

Cloud Data Warehouse

Operation Guide

Product Documentation



Copyright Notice

©2013-2024 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

Trademark Notice



All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

Contents

Operation Guide

Cluster Management

Creation and Termination

Horizontal Scaling and Vertical Configuration Adjustment

Parameter Configuration

Monitoring and Alarming

Basic Monitoring

Cluster Monitoring

Alarm Configuration

Hot/Cold Data Tiering

Account and Authorization

Query Management

Log Search

Data Dictionary

Backup and Restore

Multi-ZooKeeper Cluster

External Data Import

Real-Time Import

Kafka Data import

Offline Import

HDFS Data Import

COS Data Import

Local File import

Configuring DDL on Cluster Feature

Data Redistribution

Scale-in and Migration

Operation Guide

Cluster Management

Creation and Termination

Last updated : 2024-01-19 16:45:29

Creating Cluster

Click **Buy Now** on the [Cloud Data Warehouse overview page](#). Alternatively, log in to the [Cloud Data Warehouse console](#), click **New Cluster** to enter the purchase page, and configure and purchase a cluster as prompted. For more information, see [Getting Started](#).

Viewing Cluster Information

After the cluster is created, go to the [Cloud Data Warehouse console](#) and click the **Cluster ID/Name** on the cluster list page to enter the cluster details page.

On the cluster details page, view basic cluster information, monitoring information, and operation log, and configure parameters.

You can click the icon next to the cluster name to edit it.

Basic Information

Cluster ID: [redacted]
Cluster Name: [redacted]
Billing Mode: Pay-as-you-go
Creation Time: 2022-02-21 10:18:38
Tag: 33321:3211 123:2222 [Modify](#)

Cluster Status

Cluster Status: Servicing

Configuration Information

Kernel Version: 21.3.9.84
High availability: High availability
Compute Node Type: High I/O
Compute Node Spec: 32-core 128 GB
Compute Node Count: 2
Compute Node Storage: LOCAL_BASIC7140GB
Mount Storage to Node: 2 * CLOUD_HSSD 1000GB

Network Information

Availability Zone: [redacted]
VPC ID: [redacted]
Subnet ID: [redacted]
Cluster Access Address (http): [redacted]
Cluster Access Address (tcp): [redacted]

Node Information

No.	Node IP	V-cluster
1	[redacted]	default_cluster
2	[redacted]	default_cluster

Total items: 2 10 / page 1 / 1 page

Viewing Monitoring Information

On the cluster details page, select the **Cluster Monitoring** tab to view monitoring metrics.

Cluster Monitoring

Main Metrics ZooKeeper Metrics

1 hour [Disable](#) Show legends

zookeeperrequest(Count) 16:47 0

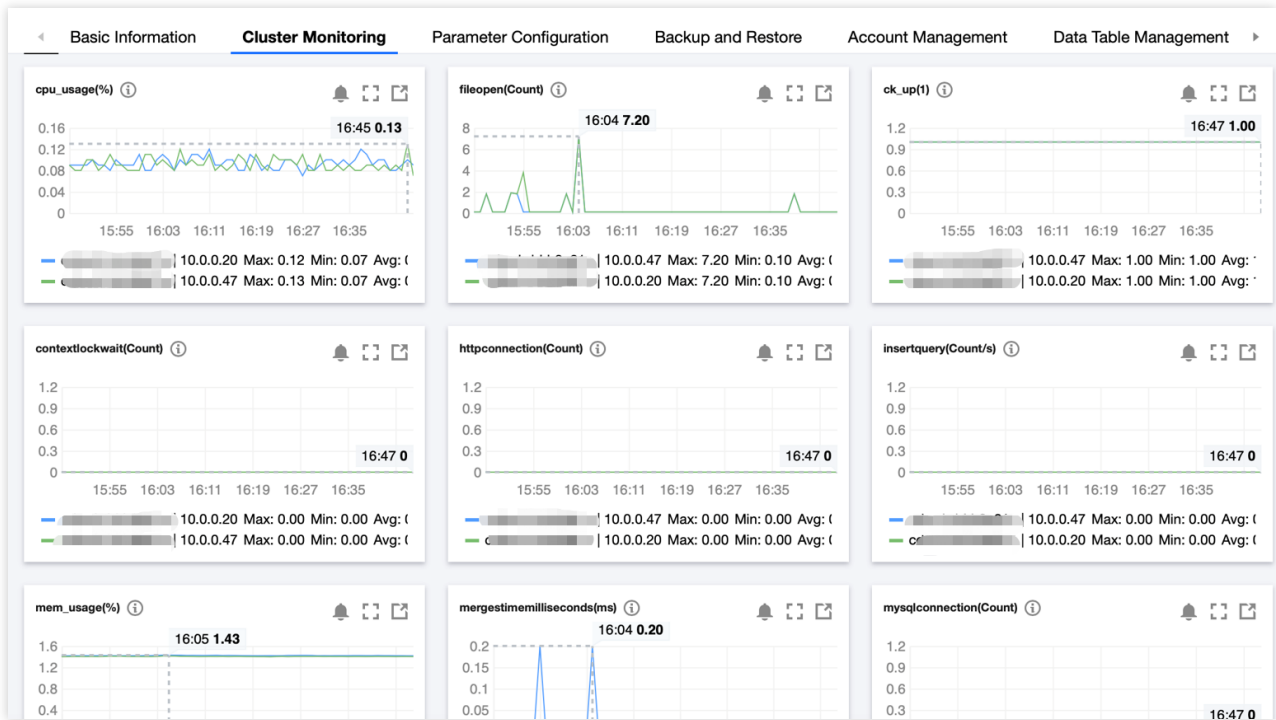
zookeeperpersession(Count) 16:47 1.00

zookeeperwatch(Count) 16:47 1.00

cpu_usage(%) 16:45 0.13

fileopen(Count) 16:04 7.20

ck_up(1) 16:47 1.00



Terminating Cluster

Select **Operation > Terminate** in the cluster list to terminate a cluster as prompted.

Pay-as-you-go clusters: resources will be released and data will be cleared in 24 hours upon cluster termination.

Terminate ✕

1

Information

>

2

Terminate

Cluster ID	Cluster Name	Spec	Node C...
[blurred]	[blurred]	32-core 128 GB...	2

I have read and agree to [Refund Rules](#)

Next: Terminate

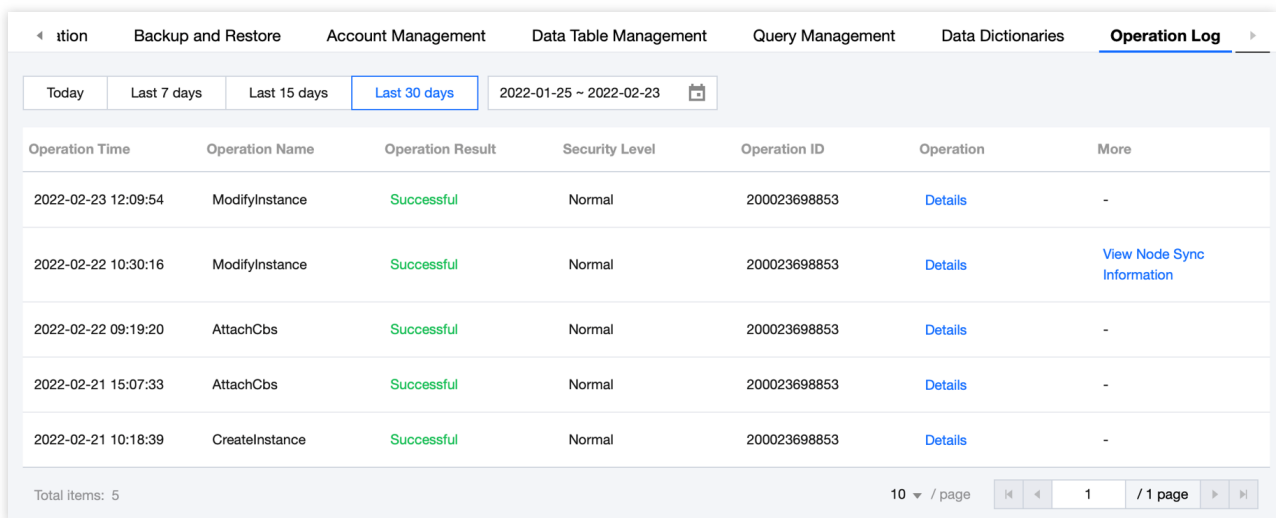
Cancel

Configuring Parameter

On the cluster details page, select the **Parameter Configuration** tab to view parameters in the config.xml file. After a parameter is modified, click **Apply to Cluster** for the modification to take effect.

Viewing Operation Log

On the cluster details page, select the **Operation Log** tab to view all operations over a specified period of time.



Operation Time	Operation Name	Operation Result	Security Level	Operation ID	Operation	More
2022-02-23 12:09:54	ModifyInstance	Successful	Normal	200023698853	Details	-
2022-02-22 10:30:16	ModifyInstance	Successful	Normal	200023698853	Details	View Node Sync Information
2022-02-22 09:19:20	AttachObs	Successful	Normal	200023698853	Details	-
2022-02-21 15:07:33	AttachObs	Successful	Normal	200023698853	Details	-
2022-02-21 10:18:39	CreateInstance	Successful	Normal	200023698853	Details	-

Total items: 5

10 / page

Horizontal Scaling and Vertical Configuration Adjustment

Last updated : 2024-01-19 16:45:29

Feature Overview

ClickHouse cluster management provides the features of horizontal scaling and vertical configuration adjustment if the current cluster size and configuration do not meet your needs.

Note:

The manipulated cluster needs to be in a stable running status.

There are no overdue payments or orders pending payment under the root account.

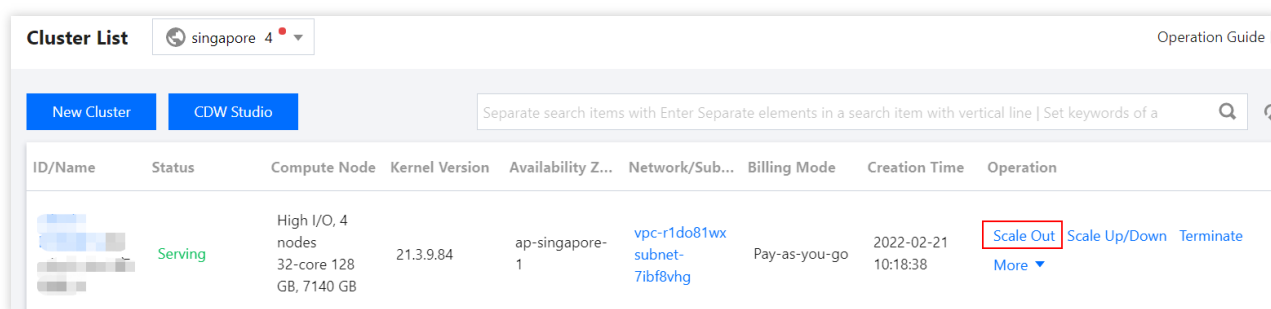
Horizontal Scaling

Note:

A horizontal scaling operation takes 5–15 minutes, during which the system is readable but unwritable. We recommend you perform it during off-peak hours.

Horizontal scaling is preferred when both data storage and query volume are increasing.

1. Log in to the [Cloud Data Warehouse console](#) and select **Operation > Scale In/Out** for the target cluster.



ID/Name	Status	Compute Node	Kernel Version	Availability Z...	Network/Sub...	Billing Mode	Creation Time	Operation
[Redacted]	Serving	High I/O, 4 nodes 32-core 128 GB, 7140 GB	21.3.9.84	ap-singapore-1	vpc-r1do81wx subnet-71bf8vhg	Pay-as-you-go	2022-02-21 10:18:38	Scale Out Scale Up/Down Terminate More ▾

2. On the horizontal scaling page, increase the node quantity to the desired value. New nodes are added to the `default_cluster` group by default or a custom group you select. You can also choose to enable sync table creation to synchronously create database tables for the new nodes. Note that you need to select a reference node in this case. If you do not enable sync table creation, the new nodes will be empty.

Note:

Currently, common database and table engines are supported for sync table creation, including MergeTree types, non-MergeTree types (such as foreign and log tables), and materialized views.

During horizontal scaling, sync table creation may fail due to various factors (for example, a table cannot be created on multiple nodes). Therefore, after the scaling is completed, you need to sync the failed tables as applicable.

Cluster Scale-Out ✕

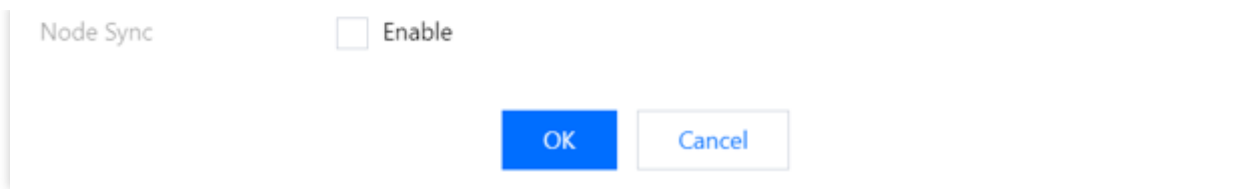
i 1. The system is readable but not writable during the scaling process.

2. The scaling is only for compute nodes. The scaling result applies to all compute nodes in the cluster.

3. You can set new nodes as empty nodes or nodes with synced tables by setting Node Sync as required.

4. You can enable Node Sync to create databases and tables (excluding system tables) as well as v_cluster accounts and permissions according to the selected IP node.

Cluster ID/Name	
Region/Availability Zone	
Billing Mode	Pay-as-you-go
High Availability	High availability
Compute Node Spec	32-core 128GB
Storage Spec	LOCAL_BASIC7140GB
Mount Storage to Node	2 * CLOUD_HSSD 1000GB
Current Nodes	4
Change To	<input type="button" value="-"/> <input type="text" value="6"/> <input type="button" value="+"/> In high availability mode, the number of nodes to change must be a multiple of 2
Cost	
v_cluster Group	<input type="text" value="default_cluster"/> New nodes will be added to the selected v_cluster group



- Click **OK** and make the payment on the confirmation page. Then, the cluster will be in the "changing" status for 5–15 minutes.
- After the scaling is completed, enter the cluster details page, switch to the **Operation Log** tab, and click **View Node Sync Information**. On the displayed page, confirm the execution result of sync table creation and create the failed tables on your own according to the corresponding SQL statement.

Vertical Configuration Adjustment

A ClickHouse cluster supports vertical configuration adjustment, including specification upgrade/downgrade and storage capacity expansion for compute nodes as well as for ZooKeeper nodes in HA clusters.

Note:

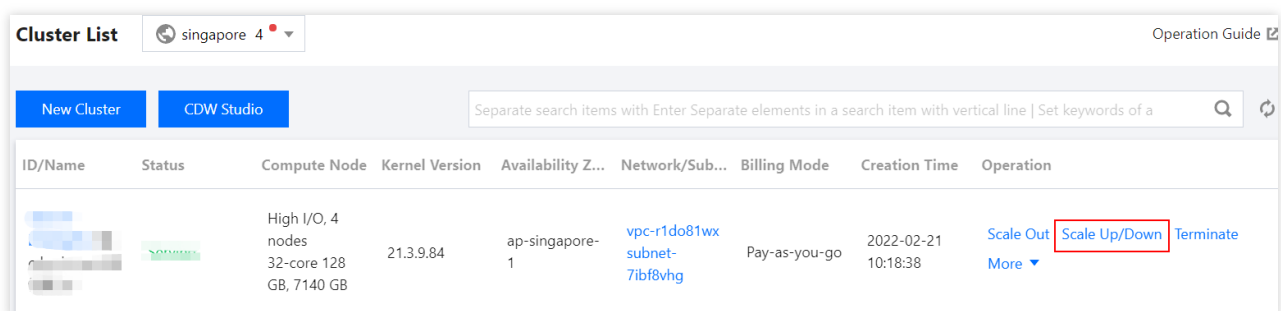
A vertical configuration adjustment takes 5–15 minutes, during which the system is unreadable and unwritable. We recommend you perform it during off-peak hours.

Vertical configuration adjustment is preferred when the node performance metrics (including ZooKeeper metrics) of a cluster cannot meet the requirements of a business scenario.

Vertical configuration adjustment is available only for standard clusters. Either compute or ZooKeeper nodes can be adjusted each time.

Note that the storage capacity of compute and ZooKeeper nodes can only be expanded but not reduced.

- Log in to the Cloud Data Warehouse console and select **Operation > Scale Up/Down** for the target cluster.



- On the page displayed, adjust the compute and storage specification as needed.

Note:

You can upgrade either the compute or storage specification or both of them at the same time, but when you downgrade the compute specification, you cannot adjust the storage specification.

Cluster Scale-Up/Down



- 1. The system is not readable or writable during scale-up.
- 2. You can increase node specs and storage specs as needed. The scaling result takes effect for all cluster nodes.
- 3. You can increase or reduce node specs, and increase storage specs.
- 4. Cost description: the cost is the single-month difference cost for monthly subscription billing, and is the hourly unit price after the scaling for pay-as-you-go billing.

Cluster ID/Name ██

Region/Availability Zone ████████████████████████████████████

Billing Mode Pay-as-you-go

High Availability High availability

Compute Node Spec 32-core 128GB

Storage Spec LOCAL_BASIC7140GB

Compute Node Count 4

Mount Storage to Node 2 * CLOUD_HSSD 1000GB

Node to Scale Compute Node ZooKeeper Node

Unable to scale up/down the compute nodes for the High I/O cluster

Compute Spec

4-core 16 GB	8-core 32 GB	16-core 64 GB
32-core 128 GB	64-core 256 GB	128-core 256 GB

Storage Spec CLOUD_HSSD - 100 + GB
A single node supports 32000 GB at most.

Cost ████████████████████

3. Click **OK** and make the payment on the confirmation page. Then, the cluster will be in the "changing" status for 5–15 minutes.

Parameter Configuration

Last updated : 2024-01-19 16:45:30

Feature Overview

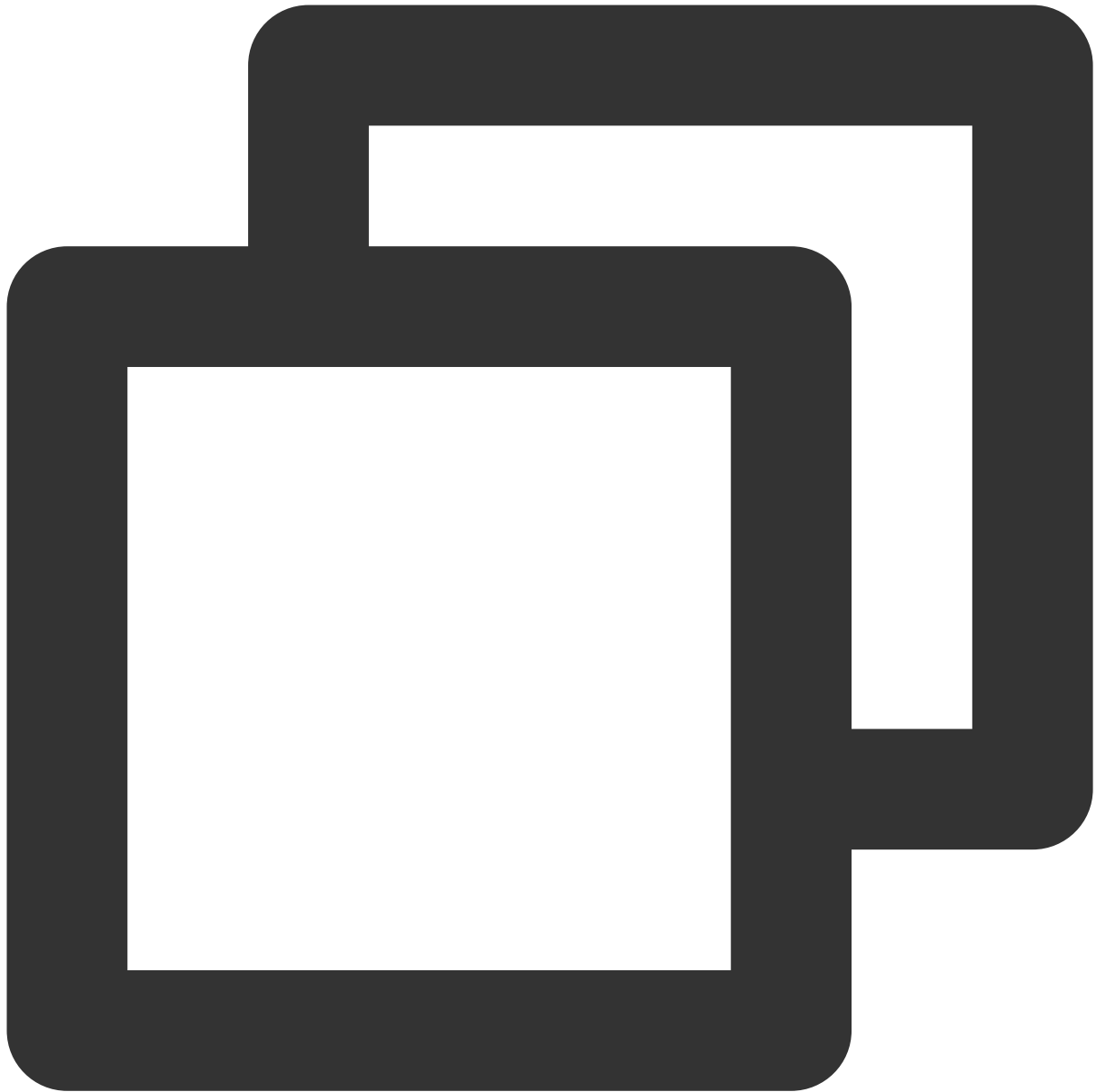
This document describes how to manage clusters in the console. You can adopt either the key-value or XML mode in the ClickHouse cluster configuration management feature based on your usage habits to modify key parameters and push the modified configuration file content to the nodes in the cluster.

Feature Description

By default, the key-value mode applies to parameter configuration for legacy clusters created before August 2021, while the XML mode applies to those created afterward. If you need to manage a legacy cluster in XML mode, [contact us](#) for assistance.

Note:

Currently, cluster parameter configuration is available in either key-value or XML mode. New clusters support the key-value mode by default, but you can also switch to the XML mode irreversibly.



Before using the following features, we recommend you contact technical support to

Directions

1. Log in to the [Cloud Data Warehouse console](#), select the target cluster in **Cluster List**, click **Name/ID** to enter the cluster details page, and click **Parameter Configuration**.

2. On the **Parameter Configuration** tab for existing clusters, parameters can be configured in key-value mode and come from the `config.xml` file. After a parameter is modified, click **Apply to Cluster** for the modification to take effect.

The screenshot shows the 'Parameter Configuration' tab for cluster `cdwch-zmpvhzpo`. The interface includes a navigation bar with tabs for 'Basic Information', 'Cluster Monitoring', 'Parameter Configuration', 'Backup and Restore', and 'Account'. Below the navigation bar, there are two sub-tabs: 'Modify Configuration' (highlighted with a red box) and 'Modification History'. The 'Modify Configuration' sub-tab contains a 'Configuration File' list on the left with items `config.xml`, `metrika.xml`, and `users.xml`. On the right, there are buttons for 'Apply to Cluster' (highlighted with a red box) and 'Reset All Changes', along with a 'Changed items: 0' indicator. Below these buttons is a table of parameters:

Parameter	Reference Value	Current Running Value
<code>max_connections</code> ⓘ	4096	4096 ✎
<code>keep_alive_timeout</code> ⓘ	120	120 ✎
<code>max_concurrent_queries</code> ⓘ	100	100 ✎
<code>uncompressed_cache_size</code> ⓘ	8589934592	8589934592 ✎
<code>mark_cache_size</code> ⓘ	5368709120	5368709120 ✎
<code>builtin_dictionaries_reload_interval</code> ⓘ	3600	3600 ✎
<code>max_session_timeout</code> ⓘ	3600	3600 ✎

3. For new clusters or existing clusters with the parameter configuration mode switched to XML, the **Parameter Configuration** tab displays parameter configurations in XML mode. Currently, you can view and edit parameters in `config.xml`, `metrika.xml`, and `users.xml` files as shown below.

← cdwch-273pc58g

Basic Information Cluster Monitoring **Parameter Configuration** Backup and Restore Account

Modify Configuration Modification History

Note: When you modify configuration items in the console, if a value contains special characters such as &, the console will not escape them.

Configuration File

- config.xml
- metrika.xml
- users.xml

CLIKHOUSE: config.xml Takes effect only for this cluster

```
1 <?xml version="1.0"?>
2 <!--
3 NOTE: User and query level settings are set up in "users.xml" file
4 -->
5 <yandex>
6   <access_control_path>/data/clickhouse/clickhouse-server/access/
7   <logger>
8     <!-- Possible levels: https://github.com/pocoproject/poco/blob/master/Foundation/include/Poco/Logger.h#L105 -->
9     <level>trace</level>
10    <log>/data/clickhouse/clickhouse-server/logs/clickhouse-server
11    <errorlog>/data/clickhouse/clickhouse-server/logs/clickhouse-server
12    <size>100M</size>
13    <count>10</count>
14    <!-- <console>1</console> -->
15    <!-- Default behavior is autodetection (log to console if not set) -->
16  </logger>
17  <!--display_name>production</display_name-->
18  <!-- It is the name that will be shown in the client -->
```

Save Configuration Reset

You can edit a configuration file and click **Save Configuration**.

Save Configuration

Configuration Comparison

[Details](#)

Configuration File

CLICKHOUSE: users.xml

Dimension

Cluster

Reason for Modification

Optional, enter the reason for the

Save and Deliver

Cancel

You can view the modification through **Details**.

Version Comparison

<pre> 1 <?xml version="1.0" encoding="UTF-8"?> 2 <yandex> 3 <clickhouse_remote_servers> 4 <default_cluster> 5 <shard> 6 <internal_replication>>false</internal_ 7 <replica> 8 <host>9.0.16.2</host> 9 <port>9000</port> 10 </replica> 11 </shard> 12 <shard> 13 <internal_replication>>false</internal_ 14 <replica> 15 <host>9.0.16.4</host> 16 <port>9000</port> 17 </replica> 18 </shard> 19 </default_cluster> 20 </clickhouse_remote_servers> 21 <zookeeper_servers> 22 <node> 23 <host>9.0.16.2</host> 24 <port>2181</port> 25 </node> 26 </zookeeper_servers> 27 </yandex> </pre>	<pre> 1 <?xml version="1.0" e 2 <yandex> 3 <clickhouse_remot 4 <default_clus 5 <shard> 6 <inte 7 <repl 8 < 9 < 10 </rep 11 </shard> 12 13 14 15 16 17 18 19 20 </yandex> </pre>
---	--

After confirming that the modification is correct, click **Save and Deliver** to commit it to the nodes in the cluster.

4. The cluster will enter the **Changing** status, and the prompt "Configuration has changed. Please restart" will be displayed. Determine whether a restart is needed according to the specific modification, and if so, click **Restart** in the top-right corner.

← cdwch-273pc58g
Basic Information Cluster Monitoring **Parameter Configuration** Backup and Restore Account

Modify Configuration
Modification History

ⓘ Note: When you modify configuration items in the console, if a value contains special characters such as &, the console will not esc

Configuration has changed. Please restart

config.xml ⓘ

metrika.xml

users.xml

CLICKHOUSE: config.xml
Takes effect only for this cluster

```

1  <?xml version="1.0"?>
2  <!--
3  | NOTE: User and query level settings are set up in "users.xml" fil
4  -->
5  <yandex>
6  |   <access_control_path>/data/clickhouse/clickhouse-server/access/
7  |   <logger>
8  |   | <!-- Possible levels: https://github.com/pocoproject/poco/b
9  |   | Foundation/include/Poco/Logger.h#L105 -->
10 |   |   <level>trace</level>
11 |   |   <log>/data/clickhouse/clickhouse-server/logs/clickhouse-ser
12 |   |   <errorlog>/data/clickhouse/clickhouse-server/logs/clickhous
13 |   |   <size>1000M</size>
14 |   |   <count>10</count>
15 |   |   <!-- <console>1</console> -->
16 |   |   <!-- Default behavior is autodetection (log to console if n
17 |   |   -->
18 |   </logger>
19 |   <!--display_name>production</display_name-->
20 |   <!-- It is the name that will be shown in the client -->
21 | </yandex>
22 </xml>
```

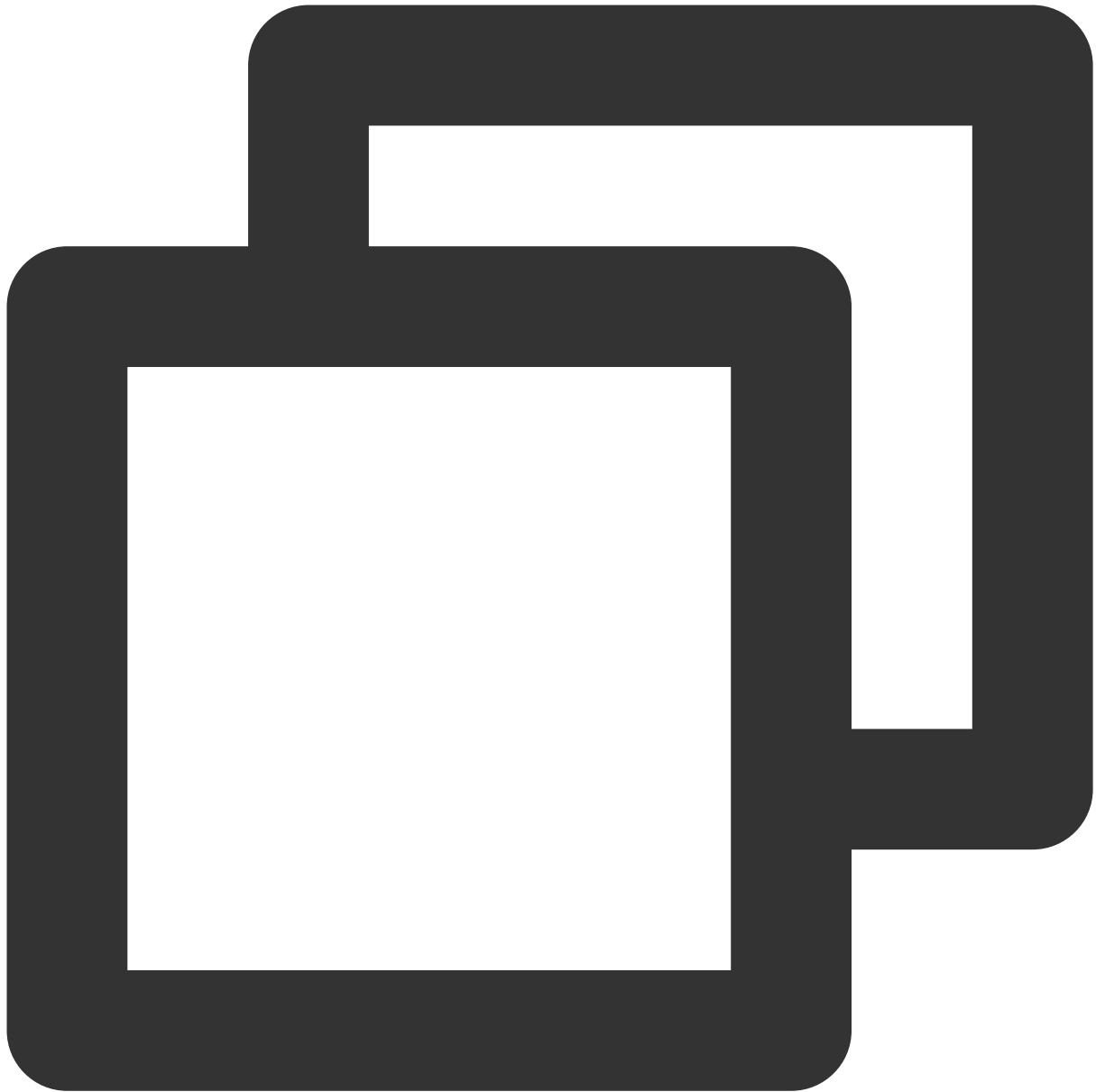
Save Configuration
Reset

5. After the modification is committed, you can view the modification history of the configuration file and the comparison details of each modification in **Modification History**.

Notes

Configuration items should not be added to the configuration file if they are not supported by the ClickHouse kernel, so as to avoid startup failures or other issues. For more information on supported configurations, see [Server Settings](#).

-In the `metrika.xml` file in XML mode, you can only modify the following configuration items and content.



```
<yandex>
  <clickhouse_remote_servers>
    <default_cluster>
      <shard>
        <internal_replication></internal_replication>
        <replica>
          <host></host>
          <second_host></second_host>
          <port></port>
          <user></user>
          <password></password>
```

```
        </replica>
    </shard>
</default_cluster>
<new_cluster_1>
    <shard>
        <internal_replication></internal_replication>
        <replica>
            <host></host>
            <second_host></second_host>
            <port></port>
            <user></user>
            <password></password>
        </replica>
    </shard>
</new_cluster_1>
</clickhouse_remote_servers>
<zookeeper-servers>
    <node>
        <host></host>
        <port></port>
    </node>
</zookeeper-servers>
</yandex>
```

Here, `new_cluster_1` indicates the name of a new cluster supported and `second_host` is an enhancement parameter for the ClickHouse kernel (v20.8.7.15 only for the time being).

