

腾讯云数据仓库 TCHouse-D

快速入门

产品文档



腾讯云

【版权声明】

©2013-2024 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

【商标声明】

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

文档目录

快速入门

通过控制台使用腾讯云数据仓库 TCHouse-D

通过客户端使用腾讯云数据仓库 TCHouse-D

快速入门

通过控制台使用腾讯云数据仓库 TCHouse-D

最近更新时间：2024-06-27 10:39:58

本文档将展示如何通过控制台使用腾讯云数据仓库 TCHouse-D，进行集群创建、管理等操作，并可通过 SQL 工作区执行 SQL 命令。

新建集群

1. 在 腾讯云数据仓库 TCHouse-D 介绍页单击**立即选购**。或登录 腾讯云数据仓库 TCHouse-D 控制台单击**新建集群**，可进入购买页进行配置与购买。
2. 逐项完成基础配置、集群配置、日志配置等配置的选择后，系统将展示相应的配置费用，单击**立即购买**可实现集群创建。

集群配置

集群名称

长度限制为6-36个字符，只允许包含中文、字母、数字、-、_

内核版本

高可用 启用高可用

启用高可用后，系统会部署3个FE，从而实现读写高可用（HA）。
非高可用的情况，只有一个FE，不建议生产环境中使用，特别是在线查询或者实时读写的场景。

FE节点类型 **标准型** 高性能型

计算规格

存储规格 GB

单节点最小支持200GB，最大支持32000GB

FE节点数量

如果开启高可用节点数量大于等于3台
可配置的节点数量不能超过所选网络可用子网IP数0 - BE节点数量，如若子网IP数量不足请切换子网或vpc尝试。

BE节点类型 **标准型** 高性能型

计算规格

存储规格 GB

单节点最小支持200GB，最大支持32000GB

BE节点数量

节点数量大于等于3台
可配置的节点数量不能超过所选网络可用子网IP数0 - FE节点数量，如若子网IP数量不足请切换子网或vpc尝试。

查看集群信息

1. 集群创建后，您即可进入 [腾讯云数据仓库 TCHouse-D 控制台](#)，选择集群所在地域后，可从集群列表中查看该地域下全部集群的状态信息。

集群列表 香港 1 其它地域 2

[新建集群](#)

| ID/名称 | 状态(进度) | 健康状态 | FE节点 | BE节点 | 内核版本 | 可用区 | 网络 | 子网 | 作 |
|-------|--------|------|--------------------------|--------------------------|------|------|----|----|---|
| | 运行中 | 运行良好 | 标准型, 1个节点 4核16G, 200G | 标准型, 4个节点 4核16G, 200G | 1.2 | 香港二区 | | | 投 |

共 1 条

2. 在集群列表单击 **集群 ID/名称**，可以查看集群基本信息、集群状态、配置信息和网络信息等。

集群信息
基本信息

集群ID ██████████ [🔗](#)
 集群名称 ██████ [✎](#)
 付费类型 **按量计费**
 创建时间 **2023-03-31 18:10:33**
 标签 [修改](#)

配置信息

内核版本 **1.2 (tencent-cdw-doris-1.2.3-rc01-8ec14a7)**
 高可用 **非高可用**
 FE节点配置 **标准型, 4核16G / 1个节点 / 增强型SSD云硬盘200GB**
 BE节点配置 **标准型, 4核16G / 4个节点 / 增强型SSD云硬盘200GB**

集群状态

集群状态 **运行中**

网络信息

可用区 **香港二区**
 VPC ID ██████████
 子网 ID ██████████
 JDBC连接地址 ██████████ [复制连接!](#)
 HTTP连接地址 ██████████ [0 复制连接!](#)

节点信息

| 序号 | 节点类型 |
|----|------|
| 1 | FE |
| 2 | BE |
| 3 | BE |
| 4 | BE |
| 5 | BE |

共 5 条

集群操作

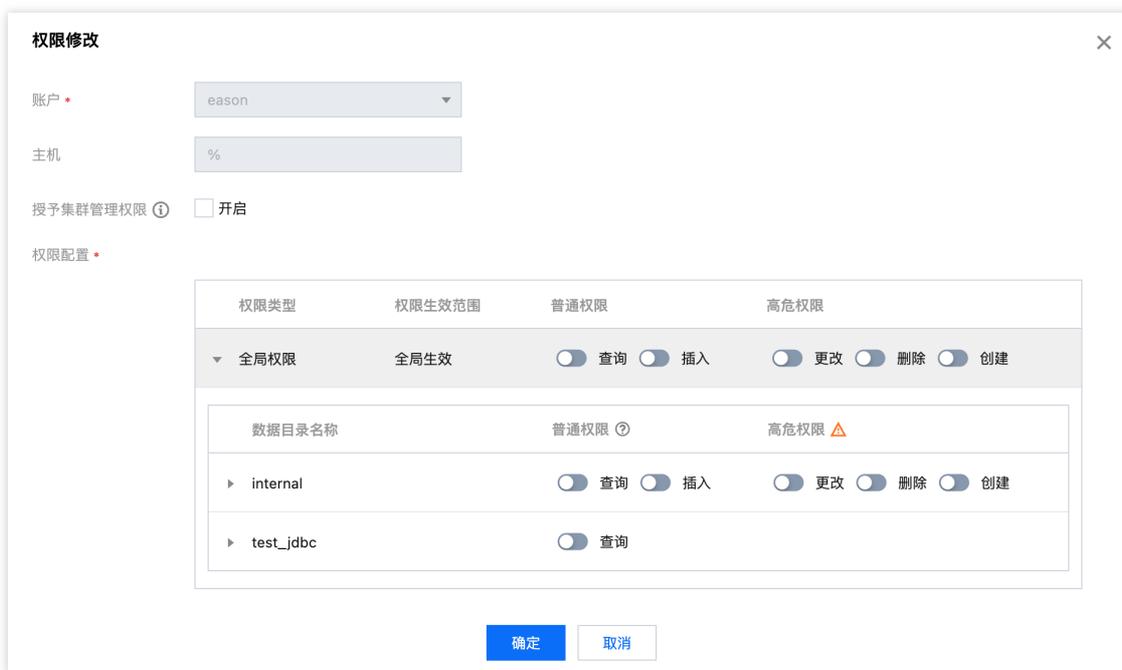
在控制台中，您可以方便地进行账户权限管理、监控告警配置、配置修改、节点管理等操作，详情可参见 [操作指南](#) 章节。

账户权限管理

1. 腾讯云数据仓库 TCHouse-D 在控制台中提供了一个可视化界面，能够对集群便捷、高效地开展账户管理、权限管理操作。



2. 单击**新增账户**按钮，填写数据库账户、密码、确认密码和描述（选填），单击**确定**即可新增账户。单击**修改权限**可进到权限修改窗口，该窗口也可用来查看权限。

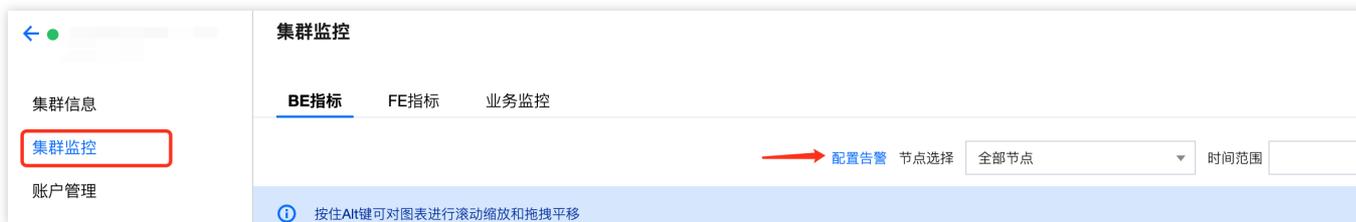


监控告警配置

1. 腾讯云数据仓库 TCHouse-D 提供了丰富的监控指标，方便您掌握集群的运行状态，并可针对指标配置告警以实时关注运行信息，实现快速响应。单击**集群 ID/名称**进入集群详情页，单击**集群监控**可查看集群的各项性能指标。



2. 支持创建灵活全面的报警策略，可以对所有监控指标进行异常告警。在集群监控页面内，单击**配置告警**可新建告警策略。



配置修改

1. 腾讯云数据仓库 TCHouse-D 支持灵活的参数配置方式，您可以在控制台中全面、直观地设置 BE/FE/Broker 等参数，并支持查看历史修改记录。在集群列表中单击**集群 ID/名称**查看集群详情，在左侧列表中选择**配置管理 > 修改配置**进入参数配置页。

- 集群信息
- 集群监控
- 账户管理
- 数据管理
 - 备份恢复
 - 数据库审计
- 查询管理
 - 查询分析
 - SQL工作区
- 配置管理
 - **修改配置**
 - 修改历史
- 节点管理
- 操作记录

修改配置

配置文件 上传

- apache_hdfs_broker.conf
- be.conf
- fe.conf
- core-site.xml
- hdfs-site.xml
- hive-site.xml
- odbcinst.ini

集群维度
节点维度

应用到集群
重置所有修改
重启服务

| 参数名 | 参考值 | 当前运行值 | 修改参? |
|-------------------------|-----|---|---------|
| broker_ipc_port ⓘ | | 8000 | |
| client_expire_seconds ⓘ | | 300 | |
| XXMX ⓘ | 2g | 2g ✎ | >=2g, < |

[新增参数](#)

2. 针对 Broker、BE、FE 对应的配置参数，支持配置文件修改、参数热更新。针对 XML 配置文件支持整文件修改。

节点操作

1. 节点管理页面展示集群中各角色的状态列表，角色包含 FE、BE、BROKER。对每个角色，会展示健康状态、进程状态、节点IP及最近重启时间（在角色管理页面最后一次手动进行服务重启操作的时间）。

版权所有：腾讯云计算（北京）有限责任公司

第9 共41页

- 集群信息
- 集群监控
- 账户管理
- 数据管理
 - 备份恢复
 - 数据库审计
- 查询管理
 - 查询分析
 - SQL工作区
- 配置管理
 - 修改配置
 - 修改历史
- 节点管理
- 操作记录

节点管理

立即重启
优雅重启
滚动重启
启动

| 角色 | 节点健康状态 | 角色进程状态 | 节点IP |
|---------------------------------|--------|--------|------|
| <input type="checkbox"/> FE | 运行良好 | 正常 | |
| <input type="checkbox"/> BROKER | 运行良好 | 正常 | |
| <input type="checkbox"/> BE | 运行良好 | 正常 | |
| <input type="checkbox"/> BROKER | 运行良好 | 正常 | |
| <input type="checkbox"/> BE | 运行良好 | 正常 | |
| <input type="checkbox"/> BROKER | 运行良好 | 正常 | |
| <input type="checkbox"/> BE | 运行良好 | 正常 | |
| <input type="checkbox"/> BROKER | 运行良好 | 正常 | |

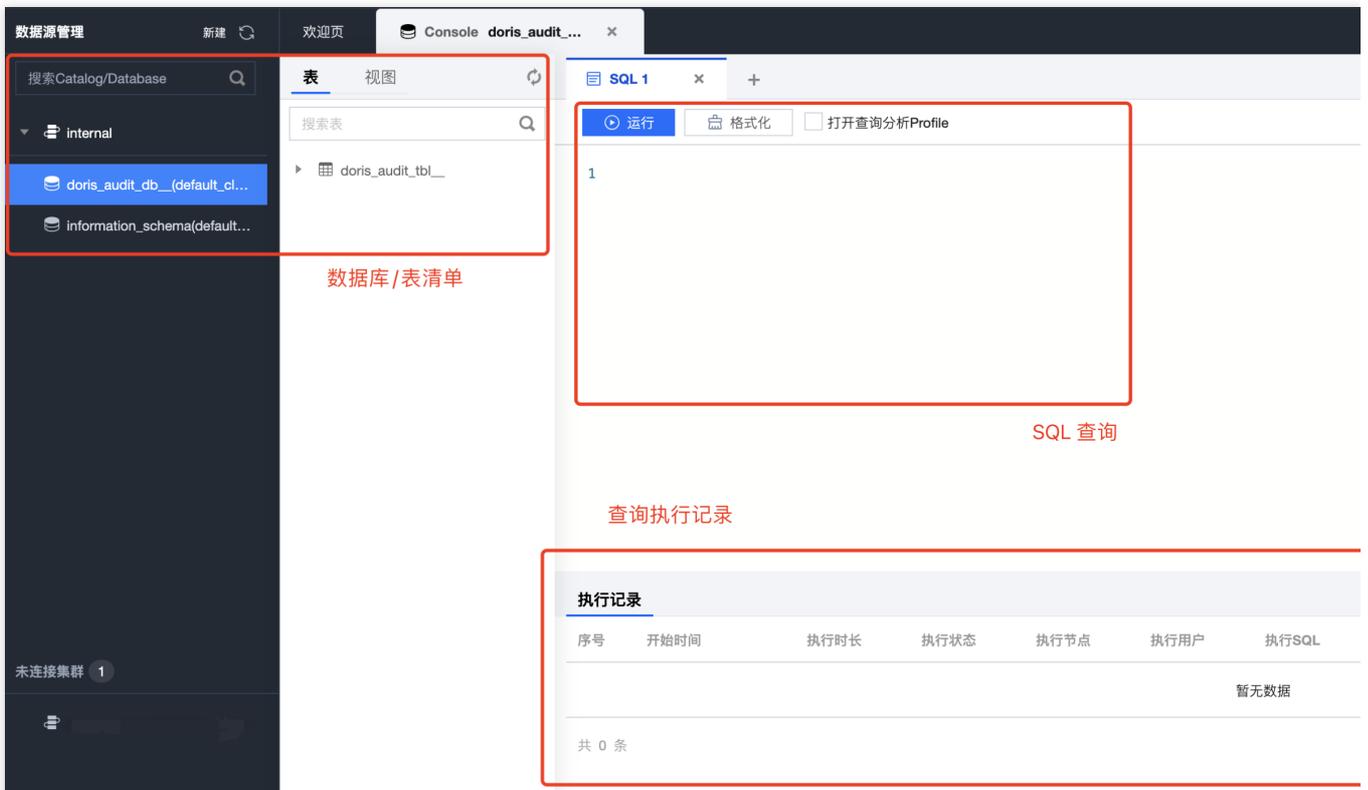
2. 您可对各节点执行重启操作，重启方式包含立即重启、优雅重启、滚动重启三种，支持对同一类节点批量执行。

