and <i>M</i> are positive integers, $N > M$ 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 enum value set: [WriteBack, WriteThrough, WriteAround].	写策略非法。
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	'operation' 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 poolName.	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 when a specific server list is designated as the affinity priority.	仅支持针对单个 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	'operation' 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 OOS 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 resource: <i>bucketName</i> .	无权限获得存储桶的生命周期配置，请更新 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	NoSuchBackupConfiguration	Backup is not configured.	没有配置备份到 OOS。
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	DeleteLUNEnabled	Cannot delete it because the iSCSI LUN is enabled, please disable first.	不能删除正在启用的卷。
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 for the LUN <i>lunname,lunname,....</i>	操作失败，Target IQNs 状态异常，请检查后重试。
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。
500	CloudInternalError	Cloud encountered an internal error. Please try again.	云内部错误，请重试。

## 4.4 iSCSI Target 管理

### 4.4.1 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
  }
}
```

#### 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",...]
}
    
```

● 请求参数

参数	描述	是否必须
targetName	iSCSI Target 名称。 类型：字符串 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	iSCSI Target 下每个 IQN 允许建立的最大会话数。 类型：整型 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。 <b>注意：</b> 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	指定 iSCSI Target 的回收策略。 类型：枚举 取值：	否

	<ul style="list-style-type: none"> <li>● <b>Delete:</b> 当 iSCSI Target 关联的卷全部删除后, iSCSI Target 自动删除。</li> <li>● <b>Retain:</b> 当 iSCSI Target 关联的卷全部删除后, iSCSI Target 仍然保留。</li> </ul> <p>默认值为 <b>Retain</b>。</p> <p><b>说明:</b> 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。</p>	
chap.name	<p>CHAP 名称。</p> <p>类型: 字符串</p> <p>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。</p>	否
chap.password	<p>CHAP 密码。</p> <p>类型: 字符串</p> <p>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。</p>	否
chap.status	<p>是否启用 CHAP 认证。</p> <p>类型: 枚举</p> <p>取值:</p> <ul style="list-style-type: none"> <li>● <b>Enabled:</b> 启用 CHAP 认证。</li> <li>● <b>Disabled:</b> 禁用 CHAP 认证。</li> </ul> <p><b>注意:</b> 如果启用 CHAP 认证, name 和 password 必须设置。</p>	是
serverNumbers	<p>Target 所在的服务器数量 (仅集群版支持)。</p> <p>类型: 整型</p> <p>取值: [2, n], n 为集群内服务器的数量。默认值是 2。</p>	否

server	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
--------	--	---

### ● 响应结果

名称	描述
iSCSITargets	iSCSI Target 属性集合。包含：iqn、ips、 类型：数组
iqn	iSCSI Target IQN。
ips	iSCSI Target 的 IP 属性集合，包含：ip、port。 类型：数组
ip	iSCSI Target 的 IP。
port	iSCSI Target 的端口号。

### ● 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

集群版：创建一个名称为 **target06** 的 Target。

```
POST /rest/v1/block/target HTTP/1.1
Date: Thu, 28 Sep 2023 07:39:52 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 252
```

```
Host: 192.168.0.192:1443

{
  "targetName": "target06",
  "maxSessions": 6,
  "chap": {
    "name": "test",
    "password": "12345678901w",
    "status": "Enabled"
  },
  "serverNumbers": 3,
  "server": ["hblock_1", "hblock_2"]
}
```

## ● 响应示例 2

```
HTTP/1.1 200 OK
x-hblock-request-id: e1e127f0345c4b9ebc36d168998901a9
Connection: keep-alive
Content-Length: 303
Date: Thu, 28 Sep 2023 07:39:52 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "iSCSITargets": [
      {
        "iqn": "iqn.2012-08.cn.ctyunapi.oos:target06.6",
        "ips": [
          {
            "ip": "192.168.0.110",
            "port": 3260
          }
        ]
      }
    ]
  },
}
```

```
{
  "iqn": "iqn.2012-08.cn.ctyunapi.oos:target06.7",
  "ips": [
    {
      "ip": "192.168.0.117",
      "port": 3260
    }
  ]
},
{
  "iqn": "iqn.2012-08.cn.ctyunapi.oos:target06.8",
  "ips": [
    {
      "ip": "192.168.0.102",
      "port": 3260
    }
  ]
}
]
```

## 4.4.2 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	描述	是否必须
targetName	iSCSI Target 名称。 类型：字符 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	CHAP 名称。 类型：字符串 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	CHAP 密码。 类型：字符串 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	是否启用 CHAP 认证。 类型：枚举 取值： <ul style="list-style-type: none"> <li>● Enabled: 启用 CHAP 认证。</li> <li>● Disabled: 禁用 CHAP 认证。</li> </ul> 注意：如果是首次启用 CHAP 认证，name 和 password 必须设置。	是

● 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
```

### 4.4.3 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。

#### ● 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	描述	是否必须
targetName	iSCSI Target 名称。 类型：字符串 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	是否强制迁移 iSCSI Target。 <b>注意：</b> 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。 类型：布尔 取值： <b>true</b> ：强制迁移 iSCSI Target。	是
source	iSCSI Target 的源服务器 ID。 类型：字符串	是
dest	iSCSI Target 的目标服务器 ID。 类型：字符串	是

### ● 响应结果

名称	描述
luns	卷的属性集合，包含：lunName、iSCSITargets、 类型：数组
lunName	关联卷的名称。
iSCSITargets	iSCSI Target 属性集合。包含：iqn、status、ips、targetPortalIP。 类型：数组
iqn	iSCSI Target IQN。
status	卷对应的 iSCSI Target 的状态： <ul style="list-style-type: none"> <li>● <b>Active</b>：主 Target。</li> <li>● <b>Standby</b>：热备 Target。</li> </ul>



	<ul style="list-style-type: none"> <li>● <b>Offline</b>: 离线 Target。</li> <li>● <b>Unavailable</b>: 冷备 Target。</li> </ul>
ips	iSCSI Target 的 IP 属性集合, 包含: ip、port。 类型: 数组
ip	iSCSI Target 的 IP。
port	iSCSI Target 的端口号。
targetPortalIP.ips	iSCSI 目标门户的 IP 属性集合, 包含: ip、port。 类型: 数组
targetPortalIP.ips.ip	iSCSI 目标门户的 IP 地址。 说明: 如果未配置, 不返回该项。
targetPortalIP.ips.port	iSCSI 目标门户的端口号。 说明: 如果未配置, 不返回该项。
targetPortalIP.status	iSCSI 目标门户的状态: <ul style="list-style-type: none"> <li>● <b>Enabled</b>: 启用状态。</li> <li>● <b>Disabled</b>: 禁用状态。</li> </ul>

### ● 请求示例

迁移 target01。

```

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 userName:signature
Content-Length: 95
Host: 192.168.0.192:1443

{
  "server": [
    {
      "source": "hblock_1",
      "dest": "hblock_2"
    }
  ]
}
    
```

```
}  
]  
}
```

### ● 响应示例

```
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  
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。

#### 4.4.4 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	描述	是否必须
targetName	iSCSI Target 名称。 类型：字符 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	iSCSI Target 下每个 IQN 允许建立的最大会话数。 类型：整型 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。 <b>说明：</b> 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。	否

	<b>注意：</b> 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。	
--	--	--

- 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
```

## 4.4.5 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	描述	是否必须
targetName	iSCSI Target 名称。 类型：字符 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	指定 iSCSI Target 的回收策略。 类型：枚举 取值： <ul style="list-style-type: none"> <li>● Delete: 当 iSCSI Target 关联的卷全部删除后，iSCSI Target 自动删除。</li> <li>● Retain: 当 iSCSI Target 关联的卷全部删除后，iSCSI Target 仍然保留。</li> </ul>	否

- 请求示例

将 target01 的回收策略修改为 Delete。

```
PUT / rest/v1/block/target/target01HTTP/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
```

## 4.4.6 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	描述	是否必须
targetName	iSCSI Target 名称。 类型：字符串 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

## 4.4.7 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	描述	是否必须
targetName	iSCSI Target 名称。 类型：字符串 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
```

## 4.4.8 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	描述	是否必须
filter	查询的过滤条件。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： <ul style="list-style-type: none"> <li>● lunName: iSCSI Target 关联的卷名称。</li> <li>● targetName: iSCSI Target 名称。</li> <li>● iqn: iSCSI Target IQN。</li> </ul>	否
range	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	描述
targetName	iSCSI Target 名称。
maxSessions	iSCSI Target 下每个 IQN 允许建立的最大会话数。

iSCSITargets	iSCSI Target 属性集合。包含：iqn、ips、targetPortalIP、serverId。 类型：数组
iqn	iSCSI Target IQN。
ips	iSCSI Target 的 IP 属性集合，包含：ip、port。 类型：数组
ip	iSCSI Target 的 IP。
port	iSCSI Target 的端口号。
targetPortalIP.ips	iSCSI 目标门户 IP 属性集合，包含：ip、port。 类型：数组
ip	iSCSI 目标门户 IP。 说明：如果未配置，不返回该项。
port	iSCSI 目标门户端口号。 说明：如果未配置，不返回该项。
targetPortalIP.status	iSCSI 目标门户 IP 状态： <ul style="list-style-type: none"> <li>● Enabled: 启用状态。</li> <li>● Disabled: 停用状态。</li> </ul> 说明：如果未配置，不返回该项。
serverId	iSCSI Target 的 IQN 所在的服务器 ID。
chap.name	CHAP 认证名称。
chap.status	CHAP 认证状态： <ul style="list-style-type: none"> <li>● Enabled: 启用 CHAP 认证。</li> <li>● Disabled: 禁用 CHAP 认证。</li> </ul>
createTime	创建 iSCSI Target 的时间，unix 时间戳（UTC），精确到毫秒。
serverIds	iSCSI Target 对应的服务器 ID（仅集群版支持）。
num	Target 所在的服务器数量（仅集群版支持）。

reclaimPolicy	iSCSI Target 的回收策略： <ul style="list-style-type: none"> <li>● <b>Delete:</b> 当 iSCSI Target 关联的卷全部删除后，iSCSI Target 自动删除。</li> <li>● <b>Retain:</b> 当 iSCSI Target 关联的卷全部删除后，iSCSI Target 仍然保留。</li> </ul>
status	iSCSI Target 的状态： <ul style="list-style-type: none"> <li>● <b>Deleting:</b> iSCSI Target 正在删除。</li> </ul> 仅 iSCSI Target 处于删除中会返回此项。

● 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 100 个的查询结果。

```
GET /rest/v1/block/target?filter=targetName:target and lunName:lun&range=1-100 HTTP/1.1
Date: Thu, 28 Nov 2024 08:50:51 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.32:1443
```

● 响应示例 1

```
HTTP/1.1 200 OK
x-hblock-request-id: 109f3dbea7f147c5b4b0b006a01def65
Connection: keep-alive
Content-Length: 1262
Date: Thu, 28 Nov 2024 08:50:51 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": "192.168.0.32",
        "port": 3260
      }
    ],
    "targetPortalIP": {
      "ips": [
        {
          "ip": "10.1.1.10",
          "port": 5000
        }
      ],
      "status": "Disabled"
    },
    "serverId": "hblock_1"
  }
],
"createTime": 1729751771424,
"num": 1,
"reclaimPolicy": "Retain"
},
{
  "targetName": "targetb",
  "maxSessions": 3,
  "iSCSITargets": [
    {
      "iqn": "iqn.2012-08.cn.ctyunapi.oos:targetb.2",
      "ips": [
        {
```

```
        "ip": "192.168.0.32",
        "port": 3260
      }
    ],
    "targetPortalIP": {
      "ips": [
        {
          "ip": "10.1.1.10",
          "port": 5000
        }
      ],
      "status": "Disabled"
    },
    "serverId": "hblock_1"
  }
],
"createTime": 1729751837717,
"num": 1,
"reclaimPolicy": "Retain"
},
{
  "targetName": "targetc1",
  "maxSessions": 1,
  "iSCSITargets": [
    {
      "iqn": "iqn.2012-08.cn.ctyunapi.oos:targetc1.3",
      "ips": [
        {
          "ip": "192.168.0.32",
          "port": 3260
        }
      ],
      "targetPortalIP": {
        "ips": [
          {
```

```
        "ip": "10.1.1.10",
        "port": 5000
      }
    ],
    "status": "Disabled"
  },
  "serverId": "hblock_1"
}
],
"createTime": 1729751942031,
"num": 1,
"reclaimPolicy": "Retain"
},
{
  "targetName": "targetwry1",
  "maxSessions": 1,
  "iSCSITargets": [
    {
      "iqn": "iqn.2012-08.cn.ctyunapi.oos:targetwry1.4",
      "ips": [
        {
          "ip": "192.168.0.32",
          "port": 3260
        }
      ],
      "targetPortalIP": {
        "ips": [
          {
            "ip": "10.1.1.10",
            "port": 5000
          }
        ]
      },
      "status": "Disabled"
    }
  ],
  "serverId": "hblock_1"
}
```



```

        }
    ],
    "createTime": 1730274298224,
    "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 100 个的查询结果。

```

GET /rest/v1/block/target?filter=targetName:target0 and lunName:lun0&range=1-100 HTTP/1.1
Date: Fri, 29 Nov 2024 11:41:22 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.192:1443

```

### ● 响应示例 2

```

HTTP/1.1 200 OK
x-hblock-request-id: 2fe3309a0e7c4171bde34ad4923222d0
Connection: keep-alive
Content-Length: 1774
Date: Fri, 29 Nov 2024 11:41:22 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "targets": [
      {
        "targetName": "target01",
        "maxSessions": 1,

```

```
"iSCSITargets": [  
  {  
    "iqn": "iqn.2012-08.cn.ctyunapi.oos:target01.3",  
    "ips": [  
      {  
        "ip": "192.168.0.117",  
        "port": 3260  
      }  
    ],  
    "serverId": "hblock_2"  
  },  
  {  
    "iqn": "iqn.2012-08.cn.ctyunapi.oos:target01.4",  
    "ips": [  
      {  
        "ip": "192.168.0.102",  
        "port": 3260  
      }  
    ],  
    "targetPortalIP": {  
      "ips": [  
        {  
          "ip": "10.0.1.10",  
          "port": 3223  
        }  
      ],  
      "status": "Disabled"  
    },  
    "serverId": "hblock_3"  
  }  
],  
"chap": {  
  "name": "test",  
  "status": "Disabled"  
},
```

```
"createTime": 1732591782678,
"serverIds": [
  "hblock_2",
  "hblock_3"
],
"num": 2,
"reclaimPolicy": "Delete"
},
{
  "targetName": "target02",
  "maxSessions": 1,
  "iSCSITargets": [
    {
      "iqn": "iqn.2012-08.cn.ctyunapi.oos:target02.5",
      "ips": [
        {
          "ip": "192.168.0.192",
          "port": 3260
        }
      ],
      "serverId": "hblock_1"
    },
    {
      "iqn": "iqn.2012-08.cn.ctyunapi.oos:target02.6",
      "ips": [
        {
          "ip": "192.168.0.117",
          "port": 3260
        }
      ],
      "serverId": "hblock_2"
    }
  ],
  "createTime": 1732592183413,
  "serverIds": [
```

```
        "hblock_1",
        "hblock_2"
    ],
    "num": 2,
    "reclaimPolicy": "Delete"
},
{
    "targetName": "target03",
    "maxSessions": 1,
    "iSCSITargets": [
        {
            "iqn": "iqn.2012-08.cn.ctyunapi.oos:target03.7",
            "ips": [
                {
                    "ip": "192.168.0.192",
                    "port": 3260
                }
            ],
            "serverId": "hblock_1"
        },
        {
            "iqn": "iqn.2012-08.cn.ctyunapi.oos:target03.8",
            "ips": [
                {
                    "ip": "192.168.0.102",
                    "port": 3260
                }
            ],
            "targetPortalIP": {
                "ips": [
                    {
                        "ip": "10.0.1.10",
                        "port": 3223
                    }
                ]
            },
        ],
    ]
}
```

```
        "status": "Disabled"
      },
      "serverId": "hblock_3"
    }
  ],
  "createTime": 1732592245821,
  "serverIds": [
    "hblock_1",
    "hblock_3"
  ],
  "num": 2,
  "reclaimPolicy": "Delete"
},
{
  "targetName": "target05",
  "maxSessions": 1,
  "iSCSITargets": [
    {
      "iqn": "iqn.2012-08.cn.ctyunapi.oos:target05.9",
      "ips": [
        {
          "ip": "192.168.0.117",
          "port": 3260
        }
      ],
      "serverId": "hblock_2"
    },
    {
      "iqn": "iqn.2012-08.cn.ctyunapi.oos:target05.10",
      "ips": [
        {
          "ip": "192.168.0.192",
          "port": 3260
        }
      ],
    },
  ],
}
```

```
        "serverId": "hblock_1"
      }
    ],
    "createTime": 1732843975141,
    "serverIds": [
      "hblock_1",
      "hblock_2"
    ],
    "num": 2,
    "reclaimPolicy": "Delete",
    "status": "Deleting"
  }
]
}
```

## 4.4.9 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	描述	是否必须
targetName	iSCSI Target 名称。 说明：如果不填写此项，会返回所有 Target 信息。 类型：字符串 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	描述
targetName	iSCSI Target 名称。
maxSessions	iSCSI Target 下每个 IQN 允许建立的最大会话数。
iSCSITargets	iSCSI Target 属性集合。包含：iqn、ips、targetPortalIP、serverId。 类型：数组
iqn	iSCSI Target IQN。
ips	iSCSI Target 的 IP 属性集合，包含：ip、port。 类型：数组

ip	iSCSI Target 的 IP。
port	iSCSI Target 的端口号。
targetPortalIP.ips	iSCSI 目标门户 IP 属性集合，包含：ip、port。 类型：数组
ip	iSCSI 目标门户 IP。 说明：如果未配置，不返回该项。
port	iSCSI 目标门户端口号。 说明：如果未配置，不返回该项。
targetPortalIP.status	iSCSI 目标门户 IP 状态： <ul style="list-style-type: none"> <li>● Enabled：启用状态。</li> <li>● Disabled：停用状态。</li> </ul> 说明：如果未配置，不返回该项。
serverId	iSCSI Target 的 IQN 所在的服务器 ID。
luns	关联卷的属性，包含：lunName、number。 类型：数组
lunName	关联卷的名称。
number	卷编号。
chap.name	CHAP 认证名称。
chap.status	CHAP 认证状态： <ul style="list-style-type: none"> <li>● Enabled：启用 CHAP 认证。</li> <li>● Disabled：禁用 CHAP 认证。</li> </ul>
createTime	创建 iSCSI Target 的时间，unix 时间戳（UTC），精确到毫秒。
serverIds	iSCSI Target 对应的服务器 ID（仅集群版支持）。
num	Target 所在的服务器数量（仅集群版支持）。
reclaimPolicy	iSCSI Target 的回收策略：



	<ul style="list-style-type: none"> <li>● <b>Delete:</b> 当 iSCSI Target 关联的卷全部删除后, iSCSI Target 自动删除。</li> <li>● <b>Retain:</b> 当 iSCSI Target 关联的卷全部删除后, iSCSI Target 仍然保留。</li> </ul>
<b>status</b>	iSCSI Target 的状态: <ul style="list-style-type: none"> <li>● <b>Deleting:</b> iSCSI Target 正在删除。</li> </ul> 仅 iSCSI Target 处于删除中会返回此项。

### ● 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: Thu, 28 Nov 2024 08:50:36 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: 06d040d876de41f6b0cb1f6cd4a21e30
Connection: keep-alive
Content-Length: 356
Date: Thu, 28 Nov 2024 08:50:36 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": "192.168.0.32",  
        "port": 3260  
      }  
    ],  
    "targetPortalIP": {  
      "ips": [  
        {  
          "ip": "10.1.1.10",  
          "port": 5000  
        }  
      ],  
      "status": "Disabled"  
    },  
    "serverId": "hblock_1"  
  }  
],  
"luns": [  
  {  
    "lunName": "LUNa1",  
    "number": 0  
  }  
],  
"createTime": 1729751771424,  
"num": 1,  
"reclaimPolicy": "Retain"  
}  
}
```

## ● 请求示例 2

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

```
GET /rest/v1/block/target/target01 HTTP/1.1
```

```
Date: Thu, 28 Nov 2024 08:19:14 GMT
```

```
Authorization: HBlock userName:signature  
Host: 192.168.0.117:1443  
Connection: keep-alive
```

## ● 响应示例 2

```
HTTP/1.1 200 OK  
x-hblock-request-id: c97e1e4472194522a22bd6d1577480a2  
Connection: keep-alive  
Content-Length: 585  
Date: Thu, 28 Nov 2024 08:19:14 GMT  
Content-Type: application/json;charset=utf-8  
Server: HBlock  
  
{  
  "data": {  
    "targetName": "target01",  
    "maxSessions": 1,  
    "iSCSITargets": [  
      {  
        "iqn": "iqn.2012-08.cn.ctyunapi.oos:target01.3",  
        "ips": [  
          {  
            "ip": "192.168.0.117",  
            "port": 3260  
          }  
        ],  
        "serverId": "hblock_2"  
      },  
      {  
        "iqn": "iqn.2012-08.cn.ctyunapi.oos:target01.4",  
        "ips": [  
          {  
            "ip": "192.168.0.102",  
            "port": 3260  
          }  
        ]  
      }  
    ]  
  }  
}
```

```
    ],
    "targetPortalIP": {
      "ips": [
        {
          "ip": "10.0.1.10",
          "port": 3223
        }
      ],
      "status": "Disabled"
    },
    "serverId": "hblock_3"
  }
],
"luns": [
  {
    "lunName": "lun01a",
    "number": 0
  },
  {
    "lunName": "lun01b",
    "number": 1
  }
],
"chap": {
  "name": "test",
  "status": "Disabled"
},
"createTime": 1732591782678,
"serverIds": [
  "hblock_2",
  "hblock_3"
],
"num": 2,
"reclaimPolicy": "Delete"
}
```

```
}

```

#### 4.4.10 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	描述	是否必须
filter	查询的过滤条件。 支持的查询条件为： targetName: iSCSI Target 名称。	否
range	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	描述
connections	连接信息集合，包含：targetName、targetConnections。 类型：数组
targetName	iSCSI Target 名称。

targetConnections	iSCSI Target 连接信息集合，包含：sessionId、iqn、clientIP、clientPort、initiatorName、targetIP。 类型：数组
sessionId	会话 ID。
iqn	IQN。
clientIP	客户端的 IP。
clientPort	客户端的端口号。
initiatorName	连接所属的 initiator 名称。
targetIP	服务器端 IP。

### ● 请求示例

查询 iSCSI Target 名称含 target 的连接。

```
GET /rest/v1/block/connection/?filter=targetName:target HTTP/1.1
Date: Wed, 16 Mar 2022 02:52:21 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Host:192.168.0.121:1443
```