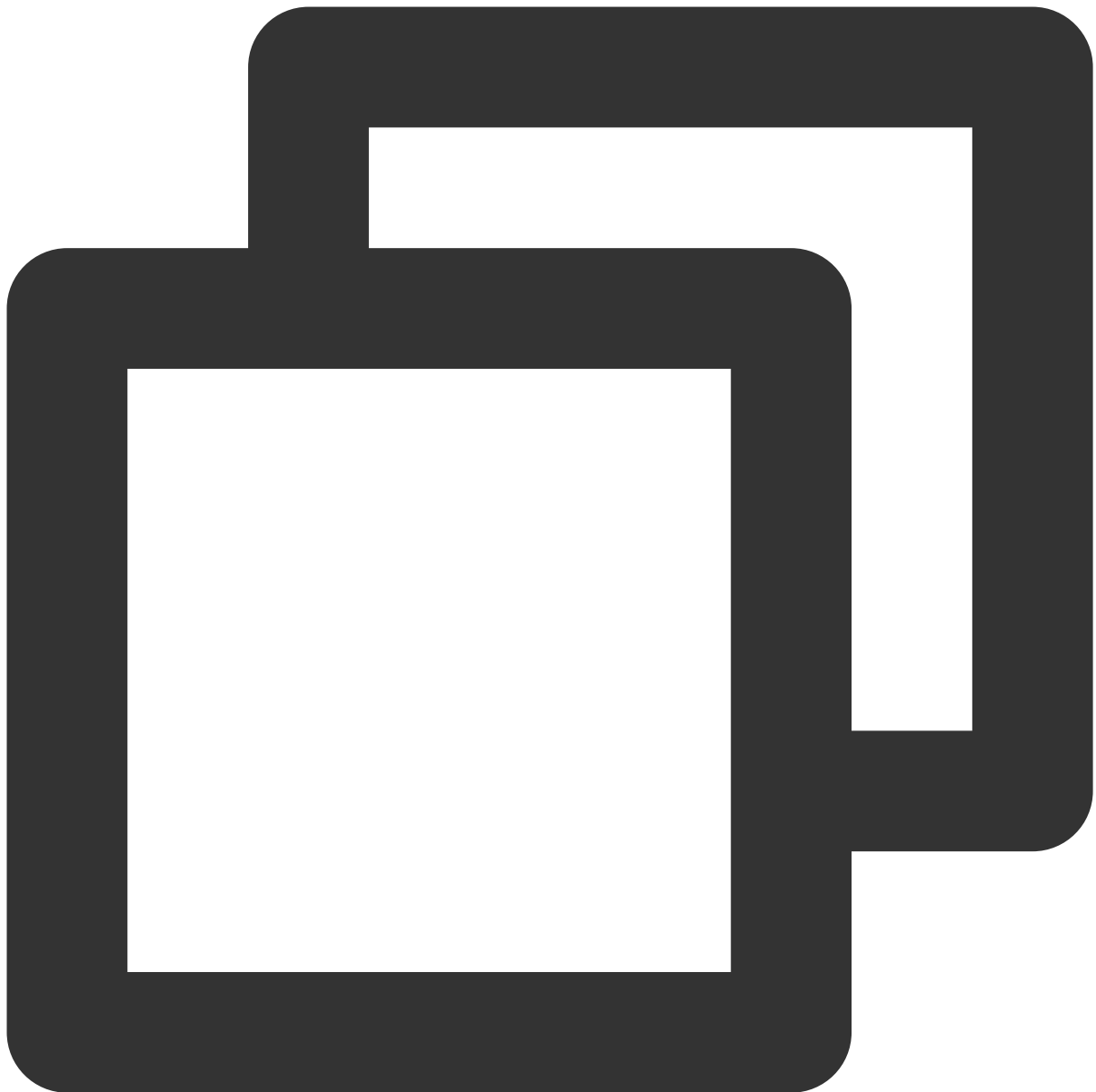
The port cannot be modified, and the host IP address cannot be set to other IP addresses or those of other clusters.

There are two IP addresses for one node: one is the host IP in the `metrika.xml` file generated by default, and the other can be found in **Basic Information > Network Information**. The former is used for backend management, while the latter for user access. IP addresses in `metrika.xml` should be those in non-user VPCs.

We recommend you use only `metrika.xml` to maintain the above information related to clusters and ZooKeeper.

If unsupported configuration items are added to `metrika.xml`, you may face the risk of invalidity or loss even if the modification is committed successfully. More configuration items will be supported in `metrika.xml` in the future, which will be described in the documentation accordingly.

In addition, you should not modify ClickHouse management and control information such as port numbers and log paths in `config.xml`, so as to avoid possible operation and management exceptions. We recommend you only modify operation configuration items such as those related to performance, compression algorithms, and MergeTree, but do not modify the following items:



```
<yandex>
  <logger>
    <level>trace</level>
    <log>/data/clickhouse/clickhouse-server/logs/clickhouse-server.log</log>
    <errorlog>/data/clickhouse/clickhouse-server/logs/clickhouse-server.err.log
    <size>1000 MB</size>
    <count>10</count>
  </logger>
  <http_port>8123</http_port>
  <tcp_port>9000</tcp_port>
  <mysql_port>9004</mysql_port>
```

```
<interserver_http_port>9009</interserver_http_port>
<path>/data/clickhouse/clickhouse-server/</path>
<tmp_path>/data/clickhouse/clickhouse-server/tmp/</tmp_path>
<user_files_path>/data/clickhouse/clickhouse-server/user_files/</user_files_path>
<users_config>users.xml</users_config>
<timezone>Asia/Shanghai</timezone>
<mlock_executable>>true</mlock_executable>
<remote_servers incl="clickhouse_remote_servers" />
<zookeeper incl="zookeeper-servers" optional="true" />
<macros incl="macros" optional="true" />
<prometheus>
  <endpoint>/metrics</endpoint>
  <port>9363</port>
  <metrics>true</metrics>
  <events>true</events>
  <asynchronous_metrics>true</asynchronous_metrics>
</prometheus>
<query_log>
  <database>system</database>
  <table>query_log</table>
  <partition_by>toYYYYMM(event_date)</partition_by>
  <flush_interval_milliseconds>7500</flush_interval_milliseconds>
</query_log>
<trace_log>
  <database>system</database>
  <table>trace_log</table>
  <partition_by>toYYYYMM(event_date)</partition_by>
  <flush_interval_milliseconds>7500</flush_interval_milliseconds>
</trace_log>
<query_thread_log>
  <database>system</database>
  <table>query_thread_log</table>
  <partition_by>toYYYYMM(event_date)</partition_by>
  <flush_interval_milliseconds>7500</flush_interval_milliseconds>
</query_thread_log>
<compression incl="clickhouse_compression">
</compression>
<distributed_ddl>
  <path>/clickhouse/task_queue/ddl</path>
</distributed_ddl>
<listen_host>0.0.0.0</listen_host>
<zookeeper incl="zookeeper-servers" optional="true" />
<macros incl="macros" optional="true" />
<include_from>/etc/clickhouse-server/metrika.xml</include_from>
</yandex>
```

Monitoring and Alarming

Basic Monitoring

Last updated : 2024-01-19 16:45:30

Basic Monitoring

Description

ClickHouse provides 46 service and performance monitoring metrics for you to stay up to date with cluster status. You can configure metrics to get real-time alarms for fast response to problems.

Cluster monitoring

Log in to the [CDW console](#), click a **Cluster ID/Name** in the **Cluster List** to enter the cluster details page, and switch to **Cluster Monitoring** to view performance metrics.

Note:

You can configure an alarm threshold and export monitoring data for specific metrics. Metrics are collected once every 10 seconds, the minimum time granularity for data display is 1 minute, and the maximum value of data points collected within 1 minute is displayed.

Cluster alarm configuration

1. On the cluster monitoring page, select a metric and configure its threshold for alarming.
2. On the CM console window that pops up, filter the instance objects (by ClickHouse cluster name) and configure the metric thresholds as instructed in the configuration template. The configuration items for an alarm policy are as displayed below:

Item	Description
Policy name	Name of the alarm policy
Monitor Type	Default: Cloud Product Monitoring
Policy type	Default: CDWCH/CK alarm
Alarm Object	Default: Instance ID
	Select a ClickHouse cluster in the drop-down list
Trigger Condition	Default: Manual Configuration
	Thresholds and alarm policies configured in Metric alarm

Notification template

Select an existing notification template or create one

Note:

Do not modify the default values.

3. Click **Complete** to submit the alarm policy. For more CM alarm policies, see [Creating Alarm Policy](#).

Monitoring Metrics

Node metrics

Metric	Description	Remarks
CPU utilization	CPU utilization	Average value
File opens	Number of file opens per unit of time	Average value
Memory utilization	Memory utilization	Average value
1-min node load	Node load in 1 minute	Average value
5-min node load	Node load in 5 minutes	Average value
15-min node load	Node load in 15 minutes	Average value
Data disk utilization	Data disk utilization	Average value
Inbound node traffic	Amount of data received by the node per unit of time	Average value
Outbound node traffic	Amount of data sent by the node per unit of time	Average value

Service metrics

Metric	Description	Remarks
Survival	Detects the survival of the CH process on a node	Instantaneous value
Contextual lock waits	Number of contextual lock waits	ClickHouseMetrics_ContextLockWait, instantaneous value
HTTP connections	Number of HTTP connections	ClickHouseMetrics_HTTPConnection, instantaneous value
TCP connections	Number of TCP connections	ClickHouseMetrics_TCPConnection, instantaneous value

Insertions per unit of time	Number of insertions per unit of time	ClickHouseProfileEvents_InsertQuery, average value
Merge time (rate)	Time consumed by merges per unit of time	ClickHouseProfileEvents_MergesTimeMilliseconds, average value
MySQL connections	Number of JDBC connections	ClickHouseMetrics_MySQLConnection, instantaneous value
CRUD queries	Number of queries including CRUDs per unit of time	ClickHouseProfileEvents_Query, average value
Query threads	Number of current query threads	ClickHouseMetrics_QueryThread, instantaneous value
Replica blocks merged per unit of time	Number of replica blocks merged per unit of time	ClickHouseProfileEvents_ReplicatedPartMerges, average value
Replica blocks modified per unit of time	Number of replica blocks modified per unit of time	ClickHouseProfileEvents_ReplicatedPartMutations, average value
Failed insertions	Number of failed insertions per unit of time	ClickHouseProfileEvents_FailedInsertQuery, average value
Failed queries	Number of failed queries per unit of time	ClickHouseProfileEvents_FailedSelectQuery, average value
Merges	Number of current merges	ClickHouseMetrics_Merge, instantaneous value

CK ZooKeeper metrics

Metric	Description	Remarks
ZooKeeper requests	Number of current ZooKeeper requests	ClickHouseMetrics_ZooKeeperRequest, instantaneous value
Current ZooKeeper sessions	Number of current ZooKeeper sessions	ClickHouseMetrics_ZooKeeperSession, instantaneous value
ZooKeeper watches	Number of current ZooKeeper watches	ClickHouseMetrics_ZooKeeperWatch, instantaneous value

ZooKeeper metrics

--	--	--

Metric	Description	Remarks
Sent packets	Number of packets sent by ZooKeeper nodes per unit of time	packets_sent, average value
Received packets	Number of packets received by ZooKeeper nodes per unit of time	packets_received, average value
ZooKeeper process survival	ZooKeeper process monitoring. 1: alive; 0: dead	Instantaneous value
Global sessions	Number of current global sessions	global_sessions, instantaneous value
Rejected connections	Number of connections rejected by ZooKeeper per unit of time	connection_rejected, average value
Committed queue requests	Number of current requests committed to the queue	request_commit_queued, average value
Wait time for queue preprocessing	Wait time for queue preprocessing per unit of time	prep_processor_queue_time_ms, average value
Preprocessing time	Preprocessing time per unit of time	prep_process_time, average value
ZooKeeper watches	Number of current ZooKeeper watches	watch_count, instantaneous value
JVM memory pool utilization	JVM memory pool usage details	jvm_memory_pool_bytes_used, instantaneous value

For metric descriptions, see [system.metrics](#).

Details

1. ZooKeeper monitoring metrics are available for HA clusters on v21.3.9.84 or later.
2. When advanced Grafana monitoring is enabled, the basic monitoring page will be updated to the advanced monitoring page.

Cluster Monitoring

Last updated : 2024-01-19 16:45:30

Background

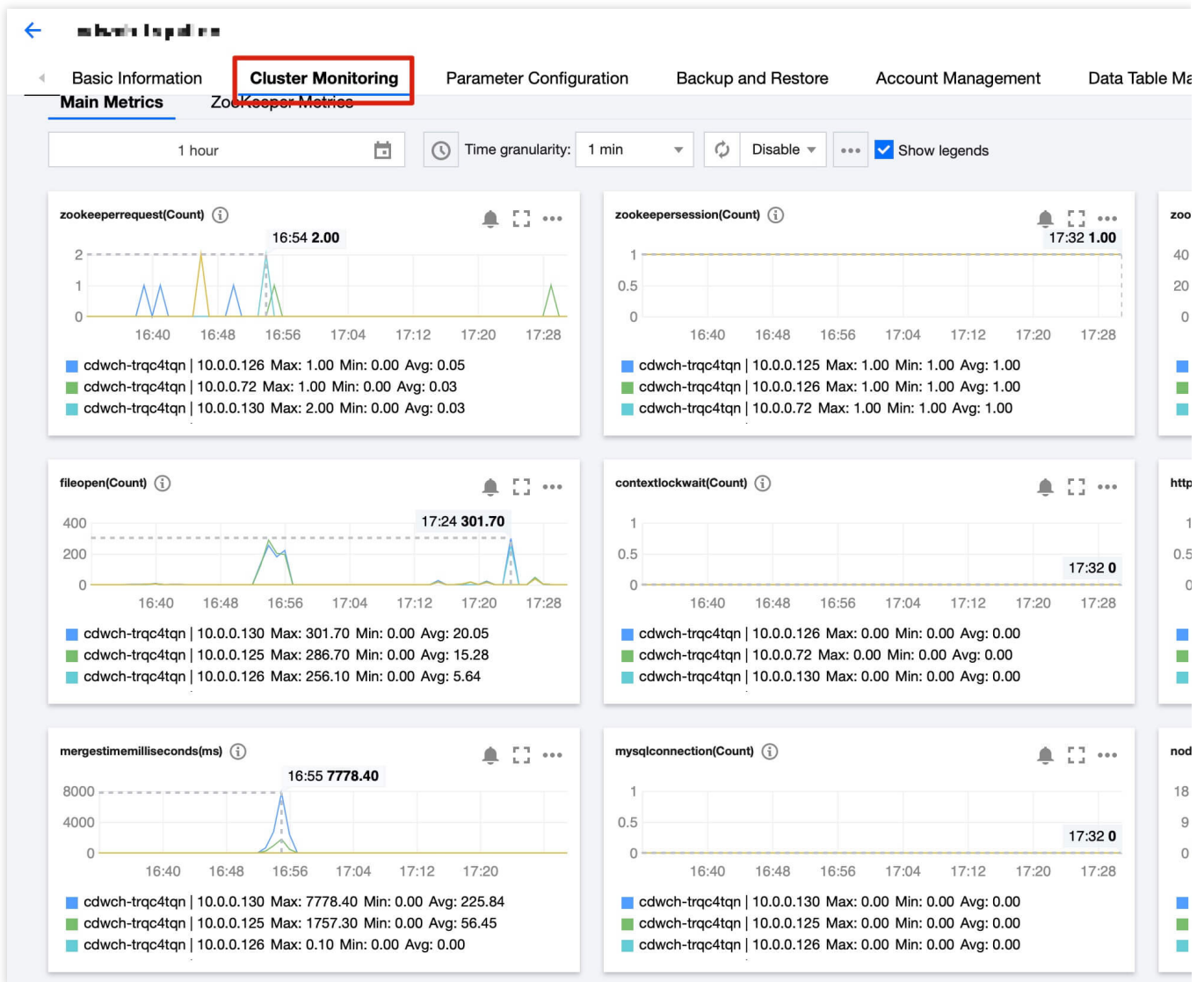
You can monitor the performance of ClickHouse clusters in the following two ways:

Use the default monitoring page if you don't enable Grafana monitoring when purchasing a cluster.

Use the advanced monitoring system with cluster alarm policies if you enable Grafana monitoring when purchasing a cluster.

Grafana Monitoring Disabled

Go to the [CDWCH console](#), click a **Cluster ID/Name** in the **Cluster List** to enter the cluster details page, and switch to the **Cluster Monitoring** tab to view performance metrics.



Metric	Description
Network connections	Total number of server connections
SELECT queries	Number of queries executed per unit of time
Total file opens	Number of file opens
Inserted rows	Number of insertions executed per unit of time
Merges executed on backend	Number of threads being merged
Total threads processing queries	Number of threads to start query processing
CPU utilization	CPU utilization of each node
1-min CPU load	CPU load in 1 minute of each node

Disk space utilization	Ratio of used disk space to the maximum available disk space * 100%
Memory utilization	Memory utilization of each node
Outbound network traffic rate	Rate of data sent by the ENI
Inbound network traffic rate	Rate of data received by the ENI

Granafa Monitoring Enabled

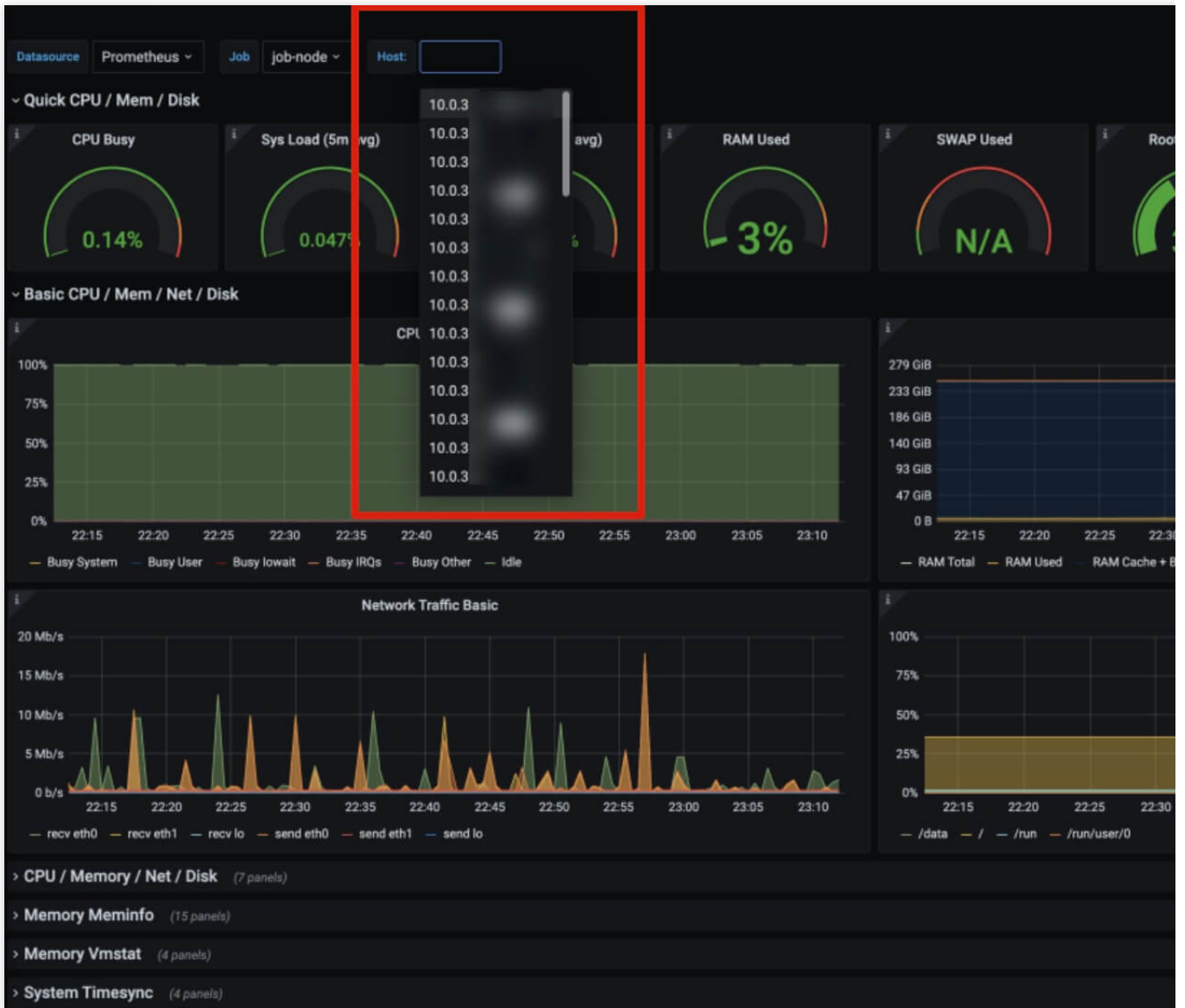
Monitoring dashboard

ClickHouse is preconfigured with four monitoring dashboards (**ClickHouse cluster**, **Single-Node server**, **Multi-Node server**, and **Node overview**). You can also customize dashboards as needed. The following describes the metrics and formulas for each dashboard.

Clickhouse cluster dashboard: See [Metric Description](#) for details. Click **ClickHouse Monitoring** in the top-right corner to switch to other dashboards.



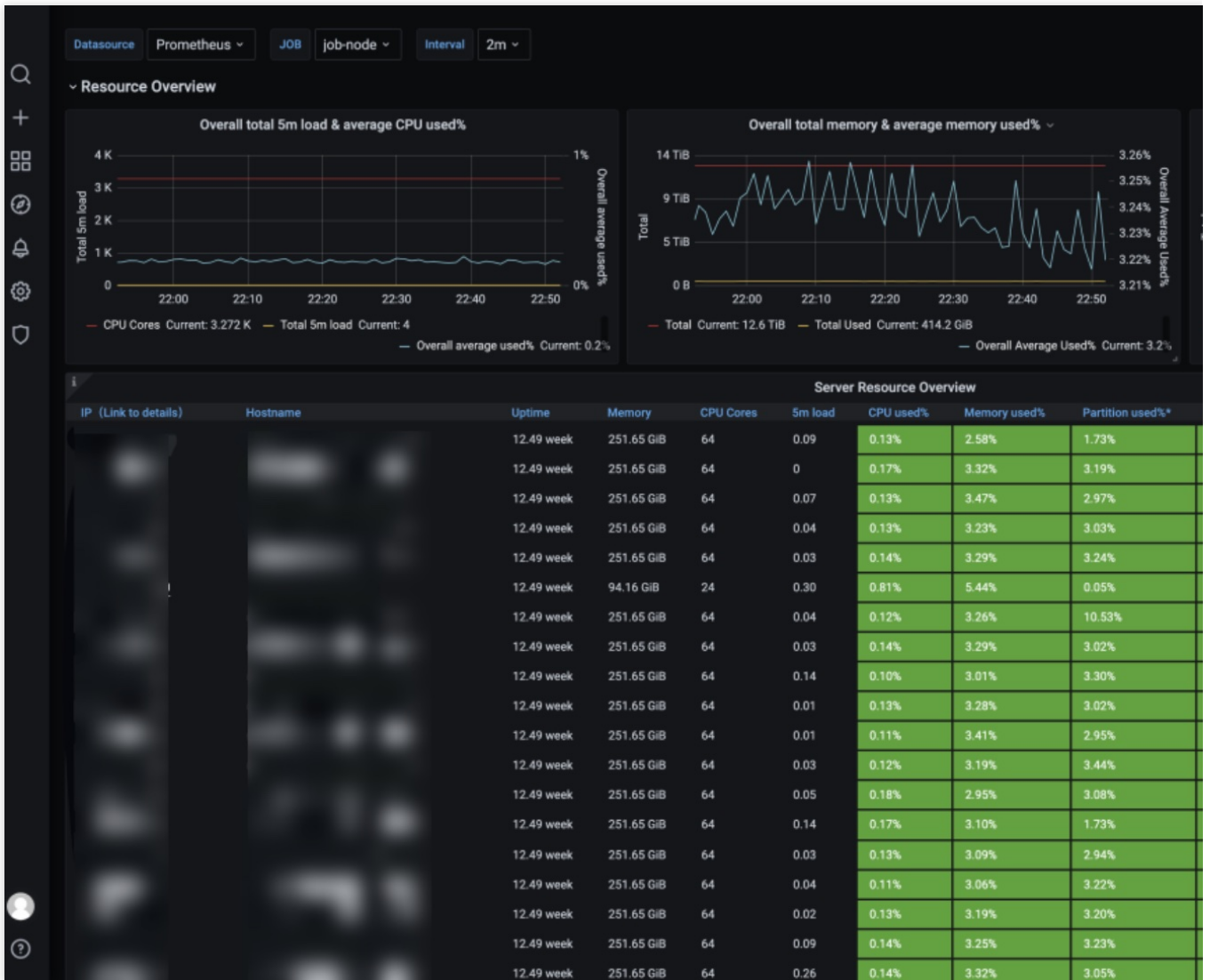
Single-node server dashboard: Details server metrics by IP.



Multi-node server dashboard: Horizontally compares 8 basic server metrics by IP.

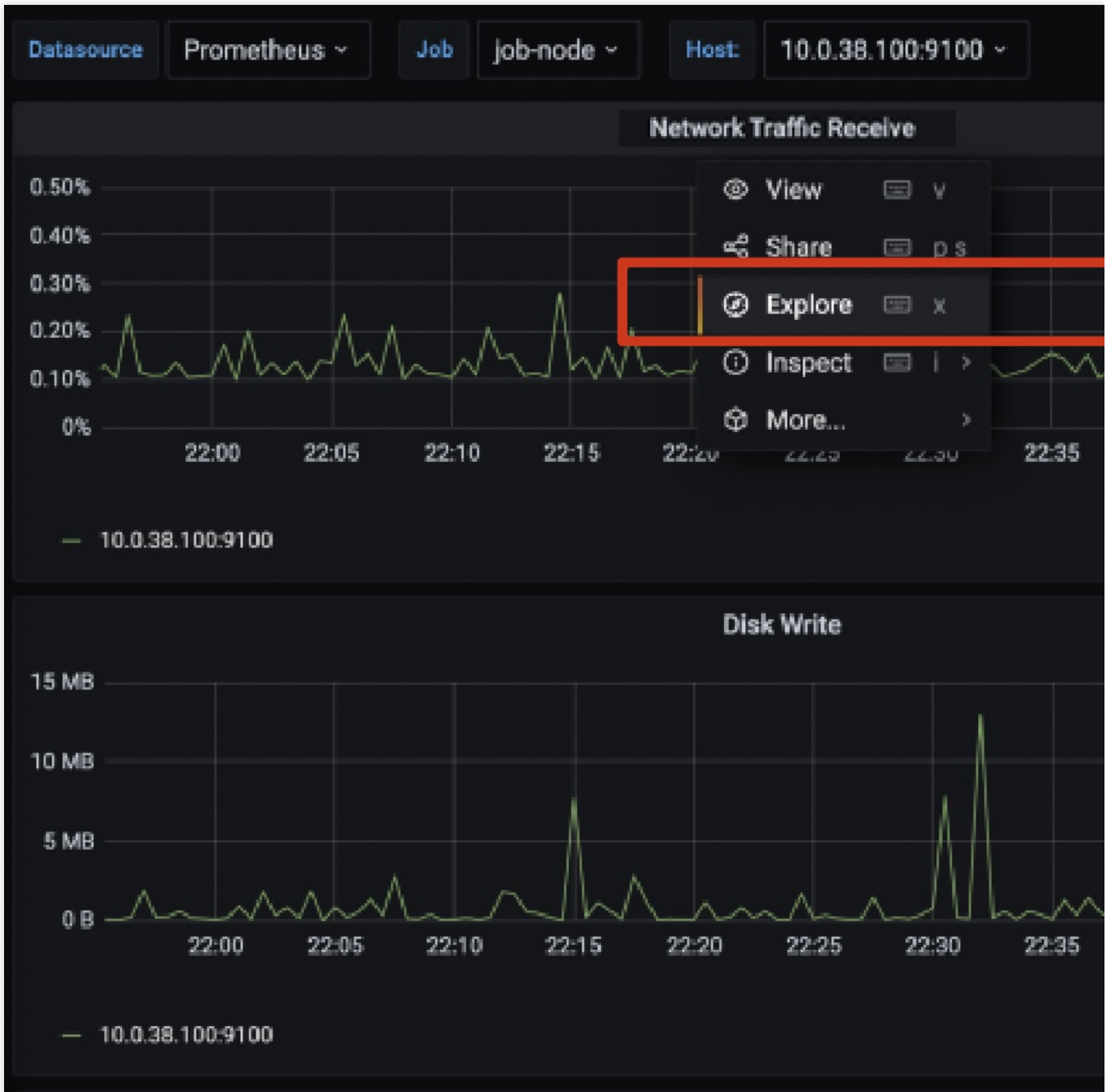


Node overview dashboard: Summarizes the basic server conditions of all nodes to offer a holistic picture of the entire cluster.

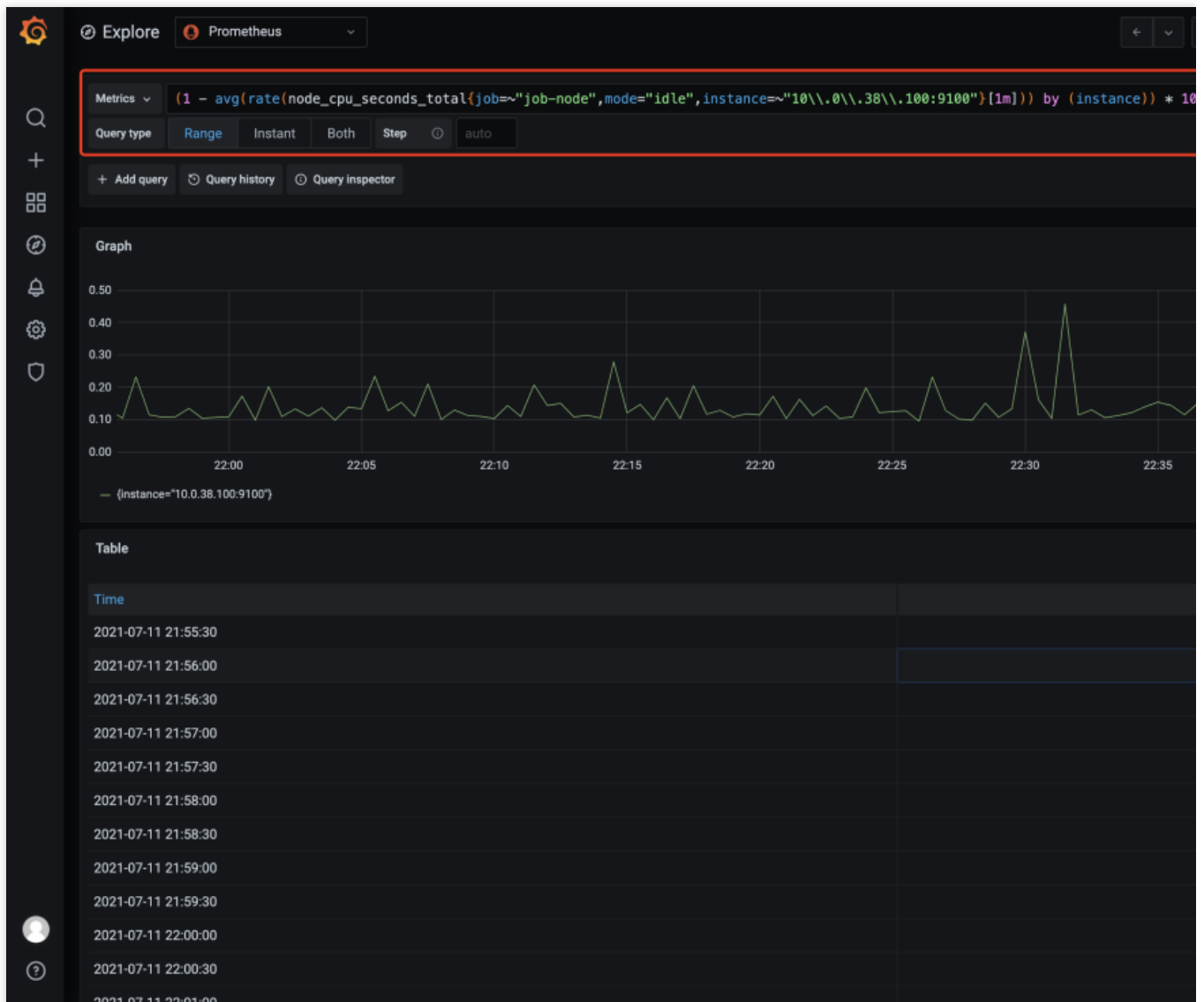


Metric calculation formula

Click a dashboard name and select **Explore** in the drop-down list to learn the details of a metric.



In a specific calculation, `node_cpu_seconds_total` is the metric. For more metrics, see [metrics](#).



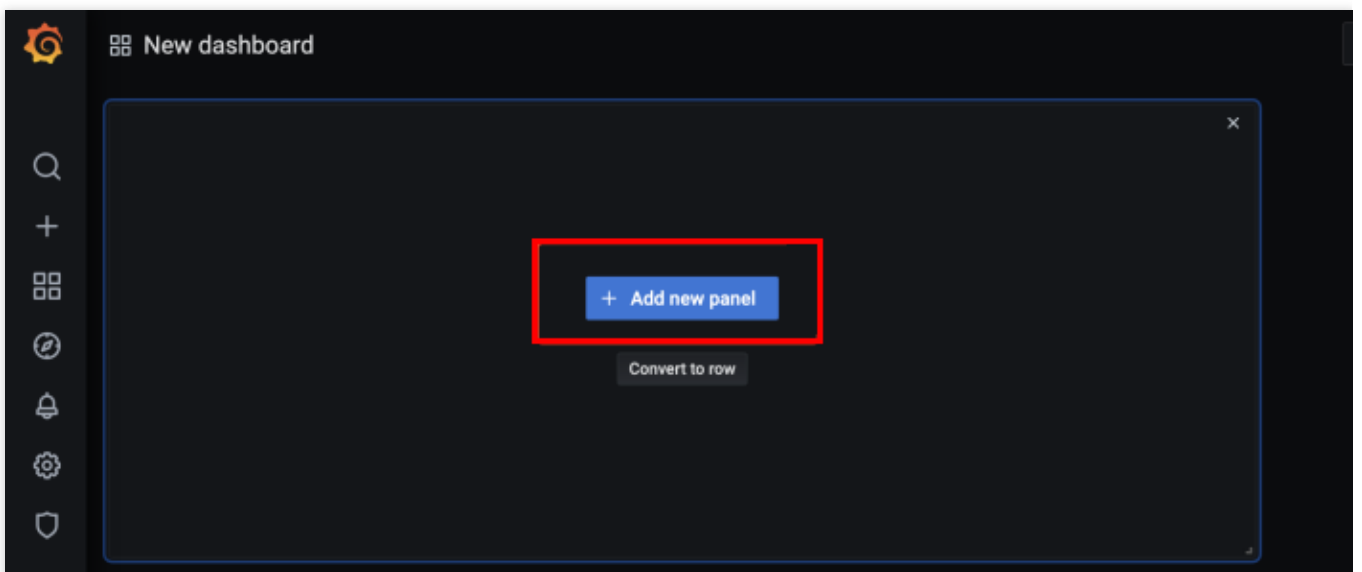
Custom panel configuration

You can personalize a panel to fit your usage habits.