执行SQL命令

1. 通过 SQL 工作区，您可以快捷地连接集群，使用 SQL 命令开展一系列操作，详情参见 [SQL 工作区](#)。在控制台左侧列表单击即可访问 **SQL 工作区**。在集群列表中选中目标**集群 ID/名称**，在查询管理下也可找到入口。



2. 在 SQL 工作区内，您可以执行 SQL 语句，查看执行记录等操作。



通过客户端使用腾讯云数据仓库 TCHouse-D

最近更新时间：2024-06-27 10:45:22

除控制台操作外，您也可基于 JDBC 和 WebUI 两种方式，通过客户端使用腾讯云数据仓库 TCHouse-D。购买创建集群后，从 [集群列表](#) 页面单击 **集群 ID/名称** 可以进入集群详情页，网络信息部分提供两个访问地址：JDBC & HTTP，分别对应了两类客户端。其中：

JDBC 连接地址适合 JDBC 类客户端，如常用的 MySQL 客户端。

HTTP 连接地址则用于通过 WebUI 调用集群的 REST API 进行各种操作。



The screenshot displays the 'Cluster Information' (集群信息) page in the Tencent Cloud console. It is divided into several sections:

- 集群信息 (Cluster Information):**
 - 基本信息 (Basic Information):**
 - 集群ID: [Redacted]
 - 集群名称: [Redacted]
 - 付费类型: 按量计费
 - 创建时间: 2024-03-13 15:08:19
 - 标签: [修改](#)
 - 配置信息 (Configuration Information):**
 - 内核版本: 1.2 (tencent-cdw-doris-1.2.8-4a67fdc-a7d2ef9)
 - 高可用: 非高可用
 - FE节点配置: 标准型, 4核16G / 1个节点 / 增强型SSD云硬盘200GB
 - BE节点配置: 标准型, 4核16G / 3个节点 / 增强型SSD云硬盘200GB
- 集群状态 (Cluster Status):**
 - 集群状态: 运行中
- 网络信息 (Network Information):**
 - 可用区: 曼谷二区
 - VPC ID: [Redacted]
 - 子网 ID: [Redacted]
 - JDBC连接地址: [Redacted]
 - HTTP连接地址: [Redacted]
 - 节点信息 (Node Information):**

| 序号 | 节点类型 |
|----|------|
| 1 | FE |
| 2 | BE |
| 3 | BE |
| 4 | BE |

共 4 条

JDBC 类客户端（MySQL Client 等）

腾讯云数据仓库 TCHouse-D 采用 MySQL 协议进行通信，用户可通过 MySQL Client 或者 MySQL JDBC 连接到集群。选择 MySQL Client 版本时建议采用 5.1 之后的版本，因为 5.1 之前不能支持长度超过 16 个字符的用户名。

注意：

只要支持 MySQL JDBC 协议的客户端都可使用，不过可能存在集群返回信息没有被此客户端原样展示的情况。该现象一般不会造成问题，但可能由于部分信息未被展示，造成对操作结果的误判。

例如，执行 Broker Load 后返回信息中的 Warning 在某些客户端中未被展示，可能会让使用者误以为没有 Warning（即没有被过滤的数据）。

下面展示通过 MySQL Client 连接到腾讯云数据仓库 TCHouse-D 的基本流程。

步骤1：创建用户

1. 购买并创建腾讯云数据仓库TCHouse-D 集群时，需设定 admin 用户的密码。



2. 集群创建后，您可在控制台中便捷查看集群信息，开展新建账户等操作。您可以使用 admin 用户连接集群，也可以新建用户并以子用户连接。

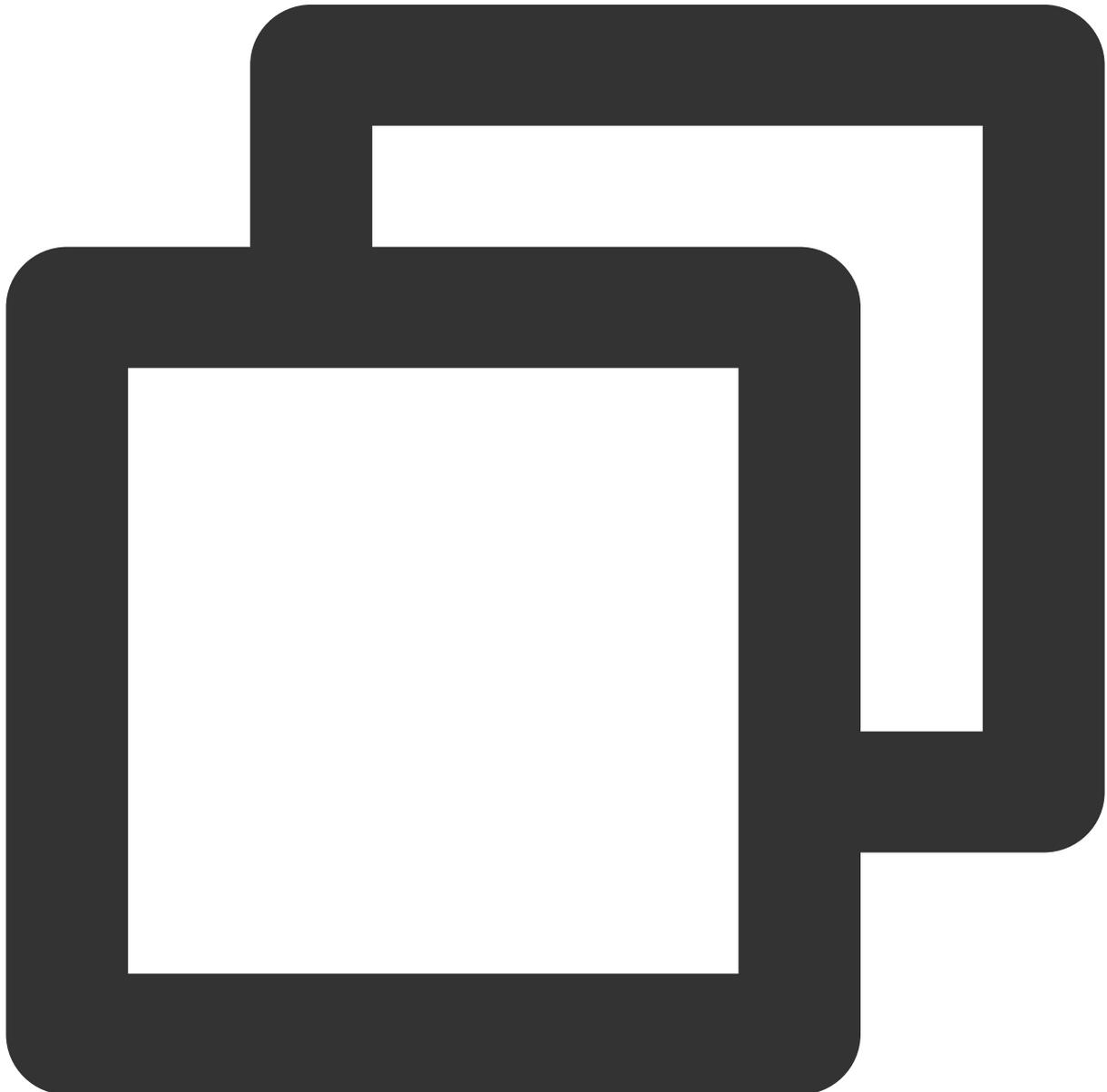


| 账户 | 描述 |
|-------|-------------------|
| admin | 系统用户，权限不能修改，也不能删除 |
| | - |
| | - |

3. 集群运行中，您可在网络联通的环境下使用 MySQL Client 连接到集群。集群详情页中，支持一键复制连接地址，若您希望通过子用户连接或重置密码，均可在控制台中 **账户管理** 页面操作。

| 网络信息 | |
|----------|---|
| 可用区 | 广州四区 |
| VPC ID | mysql-h172.16.1.155 -P9030 -uadmin -p   |
| 子网 ID | jdbc:mysql://172.16.1.155:9030  |
| JDBC连接地址 | 172.16.1.155:9030 复制连接串 |
| HTTP连接地址 | 172.16.1.155:8030 复制连接串 |

4. 输入以下命令，可登录到集群：

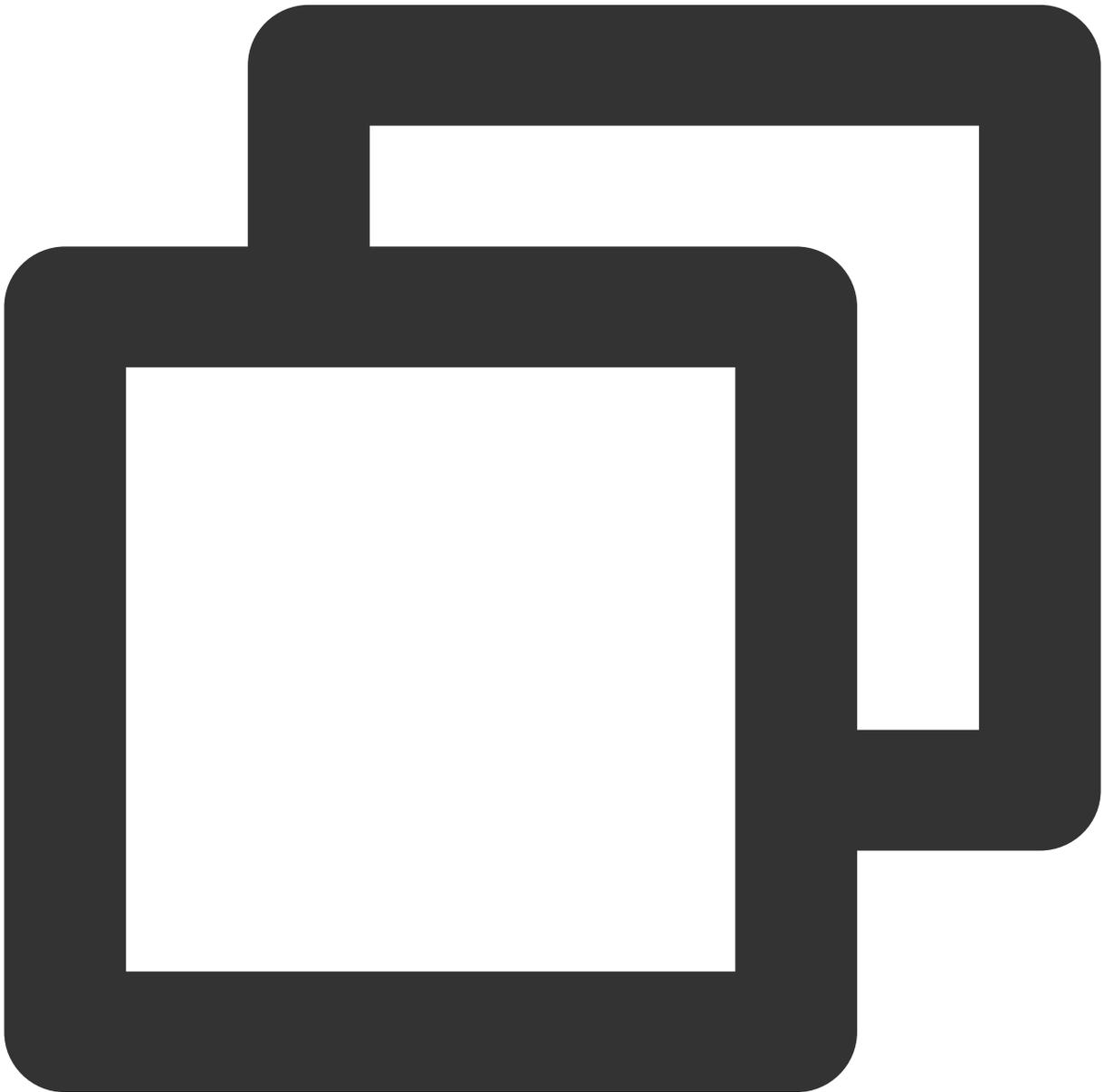


```
mysql -h FE_HOST -P9030 -uadmin -p
```

说明：

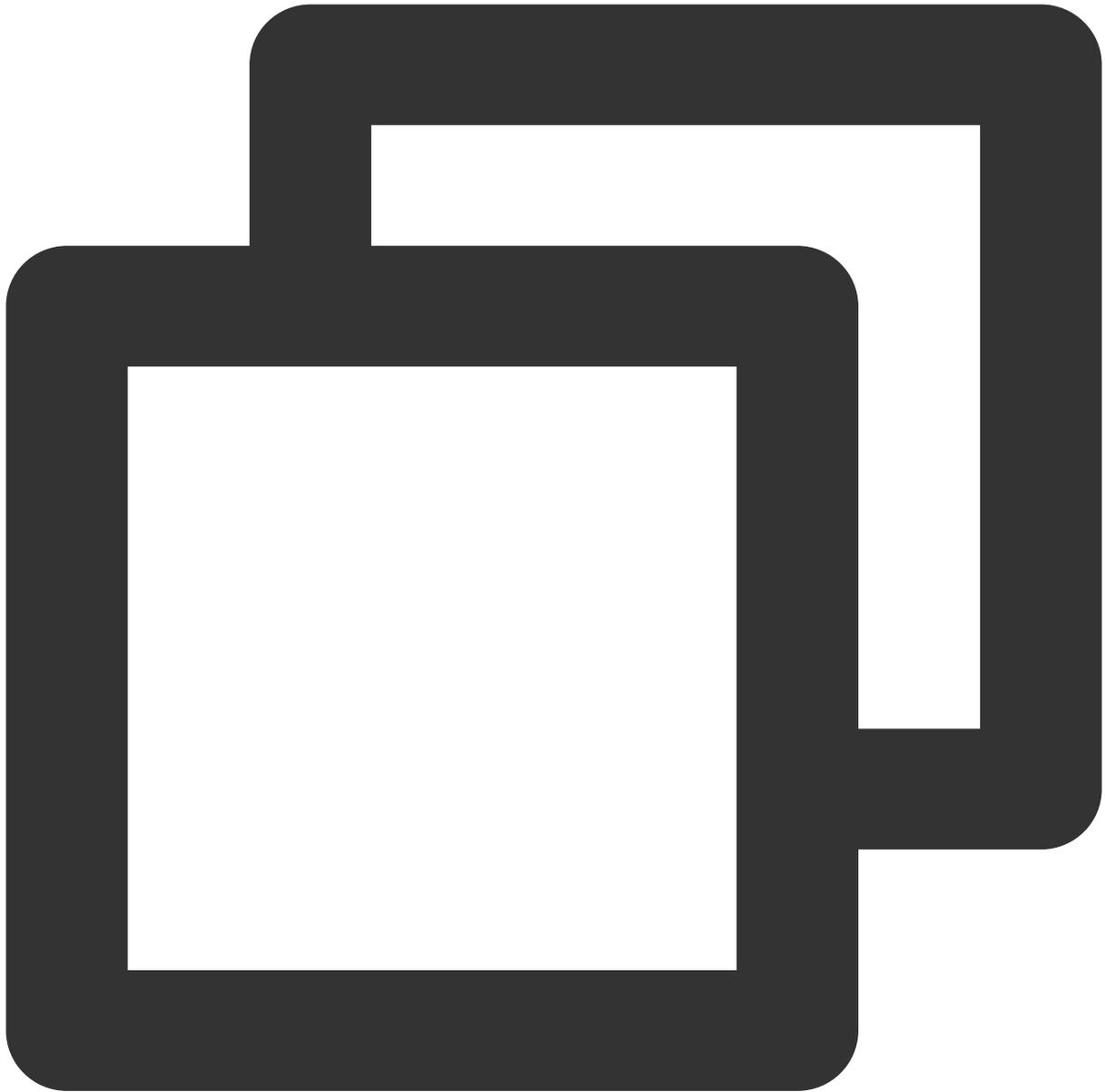
FE_HOST 是任一 FE 节点的 IP 地址。9030是 fe.conf 中的 query_port 配置。