### ● 响应示例

```
HTTP/1.1 200 OK
x-hblock-request-id: 596a0b14ab0b482b9ee88d28c701b90d
Connection: keep-alive
Content-Length: 457
Date: Wed, 16 Mar 2022 02:52:21 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "connections": [
      {
```

```
"targetName": "target02",
"targetConnections": [
  {
    "sessionId": "0x4",
    "iqn": "iqn.2012-08.cn.ctyunapi.oos:target02.7",
    "clientIP": "192.168.0.46",
    "clientPort": 51978,
    "initiatorName": "iqn.1991-05.com.microsoft:ecs-28f3",
    "targetIP": "192.168.0.121"
  },
  {
    "sessionId": "0x5",
    "iqn": "iqn.2012-08.cn.ctyunapi.oos:target02.8",
    "clientIP": "192.168.0.46",
    "clientPort": 51980,
    "initiatorName": "iqn.1991-05.com.microsoft:ecs-28f3",
    "targetIP": "192.168.0.72"
  }
]
}
}
```

#### 4.4.11 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	描述	是否必须
targetName	删除的连接所属的 iSCSI Target 名称。 类型：字符串 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	删除的连接所属的 initiator 名称。 类型：字符串 取值： <ul style="list-style-type: none"> <li>● 若使用 Windows iSCSI 发起程序，initiatorName 为“发起程序名称”。</li> <li>● 若使用 Linux iSCSI 发起程序，initiatorName 为使用 <b>cat /etc/iscsi/initiatorname.iscsi</b> 指令取得“InitiatorName”。</li> </ul>	否
targetIP	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
```

#### 4.4.12 错误码列表

HTTP status	错误码	错误信息	描述
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	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 must contain at least 2 of the following: uppercase letters, lowercase letters, digits, underscores(_), and the length is between 12 and 16.	密码不合法：字符串形式，长度范围是 12~16，必须包含大写字母、小写字母、数字、下划线(_)中的至少两种字符，字母区分大小写。
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(.) or hyphens(-), 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	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 exist when migrating Target server.	迁移 Target 时，源和目的服务器 ID 必须存在。
400	StandaloneModeNotAllowed	'operation' 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	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	InvalidServerStatus	The server 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 again.	Target 有关联的卷。
409	TargetAlreadyExists	The Target with name <i>targetName</i> already exists.	Target 已经存在。

## 4.5 存储池（集群版）

### 4.5.1 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
}
```

#### ● 请求参数

参数	描述	是否必须
poolName	存储池名称。 类型：字符串 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	指定存储池的故障域级别。 类型：枚举 取值： <ul style="list-style-type: none"> <li>● path：数据目录级别。</li> </ul>	否

	<ul style="list-style-type: none"> <li>● <b>server</b>: 服务器级别。</li> <li>● <b>rack</b>: 机架级别。</li> <li>● <b>room</b>: 机房级别。</li> </ul> 默认值为 <b>server</b> 。	
<b>description</b>	存储池的描述信息。 类型: 字符串 取值: 1~50 位字符串。	否

### ● 请求示例

创建存储池 **pool4**。

```

POST /rest/v1/system/storagepool HTTP/1.1
Date: Tue, 18 Jun 2024 08:41:36 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 104
Host: 192.168.0.110:1443

{
  "poolName": "pool4",
  "faultDomain": "path",
  "description": "The pool is path level."
}
    
```

### ● 响应示例

```

HTTP/1.1 200 OK
x-hblock-request-id: 2a6ccdba8d284dbd8d8aa7aaadc3f050
Connection: keep-alive
Content-Length: 0
Date: Tue, 18 Jun 2024 08:41:36 GMT
Server: HBlock
    
```

## 4.5.2 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	描述	是否必须
poolName	存储池名称。 类型：字符串 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	指定加入存储池的节点，该节点必须属于集群拓扑节点。 <b>说明：</b> <ul style="list-style-type: none"> <li>● 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。</li> <li>● 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。</li> </ul> 类型：字符串列表	是

## ● 请求示例

为存储池 pool14 添加存储节点。

```

PUT /rest/v1/system/storagepool/pool14/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: bcdb355abb8d481fb17bd2c40affb5bc
Connection: keep-alive
Date: Tue, 18 Jun 2024 08:41:36 GMT
Server: HBlock
```

### 4.5.3 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	描述	是否必须
poolName	存储池名称。 类型：字符串	是
newName	存储池的新名称。 类型：字符串 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	存储池的描述信息。 类型：字符串 取值：1~50 位字符串。	否

- 请求示例

修改存储池 `pool4` 的名称为 `storagepool-4`，并修改描述信息。

```
PUT /rest/v1/system/storagepool/pool4 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
```

## 4.5.4 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，强制移除节点可能造成数据丢失。

### ● 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	描述	是否必须
poolName	存储池名称。 类型：字符串 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	要移除的节点。一次可以移除多个同一故障域的节点。一次不能移除涉及多个故障域的节点。 类型：字符串列表	是
force	是否强制移除存储池内的节点。 <b>注意：</b> 强制将节点从存储池中移除，可能造成数据丢失。 类型：布尔 取值： <ul style="list-style-type: none"> <li>● <b>true</b>：强制移除存储池内的节点。</li> <li>● <b>false</b>：不强制移除存储池内的节点。</li> </ul> 默认值为 <b>false</b> 。	否
deleteLocalData	将节点从存储池移除，是否删除存储在该节点上的 HBlock 数据。 类型：布尔 取值： <ul style="list-style-type: none"> <li>● <b>true</b>：删除存储在该节点上的 HBlock 数据。</li> <li>● <b>false</b>：不删除存储在该节点上的 HBlock 数据。</li> </ul> 默认值为 <b>false</b> 。	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
```

## 4.5.5 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	描述	是否必须
poolName	存储池名称。 类型：字符串 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
```

## 4.5.6 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	描述	是否必须
filter	查询的过滤条件。 支持的过滤条件类型包括： <ul style="list-style-type: none"> <li>● poolName: 存储池的节点名称。</li> </ul>	否

### ● 响应结果

名称	描述
storagePool	存储池信息集合，包含：name、id、isBasePool、status、faultDomain、description、totalCapacity、usedCapacity、createTime、luns、topology。
name	存储池名称。
id	存储池序号。
isBasePool	是否是基础存储池： <ul style="list-style-type: none"> <li>● true: 是基础存储池。</li> </ul>

	<ul style="list-style-type: none"> <li>● <b>false</b>: 不是基础存储池。</li> </ul>
status	存储池状态: <ul style="list-style-type: none"> <li>● <b>Normal</b>: 正常。</li> <li>● <b>Deleting</b>: 删除中。</li> </ul>
faultDomain	存储池故障域级别: <ul style="list-style-type: none"> <li>● <b>path</b>: 数据目录级别。</li> <li>● <b>server</b>: 服务器级别。</li> <li>● <b>rack</b>: 机架级别。</li> <li>● <b>room</b>: 机房级别。</li> </ul>
description	存储池描述信息。
totalCapacity	存储池总容量, 单位是字节。
usedCapacity	存储池已用容量, 单位是字节。
createTime	存储池创建时间。
luns.lunNamesForCachePool	缓存存储池卷列表。
luns.lunNamesForPool	存储池卷列表。
luns.status.number	存储池关联卷的个数。
luns.status.cacheNumber	缓存模式卷的个数。
luns.status.storageNumber	存储模式卷的个数。
luns.status.localNumber	本地模式卷的个数。
luns.status.cacheCapacity	缓存模式卷容量, 单位是字节。
luns.status.storageCapacity	存储模式卷容量, 单位是字节。
luns.status.localCapacity	本地模式卷容量, 单位是字节。
topology 为存储池拓扑信息	
name	拓扑节点或者子节点名称。
fullname	拓扑节点或者子节点的全路径名称。
type	拓扑节点或者子节点的类型。
description	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		故障域状态： <ul style="list-style-type: none"> <li>● <b>Healthy</b>: 健康状态，可正常读写。</li> <li>● <b>Warning</b>: 警告状态，可读。</li> <li>● <b>Error</b>: 错误状态，无法访问。</li> </ul>
如果拓扑节点或者子节点的类型是服务器 server	clusterAddress	拓扑图集群网络 IP。
	publicAddress	拓扑图业务网络 IP。
	allDiskPaths.usedCapacity	拓扑图中所有数据目录的已用容量。
	allDiskPaths.totalCapacity	拓扑图中所有数据目录的总容量。
	serverStatus	拓扑图的服务器状态： <ul style="list-style-type: none"> <li>● <b>Connected</b>: 已连接。</li> <li>● <b>Disconnected</b>: 未连接。</li> <li>● <b>Removing</b>: 移除中。</li> </ul>
	serverId	拓扑图的服务器 ID。
	childNodes	拓扑图中子节点。
如果拓扑节点或者子节点的类型是数据目录 path	usedCapacity	HBlock 数据目录对应分区的已用容量，单位是字节。
	totalCapacity	HBlock 数据目录对应分区的总容量，单位是字节。
	usedCapacityQuota	数据目录已用容量配额，单位是字节。
	capacityQuota	数据目录容量配额，单位是字节。
	managementStatus	数据目录的管理状态： <ul style="list-style-type: none"> <li>● <b>Added</b>: 已添加。</li> <li>● <b>Removing</b>: 正在移除。</li> </ul>
	healthStatus	数据目录健康状态：

		<ul style="list-style-type: none"> <li>● <b>Healthy:</b> 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 90%。</li> <li>● <b>Warning:</b> 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 90%；磁盘剩余空间不足 1GiB；或者 HBlock 对这个目录停写。</li> <li>● <b>Error:</b> 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。</li> </ul>
	healthDetail	数据目录健康状态详情： <ul style="list-style-type: none"> <li>● 如果健康状态为 <b>Healthy</b>，此字段为空。</li> <li>● 如果健康状态为 <b>Warning</b> 或 <b>Error</b>，显示警告或错误的详细信息。</li> </ul>
	serverId	数据目录所属的服务器 ID。

### ● 请求示例

查询所有存储池。

```
GET /rest/v1/system/storagepool HTTP/1.1
Date: Fri, 02 Aug 2024 09:59:58 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: 01964c0c63ed47e88bf2dea0820baac3
Connection: keep-alive
Content-Length: 3994
Date: Fri, 02 Aug 2024 09:59:58 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "storagePool": [
      {
        "name": "default",
        "id": 0,
        "isBasePool": true,
        "status": "Normal",
        "faultDomain": "server",
        "description": "base pool",
        "totalCapacity": 336471511040,
        "usedCapacity": 38534676480,
        "createTime": 1722494921245,
        "luns": {
          "lunNamesForCachePool": [
            "lun03a"
          ],
          "lunNamesForPool": [
            "lun02a",
            "lun01b",
            "lun01a"
          ],
          "status": {
            "number": 4,
            "cacheNumber": 1,
            "storageNumber": 1,
            "localNumber": 2,
            "cacheCapacity": 214748364800,
            "storageCapacity": 322122547200,
            "localCapacity": 214748364800
          }
        }
      },
      "topology": [
```

```
{
  "name": "server1",
  "fullname": "default:room1:rack2:server1",
  "type": "server",
  "healthStatus": "Healthy",
  "clusterAddress": "192.168.0.192",
  "publicAddress": "192.168.0.192",
  "allDiskPaths": {
    "usedCapacity": 18884603904,
    "totalCapacity": 136132825088
  },
  "serverStatus": "Connected",
  "serverId": "hblock_1",
  "childNodes": [
    {
      "name": "/mnt/stor",
      "fullname": "default:room1:rack2:server1:/mnt/stor",
      "type": "path",
      "usedCapacity": 11510775808,
      "totalCapacity": 100169342976,
      "usedCapacityQuota": 172032,
      "capacityQuota": -1,
      "managementStatus": "Added",
      "healthStatus": "Healthy",
      "healthDetail": [],
      "serverId": "hblock_1"
    },
    {
      "name": "/mnt/storage01",
      "fullname":
"default:room1:rack2:server1:/mnt/storage01",
      "type": "path",
      "usedCapacity": 7373828096,
      "totalCapacity": 35963482112,
      "usedCapacityQuota": 0,
      "capacityQuota": -1,
      "managementStatus": "Added",
      "healthStatus": "Healthy",
      "healthDetail": [],
      "serverId": "hblock_1"
    }
  ]
}
```

```
]
},
{
  "name": "server2",
  "fullname": "default:room1:server2",
  "type": "server",
  "healthStatus": "Healthy",
  "clusterAddress": "192.168.0.110",
  "publicAddress": "192.168.0.110",
  "allDiskPaths": {
    "usedCapacity": 8582782976,
    "totalCapacity": 100169342976
  },
  "serverStatus": "Connected",
  "serverId": "hblock_2",
  "childNodes": [
    {
      "name": "/mnt/stor",
      "fullname": "default:room1:server2:/mnt/stor",
      "type": "path",
      "usedCapacity": 8582782976,
      "totalCapacity": 100169342976,
      "usedCapacityQuota": 0,
      "capacityQuota": -1,
      "managementStatus": "Added",
      "healthStatus": "Healthy",
      "healthDetail": [],
      "serverId": "hblock_2"
    }
  ]
},
{
  "name": "server3",
  "fullname": "default:room1:server3",
  "type": "server",
  "healthStatus": "Healthy",
  "clusterAddress": "192.168.0.102",
  "publicAddress": "192.168.0.102",
  "allDiskPaths": {
    "usedCapacity": 11067289600,
    "totalCapacity": 100169342976
  }
}
```



```

        },
        "serverStatus": "Connected",
        "serverId": "hblock_3",
        "childNodes": [
            {
                "name": "/mnt/stor",
                "fullname": "default:room1:server3:/mnt/stor",
                "type": "path",
                "usedCapacity": 11067289600,
                "totalCapacity": 100169342976,
                "usedCapacityQuota": 0,
                "capacityQuota": -1,
                "managementStatus": "Added",
                "healthStatus": "Healthy",
                "healthDetail": [],
                "serverId": "hblock_3"
            }
        ]
    }
},
{
    "name": "pool12",
    "id": 1,
    "isBasePool": false,
    "status": "Normal",
    "faultDomain": "path",
    "description": "second pool",
    "totalCapacity": 127955415040,
    "usedCapacity": 29149364224,
    "createTime": 1722499235201,
    "luns": {
        "lunNamesForCachePool": [
            "lun02a"
        ],
        "lunNamesForPool": [
            "lun03a"
        ],
        "status": {
            "number": 2,
            "cacheNumber": 1,

```

```
        "storageNumber": 1,
        "localNumber": 0,
        "cacheCapacity": 214748364800,
        "storageCapacity": 322122547200,
        "localCapacity": 0
    }
},
"topology": [
    {
        "name": "/mnt/storage02",
        "fullname": "default:rack1:server4:/mnt/storage02",
        "type": "path",
        "usedCapacity": 57221120,
        "totalCapacity": 56028450816,
        "usedCapacityQuota": 69632,
        "capacityQuota": -1,
        "managementStatus": "Added",
        "healthStatus": "Healthy",
        "healthDetail": [],
        "serverId": "hblock_4"
    },
    {
        "name": "/mnt/storage04",
        "fullname": "default:rack1:server4:/mnt/storage04",
        "type": "path",
        "usedCapacity": 14546071552,
        "totalCapacity": 35963482112,
        "usedCapacityQuota": 69632,
        "capacityQuota": -1,
        "managementStatus": "Added",
        "healthStatus": "Healthy",
        "healthDetail": [],
        "serverId": "hblock_4"
    },
    {
        "name": "/mnt/storage05",
        "fullname": "default:rack1:server4:/mnt/storage05",
        "type": "path",
        "usedCapacity": 14546071552,
        "totalCapacity": 35963482112,
        "usedCapacityQuota": 69632,
```

```
        "capacityQuota": -1,
        "managementStatus": "Added",
        "healthStatus": "Healthy",
        "healthDetail": [],
        "serverId": "hblock_4"
    }
]
},
{
    "name": "pool3",
    "id": 2,
    "isBasePool": false,
    "status": "Normal",
    "faultDomain": "path",
    "description": "",
    "totalCapacity": 0,
    "usedCapacity": 0,
    "createTime": 1722588962393,
    "luns": {
        "lunNamesForCachePool": [],
        "lunNamesForPool": [],
        "status": {}
    },
    "topology": []
}
]
```

## 4.5.7 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	描述	是否必须
poolName	存储池名称。 类型：字符串 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	描述
storagePool	存储池信息集合，包含：name、id、isBasePool、status、faultDomain、description、totalCapacity、usedCapacity、createTime、luns、topology。
name	存储池名称。
id	存储池序号。

isBasePool	是否是基础存储池： <ul style="list-style-type: none"> <li>● true: 是基础存储池。</li> <li>● false: 不是基础存储池。</li> </ul>
status	存储池状态： <ul style="list-style-type: none"> <li>● Normal: 正常。</li> <li>● Deleting: 删除中。</li> </ul>
faultDomain	存储池故障域级别： <ul style="list-style-type: none"> <li>● path: 数据目录级别。</li> <li>● server: 服务器级别。</li> <li>● rack: 机架级别。</li> <li>● room: 机房级别。</li> </ul>
description	存储池描述信息。
totalCapacity	存储池总容量，单位是字节。
usedCapacity	存储池已用容量，单位是字节。
createTime	存储池创建时间。
luns.lunNamesForCachePool	缓存存储池卷列表。
luns.lunNamesForPool	存储池卷列表。
luns.status.number	存储池关联卷的个数。
luns.status.cacheNumber	缓存模式卷的个数。
luns.status.storageNumber	存储模式卷的个数。
luns.status.localNumber	本地模式卷的个数。
luns.status.cacheCapacity	缓存模式卷容量，单位是字节。
luns.status.storageCapacity	存储模式卷容量，单位是字节。
luns.status.localCapacity	本地模式卷容量，单位是字节。
topology 为存储池拓扑信息	
name	拓扑节点或者子节点名称。
fullname	拓扑节点或者子节点的全路径名称。

type		拓扑节点或者子节点的类型。
description		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		故障域状态： <ul style="list-style-type: none"> <li>● <b>Healthy</b>: 健康状态，可正常读写。</li> <li>● <b>Warning</b>: 警告状态，可读。</li> <li>● <b>Error</b>: 错误状态，无法访问。</li> </ul>
如果拓扑节点者子节点的类型是服务器 server	clusterAddress	拓扑图集群网络 IP。
	publicAddress	拓扑图业务网络 IP。
	allDiskPaths.usedCapacity	拓扑图中所有数据目录的已用容量。
	allDiskPaths.totalCapacity	拓扑图中所有数据目录的总容量。
	serverStatus	拓扑图的服务器状态： <ul style="list-style-type: none"> <li>● <b>Connected</b>: 已连接。</li> <li>● <b>Disconnected</b>: 未连接。</li> <li>● <b>Removing</b>: 移除中。</li> </ul>
	serverId	拓扑图的服务器 ID。
	childNodes	拓扑图中子节点。
如果拓扑节点或者子节点的类型是数据目录 path	usedCapacity	HBlock 数据目录对应分区的总容量，单位是字节。
	totalCapacity	HBlock 数据目录对应分区的已用容量，单位是字节。
	usedCapacityQuota	数据目录容量配额，单位是字节。
	capacityQuota	数据目录已用容量配额，单位是字节。
	managementStatus	数据目录的管理状态： <ul style="list-style-type: none"> <li>● <b>Added</b>: 已添加。</li> <li>● <b>Removing</b>: 正在移除。</li> </ul>

	healthStatus	数据目录健康状态： <ul style="list-style-type: none"> <li>● <b>Healthy</b>: 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 90%。</li> <li>● <b>Warning</b>: 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 90%；磁盘剩余空间不足 1GiB；或者 HBlock 对这个目录停写。</li> <li>● <b>Error</b>: 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。</li> </ul>
	healthDetail	数据目录健康状态详情： <ul style="list-style-type: none"> <li>● 如果健康状态为 <b>Healthy</b>，此字段为空。</li> <li>● 如果健康状态为 <b>Warning</b> 或 <b>Error</b>，显示警告或错误的详细信息。</li> </ul>
	serverId	数据目录所属的服务器 ID。

- 请求示例

查询存储池 default。

```
GET /rest/v1/system/storagepool/default HTTP/1.1
Date: Fri, 2 Aug 2024 10:01:57 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: b63c500caa0141c79278d1b39aa25892
Connection: keep-alive
Content-Length: 2467
Date: Fri, 2 Aug 2024 10:01:57 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock
```

```
{
  "data": {
    "name": "default",
    "id": 0,
    "isBasePool": true,
    "status": "Normal",
    "faultDomain": "server",
    "description": "base pool",
    "totalCapacity": 336471511040,
    "usedCapacity": 38539456512,
    "createTime": 1722494921245,
    "luns": {
      "lunNamesForCachePool": [
        "lun03a"
      ],
      "lunNamesForPool": [
        "lun02a",
        "lun01b",
        "lun01a"
      ],
      "status": {
        "number": 4,
        "cacheNumber": 1,
        "storageNumber": 1,
        "localNumber": 2,
        "cacheCapacity": 214748364800,
        "storageCapacity": 322122547200,
        "localCapacity": 214748364800
      }
    }
  }
}
```



```
    }
  },
  "topology": [
    {
      "name": "server1",
      "fullname": "default:room1:rack2:server1",
      "type": "server",
      "healthStatus": "Healthy",
      "clusterAddress": "192.168.0.192",
      "publicAddress": "192.168.0.192",
      "allDiskPaths": {
        "usedCapacity": 18889383936,
        "totalCapacity": 136132825088
      },
      "serverStatus": "Connected",
      "serverId": "hblock_1",
      "childNodes": [
        {
          "name": "/mnt/stor",
          "fullname": "default:room1:rack2:server1:/mnt/stor",
          "type": "path",
          "usedCapacity": 11510775808,
          "totalCapacity": 100169342976,
          "usedCapacityQuota": 172032,
          "capacityQuota": -1,
          "managementStatus": "Added",
          "healthStatus": "Healthy",
          "healthDetail": [],
          "serverId": "hblock_1"
        },
        {
          "name": "/mnt/storage01",
          "fullname": "default:room1:rack2:server1:/mnt/storage01",
          "type": "path",
          "usedCapacity": 7378608128,
          "totalCapacity": 35963482112,
          "usedCapacityQuota": 0,
          "capacityQuota": -1,
          "managementStatus": "Added",
          "healthStatus": "Healthy",
          "healthDetail": []
        }
      ]
    }
  ]
}
```

```
        "serverId": "hblock_1"
      }
    ]
  },
  {
    "name": "server2",
    "fullname": "default:room1:server2",
    "type": "server",
    "healthStatus": "Healthy",
    "clusterAddress": "192.168.0.110",
    "publicAddress": "192.168.0.110",
    "allDiskPaths": {
      "usedCapacity": 8582782976,
      "totalCapacity": 100169342976
    },
    "serverStatus": "Connected",
    "serverId": "hblock_2",
    "childNodes": [
      {
        "name": "/mnt/stor",
        "fullname": "default:room1:server2:/mnt/stor",
        "type": "path",
        "usedCapacity": 8582782976,
        "totalCapacity": 100169342976,
        "usedCapacityQuota": 229376,
        "capacityQuota": -1,
        "managementStatus": "Added",
        "healthStatus": "Healthy",
        "healthDetail": [],
        "serverId": "hblock_2"
      }
    ]
  },
  {
    "name": "server3",
    "fullname": "default:room1:server3",
    "type": "server",
    "healthStatus": "Healthy",
    "clusterAddress": "192.168.0.102",
    "publicAddress": "192.168.0.102",
    "allDiskPaths": {
```

```
        "usedCapacity": 11067289600,
        "totalCapacity": 100169342976
    },
    "serverStatus": "Connected",
    "serverId": "hblock_3",
    "childNodes": [
        {
            "name": "/mnt/stor",
            "fullname": "default:room1:server3:/mnt/stor",
            "type": "path",
            "usedCapacity": 11067289600,
            "totalCapacity": 100169342976,
            "usedCapacityQuota": 0,
            "capacityQuota": -1,
            "managementStatus": "Added",
            "healthStatus": "Healthy",
            "healthDetail": [],
            "serverId": "hblock_3"
        }
    ]
}
}
```