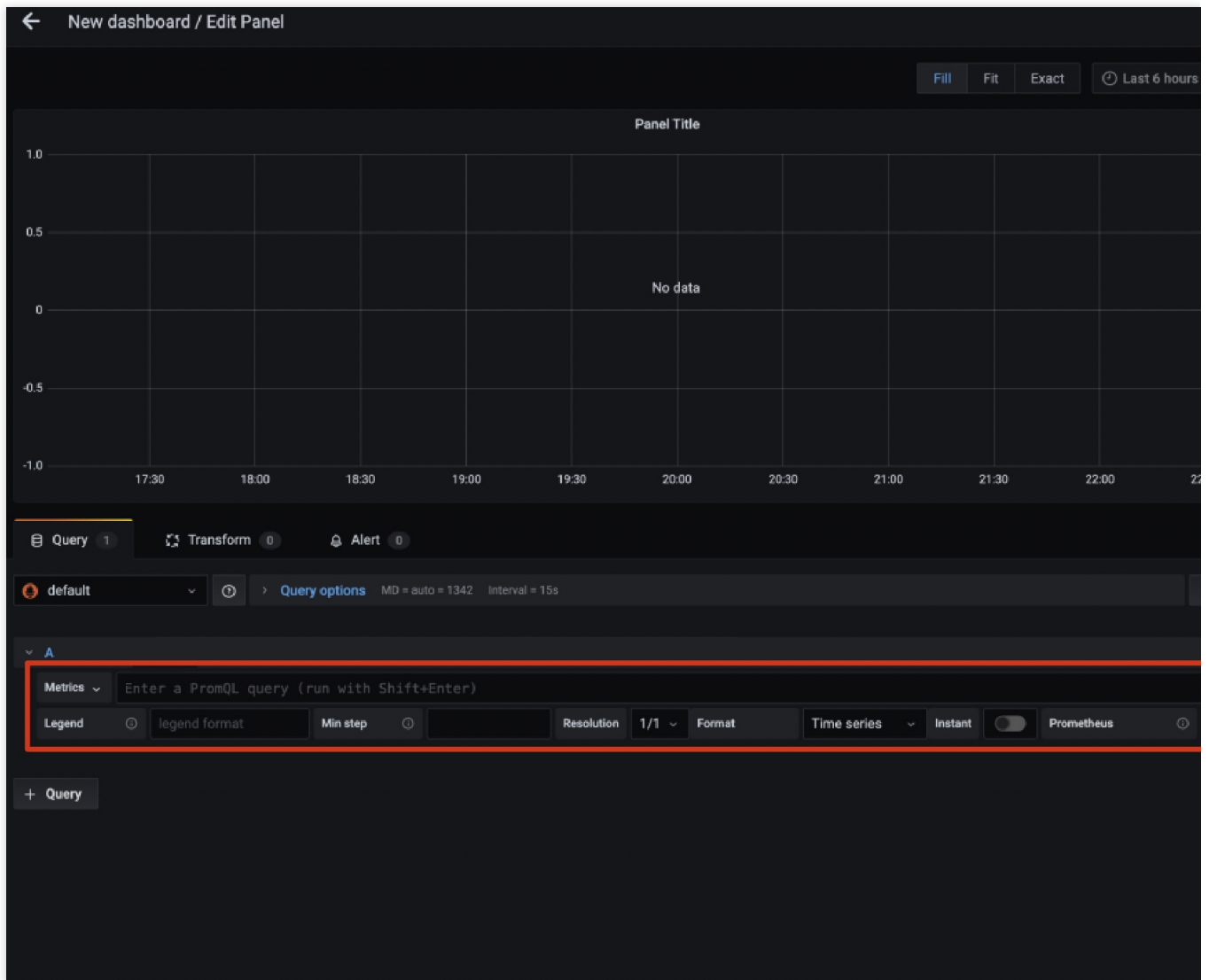
1. Click **+** on the left sidebar and select **Dashboard** in the drop-down list.



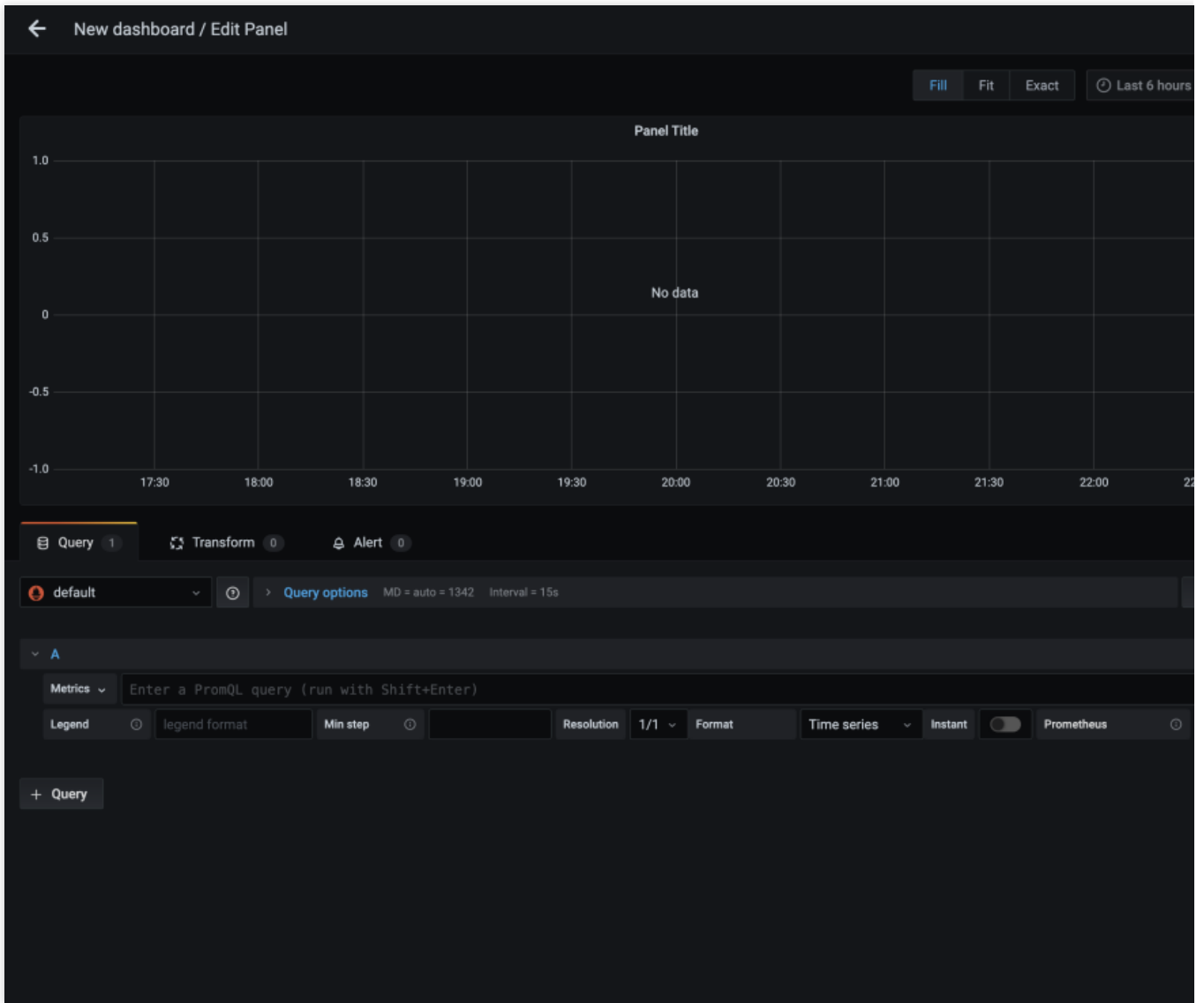
2. Click **+ Add new panel**.



3. Enter a metric or click **Metrics** to view the calculated metrics.



4. You can select a display style on the right. For more information, see [About Grafana panels](#).



5. Click **Apply** in the top-right corner and click **Save**.

Metrics

Metric	Description	Remarks
Total Query	Number of CRUD statements executed per unit of time	-
Query	Number of queries executed per unit of time	-
Replication	Sending, acquisition, and checking conditions of a single replica	-

Insert Query	Number of insertions executed per unit of time	-
Connections	Number of connections of each node	-
Read/Write Syscalls	Number of read/write system calls of each node	-
Number of Read/Write with a File Descriptor	Number of handles for file reads/writes and failed reads/writes per unit of time	-
Bytes of Read/Write with a File Descriptor	Size of files read and written per unit of time	-
Cache Rate	Cache hit rate and miss rate	Indicates repeated queries of the business
Selected Ranges	Number of index hits for a query, matching the amount of query data for a particular SQL hit	-
Selected Marks	Number of index hits for a query, matching the amount of query data for a particular SQL with a finer granularity	-
Merge1	Number of threads being merged	The number of merges should not be set too large. A high merge rate means that the amount of data imported per batch is too small, and the data is relatively concentrated and proportional to the part file directories
Merge2	Number of rows being merged	-
Merges Time	Compression and consumption time (rate)	It is related to the amount of compressed data
Parts of ReplicatedMergeTree Merged	Number of replicated parts merged per unit of time	-
Mutations	Number of replicated part	-

	mutations per unit of time	
Pool Tasks	Number of tasks performed on the backend	-
Open Files	Number of file opens per unit of time	-
Compressed Read Buffer	Size of compressed read cache used per unit of time	-
Memory	Memory usage of each node	-

Alarm Configuration

Last updated : 2024-01-19 16:45:30

Background

You should enable Grafana (Prometheus) monitoring during cluster purchase to configure the monitoring and alarming feature as needed. After a cluster is created, the cluster details page in the [CDW console](#) will display a Grafana address, where prom-xxxxxx is the name of the Prometheus instance you purchased.

Configuring an Alarm Policy

1. To configure ClickHouse cluster alarms, go to [Prometheus Monitoring](#) in the CM console, select the Prometheus monitoring instance corresponding to your ClickHouse cluster by region and instance name, and click **Instance ID/Name** to enter the details page.
2. On the instance details page, select **Alarm Policy > Create**, select preconfigured basic policies (including the number of connections, CPU, memory, and data disk), and apply them to all nodes of the cluster.
3. Configure (or create) a notification template to notify relevant users when the threshold is exceeded. An alarm policy will take effect immediately after it is saved.

Hot/Cold Data Tiering

Last updated : 2024-01-19 16:45:30

Feature Description

When there is too much data in a cluster instance, you can store some cold data in a COS bucket and hot data on a CDWCH local disk to save storage costs without compromising CDWCH's data query performance.

Terms

Hot data: Frequently accessed or recently created data, which is stored on the cloud or local disk selected at the time of cluster creation for efficient data access.

Cold data: Less frequently accessed data, which can be stored on a cold data disk to save storage costs and meet data access needs.

Tiered storage of hot/cold data provides three data moving methods. Parts that satisfy the TTL policy will be moved first; if the storage capacity is still exceeded, large parts will be moved first, as further described below:

Storage capacity policy: Newly written data is stored to a hot data disk for efficient query. When the amount of stored hot data reaches the usage threshold, the data on the hot data disk will be automatically moved to the cold data disk to free up the storage space of the hot data disk.

TTL-based tiered storage policy: A TTL statement is added to the default storage policy to automatically move all data older than the time interval to a cold data disk.

Data can be manually moved between hot and cold data disks.

Notes

This feature is applicable to CDWCH v21.3.9.83 or later. If your instance is not supported, upgrade it to v21.3.9.83 or later first.

During tiering, the cluster will be restarted and become inaccessible.

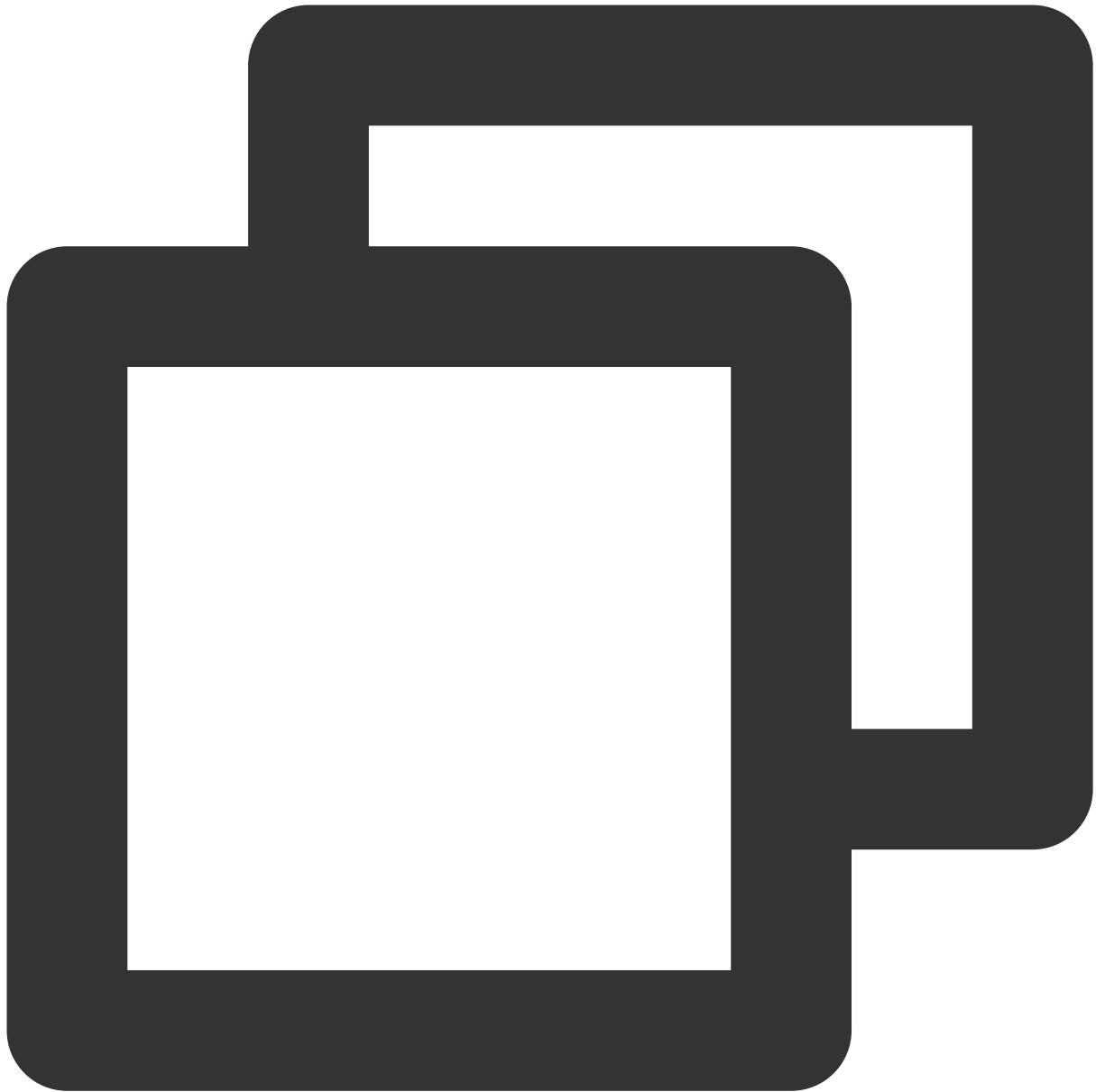
To use tiered storage, add the `SETTINGS storage_policy = 'hot_to_cold';` statement to specify the storage policy during table creation or dynamically modify the storage policy of the table (which must be `default` before the change).

The storage policy cannot be modified once specified.

The tiered storage feature cannot be disabled once enabled.

Due to network restraints, you need to have a COS bucket in the same region as the cluster to perform tiered storage.

Directions



Log in to the [CDWCH console] (<https://console.intl.cloud.tencent.com/cdwch>). When c

According to the cluster policy, when the used storage space reaches 90%, a cold data tiering mechanism will be automatically triggered by default, where large data parts will be transferred from the local disk to the configured COS bucket first until the used space drops below the threshold. You can configure the tiering table with the following command as needed: `ALTER TABLE xx MODIFY SETTINGS storage_policy = 'hot_to_cold' .`

Hot-Cold Tiered Storage Configuration

Cold Backup Storage Enable

After cold backup storage is enabled, data will be classified as hot and cold based on the capacity policy. When more than 90% of the node storage capacity is use

COS Disk for Cold Backup

Select an existing COS disk, or [create a COS disk](#)

1. To enable data tiering after a cluster is created, select **Operation > More > Configure Hot-Cold Tiered Storage** in the cluster list.

New Cluster
CDW Studio

Separate search items with Enter Separate e

ID/Name	Status	Compute Node	Kernel Version	Availability Zone	Network/Subnet	Billing Mode
	Serving	Standard, 2 nodes 4-core 16 GB, 200 GB	21.3.9.84	ap-singapore-1		Pay-as-you-go

Total items: 1

2. After tiering is configured, view or adjust the hot data tiering coefficient on the cluster details page.

←
Cluster Monitoring
Parameter Configuration
Backup and Restore
Account I

Basic Information

Cluster ID: [redacted]

Cluster Name: [redacted]

Billing Mode: Pay-as-you-go

Creation Time: 2022-06-28 14:27:07

Tag: [Modify](#)

Cluster Status

Cluster Status: Serving

Configuration Information

Kernel Version: 21.8.12.29

High availability: High availability

Compute Node Type: Standard

Compute Node Spec: 4-core 16 GB

Compute Node Count: 2

Compute Node Storage: CLOUD_HSSD200GB

ZooKeeper Node Spec: 4-core 16 GB

ZooKeeper Node Storage: CLOUD_HSSD100GB

ZooKeeper Node Count: 3 [Increase multi-ZooKeeper Nodes](#)

Network Information

Availability Zone: [redacted]

VPC ID: [redacted]

Subnet ID: [redacted]

Cluster Access Address (http): [redacted]

Cluster Access Address (tcp): [redacted]

Node Information

No.	No.
1	[redacted]
2	[redacted]

Total items: 2

Hot-Cold Tiered Storage

COS Bucket: [redacted]

Capacity Tiering Policy: 90% (When more than 90% of the node storage capacity is used, earlier data will be migrated to the cold backup storage)

[Modify Policy](#)

3. If tiering is triggered, you can log in to the COS console and go to the COS bucket details page to query cold data files.

Moving Hot-Cold Tiered Data

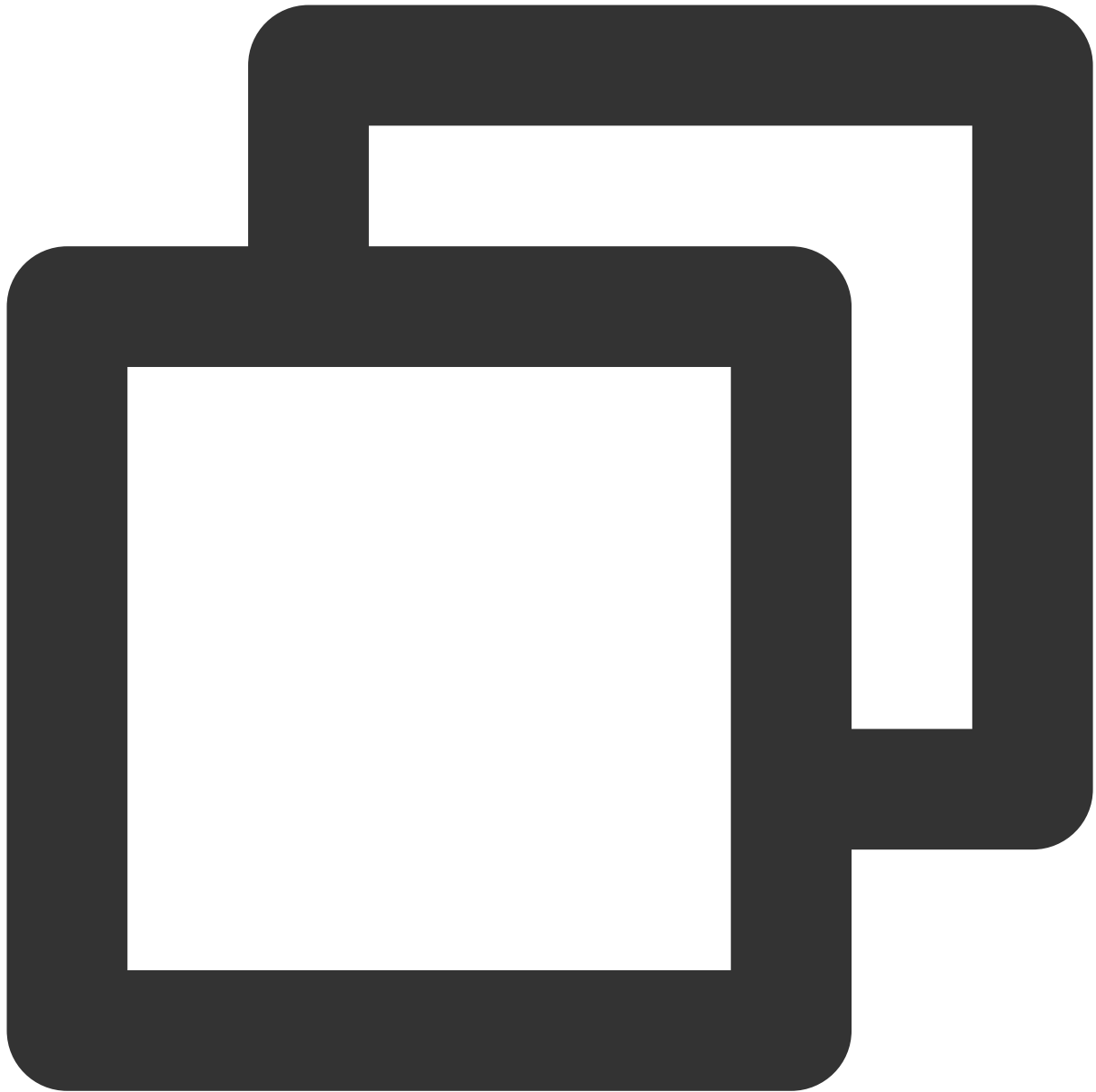
1. After tiered storage is enabled, the parameters of the default storage policy are as described below:

Parameter	Description
<code>move_factor</code>	When the proportion of the available storage space of the hot data disk is below this value, the oldest data written to the hot data disk will be automatically moved to the cold data disk. The valid value is a floating point number between 0 (exclusive) and 1 (inclusive). The default value is 0.1, indicating that data will be automatically moved when the proportion of available storage space is below 10%. This parameter can be modified through the tiered storage policy in the console.
<code>prefer_not_to_merge</code>	Specifies whether to merge the data on the cold data disk. Valid values: true: nofalse (default): yes.
<code>max_data_part_size</code>	Maximum part size. Parts on the hot data disk exceeding this size will be moved to the cold data disk. The default value is 0 (unlimited).

2. Set the TTL-based tiered storage policy.

You can add a TTL statement to the default storage policy to automatically transfer all data older than the time interval to the cold data disk. You can also set the part expiration time by setting the TTL at the **table level** for data migration. The calculation result of the expression must be of the `Date` or `Datetime` type. **The TTL needs to be expressed by the `INTERVAL` operator of a `Datetime` or `Date` field.** For the moving characteristics of a part, all rows in the part must meet the moving expression.

You can add a TTL statement according to the following syntax. The TTL can be set to `TO DISK 'cold_disk'` or `TO VOLUME 'cold'` (with the same effect), and the name can be viewed through the `system.disks` and `system.storage_policies` system tables.



```
TTL
+ INTERVAL
TO DISK 'cold_disk'
```

Parameter description

Parameter	Description
time_column	Column of `Date` or `Datetime` type .
number	Time interval in days, weeks, months, or years.

Example To move all data older than 90 days (based on the date column) to the cold data disk, use the following table creation statement: ``CREATE TABLE ttl_test_tbl(`f1` String, `f2` String, `f3` Int64, `f4` Float64, `date` Date)ENGINE = MergeTree()PARTITION BY dateORDER BY f1TTL date + INTERVAL 90 DAY TO DISK 'cold_disk'SETTINGS storage_policy = 'hot_to_cold';``

Change the TTL-based tiered storage policy

You can change the column of `Date` or `Datetime` type and time interval of the TTL-based tiered storage policy.

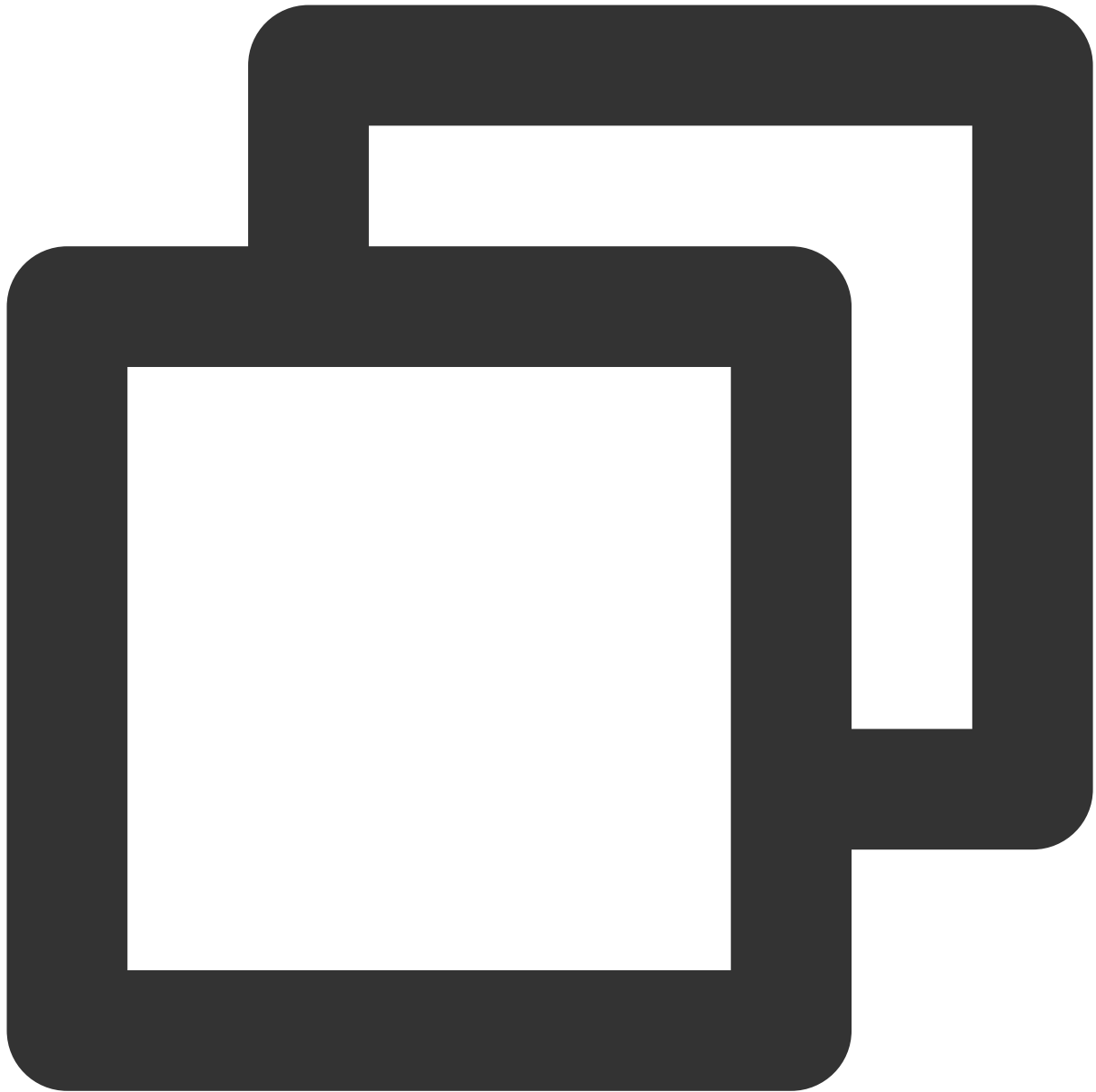
Note:

After the TTL-based tiered storage policy is modified, all existing and new data will be stored according to the new policy by default.

If you don't want the change to the TTL-based tiered storage policy to take effect for existing data, run the `set materialize_ttl_after_modify=0;` statement first before modifying the policy. In this way, only new data will be stored according to the new policy.

After the TTL-based tiered storage policy is modified, data on the cold data disk will not be automatically moved to the hot data disk, but it can be manually moved.

You can modify the TTL-based tiered storage policy according to the following syntax:



```
ALTER TABLE  
MODIFY TTL  
+ INTERVAL  
TO DISK 'cold_disk';
```

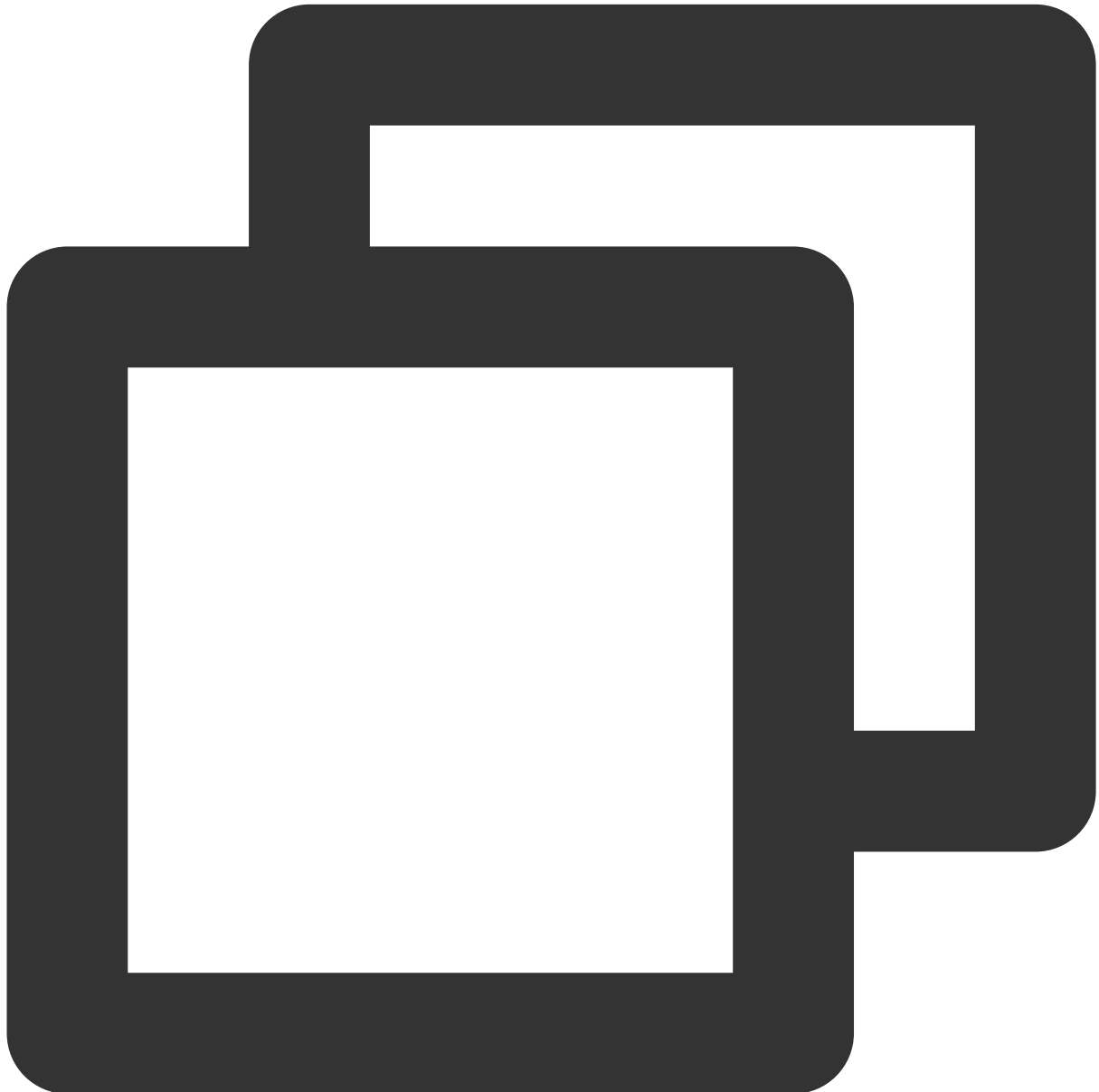
Parameter description

Parameter	Description
table_name	Table name.

time_column	Column of `Date` or `Datetime` type after the change.
number	Time interval in days, weeks, months, or years after the change.

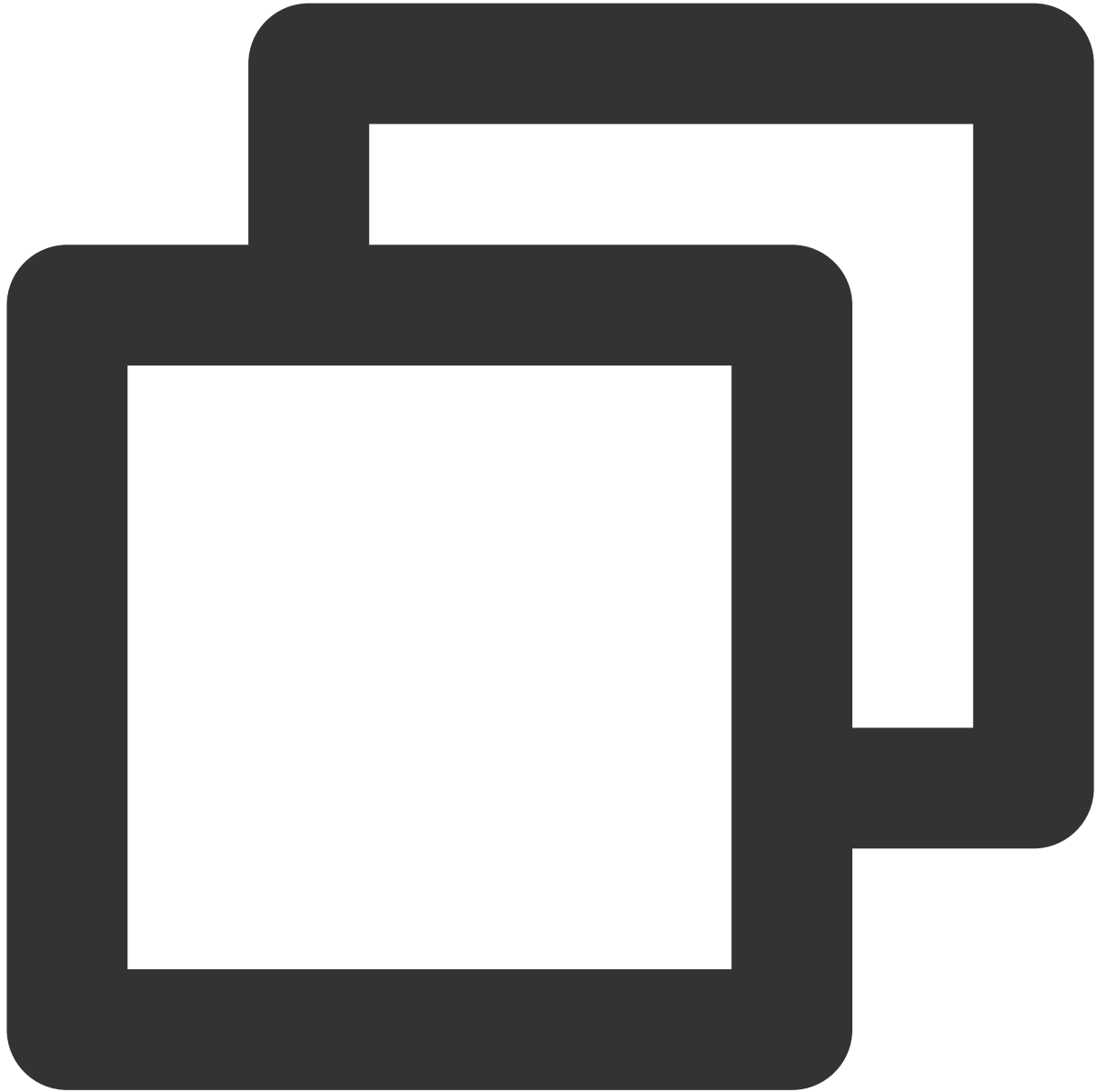
3. Move the data on the hot or cold data disk.