5. 登录后，也可以通过以下命令修改 admin 密码：



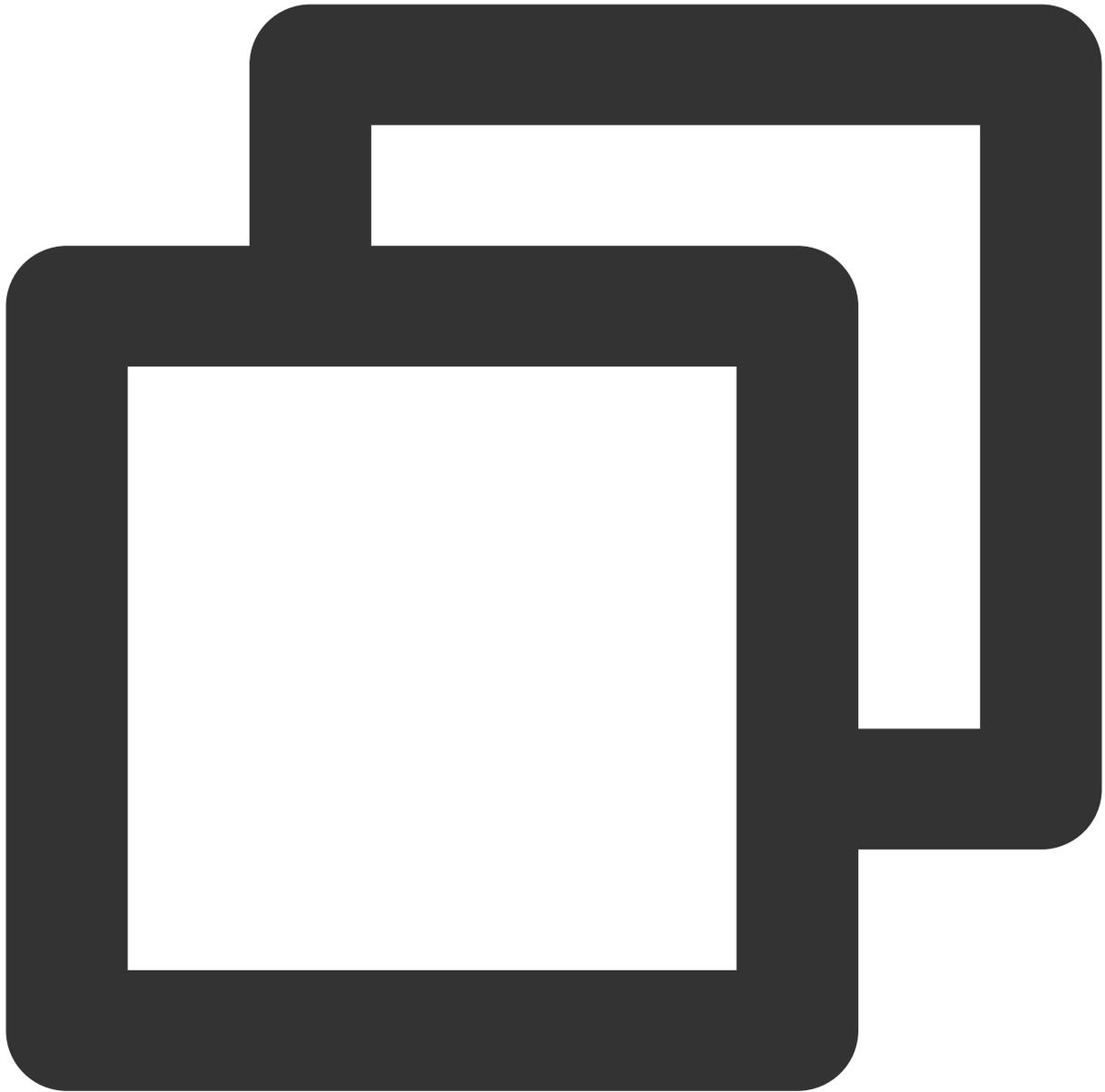
```
SET PASSWORD FOR 'admin' = PASSWORD('your_password');
```

6. 创建新用户：



```
CREATE USER 'test' IDENTIFIED BY 'test_passwd';
```

7. 可通过以下命令，使用新创建用户连接集群：



```
mysql -h FE_HOST -P9030 -utest -ptest_passwd
```

说明：

新创建的普通用户默认没有任何权限，您可在控制台 **账户管理** 页面操作授权。

步骤2：数据表创建

1. 创建数据库。

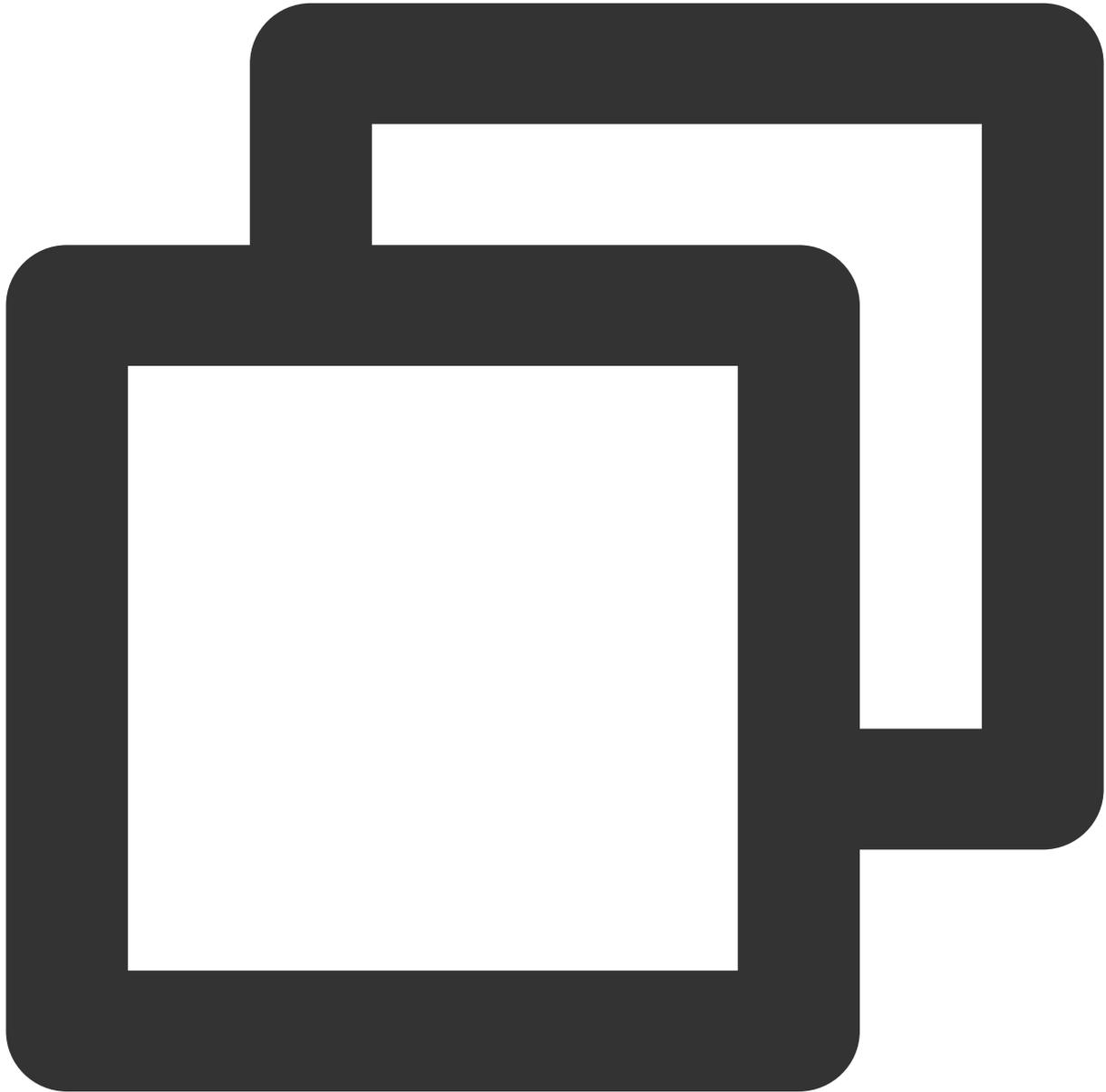
初始可以通过 admin 用户创建数据库：`CREATE DATABASE example_db;`。

说明：

所有命令都可以使用 'HELP command;' 查看到详细的语法帮助。如：`HELP CREATE DATABASE;`。

如果不清楚命令的全名，可以使用 "help 命令某一字段" 进行模糊查询。如键入 'HELP CREATE'，可以匹配到 `CREATE DATABASE`，`CREATE TABLE`，`CREATE USER` 等命令。

数据库创建完成之后，可以通过 `SHOW DATABASES;` 查看数据库信息。



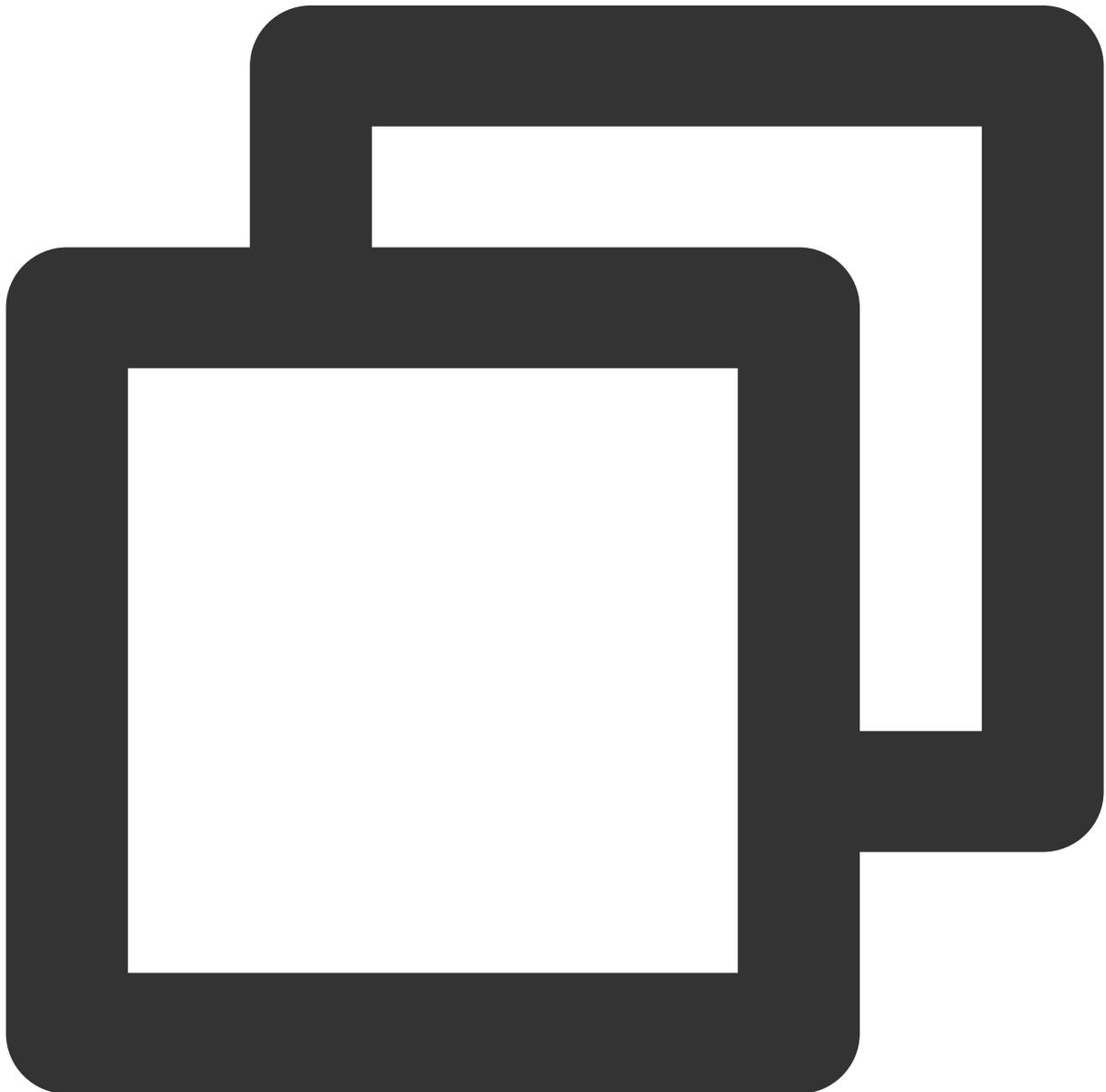
```
MySQL> SHOW DATABASES;
+-----+
| Database          |
+-----+
| doris_audit_db__  |
```

```
| example_db      |
| information_schema |
+-----+
3 rows in set (0.00 sec)
```