### 4.5.8 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	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 storage pool, or the node is related to multiple fault domains in the base pool,	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



		removing node may result in a system crash.	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。

## 4.6 集群拓扑（集群版）

### 4.6.1 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	描述	是否必须
nodeName	拓扑节点名称。 类型：字符串 取值：字符串形式，长度范围 1~63，只能由字母、数字、句点(.)、下划线（_）和短横线(-)组成，字母区分大小写，且仅支持以字母或数字开头。	是
nodeType	拓扑节点的类型。 类型：枚举 取值： <ul style="list-style-type: none"> <li>● rack：机架类型。</li> </ul>	是

	<ul style="list-style-type: none"> <li>● <b>room</b>: 机房类型。</li> </ul>	
description	节点描述信息。 类型: 字符串 取值: 1~50 位字符串。	否
parentName	父节点名称。 类型: 字符串 取值: 字符串形式, 长度范围 1~63, 只能由字母、数字、句点(.)、下划线(_)和短横线(-)组成, 字母区分大小写, 且仅支持以字母或数字开头。 默认为根节点。	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

## 4.6.2 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	描述	是否必须
nodeName	拓扑节点名称。 类型：字符串 取值：字符串形式，长度范围 1~63，只能由字母、数字、句点(.)、下划线(_)和短横线(-)组成，字母区分大小写，且仅支持以字母或数字开头。	是
newName	拓扑节点的新名称。 类型：字符串 取值：字符串形式，长度范围 1~63，只能由字母、数字、句点(.)、下划线(_)和短横线(-)组成，字母区分大小写，且仅支持以字母或数字开头。	否

description	节点描述信息。 类型：字符串 取值：1~50 位字符串。	否
parentName	父节点名称。 类型：字符串 取值：字符串形式，长度范围 1~63，只能由字母、数字、句点(.)、下划线(_)和短横线(-)组成，字母区分大小写，且仅支持以字母或数字开头。 默认为根节点。	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
    
```

### 4.6.3 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	描述	是否必须
nodeName	要删除的拓扑节点。 类型：字符串	是
force	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。 <b>注意：</b> 强制删除拓扑节点，可能造成数据丢失。 类型：布尔 取值：	否

	<ul style="list-style-type: none"> <li>● <b>true</b>: 是。</li> <li>● <b>false</b>: 否。</li> </ul> 默认值为 <b>false</b> 。	
<code>deleteLocalData</code>	删除 <code>server</code> 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 <code>server</code> 类型的节点时支持此参数。 类型：布尔 取值： <ul style="list-style-type: none"> <li>● <b>true</b>: 删除拓扑节点上 HBlock 数据。</li> <li>● <b>false</b>: 不删除拓扑节点上 HBlock 数据。</li> </ul> 默认值为 <b>false</b> 。	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
```



## 4.6.4 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	描述	
name	拓扑节点名称。	
fullname	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	拓扑节点 ID。	
type	节点的类型： <b>root</b> ：根节点。	
description	节点描述。	
childNodes	子节点信息集合，根据子节点类型不同，展示的信息不同。	
子节点类 型为 room	name	子节点名称。
	fullname	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	节点 ID。
	type	节点类型。
	description	节点描述。

子节点类 型为 rack	name	子节点名称。
	fullname	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	节点 ID。
	type	节点类型。
	description	节点描述。
子节点类 型为 server	name	子节点名称。
	fullname	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	节点 ID。
	serverId	服务器 ID。
	type	节点类型。
	description	节点描述。
	clusterAddress	拓扑图集群网络 IP。
	publicAddress	拓扑图业务网络 IP。
	allDiskPaths.usedCapacity	服务器上 HBlock 数据目录对应分区的已用容量，单位是字节。
	allDiskPaths.totalCapacity	服务器上 HBlock 数据目录对应分区的总容量，单位是字节。
	serverStatus	服务器状态： <ul style="list-style-type: none"> <li>● <b>Connected:</b> 已连接。</li> <li>● <b>Disconnected:</b> 未连接。</li> <li>● <b>Removing:</b> 移除中。</li> </ul>
子节点类 型为 path	name	子节点名称。
	fullname	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	节点 ID。

<code>serverId</code>	服务器 ID。
<code>type</code>	节点类型。
<code>totalCapacity</code>	HBlock 数据目录对应分区的总容量，单位是字节。
<code>usedCapacity</code>	HBlock 数据目录对应分区的已用容量，单位是字节。
<code>capacityQuota</code>	数据目录容量配额。 负整数表示无限制写入。
<code>usedCapacityQuota</code>	数据目录已用容量配额。
<code>storagePool</code>	所属资源池。
<code>healthStatus</code>	数据目录健康状态： <ul style="list-style-type: none"> <li>● <b>Healthy</b>: 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 90%。</li> <li>● <b>Warning</b>: 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 90%；磁盘剩余空间不足 1GiB；或者 HBlock 对这个目录停写。</li> <li>● <b>Error</b>: 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。</li> </ul>
<code>healthDetail</code>	数据目录健康状态详情： <ul style="list-style-type: none"> <li>● 如果健康状态为 <b>Healthy</b>，此字段为空。</li> <li>● 如果健康状态为 <b>Warning</b> 或 <b>Error</b>，显示警告或错误的详细信息。</li> </ul>

● 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": []
  }
]
}
}
```

## 4.6.5 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	描述	是否必须
nodeName	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	描述
name	拓扑节点名称。
fullname	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	拓扑节点 ID。
type	节点的类型： <ul style="list-style-type: none"> <li>● server: 服务器类型。</li> <li>● rack: 机架类型。</li> <li>● room: 机房类型。</li> <li>● root: 根节点。</li> </ul>

description		节点描述。
childNodes		子节点信息集合，根据子节点类型不同，展示的信息不同。
子节点类型为 room	name	子节点名称。
	fullname	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	节点 ID。
	type	节点类型。
	description	节点描述。
子节点类型为 rack	name	子节点名称。
	fullname	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	节点 ID。
	type	节点类型。
	description	节点描述。
子节点类型为 server	name	子节点名称。
	fullname	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	节点 ID。
	serverId	服务器 ID。
	type	节点类型。
	description	节点描述。
	clusterAddress	拓扑图集群网络 IP。
	publicAddress	拓扑图业务网络 IP。
	allDiskPaths.used Capacity	服务器上 HBlock 数据目录对应分区的已用容量，单位是字节。

	<code>allDiskPaths.totalCapacity</code>	服务器上 HBlock 数据目录对应分区的总容量，单位是字节。
	<code>serverStatus</code>	服务器状态： <ul style="list-style-type: none"> <li>● <b>Connected</b>: 已连接。</li> <li>● <b>Disconnected</b>: 未连接。</li> <li>● <b>Removing</b>: 移除中。</li> </ul>
子节点类型为 <code>path</code>	<code>name</code>	子节点名称。
	<code>fullname</code>	节点在集群中的全路径名称，从根节点开始，使用 <code>name:name:name</code> 格式来唯一标识该节点的名称。
	<code>id</code>	节点 ID。
	<code>serverId</code>	服务器 ID。
	<code>type</code>	节点类型。
	<code>totalCapacity</code>	HBlock 数据目录对应分区的总容量，单位是字节。
	<code>usedCapacity</code>	HBlock 数据目录对应分区的已用容量，单位是字节。
	<code>capacityQuota</code>	数据目录容量配额。 负整数表示无限制写入。
	<code>usedCapacityQuota</code>	数据目录已用容量配额。
	<code>storagePool</code>	所属存储池。
	<code>healthStatus</code>	数据目录健康状态： <ul style="list-style-type: none"> <li>● <b>Healthy</b>: 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 90%。</li> <li>● <b>Warning</b>: 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 90%；磁盘剩余空间不足 1GiB；或者 HBlock 对这个目录停写。</li> </ul>

		<ul style="list-style-type: none"> <li>● <b>Error</b>: 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。</li> </ul>
	healthDetail	数据目录健康状态详情: <ul style="list-style-type: none"> <li>● 如果健康状态为 <b>Healthy</b>, 此字段为空。</li> <li>● 如果健康状态为 <b>Warning</b> 或 <b>Error</b>, 显示警告或错误的详细信息。</li> </ul>

- 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": []
  }
]
}
]
```



#### 4.6.6 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(_) or hyphens(-),	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

		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。
--	--	--	---

## 4.7 服务器管理

### 4.7.1 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	描述	是否必须
nodeName	节点名称。 类型：字符串 取值：字符串形式，长度范围 1~63，只能由字母、数字、句点(.)、下划线(_)和短横线(-)组成，字母区分大小写，且仅支持以字母或数字开头。默认使用服务器 ID 作为节点名称。	否

parentName	父节点名称。 类型：字符串 默认映射给根节点。	否
description	服务器的描述信息。 类型：字符串 取值：1~50 位字符串。	否
ip	要添加的服务器 IP。	是
apiPort	指定管理 API 端口号。 类型：整型 取值：[1, 65535]，默认值为 1443。	否
iSCSIPort	指定 iSCSI 端口号。 类型：整型 取值：[1, 65535]，默认值为 3260。	否
portRange	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。 类型：整型 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。 <b>说明：</b> 建议指定的端口范围至少包含 500 个端口。	否
diskPaths	数据目录属性集合。包括 <i>path</i> 、 <i>capacityQuota</i> 。 类型：数组	否
path	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。 <b>注意：</b> 数据目录中不能有逗号（,）。 类型：字符串列表	否

capacityQuota	<p>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。</p> <p>类型：长整型</p> <p>取值：小于数据目录所在磁盘的总容量，单位是字节。负整数表示无限制写入，0 表示禁止写入。默认不限制写入。</p> <p><b>注意：</b>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。</p>	否
ports.dataPort1	<p>数据端口 1。</p> <p>类型：整型</p> <p>取值：[1, 65535]。</p>	否
ports.managementPort1	<p>管理服务端口 1。</p> <p>类型：整型</p> <p>取值：[1, 65535]。</p>	否
ports.managementPort2	<p>管理服务端口 2。</p> <p>类型：整型</p> <p>取值：[1, 65535]。</p>	否
ports.managementPort3	<p>管理服务端口 3。</p> <p>类型：整型</p> <p>取值：[1, 65535]。</p>	否
ports.managementPort4	<p>管理服务端口 4。</p> <p>类型：整型</p> <p>取值：[1, 65535]。</p>	否
ports.managementPort5	<p>管理服务端口 5。</p> <p>类型：整型</p>	否

	取值：[1, 65535]。	
ports.managementPort6	管理服务端口 6。 类型：整型 取值：[1, 65535]。	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: Tue, 23 Jan 2024 08:16:17 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: 2707c6619d4c4eb9897ec7d826d8c901
Connection: keep-alive
Date: Tue, 23 Jan 2024 08:16:17 GMT
    
```



```
Content-Length: 0
```

```
Server: HBlock
```

## 4.7.2 修改服务器属性

此操作用来修改服务器属性。

**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。

### ● 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	描述	是否必须
----	----	------

serverId	要修改的服务器 ID。 类型：字符串	是
targetPortalIP.ips	iSCSI 目标门户 IP 属性集合，包含：ip、port。 类型：数组 <b>说明：</b> 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。	否
ip	iSCSI 目标门户 IP。 <b>说明：</b> ip 和 port 必须同时存在，或者同时不存在。	否
port	iSCSI 目标门户端口号。 <b>说明：</b> ip 和 port 必须同时存在，或者同时不存在。 类型：整型 取值：[1, 65535]。	否
targetPortalIP.status	是否启用 iSCSI 目标门户 IP。 类型：枚举 取值： <ul style="list-style-type: none"> <li>● Enabled: 启用。</li> <li>● Disabled: 禁用。</li> </ul>	是
defaultPath	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	指定端口范围，用于相关服务。 取值：整型，取值范围为[1, 65535]，port1 为端口范围最小值，port2 为端口范围最大值，且 port1 < port2。	否

	说明：建议指定的端口范围至少包含 500 个端口。	
--	---------------------------	--

- 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
```

### ● 请求示例 3

修改服务器 hblock\_1 的端口范围为 19500-20500。

```
PUT /rest/v1/system/server/hblock_1 HTTP/1.1
Date: Tue, 26 Sep 2023 06:45:38 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
```

```
Content-Length: 39
Host: 192.168.0.121:1443

{
  "portRange": "19500-20500"
}
```

- 响应示例 3

```
HTTP/1.1 204 No Content
x-hblock-request-id: 2271ef7c175c44cca2e197c12f2cb742
Connection: keep-alive
Date: Tue, 26 Sep 2023 06:45:38 GMT
Server: HBlock
```

### 4.7.3 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	描述	是否必须
serverId	要删除 targetPortalIP 配置的服务器 ID。 类型：字符串	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
```

#### 4.7.4 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	描述	是否必须
serverId	源 HBlock 服务器的 ID。 类型：字符串	是
destServerId	目的服务器 ID。 类型：字符串	是
service	需要迁移的基础服务。 类型：枚举 取值： <ul style="list-style-type: none"> <li>● mdm：元数据管理服务。</li> <li>● ls：日志服务。</li> <li>● cs：协调服务。</li> </ul>	是
metaDir	迁移服务的数据目录，用于存储基础服务的相关数据信息。 <b>说明：</b> 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。 类型：字符串 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

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
```

## 4.7.5 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。

### ● 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	描述	是否必须
serverId	要移除服务器的服务器 ID。 类型：字符串	是
force	是否强制移除服务器。 <b>注意：</b> 强制移除服务器，可能造成数据丢失。 类型：布尔 取值： <ul style="list-style-type: none"> <li>● <b>true</b>：是。</li> <li>● <b>false</b>：否。</li> </ul> 默认值为 <b>false</b> 。	否
deleteLocalData	移除服务器时，是否删除服务器上 HBlock 数据目录中的数据。 类型：布尔 取值： <ul style="list-style-type: none"> <li>● <b>true</b>：删除服务器上 HBlock 数据目录中的数据。</li> <li>● <b>false</b>：不删除服务器上 HBlock 数据目录中的数据。</li> </ul> 默认值为 <b>false</b> 。	否

### ● 响应结果

名称	描述
luns	被移除服务器上卷的属性信息集合，包含： <b>lunName</b> 、 <b>iSCSITargets</b> 。 类型：数组
lunName	被移除服务器上的卷名称。
iSCSITargets	被移除服务器上卷对应的 iSCSI Target 属性集合。包含： <b>iqn</b> 、 <b>status</b> 、 <b>ips</b> 。 类型：数组

iqn	被移除服务器上的卷对应的新 Target IQN。
status	被移除服务器上的卷对应的 Target 的状态： <ul style="list-style-type: none"> <li>● Active: 主 Target。</li> <li>● Standby: 热备 Target。</li> <li>● Offline: 离线 Target。</li> <li>● Unavailable: 冷备 Target。</li> </ul>
ips	被移除服务器上卷对应 iSCSI Target 的 IP 属性集合，包含：ip、port。 类型：数组
ip	被移除服务器上的卷对应的新 iSCSI Target 的 IP。
port	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 userName:signature
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
        }
      ]
    }
  ]
}
```

## 4.7.6 重启服务器上的 HBlock

此操作用来重启服务器上的 HBlock。

### ● 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

{
  "force": force
}
    
```

### ● 请求参数

参数	描述	是否必须
serverId	要重启的服务器 ID。	是
force	是否强制重启。 <b>注意：</b> 强制重启服务器，可能造成数据丢失。 类型：布尔 取值： <ul style="list-style-type: none"> <li>● true：强制重启。</li> <li>● false：不强制重启。</li> </ul> 默认值为 false。	否

### ● 请求示例

重启服务器 hblock\_4 上的 HBlock 服务。

```

PUT /rest/v1/system/server/hblock_4/restart HTTP/1.1
    
```

```
Date: Thu, 17 Mar 2022 02:00:12 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.121:1443
```

- 响应示例

```
HTTP/1.1 202 Accepted
x-hblock-request-id: c221ede9cf764ce39a06c3e8ea930fdd
Connection: keep-alive
Date: Thu, 17 Mar 2022 02:00:16 GMT
Content-Length: 0
Server: HBlock
```



## 4.7.7 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&range=i-j HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

### ● 请求参数

参数	描述	是否必须
filter	查询的过滤条件。 支持的查询条件为： <ul style="list-style-type: none"> <li>● <b>serverName</b>: 服务器名称。</li> <li>● <b>status</b>: 服务器状态（<b>Connected</b>: 已连接; <b>Disconnected</b>: 未连接; <b>Removing</b>: 移除中）。</li> </ul>	否
range	查询服务器的范围，格式为 <i>i-j</i> ， <i>i</i> 和 <i>j</i> 为正整数， $0 < j-i < 1000$ 。按照服务器 ID 进行排序，返回第 <i>i</i> 到第 <i>j</i> 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 <b>range</b> ，则返回所有符合条件的服务器。	否

### ● 响应结果

名称	描述
servers	服务器 HBlock 属性集合，包含： <b>serverName</b> 、 <b>serverId</b> 、 <b>status</b> 、 <b>version</b> 、 <b>isMasterServer</b> 、 <b>isBaseServer</b> 、

	publicAddress、clusterAddress、 allDiskPaths、defaultPath、 recentStartTime、ports、portRange、 iSCSIPort、storagePorts、nodeName、 parentName、description、baseServices。 类型：数组
serverName	服务器名称。
serverId	服务器 ID。
status	服务器状态： <ul style="list-style-type: none"> <li>● <b>Connected</b>: 已连接。</li> <li>● <b>Disconnected</b>: 未连接。</li> <li>● <b>Removing</b>: 移除中。</li> </ul>
version	服务器上的 HBlock 版本。
isMasterServer	该服务器是否为 Master 节点： <ul style="list-style-type: none"> <li>● <b>true</b>: 是。</li> <li>● <b>false</b>: 否。</li> </ul>
isBaseServer	该服务器是否为基础节点： <ul style="list-style-type: none"> <li>● <b>true</b>: 是。</li> <li>● <b>false</b>: 否。</li> </ul>
publicAddress.ip	业务网络的 IP。
publicAddress.port	业务网络端口号。
clusterAddress.ip	集群网络的 IP。
clusterAddress.port	集群网络端口号。
allDiskPaths.usedCapacity	当前服务器所有 HBlock 数据目录对应分区的已用容量，单位为字节。
allDiskPaths.totalCapacity	当前服务器所有 HBlock 数据目录对应分区的总容量，单位是字节。

defaultPath	默认的数据目录（仅单机版支持）。
recentStartTime	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	HBlock 端口的集合。数组，包括 portName、port。
portName	HBlock 端口名称。
port	HBlock 在当前服务器上使用的端口号。
portRange	端口范围。
iSCSIPort	iSCSI 端口号。
storagePorts	存储服务端口集合（仅集群版支持）。 数组，包括 portN、diskpath、description、pathId。
portN	存储服务端口。
diskpath	存储服务的数据目录。
pathId	数据目录 ID。
nodeName	节点名称（仅集群版支持）。
parentName	父节点名称（仅集群版支持）。
description	节点描述（仅集群版支持）。
baseServices	基础服务属性的集合（仅集群版基础服务器支持）。数组，包括 name、status、metaDir。
name	基础服务名称： <ul style="list-style-type: none"> <li>● mdm: 元数据管理服务。</li> <li>● ls: 日志服务。</li> <li>● cs: 协调服务。</li> </ul>
status	基础服务的状态： <ul style="list-style-type: none"> <li>● Up: 可用。</li> </ul>

	<ul style="list-style-type: none"> <li>● <b>Down</b>: 不可用。</li> <li>● <b>Migrating</b>: 正在迁移。</li> <li>● <b>Unkown</b>: 未知。</li> </ul>
metaDir	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。

```
GET /rest/v1/system/server HTTP/1.1
Date: Tue, 14 Jan 2025 08:41:57 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.32:1443
```

### ● 响应示例 1

```
HTTP/1.1 200 OK
Date: Tue, 14 Jan 2025 08:41:57 GMT
x-hblock-request-id: 9fac0d6f51a647bda31c8b68280652e1
Connection: keep-alive
Content-Length: 695
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "servers": [
      {
        "serverName": "ecs-9689-0915127",
        "serverId": "hblock_1",
        "status": "Connected",
        "version": "3.8.0",
        "publicAddress": {
          "ip": "192.168.0.32",
          "port": 3260
        }
      }
    ]
  }
}
```

```
    },
    "clusterAddress": {
      "ip": "192.168.0.32",
      "port": 20004
    },
    "allDiskPaths": {
      "usedCapacity": 35861303296,
      "totalCapacity": 100169342976
    },
    "defaultPath": "/mnt/stor",
    "recentStartTime": 1735873608983,
    "ports": [
      {
        "portName": "apiPort",
        "port": 1443
      },
      {
        "portName": "webPort",
        "port": 2443
      },
      {
        "portName": "managementPort1",
        "port": 20004
      },
      {
        "portName": "managementPort2",
        "port": 20000
      },
      {
        "portName": "managementPort3",
        "port": 20002
      },
      {
        "portName": "managementPort4",
        "port": 20003
      }
    ]
  }
}
```

```
    },  
    {  
      "portName": "managementPort6",  
      "port": 20001  
    }  
  ],  
  "portRange": "20000-20500",  
  "iSCSIPort": 3260  
}  
]  
}
```

### ● 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: Mon, 25 Nov 2024 09:47:00 GMT  
Authorization: HBlock userName:signature  
Host: 192.168.0.192:1443
```

### ● 响应示例 2

```
HTTP/1.1 200 OK  
Date: Mon, 25 Nov 2024 09:47:00 GMT  
x-hblock-request-id: 79a2d7b33f55452e910217c2d0cd51af  
Connection: keep-alive  
Content-Length: 2716  
Content-Type: application/json;charset=utf-8  
Server: HBlock  
  