Check the location of the current part or partition before moving:



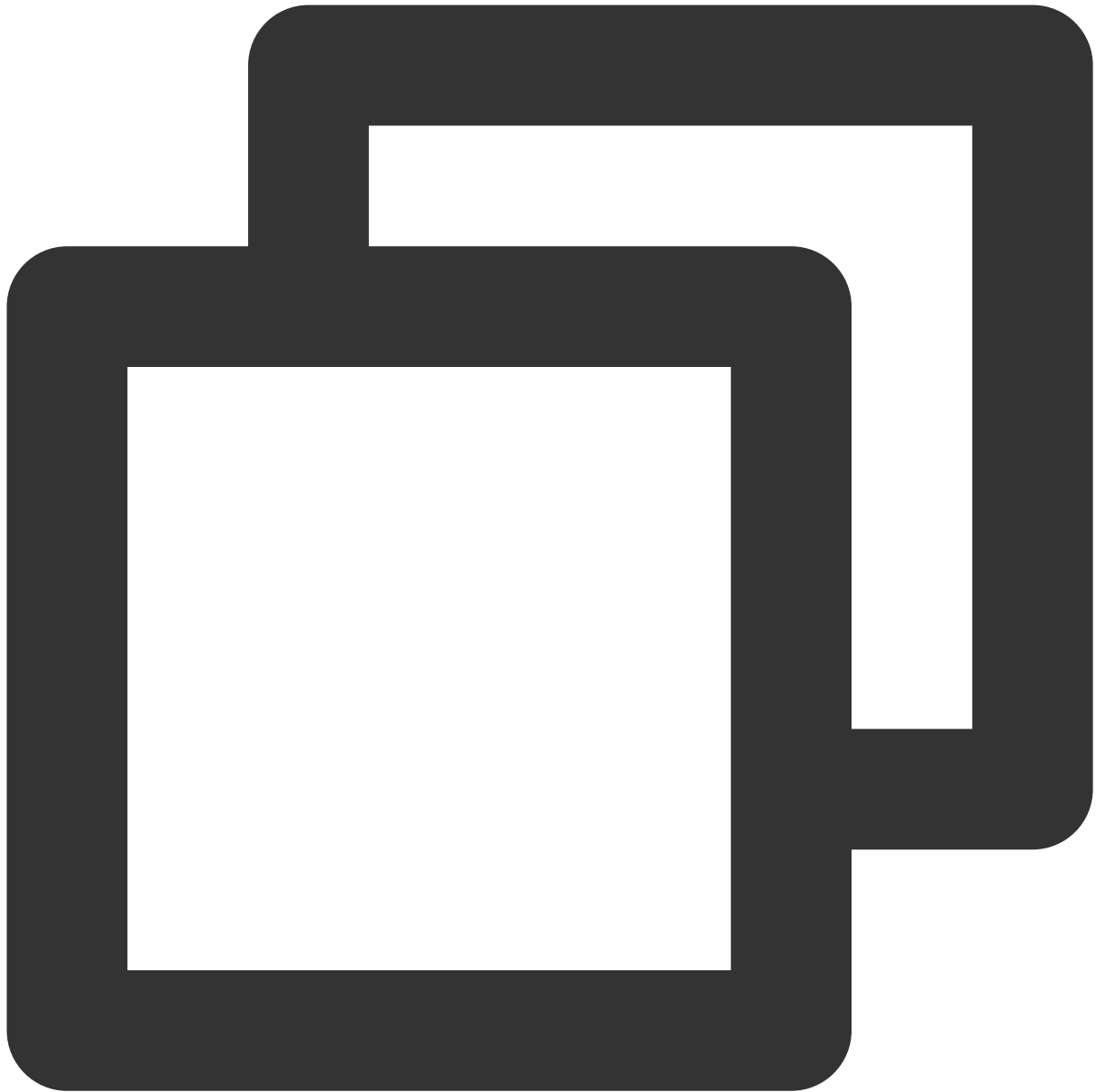
```
select partition,name,table,disk_name,database from system.parts where active=1
```

Find the part or partition to be moved by using the following syntax:



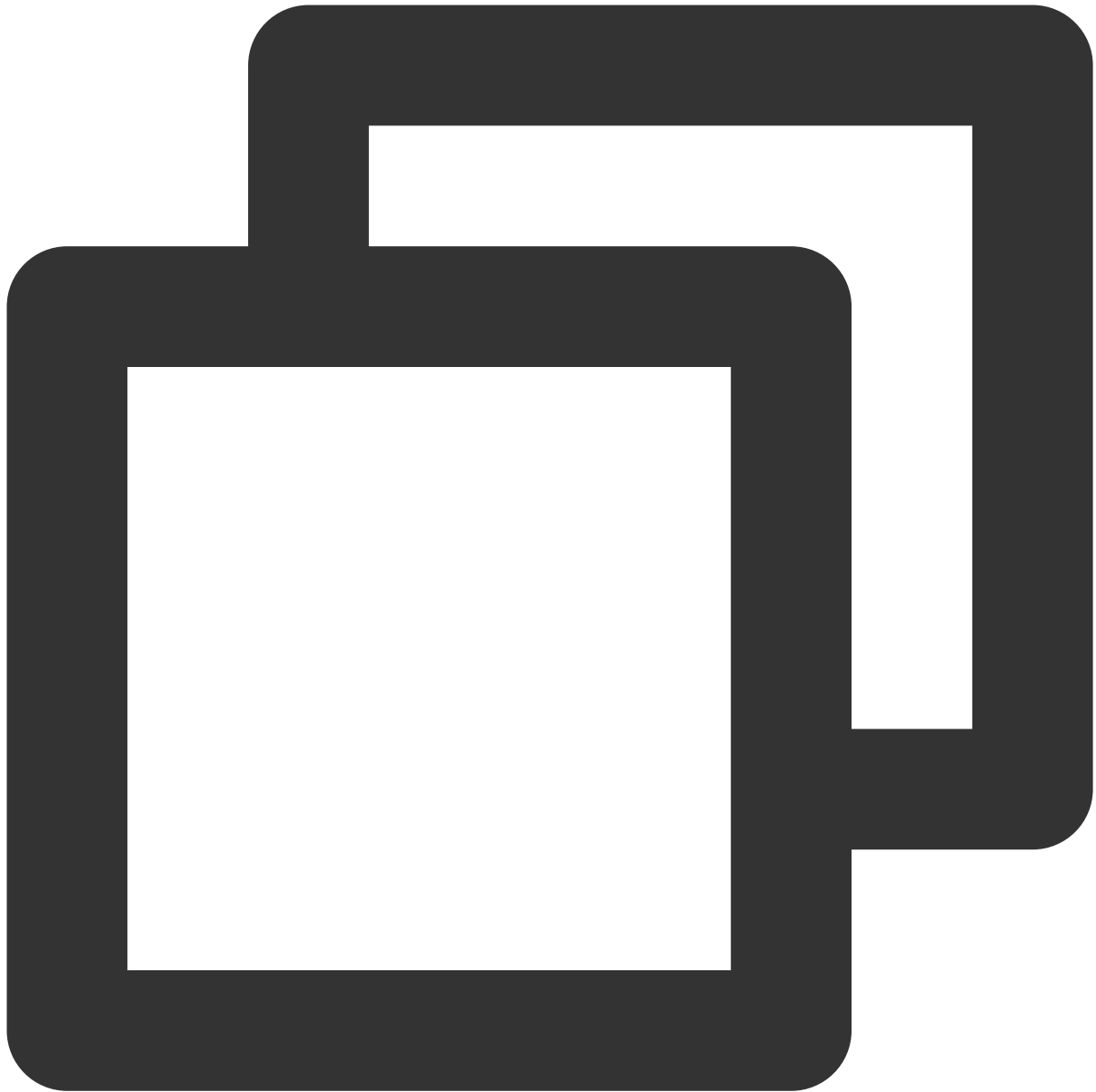
```
ALTER TABLE table_name MOVE PART | PARTITION partition_expr TO VOLUME ' volume_name
```

Move data from the hot data disk to the cold data disk.



```
ALTER TABLE
MOVE PARTITION <partition>
  TO DISK 'cold_disk'; ALTER TABLE
MOVE PARTITION <partition>
  TO VOLUME 'cold';
```

Move data from the cold data disk to the hot data disk.



```
ALTER TABLE
ON CLUSTER default MOVE PARTITION
TO DISK 'default'; ALTER TABLE
ON CLUSTER default MOVE PARTITION
TO VOLUME 'hot';
```

Possible Problems During Use

1. After tiering, data will be first written to the hot data disk. If the data imported at a time exceeds the remaining available capacity of the hot data disk, an error may be reported, indicating that the disk has no remaining space.
2. Only the kernel on v21.6 or later supports the COS connection retry mechanism. If time-consuming operations such as join are performed on an earlier version, the following error may be reported. This is because it takes CDWCH more than a certain period of time to read files after connecting to COS. In this case, you can adjust the SQL statement or upgrade the version.

```
Received exception from server (version 21.3.9):  
Code: 23, DB::Exception: Received from localhost:9000, DB::Exception: Cannot read from istream at offset 0: (while reading column LO_CUSTKEY): (while reading  
f-c69f36e61f2f/1994_2336_3381_3/ from mark 96 with max_rows_to_read = 8192): While executing MergeTreeThread.
```

Account and Authorization

Last updated : 2024-01-19 16:45:30

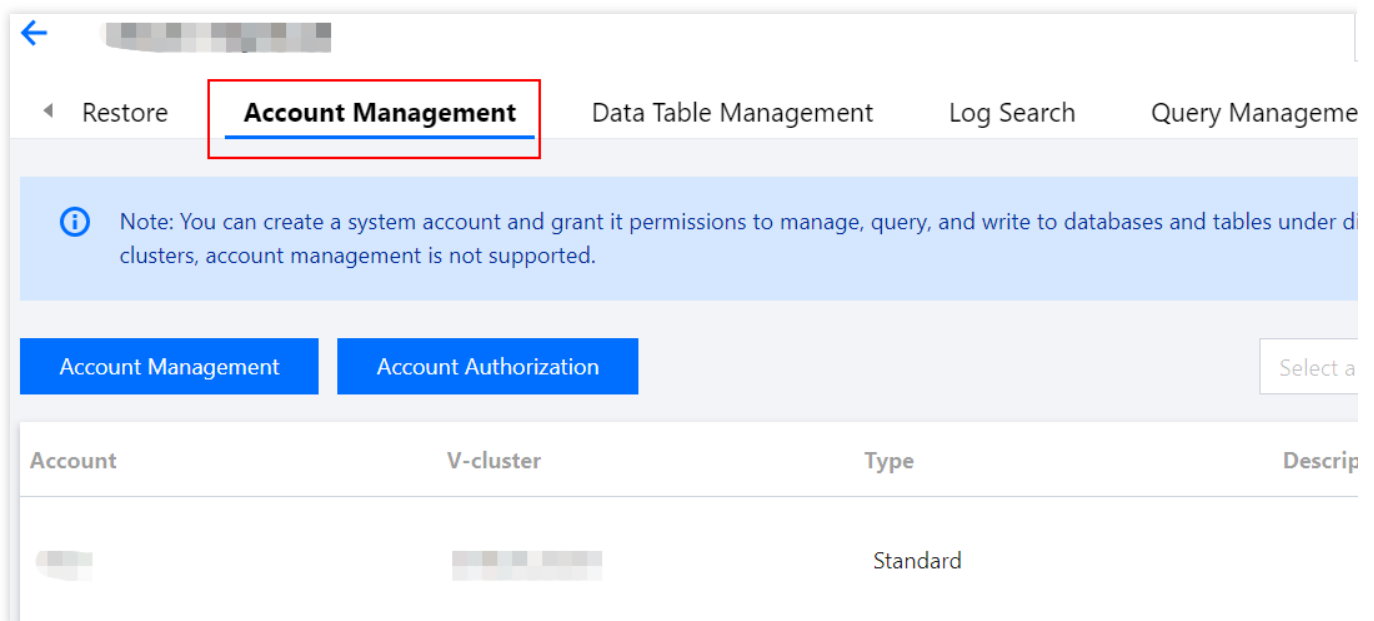
Overview

CDW provides a visual UI for you to create RBAC accounts and configure their permissions, quotas, settings profiles, and row policies. This helps you control account permissions in a CDW cluster more easily and efficiently.

Account

CDW allows you to create and delete accounts and view and reset passwords.

1. Log in to the [CDW console](#), click a **Cluster ID/Name** in the cluster list to enter the cluster details page, and switch to **Account Management**.



2. You can add and delete accounts and change their passwords.

Account Management

Database Account * Add account ▼ 2-16 characters; supports lowercase letters, di

Account Type * Add account

Password * Change passw...

Confirm Password * Delete account Enter the password again

Description The description cannot exceed 256 characters and cannot start wit

OK Cancel

Note:

Account name rule: It can contain 2–16 lowercase letters, digits, and underscores and must start with a letter and end with a letter or digit.

Password rule: It must contain 8–30 (preferably more than 12) characters in at least three of the following types: lowercase letters, uppercase letters, digits, and symbols `()!@#%$%^&*|?><`, and cannot start with `/`.

An XML account is a user defined in the `user.xml` configuration file, while an RBAC account is an account created on the **Account Management** page. The two types of accounts are separated. We recommend you not use duplicate account names; otherwise, account permissions may be inconsistent.

XML accounts can be only queried but cannot be modified. You can only perform modifications in the `user.xml` configuration file.

Authorization

1. Log in to the [CDW console](#), click the target **ID/Name** in the cluster list to enter the cluster details page, and switch to **Account Management**. Click **Authorize Now** to enter the **Account Authorization** tab and authorize the account (by selecting a V-cluster and granting permissions of databases and tables):

Account Authorization

Database Account *

V-cluster *

Access Scope * All databases and tables Some databases and

General Permissions * Query Set ⓘ Insert ⓘ

High Risk Permissions ⚠ Table: Create Clear Delete
 Database: Create Delete

Access Scope: You can grant the permissions of all or certain databases and tables.

General Permissions:

Query (enabled by default): SELECT, which is used to query databases, tables, and views.

Configuration: ALTER, which is used to modify table structures.

Insertion: INSERT, which is used to insert table data.

High-Risk Permissions:

Table permissions

Table creation: CREATE TABLE, CREATE VIEW, and CREATE DICTIONARY, which are used to create tables, views, and dictionaries respectively.

Clearing: TRUNCATE and OPTIMIZE, which are used to clear and merge the table data respectively.

Table deletion: DROP TABLE, DROP VIEW, and DROP DICTIONARY, which are used to delete tables, views, and dictionaries respectively.

Database permissions

Database creation: CREATE DATABASE, which is used to create databases. This option takes effect globally. If it is selected, you can see all databases.

Database deletion: DROP DATABASE, which takes effect only for specific databases.

2. Authorize certain databases and tables.

Note:

Some databases and tables supports finer-granularity and more flexible database/table permission configuration; that is, you can configure different permissions for any databases/tables.

Each permission configuration is an incremental operation. Compared with previous permissions, unchanged permissions are not deleted, only removed permissions are deleted, and new permissions are added. This avoids affecting business operations during the permission change.

Configuration

On ClickHouse 20.3 and later, items that previously can only be configured through XML such as row policy, settings profile, and quota can be created, modified, and deleted through SQL statements.

1. Quota

Note:

A quota offers seven configuration items. You can hover over each of them to view the description.

In the console, you can select multiple configuration items among the seven items as a combination.

2. Settings profile

Each settings profile contains a certain number of configuration items, and all 335 configuration items can be found in the `settings` table in the `system` database. In the console, you can select multiple configuration items as a combination and name them as a single settings profile, which can be applied to one or multiple RBAC accounts.

3. Row policy

A row policy is a filter defining the rows that can be seen by users.

Note:

If multiple row policies are configured for a table in the cluster, users with no policies configured cannot query the data of the table.

Notes

1. Operations on the **Account Management** page involve only RBAC accounts, but not XML accounts.

2. You can grant one account the permissions of multiple databases/tables in the V-cluster. Each operation generates an authorization record.
3. After authorization, you need to modify the password of the default passwordless XML account to avoid potential security risks.
4. After you configure an RBAC account on the **Account Management** page, you can run common ClickHouse commands on the **CDW Studio platform** in the **CDW console**.
5. The new CDW Studio platform is available now, and XML users can log in to it to perform operations.

Query Management

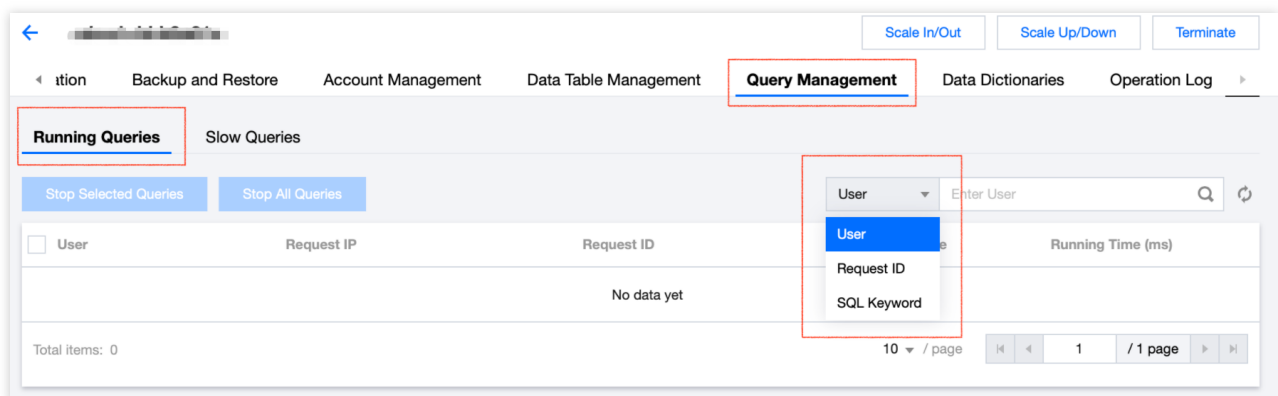
Last updated : 2024-01-19 16:45:30

Feature Overview

You can quickly get started with SQL query execution in the Cloud Data Warehouse console, stay on top of slow queries, and optimize or terminate running queries to reduce the usage of cluster resources. You can also manipulate original query log tables on the interactive UI in a simplified and efficient manner.

Running Queries

1. Log in to the [Cloud Data Warehouse console](#), click the **ID/Name** of the target cluster in the cluster list to enter the cluster details page, and switch to the **Query Management** tab.
2. Click the **Running Queries** tab, which displays the details of all running SQL queries by default. You can filter the queries by type, terminate selected queries as needed, or terminate all queries to release resources temporarily.



Slow Queries

1. Log in to the [Cloud Data Warehouse console](#), click the **ID/Name** of the target cluster in the cluster list to enter the cluster details page, and switch to the **Query Management** tab.
2. Click the **Slow Queries** tab, which displays queries taking more than 500 ms to run. You can adjust the threshold of slow query execution duration, timeout period, and set the filter time (including last 15 minutes, last 30 minutes, last hour, and custom time period) as needed and then click the refresh icon.

Note:

500 ms is the minimum value of slow query execution duration.

You can filter slow queries by date or time (accurate to the minute).

The screenshot shows the 'Query Management' interface in the Tencent Cloud console. The 'Query Management' tab is highlighted with a red box. Underneath, the 'Slow Queries' sub-tab is also highlighted with a red box. The interface includes a navigation bar with options like 'Scale In/Out', 'Scale Up/Down', and 'Terminate'. Below the navigation bar, there are tabs for 'Running Queries' and 'Slow Queries'. The 'Slow Queries' section has a filter for 'Slow Query Running Time (ms)' set to 500, and a time range filter set to 'Last hour' (2022-02-22 19:53:54 - 2022-02-22 20:53:54). A table displays the following data:

User	Access Source Address	Initial Request ID	SQL Statement	Execution Start Time	Running Time (ms) †	Read Rows	Read Data Size	Return Result Size	Memory Usage
default		8...	select sleepEachRow(...)	2022-02-22 20:53:10	6002	810	810	34	0
default		...	select sleepEachRow(...)	2022-02-22 20:52:55	8002	804	804	36	0
default		3...	select sleepEachRow(...)	2022-02-22 20:52:04	5002	804	804	33	0
default		0...	select sleepEachRow(...)	2022-02-22 20:51:53	6002	804	804	34	0

At the bottom of the table, it indicates 'Total Items: 4' and a pagination control showing '10 / page' and '1 / 1 page'.

Log Search

Last updated : 2024-01-19 16:45:30

Configuring Cluster Log Service

Cluster logging configuration

You can collect ClickHouse node logs under your account to Cloud Log Service (CLS) as well as enable and use the log search feature as needed.

Note:

CLS is postpaid, so make sure that your account balance is sufficient to ensure normal ClickHouse log upload and display. For more information on CLS fees, see [Billing Overview](#). When CLS is enabled, the authorized permissions will take effect simultaneously.

Enabling cluster logging

Enable CLS for a new cluster

You can configure CLS when creating a cluster. Select an existing CLS logset or create one in the same region as the cluster. Logs are stored for 30 days by default.

Log Configuration

CLS Enable

After you enable CLS, the system will charge you on a pay-as-you-go basis. View [Purchase Guide](#) for details.

Logset

Select an existing logset, or [create a logset](#)

Note: To ensure network interconnectivity, ensure the CLS service and cluster are in the same region.

Note:

A log topic will be created in your configured logset to store ClickHouse logs. You can view the topic on the CLS page. Do not delete this topic configured for the ClickHouse cluster; otherwise, log queries will fail on the ClickHouse log search page.

Enable or modify CLS for an existing cluster

If CLS is not enabled for a ClickHouse cluster during cluster creation, you can enable it by selecting **Operation >**

More > Create Log Service in the cluster list. Note that you need to be authorized first before configuring a logset.

Create Log Service ×

1 **Activate and Authorize** > 2 **Select Logset**

i After you enable CLS, the system will charge you on a pay-as-you-go basis. View [Purchase Guide](#) for details.

Logset

Select an existing logset, or [create a logset](#)

Note: To ensure network interconnectivity, ensure the CLS service and cluster are in the same region.

OK **Cancel**

Using Log Search

1. Log in to the [Cloud Data Warehouse ClickHouse console](#), click a **Cluster ID/Name** in the **Cluster List** to enter the cluster details page, and switch to the **Log Search** tab.
2. The **Page Mode** can be **Node log** or **Search**. The **Log Type** can be **Running log** or **Error log**. You can view logs of each node as needed.

Scale Out Scale Up/Down Terminate

Restore Account Management Data Table Management **Log Search** Query Management Data Dictionaries Operation Log

Page Mode **Node log** Search Node IP 10.0.0.12 Log Type **Running log** Error log

Time Range **Today** Last 7 days Last 15 days Last 30 days 2022-02-23 00:00:00 ~ 2022-02-23 20:12:43

Auto Line Feed Auto Refresh Keyword Supports AND, OR **Search All Nodes**

Current node: 10.0.0.12

```
PrometheusHandler-factory. Method: GET, Address: 10.0.0.12:60612, User-Agent: Go-http-client/1.1, Content Type: ,
Transfer Encoding: identity, X-Forwarded-For: (none)
2022.02.23 20:12:12.163526 [ 2785 ] {} <Debug> DNSResolver: Updating DNS cache
2022.02.23 20:12:12.163569 [ 2785 ] {} <Debug> DNSResolver: Updated DNS cache

[Tips]: ----- Above is the newly loaded data (100 entries) -----
```

3. You can query logs by keyword. The results are grouped by node IP. The last 100 logs are displayed for each node by default.

Scale Out Scale Up/Down Terminate

Restore Account Management Data Table Management **Log Search** Query Management Data Dictionaries Operation Log

Page Mode **Node log** Search Log Type **Running log** Error log

Time Range **Today** Last 7 days Last 15 days Last 30 days 2022-02-23 00:00:00 ~ 2022-02-23 20:15:39

Auto Refresh Keyword Supports AND, OR **Search All Nodes**

Search keyword: empty, found 2 nodes, 200 logs (up to the first 100 logs are displayed per node)

Node List	Time	Log Content
10.0.0.12	▶ 2022-02-23 20:15:13	2022.02.23 20:15:12.165416 [2781] {} <Debug> DNSResolver: Updated DNS cache
10.0.0.8	▶ 2022-02-23 20:15:13	2022.02.23 20:15:12.165374 [2781] {} <Debug> DNSResolver: Updating DNS cache

Data Dictionary

Last updated : 2024-01-19 16:45:30

Feature Overview

ClickHouse offers [internal dictionaries](#) and [external dictionaries](#). The former predefines the content and the latter offers various definitions from external data sources. You can use special functions along with a dictionary for query, which is simpler and more efficient than the combination of `JOIN` and a reference table. This document describes how to use external dictionaries in the console.

Notes

1. A data dictionary maintained in the console will take effect on all ClickHouse nodes.
2. A data dictionary added in the console can only be maintained (modified and deleted) in the console.
3. A data dictionary added using SQL commands or other backend methods cannot be maintained (modified and deleted) in the console.
4. A dictionary table created in the console will be added to the Cloud Data Warehouse global database `cdwch_{instanceId}_dictionary_database` by default.

Note:

Do not maintain a dictionary via both the console and other means (such as SQL commands and other backend means). Otherwise, content may be overwritten or lost (such as due to accidental deletion).

Directions in Console

1. Log in to the [Cloud Data Warehouse console](#), select the target cluster in **Cluster List**, and click the **Data Dictionaries** tab on the cluster details page.
2. View the list of existing external dictionaries on the current page.

←
Restore Account Management Data Table Management Log Search Query Management

i 1. The data dictionaries below take effect on all ClickHouse nodes.
 2. For a data dictionary added via the console, you can maintain (modify, delete, etc.) it only via the console.
 3. For a data dictionary added using other methods (such as SQL commands or other backend methods), you cannot maintain
 4. A new dictionary table successfully created via the console will be added to the global database `cdwch_7zijmba6_dictionary` by default.

! Do not maintain a dictionary via both the console and other means (such as SQL commands and other backend means). Other to accidental deletion).

Add Dictionary

Dictionary Name	Status	Data Source	layout	
No data yet				

3. Click **Create Dictionary** in the top-left corner, set the fields as prompted, and click **OK** to create a dictionary.

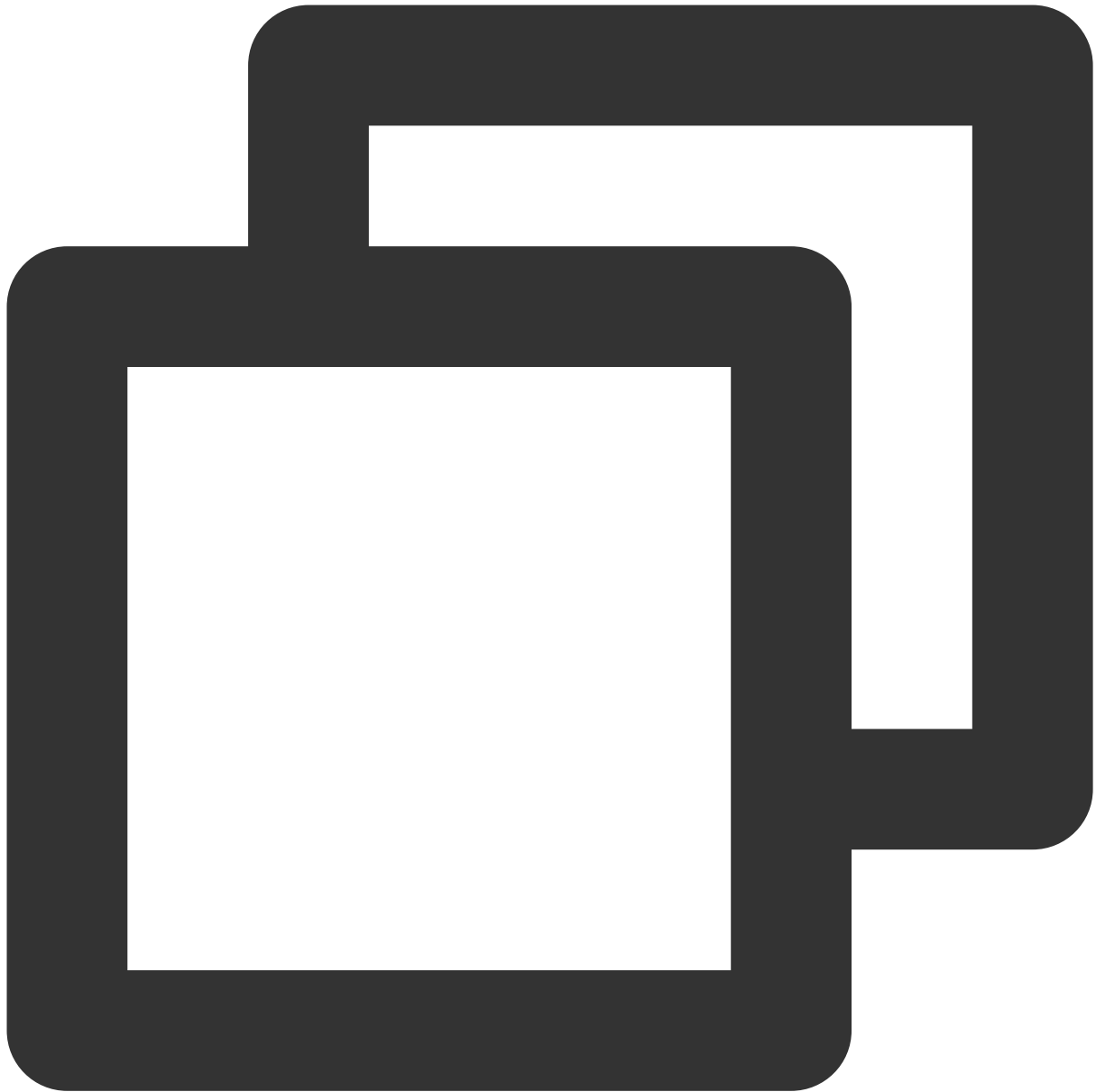
Parameter	Description	Reference
Dictionary Name	Enter a custom external dictionary name, which can contain 2–16 lowercase letters, digits, and underscores and must start with a letter and end with a letter or digit.	-
Data Source	Select the type of data source for the external dictionary. Currently, the Cloud Data Warehouse console supports the MySQLClickHouse data source.	Sources of External Dictionaries
Data Source Connection	Configure data source information to verify the connectivity of the configured external source. Configuration items include: HOST: IP address or domain name. Currently, only VPC connections are supported. TCP PORT: TCP port USER: external source account PASSWORD: password of the external source account	-
Source Table Information	Select the databases and tables of the dependent external source and enter `WHERE` (table filter condition) and `INVALIDATEQUERY` (for querying/checking the dictionary status, with only the updated data extracted).	Dictionary Updates
Data	Set the primary key and general fields of the external dictionary, including:	Dictionary

Structure	<p>PRIMARY_KEY: single or composite primary key.</p> <p>COLUME_NAME: field type. Currently, the console supports the following types: UInt8, UInt16, UInt32, UInt64, Int8, Int16, Int32, Int64, Float32, Float64, Date, DateTime, UUID, and String.</p> <p>DEFAULT_VALUE: default value for empty fields</p> <p>EXPRESSION: expression to describe fields (if applicable)</p> <p>IS_HIERARCHICAL: indicates the support for hierarchy. The default value is `false`.</p> <p>S_INJECTIVE: indicates the inline mapping "id -> attribute". The default value is `false`.</p>	Key and Fields
Storage Format	<p>Select the type of memory layout for the external dictionary, including:</p> <p>FLAT: stores the entire dictionary in memory as a flat array, which is suitable for a single primary key.</p> <p>HASHED: stores the entire dictionary in memory as a hash table, which is suitable for a single primary key.</p> <p>RANGE_HASHED: stores the entire dictionary in memory as a hash table. It comes with an ordered array of ranges and corresponding values, which is suitable for a single primary key. You need to set the fields to represent the ranges.</p> <p>CACHE: stores the entire dictionary in a cache with a certain number of cells, which is suitable for a single primary key. You need to set the cache size.</p> <p>COMPLEX_KEY_HASHED: similar to HASHED and suitable for composite primary keys.</p> <p>COMPLEX_KEY_CACHE: similar to CACHE and suitable for composite primary keys. You need to set the cache size.</p>	Storing Dictionaries in Memory
Update Interval	Set the frequency of updating data in the dictionary. The unit is s.	Dictionary Updates