information_schema 是为了兼容 MySQL 协议而存在，实际中信息可能不是很准确，所以关于具体数据库的信息建议通过直接查询相应数据库而获得。

2. 账户授权。

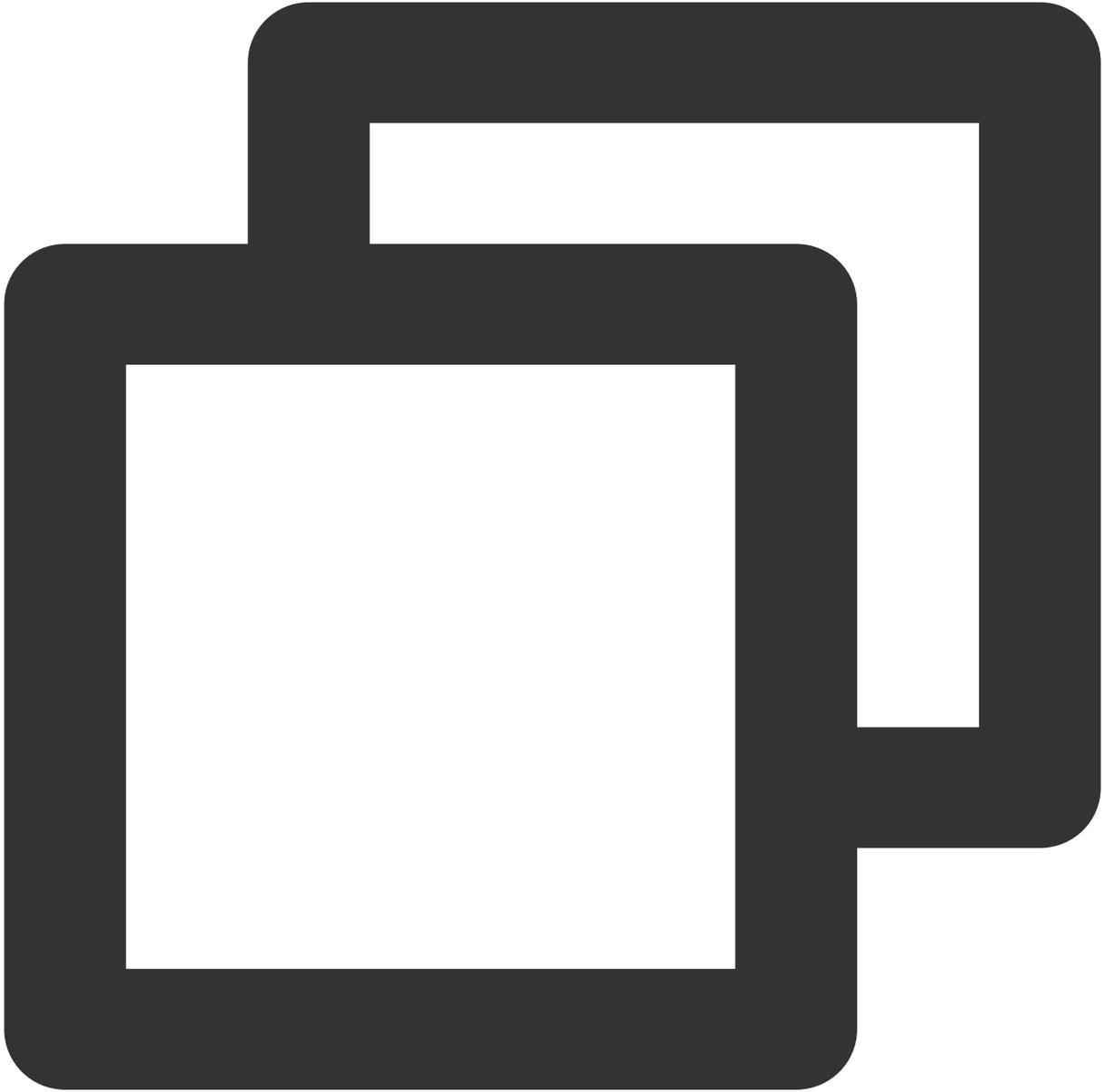
example_db 创建完成之后，可以通过 admin 账户将 example_db 读写权限授权给普通账户，如 test。授权之后采用 test 账户登录就可以操作 example_db 数据库了。



```
GRANT ALL ON example_db TO test;
```

3. 建表。

首先切换数据库：



```
USE example_db;
```

使用 `CREATE TABLE` 命令建立一个表。腾讯云数据仓库 TCHouse-D 支持单分区和复合分区两种建表方式，具体参见 [数据分区和分桶](#)。下面以聚合模型为例，分别演示两种分区的建表语句。

单分区

建立一个名字为 table1 的逻辑表。分桶列为 siteid，桶数为 10，表的 schema 如下：

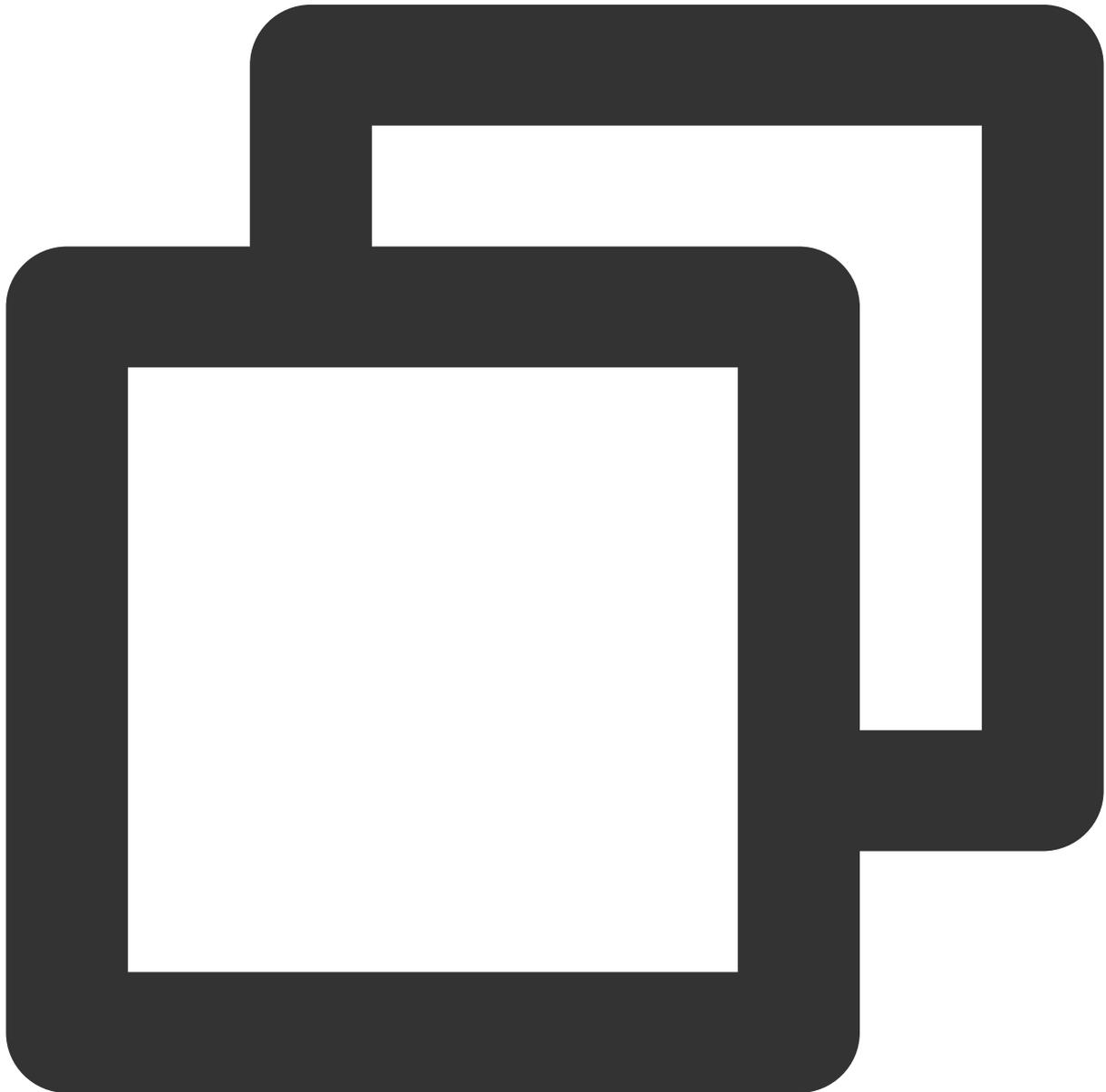
siteid：类型是 INT（4字节），默认值为10。

citycode：类型是 SMALLINT（2字节）。

username：类型是 VARCHAR，最大长度为32，默认值为空字符串。

pv：类型是 BIGINT（8字节），默认值是0；这是一个指标列，数据会默认做聚合操作，聚合方法是求和（SUM）。

建表语句如下：



```
CREATE TABLE table1
```

```
(
  siteid INT DEFAULT '10',
  citycode SMALLINT,
  username VARCHAR(32) DEFAULT '',
  pv BIGINT SUM DEFAULT '0'
)
AGGREGATE KEY(siteid,citycode,username)
DISTRIBUTED BY HASH(siteid) BUCKETS 10
PROPERTIES("replication_num" = "1");
```

复合分区

建立一个名字为 table2 的逻辑表，表的 schema 如下：

event_day：类型是 DATE，无默认值。

siteid：类型是 INT（4字节），默认值为10。

citycode：类型是 SMALLINT（2字节）。

username：类型是 VARCHAR，最大长度为32，默认值为空字符串。

pv：类型是 BIGINT（8字节），默认值是0；这是一个指标列，数据会默认做聚合操作，聚合方法是求和（SUM）。

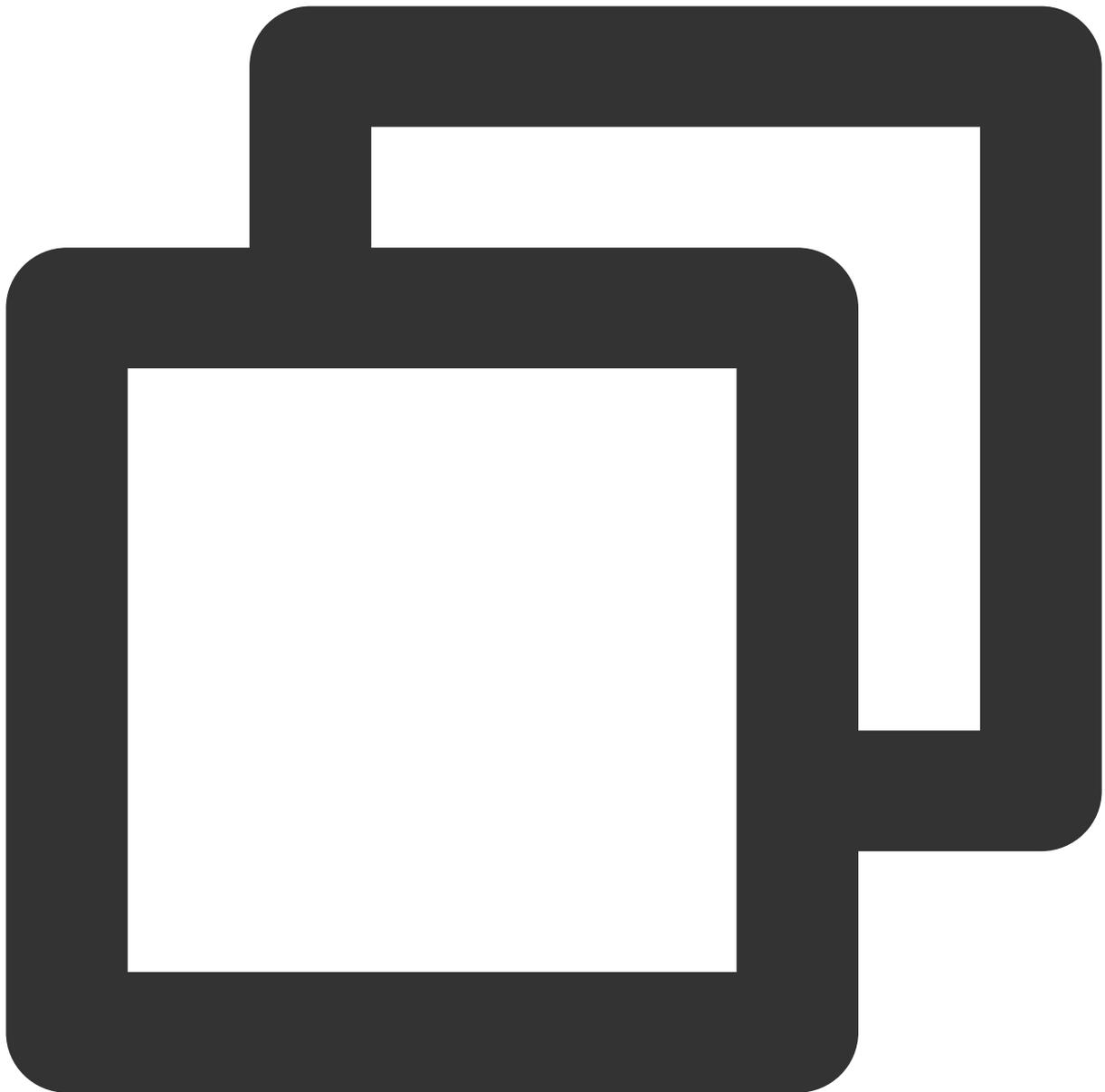
使用 event_day 列作为分区列，建立3个分区: p201706、p201707、p201708，每个分区使用 siteid 进行哈希分桶，桶数为10。

p201706：范围为 [最小值, 2017-07-01)

p201707：范围为 [2017-07-01, 2017-08-01)

p201708：范围为 [2017-08-01, 2017-09-01)