{  
  "data": {  
    "servers": [  

```

```
{
  "serverName": "ecs-9689-0915142",
  "serverId": "hblock_2",
  "status": "Connected",
  "version": "3.8.0",
  "isMasterServer": false,
  "isBaseServer": true,
  "publicAddress": {
    "ip": "192.168.0.117",
    "port": 3260
  },
  "clusterAddress": {
    "ip": "192.168.0.117",
    "port": 20008
  },
  "allDiskPaths": {
    "usedCapacity": 18037145600,
    "totalCapacity": 35963482112
  },
  "recentStartTime": 1732519209900,
  "ports": [
    {
      "portName": "apiPort",
      "port": 1443
    },
    {
      "portName": "webPort",
      "port": 2443
    },
    {
      "portName": "managementPort1",
      "port": 20008
    },
    {
      "portName": "managementPort2",
```

```
        "port": 20013
    },
    {
        "portName": "managementPort3",
        "port": 20004
    },
    {
        "portName": "managementPort4",
        "port": 20011
    },
    {
        "portName": "managementPort5",
        "port": 20000
    },
    {
        "portName": "managementPort6",
        "port": 20010
    },
    {
        "portName": "dataPort1",
        "port": 20002
    },
    {
        "portName": "metadataPort4",
        "port": 20005
    },
    {
        "portName": "metadataPort5",
        "port": 20001
    },
    {
        "portName": "metadataPort6",
        "port": 20006
    },
    {
```



```
        "portName": "metadataPort7",
        "port": 20012
    },
    {
        "portName": "metadataPort8",
        "port": 20007
    }
],
"portRange": "20000-20500",
"iSCSIPort": 3260,
"storagePorts": [
    {
        "port1": 20003,
        "port2": 20014,
        "port3": 20009,
        "diskpath": "/mnt/storage02",
        "pathId": "1"
    }
],
"nodeName": "default:hblock_2",
"parentName": "default",
"baseServices": [
    {
        "name": "ls",
        "status": "Up",
        "metaDir": "/mnt/storage02"
    },
    {
        "name": "cs",
        "status": "Up",
        "metaDir": "/mnt/storage02"
    }
]
},
{
```

```
"serverName": "ecs-9689-0915140",
"serverId": "hblock_3",
"status": "Connected",
"version": "3.8.0",
"isMasterServer": false,
"isBaseServer": true,
"publicAddress": {
  "ip": "192.168.0.102",
  "port": 3260
},
"clusterAddress": {
  "ip": "192.168.0.102",
  "port": 20008
},
"allDiskPaths": {
  "usedCapacity": 63451136,
  "totalCapacity": 100169342976
},
"recentStartTime": 1732519207564,
"ports": [
  {
    "portName": "apiPort",
    "port": 1443
  },
  {
    "portName": "webPort",
    "port": 2443
  },
  {
    "portName": "managementPort1",
    "port": 20008
  },
  {
    "portName": "managementPort2",
    "port": 20013
  }
]
```

```
    },  
    {  
      "portName": "managementPort3",  
      "port": 20004  
    },  
    {  
      "portName": "managementPort4",  
      "port": 20011  
    },  
    {  
      "portName": "managementPort5",  
      "port": 20000  
    },  
    {  
      "portName": "managementPort6",  
      "port": 20010  
    },  
    {  
      "portName": "dataPort1",  
      "port": 20002  
    },  
    {  
      "portName": "metadataPort1",  
      "port": 20014  
    },  
    {  
      "portName": "metadataPort2",  
      "port": 20003  
    },  
    {  
      "portName": "metadataPort3",  
      "port": 20009  
    },  
    {  
      "portName": "metadataPort4",
```

```
        "port": 20005
      },
      {
        "portName": "metadataPort5",
        "port": 20001
      },
      {
        "portName": "metadataPort6",
        "port": 20006
      },
      {
        "portName": "metadataPort7",
        "port": 20012
      },
      {
        "portName": "metadataPort8",
        "port": 20007
      }
    ],
    "portRange": "20000-20500",
    "iSCSIPort": 3260,
    "storagePorts": [
      {
        "port1": 20015,
        "port2": 20017,
        "port3": 20016,
        "diskpath": "/mnt/stor01",
        "pathId": "1"
      }
    ],
    "nodeName": "default:hblock_3",
    "parentName": "default",
    "baseServices": [
      {
        "name": "mdm",
```

```
        "status": "Up",
        "metaDir": "/mnt/stor02"
    },
    {
        "name": "ls",
        "status": "Up",
        "metaDir": "/mnt/stor02"
    },
    {
        "name": "cs",
        "status": "Up",
        "metaDir": "/mnt/stor02"
    }
]
}
}
```

## 4.7.8 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	描述	是否必须
serverId	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。 类型：字符串	否

### ● 响应结果

名称	描述
serverName	服务器名称。
serverId	服务器 ID。
status	服务器状态： <ul style="list-style-type: none"> <li>● Connected: 已连接。</li> <li>● Disconnected: 未连接。</li> <li>● Removing: 移除中。</li> </ul>
version	服务器上的 HBlock 版本。
isMasterServer	该服务器是否为 Master 节点（仅集群版支持）： <ul style="list-style-type: none"> <li>● true: 是。</li> <li>● false: 否。</li> </ul>

isBaseServer	该服务器是否为基础节点（仅集群版支持）： <ul style="list-style-type: none"> <li>● true: 是。</li> <li>● false: 否。</li> </ul>
publicAddress.ip	业务网的 IP。
publicAddress.port	业务网端口号。
clusterAddress.ip	集群网 IP 地址。
clusterAddress.port	集群网端口号。
targetPortalIP.ips	iSCSI 目标门户 IP 属性集合，包含：ip、port。 类型：数组
ip	iSCSI 目标门户 IP。
port	iSCSI 目标门户端口号。
targetPortalIP.status	iSCSI 目标门户 IP 状态： <ul style="list-style-type: none"> <li>● Enabled: 启用状态。</li> <li>● Disabled: 停用状态。</li> </ul>
diskPaths	HBlock 数据目录信息集合，包含：path、totalCapacity、usedCapacity、usedCapacityQuota、capacityQuota、managementStatus、healthStatus、healthDetail。 类型：数组
path	HBlock 数据目录。
totalCapacity	当前 HBlock 数据目录对应分区的总容量，单位是字节。
usedCapacity	服务器上 HBlock 数据目录对应分区的已用容量，单位是字节。
usedCapacityQuota	HBlock 在该数据目录的已写入量，即已用容量配额。单位是字节。

capacityQuota	数据目录的容量配额。单位是字节。 0 表示禁止写入。不输出，表示没有限制。
managementStatus	数据目录管理状态： <ul style="list-style-type: none"> <li>● Added: 数据目录已添加至系统中。</li> <li>● Removing: 数据目录正在移除中。</li> </ul>
subStatus.stage	移除数据目录所处的阶段： <ul style="list-style-type: none"> <li>● Reconfiguration: 重置中。</li> <li>● 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）。</li> <li>● Executing: 执行移除数据目录</li> </ul>
subStatus.totalFD	总的故障域个数。
subStatus.healthyFD	状态为 Healthy 的故障域个数。
subStatus.warningFD	状态为 Warning 的故障域个数。
subStatus.errorFD	状态为 Error 的故障域个数。
subStatus.safeData	安全数据百分比，单位是%。
subStatus.reconstructionWaitData	需要重建的数据百分比，单位是%。
subStatus.availFDWaitData	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	单副本数据百分比，单位是%。
subStatus.corruptedData	已经损坏的数据百分比，单位是%。



subStatus.availFDWaitWarning	告警信息。
subStatus.singleCopyLUNWarning	根据状态不同，显示的告警信息不同：
subStatus.corruptedWarning	<ul style="list-style-type: none"> <li>● <b>availFDWaitWarning</b>: 移除磁盘或者服务器时，数据重建需要的故障域个数不足。</li> <li>● <b>singleCopyLUNWarning</b>: 单副本卷数据所在磁盘或者服务器被移除。</li> <li>● <b>corruptedWarning</b>: 故障域异常导致数据损坏。</li> </ul>
healthStatus	数据目录健康状态： <ul style="list-style-type: none"> <li>● <b>Healthy</b>: 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 90%。</li> <li>● <b>Warning</b>: 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 90%；磁盘剩余空间不足 1GiB；或者 HBlock 对这个目录停写。</li> <li>● <b>Error</b>: 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。</li> </ul>
healthDetail	数据目录健康状态详情： <ul style="list-style-type: none"> <li>● 如果健康状态为 <b>Healthy</b>，此字段为空。</li> <li>● 如果健康状态为 <b>Warning</b> 或 <b>Error</b>，显示警告或错误的详细信息。</li> </ul>
defaultPath	默认的数据目录（仅单机版支持）。
recentStartTime	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	HBlock 端口的集合。数组，包括 portName、port。
portName	HBlock 端口名称。
port	HBlock 在当前服务器上使用的端口号。
portRange	端口范围。
iSCSIPort	iSCSI 端口号。
storagePorts	存储服务端口集合（仅集群版支持）。 数组，包括 portN、diskpath、description、pathId。
portN	存储服务端口。
diskpath	存储服务的数据目录。
pathId	数据目录 ID。
nodeName	节点名称（仅集群版支持）。
parentName	父节点名称（仅集群版支持）。
description	节点描述（仅集群版支持）。
baseServices	基础服务属性的集合（仅集群版基础服务器支持）。数组，包括 name、status、metaDir。
name	基础服务名称： <ul style="list-style-type: none"> <li>● mdm: 元数据管理服务。</li> <li>● ls: 日志服务。</li> <li>● cs: 协调服务。</li> </ul>
status	基础服务的状态： <ul style="list-style-type: none"> <li>● Up: 可用。</li> <li>● Down: 不可用。</li> <li>● Migrating: 正在迁移。</li> <li>● Unkown: 未知。</li> </ul>
metaDir	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: Tue, 14 Jan 2025 08:45:46 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.32:1443
```

- 响应示例 1

```
HTTP/1.1 200 OK
x-hblock-request-id: 1bc58ab26322494fb2d2e6a5837f6193
Connection: keep-alive
Content-Length: 794
Date: Tue, 14 Jan 2025 08:45:46 GMT
Content-Type: application/json; charset=utf-8
Server: HBlock

{
  "data": {
    "serverName": "ecs-9689-0915127",
    "serverId": "hblock_1",
    "status": "Connected",
    "version": "3.8.0",
    "publicAddress": {
      "ip": "192.168.0.32",
      "port": 3260
    },
    "clusterAddress": {
      "ip": "192.168.0.32",
      "port": 20004
    },
    "diskPaths": [
      {
        "path": "/mnt/stor",
```

```
        "usedCapacity": 35861303296,
        "totalCapacity": 100169342976,
        "usedCapacityQuota": 356576,
        "managementStatus": "Added",
        "healthStatus": "Healthy",
        "healthDetail": []
    }
],
"defaultPath": "/mnt/stor",
"recentStartTime": 1735873608983,
"ports": [
    {
        "portName": "apiPort",
        "port": 1443
    },
    {
        "portName": "webPort",
        "port": 2443
    },
    {
        "portName": "managementPort1",
        "port": 20004
    },
    {
        "portName": "managementPort2",
        "port": 20000
    },
    {
        "portName": "managementPort3",
        "port": 20002
    },
    {
        "portName": "managementPort4",
        "port": 20003
    },
],
```

```
{
  "portName": "managementPort6",
  "port": 20001
},
"portRange": "20000-20500",
"iSCSIPort": 3260
}
```

### ● 请求示例 2

集群版：查询服务器 ID 为 hblock\_2 的信息。

```
GET /rest/v1/system/server/hblock_2 HTTP/1.1
Date: Mon, 25 Nov 2024 09:29:43 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.117:1443
```

### ● 响应示例 2

```
HTTP/1.1 200 OK
x-hblock-request-id: 23faa39b4637441194e988a0b40cf266
Connection: keep-alive
Content-Length: 1414
Date: Mon, 25 Nov 2024 09:29:43 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "serverName": "ecs-9689-0915142",
    "serverId": "hblock_2",
    "status": "Connected",
    "version": "3.8.0",
    "isMasterServer": false,
    "isBaseServer": true,
```

```
"publicAddress": {
  "ip": "192.168.0.117",
  "port": 3260
},
"clusterAddress": {
  "ip": "192.168.0.117",
  "port": 20008
},
"diskPaths": [
  {
    "path": "/mnt/storage02",
    "storagePool": "default",
    "usedCapacity": 18030116864,
    "totalCapacity": 35963482112,
    "usedCapacityQuota": 69632,
    "managementStatus": "Added",
    "healthStatus": "Healthy",
    "healthDetail": []
  }
],
"recentStartTime": 1732519209900,
"ports": [
  {
    "portName": "apiPort",
    "port": 1443
  },
  {
    "portName": "webPort",
    "port": 2443
  },
  {
    "portName": "managementPort1",
    "port": 20008
  },
  {
```

```
    "portName": "managementPort2",
    "port": 20013
  },
  {
    "portName": "managementPort3",
    "port": 20004
  },
  {
    "portName": "managementPort4",
    "port": 20011
  },
  {
    "portName": "managementPort5",
    "port": 20000
  },
  {
    "portName": "managementPort6",
    "port": 20010
  },
  {
    "portName": "dataPort1",
    "port": 20002
  },
  {
    "portName": "metadataPort4",
    "port": 20005
  },
  {
    "portName": "metadataPort5",
    "port": 20001
  },
  {
    "portName": "metadataPort6",
    "port": 20006
  },
}
```

```
{
  "portName": "metadataPort7",
  "port": 20012
},
{
  "portName": "metadataPort8",
  "port": 20007
}
],
"portRange": "20000-20500",
"iSCSIPort": 3260,
"storagePorts": [
  {
    "port1": 20003,
    "port2": 20014,
    "port3": 20009,
    "diskpath": "/mnt/storage02",
    "pathId": "1"
  }
],
"nodeName": "default:hblock_2",
"parentName": "default",
"baseServices": [
  {
    "name": "ls",
    "status": "Up",
    "metaDir": "/mnt/storage02"
  },
  {
    "name": "cs",
    "status": "Up",
    "metaDir": "/mnt/storage02"
  }
]
}
```



```
}
```

## 4.7.9 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	描述	是否必须
serverId	要添加的数据目录所属服务器 ID。 类型：字符串	是

diskPaths	数据目录属性集合。包括 path、capacityQuota。 类型：数组	是
path	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。 <b>注意：</b> 数据目录中不能有逗号（,）。 类型：字符串列表	是
capacityQuota	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。 类型：长整型 取值：小于数据目录所在磁盘的总容量，单位是字节。负整数表示无限制写入，0 表示禁止写入。默认无限制写入。 <b>注意：</b> 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\_2 的数据目录。

```

POST /rest/v1/system/server/hblock_2/diskpaths HTTP/1.1
Date: Wed, 27 Sep 2023 05:51:20 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 137
Host: 192.168.0.117:1443

{
  "diskPaths": [
    { "path": "/mnt/storage01",
      "capacityQuota": 1024000000
    },
  ],
}
    
```

```
{ "path": "/mnt/storage02"}  
]  
}
```

- 响应示例

```
HTTP/1.1 200 OK  
x-hblock-request-id: f0ec3fbbf3154cc99a671c3be437852a  
Connection: keep-alive  
Date: Wed, 27 Sep 2023 05:51:20 GMT  
Content-Length: 0  
Server: HBlock
```

## 4.7.10 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	描述	是否必须
serverId	要修改的数据目录所属服务器 ID。 类型：字符串	是
diskPaths	数据目录属性集合。包括 path、capacityQuota。 类型：数组	是
path	指定要修改容量配额的数据目录。	是

	类型：字符串列表	
capacityQuota	<p>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。</p> <p>类型：长整型</p> <p>取值：小于数据目录所在磁盘的总容量，单位是字节。负整数表示无限制写入，0 表示禁止写入。默认不限制写入。</p> <p><b>注意：</b>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。</p>	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
```

### 4.7.11 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	描述	是否必须
serverId	要移除数据目录的服务器 ID。 类型：字符串	是
diskPaths	要移除的 HBlock 数据目录。一次可以移除多个数据目录。 类型：数组	是
force	是否强制移除数据目录。 <b>注意：</b> 强制移除数据目录，可能造成数据丢失。 类型：布尔 取值： <ul style="list-style-type: none"> <li>● true: 强制移除数据目录。</li> <li>● false: 不强制移除数据目录。</li> </ul> 默认值为 false。	否
deleteLocalData	是否删除数据目录上的数据。 类型：布尔 取值： <ul style="list-style-type: none"> <li>● true: 删除数据目录上的数据。</li> <li>● false: 不删除数据目录上的数据。</li> </ul> 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
```

### 4.7.12 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	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 一个数据目录。

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	<p>The disk paths do not exist:</p> <pre><i>serverIP:diskPaths</i> [<i>,diskPaths...</i>] [; <i>serverIP:diskPaths[,diskPaths. ..]</i>.</pre> <p>The available disk space must be greater than or equal to 1 GiB for each disk path:</p> <pre><i>serverIP:diskPaths</i> [<i>,diskPaths...</i>] [; <i>serverIP:diskPaths[,diskPaths. ..]</i>.</pre> <p>The disk paths can not access:</p> <pre><i>serverIP:diskPaths</i> [<i>,diskPaths...</i>] [; <i>serverIP:diskPaths[,diskPaths. ..]</i>.</pre> <p>The disk paths must be directory: <i>serverIP:diskPaths</i> [<i>,diskPaths...</i>] [; <i>serverIP:diskPaths[,diskPaths. ..]</i>.</p> <p>The disk paths include incompatible data:</p>	因数据目录有问题导致操作失败。

		<p><i>serverIP:diskPaths</i> [,<i>diskPaths...</i>] [;</p> <p><i>serverIP:diskPaths</i>[,<i>diskPaths</i>.. ..]].</p> <p>The disk paths can not contain commas:</p> <p><i>serverIP:diskPaths</i> [,<i>diskPaths...</i>] [;</p> <p><i>serverIP:diskPaths</i>[,<i>diskPaths</i>.. ..]].</p> <p>The disk paths is already on the server and the status is removing:</p> <p><i>serverIP:diskPaths</i> [,<i>diskPaths...</i>]</p>	
400	InvalidDiskPathCapacityQuota	<p>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.</p> <p><i>serverIP:diskPaths</i> [,<i>diskPaths...</i>] [;<i>serverIP:diskPaths</i>[,<i>diskPaths...</i>.. ..]].</p>	容量配额参数错误。
400	InvalidLong	<p>Value <i>value</i> at '<i>argument</i>' failed to satisfy constraint:</p>	参数填写错误，取值必须为长整型。

		Argument must be of type long.	
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 data loss. You can use force remove. When using this option, there may be a risk of data loss.	数据目录状态不正常，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。

400	RemovePathRisk	<p>There is LUN with redundancy mode, which has the minimum requirement for the number of paths.</p> <p>LUN: <i>lunName1</i>[, <i>lunName2</i>....]</p> <p>Removing the path may cause data loss. You can use force remove. When using this option, there may be a risk of data loss.</p>	<p>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。</p> <p>数据目录移除后会造成卷数据损毁：<i>lunName1</i>[, <i>lunName2</i>....]。</p>
400	RemoveServerRisk	<p>There is LUN with high availability disabled, and the LUN is connected with the client.</p> <p>LUN: <i>lunName1</i>[, <i>lunName2</i>....]</p> <p>There is LUN with redundancy mode, which has the minimum requirement for the number of servers.</p> <p>LUN: <i>lunName1</i>[, <i>lunName2</i>....]</p> <p>There is Target IQN which has no available server to migrate, removing server will cause the IQN deleted.</p>	<p>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。</p> <p>移除服务器会导致卷冗余度失败。</p> <p>移除服务器之后，该服务器上面部分的IQN无法在其他机器上重建。</p> <p>移除服务器时，有可能丢数据。如果必须</p>



		<p>Target: TargetIQN1_[, TargetIQN2....]_</p> <p>Removing the server may cause data loss. You can use force remove. When using this option, there may be a risk of data loss.</p>	移除，请使用强制移除。
400	RestartServerRisk	<p>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.</p>	重启服务器失败。可以强制重启，但有丢失数据风险。
400	StandaloneModeNotAllowed	<p>'<i>operation</i>' is not supported by standalone mode of HBlock.</p>	单机版本 HBlock 不支持该操作。
400	SystemCrashRisk	<p>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.</p>	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	<p>The path <i>path</i> does not exist in the HBlock.</p>	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	TargetPortalIP 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 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. <i>server_ID/IP/local server: port_name port[, port_name port...]</i> .	端口冲突。
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	有服务器正在移除时，不能再移除其他服务器。如果必须移

		option, there may be a risk of data loss.	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。

## 4.8 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	描述
storId	HBlock ID。
storName	HBlock 的名称。
serialId	HBlock 序列号。
serverNumbers	HBlock 服务器数量。
basePoolName	基础存储池的名字（仅集群版）。
faultDomain	故障域类型（仅集群版）： <ul style="list-style-type: none"> <li>● rack: 机架级别的故障域。</li> <li>● room: 机房级别的故障域。</li> <li>● server: 服务器级别的故障域。</li> <li>● path: 数据目录级别的故障域。</li> </ul>
setupTime	系统初始化时间。unix 时间戳（UTC），精确到毫秒。
trialStatus	试用期标识： <ul style="list-style-type: none"> <li>● Trial: 处于试用期内，未导入软件许可证。</li> <li>● TrialExpired: 试用期已过，未导入软件许可证。</li> <li>● NonTrialVersion: 非试用版本，已导入软件许可证。</li> </ul>

trialExpireTime	试用期到期时间。unix 时间戳（UTC），精确到毫秒。
userName	HBlock 的管理员用户名。

- 请求示例 1

单机版：查询 HBlock 信息。

```
GET /rest/v1/system/info HTTP/1.1
Date: Tue, 6 Aug 2024 08:41:41 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.32:1443
```

- 响应示例 1

```
HTTP/1.1 200 OK
x-hblock-request-id: 62d8830817844ebba996c0b195bff655
Connection: keep-alive
Content-Length: 269
Date: Tue, 6 Aug 2024 08:41:41 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "storId": "D28969D0-376B-4BCC-B976-200E500802DB",
    "storName": "stor2",
    "serialId": "D28969D0-376B-4BCC-B976-200E500802DB-0201-030700",
    "serverNumbers": 1,
    "setupTime": 1722587327264,
    "trialStatus": "NonTrialVersion",
    "trialExpireTime": 1725179327264,
    "userName": "storuser"
  }
}
```

- 请求示例 2

集群版：查询 HBlock 信息。

```
GET /rest/v1/system/info HTTP/1.1
Date: Tue, 6 Aug 2024 08:39:16 GMT
Authorization: HBlock userName:signature
Host:192.168.0.110:1443
```