3. After the external dictionary is created, you can view, modify, and delete it in the list.

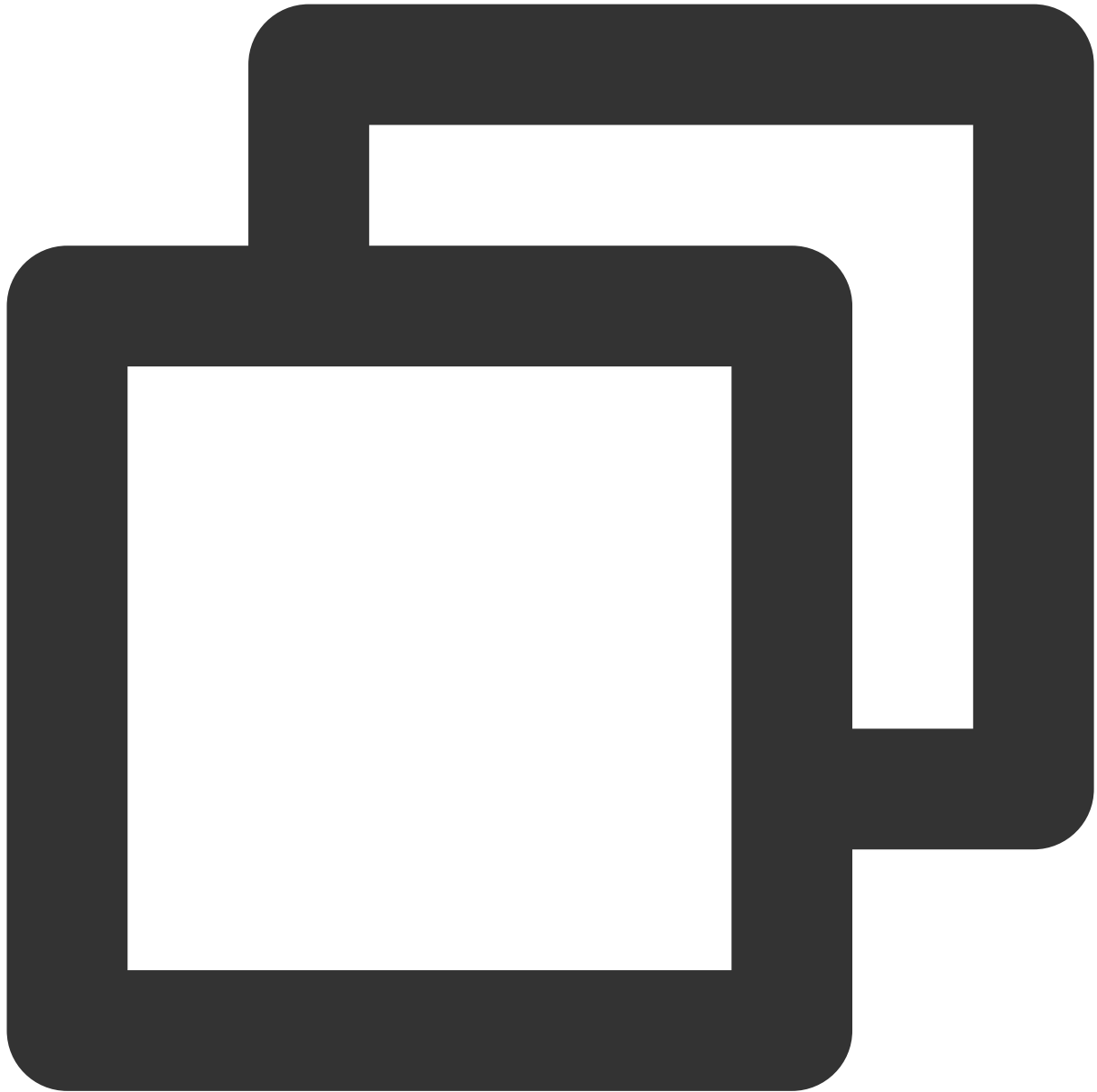
Managing External Dictionary via SQL

1. Create a dictionary.



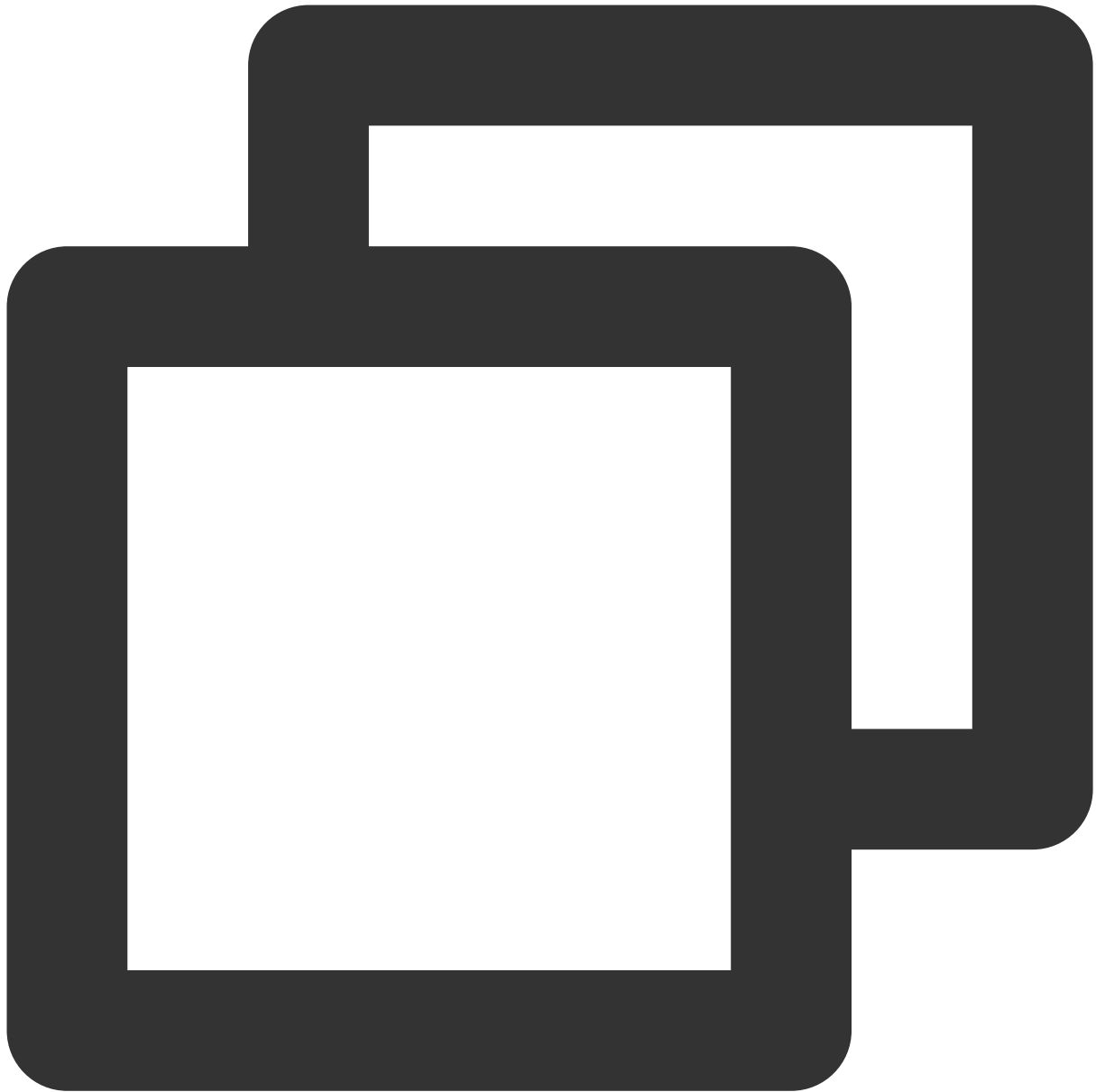
```
CREATE DICTIONARY [IF NOT EXISTS] [db.]dictionary_name [ON CLUSTER cluster]
(
  key1 type1 [DEFAULT|EXPRESSION expr1] [IS_OBJECT_ID],
  key2 type2 [DEFAULT|EXPRESSION expr2],
  attr1 type2 [DEFAULT|EXPRESSION expr3] [HIERARCHICAL|INJECTIVE],
  attr2 type2 [DEFAULT|EXPRESSION expr4] [HIERARCHICAL|INJECTIVE]
)
PRIMARY KEY key1, key2
SOURCE(SOURCE_NAME([param1 value1 ... paramN valueN]))
LAYOUT(LAYOUT_NAME([param_name param_value])``)
LIFETIME({MIN min_val MAX max_val | max_val})
```

2. View the dictionary.



```
SELECT * FROM system.dictionaries
```

3. Delete the dictionary.

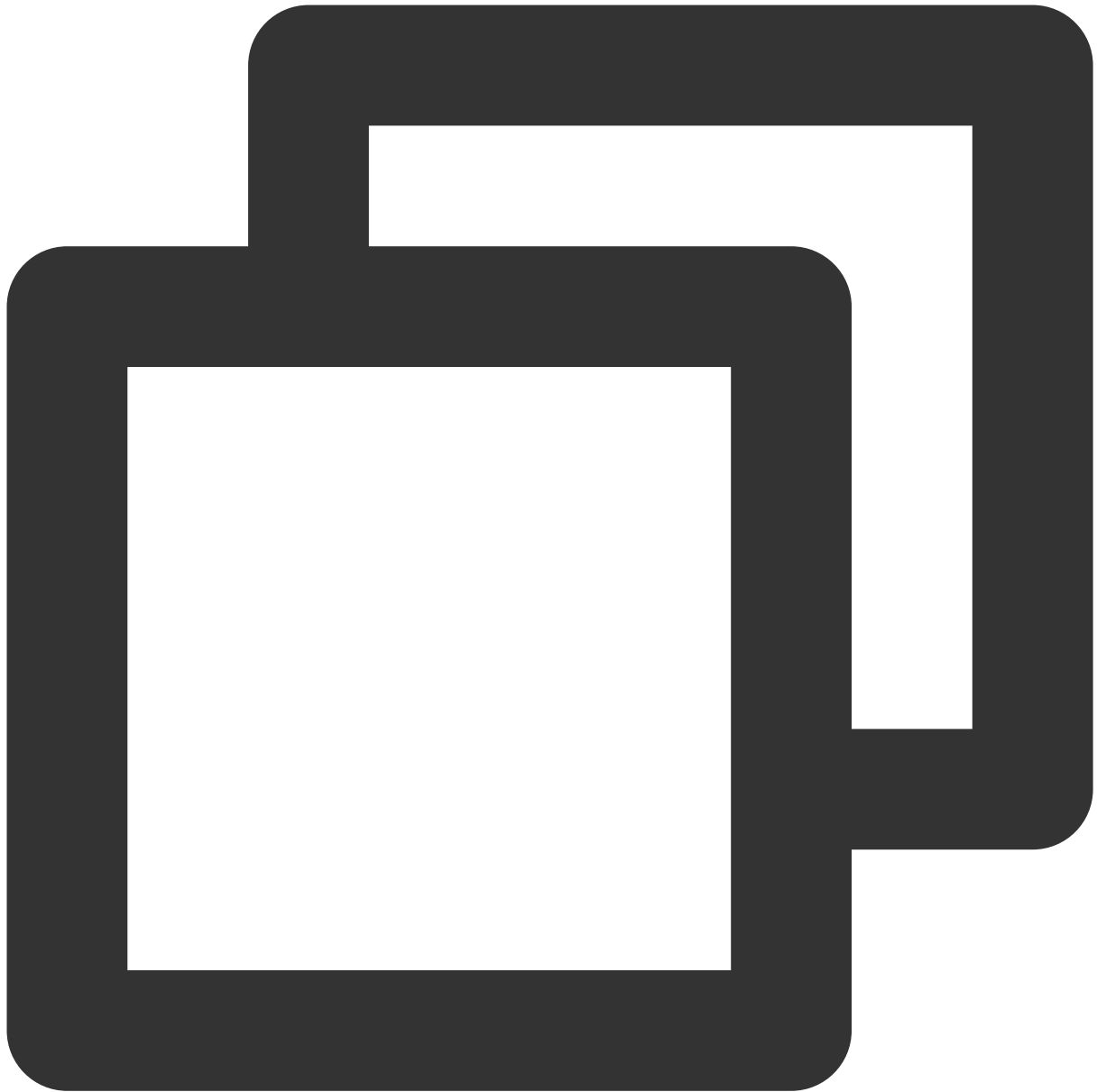


```
DROP DICTIONARY <database name>.<dictionary name>
```

For more information on dictionary SQL, see [CREATE DICTIONARY](#).

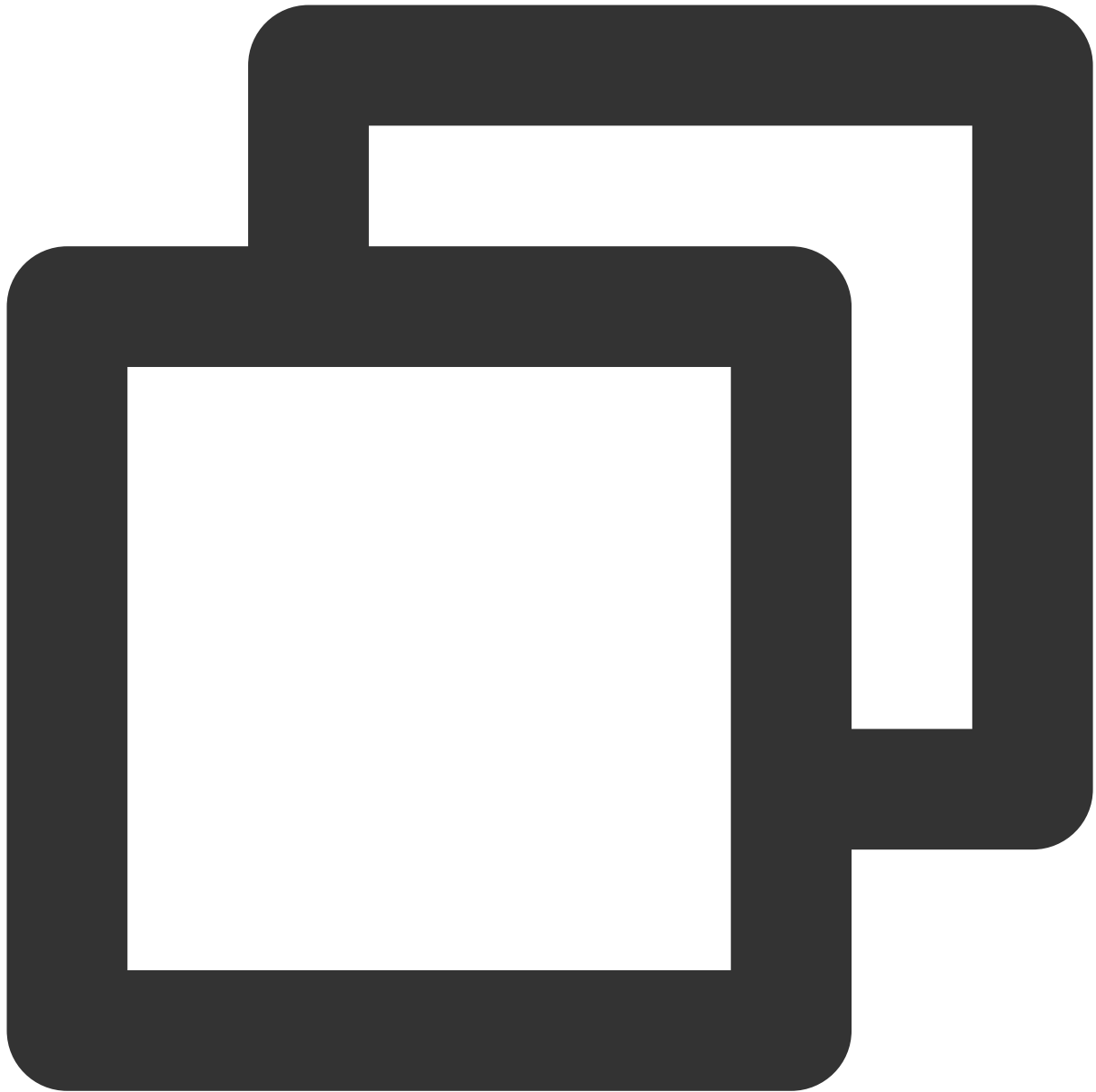
Using External Dictionary

You can query some types of dictionaries using general `SELECT` :



```
SELECT * FROM <database name>.<dictionary name>
```

You can also query them using dictionary functions:



```
dictGet('dict_name', attr_names, id_expr)
dictGetOrDefault('dict_name', attr_names, id_expr, default_value_expr)
dictGetOrNull('dict_name', attr_name, id_expr)
```

For more information on dictionary functions, see [Functions for Working with External Dictionaries](#).

Backup and Restore

Last updated : 2024-01-19 16:45:29

Feature Overview

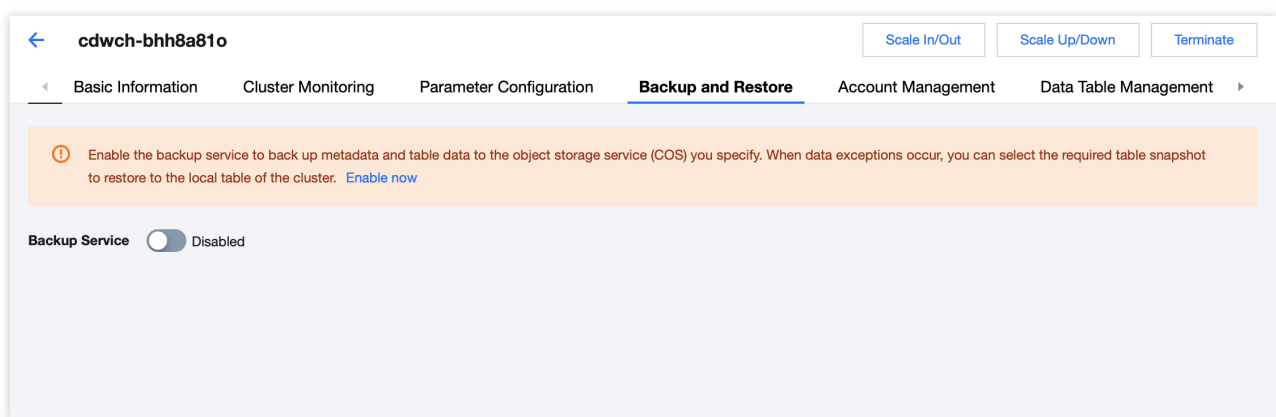
This document describes how to back up and restore ClickHouse cluster tables in the console. In particular, you can periodically back up metadata and data tables, where metadata can be backed up to COS to restore table structures. You can also back up table data and restore it with the backup snapshot as needed. Data will be restored to a new database without affecting the existing data.

Notes

1. During a backup, make sure that the cluster has a unique `v_cluster` that contains all nodes.
2. You can back up metadata several times per week and table data once a week at most.
3. You can restore data backups but not metadata backups. You can restore a table only once. Delete or rename the already restored table if you want to restore it again.
4. For ReplicatedMergeTree tables, you can only back up the data table of the first replica of each shard defined by `v_cluster`.
5. A data table will always be restored to a MergeTree local table.

Directions in Console

1. Log in to the [Cloud Data Warehouse console](#), select the target cluster in **Cluster List**, and click the **Backup and Restore** tab on the cluster details page.



2. Toggle on data backup in the top-left corner.

cdwch-bhh8a81o

Scale In/Out Scale Up/Down Terminate

Basic Information Cluster Monitoring Parameter Configuration **Backup and Restore** Account Management Data Table Management

Backup Service Enabled

Metadata Backup Policy

COS Bucket

Validity Period {{attr0}} day(s)

Backup Day Mon

Backup Time 01: 00 - 02: 00

Backup Content Full metadata and meta information

Data Backup Policy

No data backup policy. [Enable data backup](#)

Backup Management

Snapshot Generation Start Time: 2022-02-15 ~ 2022-02-22

Backup Task Name	Backup Status	Task Type	Backup Size	Snapshot Generation ...	Snapshot Expiration T...	Operation
No data yet						

3. Then, a metadata backup policy will be generated by default, which can be edited in the **Metadata Backup Policy** module.

Metadata Backup Policy

Only full metadata backup is supported.

COS Bucket

Backup Day Mon Tue Wed Thu Fri Sat Sun

Backup Time 01: 00 - 02: 00

Validity Period 7 day(s)

OK Cancel

4. You can also click a data backup policy to edit it. Configure the backup day, backup start time, and validity period, select the table to be backed up, and pay attention to the IP of the node where the data table resides.

Data Backup Configuration ✕

i You can configure data backup for a cluster for which compute is not separated from storage. Considering the large data size, we only support weekly backup.

Data backup selects one replica of table data in the shard, and the data will be restored to the selected backup replica node in the form of local table.

COS Bucket [Placeholder]

Backup Day Mon Tue Wed Thu Fri Sat Sun

Backup Time 01: 00 - 02: 00

Validity Period **i** - 7 + day(s)

Backup Scope

Select Data Table

🔍

<input type="checkbox"/> Database/Table	IP/Data Size
No data yet	

Selected: 0

Database/Table	IP/Data Size
Select a data table	

↔

5. After setting the table backup policy, you can click **View Details** to view and confirm the backup status.

Data Backup Details ✕

Table	Database	v_cluster	Ip
lineorder	default	default_cluster	10.0.0.41

6. A backup record will be generated for each periodic backup.

Backup Management

Snapshot Generation Start Time:

Backup Task Name	Backup Status	Task Type	Backup Size	Snapshot Generation ...	Snapshot Expiration T...	Operation ⓘ
	backup succeed	metadata	0B	2022-02-21 01:24:32...	2022-02-28 01:24:32	Restore Data View Delete
	backup succeed	data	1.75GB	2022-02-18 16:15:01...	2022-02-28 16:15:01	Restore Data View Delete

Data restoration and task deletion can be performed for backups. Data restoration will restore the backed up tables as local tables to a new database on the corresponding ClickHouse node in the IP dimension. Task deletion is only a soft deletion, where COS data will still be stored in the corresponding directory you select and need to be deleted manually.

Multi-ZooKeeper Cluster

Last updated : 2024-01-19 16:45:30

Overview

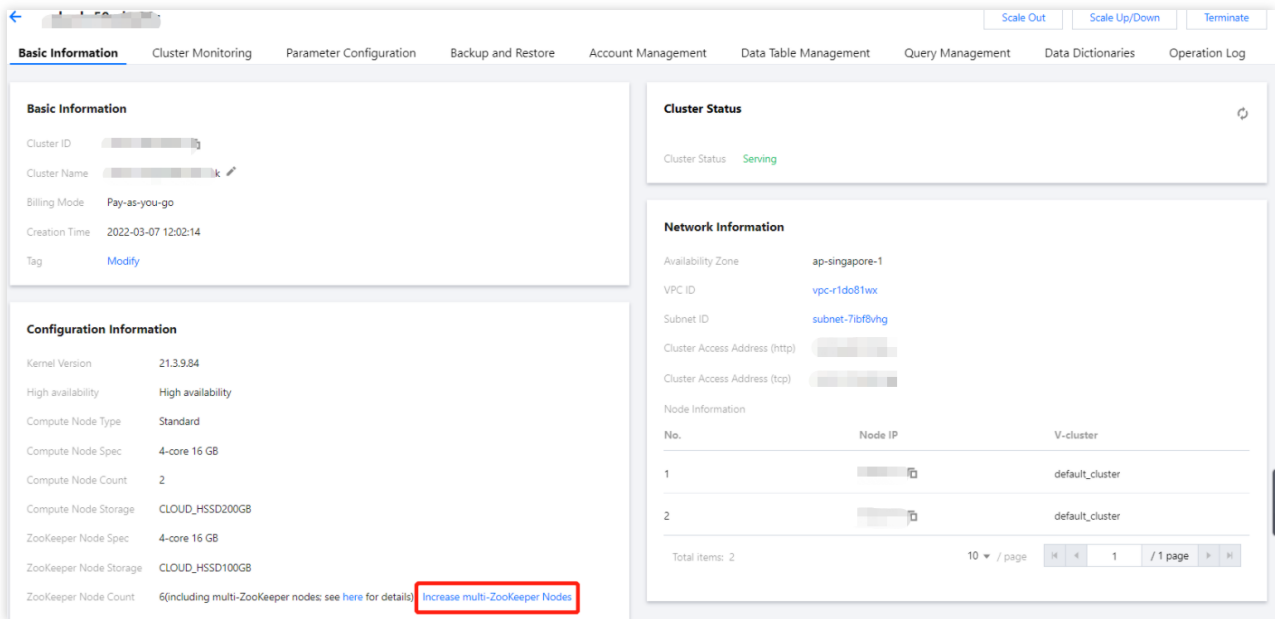
After you create an HA ClickHouse cluster on Tencent Cloud, a ZooKeeper cluster will be created by default to serve the ClickHouse cluster. However, if the load of the ZooKeeper cluster is too high, the ClickHouse cluster will experience write congestion or even a crash in a severe case. Therefore, Cloud Data Warehouse offers a multi-ZooKeeper solution, where you can configure several sets of ZooKeeper clusters to serve the ClickHouse cluster by load.

Notes

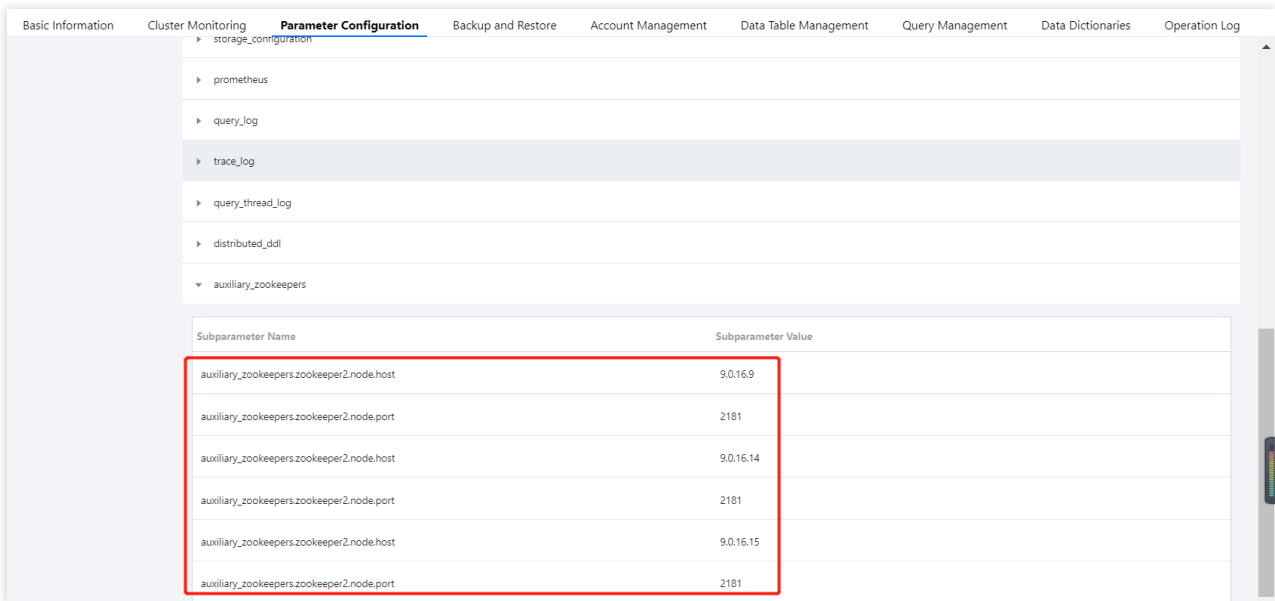
1. The multi-ZooKeeper solution applies only to HA clusters.
2. You can configure up to seven sets of additional ZooKeeper clusters.
3. All Zookeeper nodes in a cluster must be in the same spec (including the spec after scale-up/scale-down).
4. Additional ZooKeeper clusters differ from the default ZooKeeper cluster only in terms of table creation.

Directions

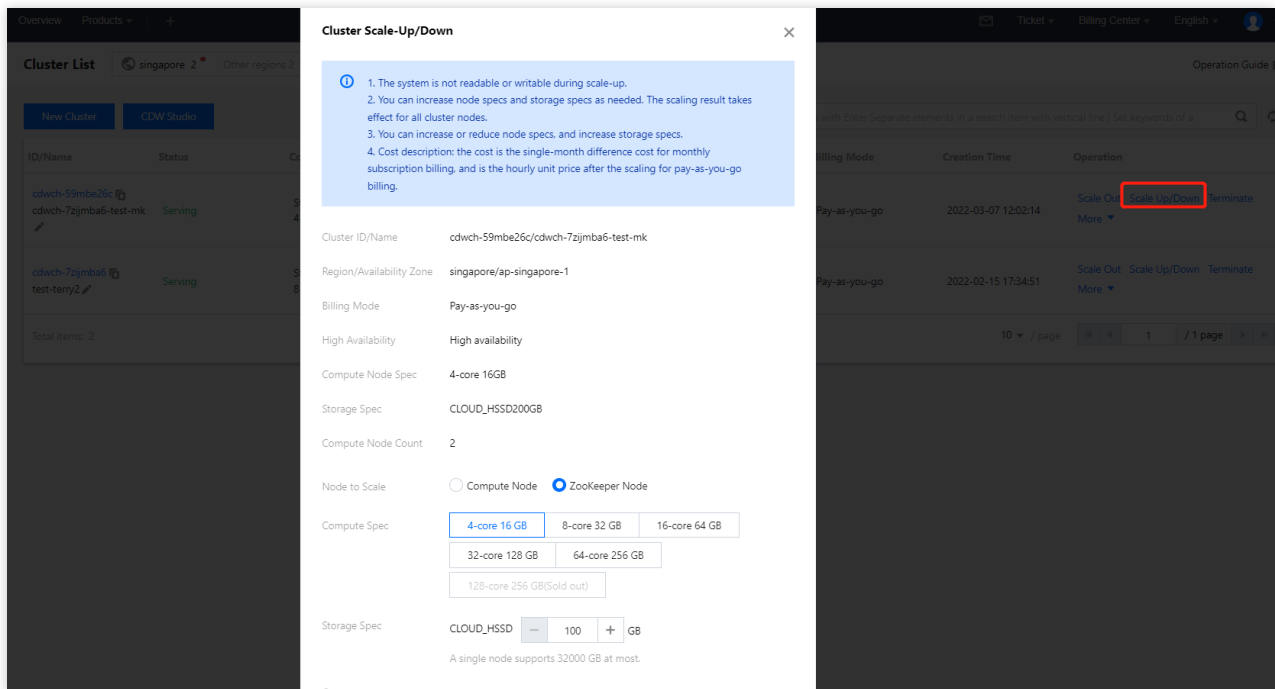
1. Log in to the [Cloud Data Warehouse console](#), select the target cluster in **Cluster List**, and click **Upgrade to Multi-ZooKeeper** on the cluster details page.



2. View the information on existing additional ZooKeeper clusters in the configuration file config.xml.



3. After you create an additional ZooKeeper cluster, you can scale up/down or expand the disk capacity of the ZooKeeper nodes in the cluster.



Using Multi-ZooKeeper

1. Create a database.



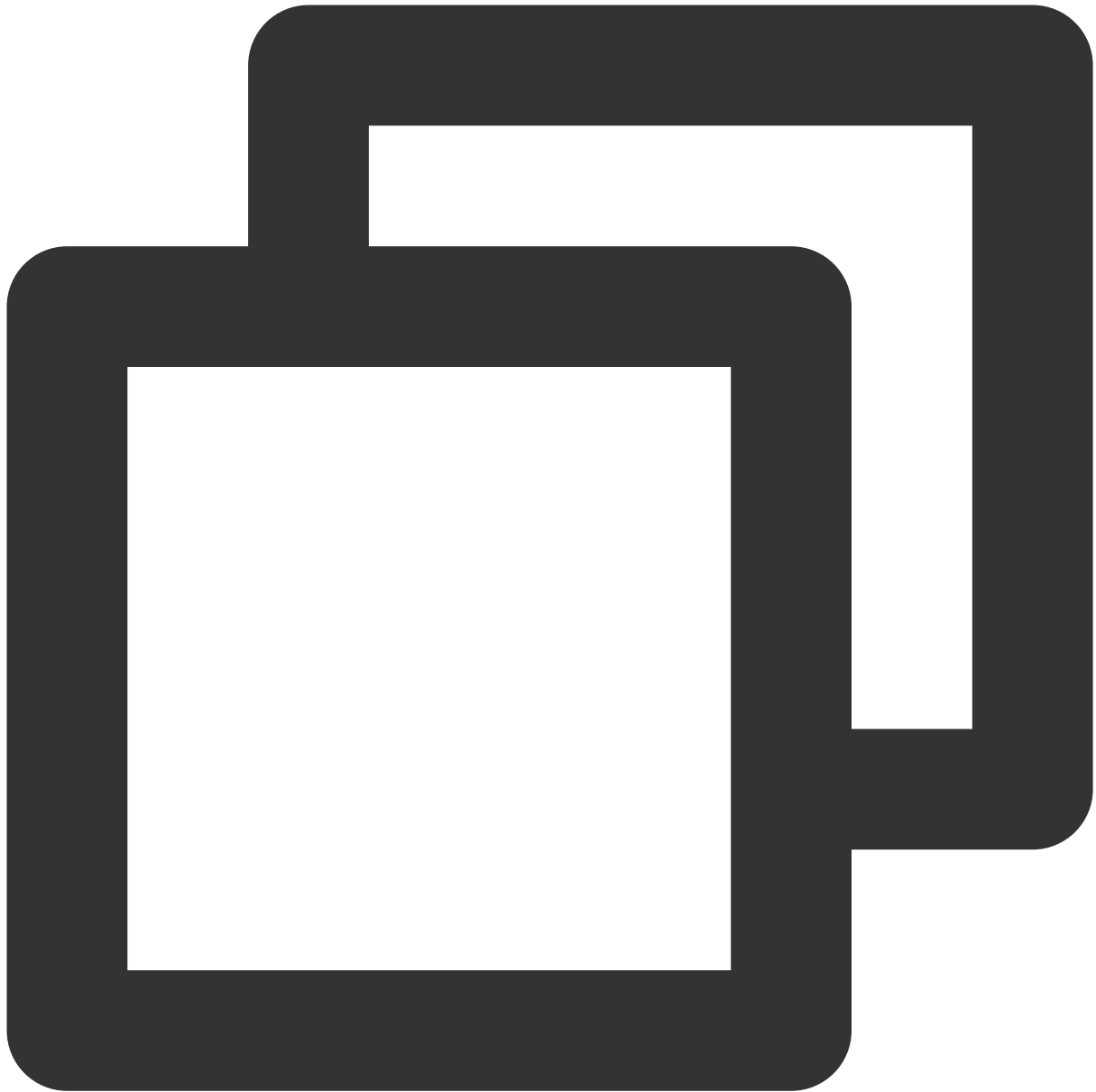
```
CREATE DATABASE IF NOT EXISTS testdb ON CLUSTER default_cluster;
```

2. Create a table that uses the default ZooKeeper cluster.



```
CREATE TABLE testdb.account ON CLUSTER default_cluster(accountid UInt16,name String
```

3. Create a table that uses an additional ZooKeeper cluster.



```
--Prefix the path with 'zookeeper2:' for a table that uses the second set of ZooKeeper
CREATE TABLE testdb.account2 ON CLUSTER default_cluster(accountid UInt16,name String)
```

External Data Import

Real-Time Import

Kafka Data import

Last updated : 2024-01-19 16:45:30

This document describes how to consume data from [Kafka](#) to Cloud Data Warehouse in real time.

Prerequisites

The Kafka data source cluster and the target Cloud Data Warehouse cluster must be in the same VPC.