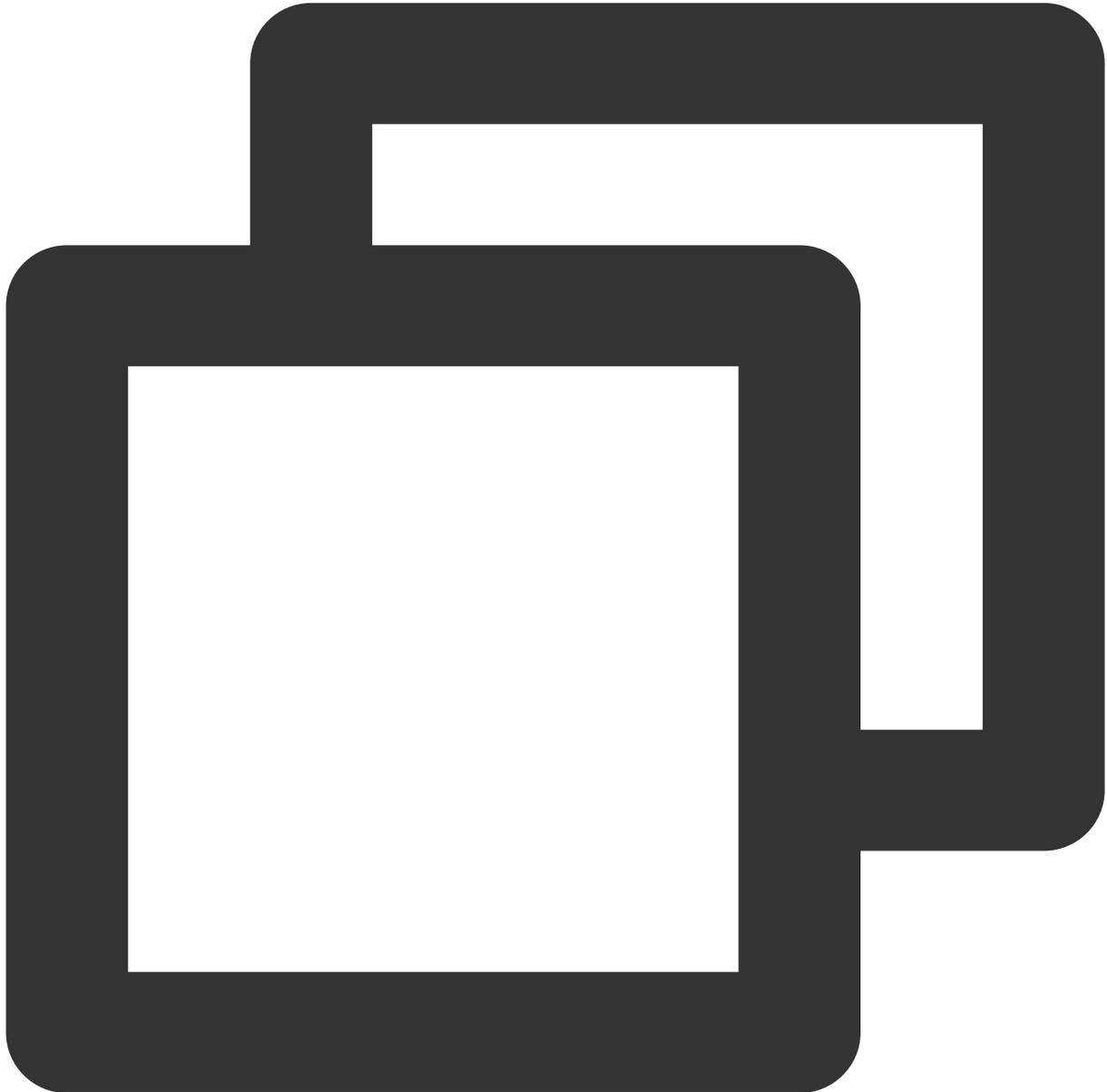
建表语句如下：



```
CREATE TABLE table2
(
  event_day DATE,
  siteid INT DEFAULT '10',
  citycode SMALLINT,
  username VARCHAR(32) DEFAULT '',
  pv BIGINT SUM DEFAULT '0'
)
AGGREGATE KEY(event_day, siteid, citycode, username)
PARTITION BY RANGE(event_day)
(
```

```
PARTITION p201706 VALUES LESS THAN ('2017-07-01'),  
PARTITION p201707 VALUES LESS THAN ('2017-08-01'),  
PARTITION p201708 VALUES LESS THAN ('2017-09-01')  
)  
DISTRIBUTED BY HASH(siteid) BUCKETS 10  
PROPERTIES("replication_num" = "1");
```

表建完之后，可以查看 `example_db` 中表的信息：



```
MySQL> SHOW TABLES;  
+-----+  
| Tables_in_example_db |
```

```

+-----+
| table1          |
| table2          |
+-----+
2 rows in set (0.01 sec)

MySQL> DESC table1;
+-----+-----+-----+-----+-----+-----+
| Field          | Type                | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| siteid         | int(11)             | Yes  | true | 10       |       |
| citycode       | smallint(6)         | Yes  | true | N/A      |       |
| username       | varchar(32)         | Yes  | true |          |       |
| pv             | bigint(20)          | Yes  | false| 0        | SUM   |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

MySQL> DESC table2;
+-----+-----+-----+-----+-----+-----+
| Field          | Type                | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| event_day      | date                | Yes  | true | N/A      |       |
| siteid         | int(11)             | Yes  | true | 10       |       |
| citycode       | smallint(6)         | Yes  | true | N/A      |       |
| username       | varchar(32)         | Yes  | true |          |       |
| pv             | bigint(20)          | Yes  | false| 0        | SUM   |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
    
```

注意：

上述表通过设置 `replication_num` 建的都是单副本的表，腾讯云数据仓库 TCHouse-D 建议用户采用默认的 3 副本设置，以保证高可用。

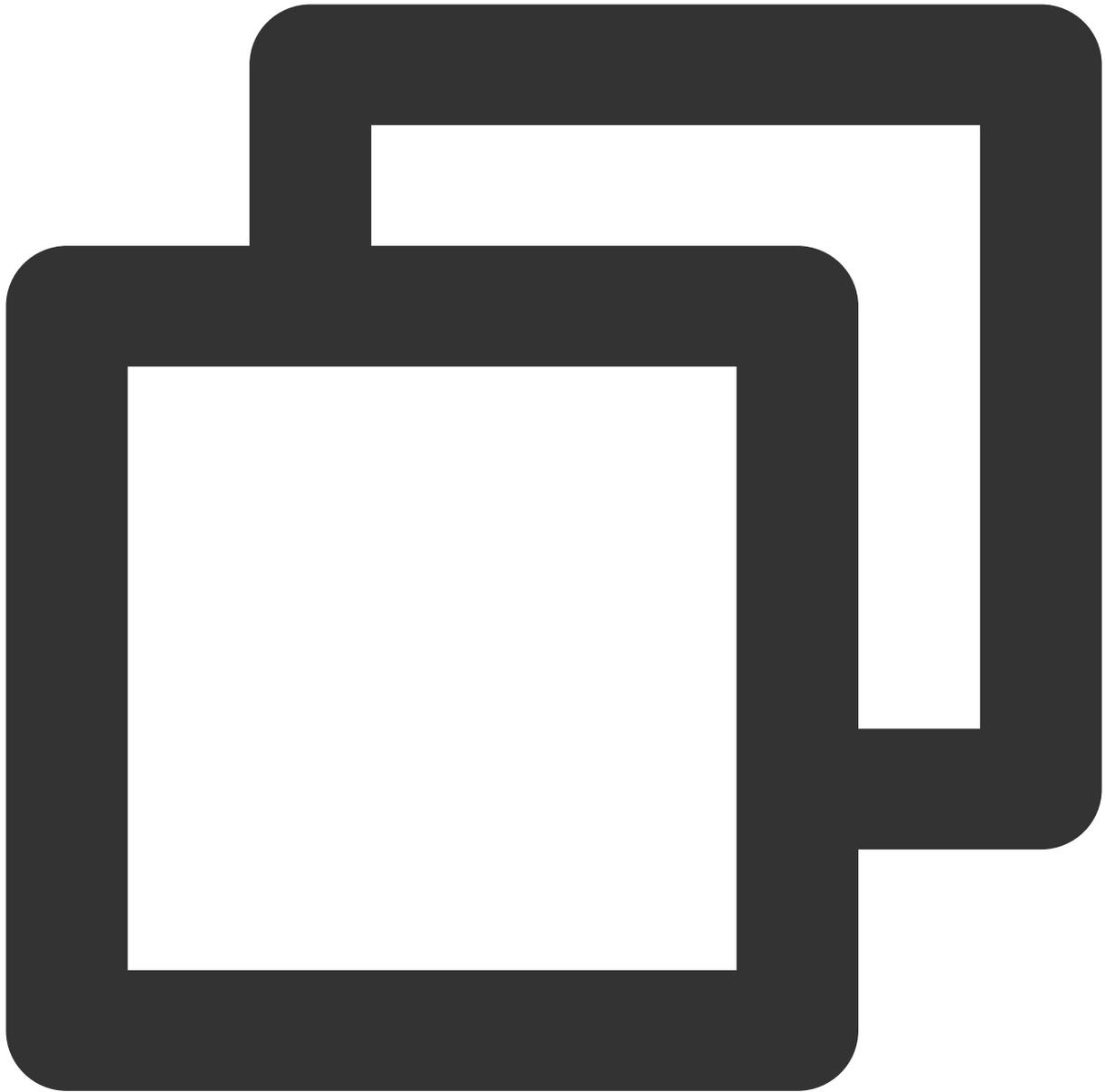
步骤3：数据导入

腾讯云数据仓库 TCHouse-D 支持多种数据导入方式。具体可以参阅[数据导入](#)。这里给出流式导入和 Broker 导入的示例。

流式导入

流式导入通过 HTTP 协议向 TCHouse-D 传输数据，可以不依赖其他系统或组件直接导入本地数据。详细语法帮助可以参阅 `HELP STREAM LOAD;`。

示例1：以 "table1_20170707" 为 Label，使用本地文件 `table1_data` 导入 `table1` 表。



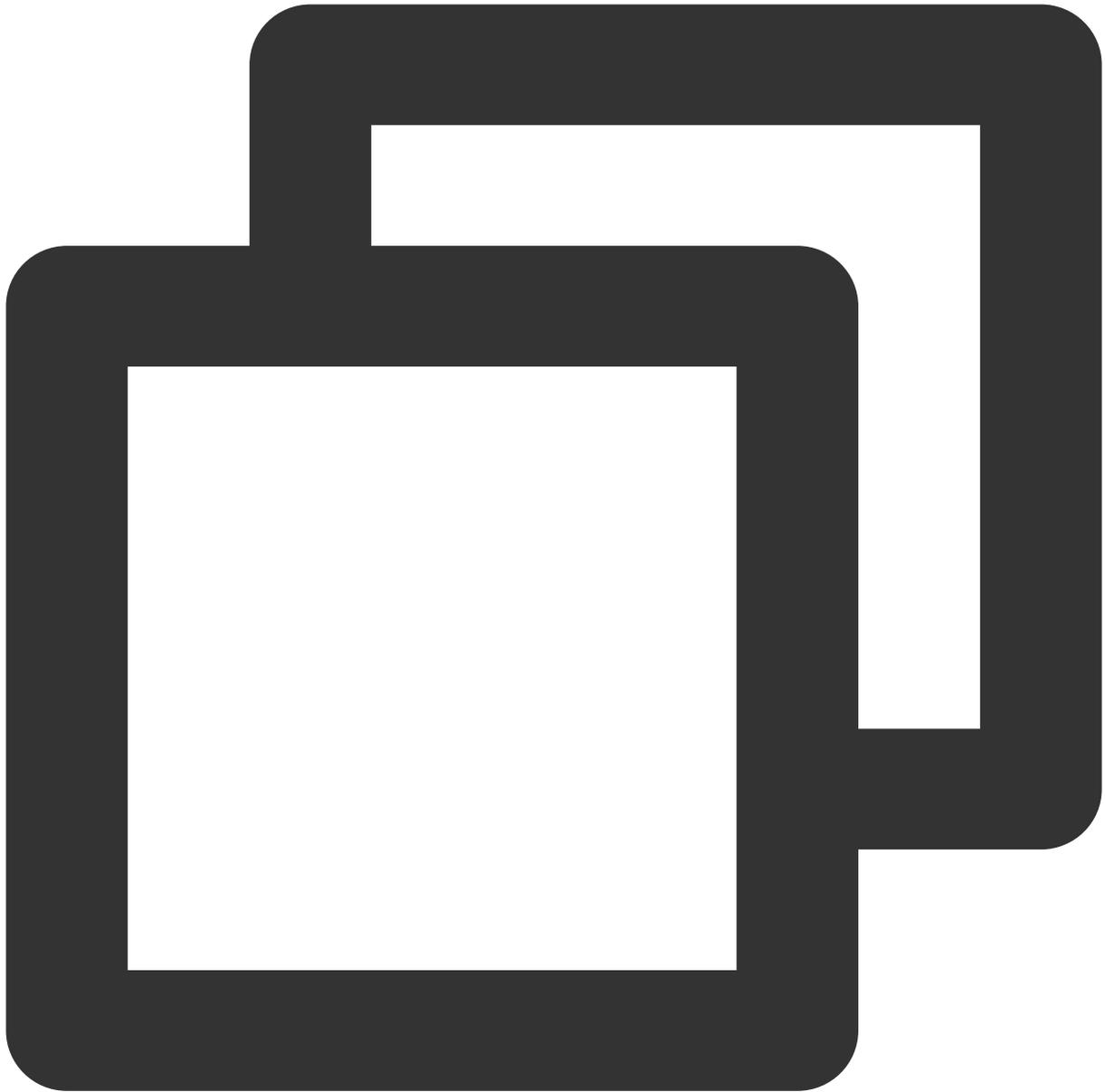
```
curl --location-trusted -u test:test_passwd -H "label:table1_20170707" -H "column_s
```