### ● 响应示例 2

```
HTTP/1.1 200 OK
x-hblock-request-id: 1b0d2c47fbb04475940bc7756f3ef36d
Connection: keep-alive
Content-Length: 306
Date: Tue, 6 Aug 2024 08:39:16 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "storId": "6DCDE5A5-B0EE-496B-9C1A-6013CDBD844B",
    "storName": "stor",
    "serialId": "6DCDE5A5-B0EE-496B-9C1A-6013CDBD844B-0202-030700",
    "serverNumbers": 4,
    "basePoolName": "default",
    "faultDomain": "server",
    "setupTime": 1722494887405,
    "trialStatus": "Trial",
    "trialExpireTime": 1725086887405,
    "userName": "storuser"
  }
}
```

## 4.9 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/v1/system/status HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

### ● 响应结果

名称	描述
system.status	HBlock 服务状态： <ul style="list-style-type: none"> <li>● <b>Upgrading</b>: 升级中。</li> <li>● <b>Uninstalling</b>: 卸载中。</li> <li>● <b>Working</b>: 运行中。</li> </ul>
system.licenseStatus	软件许可证状态： <ul style="list-style-type: none"> <li>● <b>Expired</b>: 软件许可证已过期。</li> <li>● <b>Effective</b>: 软件许可证已生效。</li> <li>● <b>None</b>: 还未加载软件许可证。</li> </ul>
server.status.connected	状态为连接的服务器个数。
server.status.disconnected	状态为断开连接的服务器个数。
server.status.removing	状态为正在移除的服务器个数。
disk.status.healthy	状态为健康的磁盘个数（仅单机版支持）。
disk.status.warning	状态为告警的磁盘个数（仅单机版支持）。
disk.status.error	状态为错误的磁盘个数（仅单机版支持）。

disk.usage.used	HBlock 数据目录对应分区的已用容量（仅单机版支持），单位是字节。
disk.usage.total	HBlock 数据目录对应分区的总容量（仅单机版支持），单位是字节。
lun.status.number	卷的总个数。
lun.status.cacheNumber	缓存模式卷的个数。
lun.status.storageNumber	存储模式卷的个数。
lun.status.localNumber	本地模式卷的个数。
lun.status.cacheCapacity	缓存模式卷的容量，单位是字节。
lun.status.storageCapacity	存储模式卷的容量，单位是字节。
lun.status.localCapacity	本地模式卷的容量，单位是字节。
lun.data.normal	正常状态的卷数量百分比。
lun.data.lowRedundancy	低冗余状态的卷数据量百分比。
lun.data.error	错误状态的卷数据量百分比。
storagePool.number	存储池个数（仅集群版支持）。
storagePool.basePool	基础存储池名称（仅集群版支持）。
storagePool.basePoolDetail.faultDomain.level	基础存储池的故障域级别（仅集群版支持）。
storagePool.basePoolDetail.faultDomain.status.healthy	状态为健康的故障域个数（仅集群版支持）。
storagePool.basePoolDetail.faultDomain.status.warning	状态为告警的故障域个数（仅集群版支持）。
storagePool.basePoolDetail.faultDomain.status.error	状态为错误的故障域个数（仅集群版支持）。
storagePool.basePoolDetail.disk.status.healthy	状态为健康的磁盘个数（仅集群版支持）。



storagePool.basePoolDetail. disk.status.warning	状态为告警的磁盘个数（仅集群版支持）。
storagePool.basePoolDetail. disk.status.error	状态为错误的磁盘个数（仅集群版支持）。
storagePool.basePoolDetail. disk.usage.used	基础存储池中 HBlock 数据目录对应分区的已用容量（仅集群版支持），单位是字节。
storagePool.basePoolDetail. disk.usage.total	基础存储池中 HBlock 数据目录对应分区的总容量（仅集群版支持），单位是字节。

### ● 请求示例 1

单机版：查询 HBlock 状态。

```
GET /rest/v1/system/status HTTP/1.1
Date: Mon, 13 May 2024 11:06:55 GMT
Authorization: HBlock userName:signature
Host:192.168.0.32:1443
```

### ● 响应示例 1

```
HTTP/1.1 200 OK
x-hblock-request-id: b9a47583b571487697f327cfd70723fe
Connection: keep-alive
Content-Length: 443
Date: Mon, 13 May 2024 11:06:55 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "system": {
      "status": "Working",
      "licenseStatus": "Effective"
    },
  },
}
```

```
"server": {
  "status": {
    "connected": 1,
    "disconnected": 0,
    "removing": 0
  }
},
"disk": {
  "status": {
    "healthy": 1,
    "warning": 0,
    "error": 0
  },
  "usage": {
    "used": 63168512,
    "total": 100167352320
  }
},
"lun": {
  "status": {
    "number": 5,
    "cacheNumber": 0,
    "storageNumber": 2,
    "localNumber": 3,
    "cacheCapacity": 0,
    "storageCapacity": 654982512640,
    "localCapacity": 665719930880
  },
  "data": {
    "normal": 100,
    "lowRedundancy": 0,
    "error": 0
  }
}
}
```

```
}
```

### ● 请求示例 2

集群版：查询 HBlock 状态。

```
GET /rest/v1/system/status HTTP/1.1
Date: Wed, 7 Aug 2024 06:20:45 GMT
Authorization: HBlock userName:signature
Host:192.168.0.32:1443
```

### ● 响应示例 2

```
HTTP/1.1 200 OK
x-hblock-request-id: 34e45fb7348647a6afbfb1dcb5b602cf
Connection: keep-alive
Content-Length: 597
Date: Wed, 7 Aug 2024 06:20:45 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "system": {
      "status": "Working",
      "licenseStatus": "None"
    },
    "server": {
      "status": {
        "connected": 4,
        "disconnected": 0,
        "removing": 0
      }
    },
    "storagePool": {
      "number": 4,
      "basePool": "default",
```

```
"basePoolDetail": {
  "faultDomain": {
    "level": "server",
    "status": {
      "healthy": 3,
      "warning": 0,
      "error": 0
    }
  },
  "disk": {
    "status": {
      "healthy": 4,
      "warning": 0,
      "error": 0
    },
    "usage": {
      "used": 40795934720,
      "total": 336471511040
    }
  }
},
"lun": {
  "status": {
    "number": 5,
    "cacheNumber": 1,
    "storageNumber": 1,
    "localNumber": 3,
    "cacheCapacity": 214748364800,
    "storageCapacity": 322122547200,
    "localCapacity": 429496729600
  },
  "data": {
    "normal": 100,
    "lowRedundancy": 0,
```

```
    "error": 0  
  }  
}  
}
```

## 4.10 监控

### 4.10.1 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	描述	是否必须
dimension	监控对象。 类型：枚举 取值： <ul style="list-style-type: none"> <li>● system: 系统。</li> <li>● pool: 存储池（仅集群版支持）。</li> <li>● server: 服务器。</li> <li>● disk: 数据目录所在磁盘分区。</li> <li>● LUN: 卷。</li> </ul> 单机版默认值为 <b>system</b> ，集群版默认值为 <b>pool</b> 。	否
instanceId	监控对象实例的唯一标识，可以填写多个，以英文逗号隔开。 类型：字符串 取值： <ul style="list-style-type: none"> <li>● 监控对象为 <b>system</b>，没有实例。</li> </ul>	否

	<ul style="list-style-type: none"> <li>● 监控对象为 <code>pool</code>，实例为存储池名称。</li> <li>● 监控对象为 <code>server</code>，实例取值为服务器 ID。</li> <li>● 监控对象为 <code>disk</code>，实例取值为“<code>serverId+路径名称</code>”，格式为 <code>serverId:/diskpath</code>。</li> <li>● 监控对象为 <code>LUN</code>，实例取值为卷名称。</li> </ul> <p>如果不填写，默认查看监控对象下的所有实例性能数据。</p>	
<p><code>metric</code></p>	<p>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<a href="#">监控指标</a>。</p> <p>类型：字符串</p> <p>取值：</p> <ul style="list-style-type: none"> <li>● 监控对象为 <code>system</code>:                             <ul style="list-style-type: none"> <li>■ IOPS</li> <li>■ R_IOPS</li> <li>■ W_IOPS</li> <li>■ Bandwidth</li> <li>■ R_Bandwidth</li> <li>■ W_Bandwidth</li> <li>■ Latency</li> <li>■ W_Latency</li> <li>■ R_Latency</li> <li>■ Path_Cap</li> <li>■ Path_Used</li> <li>■ Path_Rate</li> <li>■ Path_Cap_Quota</li> <li>■ Path_Cap_Quota_Used</li> <li>■ Path_Cap_Quota_Rate</li> <li>■ Cloud_Bandwidth</li> </ul> </li> </ul>	<p>否</p>

	<ul style="list-style-type: none"><li>■ Cloud_U_Bandwidth</li><li>■ Cloud_D_Bandwidth</li><li>● 监控对象为 pool:<ul style="list-style-type: none"><li>■ IOPS</li><li>■ R_IOPS</li><li>■ W_IOPS</li><li>■ Bandwidth</li><li>■ R_Bandwidth</li><li>■ W_Bandwidth</li><li>■ Latency</li><li>■ W_Latency</li><li>■ R_Latency</li><li>■ Path_Cap</li><li>■ Path_Used</li><li>■ Path_Rate</li><li>■ Path_Cap_Quota</li><li>■ Path_Cap_Quota_Used</li><li>■ Path_Cap_Quota_Rate</li></ul></li><li>● 监控对象为 server:<ul style="list-style-type: none"><li>■ CPU_Rate</li><li>■ Mem_Rate</li><li>■ Mem_Total</li><li>■ Mem_Used</li><li>■ IOPS</li><li>■ R_IOPS</li><li>■ W_IOPS</li><li>■ Bandwidth</li></ul></li></ul>	
--	--	--



	<ul style="list-style-type: none"><li>■ R_Bandwidth</li><li>■ W_Bandwidth</li><li>■ Latency</li><li>■ W_Latency</li><li>■ R_Latency</li><li>■ Path_Cap</li><li>■ Path_Used</li><li>■ Path_Rate</li><li>■ Path_Cap_Quota</li><li>■ Path_Cap_Quota_Used</li><li>■ Path_Cap_Quota_Rate</li><li>■ Cloud_Bandwidth</li><li>■ Cloud_U_Bandwidth</li><li>■ Cloud_D_Bandwidth</li><li>● 监控对象为 disk:<ul style="list-style-type: none"><li>■ Path_Cap</li><li>■ Path_Used</li><li>■ Path_Rate</li><li>■ Path_Cap_Quota</li><li>■ Path_Cap_Quota_Used</li><li>■ Path_Cap_Quota_Rate</li></ul></li><li>● 监控对象为 LUN:<ul style="list-style-type: none"><li>■ IOPS</li><li>■ R_IOPS</li><li>■ W_IOPS</li><li>■ Bandwidth</li><li>■ R_Bandwidth</li></ul></li></ul>	
--	---	--

	<ul style="list-style-type: none"> <li>■ W_Bandwidth</li> <li>■ Latency</li> <li>■ W_Latency</li> <li>■ R_Latency</li> <li>■ Cloud_Bandwidth</li> <li>■ Cloud_U_Bandwidth</li> <li>■ Cloud_D_Bandwidth</li> <li>■ Wait_Upload</li> </ul> <p>不填写，默认查询监控对象的全部指标。</p>	
--	--	--

● 响应结果

名称	描述
monitoring	监控信息集合，包含： <code>dimension</code> 、 <code>instanceId</code> 、 <code>metric</code> 、 <code>avgValueSeries</code> 。 类型：数组
dimension	监控对象： <ul style="list-style-type: none"> <li>● <code>system</code>：系统。</li> <li>● <code>pool</code>：存储池（仅集群版支持）。</li> <li>● <code>server</code>：服务器。</li> <li>● <code>disk</code>：数据目录所在磁盘分区。</li> <li>● <code>LUN</code>：卷。</li> </ul>
instanceId	监控对象实例： <ul style="list-style-type: none"> <li>● 监控对象为 <code>system</code>，显示 HBlock 名称。</li> <li>● 监控对象为 <code>pool</code>，显示存储池名称（仅集群版支持）。</li> <li>● 监控对象为 <code>server</code>，显示为服务器 ID。</li> <li>● 监控对象为 <code>disk</code>，显示为数据目录。</li> <li>● 监控对象为 <code>LUN</code>，显示为卷名称。</li> </ul>

metric	指标名称。具体详见 <a href="#">监控指标</a> 。
avgValueSeries	监控数据的[时间,数值]序列。 时间为 <b>unix</b> 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,
      97.78
    ]
  ]
}
```

```
    }  
  ]  
}  
}
```

## 4.10.2 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	描述	是否必须
startTime	查询起始时间。 类型：时间戳 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	查询结束时间。 类型：时间戳 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	监控对象。 类型：枚举 取值： <ul style="list-style-type: none"> <li>● system: 系统。</li> <li>● pool: 存储池（仅集群版支持）。</li> </ul>	否

	<ul style="list-style-type: none"> <li>● <b>server</b>: 服务器。</li> <li>● <b>disk</b>: 数据目录所在磁盘分区。</li> <li>● <b>LUN</b>: 卷。</li> </ul> <p>单机版默认值为 <b>system</b>，集群版默认值为 <b>pool</b>。</p>	
<b>instanceId</b>	<p>监控对象实例的唯一标识。</p> <p>类型：字符串</p> <p>取值：</p> <ul style="list-style-type: none"> <li>● 监控对象为 <b>system</b>，没有实例。</li> <li>● 监控对象为 <b>pool</b>，必填，实例为存储池名称（仅集群版支持）。</li> <li>● 监控对象为 <b>server</b>，必填，实例取值为服务器 ID。</li> <li>● 监控对象为 <b>disk</b>，必填，实例取值为“<b>serverId+路径名称</b>”，格式为 <b>serverId:/diskpath</b>。</li> <li>● 监控对象为 <b>LUN</b>，必填，实例取值为卷名称。</li> </ul>	否
<b>metric</b>	<p>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<a href="#">监控指标</a>。</p> <p>类型：字符串</p> <p>取值：</p> <ul style="list-style-type: none"> <li>● 监控对象为 <b>system</b>:             <ul style="list-style-type: none"> <li>■ IOPS</li> <li>■ R_IOPS</li> <li>■ W_IOPS</li> <li>■ Bandwidth</li> <li>■ R_Bandwidth</li> <li>■ W_Bandwidth</li> <li>■ Latency</li> <li>■ W_Latency</li> </ul> </li> </ul>	否

	<ul style="list-style-type: none"> <li>■ R_Latency</li> <li>■ Path_Cap</li> <li>■ Path_Used</li> <li>■ Path_Rate</li> <li>■ Path_Cap_Quota</li> <li>■ Path_Cap_Quota_Used</li> <li>■ Path_Cap_Quota_Rate</li> <li>■ Cloud_Bandwidth</li> <li>■ Cloud_U_Bandwidth</li> <li>■ Cloud_D_Bandwidth</li> <li>● 监控对象为 pool:                             <ul style="list-style-type: none"> <li>■ IOPS</li> <li>■ R_IOPS</li> <li>■ W_IOPS</li> <li>■ Bandwidth</li> <li>■ R_Bandwidth</li> <li>■ W_Bandwidth</li> <li>■ Latency</li> <li>■ W_Latency</li> <li>■ R_Latency</li> <li>■ Path_Cap</li> <li>■ Path_Used</li> <li>■ Path_Rate</li> <li>■ Path_Cap_Quota</li> <li>■ Path_Cap_Quota_Used</li> <li>■ Path_Cap_Quota_Rate</li> </ul> </li> <li>● 监控对象为 server:</li> </ul>	
--	--	--



	<ul style="list-style-type: none"><li>■ CPU_Rate</li><li>■ Mem_Rate</li><li>■ Mem_Total</li><li>■ Mem_Used</li><li>■ IOPS</li><li>■ R_IOPS</li><li>■ W_IOPS</li><li>■ Bandwidth</li><li>■ R_Bandwidth</li><li>■ W_Bandwidth</li><li>■ Latency</li><li>■ W_Latency</li><li>■ R_Latency</li><li>■ Path_Cap</li><li>■ Path_Used</li><li>■ Path_Rate</li><li>■ Path_Cap_Quota</li><li>■ Path_Cap_Quota_Used</li><li>■ Path_Cap_Quota_Rate</li><li>■ Cloud_Bandwidth</li><li>■ Cloud_U_Bandwidth</li><li>■ Cloud_D_Bandwidth</li><li>● 监控对象为 disk:<ul style="list-style-type: none"><li>■ Path_Cap</li><li>■ Path_Used</li><li>■ Path_Rate</li><li>■ Path_Cap_Quota</li></ul></li></ul>	
--	--	--

	<ul style="list-style-type: none"> <li>■ Path_Cap_Quota_Used</li> <li>■ Path_Cap_Quota_Rate</li> <li>● 监控对象为 LUN:                             <ul style="list-style-type: none"> <li>■ IOPS</li> <li>■ R_IOPS</li> <li>■ W_IOPS</li> <li>■ Bandwidth</li> <li>■ R_Bandwidth</li> <li>■ W_Bandwidth</li> <li>■ Latency</li> <li>■ W_Latency</li> <li>■ R_Latency</li> <li>■ Cloud_Bandwidth</li> <li>■ Cloud_U_Bandwidth</li> <li>■ Cloud_D_Bandwidth</li> <li>■ Wait_Upload</li> </ul> </li> </ul> <p>不填写，默认获取监控对象的全部指标。</p>	
--	--	--

● 响应结果

名称	描述
monitoring	监控信息集合，包含：dimension、instanceId、metric、avgValueSeries。 类型：数组
dimension	监控对象： <ul style="list-style-type: none"> <li>● system: 系统。</li> <li>● pool: 存储池（仅集群版支持）。</li> <li>● server: 服务器。</li> </ul>

	<ul style="list-style-type: none"> <li>● <b>disk</b>: 数据目录所在磁盘分区。</li> <li>● <b>LUN</b>: 卷。</li> </ul>
<b>instanceId</b>	监控对象实例: <ul style="list-style-type: none"> <li>● 监控对象为 <b>system</b>, 显示 HBlock 名称。</li> <li>● 监控对象为 <b>pool</b>, 显示存储池名称 (仅集群版支持)</li> <li>● 监控对象为 <b>server</b>, 显示为服务器 ID。</li> <li>● 监控对象为 <b>disk</b>, 显示为 “<b>serverId+路径名称</b>”, 格式为 <b>serverId:/path/dir</b>。</li> <li>● 监控对象为 <b>LUN</b>, 显示为卷名称。</li> </ul>
<b>metric</b>	指标名称。具体详见 <b>监控指标</b> 。
<b>avgValueSeries</b>	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  
    ]  
  ]  
}
```

### 4.10.3 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。



## 4.11 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。

### 4.11.1 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	描述	是否必须
alarmStatus	告警状态： <ul style="list-style-type: none"> <li>● Unresolved: 告警中。</li> <li>● Resolved: 告警已解除。</li> <li>● Expired: 告警已失效。</li> </ul> 默认值为 Unresolved。 <b>说明：</b> 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	设置查询的过滤条件。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： <ul style="list-style-type: none"> <li>● instanceSnapshot: 告警实例的详情。</li> <li>● alarmRule: 告警规则编码名称，详见告警列表。</li> </ul>	否

	<ul style="list-style-type: none"> <li>● severity: 告警级别:                             <ul style="list-style-type: none"> <li>■ Warning: 警告。</li> <li>■ Major: 重要。</li> <li>■ Critical: 严重。</li> </ul> </li> <li>● muteStatus: 静默状态:                             <ul style="list-style-type: none"> <li>■ Normal: 正常。</li> <li>■ Muted: 静默。</li> </ul> </li> <li>● alarmTime: 告警发生时间, 通过&gt;和&lt;支持范围查询。 unix 时间戳 (UTC), 精确到毫秒。</li> <li>● resolveTime: 解除时间, 通过&gt;和&lt;支持范围查询。unix 时间戳 (UTC), 精确到毫秒。</li> <li>● resolveType: 告警解除方式:                             <ul style="list-style-type: none"> <li>■ Manual: 手动解除警告。</li> <li>■ Auto: 自动解除告警。</li> </ul> </li> <li>● expireTime: 告警失效时间, 通过&gt;和&lt;支持范围查询。 unix 时间戳 (UTC), 精确到毫秒。</li> </ul>	
range	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	描述
totalCount	符合查询条件的告警条数。
alarmList	告警列表, 包含: alarmId、instanceId、instanceSnapshot、alarmRule、severity、alarmStatus、duration、alarmTime、muteStatus、alarmValue、currentValue、resolveTime、

	resolveValue、resolveType、expireTime、reason、muteDueTime、muteOperations。 类型：数组
alarmId	告警 ID。
instanceId	告警实例 ID。
instanceSnapshot	告警实例快照，即告警发生时告警实例的详细信息。
alarmRule	告警规则。
severity	告警级别： <ul style="list-style-type: none"> <li>● Warning: 警告。</li> <li>● Major: 重要。</li> <li>● Critical: 严重。</li> </ul>
alarmStatus	告警状态： <ul style="list-style-type: none"> <li>● Unresolved: 告警中。</li> <li>● Resolved: 告警已解除。</li> <li>● Expired: 告警已失效。</li> </ul>
duration	告警持续时长。unix 时间戳（UTC），精确到毫秒。
alarmTime	告警发生时间。unix 时间戳（UTC），精确到毫秒。
muteStatus	静默状态： <ul style="list-style-type: none"> <li>● Muted: 静默。</li> <li>● Normal: 正常。</li> </ul>
alarmValue	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： <ul style="list-style-type: none"> <li>● 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%。</li> </ul>

	<ul style="list-style-type: none"> <li>● <b>ResourceUsageApproachingLimit:</b> 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%。</li> <li>● <b>CapacityQuotaUsageExceedsThreshold:</b> 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%。</li> <li>● <b>CapacityQuotaUsageApproachLimit:</b> 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%。</li> <li>● <b>DiskUsageExceedsThreshold:</b> 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%。</li> <li>● <b>InsufficientSpaceonInstallationPath:</b> 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。</li> <li>● <b>InsufficientSpaceonMetaDir:</b> 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。</li> </ul>
<b>currentValue</b>	<p>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：</p> <ul style="list-style-type: none"> <li>● <b>AlarmNumberApproachingLimit:</b> 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%。</li> <li>● <b>ResourceUsageApproachingLimit:</b> 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%。</li> <li>● <b>CapacityQuotaUsageExceedsThreshold:</b> 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%。</li> </ul>

	<ul style="list-style-type: none"> <li>● <b>CapacityQuotaUsageApproachLimit:</b> 配额用尽, 显示存储池或数据目录关联磁盘的配额使用率 (<b>Path_Cap_Quota_Rate</b>), 百分数, 单位是%。</li> <li>● <b>DiskUsageExceedsThreshold:</b> 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 (<b>Path_Rate</b>), 百分数, 单位是%。</li> <li>● <b>InsufficientSpaceonInstallationPath:</b> 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 <b>GiB</b>。</li> <li>● <b>InsufficientSpaceonMetaDir:</b> 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 <b>GiB</b>。</li> </ul>
<b>resolveTime</b>	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。
<b>resolveValue</b>	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: <ul style="list-style-type: none"> <li>● <b>AlarmNumberApproachingLimit:</b> 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%。</li> <li>● <b>ResourceUsageApproachingLimit:</b> 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%。</li> <li>● <b>CapacityQuotaUsageExceedsThreshold:</b> 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 (<b>Path_Cap_Quota_Rate</b>), 百分数, 单位是%。</li> <li>● <b>CapacityQuotaUsageApproachLimit:</b> 配额用尽, 显示存储池或数据目录关联磁盘的配额使用率 (<b>Path_Cap_Quota_Rate</b>), 百分数, 单位是%。</li> </ul>