Directions

1. Log in to the [Cloud Data Warehouse](#) cluster and create a Kafka consumption table.



```
CREATE TABLE queue (  
    timestamp UInt64,  
    level String,  
    message String  
) ENGINE = Kafka  
SETTINGS  
    kafka_broker_list = 'localhost:9092',  
    kafka_topic_list = 'topic',  
    kafka_group_name = 'group',  
    kafka_format = 'JSONEachRow',  
    kafka_num_consumers = 1,
```

```
kafka_max_block_size = 65536,
kafka_skip_broken_messages = 0,
kafka_auto_offset_reset = 'latest';
```

The common parameters are described as follows:

Name	Required	Description
kafka_broker_list	Yes	A list of Kafka service brokers separated by commas. We recommend you use `Ip:port` instead of domain name to avoid possible DNS resolution issues.
kafka_topic_list	Yes	Kafka topics separated by commas
kafka_group_name	Yes	Kafka consumption group name
kafka_format	Yes	Kafka data format. For information on ClickHouse-supported formats, see the parameters in Formats for Input and Output Data .
kafka_row_delimiter	No	Row delimiter used to split data rows. The default value is `\n`, but you can also set it to another value according to the actual segmentation format during data write.
kafka_num_consumers	No	Number of consumers for a single Kafka engine. You can increase the consumption data throughput by increasing this parameter value, but it cannot exceed the total number of partitions in the corresponding topic.
kafka_max_block_size	No	Block size of the target table to which Kafka data is written. It is 65536 bytes by default. If the data size exceeds this value, the data will be flushed.
kafka_skip_broken_messages	No	Number of data records with parsing exceptions that can be ignored. If the number of exceptions exceeds the specified value (`N`), the backend thread will stop. The default value is 0.
kafka_commit_every_batch	No	Frequency of Kafka commit execution. 0: commits only after the data of an entire block is written. 1: commits after the data of each batch is written.
kafka_auto_offset_reset	No	The offset from which to read Kafka data. Its value can be `earliest` or `latest`.

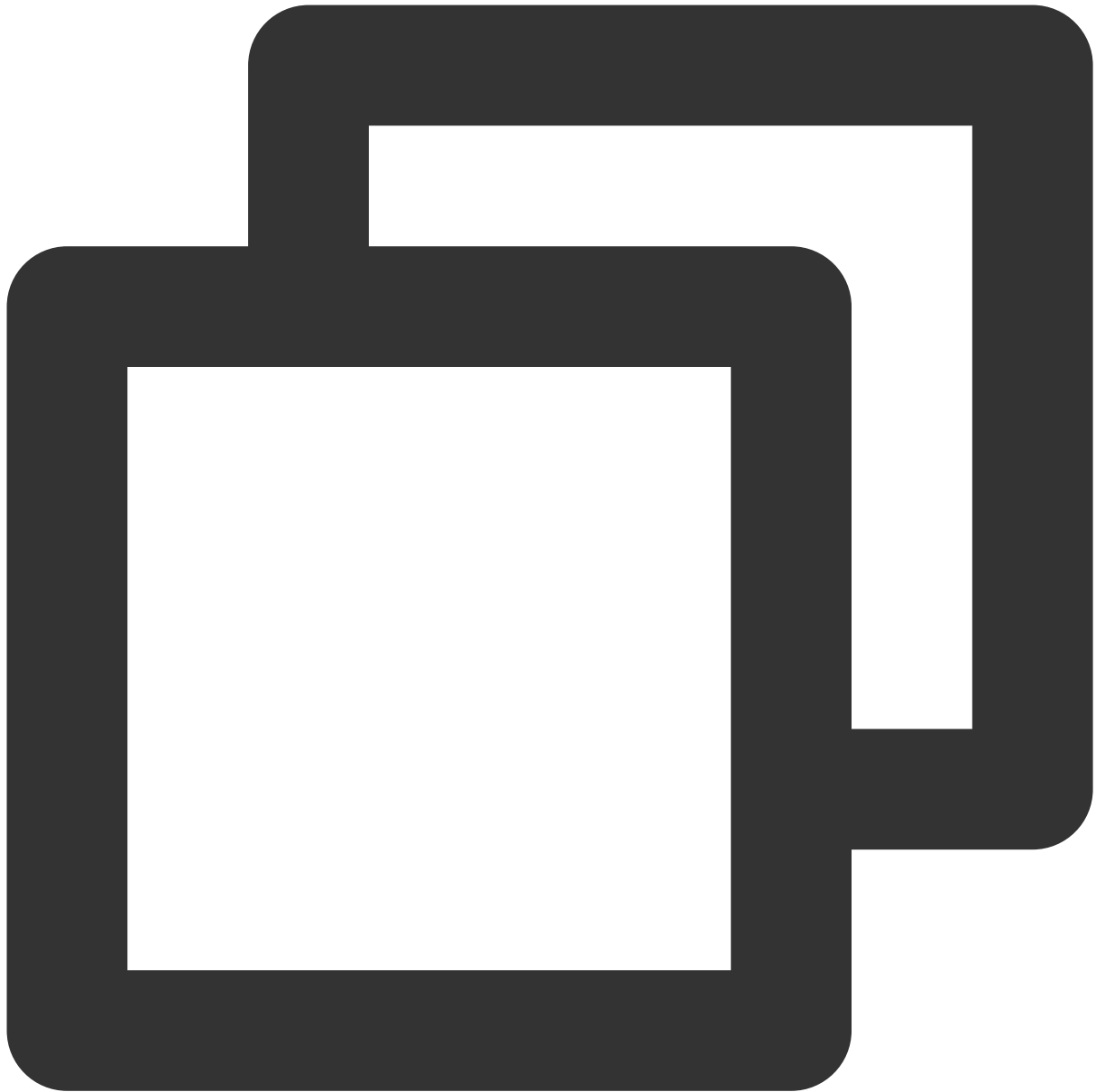
2. Create a ClickHouse local table (target table).

If your cluster has one replica:



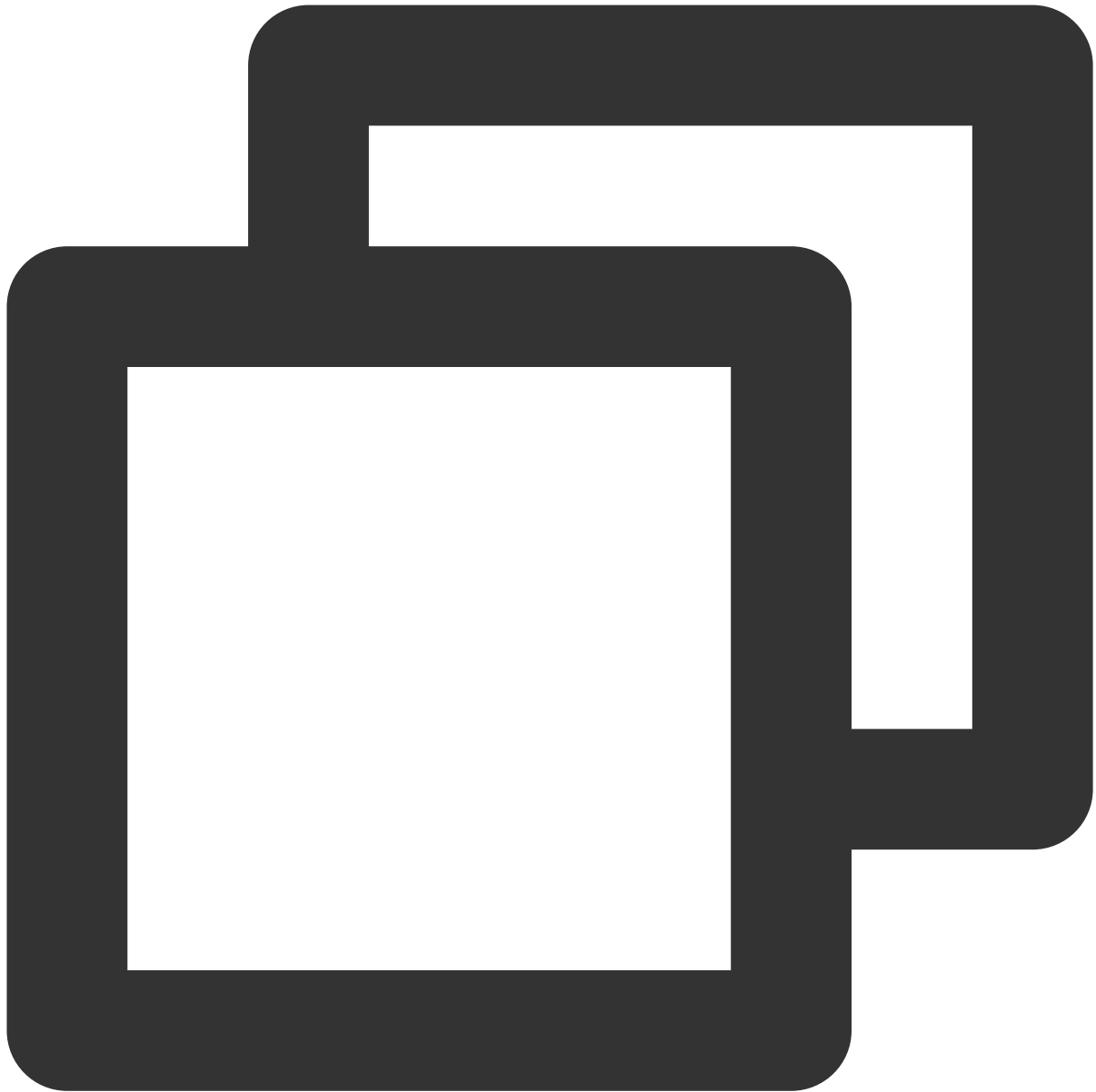
```
CREATE TABLE daily on cluster default_cluster
(
  day Date,
  level String,
  total UInt64
)
engine = SummingMergeTree()
order by int_id;
```

If your cluster has two replicas:



```
create table daily on cluster default_cluster
(
    day Date,
    level String,
    total UInt64
)
engine = ReplicatedSummingMergeTree('/clickhouse/tables/test/test/{shard}', '{repli
order by int_id;`
```

Create a distributed table:



```
create table daily_dis on cluster default_cluster
AS test.test
engine = Distributed('default_cluster', 'default', 'daily', rand());
```

3. Create a materialized view to sync data consumed by the Kafka consumption table to the ClickHouse target table.



```
CREATE MATERIALIZED VIEW consumer TO daily
AS SELECT toDate(toDateTime(timestamp)) AS day, level, count() as total
FROM queue GROUP BY day, level;
```

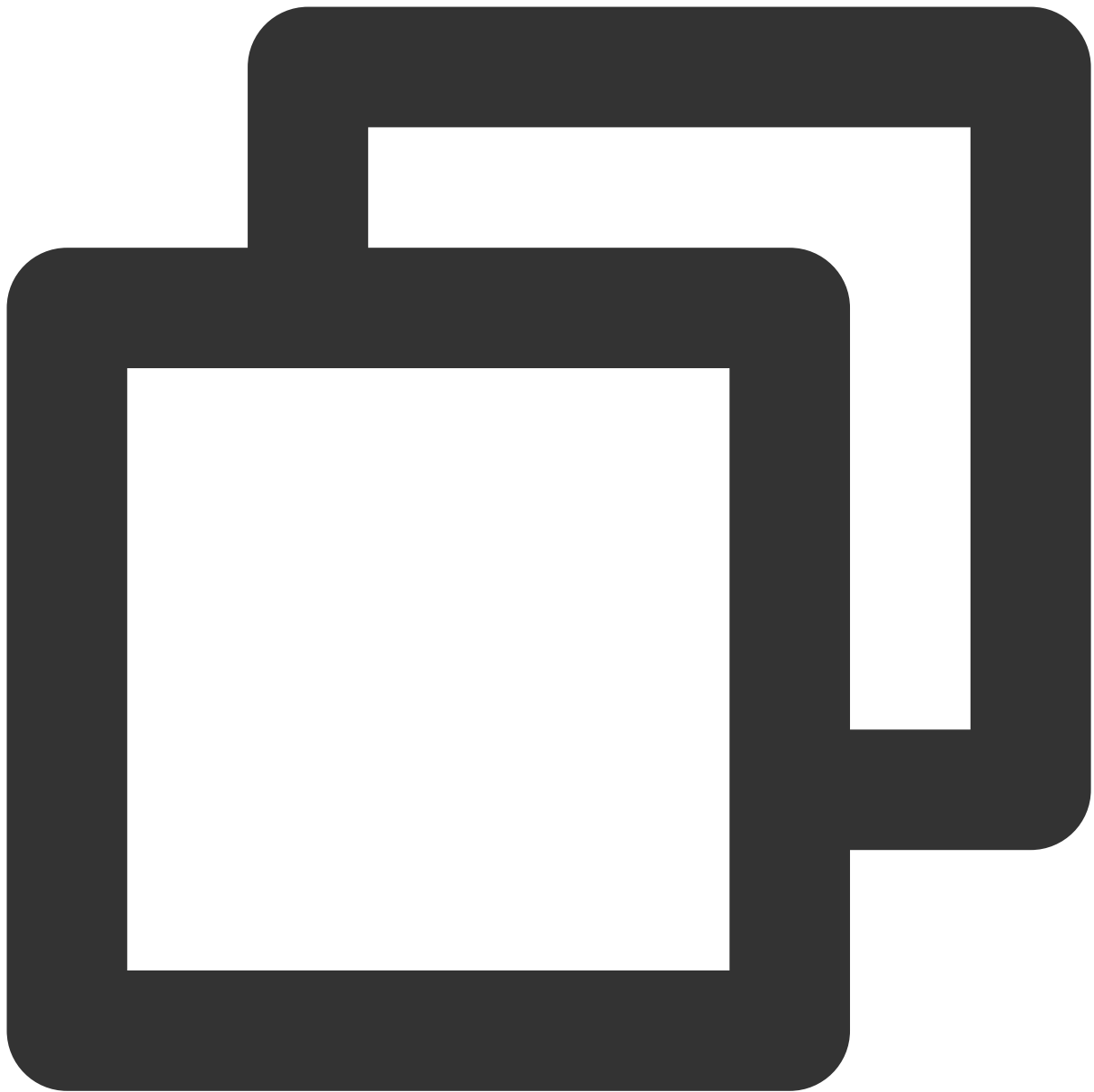
4. Query the data.



```
SELECT level, sum(total) FROM daily GROUP BY level;
```

Notes

If you want to stop receiving topic data or change the conversion logic, perform `detach` and `attach` view operations.



```
DETACH TABLE consumer;  
ATTACH TABLE consumer;
```

Offline Import

HDFS Data Import

Last updated : 2024-01-19 16:45:30

This document describes how to import data from HDFS to Cloud Data Warehouse.

Prerequisites

1. Read permissions of HDFS are required for HDFS data access. See [Access Control Overview](#) for how to set permissions.
2. The HDFS instance and Cloud Data Warehouse cluster must be in the same VPC.

Directions

1. Log in to Cloud Data Warehouse and create an HDFS table.



```
CREATE TABLE hdfs_engine_table
(
  `int_id` UInt32
)
ENGINE = ENGINE=HDFS('hdfs://hdfs1:9000/other_storage', 'TSV')
```

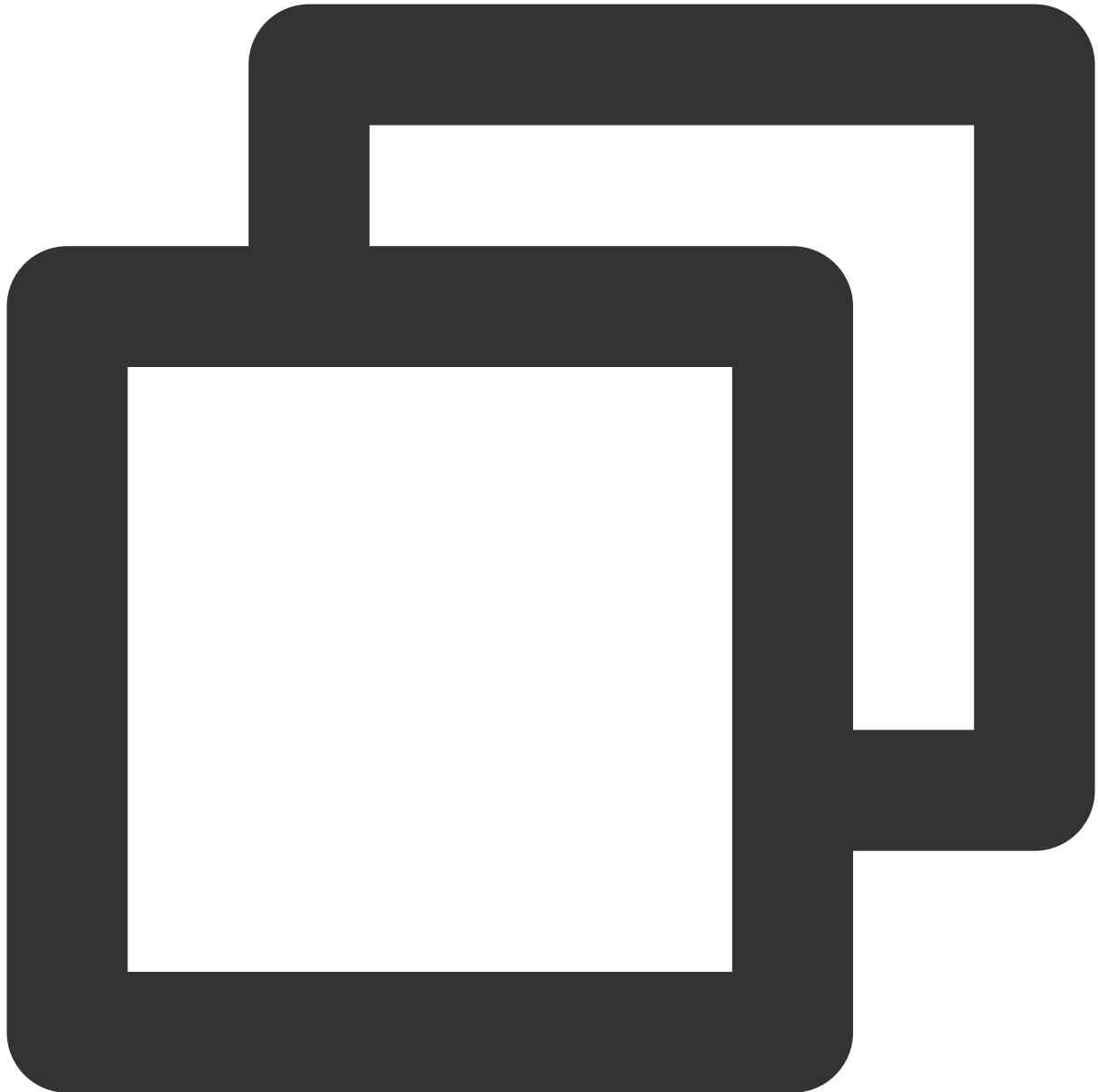
Reference

ENGINE = HDFS(URI, format)

`URI` is the URI of the entire file in HDFS, and `format` specifies an available file format. For more formats, see [Formats for Input and Output Data](#). A path URI may contain glob wildcards. In this case, the table will be read-only.

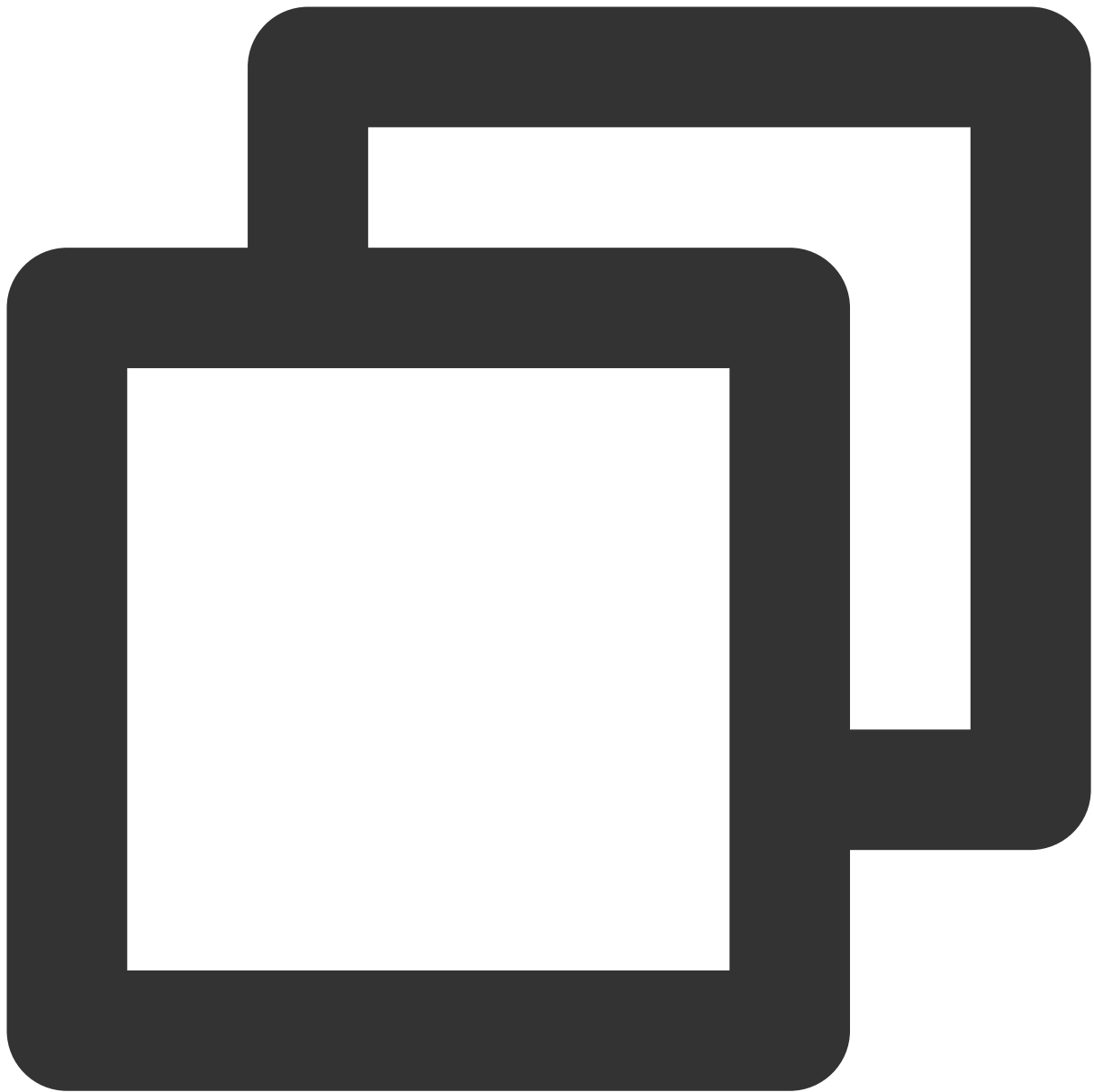
2. Create a ClickHouse target table.

If your cluster has one replica:



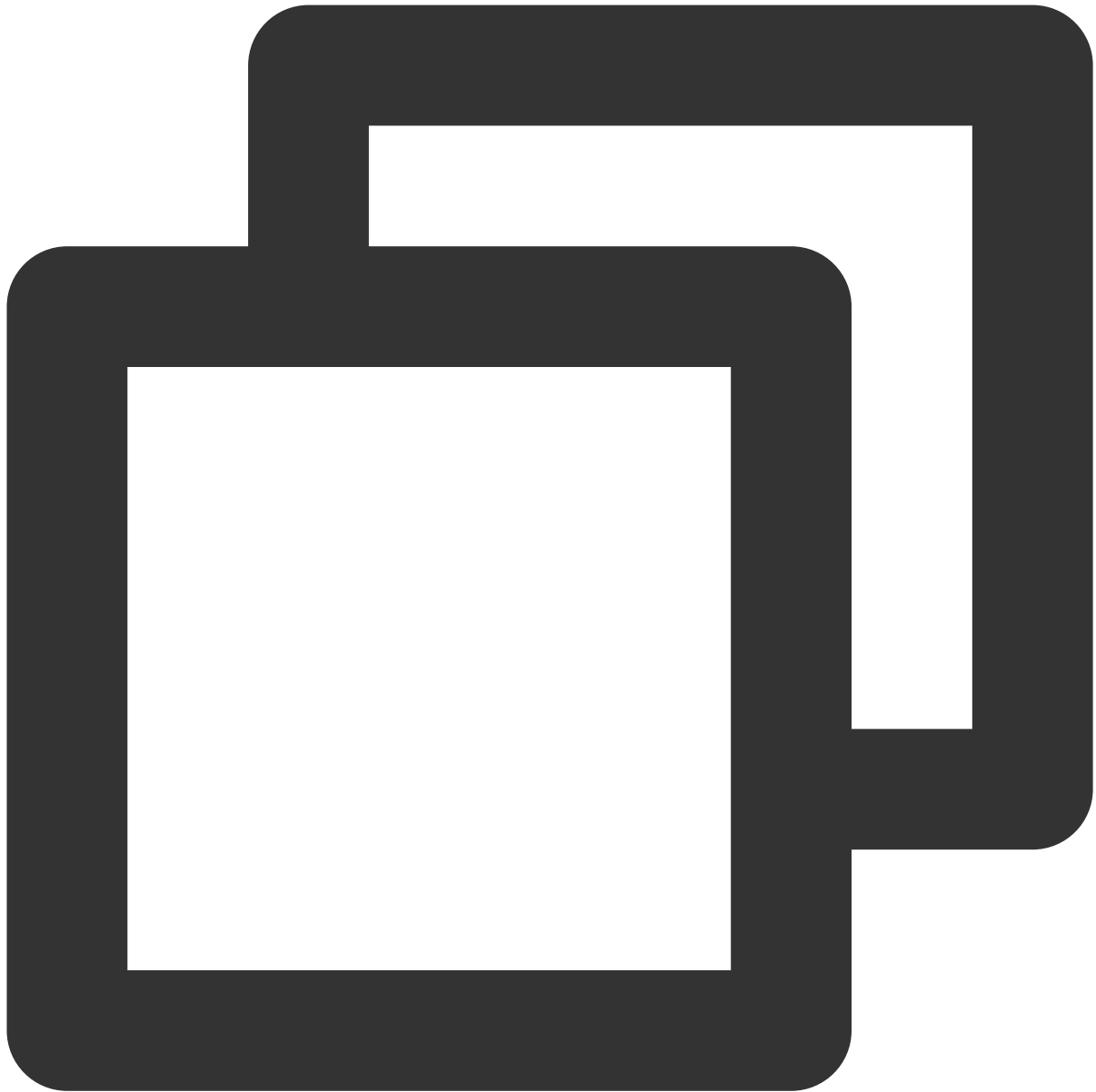
```
CREATE TABLE test.test on cluster default_cluster
(
  `int_id` UInt32
)
engine = MergeTree()
order by int_id;
```

If your cluster has two replicas:



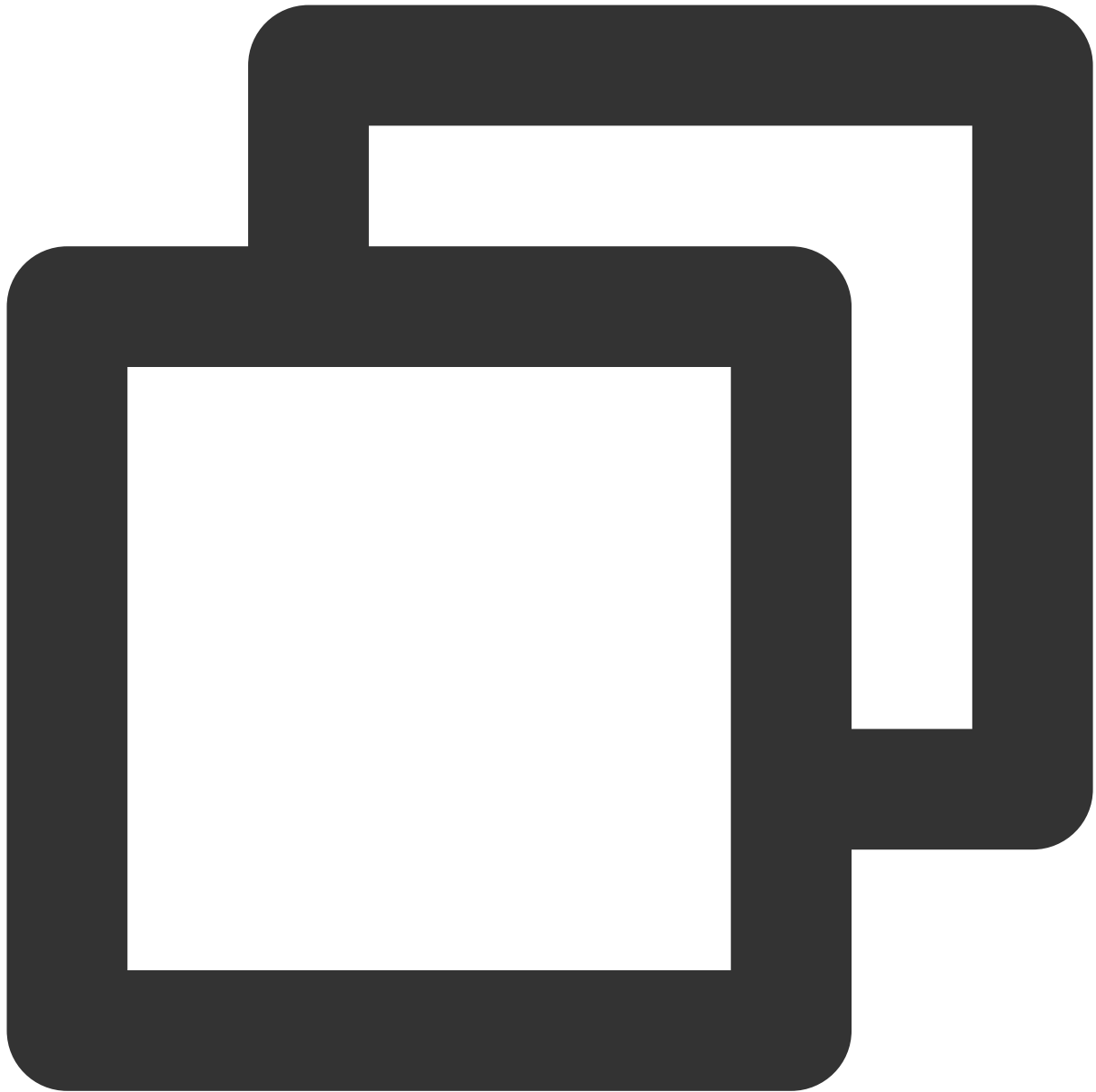
```
create table test.test on cluster default_cluster
(
  `int_id` UInt32
)
engine = ReplicatedMergeTree('/clickhouse/tables/test/test/{shard}', '{replica}')
order by int_id;
```

Create a distributed table:



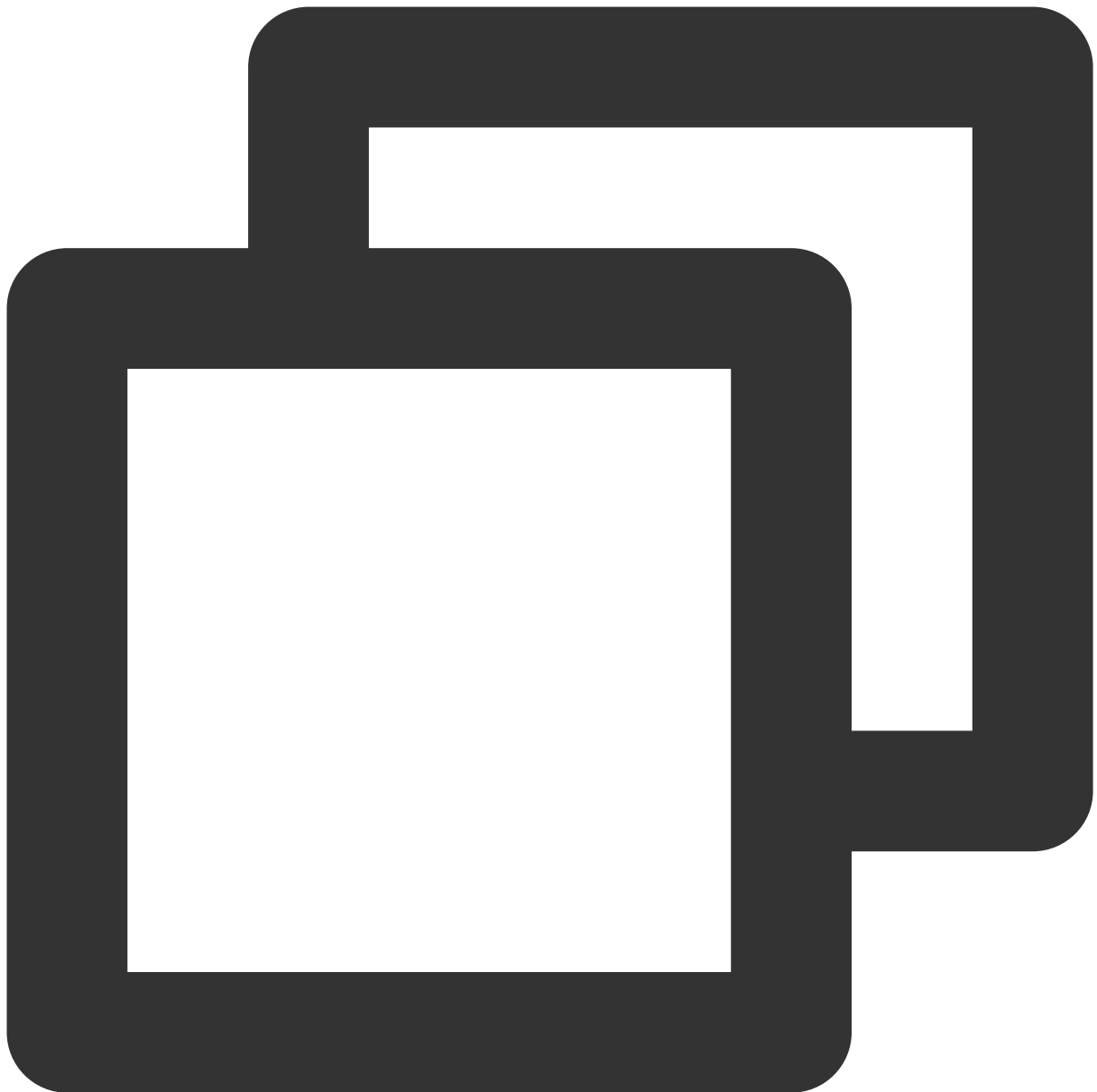
```
create table test.test_dis on cluster default
AS test.test
engine = Distributed('default_cluster', 'test', 'test', rand());
```

3. Write data to the target table.



```
INSERT INTO test.test SELECT * FROM hdfs_engine_table;
```

4. Query the data.



```
select * from test.test
```

COS Data Import

Last updated : 2024-01-19 16:45:30

This document describes how to import data from [COS](#) to Cloud Data Warehouse.