注意：

FE_HOST 是任一 FE 所在节点 IP，8030 为 fe.conf 中的 http_port。

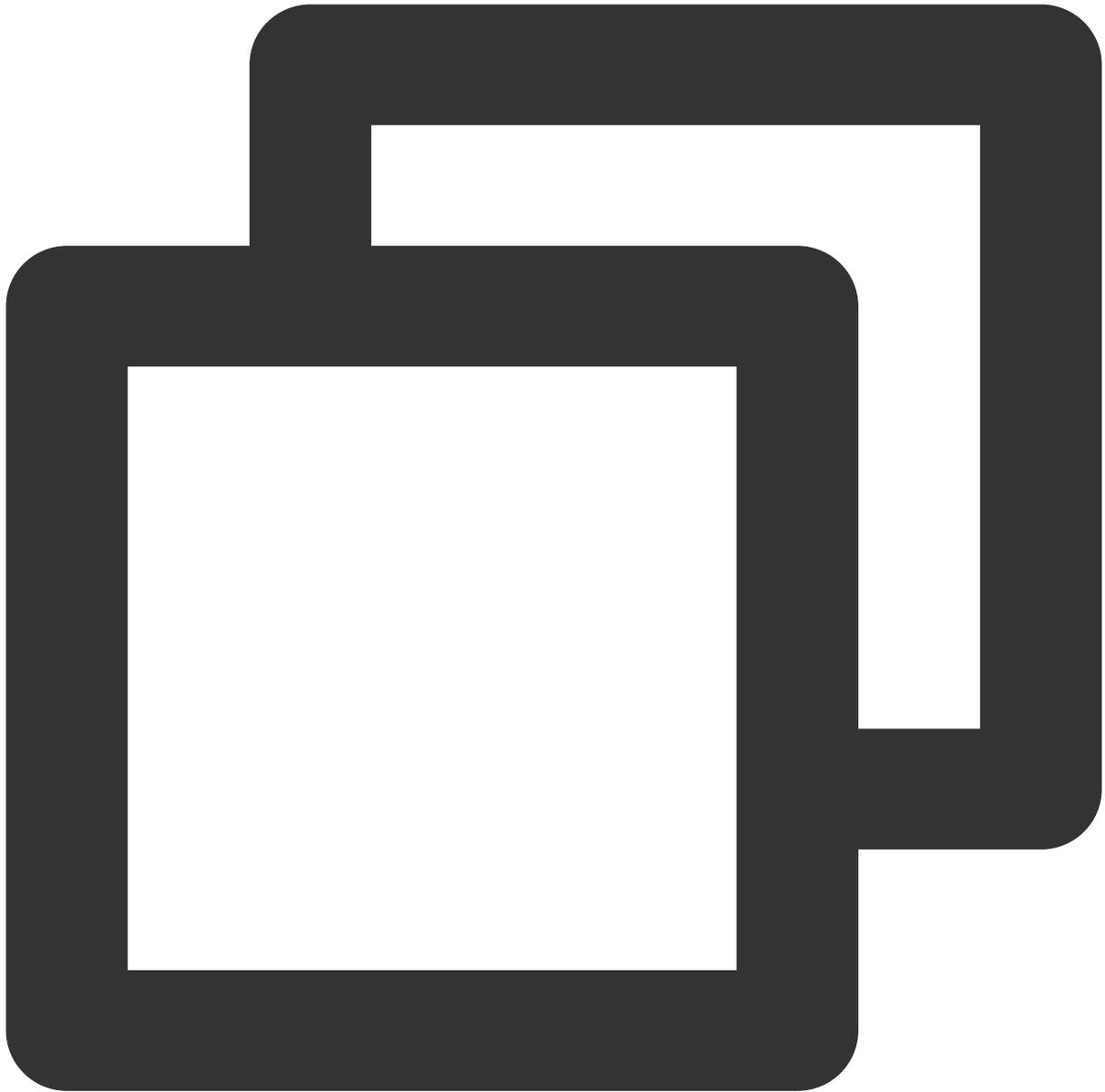
可以使用任一 BE 的 IP，以及 be.conf 中的 webserver_port 进行导入。如：BE_HOST:8040。

本地文件 table1_data 以，作为数据之间的分隔，具体内容如下：



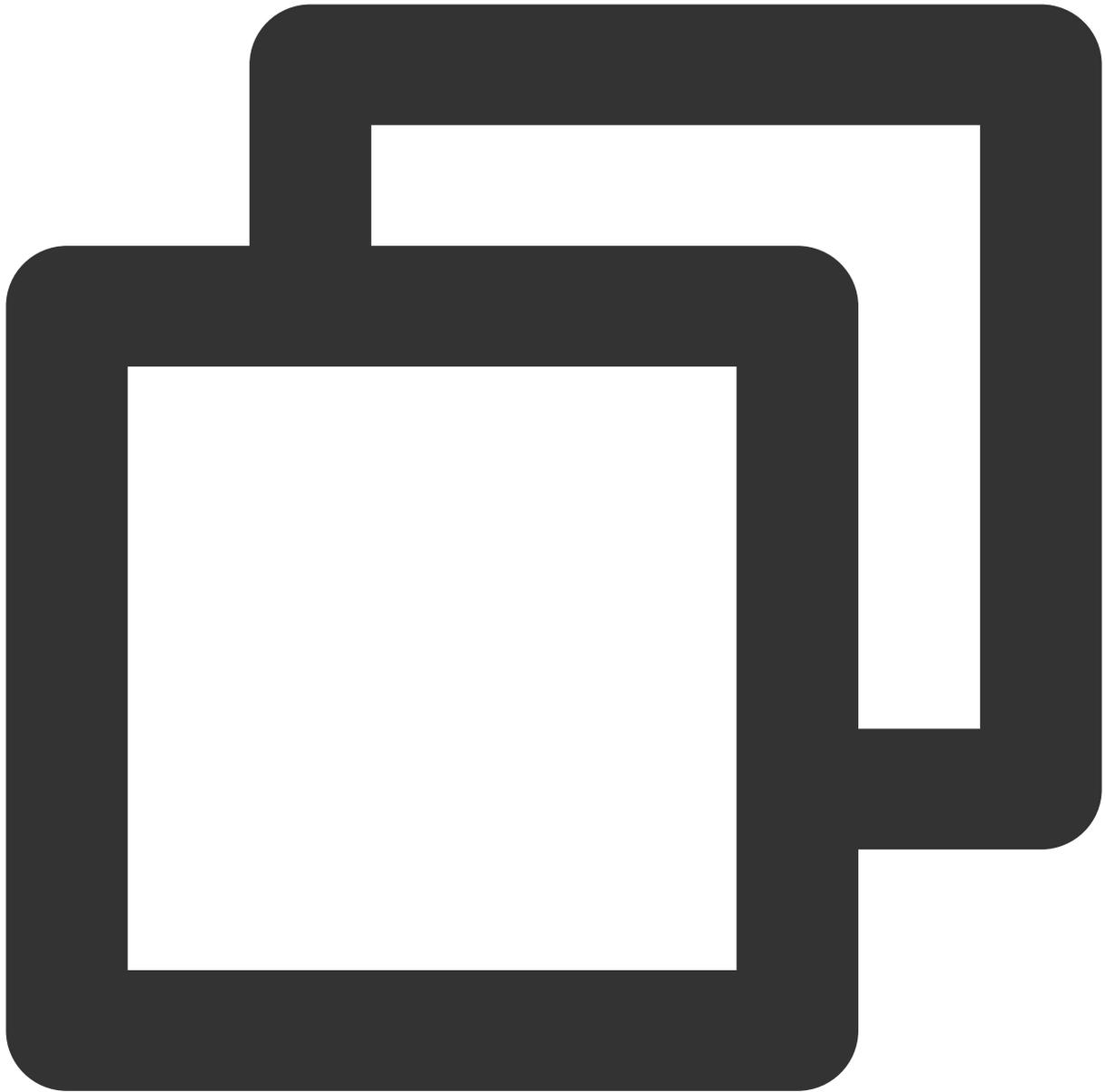
```
1,1,jim,2  
2,1,grace,2  
3,2,tom,2  
4,3,bush,3  
5,3,helen,3
```

示例2：以 "table2_20170707" 为 Label，使用本地文件 table2_data 导入 table2 表。



```
curl --location-trusted -u test:test -H "label:table2_20170707" -H "column_separato  
http://127.0.0.1:8030/api/example_db/table2/_stream_load
```

本地文件 `table2_data` 以 `|` 作为数据之间的分隔，具体内容如下：



```
2017-07-03|1|1|jim|2  
2017-07-05|2|1|grace|2  
2017-07-12|3|2|tom|2  
2017-07-15|4|3|bush|3  
2017-07-12|5|3|helen|3
```

注意：

采用流式导入建议文件大小限制在 10GB 以内，过大的文件会导致失败重试代价变大。

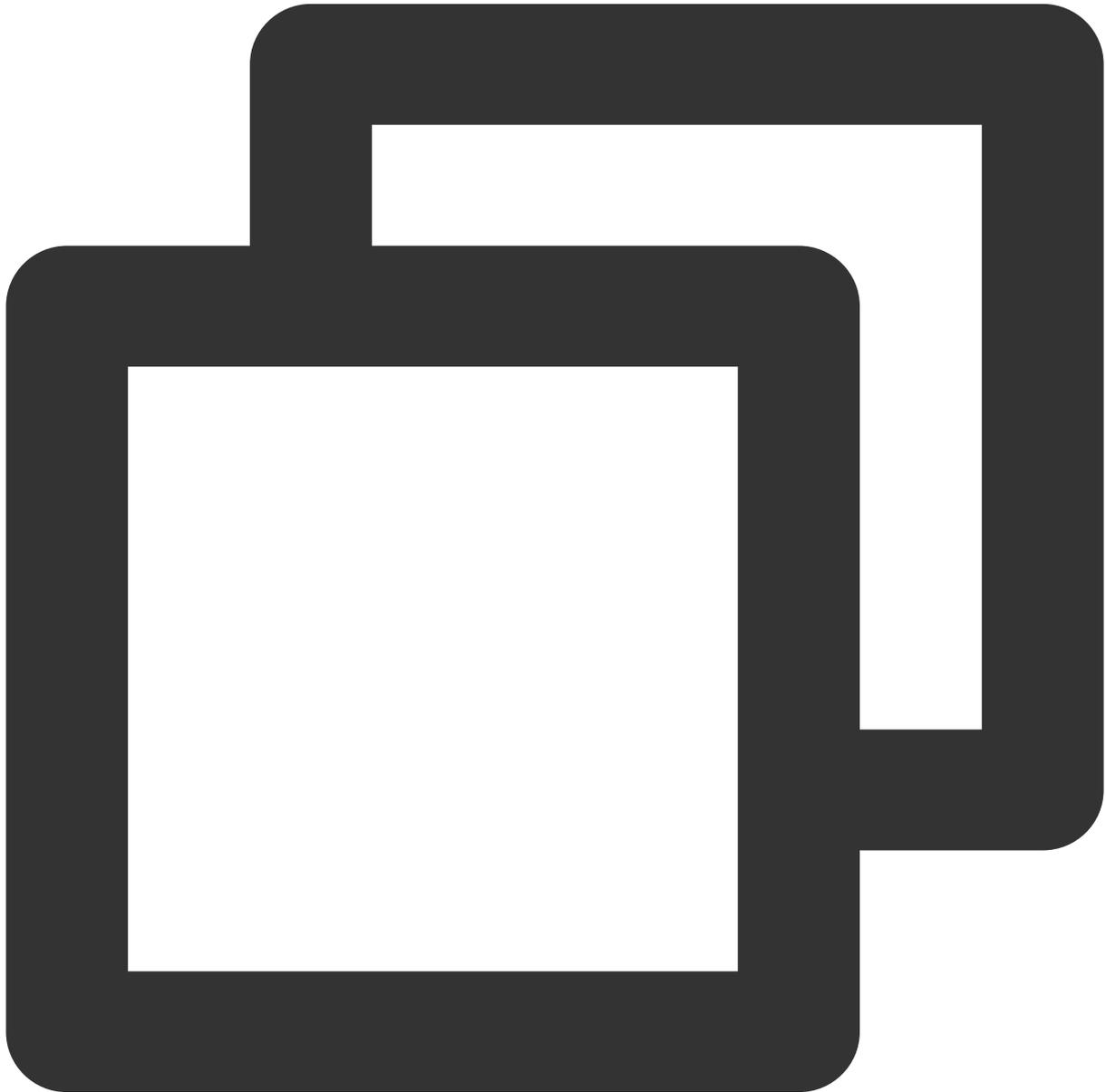
每一批导入数据都需要取一个 Label，Label 最好是一个和一批数据有关的字符串，方便阅读和管理。腾讯云数据仓库 TCHouse-D 基于 Label 保证在一个 Database 内，同一批数据只可导入成功一次。失败任务的 Label 可以重用。