	<ul style="list-style-type: none"> <li>● <b>DiskUsageExceedsThreshold</b>: 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%。</li> <li>● <b>InsufficientSpaceonInstallationPath</b>: 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。</li> <li>● <b>InsufficientSpaceonMetaDir</b>: 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。</li> </ul>
resolveType	告警解除方式： <ul style="list-style-type: none"> <li>● <b>Auto</b>: 自动解除。</li> <li>● <b>Manual</b>: 手动解除。</li> </ul>
expireTime	告警失效时间。 Expired 状态的告警会返回此字段。
reason	告警解除原因或告警失效原因。
muteDueTime	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	静默操作，包含：operTime、operType、reason、dueTime。 类型：数组
operTime	静默操作时间。
operType	静默操作类型： <ul style="list-style-type: none"> <li>● <b>Mute</b>: 静默。</li> <li>● <b>AutoUnmute</b>: 自动解除静默。</li> <li>● <b>ManualUnmute</b>: 手动解除静默。</li> </ul>
reason	静默/解除静默的原因。
dueTime	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": "4jlhtUe1",
```



```
        "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": "4j1htUel",
        "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"
      }
    ]
  }
}
```

## 4.11.2 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	描述	是否必须
alarmId	告警 ID。 仅 Unresolved 状态的告警支持单条查询。	是

### ● 响应结果

名称	描述
alarmId	告警 ID。
instanceId	告警实例 ID。
instanceSnapshot	告警实例快照，即告警发生时告警实例的详细信息。
alarmRule	告警规则。
severity	告警级别： <ul style="list-style-type: none"> <li>● Warning: 警告。</li> <li>● Major: 重要。</li> <li>● Critical: 严重。</li> </ul>
alarmStatus	告警状态： <ul style="list-style-type: none"> <li>● Unresolved: 告警中。</li> <li>● Resolved: 告警已解除。</li> </ul>

	<ul style="list-style-type: none"> <li>● <b>Expired:</b> 告警已失效。</li> </ul>
duration	告警持续时长。unix 时间戳 (UTC)，精确到毫秒。
alarmTime	告警发生时间。unix 时间戳 (UTC)，精确到毫秒。
alarmValue	<p>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：</p> <ul style="list-style-type: none"> <li>● <b>AlarmNumberApproachingLimit:</b> 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：<math>(\text{告警中的告警总条数} / \text{告警中条数上限}) * 100\%</math>。</li> <li>● <b>ResourceUsageApproachingLimit:</b> 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：<math>(\text{本地卷总容量} / \text{许可证允许的容量}) * 100\%</math>。</li> <li>● <b>CapacityQuotaUsageExceedsThreshold:</b> 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 (<math>\text{Path\_Cap\_Quota\_Rate}</math>)，百分数，单位是%。</li> <li>● <b>CapacityQuotaUsageApproachLimit:</b> 配额用尽，显示存储池或数据目录关联磁盘的配额使用率 (<math>\text{Path\_Cap\_Quota\_Rate}</math>)，百分数，单位是%。</li> <li>● <b>DiskUsageExceedsThreshold:</b> 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 (<math>\text{Path\_Rate}</math>)，百分数，单位是%。</li> <li>● <b>InsufficientSpaceonInstallationPath:</b> 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。</li> <li>● <b>InsufficientSpaceonMetaDir:</b> 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。</li> </ul>
currentValue	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：



	<ul style="list-style-type: none"> <li>● <b>AlarmNumberApproachingLimit</b>: 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：<math>(\text{告警中的告警总条数}/\text{告警中条数上限}) * 100\%</math>。</li> <li>● <b>ResourceUsageApproachingLimit</b>: 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：<math>(\text{本地卷总容量}/\text{许可证允许的容量}) * 100\%</math>。</li> <li>● <b>CapacityQuotaUsageExceedsThreshold</b>: 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 <math>(\text{Path\_Cap\_Quota\_Rate})</math>，百分数，单位是%。</li> <li>● <b>CapacityQuotaUsageApproachLimit</b>: 配额用尽，显示存储池或数据目录关联磁盘的配额使用率 <math>(\text{Path\_Cap\_Quota\_Rate})</math>，百分数，单位是%。</li> <li>● <b>DiskUsageExceedsThreshold</b>: 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 <math>(\text{Path\_Rate})</math>，百分数，单位是%。</li> <li>● <b>InsufficientSpaceonInstallationPath</b>: 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。</li> <li>● <b>InsufficientSpaceonMetaDir</b>: 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。</li> </ul>
muteStatus	静默状态： <ul style="list-style-type: none"> <li>● <b>Muted</b>: 静默。</li> <li>● <b>Normal</b>: 正常。</li> </ul>
muteDueTime	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	静默操作，包含：operTime、operType、reason、dueTime。 类型：数组
operTime	静默操作时间。

operType	静默操作类型： <ul style="list-style-type: none"> <li>● Mute: 静默。</li> <li>● AutoUnmute: 自动解除静默。</li> <li>● ManualUnmute: 手动解除静默。</li> </ul>
reason	静默/解除静默的原因。
dueTime	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"  
  }  
}
```

### 4.11.3 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	描述	是否必须
alarmId	告警 ID。	是
reason	解除告警的原因。 类型：字符串 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
```

#### 4.11.4 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	描述	是否必须
alarmId	告警 ID。	是
operType	静默操作。 类型：字符串 取值： <ul style="list-style-type: none"> <li>● Mute: 静默告警。</li> <li>● ManualUnmute: 解除告警静默。</li> </ul>	是
dueTime	静默截止时间。 类型：时间戳 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	静默/解除静默的原因。 类型：字符串 取值：1~50 位字符串。	是
--------	--	---

### ● 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
```



## 4.11.5 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	描述	是否必须
alarmStatus	告警状态。 类型：字符串 取值： <ul style="list-style-type: none"> <li>● Unresolved：告警中。</li> <li>● Resolved：告警已解除。</li> <li>● Expired：告警已失效。</li> </ul> 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,
```

### 4.11.6 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 ' <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 '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 ' <i>argument</i> ' 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。

## 4.12 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。

### 4.12.1 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	描述	是否必须
number	指定最近事件的查询个数。 类型：整型 取值：[1, 1000]，默认值为 1000。	否
type	指定事件的类型。 类型：字符串 取值： <ul style="list-style-type: none"> <li>● user：用户事件。</li> <li>● system：系统事件。</li> </ul> 默认值为 user。	否

#### ● 响应结果

名称	描述
----	----

events	事件信息集合，包含：eventId、module、eventTime、name、requestId、requesterIP、statusCode、errorCode、errorMessage、instanceId。 类型：数组
eventId	事件 ID。
module	事件所属模块。
eventTime	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	事件名称。 <b>说明：</b> 事件名称描述详见用户事件列表和系统事件列表。
requestId	用户事件请求 ID。
requesterIP	用户事件中，发起请求的源 IP 地址。
statusCode	用户事件响应状态码。
errorCode	用户事件错误码。 -：表示无错误码。
errorMessage	用户事件错误信息。 -：表示无错误信息。
detail.method	用户事件的请求方法。
detail.url	用户事件的 URL。
detail.body	事件的请求体。
instanceId	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\", \"hb1
            ock_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\": \"0%\"}"
        }
      },
      {
        "eventId": "6b6d9ec4-4a20-44d2-b00c-f3f154373a11",
        "module": "Disk",
        "eventTime": 1659343495978,
        "name": "PathRemoved",
        "instanceId": "hblock_2:/mnt/storage02"
      }
    ]
  }
}
```



## 4.12.2 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	描述	是否必须
number	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。 类型：整型 取值：[1, 10000]。	否
type	指定事件的类型。 类型：字符串 取值： <ul style="list-style-type: none"> <li>● user：用户事件。</li> <li>● system：系统事件。</li> </ul> 默认值为 user。	否

### ● 响应头

名称	描述
Content-Disposition	响应内容的显示形式。
Transfer-Encoding	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,{}
```

### 4.12.3 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	描述	是否必须
startTime	HBlock 日志采集的起始时间。 类型：时间戳 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： <ul style="list-style-type: none"> <li>● HBlock 日志采集的起始时间必须早于 HBlocks 日志采集的结束时间。</li> </ul>	否

	<ul style="list-style-type: none"> <li>● 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。</li> <li>● 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。</li> </ul>	
endTime	<p>HBlock 日志采集的结束时间。</p> <p>类型：时间戳</p> <p>取值：unix 时间戳（UTC），精确到毫秒。默认值为当前时间。</p> <p>注意：</p> <ul style="list-style-type: none"> <li>● HBlock 日志采集的起始时间必须早于 HBlocks 日志采集的结束时间。</li> <li>● HBlock 日志采集的结束时间必须晚于 HBlock 初始化时间。</li> <li>● 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。</li> </ul>	否
servers	<p>要采集 HBlock 日志的服务器 ID。</p> <p>类型：数组</p> <p>取值：可以填写多个服务器 ID，以英文逗号（,）隔开。</p> <p>默认采集所有服务器。</p>	否
logTypes	<p>采集的日志类型</p> <p>类型：数组</p> <p>取值：</p> <ul style="list-style-type: none"> <li>● Config: 配置相关的日志。</li> <li>● System: 系统相关的日志。</li> <li>● Data: 数据处理相关的日志（仅集群版支持）。</li> </ul>	否

	<ul style="list-style-type: none"> <li>● <b>Coordination:</b> 内部协调服务相关的日志（仅集群版支持）。</li> </ul> 默认采集所有类型的日志。	
outputDirectory	HBlock 日志采集后存放的目录，为绝对路径。 类型：字符串 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： <ul style="list-style-type: none"> <li>● <b>id:</b> 本次日志请求的唯一标识符。</li> <li>● <b>yyyyMMddHHmmss:</b> 日志采集的起始时间和结束时间，UTC+0 时间。</li> </ul>	否

### ● 响应结果

名称	描述
id	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 <i>userName:signature</i> Content-Length: 192 Content-Type: text/plain Host:192.168.0.121:1443  {   "startTime": "1659837600000",                 </pre>
---

```
"endTime": "1659880800000",  
"servers": ["hblock_1", "hblock_2"],  
"logTypes": ["Config", "System"],  
"outputDirectory": "/mnt/storage01"  
}
```

### ● 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"
  }
}
```



## 4.12.4 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	描述	是否必须
filter	查询的过滤条件。 支持的查询条件为： <ul style="list-style-type: none"> <li>● status: 日志采集的状态。                             <ul style="list-style-type: none"> <li>■ Processing: 日志采集中。</li> <li>■ Succeeded: 采集成功。</li> <li>■ PartiallySucceeded: 部分采集成功。</li> <li>■ Failed: 采集失败。</li> </ul> </li> <li>● logTypes: 采集日志的类型                             <ul style="list-style-type: none"> <li>■ Config: 配置相关的日志。</li> <li>■ System: 系统相关的日志。</li> <li>■ Data: 数据处理相关的日志（仅集群版支持）。</li> <li>■ Coordination: 内部协调服务相关的日志（仅集群版支持）。</li> </ul> </li> </ul>	否

### ● 响应结果

名称	描述
----	----

logs	日志列表信息，包含：id、startTime、endTime、servers、logTypes、file、status、size、createTime。 类型：数组
id	日志 ID。
startTime	HBlock 日志采集的起始时间。
endTime	HBlock 日志采集的结束时间。
servers	HBlock 日志采集的服务器 ID。
logTypes	采集的 HBlock 日志类型： <ul style="list-style-type: none"> <li>● Config: 配置相关的日志。</li> <li>● System: 系统相关的日志。</li> <li>● Data: 数据处理相关的日志。</li> <li>● Coordination: 内部协调服务相关的日志（仅集群版支持）。</li> </ul>
file	日志文件存储路径及名称。
status	日志采集的状态： <ul style="list-style-type: none"> <li>● Processing: 日志采集中。</li> <li>● Succeeded: 采集成功。</li> <li>● PartiallySucceeded: 部分采集成功。</li> <li>● Failed: 采集失败。</li> </ul>
size	采集到的日志文件大小，单位为字节。
createTime	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_3.7.0_x64/collected_logs/hblock_logs_ea
35db6a939d40b9825c470a9f8826bb_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_54
dfd39212fc476dba61f761c08d8a8a_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
      }
    ]
  }
}
```

## 4.12.5 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	描述	是否必须
id	日志 ID。 如果不填写，查询所有的采集日志。	否

### ● 响应结果

名称	描述
logs	日志列表信息，包含：id、startTime、endTime、servers、logTypes、file、status、size、createTime。 类型：数组
id	日志 ID。
startTime	HBlock 日志采集的起始时间。
endTime	HBlock 日志采集的结束时间。
servers	日志采集的服务器 ID。
logTypes	采集的日志类型： <ul style="list-style-type: none"> <li>● Config: 配置相关的日志。</li> <li>● System: 系统相关的日志。</li> <li>● Data: 数据处理相关的日志（仅集群版支持）。</li> </ul>

	<ul style="list-style-type: none"> <li>● <b>Coordination:</b> 内部协调服务相关的日志（仅集群版支持）。</li> </ul>
<b>file</b>	日志文件存储路径及名称。
<b>status</b>	日志采集的状态： <ul style="list-style-type: none"> <li>● <b>Processing:</b> 日志采集中。</li> <li>● <b>Succeeded:</b> 采集成功。</li> <li>● <b>PartiallySucceeded:</b> 部分采集成功。</li> <li>● <b>Failed:</b> 采集失败。</li> </ul>
<b>size</b>	采集到的日志文件大小，单位为字节。
<b>createTime</b>	日志采集请求的创建时间。

### ● 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_3.7.0_x64/collected_logs/hblock_logs_54
dfd39212fc476dba61f761c08d8a8a_20240806034633_20240807054633.zip",
  "status": "Succeeded",
  "size": 104534,
  "createTime": 1723009566777
}
}
```



## 4.12.6 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	描述	是否必须
id	日志 ID。 类型：字符串	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_2022090  
5082242.zip  
Content-Type: application/zip;charset=utf-8  
Server: HBlock  
  
{ ..... }
```

## 4.12.7 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	描述	是否必须
id	日志 ID。 如果不填写，删除所有采集日志。 类型：字符串	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
```

## 4.12.8 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
```

### 4.12.9 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</i> [, <i>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 must be positive integer.	取值必须为正整数。
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。

## 4.13 HBlock 系统设置

### 4.13.1 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	描述	是否必须
userName	HBlock 的管理员用户名。 类型：字符串 取值：与初始化设置的用户名保持一致。	是
newPassword	用户设置的新密码的明文。 类型：字符串 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
```



## 4.13.2 邮件设置

### 4.13.2.1 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": reciverEmail
}
    
```

#### ● 请求参数

参数	描述	是否必须
status	是否启用邮件通知功能。 类型：枚举 取值： <ul style="list-style-type: none"> <li>● Enabled: 启用。</li> </ul>	是

	<ul style="list-style-type: none"> <li>● <b>Disabled:</b> 禁用。</li> </ul>	
smtpHost	SMTP 服务器。 取值: SMTP 服务器域名或 IP。 <b>注意:</b> 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	SMTP 端口号。 类型: 整型 取值: [1,65535]。 <b>说明:</b> <ul style="list-style-type: none"> <li>● 如果启用了 SSL, 默认端口为 465。</li> <li>● 如果禁用 SSL, 默认端口为 25。</li> </ul>	否
SSL	是否启用 SSL。 类型: 枚举 取值: <ul style="list-style-type: none"> <li>● <b>Enabled:</b> 启用 SSL。</li> <li>● <b>Disabled:</b> 禁用 SSL。</li> </ul> 默认值为 Disabled。	否
senderEmail	发件箱。 邮箱格式 <i>Local-part@domain</i> : <ul style="list-style-type: none"> <li>● <b>Local-part:</b> 字符串形式, 长度 1~64, 可包含字母、数字、特殊字符 (!# \$ % &amp; * + - / = ? ^ _ ` {   } ~ .), 字母区分大小写。句点 (.) 不能作为首尾字符, 也不能连续出现。</li> <li>● <b>domain:</b> 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:                             <ul style="list-style-type: none"> <li>■ 长度 1~63。</li> <li>■ 可包含字母、数字、短横线 (-), 字母区分大小写。</li> </ul> </li> </ul>	否

	<ul style="list-style-type: none"> <li>■ 顶级域名不能是纯数字。</li> <li>■ 短横线 (-) 不能作为首尾字符。</li> </ul> <p><b>注意：</b>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。</p>	
password	<p>邮箱授权码。</p> <p><b>说明：</b>授权码是邮箱推出的，用于第三方客户端登录的专用密码。</p> <p><b>注意：</b>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。</p>	否
receiverEmail	<p>收件箱。可以填写多个收件箱。</p> <p>邮箱格式 <i>Local-part@domain</i>：</p> <ul style="list-style-type: none"> <li>● <i>Local-part</i>：字符串形式，长度 1~64，可包含字母、数字、特殊字符 (! # \$ % &amp; * + - / = ? ^ _ ` {   } ~ .)，字母区分大小写。句点 (.) 不能作为首尾字符，也不能连续出现。</li> <li>● <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：                     <ul style="list-style-type: none"> <li>■ 长度 1~63。</li> <li>■ 可包含字母、数字、短横线 (-)，字母区分大小写。</li> <li>■ 顶级域名不能是纯数字。</li> <li>■ 短横线 (-) 不能作为首尾字符。</li> </ul> </li> </ul> <p><b>注意：</b>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。</p>	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
```

### 4.13.2.2 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
```

### 4.13.2.3 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	描述
status	是否启用邮件通知功能： <ul style="list-style-type: none"> <li>● Enabled: 启用。</li> <li>● Disabled: 禁用。</li> </ul>
smtpHost	SMTP 服务器。
smtpPort	SMTP 服务器端口号。
SSL	是否启用 SSL： <ul style="list-style-type: none"> <li>● Enabled: 启用。</li> <li>● Disabled: 禁用。</li> </ul>
senderEmail	发件箱。
receiverEmail	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"
  }
}
```



#### 4.13.2.4 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
```

## 4.13.2.5 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 satisfy constraint: Argument must not be null when enable email configuration.	启用邮件设置时，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。

### 4.13.3 远程协助

#### 4.13.3.1 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	描述	是否必须
serverId	服务器 ID。 如果不指定，默认设置当前被请求的服务器。 类型：字符串	否
status	是否启用远程协助。 类型：枚举 取值： <ul style="list-style-type: none"> <li>● Enabled: 启用。</li> <li>● Disabled: 禁用。</li> </ul>	是

host	远程协助的服务端 IP 或域名。 <b>说明：</b> 请联系软件供应商获取 host 信息。 除下列情况 host 必填外，其他情况选填： <ul style="list-style-type: none"> <li>● 如果启用远程协助时，此项在服务端没有保存过。</li> <li>● 如果设置远程协助时，输入了具体的端口号。</li> </ul>	否
port	远程协助的端口号。 <b>说明：</b> 请联系软件供应商获取端口号。 类型：整型 取值： [1,65535]，默认 18100。	否

### ● 响应结果

名称	描述
code	远程协助码，6 位数字。
host	远程协助的服务端 IP。
port	远程协助端口号。

### ● 请求示例

设置远程协助。

```

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 userName:signature

{
  "serverId": "hblock_1",
  "status": "Enabled",
  "host": "182.151.21.174",
  "port": 18100
    
```

```
}
```

- 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
  }
}
```

### 4.13.3.2 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	描述	是否必须
filter	查询的过滤条件。 支持的查询条件为：serverId（服务器 ID）。	否
range	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	描述
serverId	服务器 ID，若还没有初始化，则不显示此项。
status	远程协助是否启用。 <ul style="list-style-type: none"> <li>● Enabled: 启用。</li> <li>● Disabled: 禁用。</li> </ul>
host	远程协助的服务器 Host。

port	远程协助的端口号。
code	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"  
  }  
]  
}  
}
```

### 4.13.3.3 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
```

#### 4.13.3.4 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。

## 4.13.4 pushgateway 监控配置

### 4.13.4.1 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	描述	是否必须
serverId	指定添加 pushgateway 监控配置的 HBlock 服务器 ID。一次可以指定多个 HBlock 服务器的 ID，以英文逗号 (,) 分隔。	否

	分开。如果不填写，默认为所有 HBlock 服务器添加 pushgateway 监控配置。 类型：字符串	
pushgateway	指定 pushgateway 的地址和接口。 类型：字符串 取值：格式为 IPv4:port、[IPv6]:port 或者 domain-name:port。	是
pushgatewayLabels	指定 pushgateway 对应的标签值。 类型：字符串 取值：格式为 "key": "value"。	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
```

#### 4.13.4.2 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	描述	是否必须
serverId	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。 类型：字符串	否
pushgateway	指定 pushgateway 的地址和接口。 类型：字符串 取值：格式为 IPv4:port、[IPv6]:port 或者 domain-name:port。	否
pushgatewayTimeout	向 pushgateway 发送数据的超时时间。 类型：长整型 取值：[1000, 1000000]，单位为毫秒。	否
pushgatewayLabels	指定 pushgateway 对应的标签值。 类型：字符串 取值：格式为 "key": "value"。	否
changeAll	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 pushgatewayLabels 和 collectMetricItems 参数的设置。 注意：如果未指定 collectMetricItems，也未指定 pushgatewayLabels，则此参数不起作用。 类型：布尔 取值： <ul style="list-style-type: none"> <li>● true: 配置以覆盖方式更新。</li> <li>● false: 配置以添加方式更新。</li> </ul>	否



	默认值为 <b>false</b> 。	
<b>collectInterval</b>	指定采集时间间隔。 类型：长整型 取值：负整数或[2000, 1000000]范围的，单位为毫秒。 说明： <ul style="list-style-type: none"> <li>● 如果与 <b>collectMetric</b> 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。</li> <li>● 如果不指定 <b>collectMetric</b> 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。</li> </ul>	否
<b>collectMetric</b>	指定采集的监控指标。 类型：枚举 取值为： <b>server</b> 、 <b>fileSystem</b> 、 <b>interface</b> 、 <b>load</b> 、 <b>disk</b> 、 <b>tcp</b> 、 <b>os</b> 。 默认采集的上述所有监控指标。	否
<b>collectMetricItems</b>	指定监控指标下的配置项。目前仅监控指标 <b>disk</b> 、 <b>fileSystem</b> 、 <b>interface</b> 可以指定配置项。 说明： <ul style="list-style-type: none"> <li>● 如果指定 <b>collectMetric</b>，未指定配置项，默认修改指定 <b>collectMetric</b> 下的所有配置项。</li> <li>● 如果未指定 <b>collectMetric</b>，不能指定此参数。</li> </ul> 类型：字符串	否

● 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
```