Prerequisites

1. The COS data source and the Cloud Data Warehouse cluster must be in the same VPC.
2. The `access-key-id` and `access-key-secret` entered in the table function must have permission to read the corresponding `oss-file-path`.
3. The `oss-file-path` parameter needs to meet the OSS path format specification, which is `oss://<bucket-name>/<path-to-file>` typically.

Directions

In the following example, data is imported from the S3 system (with COS as an example in this document) to Cloud Data Warehouse.

1. Log in to Cloud Data Warehouse and create an S3 table.



```
CREATE TABLE cos_engine_table
(
  `int_id` UInt32
)
ENGINE = S3('http://clickhouse-xxx.myqcloud.com/clickhouse-xxx/cos/data.csv.gz', 'C
```

Sample S3 engine parameters:

The S3 table engine provides integration with the Amazon S3 ecosystem. Its parameter follows the format of

```
S3(path, [aws_access_key_id, aws_secret_access_key,] format, [compression]) .
```

path: bucket URL with file path. In read-only mode, it supports *, ?, {abc,def}, and {N..M} wildcards, where N and M are digits and 'abc' and 'def' are strings.

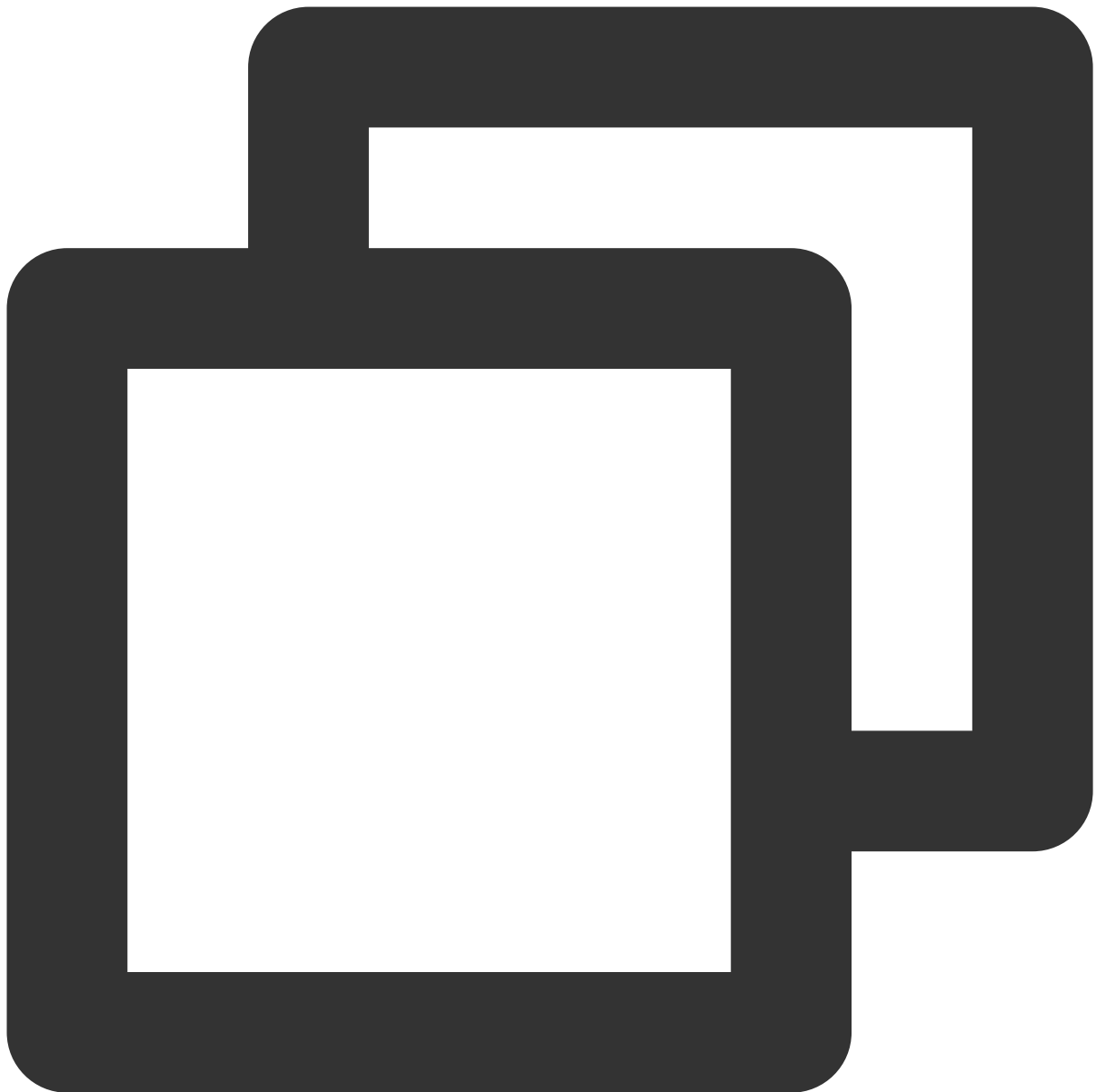
format: file format.

`aws_access_key_id` and `aws_secret_access_key` : permanent credentials of COS account. You can use them to authenticate your request. These parameters are optional. If no credentials are specified, credentials will be read from the configuration file. For more information, see [Using S3 to Store Data](#).

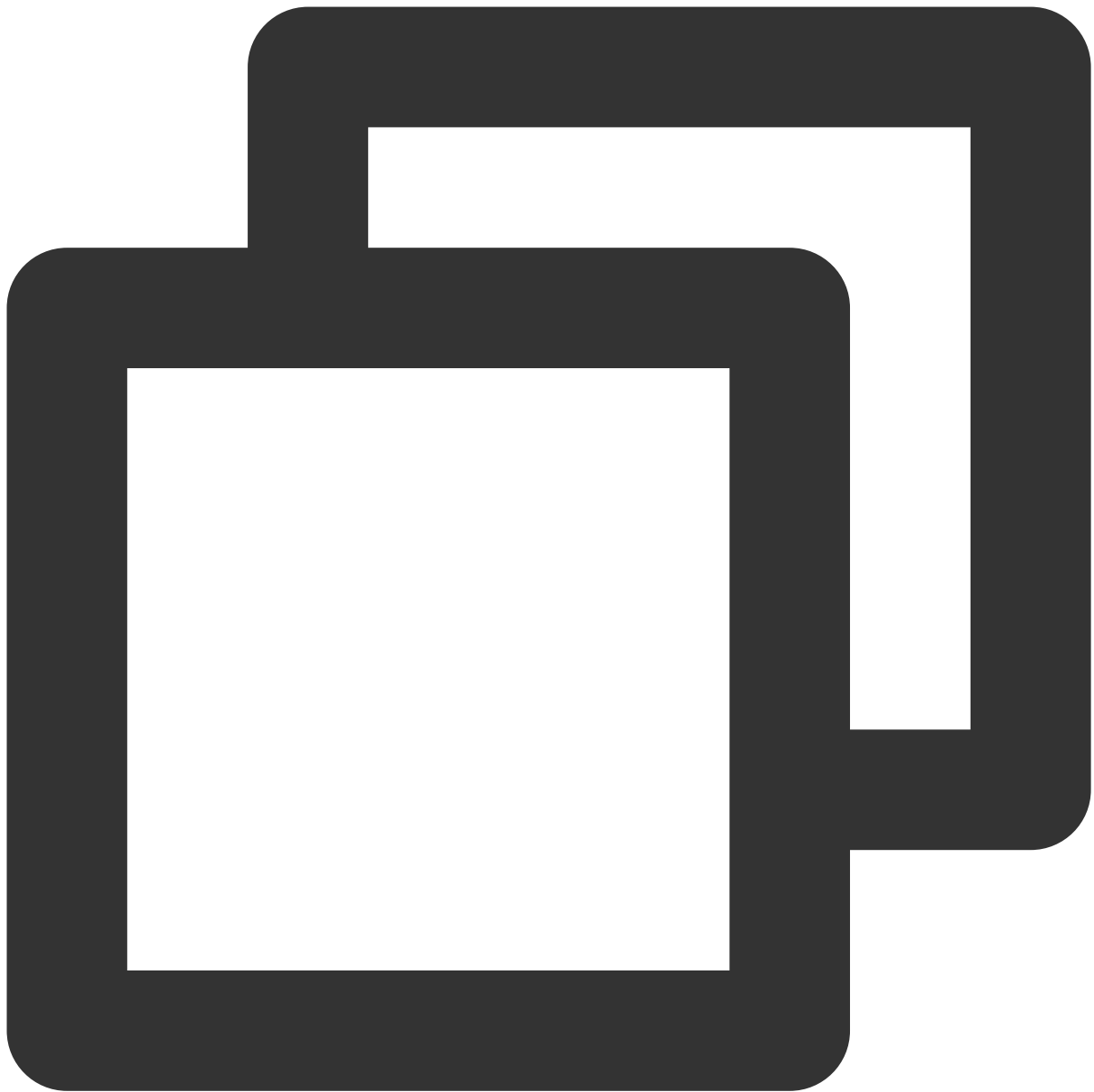
compression: compression type. Supported values are `none` , `gzip/gz` , `br/br` , `xz/LZMA` , and `zstd/zst` . This parameter is optional. The compression type is automatically detected based on the file extension by default.

2. Create a target table.

If your cluster has one replica:



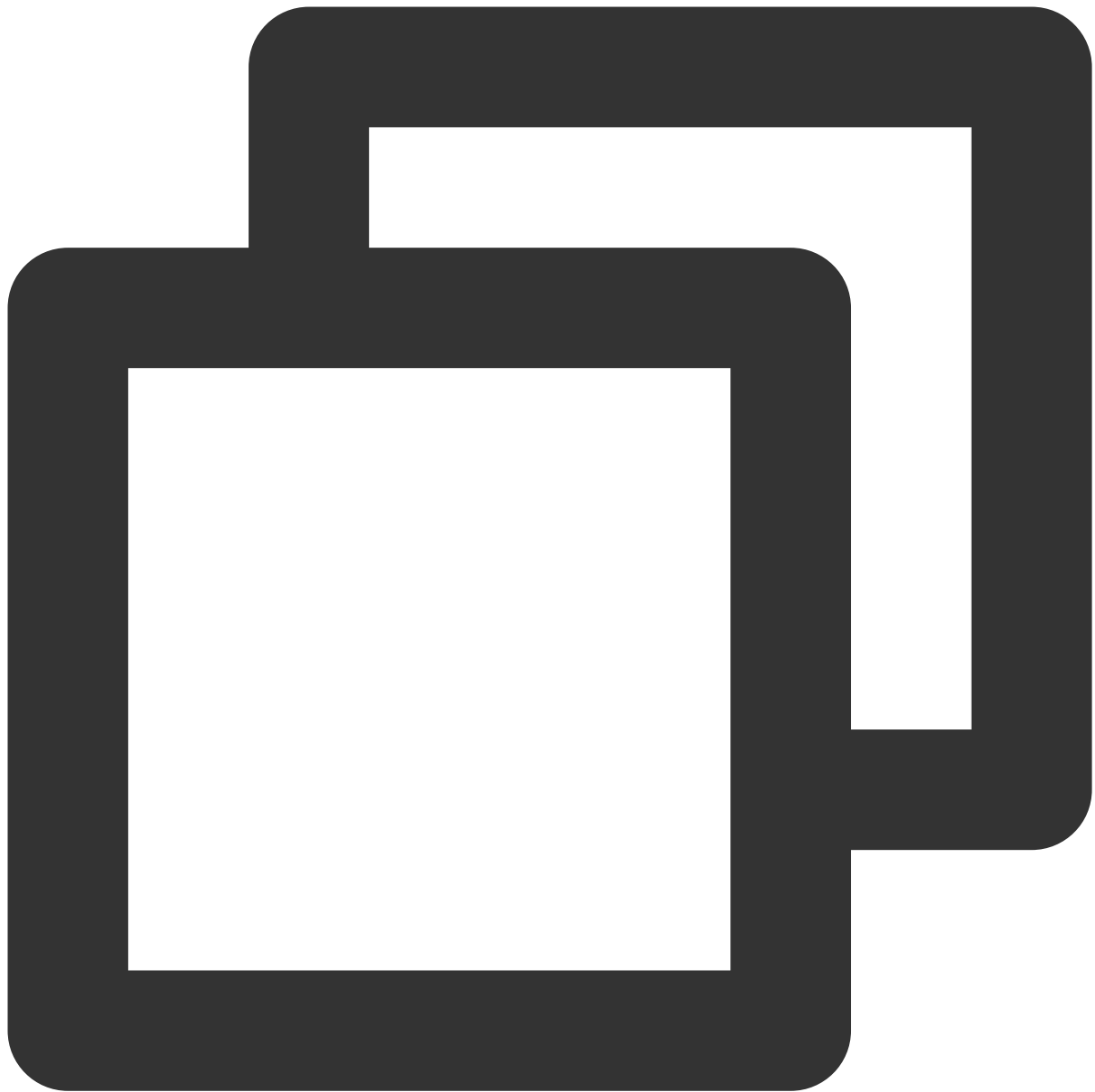
```
CREATE TABLE test.test on cluster default_cluster
(
  `int_id` UInt32
)
engine = MergeTree()
order by int_id;
```



- If your cluster has two replicas:



```
create table test.test on cluster default_cluster
(
`int_id` UInt32
)
engine = ReplicatedMergeTree('/clickhouse/tables/test/test/{shard}', '{replica}')
order by int_id;
```



- Create a distributed table:



```
create table test.test_dis on cluster default_cluster
AS test.test
engine = Distributed('default_cluster', 'test', 'test', rand());
```

3. Write data to the target table.



```
INSERT INTO test.test SELECT * FROM cos_engine_table;
```

4. Query the data.



```
select * from test.test
```

Local File import

Last updated : 2024-01-19 16:45:30

This document describes how to import data from a local file to Cloud Data Warehouse.

Prerequisites

1. The ClickHouse client has been installed. If not, you can [download](#) and install it.
2. The following common file formats can be imported to Cloud Data Warehouse: TabSeparated, TabSeparatedRaw, TabSeparatedWithNames, TabSeparatedWithNamesAndTypes, Template, CSV, and CSVWithNames. For more supported formats, see [Formats for Input and Output Data](#).
3. Make sure that the local server where the ClickHouse client is installed and the Cloud Data Warehouse cluster are in the same VPC.

Note:

Client and server versions are compatible, but some features may not be available on earlier clients. We recommend you use a client of the same version as the server. When you use a legacy client, the client on the server will display the following message: "ClickHouse client version is older than ClickHouse server. It may lack support for new features." In this case, import a file using the following command:

```
cat <data_file> | ./clickhouse-client
--host=<host> --port=<port> --user=<username> --password=<password> --query="INSERT
INTO <table_name> FORMAT <format>";
```

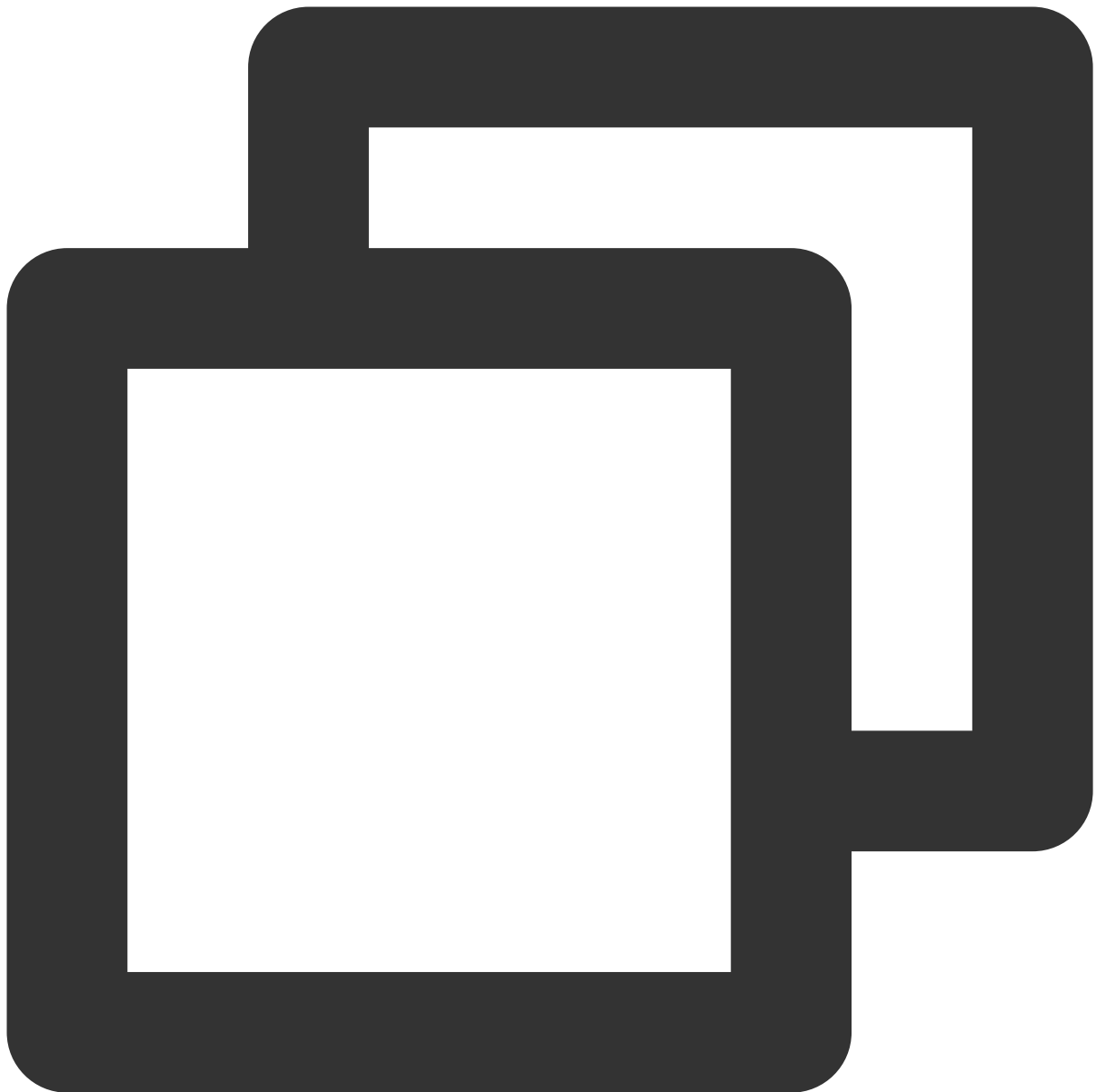
In batch mode, the default data format is TabSeparated. You can set a format flexibly based on the query.

By default, the batch mode supports only a single query. In order to execute multiple queries from a single script, use the `--multiquery` parameter, which is useful in most cases except for INSERT requests. The query results are output continuously without delimiters. Similarly, to execute large-scale queries, start a ClickHouse client for each query. Note that each startup takes tens of milliseconds.

Directions

The following is the entire process of importing the local file `test.csv`.

1. Prepare `test.csv` and write the following data.



```
$ cat /test.csv  
1,2,3  
3,2,1  
78,43,45
```

2. Create a ClickHouse target table.

If your cluster has one replica:



```
CREATE TABLE test.test on cluster default_cluster
(
  `column1` UInt32,
  `column2` UInt32,
  `column3` UInt32
)
engine = MergeTree()
order by column1;
```

If your cluster has two replicas:



```
create table test.test on cluster default_cluster
(
  `column1` UInt32,
  `column2` UInt32,
  `column3` UInt32
)
engine = ReplicatedMergeTree('/clickhouse/tables/test/test/{shard}', '{replica}')
order by column1;
```

Create a distributed table:



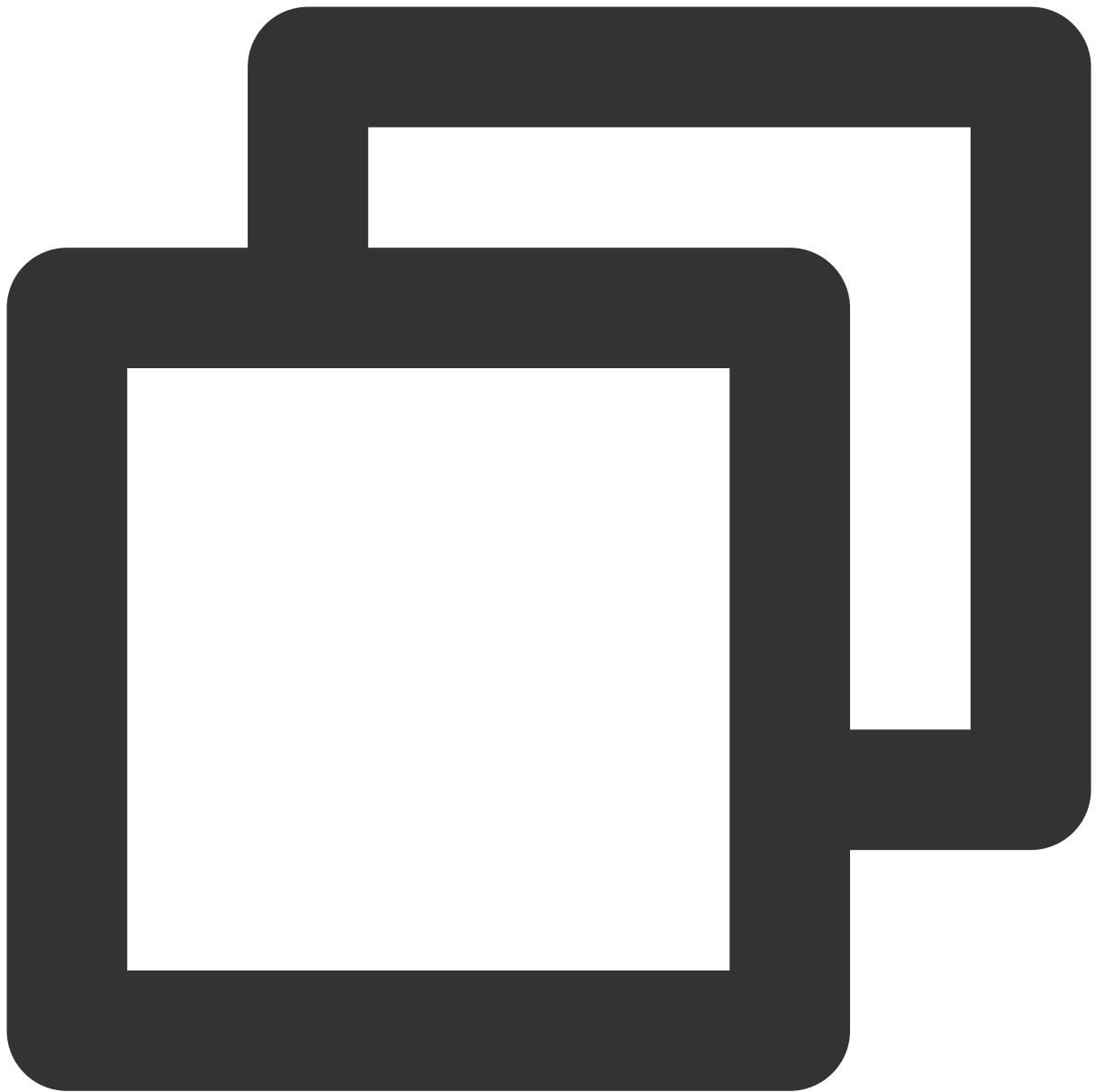
```
create table test.test_dis on cluster default
AS test.test
engine = Distributed('default_cluster', 'test', 'test', rand());
```

3. Write data to the target table.



```
cat data.csv | clickhouse-client --query="INSERT INTO test FORMAT CSV"
```

4. Query the data.



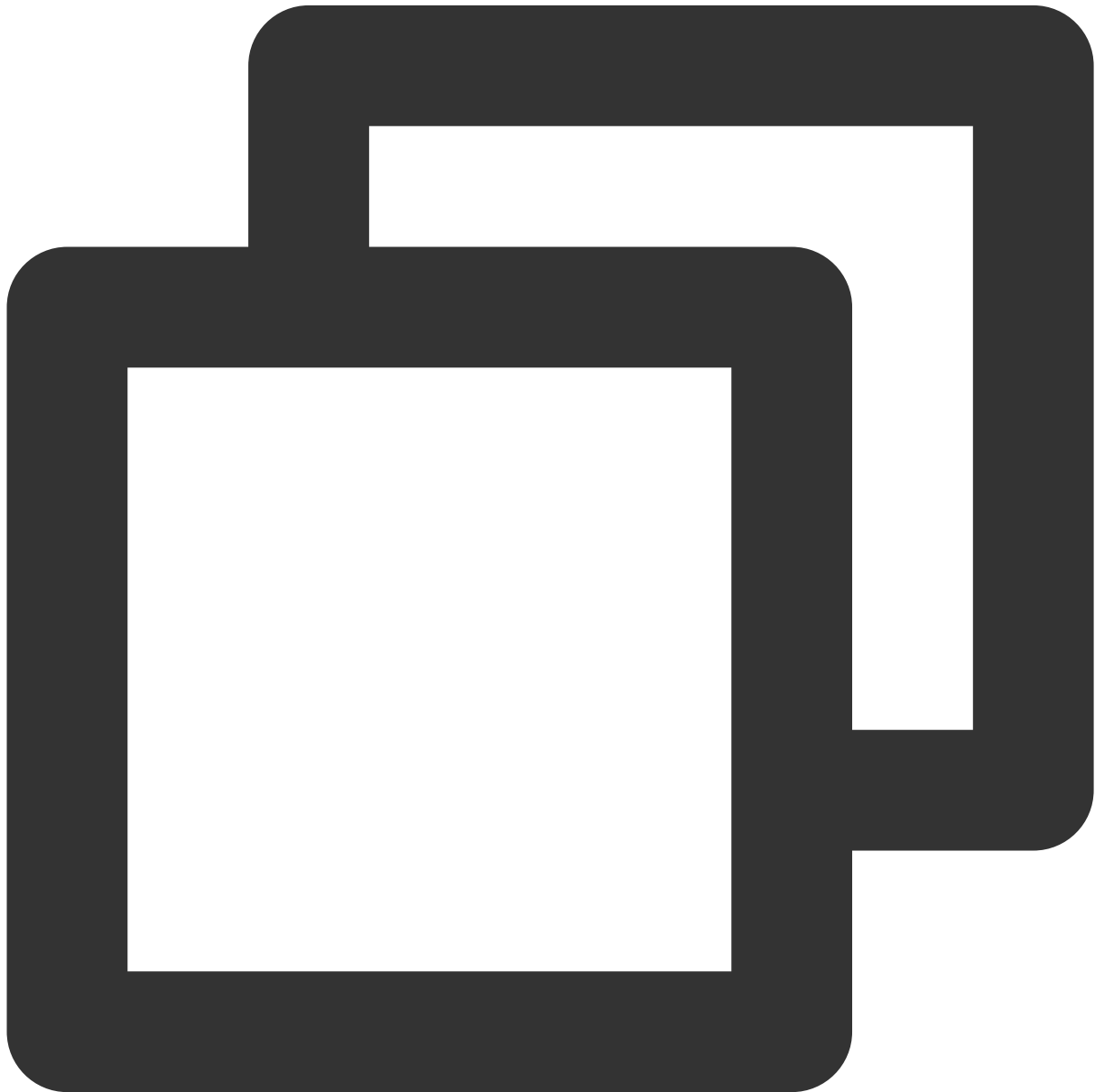
```
select * from test;
```

Configuring DDL on Cluster Feature

Last updated : 2024-01-19 16:45:30

By default, you need to manually add the `ON CLUSTER XXX` syntax when performing DDL operations such as table creation at the cluster level. CDWCH 21.8.8.12 offers the new feature of DDL On Cluster, which can automatically add a cluster when performing DDL operations. This feature is disabled by default.

New ClickHouse parameters:



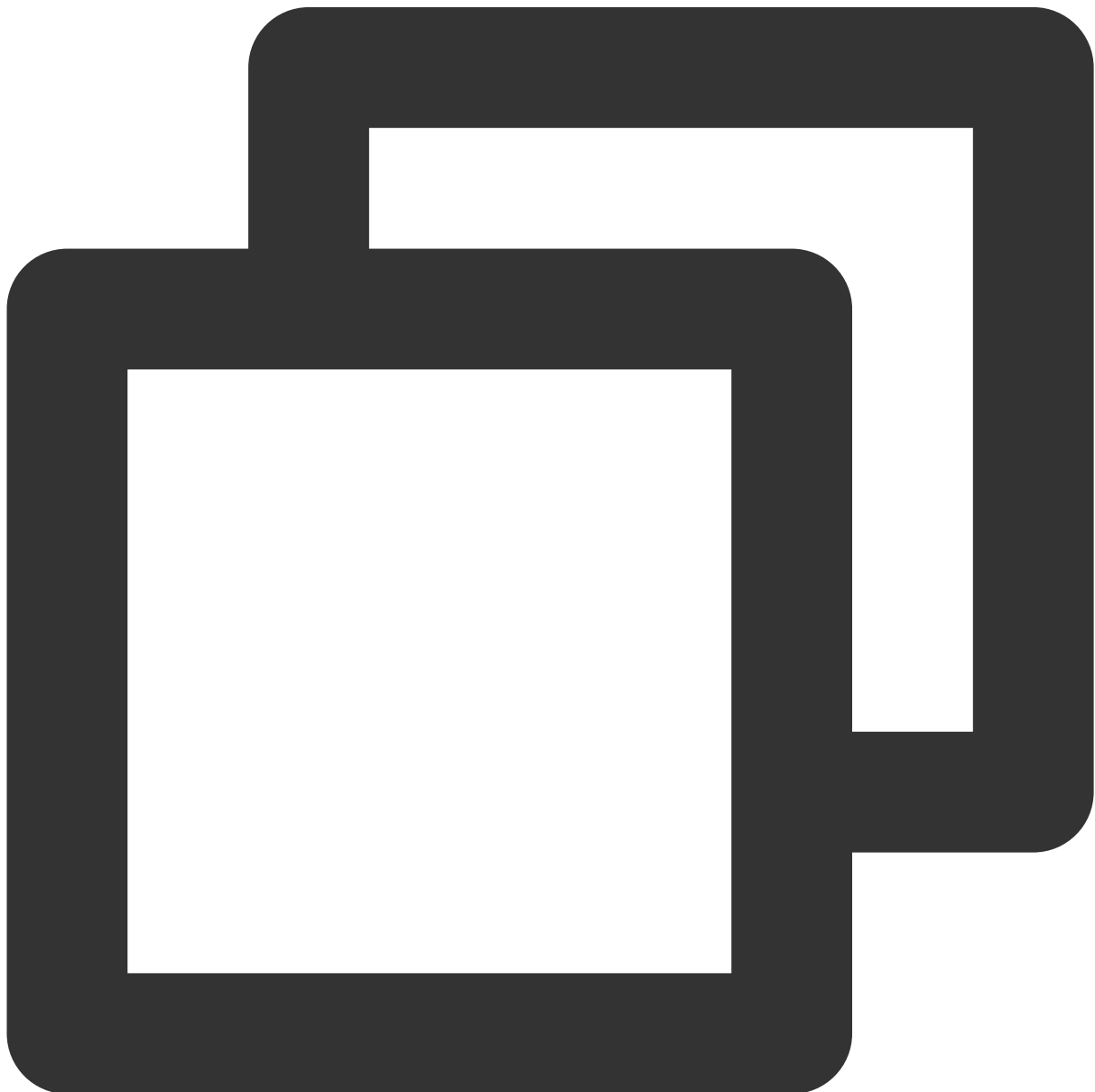
```
<!--`default_on_cluster` specifies the added cluster, which is "" by default.-->
```

```
<default_on_cluster>default_cluster</default_on_cluster>  
<!--`enable_default_on_cluster` specifies whether to enable the DDL On Cluster feat  
<enable_default_on_cluster>1</enable_default_on_cluster>
```

Directions

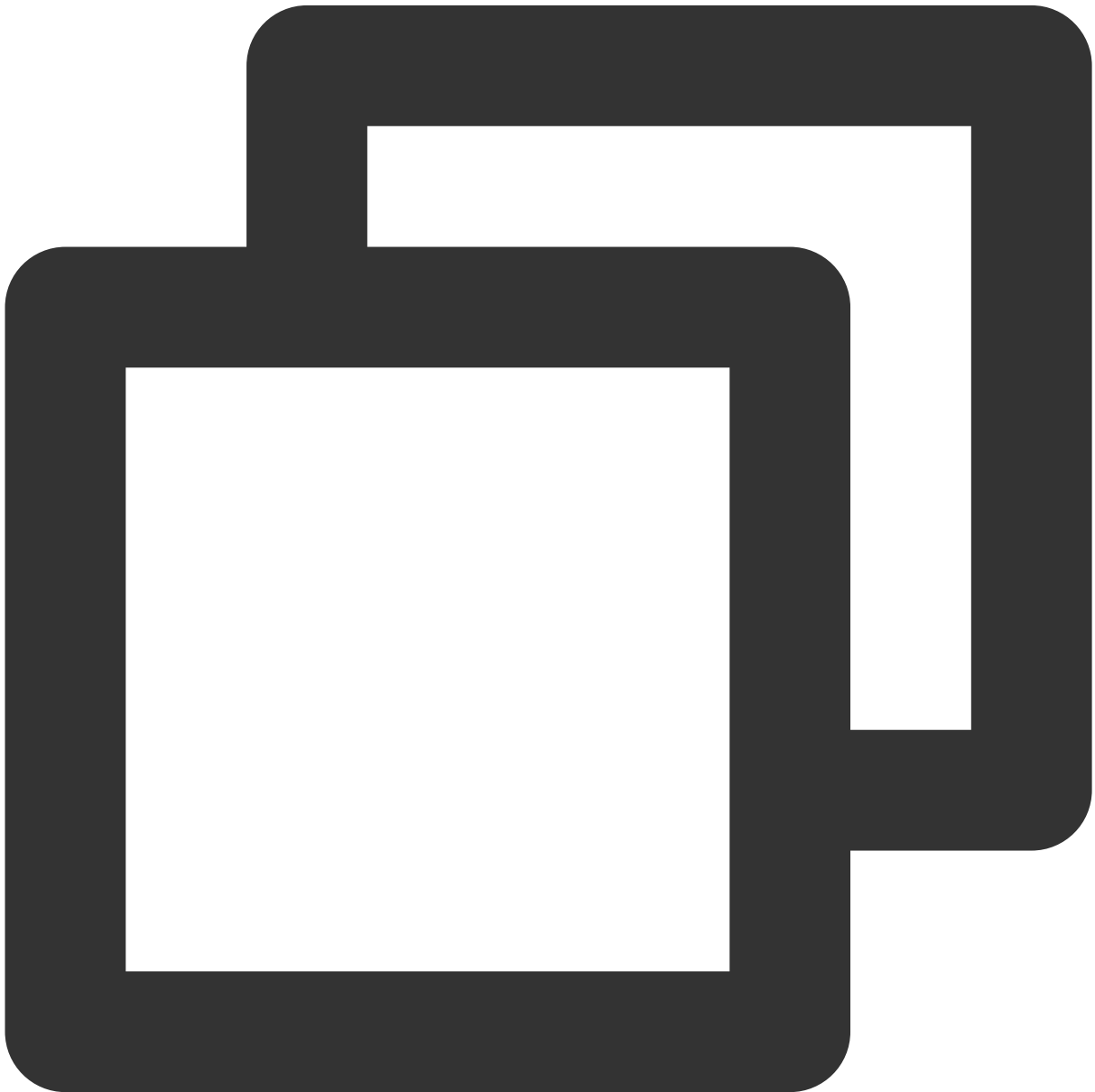
1. Configure in the configuration file permanently

Add the following parameters to `users.xml` , with no restart required after the configuration:



```
<yandex>
  <!-- Profiles of settings. -->
  <profiles>
    <default_on_cluster>default_cluster</default_on_cluster>
    <enable_default_on_cluster>1</enable_default_on_cluster>
  </profiles>
</yandex>
```

2. Modify temporarily at the session level



```
set default_on_cluster='default_cluster';  
set enable_default_on_cluster=1;
```


Data Redistribution

Last updated : 2024-01-19 16:45:30

Feature Overview

This document describes how to redistribute data in the CDWCH console. Currently, CDWCH supports horizontally scaling out a cluster to add nodes for higher computing and storage capabilities. However, ClickHouse doesn't support automatic data rebalancing in cluster datasets and requires human intervention to redistribute the data. How to automate data redistribution has always been a challenge in ClickHouse use and Ops. In contrast, CDWCH offers an data redistribution feature to implement unattended cluster data rebalancing in part mode or resharding mode.