流式导入是同步命令。命令返回成功则表示数据已经导入，返回失败表示这批数据没有导入。

Broker 导入

Broker 导入通过部署的 Broker 进程，读取外部存储上的数据进行导入。更多帮助请参阅 `HELP BROKER LOAD;`。

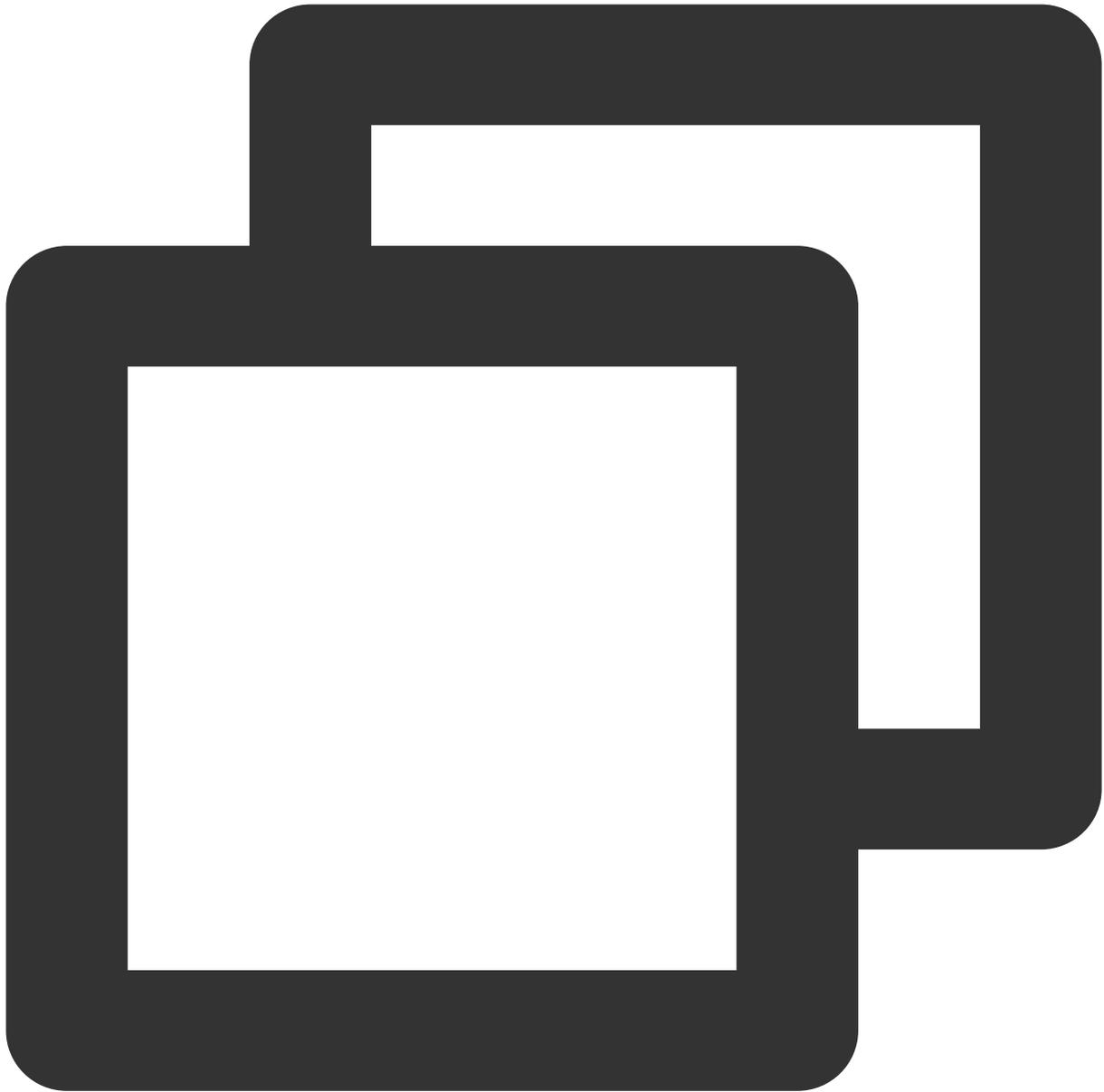
示例：以 "table1_20170708" 为 Label，将 HDFS 上的文件导入 table1 表。



```
LOAD LABEL table1_20170708
(
  DATA INFILE ("hdfs://your.namenode.host:port/dir/table1_data")
```

```
INTO TABLE table1
)
WITH BROKER hdfs
(
  "username"="hdfs_user",
  "password"="hdfs_password"
)
PROPERTIES
(
  "timeout"="3600",
  "max_filter_ratio"="0.1"
);
```

Broker 导入是异步命令。以上命令执行成功只表示提交任务成功。导入是否成功需要通过 `SHOW LOAD;` 查看。
如：



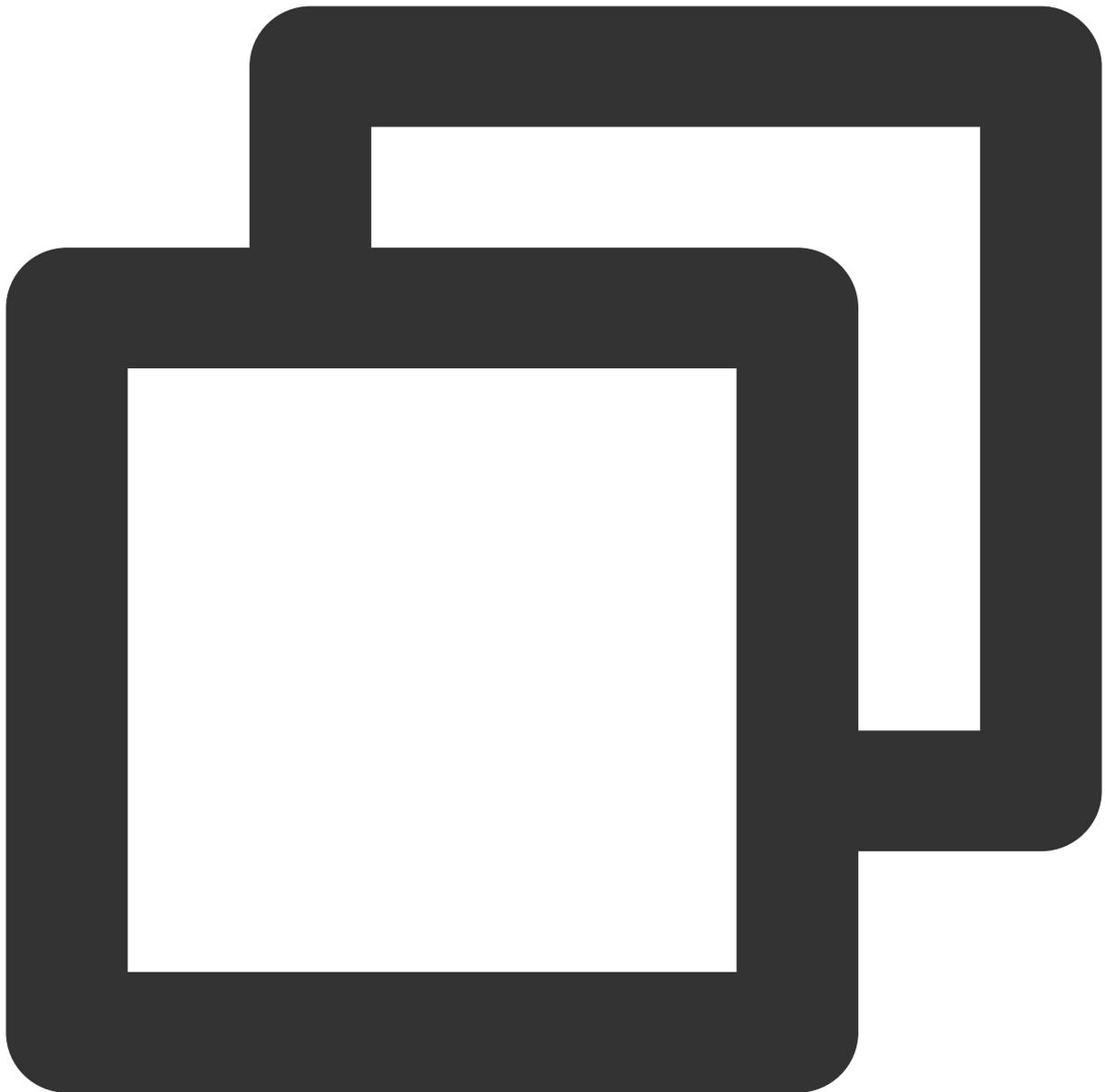
```
SHOW LOAD WHERE LABEL = "table1_20170708";
```

返回结果中，`State` 字段为 `FINISHED` 则表示导入成功。关于 `SHOW LOAD` 的更多说明，可以参阅 `HELP SHOW LOAD`；。

异步的导入任务在结束前可以取消：`CANCEL LOAD WHERE LABEL = "table1_20170708";`。

步骤4：数据查询

1. 简单查询：



```
MySQL> SELECT * FROM table1 LIMIT 3;
```

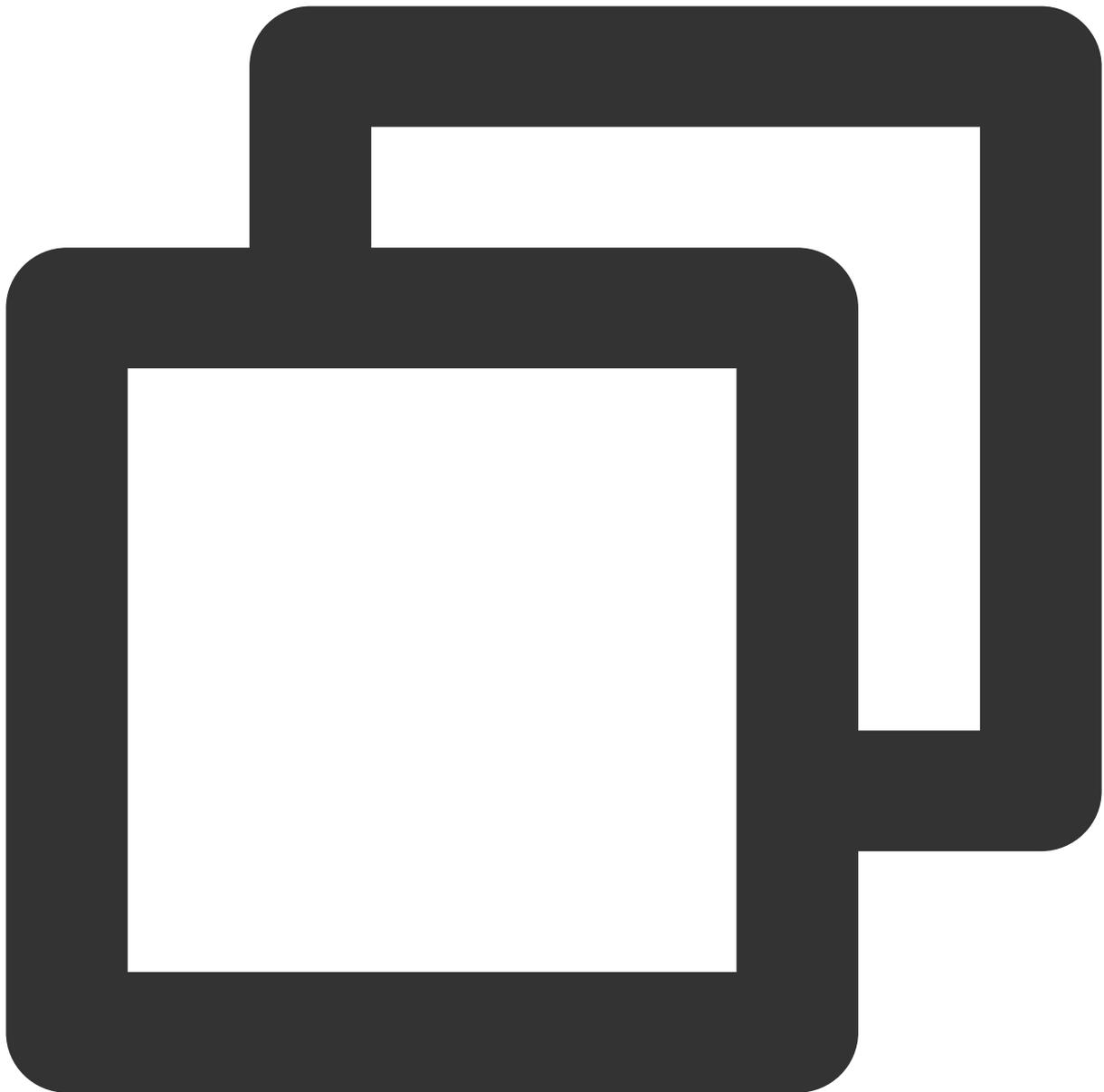
```
+-----+-----+-----+-----+
| siteid | citycode | username | pv |
+-----+-----+-----+-----+
|      2 |      1 | 'grace'  |  2 |
|      5 |      3 | 'helen'  |  3 |
|      3 |      2 | 'tom'    |  2 |
+-----+-----+-----+-----+
```

```
3 rows in set (0.01 sec)
```

```
MySQL> SELECT * FROM table1 ORDER BY citycode;
```

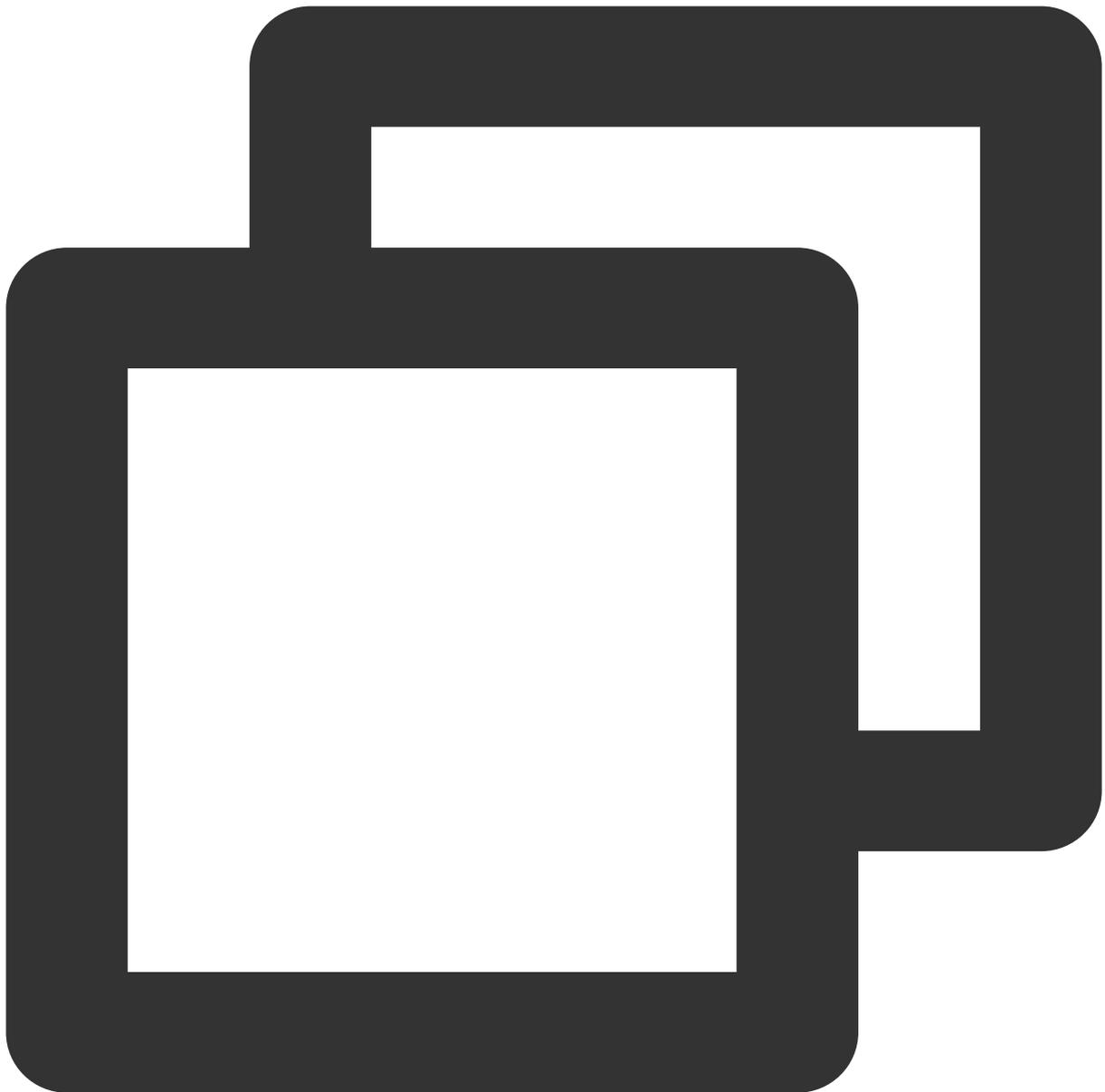
```
+-----+-----+-----+-----+
| siteid | citycode | username | pv |
+-----+-----+-----+-----+
|      2 |         1 | 'grace'  |  2 |
|      1 |         1 | 'jim'    |  2 |
|      3 |         2 | 'tom'    |  2 |
|      4 |         3 | 'bush'   |  3 |
|      5 |         3 | 'helen'  |  3 |
+-----+-----+-----+-----+
5 rows in set (0.01 sec)
```