### 4.13.4.3 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	描述	是否 必须

serverId	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。 类型：字符串	否
pushgateway	指定 pushgateway 的地址和接口。 类型：字符串 取值：格式为 IPv4:port、[IPv6]:port 或者 domain-name:port。	是
pushgatewayLabels	指定 pushgateway 对应的标签值。 类型：字符串 取值：label 项。	否
collectMetric	指定采集的监控指标。 类型：枚举 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	指定监控指标下的配置项。 类型：字符串	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
```

#### 4.13.4.4 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	描述	是否必须
serverId	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。 类型：字符串	否

##### ● 响应结果

名称	描述
serverId	服务器 ID。
send.pushgatewayTimeout	向所有 pushgateway 推送数据的超时时长，超时即认为推送失败。
send.pushgateways.address	pushgateway 的 IP 或域名。
send.pushgateways.labels	pushgateway 对应的标签。
collect.interval	硬件指标采集的采集间隔。



collect.metrics.name	采集间隔或采集对象的硬件指标类别名称。
collect.metrics.interval	硬件指标类别单独设置的采集间隔。
collect.metrics.items	硬件指标类别单独设置的采集对象。

### ● 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"
    ]
  }
]
}
}
```

## 4.13.4.5 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。

## 4.13.5 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。

### 4.13.5.1 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	描述	是否必须
url	智维平台第三方告警的推送地址。 类型：字符串	否

labels	指定推送至智维平台的告警数据中携带的标签信息。 类型：字符串 取值： <i>"key": "value"</i> 格式。 注意：标签 idc 和 mid 必须添加。	是
--------	---	---

● 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
    
```

### 4.13.5.2 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	描述	是否必须
url	智维平台第三方告警的推送地址。 类型：字符串	是
labels	指定推送至智维平台的告警数据中携带的标签信息。 类型：字符串 取值：“key”:“value”格式。	是
all	修改配置时，以覆盖方式更新，还是以添加方式更新。	否

	<p>类型：布尔</p> <p>取值：</p> <ul style="list-style-type: none"><li>● <b>true</b>：配置以覆盖方式更新。</li><li>● <b>false</b>：配置以添加方式更新。</li></ul> <p>默认值为 <b>false</b>。</p>	
--	--	--

### ● 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
```

### 4.13.5.3 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	描述	是否必须
url	智维平台第三方告警的推送地址。 类型：字符串	是
labels	指定要删除的标签信息。 <b>注意：</b> idc 和 mid 不能删除。 <b>说明：</b> 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。 类型：字符串	否



	取值: "key"。	
--	------------	--

- 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
```

#### 4.13.5.4 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	描述
CtyunCMS url	智维平台第三方告警的推送地址。
CtyunCMS labels	推送至智维平台的告警数据中携带的标签信息。

- 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"
      }
    }
  ]
}
```

#### 4.13.5.5 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。

## 4.14 调整 HBlock 性能参数（集群版适用）

### 4.14.1 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
x-update-parameter: key1=value1;key2=value2
Host: ip:port
Authorization: authorization

{
  "perfScen": perfScen,
  "haSens": haSens
}
```

#### ● 请求参数

参数	描述	是否必须
perfScen	设置性能调优针对的场景。 类型：字符串 取值： <ul style="list-style-type: none"> <li>● ICv1：智算场景。</li> <li>● VMv1：虚拟化场景。</li> <li>● default：系统默认的类型。</li> </ul>	否
haSens	设置性能调优针对的场景敏感型。 类型：字符串 取值：	否

	<ul style="list-style-type: none"> <li>● <b>High:</b> 高敏感型场景。</li> <li>● <b>Middle:</b> 中等敏感型场景。</li> <li>● <b>Low:</b> 低敏感型场景。</li> </ul> <p>默认值为 <b>Middle</b>。</p>	
--	---	--

- 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: text/plain
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
```

## 4.14.2 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	描述
perfScen	性能调优针对的场景： <ul style="list-style-type: none"> <li>● ICv1: 智算场景。</li> <li>● VMv1: 虚拟化场景。</li> <li>● default: 系统默认的类型。</li> </ul>
haSens	性能调优针对的场景敏感型： <ul style="list-style-type: none"> <li>● High: 高敏感型场景。</li> <li>● Middle: 中等敏感型场景。</li> <li>● Low: 低敏感型场景。</li> <li>● custom: 用户设置的场景。</li> <li>● Unknown: 场景未知。</li> </ul>

### ● 请求示例

查看性能调优配置。

```
GET /rest/v1/system/upgrade 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"
  }
}
```

### 4.14.3 错误码

HTTP status	错误码	错误信息	描述
400	StandaloneModeNotAllowed	' <i>operation</i> ' 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。

## 4.15 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	描述
version	HBlock 的版本号。
type	HBlock 的版本类型： <ul style="list-style-type: none"> <li>● Stand-alone: 单机版。</li> <li>● Cluster: 集群版本。</li> </ul>

### ● 请求示例

查询 HBlock 的版本信息。

```
GET /rest/v1/system/version HTTP/1.1
Date: Tue, 14 Jan 2025 08:41:57 GMT
Authorization: HBlock userName:signature
Host:192.168.0.121:1443
```

### ● 响应示例

```
HTTP/1.1 200 OK
x-hblock-request-id: ab2219e1469849fb81742a9395b9f22a
Connection: keep-alive
Content-Length: 45
```

```
Date: Tue, 14 Jan 2025 08:41:57 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock
```

```
{
  "data": {
    "version": "3.8.0",
    "type": "Cluster"
  }
}
```

## 4.16 升级 HBlock

### 4.16.1 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`。建议升级之前对系统进行检查，确保：

- 所有状态正常：
  - 所有 HBlock 服务器连接正常，没有处于删除状态的服务器。
  - HBlock 处于 `working`、`upgrading` 状态。
  - 软件许可证证书在有效期内，且本地卷总容量未超过许可证容量限制；或者试用期未结束。
- 如果卷是高可用类型，至少保证卷的主备连接正常。
- 没有处于还原状态、删除状态的卷。
- 系统整体数据冗余度不降级，正常数据百分比为 100%。并且，可用故障域数量和健康数量大于所有卷的写入需求。
- 升级监听服务（`stor:ua`）正常。
- 协议解析服务（`stor:ps`）正常。
- 基础服务正常：元数据管理服务（`stor:mdm`）、日志服务（`stor:ls`）、协调服务（`stor:cs`）。
- HBlock 使用的磁盘不存在 `Error` 状态。

**注意：**

- 执行升级 HBlock 前，确保每个服务器的 HBlock 安装路径对应的文件系统，存在至少 1 GiB 的可用空间。
- 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: application/zip
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
.....
```

- 请求参数

参数	描述	是否必须
Content-Type	请求体 MIME 类型。如果是同架构升级，取值为 application/zip; 如果混合部署升级，取值为 multipart/form-data; boundary= <i>value</i> , 其中 boundary 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。	否

<b>x-update-parameter</b>	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。  key 为参数名，value 为参数值，以“;”分割。	否
<b>upgradeFile</b>	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: Tue, 14 Jan 2025 09:40:35 GMT
Content-Type: application/zip
Authorization: HBlock userName:signature
Content-Length: 186862084
Host: 192.168.0.192:1443

upgradeFile: undefined
    
```

### ● 响应示例

```

HTTP/1.1 202 Accepted
Date: Tue, 14 Jan 2025 09:40:35 GMT
Content-Length: 0
Connection: keep-alive
Content-Type: application/json; charset=utf-8
x-hblock-request-id: c745e6cdd7264125841cf8be09571aa2
Server: HBlock
    
```

## 4.16.2 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	描述
status	HBlock 服务升级状态： <ul style="list-style-type: none"> <li>● NotUpgraded: 未进行过升级。</li> <li>● Preparation: 正在准备升级。</li> <li>● Processing: 正在升级。</li> <li>● Succeeded: 升级成功。</li> <li>● Failed: 升级失败。</li> <li>● Interrupted: 升级中断。</li> <li>● Unknown: 未知。</li> </ul>
progress	升级进程详细信息。
targetVersion	目标升级版本。
startTime	本次升级开始时间，unix 时间戳（UTC），精确到毫秒。
lastEndTime	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: Tue, 14 Jan 2025 09:33:23 GMT
```



```
Host: 192.168.0.192:1443
Authorization: HBlock userName:signature
```

## ● 响应示例

```
HTTP/1.1 200 OK
Date: Tue, 14 Jan 2025 09:33:23 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 5804
Connection:keep-alive
x-hblock-request-id: a14f668d771f4131a38d334ca7c381cc
Server: HBlock

{
  "data": {
    "status": "Succeeded",
    "progress": "Server info:\nhblock_1, 192.168.0.192, server1\nhblock_2,
192.168.0.202, server2\nhblock_3, 192.168.0.102, server3\n\n2025-01-14 17:14:54
CST+0800 [Step 1/5] Checking for system status...\n2025-01-14 17:14:54 CST+0800
Checking system info\n2025-01-14 17:14:54 CST+0800 Checking data status\n\n2025-01-14
17:14:54 CST+0800 [Step 2/5] Uploading update files...\n\r2025-01-14 17:14:58 CST+0800
Upload [100%] for hblock_1 ###[server 1/3]\n\r2025-01-14 17:15:01 CST+0800 Upload
[100%] for hblock_2 ###[server 2/3]\n\r2025-01-14 17:15:04 CST+0800 Upload [100%] for
hblock_3 ###[server 3/3]\n\n2025-01-14 17:15:04 CST+0800 [Step 3/5] Upgrade
preparing...\n\n2025-01-14 17:15:25 CST+0800 [Step 4/5] Installing updates...\n\r2025-
01-14 17:15:25 CST+0800 Upgrading re - Update[√] ### [module 1/15]\n\r2025-01-14
17:15:25 CST+0800 Upgrading lib - Update[√] ### [module 2/15]\n\r2025-01-14 17:15:25
CST+0800 Upgrading config - Update[√] ### [module 3/15]\n\r2025-01-14 17:15:39 CST+0800
Upgrading cs for hblock_1 - Check[√], Stop[√], Update[√], Start[√] ### [server 1/3]
[module 4/15]\n\r2025-01-14 17:16:24 CST+0800 Upgrading cs for hblock_2 - Check[√],
Stop[√], Update[√], Start[√] ### [server 2/3] [module 4/15]\n\r2025-01-14 17:17:10
CST+0800 Upgrading cs for hblock_3 - Check[√], Stop[√], Update[√], Start[√] ###
[server 3/3] [module 4/15]\n\r2025-01-14 17:17:55 CST+0800 Upgrading storfsBase -
```

```
Update[√] ### [module 5/15]\n\r2025-01-14 17:18:01 CST+0800 Upgrading ls for hblock_1 -
Check[√], Stop[√], Update[√], Start[√] ### [server 1/3] [module 6/15]\n\r2025-01-14
17:18:46 CST+0800 Upgrading ls for hblock_2 - Check[√], Stop[√], Update[√], Start[√]
### [server 2/3] [module 6/15]\n\r2025-01-14 17:19:31 CST+0800 Upgrading ls for
hblock_3 - Check[√], Stop[√], Update[√], Start[√] ### [server 3/3] [module
6/15]\n\r2025-01-14 17:20:46 CST+0800 Upgrading mdm for hblock_2 - Check[√], Stop[√],
Update[√], Start[√] ### [server 1/3] [module 7/15]\n\r2025-01-14 17:21:02 CST+0800
Upgrading mdm for hblock_1 - Check[√], Stop[√], Update[√], Start[√] ### [server 2/3]
[module 7/15]\n\r2025-01-14 17:21:19 CST+0800 Upgrading mdm for hblock_3 - Update[√]
### [server 3/3] [module 7/15]\n\r2025-01-14 17:21:24 CST+0800 Upgrading ds-1 for
hblock_1 - Check[√], Stop[√], Update[√], Start[√] ### [service 1/2] [server 1/3]
[module 8/15]\n\r2025-01-14 17:21:33 CST+0800 Upgrading ds-2 for hblock_1 - Check[√],
Stop[√], Update[√], Start[√] ### [service 2/2] [server 1/3] [module 8/15]\n\r2025-01-14
17:21:43 CST+0800 Upgrading ds-1 for hblock_2 - Check[√], Stop[√], Update[√], Start[√]
### [service 1/1] [server 2/3] [module 8/15]\n\r2025-01-14 17:21:53 CST+0800 Upgrading
ds-1 for hblock_3 - Check[√], Stop[√], Update[√], Start[√] ### [service 1/1] [server
3/3] [module 8/15]\n\r2025-01-14 17:22:02 CST+0800 Upgrading rb for hblock_1 -
Check[√], Stop[√], Update[√] ### [server 1/3] [module 9/15]\n\r2025-01-14 17:22:04
CST+0800 Upgrading rb for hblock_2 - Check[√], Stop[√], Update[√] ### [server 2/3]
[module 9/15]\n\r2025-01-14 17:22:06 CST+0800 Upgrading rb for hblock_3 - Check[√],
Stop[√], Update[√] ### [server 3/3] [module 9/15]\n\r2025-01-14 17:22:08 CST+0800
Upgrading ps for hblock_1 - Check[√], Stop[√], Update[√], Start[√] ### [server 1/3]
```

```
[module 10/15]\n\r2025-01-14 17:22:17 CST+0800 Upgrading ps for hblock_2 - Check[√],
Stop[√], Update[√], Start[√] ### [server 2/3] [module 10/15]\n\r2025-01-14 17:22:33
CST+0800 Upgrading ps for hblock_3 - Check[√], Stop[√], Update[√], Start[√] ###
[server 3/3] [module 10/15]\n\r2025-01-14 17:22:39 CST+0800 Upgrading tool - Update[√]
### [module 11/15]\n\r2025-01-14 17:22:45 CST+0800 Upgrading remote - Update[√] ###
[module 12/15]\n\r2025-01-14 17:22:51 CST+0800 Upgrading monitor for hblock_1 -
Check[√], Stop[√], Update[√], Start[√] ### [server 1/3] [module 13/15]\n\r2025-01-14
17:22:59 CST+0800 Upgrading monitor for hblock_2 - Check[√], Stop[√], Update[√],
Start[√] ### [server 2/3] [module 13/15]\n\r2025-01-14 17:23:06 CST+0800 Upgrading
monitor for hblock_3 - Check[√], Stop[√], Update[√], Start[√] ### [server 3/3] [module
13/15]\n\r2025-01-14 17:23:13 CST+0800 Upgrading ws for hblock_1 - Check[√], Stop[√],
Update[√], Start[√] ### [service 1/2] [server 1/3] [module 14/15]\n\r2025-01-14
17:23:19 CST+0800 Upgrading ms for hblock_1 - Check[√], Stop[√], Update[√], Start[√]
### [service 2/2] [server 1/3] [module 14/15]\n\r2025-01-14 17:23:31 CST+0800 Upgrading
ws for hblock_2 - Check[√], Stop[√], Update[√], Start[√] ### [service 1/2] [server
2/3] [module 14/15]\n\r2025-01-14 17:23:37 CST+0800 Upgrading ms for hblock_2 -
Check[√], Stop[√], Update[√], Start[√] ### [service 2/2] [server 2/3] [module
14/15]\n\r2025-01-14 17:23:48 CST+0800 Upgrading ws for hblock_3 - Check[√], Stop[√],
Update[√], Start[√] ### [service 1/2] [server 3/3] [module 14/15]\n\r2025-01-14
17:23:54 CST+0800 Upgrading ms for hblock_3 - Check[√], Stop[√], Update[√], Start[√]
### [service 2/2] [server 3/3] [module 14/15]\n\r2025-01-14 17:24:11 CST+0800 Upgrading
web - Update[√] ### [module 15/15]\n\r2025-01-14 17:24:11 CST+0800 [Step 5/5] Updating
system info...\n\r2025-01-14 17:24:12 CST+0800 [Post Upgrade] Cleaning...\n\r2025-01-14
```