Notes

1. Resharding will reinsert data, and the temporary data in the process will occupy storage space. Before using the resharding mode, ensure that the current remaining capacity exceeds the storage size of the table for the current task.
2. A data table can be associated with only one task, and a cluster supports at most one running task at a time.
3. The local table database must be atomic (default) or ordinary, and the table type must be a table from the MergeTree engine family (including non-replicated and replicated tables; a materialized view table is not supported).
4. The replica relationship of the local table must be consistent with the cluster. There must be a distributed table providing a view to the relationship between shards.
5. Resharding redistribution redistributes data according to the hash rules of distributed tables, and part redistribution balances data at part granularity regardless of the hash rules of distributed tables.
6. Redistribution will generate a temporary table, which is named in the format of "original table name_resharding_temp_task ID_random number". When the status changes to "To switch", verify that the temporary table data is correct and perform switching. After the switching is completed, the original table data will be saved in the table named "original table name_resharding_task ID_random number", and you can delete the old data after verifying it is correct.
7. During redistribution, the original table is set to **read-only** by default. At the table level, you can select some data in partition granularity to redistribute. If no data needs to be written to the redistributed partitions, you can disable read-only for the table; if data needs to be written, disabling read-only will cause data inconsistency. Currently, you cannot select all partitions to make them writable.
8. Redistributed data will be saved in a temporary table, which will replace the original table during the table exchange. Incorrect data may be read in this process. Generally, the exchange takes several seconds.
9. A task may be paused due to cluster problems. If a task is paused, you can repair or configure the cluster based on the reported error, and resume the task.

10. The strings of table creation statements for the redistributed local table and distributed table must be identical.
11. In resharding mode, all the cluster shards must have the same `internal_replication` configuration (either `true` or `false`).
12. Clusters using cloud disks for data tiering must be restarted before redistribution is performed.

Directions

1. Log in to the [CDWCH console](#), select the target cluster in the **Cluster List**, and click the **Data Redistribution** tab on the cluster details page.
2. You can view the list of existing data redistribution tasks on the current page.

Important Notes:

1. Resharding will reinsert data, and the temporary data in the process will occupy storage space. Before using the resharding mode, ensure that the current remaining capacity is sufficient.
2. A data table can be associated with only one task, and a cluster supports at most one running task at a time.
3. The local table database must be atomic (default) or ordinary, and the table type must be a table from the MergeTree engine family (including non-replicated and replicated tables).

Show ▶

New Task

Task ID/Name	V_cluster Name	Redistribute By	Status/Progress	Creation Time	Start Time
	default_cluster	Part	Initialize Completed data tables: 0/1	2022-08-09 17:20:47	-
	default_cluster	Part	Initialize Completed data tables: 0/2	2022-08-09 17:20:32	-

Total items: 2

3. Click **Create Task**, enter the task name, select the `v_cluster` of the table, migration mode (part or resharding), and distributed table for redistribution, and click **OK**.

Part mode: Parts are migrated between cluster shards to rebalance data. This mode doesn't need to follow the sharding specifications of the distributed table, and is more suitable for `rand()` distributed tables.

Resharding mode: The original table data is rewritten to the entire cluster according to the distributed table specifications to rebalance data between shards. This mode is preferred if data needs to be allocated to the same shard based on the primary key.

At the table level, you can select some data in partition granularity to redistribute. You can also set whether to make

the table writable in partition granularity. If no data needs to be written to the redistribution partition, you can disable the read-only status; otherwise, disabling the status will cause data inconsistency.

New Task

Task Name *

6-36 characters; supports Chinese characters, letters, digits, -, and _ .

v_cluster * Redistribute By * Part Resharding

Select Data Table (Note: a data table can be associated with only one task) **3 tables selected**

Distributed Table	Local Table
<input checked="" type="checkbox"/> ssb.dis_lineorder 🔗	ssb.lineorder
<input checked="" type="checkbox"/> ssb.dis_lineorder1 🔗	ssb.lineorder1
<input checked="" type="checkbox"/> ssb.dis_lineorder2 🔗	ssb.lineorder2

Distributed Table

ssb.dis_lineorder [🔗](#)

ssb.dis_lineorder1 [🔗](#)

ssb.dis_lineorder2 [🔗](#)

Hold down Shift to select multiple items

The data redistribution feature uses components with multiple concurrency mechanisms to improve the task performance. To strike a balance between the business load and the impact of data redistribution on the cluster, you can use the **task concurrency** and **node concurrency** to adjust the performance and speed of the task.


Note:

Node concurrency indicates the partition concurrency on a single ClickHouse node in resharding mode or the part concurrency on a single ClickHouse node in part mode.


Task concurrency indicates the total concurrency of all the nodes involved in the task during the task execution.


Setting either of the parameters to 0 will pause the task. In addition, the effect of the same value may differ by cluster specification or task. You can adjust the values to control the task speed, performance, and cluster load as needed.

4. After creating a data redistribution task, you can start, edit, or delete it. The edit window is similar to the creation window and allows you to modify the task information.

Task ID/Name	V_cluster Name	Redistribute By	Status/Progress	Creation Time	Start Time
	default_cluster	Part	Initialize Completed data tables: 0/1	2022-08-09 17:20:47	-

5. Click **Start** to start the task, and the CDWCH instance will enter the **Changing** status (which will also be displayed on the cluster's basic information page, without a progress bar though).

 Unable to create, edit, or start a task because the cluster is not running.

Task ID/Name	V_cluster Name	Redistribute By	Status/Progress	Creation Time	Start Time
	default_cluster	Part	Initialize Completed data tables: 0/1	2022-08-09 17:20:47	-

After the task is started, you can click **Modify Parameter** to modify the task concurrency and node concurrency to control the task speed and cluster load and even pause the task.

6. Click **Details** to view the execution details of the data redistribution task at the table level.

Task Details

Task ID	██████████	Task Name	testtest	V_
Redistribute By	Part	Status	Executing	Pr
Creation Time	2022-08-09 17:20:32	Start Time	2022-08-09 17:22:56	Up

[Switch All](#) [Cancel All Tables](#)

<input type="checkbox"/>	Data Table	Status/Progress
<input type="checkbox"/>	ssb.dis_lineorder	To switch <div style="width: 100%; height: 10px; background-color: green;"></div> 100.00% Executed <u>901.14 MB</u> <u>901.14 MB</u> in total Average S MB/s
<input type="checkbox"/>	ssb.dis_lineorder1	To switch <div style="width: 100%; height: 10px; background-color: green;"></div> 100.00% Executed <u>901.14 MB</u> <u>901.14 MB</u> in total Average S MB/s

Total items: 2 10 ▼ / page

[Back](#)

7. The table data redistribution has three phases:

<input type="checkbox"/> Data Table	Status/Progress
<input type="checkbox"/> ssb.dis_lineorder	<p>Canceled</p> <p>100.00% Executed 901.14 MB 901.14 MB in total Average MB/s</p>
<input type="checkbox"/> ssb.dis_lineorder1	<p>Original table pending deletion</p> <p>100.00% Executed 901.14 MB 901.14 MB in total Average MB/s</p>

Executing: Data in the table is being redistributed, which may involve creating and deleting temporary tables, moving and deleting parts, and reading/writing data based on the selected mode. At this point, the read/write performance of the cluster may drop.

To switch: Data in the distributed table has been redistributed, but the original table is not replaced yet. You need to verify and make sure that data in the temporary table generated during data redistribution is correct and consistent with the original data. Then, you can perform the switch to make your business access the target table.

Note:

Before switching to the data source after redistribution, ensure you have completed data consistency and accuracy checks. The data table after switching will be used as the only data source for system reads and writes.

For redistribution by partition, verify data consistency and accuracy before and after the redistribution at the partition level.

Original table pending deletion: After the switch, the data file of the original table will not be directly deleted. You need to verify the accuracy of the rebalanced data before the deletion. After the deletion, the table will enter the **Executed successfully** status.

Note:

This operation will permanently delete the physical files of the data table before data redistribution. Ensure you have completed data consistency and accuracy checks, and you have finished switching data sources.

For redistribution by partition, only the data file of the corresponding partition of the original table will be deleted.

8. In addition, before **switching** the redistributed table, you can **cancel** the table or the entire task to stop the data redistribution operation and roll back the data. A **switched** or **deleted** table cannot be rolled back.

9. During task execution, if the network jitters, the ClickHouse component is restarted, or other accidents occur in the cluster, the task may enter the **Paused** status, with the cause displayed. You can adjust the cluster accordingly or [submit a ticket](#) for assistance.

10. When all the redistributed tables enter the final status (**Canceled** or **Executed successfully**), the redistribution task status will become **Execution ended**, and the CDWCH instance status will become **Running**.

Task Details

Task ID	rb-fwzvcjbf0e	Task Name	testtest	V_
Redistribute By	Part	Status	Execution ended	Pr
Creation Time	2022-08-09 17:20:32	Start Time	2022-08-09 17:22:56	Up

<input type="checkbox"/> Data Table	Status/Progress
<input type="checkbox"/> ssb.dis_lineorder	<p>Canceled</p> <p>100.00% Executed <u>901.14 MB</u> <u>901.14 MB</u> in total Average : MB/s</p>
<input type="checkbox"/> ssb.dis_lineorder1	<p>Executed successfully</p> <p>100.00% Executed <u>901.14 MB</u> <u>901.14 MB</u> in total Average : MB/s</p>

Total items: 2 10 ▼ / page

[Back](#)

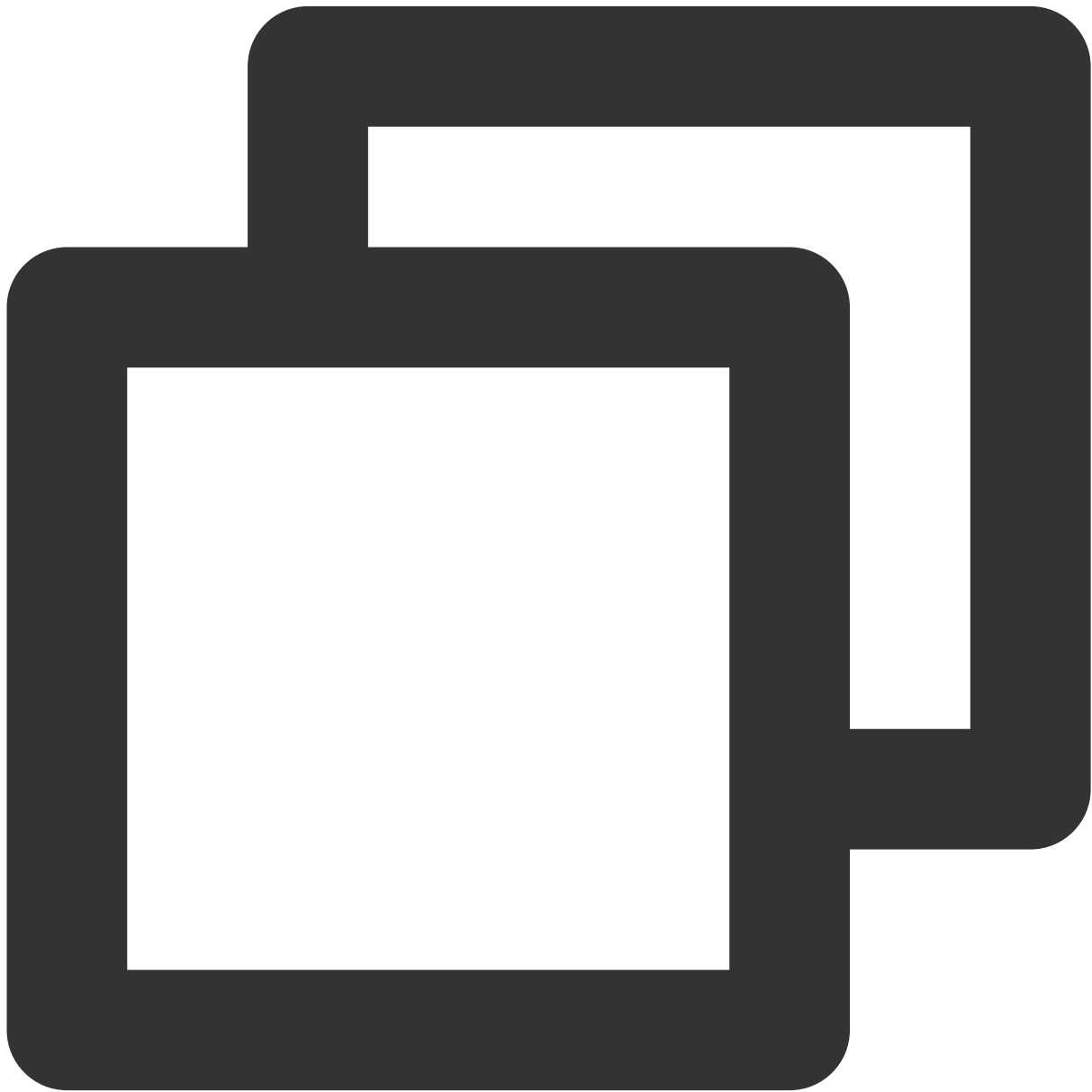
Best Practices for Data Redistribution

Scenarios

1. The data is imbalanced and the distributed table hash is incorrect after scale-out.
2. Data imbalance is caused by business data insertion.
3. You want to adjust the table data distribution policy.

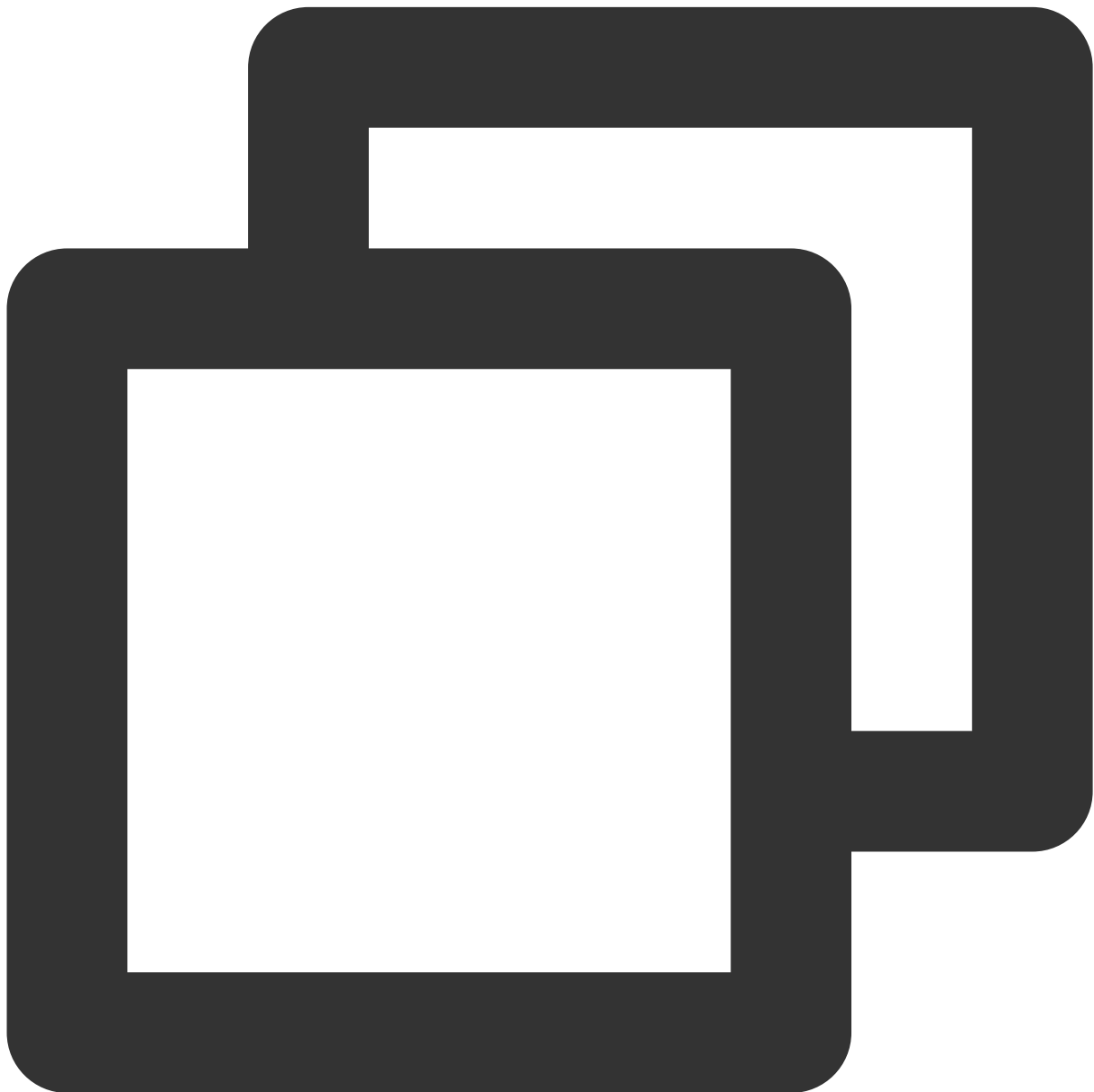
Parameter adjustment

Before a data redistribution task starts, we recommend you add the following parameters to grant permissions; otherwise, the task may pause as the following permissions are required to delete the generated temporary table.




```
<max_table_size_to_drop>0</max_table_size_to_drop>  
<max_partition_size_to_drop>0</max_partition_size_to_drop>
```

For instances with more than 32 CPU cores, we recommend you increase the following parameter values in resharding mode during data redistribution. As data reimport will result in a large number of write requests, doing so will enhance the merge capabilities of the backend to prevent a lot of parts from emerging and thereby causing the too-many-parts error.



```
<background_pool_size>64</background_pool_size>  
<background_schedule_pool_size>64</background_schedule_pool_size>
```

Suggestions

1. We recommend you perform redistribution during off-peak hours, as data will be rewritten in resharding mode and impose a higher load stress on the cluster.
2. Before using the part mode, we recommend you manually run the `optimize` statement to trigger a merge. This increases the part size, reduces the transferred data volume, and thus enhances the redistribution performance.

Performance Data

Test environment 1

Kernel version: 21.8.12.29

High availability: High availability

Compute node type: High I/O

Compute node specification: 64-core 256 GB MEM

Compute node quantity: 4 (two shards), 8 (four shards), and 16 (eight shards)

Compute node storage: 14,280 GB local disk

ZooKeeper node specification: 16-core 64 GB MEM

ZooKeeper node quantity: 3

ZooKeeper node storage: 100 GB enhanced SSD

Data generation tool: ssb-dbgen

Test scenario 1

Expanding two shards to four shards.

Resharding mode

No.	Number of Tables	Total Data Volume	Total Duration	Average Speed
1	1	335 GB	2 minutes 58 seconds	1.878 GB/s
2	1	898 GB	9 minutes 18 seconds	1.609 GB/s
3	2	670 GB	5 minutes 23 seconds	2.074 GB/s
4	2	1,796 GB	13 minutes 25 seconds	2.23 GB/s
5	4	1,340 GB	11 minutes 17 seconds	1.979 GB/s
6	4	3,592 GB	24 minutes 33 seconds	2.439 GB/s

Part mode

--	--	--	--

No.	Number of Tables	Total Data Volume (Migrated Part Volume)	Total Duration	Average Speed
1	1	335 GB (180 GB)	1 minute 44 seconds	1.737 GB/s
2	1	898 GB (476 GB)	6 minutes 27 seconds	1.229 GB/s
3	2	792 GB (397 GB)	4 minutes 50 seconds	1.370 GB/s
4	2	1,796 GB (952 GB)	11 minutes 38 seconds	1.364 GB/s
5	4	1,675 GB (903 GB)	10 minutes 28 seconds	1.438 GB/s
6	4	3,592 GB (2,379 GB)	28 minutes 6 seconds	1.411 GB/s

Test scenario 2

Expanding four shards to eight shards.

Resharding mode

No.	Number of Tables	Total Data Volume	Total Duration	Average Speed
1	1	1,034 GB	3 minutes 39 seconds	4.715 GB/s
2	2	2,068 GB	6 minutes 58 seconds	4.95 GB/s
3	4	4,136 GB	13 minutes 1 second	5.301 GB/s

Part mode

No.	Number of Tables	Total Data Volume (Migrated Part Volume)	Total Duration	Average Speed
1	1	1,034 GB (524 GB)	3 minutes 19 seconds	2.633 GB/s
2	2	2,068 GB (1,049 GB)	6 minutes 35 seconds	2.656 GB/s
3	4	4,136 GB (2,097 GB)	12 minutes 3 seconds	2.900 GB/s

Test scenario 3

Expanding four shards to four shards.

Resharding mode

No.	Number of Tables	Total Data Volume	Total Duration	Average Speed
1	1	1,034 GB	10 minutes 16 seconds	1.679 GB/s
2	2	2,068 GB	20 minutes 34 seconds	1.676 GB/s
3	4	4,136 GB	37 minutes 12 seconds	1.853 GB/s

Test environment 2

Kernel version: 21.8.12.29

High availability: High availability

Compute node type: Big Data

Compute node specification: 64-core 256 GB MEM

Compute node quantity: 16

Compute node storage: 2,142,720 GB local disk

ZooKeeper node specification: 16-core 64 GB MEM

ZooKeeper node quantity: 3

ZooKeeper node storage: 500 GB enhanced SSD

Test scenario 1

Expanding two shards to four shards.

Resharding mode

No.	Number of Tables	Total Data Volume	Total Duration	Average Speed
1	1	773 GB	7 minutes 21 seconds	1.752 GB/s
2	2	1,585 GB	15 minutes 43 seconds	1.681 GB/s
3	4	3,209 GB	32 minutes 9 seconds	1.664 GB/s

Part mode

No.	Number of Tables	Total Data Volume (Migrated Part Volume)	Total Duration	Average Speed
1	1	774 GB (393 GB)	4 minutes 47 seconds	1.371 GB/s
2	2	1,548 GB (786 GB)	9 minutes 53	1.563 GB/s

			seconds	
3	4	3,096 GB (1,572 GB)	19 minutes 18 seconds	1.358 GB/s

Test scenario 2

Expanding four shards to eight shards.

Resharding mode

No.	Number of Tables	Total Data Volume	Total Duration	Average Speed
1	1	855 GB	4 minutes 37 seconds	3.087 GB/s
2	2	1,667 GB	9 minutes 6 seconds	3.053 GB/s
3	4	3,292 GB	18 minutes 49 seconds	2.916 GB/s

Part mode

No.	Number of Tables	Total Data Volume (Migrated Part Volume)	Total Duration	Average Speed
1	1	1,072 GB (542 GB)	3 minutes 21 seconds	2.697 GB/s
2	2	2,144 GB (1,084 GB)	6 minutes 24 seconds	2.823 GB/s
3	4	4,288 GB (2,167 GB)	12 minutes 34 seconds	2.874 GB/s

Scale-in and Migration

Last updated : 2024-01-19 16:45:30

Overview

This document describes how to scale in a ClickHouse cluster and migrate data in the console. Currently, CDW supports horizontally scaling in a cluster to reduce nodes. Data on the removed nodes should be migrated to remaining nodes to avoid business data loss. Specifically, data can be migrated by part and then distributed evenly on the remaining nodes.

Notes

1. Node data migration is performed by part. When data is migrated from the source shard to the target shard, table data in the target shard will be rebalanced by part.
2. A data table can be associated with only one task, and a cluster supports at most one running task at a time.
3. The local table database must be atomic (default) or ordinary, and the table type must be a table from the MergeTree engine family (including non-replicated and replicated tables; a materialized view table is not supported).
4. The replica relationship of the local table must be consistent with the cluster. There must be a distributed table providing a view to the relationship between shards.
5. Migration will generate a temporary table, which is named in the format of "original table name_resharding_temp_task ID_random number". When the status changes to **To switch**, verify that the temporary table data is correct and perform switching. After the switching is completed, the original table data will be saved in the table named "original table name_resharding_task ID_random number", and you can delete the old data after verifying it is correct.
6. During migration, the original table is set to **read-only** by default. You can select some data in partition granularity to migrate. If no data needs to be written to the migrated partitions, you can disable read-only for the table; if data needs to be written, disabling read-only will cause data inconsistency. Currently, you cannot select all partitions to make them writable.
7. The migration task can be configured with the task concurrency and node concurrency to control the performance. If either of the values is 0, the task will be paused.
8. Migrated data will be saved in a temporary table, which will replace the original table during the table switch. Incorrect data may be read in this process. Generally, the switch takes several seconds.
9. A task may be paused due to cluster problems. If a task is paused, you can repair or configure the cluster based on the reported error, and resume the task.
10. The strings of table creation statements for the migrated local table and distributed table must be identical.

11. Clusters using cloud disks for data tiering must be restarted before scale-in and migration is performed.

Directions

1. Log in to the [CDW console](#), select the target cluster in the **Cluster List**, and click the **Scale-in and Migration** tab on the cluster details page.
2. You can view the list of existing scale-in and migration tasks on the current page.
3. Click **Create Task**, enter the task name, select the `v_cluster` of the table, source shard, target shard, and distributed table for migration, and click **OK**.

We recommend you select the source and target shards in line with the scale-in requirements and use the shard to be removed as the source shard.

The **part mode** is used by default to migrate data after scale-in. The parts of the table on the source shard are used to rebalance data between shards. This mode doesn't need to follow the sharding specifications of the distributed table, and is more suitable for `rand()` distributed tables.

At the table level, you can select some data in partition granularity to migrate. If no data needs to be written to the migrated partitions, you can disable read-only for the table; if data needs to be written, disabling read-only will cause data inconsistency.

The scale-in and migration feature uses components with multiple concurrency mechanisms to improve the task performance. To strike a balance between the business load and the impact of scale-in and migration on the cluster, you can use the **task concurrency** and **node concurrency** to adjust the performance and speed of the task.

Note:

Node concurrency indicates the part concurrency on a single ClickHouse node.

Task concurrency indicates the total concurrency of all the nodes involved in the task during the task execution.

Setting either of the parameters to 0 will pause the task. In addition, the effect of the same value may differ by cluster specification or task. You can adjust the values to control the task speed, performance, and cluster load as needed.

4. After creating a scale-in and migration task, you can start, edit, or delete it. The edit window is similar to the creation window and allows you to modify the task information.

5. Click **Start** to start the task, and the CDW instance will enter the **Changing** status (which will also be displayed on the cluster's basic information page, without a progress bar though).

After the task is started, you can click **Modify Parameter** to modify the task concurrency and node concurrency to control the task speed and cluster load and even pause the task.

6. Click **Details** to view the execution details of the scale-in and migration task at the table level.

7. The table data migration has three phases:

Executing: Data in the table is being migrated, which may involve creating and deleting temporary tables, moving and deleting parts, and reading/writing data. At this point, the read/write performance of the cluster may drop.

To switch: Data in the distributed table has been migrated, but the original table is not replaced yet. You need to verify and make sure that data in the temporary table generated in the target shard is correct and consistent with the

original data. Then, you can perform the switch to make your business access the target table.

Note:

Before switching the data source, be sure to verify the data consistency and accuracy before and after the migration. The data table after switching will be used as the only data source for system reads and writes.

For scale-in and migration by partition, verify data consistency and accuracy before and after the migration at the partition level.

Original table pending deletion: After the switch, the data file of the original table will not be directly deleted. You need to verify the accuracy of the data rebalanced in the target shard before the deletion. After the deletion, the table will enter the **Executed successfully** status.

Note:

This operation will permanently delete the physical files of the data table before data migration. Ensure you have completed data consistency and accuracy checks, and you have finished switching data sources.

For scale-in and migration by partition, only the data file of the corresponding partition of the original table will be deleted.

6. In addition, before **switching** the migrated table, you can **cancel** the table or the entire task to stop the data migration operation and roll back the data. A **switched** or **deleted** table cannot be rolled back.

7. During task execution, if the network jitters, the ClickHouse component is restarted, or other accidents occur in the cluster, the task may enter the **Paused** status, with the cause displayed. You can adjust the cluster accordingly or submit a ticket for assistance.

8. When all the migrated tables enter the final status (**Canceled** or **Executed successfully**), the migration task status will become **Execution ended**, and the CDW instance status will become **Running**.

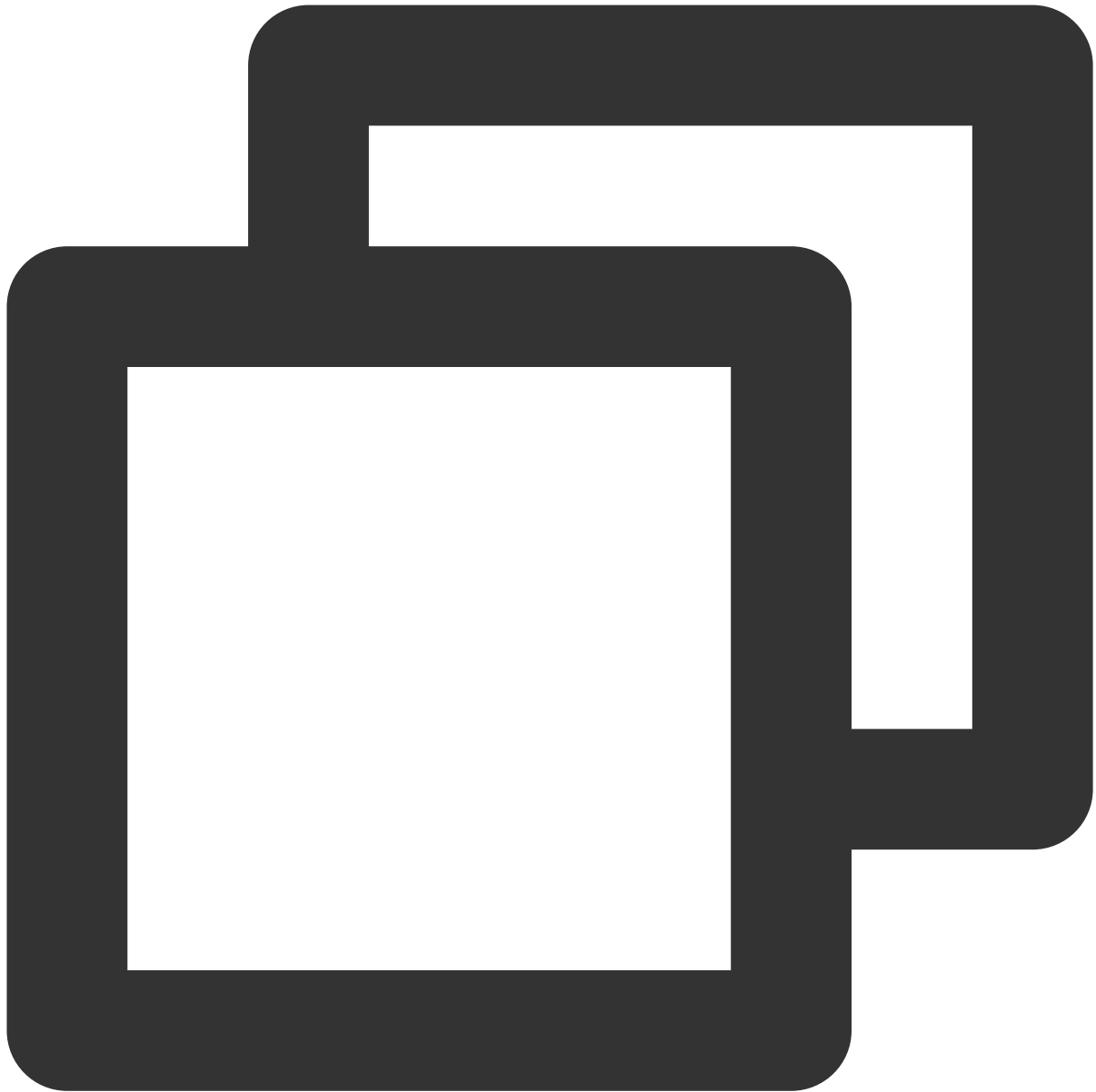
Best Practices

Scenarios

1. You want to retain the data on a node to be removed and migrate it to a remaining node before scale-in.
2. You want to redistribute the data.

Parameter adjustment

Before a scale-in and migration task starts, we recommend you add the following parameters to grant permissions; otherwise, the task may pause as the following permissions are required to delete the generated temporary table.



```
<max_table_size_to_drop>0</max_table_size_to_drop>  
<max_partition_size_to_drop>0</max_partition_size_to_drop>
```

Suggestions

Before using the part mode, we recommend you manually run the `optimize` statement to trigger a merge. This increases the part size while decreasing the transferred data volume, thus improving scale-in and migration performance.