2. Join 查询：



```
MySQL> SELECT SUM(table1.pv) FROM table1 JOIN table2 WHERE table1.siteid = table2.s
+-----+
| sum(`table1`.`pv`) |
+-----+
|                12 |
+-----+
1 row in set (0.20 sec)
```

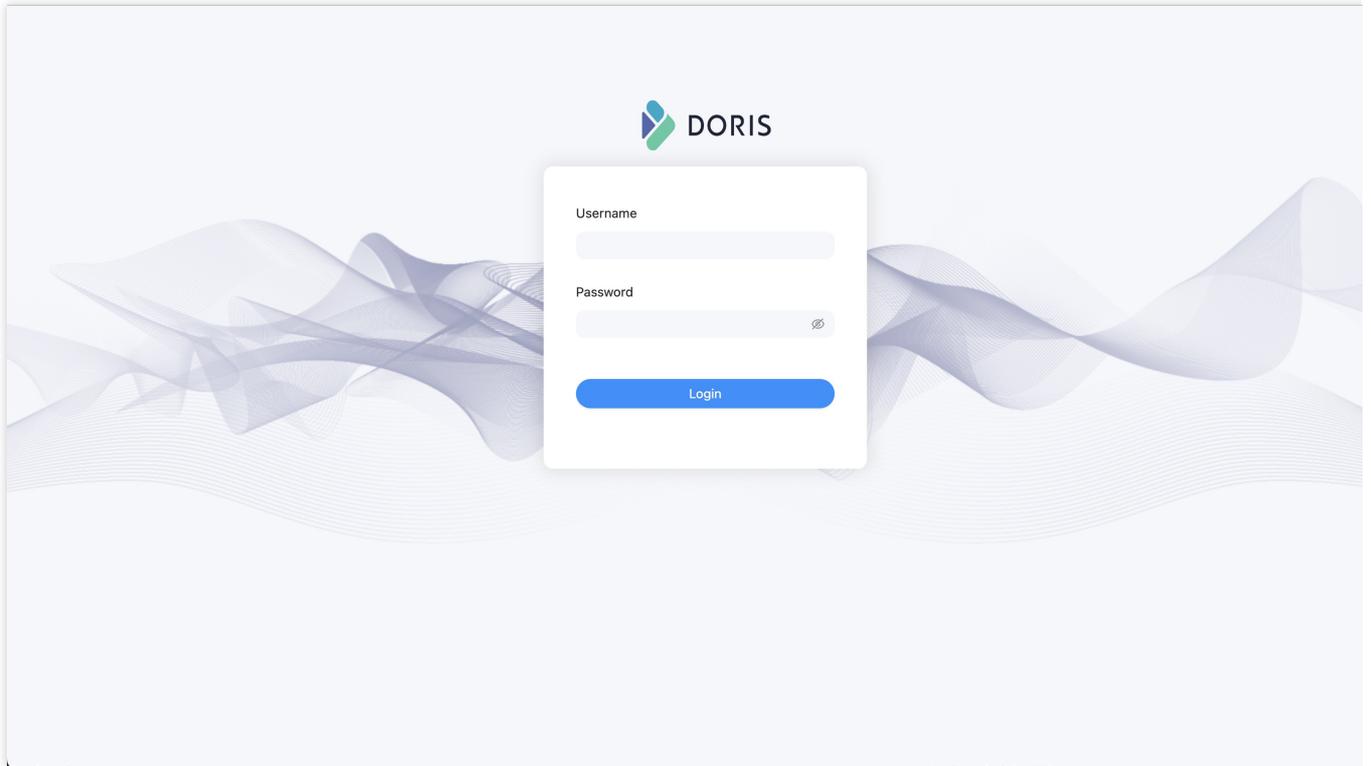
3. 子查询：



```
MySQL> SELECT SUM(pv) FROM table2 WHERE siteid IN (SELECT siteid FROM table1 WHERE
+-----+
| sum(`pv`) |
+-----+
|          8 |
+-----+
1 row in set (0.13 sec)
```

Web UI

Web UI 基于 JDBC 和 Rest API 能力实现，您可以通过在其中进行数据操作和集群管控等功能。要使用 Web UI，可在网络打通环境下输入 `https:// fe_ip:8030`，当看到如下页面时说明启动成功。



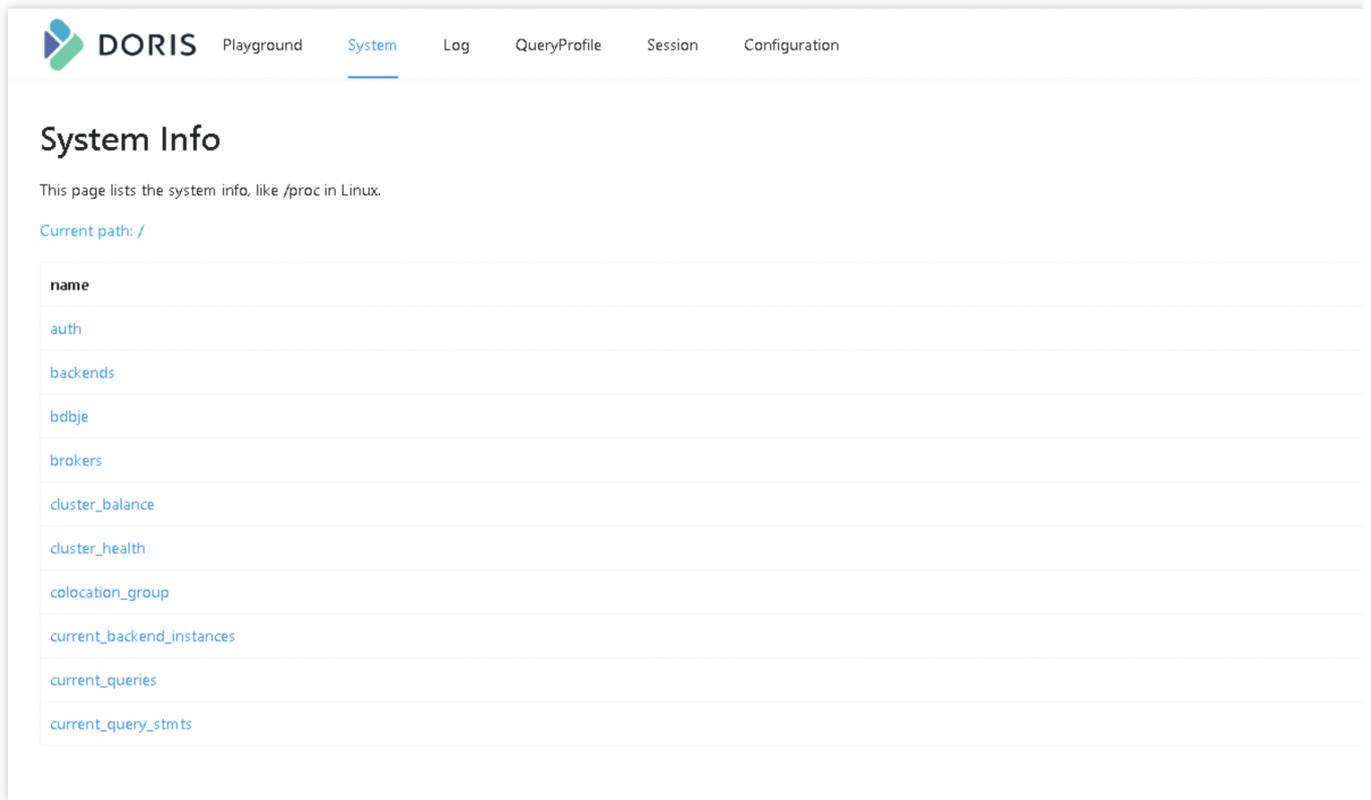
Web UI 分为 Playground、System、Log、QueryProfile、Session、Configuration 几大功能页。

Playground 中提供一个 SQL 查询编辑器，用于执行各种 SQL 命令。左侧以树形结构展示所有数据库和其下的表。双击表名可查看表的元数据和数据样例，单击 `Data Import` 后可从本地上传数据。

The screenshot shows the DORIS Playground interface. At the top, there are navigation tabs: Playground (selected), System, Log, QueryProfile, Session, and Configuration. On the left, a search bar is present above a tree view of the database structure. The main area contains an editor with a 'Format' button and a query: `select * from db1.t2`. Below the editor is an 'Execute' button. At the bottom, there are tabs for 'Table Schema', 'Data Preview' (selected), and 'Data Import'. The 'Data Preview' section shows a table with 6 columns: TABLE_CATALOG, TABLE_SCHEMA, TABLE_NAME, COLUMN_NAME, ORDINAL_POSITION, and COLUMN_DEFA. The table contains 4 rows of data.

| TABLE_CATALOG | TABLE_SCHEMA | TABLE_NAME | COLUMN_NAME | ORDINAL_POSITION | COLUMN_DEFA |
|---------------|--------------------|------------|----------------|------------------|-------------|
| | information_schema | partitions | TABLE_CATALOG | 1 | |
| | information_schema | partitions | TABLE_SCHEMA | 2 | |
| | information_schema | partitions | TABLE_NAME | 3 | |
| | information_schema | partitions | PARTITION_NAME | 4 | |

System 页面可查看集群的各种系统信息。



The screenshot shows the DORIS System Info page. At the top, there is a navigation bar with the DORIS logo and several menu items: Playground, System (which is highlighted with a blue underline), Log, QueryProfile, Session, and Configuration. Below the navigation bar, the page title is "System Info". A subtitle reads: "This page lists the system info, like /proc in Linux." Below this, it says "Current path: /". The main content area is a table with a single column titled "name". The table lists the following system info items: auth, backends, bdbje, brokers, cluster_balance, cluster_health, colocation_group, current_backend_instances, current_queries, and current_query_stmts.

| name |
|---------------------------|
| auth |
| backends |
| bdbje |
| brokers |
| cluster_balance |
| cluster_health |
| colocation_group |
| current_backend_instances |
| current_queries |
| current_query_stmts |

Log 页面提供了 FE 日志的展示与管理功能。

DORIS
Playground System **Log** QueryProfile Session Configuration

Log Configuration

Level:

Verbose Names:

Audit Names: slow_query,query,load,stream_load

Add

Delete

Log Contents

Log path is: /data/cdw/doris/fe/log/fe.warn.log

38334 bytes of log

```

2022-09-28 21:46:58,704 WARN (qtp1286393023-16618|16618) [StmtExecutionAction.executeQuery():154] failed to execute stmt
java.util.concurrent.ExecutionException: java.sql.SQLException: Socket error: (conn=81) Socket error
    at java.util.concurrent.FutureTask.report(FutureTask.java:122) ~[?:?]
    at java.util.concurrent.FutureTask.get(FutureTask.java:191) ~[?:?]
    at org.apache.doris.httpv2.rest.StmtExecutionAction.executeQuery(StmtExecutionAction.java:148) ~[doris-fe.jar:1.0-SNAPSHOT]
    at org.apache.doris.httpv2.rest.StmtExecutionAction.executeSQL(StmtExecutionAction.java:103) ~[doris-fe.jar:1.0-SNAPSHOT]
    at jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method) ~[?:?]
    at jdk.internal.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62) ~[?:?]
    at jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43) ~[?:?]
    at java.lang.reflect.Method.invoke(Method.java:566) ~[?:?]
    at org.springframework.web.method.support.InvocableHandlerMethod.doInvoke(InvocableHandlerMethod.java:205) ~[spring-web-5.3.22.jar:5.3.22]
    at org.springframework.web.method.support.InvocableHandlerMethod.invokeForRequest(InvocableHandlerMethod.java:150) ~[spring-web-5.3.22.jar:5.3.22]
                
```

QueryProfile 页面展示了记录到 Profile 的 SQL，可查看 SQL 整体以及其中各个 Instance 的执行计划及运行数据。

DORIS
Playground System Log **QueryProfile** Session Configuration

Finished Queries

This table lists the latest 100 queries

| Detail | Job ID | Query ID | User | Default Db | Sql Statement |
|-----------------------------------|--------|-----------------------------------|------|------------------------------|--|
| fc25e5792ac54798-8c08d39ef9a1bf0d | N/A | fc25e5792ac54798-8c08d39ef9a1bf0d | root | default_cluster:tpch_100g_11 | SELECT DATABASE0 |
| 2754d3a095534783-bab9937e471fc306 | N/A | 2754d3a095534783-bab9937e471fc306 | root | default_cluster:tpch_100g_11 | select l_returnflag, l_linestatus, sum(l_quantity) as sum_qty, sum(l_extendedprice) as sum_base_price, sum(l_extendedp |

Session 页面展示了当前活动的所有会话的信息。

 DORIS
Playground System Log QueryProfile Session Configuration

Session Info

This page lists the session info, there are 1 active sessions.

| Id | User | Host | Cluster | Db | Command |
|----|-------|-----------------|-----------------|----|---------|
| 80 | admin | 127.0.0.1:56628 | default_cluster | | Sleep |

Configuration 页面可查看 FE 的所有配置信息，在配置值一列中提供了过滤功能。

 DORIS
Playground System Log QueryProfile Session Configuration

Configure Info

| Name |
|---|
| sys_log_roll_interval |
| max_routine_load_job_num |
| query_colocate_join_memory_limit_penalty_factor |
| remote_fragment_exec_timeout_ms |
| max_running_rollup_job_num_per_table |
| max_allowed_in_element_num_of_delete |
| async_loading_load_task_pool_size |
| max_stream_load_timeout_second |
| max_routine_load_task_num_per_be |
| using_old_load_usage_pattern |

1-10 of 253 items