```
17:24:12 CST+0800 Cleaning hblock_2 ###[server 1/3]\n2025-01-14 17:24:12 CST+0800
Cleaning hblock_3 ###[server 2/3]\n2025-01-14 17:24:12 CST+0800 Cleaning hblock_1
###[server 3/3]\n\n2025-01-14 17:24:12 CST+0800 [Upgrade Complete] The system has been
successfully upgraded to 3.8.0rc12 version",
    "targetVersion": "3.8.0rc12",
    "startTime": 1736846094364,
    "lastEndTime": 1736846652472
  }
}
```

### 4.16.3 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 newer.	不需要升级，当前运行的版本更新。
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.	系统中存在低冗余或错误数据，不允许操作。

		Please repair fault domains and disk paths.	许操作。请修复故障域和磁盘路径。
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 for the LUN <i>lunname, lunname, ...</i>	操作失败，Target IQNs 状态异常，请检查后重试。



## 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
<	&lt;
>	&gt;
(	&#40;
)	&#41;
&	&amp;

## 5.2 HBlock 服务

服务	服务名称	作用
stor:mdm	元数据管理服务	管理整个系统的元数据
stor:fc	故障转移控制服务	进行系统健康检测，实现故障转移控制
stor:ls	日志服务	提供基于日志的数据同步功能
stor:ds	数据服务	管理用户的文件数据块
stor:cs	协调服务	监视各服务器的状态，触发通知事件，确保集群服务高可用
stor:ms	管理服务	处理请求信息，维护集群运行状态。
stor:ws	监控服务	监控各个服务的状态，并负责服务的启动
stor:ps	协议解析服务	负责 iSCSI 协议解析与数据存储
stor:ag	数据采集服务	负责采集性能数据。
stor:ua	升级监听服务	负责接收升级请求，执行升级相关操作。

## 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

## ● 卷

事件	描述
CreateLUN	创建卷
DeleteLUN	删除卷
SetLUN	设置卷属性
ExpandLUN	扩容卷
EnableLUN	启用卷
DisableLUN	禁用卷
SwitchLUN	卷主备切换
RecoverLUN	还原卷
ResumeLUNRecovery	继续还原卷
SetBatchLUN	批量卷操作

## ● 系统

事件	描述
Login	登录
SetMailConfig	设置邮件通知
DeleteMailConfig	删除邮件通知
SendTestMail	发送测试邮件
SetRemoteAccess	设置远程协助
DeleteRemoteAccess	删除远程协助
ImportLicense	导入软件许可证
SetPassword	设置密码
StartLogCollect	发起日志收集请求

DeleteLogCollect	删除日志收集请求
SetAlarmMuteStatus	修改告警静默状态
ManuallyResolveAlarm	手动解除告警
Setup	初始化
AddMonitorConfig	新增监控配置
SetMonitorConfig	修改监控配置
DeleteMonitorConfig	删除监控配置
AddCtyunCMSConfig	新增智维平台告警推送配置
SetCtyunCMSConfig	修改智维平台告警推送配置
DeleteCtyunCMSConfig	删除智维平台告警推送配置

### ● 集群拓扑

事件	描述
SetNode	修改节点信息
CreateNode	创建节点
DeleteNode	删除节点

### ● 存储池

事件	描述
CreateStoragePool	创建存储池
DeleteStoragePool	删除存储池
SetStoragePool	修改存储池信息
AddNodeToPool	添加节点入池

事件	描述
RemoveNodeFromPool	移除池中节点

- 升级

事件	描述
StartUpgrade	开始升级

## 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

### ● 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
-------------------------	---------

● 目标 (Target)

事件	描述
InitiatorConnectionFailed	客户端连接断开
InitiatorConnectionNormal	客户端连接恢复

● 系统 (System)

事件	描述
ReachLicenseMaxCapacity	许可证容量达到上限
LicenseMaintenanceExpired	许可证过保
LicenseExpired	许可证过期
LicenseImported	许可证导入
DataResumed	数据恢复
DataLowRedundancy	数据降级
DataBalanceStart	数据均衡开始
DataBalanceProgress	数据均衡过程
DataBalanceFailed	数据均衡失败
DataBalanceEnd	数据均衡结束
DataAccessFailed	数据无法访问

● 故障域模块 (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

## 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。	字节/s	精细
	R_Bandwidth	客户端从 HBlock 读取数据的带宽。	字节/s	精细
	W_Bandwidth	客户端向 HBlock 写入数据的带宽。	字节/s	精细
	Latency	客户端与 HBlock 之间的总时延。系统在一个采集周期内，读写操作平均时延，反映 HBlock 处理读写请求的时长。	ms	精细
	W_Latency	客户端向 HBlock 写入数据的时延，系统在一个采集周期内，写操作平均时延，反映 HBlock 处理写请求的时长。	ms	精细
	R_Latency	客户端从 HBlock 读取数据的时延，集群在一个采集周期内，读操作平均时延，反映 HBlock 处理读请求的时长。	ms	精细
	Path_Cap	数据目录总容量。	字节	粗糙
Path_Used	数据目录已用容量。	字节	粗糙	

	Path_Rate	数据目录平均使用率，即所有数据目录使用率的平均值。	%	粗糙
	Path_Cap_Quota	HBlock 可用空间大小，即用户给 HBlock 分配的所有目录容量配额的总和。	字节	粗糙
	Path_Cap_Quota_Used	磁盘文件系统中，HBlock 数据占用的空间大小。	字节	粗糙
	Path_Cap_Quota_Rate	数据目录容量配额平均使用率，所有数据目录 $\text{Path\_Cap\_Quota\_Used}/\text{Path\_Cap\_Quota}$ 的平均值。	%	粗糙
	Cloud_Bandwidth	HBlock 与云之间的总带宽。	字节/s	精细
	Cloud_U_Bandwidth	HBlock 向云上传数据的带宽。	字节/s	精细
	Cloud_D_Bandwidth	HBlock 从云下载数据的带宽。	字节/s	精细
server	CPU_Rate	服务器 CPU 使用率。	%	精细
	Mem_Rate	服务器内存使用率。	%	精细
	Mem_Total	服务器内存总量。	字节	精细
	Mem_Used	服务器内存使用量。	字节	精细
	IOPS	客户端与 HBlock 之间的总 IOPS。	无	精细
	R_IOPS	客户端从 HBlock 读取数据的 IOPS。	无	精细
	W_IOPS	客户端向 HBlock 写入数据的 IOPS。	无	精细
	Bandwidth	客户端与 HBlock 之间的总带宽。	字节/s	精细
	R_Bandwidth	客户端从 HBlock 读取数据的带宽。	字节/s	精细
	W_Bandwidth	客户端向 HBlock 写入数据的带宽。	字节/s	精细

Latency	客户端与 HBlock 之间的总时延。采集周期内，服务器关联卷的读写时延平均值。	ms	精细
W_Latency	客户端到 HBlock 写时延，采集周期内，服务器关联卷的写时延平均值。	ms	精细
R_Latency	客户端从 HBlock 读取数据的时延，采集周期内，服务器关联卷的平均读时延。	ms	精细
Path_Cap	服务器的数据目录总容量。	字节	粗糙
Path_Used	服务器的数据目录已用容量。	字节	粗糙
Path_Rate	服务器的数据目录平均使用率，即服务器上所有数据目录使用率的平均值。	%	粗糙
Path_Cap_Quota	HBlock 可用空间大小，即用户给 HBlock 分配的所有目录容量配额的总和。	字节	粗糙
Path_Cap_Quota_Used	磁盘文件系统中，HBlock 数据占用的空间大小。	字节	粗糙
Path_Cap_Quota_Rate	数据目录容量配额平均使用率，即所有数据目录 $\text{Path\_Cap\_Quota\_Used}/\text{Path\_Cap\_Quota}$ 的平均值。	%	粗糙
Cloud_Bandwidth	HBlock 服务器与云之间的总带宽。	字节/s	精细
Cloud_U_Bandwidth	HBlock 服务器向云上传数据的带宽。	字节/s	精细

	Cloud_D_Bandwidth	HBlock 服务器从云下载数据的带宽。	字节/s	精细
disk	Path_Cap	数据目录总容量。	字节	粗糙
	Path_Used	数据目录已用容量。	字节	粗糙
	Path_Rate	数据目录平均使用率。	%	粗糙
	Path_Cap_Quota	HBlock 可用空间大小，即用户给 HBlock 分配的容量配额。	字节	粗糙
	Path_Cap_Quota_Used	磁盘文件系统中，HBlock 数据占用的空间大小。	字节	粗糙
	Path_Cap_Quota_Rate	数据目录容量配额使用率，即 $\text{Path\_Cap\_Quota\_Used}/\text{Path\_Cap\_Quota}$ 。	%	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。	字节/s	精细
	R_Bandwidth	客户端从 HBlock 卷读取数据的带宽。	字节/s	精细
	W_Bandwidth	客户端向 HBlock 卷写入数据的带宽。	字节/s	精细
	Latency	客户端与 HBlock 之间的总时延，卷在一个采集周期内，读写操作平均时延，反映 HBlock 单卷处理读写请求的时长。	ms	精细

	W_Latency	客户端向 HBlock 卷写入数据的时延，卷在一个采集周期内，写操作平均时延，反映 HBlock 单卷处理写请求的时长。	ms	精细
	R_Latency	客户端从 HBlock 卷读取数据的时延，卷在一个采集周期内，读操作平均时延，反映 HBlock 单卷处理读请求的时长。	ms	精细
	Cloud_Bandwidth	HBlock 卷与云之间的总带宽。	字节/s	精细
	Cloud_U_Bandwidth	HBlock 卷向云上传数据的带宽。	字节/s	精细
	Cloud_D_Bandwidth	HBlock 卷从云下载数据的带宽。	字节/s	精细
	Wait_Upload	待从 HBlock 卷上传云的数据量。	字节	精细
pool（仅集群版支持）	IOPS	客户端与 HBlock 之间的总 IOPS。	无	精细
	R_IOPS	客户端从 HBlock 读取数据的 IOPS。	无	精细
	W_IOPS	客户端向 HBlock 写入数据的 IOPS。	无	精细
	Bandwidth	客户端与 HBlock 之间的总带宽。	字节/s	精细
	R_Bandwidth	客户端从 HBlock 读取数据的带宽。	字节/s	精细
	W_Bandwidth	客户端向 HBlock 写入数据的带宽。	字节/s	精细
	Latency	客户端与 HBlock 之间的总时延。系统在一个采集周期内，读写操作平均时延，反映 HBlock 处理读写请求的时长。	ms	精细
	W_Latency	客户端向 HBlock 写入数据的时延，系统在一个采集周期内，写操作平	ms	精细



		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分区的文件系统的总容量。	字节	粗糙
Path_Used		数据目录已用容量，即存储池中所有数据目录所在磁盘分区的文件的已用容量	字节	粗糙
Path_Rate		数据目录平均使用率，即存储池所有数据目录使用率的平均值。	%	粗糙
Path_Cap_Quota		HBlock 可用空间大小，即存储池中用户给 HBlock 分配的所有目录容量配额的总和。	字节	粗糙
Path_Cap_Quota_Used		存储池中所有已加入 HBlock 的数据目录中的 HBlock 数据大小。	字节	粗糙
Path_Cap_Quota_Rate		数据目录容量配额平均使用率，存储池中所有数据目录 $\text{Path\_Cap\_Quota\_Used}/\text{Path\_Cap\_Quota}$ 的平均值。	%	粗糙

## 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，或数据目录所在服务器被移除	是	每天 1 次
数据目录所在磁盘写入速度慢 DiskWriteSlow	警告	数据目录所在磁盘写入速度慢	数据目录所在盘恢复正常	数据目录被移除，数据目录从存储池中移除，或数据目录所在服务器被移除	是	每天 1 次
许可证即将到期 LicenseWillExpire	警告	当前时间（告警模块所在服务器的系统时间）距离最后导入的许可证的到期时间 $\leq 15$ 天且 $> 0$ 天	当前时间（告警模块所在服务器的系统时间）距离告警许可证的到期时间 $> 15$ 天	许可证过期，或导入新的许可证（不同 id 的许可证）	是	每天 1 次
许可证过期 LicenseExpired	严重	当前时间（告警模块所在服务器的系统时间）距离最后导入的许可证的到期时间 $\leq 0$ 天	当前时间（告警模块所在服务器的系统时间）距离告警许可证的到期时间 $> 0$ 天	导入新的许可证（不同 id 的许可证）	是	发送 1 次
许可证维保即将到期 LicenseMaintenanceWillExpire	警告	当前时间（告警模块所在服务器的系统时间）距离最后导入的许可证的维保到期时间 $\leq 15$ 天且 $> 0$ 天	当前时间（告警模块所在服务器的系统时间）距离告警许可证的维保到期时间 $> 15$ 天	许可证过期，或导入新的许可证（不同 id 的许可证）	是	每天 1 次
许可证过保 LicenseMaintenanceExpired	警告	当前时间（告警模块所在服务器的系统时间）距离最后导入的许可证的维保到期时间 $\leq 0$ 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	无解除条件，只能手动解除	导入新的许可证	是	每天 1 次

		试用期过期时间 $\leq 15$ 天且 $\geq 0$ 天				
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 $\geq 8000$	告警条数 $< 7500$	无	是	每天 1 次
告警邮件发送失败 FailToSendAlarmEmail	严重	告警邮件发送失败	告警邮件发送成功	邮件配置被删除, 或邮件发送设置为 disable	是	每天 1 次
配额使用率超阈值 CapacityQuotaUsageExceedsThreshold	警告	存储池中数据目录关联磁盘的 Path_Cap_Quota_Rate $\geq 80\%$ , 数据目录层级的数据目录关联磁盘的 Path_Cap_Quota_Rate $\geq 80\%$ <b>说明:</b> 数据目录未设置容量配额, 则按容量配额=磁盘总容量计算。	存储池中数据目录关联磁盘的 Path_Cap_Quota_Rate $< 75\%$ , 或数据目录层级的数据目录关联磁盘的 Path_Cap_Quota_Rate $< 75\%$ <b>说明:</b> 数据目录未设置容量配额, 则按容量配额=磁盘总容量计算。	<ul style="list-style-type: none"> <li>存储池名称变更</li> <li>存储池中的数据目录被全部移除</li> </ul>	是	每天 1 次
配额用尽 CapacityQuotaUsageApproachLimit	严重	基础存储池中数据目录对应磁盘总配额使用率 $\geq 95\%$	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 $\geq 95\%$ , 或数据目录对应磁盘配额使用率 $\geq 95\%$	非基础存储池中数据目录对应磁盘总配额 $< 90\%$ , 或数据目录对应磁盘配额使用率 $< 90\%$	<ul style="list-style-type: none"> <li>存储池名称变更</li> <li>存储池中的数据目录被全部移除</li> </ul>	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 $\geq 80\%$ , 或数据目录对应磁盘的 Path_Rate $\geq 80\%$	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 $< 75\%$	<ul style="list-style-type: none"> <li>存储池名称变更</li> <li>存储池中的数据目录被全部移除</li> </ul>	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	<ul style="list-style-type: none"> <li>卷删除</li> <li>存储池名称变更</li> <li>卷禁用</li> <li>卷删除失败</li> </ul>	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	<ul style="list-style-type: none"> <li>● 数据目录被移除</li> <li>● 数据目录从存储池中移除</li> <li>● 数据目录健康状态变为“Error”</li> </ul>	是	每天 1 次
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	<ul style="list-style-type: none"> <li>● 数据目录被移除。</li> <li>● 数据目录从存储池中移除</li> </ul>	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	<ul style="list-style-type: none"> <li>● 服务器被移除</li> <li>● 数据目录被移除</li> <li>● 数据目录从存储池中移除</li> <li>● 数据目录健康状态变为“Error”</li> <li>● 存储池名称变更</li> </ul>	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	<ul style="list-style-type: none"> <li>● 服务器被移除</li> <li>● 数据目录被移除</li> <li>● 数据目录从存储池中移除</li> <li>● 存储池名称变更</li> </ul>	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	<ul style="list-style-type: none"> <li>● 服务器被移除</li> <li>● Target 被删除</li> <li>● Target 被迁移</li> </ul>	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	<ul style="list-style-type: none"> <li>● 服务器被移除</li> <li>● 数据目录被移除</li> <li>● 数据目录从存储池中移除</li> <li>● 故障域健康状态变为“Error”</li> <li>● 存储池名称变更或故障域全路径名称中任</li> </ul>	是	每天 1 次

				一节点名称变更 <ul style="list-style-type: none"> <li>故障域中的数据目录被全部移除</li> </ul>		
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	<ul style="list-style-type: none"> <li>服务器被移除</li> <li>数据目录被移除</li> <li>数据目录从存储池中移除</li> <li>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</li> <li>故障域中的数据目录被全部移除</li> </ul>	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	<ul style="list-style-type: none"> <li>卷被删除或服务器被移除，卷禁用，卷删除失败</li> <li>Target 发生迁移</li> <li>还原中的卷产生的此告警，在卷还原完成后，告警失效</li> </ul>	是	每天 1 次
云账户异常 CloudAccountAbnormal	严重	云端读取或写入失败：欠费冻结、欠费冻结或违规冻结	云端读取或写入成功一次	卷被删除、卷禁用或卷删除失败	是	每天 1 次
卷云端数据冲突 LUNCloudDataConflict	严重	卷对应的云端数据出现比本地更新的版本	卷对应的云端数据版本全部比本地旧	卷被删除、卷禁用或卷删除失败	是	每天 1 次
卷云端心跳冲突 LUNCloudHeartbeatConflict	严重	卷的云端出现来自非本集群的心跳	卷的云端心跳仅来自本集群	卷被删除、卷禁用或卷删除失败	是	每天 1 次
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	该客户端与 Target 连接成功。	<ul style="list-style-type: none"> <li>告警的 IQN 所在 Target 被删除。</li> <li>Target 被迁移</li> </ul>	是	每天 1 次

		客户端主动断开连接的情况除外。				
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： <ul style="list-style-type: none"> <li>● 元数据管理服务异常：集群中 stor:mdm 服务有 2 个，仅允许一个故障，故障时发出告警</li> <li>● 故障转移控制服务异常：集群中 stor:fc 服务有 2 个，仅允许一个故障，故障时发出告警</li> <li>● 日志服务异常：集群中 stor:ls 服务有 3 个，仅允许一个故障，故障时发出告警</li> <li>● 协调服务异常：集群中 stor:cs 服务有 3 个，仅允许一个故障，故障时发出告警</li> </ul>	匹配的告警实例恢复到以下程度： <ul style="list-style-type: none"> <li>● 元数据管理服务在告警机器上恢复正常：告警机器上的 stor:mdm 服务恢复正常</li> <li>● 故障转移控制服务在告警机器上恢复正常：告警机器上的 stor:fc 服务恢复正常</li> <li>● 日志服务在告警机器上恢复正常：告警机器上的 stor:ls 服务恢复正常</li> <li>● 协调服务在告警机器上恢复正常：告警机器上的 stor:cs 服务恢复正常</li> </ul>	告警机器上的基础服务迁移完成	是	每天 1 次
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

## 5.7 OOS Endpoint 和 Region

中国大陆区域对象存储网络中的各个地区，使用统一的 OOS Endpoint: oos-cn.ctyunapi.cn，region 为 cn。

香港节点分为**香港精品网络**和**香港普通网络**两种方式：香港精品网 OOS Endpoint: oos-cnkhqnet.ctyunapi.cn，region 为 cnhk-hqnet；香港普通网 OOS Endpoint: oos-cnkh-nqnet.ctyunapi.cn，region 为 cnhk-nqnet。

**说明：**如果您的数据存储在中国大陆区域对象存储网络中的某个资源池，可以直接使用该资源池的 OOS Endpoint。OOS Endpoint 列表如下（Endpoint 列表仅为资源池 Endpoint 访问信息描述，与资源状态无关联）：

地区	OOS Endpoint	region
郑州	oos-hazz.ctyunapi.cn	hazz
沈阳	oos-lnsy.ctyunapi.cn	lnsy
四川成都	oos-sccd.ctyunapi.cn	sccd
乌鲁木齐	oos-xjwlmq.ctyunapi.cn	xjwlmq
甘肃兰州	oos-gslz.ctyunapi.cn	gslz
山东青岛	oos-sdqd.ctyunapi.cn	sdqd
贵州贵阳	oos-gzgy.ctyunapi.cn	gzgy
湖北武汉	oos-hbwh.ctyunapi.cn	hbwh
西藏拉萨	oos-xzls.ctyunapi.cn	xzls
安徽芜湖	oos-ahwh.ctyunapi.cn	ahwh
广东深圳	oos-gdsz.ctyunapi.cn	gdsz
江苏苏州	oos-jssz.ctyunapi.cn	jssz
上海 2	oos-sh2.ctyunapi.cn	sh2

## 5.8 HBlock 可推送的操作系统监控指标

类别	指标名称	含义
Status	hblock_status_system	HBlock 服务状态 <ul style="list-style-type: none"> <li>● 0: 表示 Working, 运行中。</li> <li>● 1: 表示 Upgrading, 升级中。</li> <li>● 2: 表示 Uninstalling, 卸载中。</li> <li>● 3: 表示 Unknown, 未知。</li> </ul>
	hblock_status_data	HBlock 整体数据健康状态: <ul style="list-style-type: none"> <li>● Normal: 数据正常的比例。</li> <li>● LowRedundancy: 数据低冗余的比例。</li> <li>● Error: 数据错误的比例。</li> </ul>
	hblock_status_license	软件许可证状态: <ul style="list-style-type: none"> <li>● 0: 表示订阅有效。</li> <li>● 1: 表示维保有效。</li> <li>● 2: 表示未导入许可证。</li> <li>● 3: 表示维保过期。</li> <li>● 4: 订阅过期。</li> </ul>
	hblock_status_licenseExpiredTime	软件许可证/试用期过期时间。
	hblock_status_poolDomain	存储池的故障域状态（仅集群版支持）: <ul style="list-style-type: none"> <li>● 0: 表示 Healthy, 健康状态, 可读写。</li> <li>● 1: 表示 Warning, 警告状态, 可读。</li> <li>● 2: 表示 Error, 错误状态, 无法读写。</li> <li>● 3: 表示 Unknown, 状态未知。</li> </ul>
	hblock_status_baseService	基础服务运行状态（仅集群版支持）: <ul style="list-style-type: none"> <li>● 0: 表示 Healthy, 健康状态。</li> </ul>



		<ul style="list-style-type: none"> <li>● 1: 表示 Warning, 警告状态。</li> <li>● 2: 表示 Error, 错误状态。</li> <li>● 3: 表示 Unkown, 状态未知。</li> </ul>
	hblock_status_dataService	数据服务运行状态（仅集群版支持）： <ul style="list-style-type: none"> <li>● 0: 表示 Healthy, 健康状态。</li> <li>● 1: 表示 Warning, 警告状态。</li> <li>● 2: 表示 Error, 错误状态。</li> <li>● 3: 表示 Unknown, 状态未知。</li> </ul>
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： <ul style="list-style-type: none"> <li>● 0: 表示状态为 down</li> <li>● 1: 表示状态为 up</li> <li>● 2: 表示状态为 unknown</li> <li>● 3: 表示状态为 notpresent</li> <li>● 4: 表示状态为 lowerlayerdown</li> <li>● 5: 表示状态为 testing</li> <li>● 6: 表示状态为 dormant</li> <li>● -2: 表示未识别的状态</li> </ul>
	hblock_network_up_count	端口 up 次数
hblock_network_down_count	端口 down 次数	
tcp	hblock_netstat_tcp_RetransSegs	TCP 重传的报文数量
	hblock_netstat_tcp_OutSegs	TCP 输出的报文数量
	hblock_netstat_tcp_InSegs	TCP 接收的报文数量

	hblock_netstat_tcp_ActiveOpens	当前 ActiveOpen 状态的 TCP 连接数
	hblock_netstat_tcp_CurrEstab	当前 CurrEstab 状态的 TCP 连接数
	hblock_netstat_tcp_PassiveOpens	当前 PassiveOpens 状态的 TCP 连接数
	hblock_sockstat_tcp_mem	当前 mem 状态的 TCP 连接数
	hblock_sockstat_tcp_alloc	当前 alloc 状态的 TCP 连接数
	hblock_sockstat_tcp_inuse	当前 inuse 状态的 TCP 连接数
	hblock_sockstat_tcp_orphan	当前 orphan 状态的 TCP 连接数
	hblock_sockstat_tcp_tw	当前 tw 状态的 TCP 连接数
disk	hblock_disk_read_bytes	硬盘读请求数据量
	hblock_disk_written_bytes	写请求数据量
	hblock_disk_reads_completed	读请求次数
	hblock_disk_read_time_seconds	读请求的时间
	hblock_disk_writes_completed	写请求次数
	hblock_disk_write_time_seconds	写请求的时间
	hblock_disk_io_time_seconds	处理 I/O 操作的时间
	hblock_disk_io_time_weighted_seconds	处理 I/O 操作的加权时间
	hblock_disk_io_now	当前正在运行的实际 I/O 请求数
fileSystem	hblock_fileSystem_size_bytes	文件系统总容量
	hblock_fileSystem_free_bytes	文件系统剩余容量
	hblock_fileSystem_free_inode_count	空闲的 inode 数量
	hblock_fileSystem_total_inode_count	总的 inode 数量
	hblock_fileSystem_readonly	文件系统是否为只读
OS	hblock_os_boot_time_seconds	服务器最近一次启动时间
Cloud	hblock_cloud_wait_upload_bytes	待上传云的数据
	hblock_cloud_upload_size_bytes	已上传云的数据
	hblock_cloud_download_size_bytes	从云上已下载的数据

LUN	hblock_blockDevice_read_count	IO 读次数（包含读成功、读失败）
	hblock_blockDevice_read_fail_count	IO 读失败次数
	hblock_blockDevice_read_time_seconds	IO 读累积耗时（包含读成功、读失败）
	hblock_blockDevice_read_fail_time_seconds	IO 读失败累积耗时
	hblock_blockDevice_read_bytes	IO 读成功数据量大小
	hblock_blockDevice_write_count	IO 写次数（包含写成功、写失败）
	hblock_blockDevice_write_fail_count	IO 写失败次数
	hblock_blockDevice_write_time_seconds	IO 写累积耗时（包含写成功、写失败）
	hblock_blockDevice_write_fail_time_seconds	IO 写失败累积耗时
	hblock_blockDevice_write_bytes	IO 写成功数据量大小
diskpath	hblock_dataDir_total_bytes	数据目录对应磁盘的总容量
	hblock_dataDir_avail_bytes	数据目录对应磁盘的可用容量
	hblock_dataDir_quota_total_bytes	数据目录的配额空间
	hblock_dataDir_quota_used_bytes	数据目录的配额已用空间

## 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	描述	是否必须
name	根节点名称。 类型：字符串 取值：字符串形式，长度范围 1~63，只能由字母、数字、句点(.)、下划线(_)和短横线(-)组成，字母区分大小写，且仅支持以字母或数字开头。 默认为 default。	否
type	根节点的类型。 类型：字符串 取值：root：根节点。 默认为 root。	否
childNodes	子节点信息集合，根据子节点类型不同，需要的参数不同。	是

		类型：数组	
子节点类型为 room	name	子节点名称。 类型：字符串 取值：字符串形式，长度范围 1~63，只能由字母、数字、句点(.)、下划线(_)和短横线(-)组成，字母区分大小写，且仅支持以字母或数字开头。	是
	type	子节点类型。 类型：字符串 取值：room：机房类型。	是
	description	子节点描述信息。 类型：字符串 取值：1~50 位字符串。	否
	childNodes	子节点信息集合，根据子节点类型不同，需要的参数不同，子节点可以为 rack 或者 server。 类型：数组	否
子节点类型为 rack	name	子节点名称。 类型：字符串 取值：字符串形式，长度范围 1~63，只能由字母、数字、句点(.)、下划线(_)和短横线(-)组成，字母区分大小写，且仅支持以字母或数字开头。	是
	type	子节点类型。 类型：字符串 取值：rack：机架类型。	是
	description	子节点描述信息。	否

		类型：字符串 取值：1~50 位字符串。	
	<b>childNodes</b>	子节点信息集合，子节点只能为 <b>server</b> 。 类型：数组	否
子节点类型为 <b>server</b>	<b>name</b>	子节点名称。 类型：字符串 取值：字符串形式，长度范围 1~63，只能由字母、数字、句点(.)、下划线(_)和短横线(-)组成，字母区分大小写，且仅支持以字母或数字开头。 默认为服务器 ID。	否
	<b>type</b>	子节点类型。 类型：字符串 取值： <b>server</b> ：服务器类型。	是
	<b>description</b>	子节点描述信息。 类型：字符串 取值：1~50 位字符串。	否
	<b>ip</b>	HBlock 的服务器 IP。 取值：IPv4 或 [IPv6] 地址。	是
	<b>apiPort</b>	管理 API 端口号。 类型：整型 取值：[1, 65535]，默认值为 1443。需要和该服务器安装 HBlock 时设置的 API 端口号保持一致。	否
	<b>childNodes</b>	子节点信息集合。子节点的类型为 <b>path</b> 。	是
		<b>name</b>	子节点名称。

子节点类型为 path		类型：字符串 取值：数据目录具体路径。	
	type	子节点类型。 类型：字符串 取值：path：数据目录类型。	是
	capacityQuota	指定数据目录的容量配额，即针对此数据目录，HBlock 可写入的数据总量。当 HBlock 的使用空间一旦达到配额，就立刻阻止数据写入，不允许再使用超出配额的空间。 类型：长整型 取值：小于数据目录所在磁盘的总容量，单位是字节。负整数表示无限制写入，0 表示禁止写入。默认不限制写入。 <b>注意：</b> 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"
    }
  ]
}
```

```
    ]  
  }  
] }  
] }  
] }  
}
```