TencentDB for DBbrain

Operation Guide

Product Documentation

Copyright Notice

©2013-2025 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

Tencent Cloud

All trademarks associated with Tencent Cloud and its services are owned by the Tencent corporate group, including its parent, subsidiaries and affiliated companies, as the case may be. 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
  - Cloud Access Management
    - Overview
    - Authorization Policy Syntax
    - Authorizable Resource Types
  - Instance Overview
  - Instance Management
  - Monitoring and Alarms
    - Database Inspection
    - Exception Alarms
    - Event Notification
    - Intelligent Monitoring (Monitoring Dashboard)
    - Intelligent Monitoring (Full Instance Monitoring)
  - Health Report
    - Health Report Management
    - Health Report Email Push
    - Contact Management
  - MySQL/TDSQL-C for MySQL Performance Optimization
    - Exception Diagnosis
    - Performance Trends
    - Real-Time Session
    - Slow SQL Analysis
    - Space Analysis
    - SQL Optimization
    - Deadlock Visualization
    - Event Notification
    - Best Practices
  - Redis Performance Optimization
    - Exception Diagnosis
    - Performance Trends
    - Real-Time Session
    - Slow Log Analysis
    - Memory Analysis (Big Key Analysis)
    - Latency Analysis
    - Latency Analysis (Command Word Analysis)

©2013-2025 Tencent Cloud International Pte. Ltd.

Latency Analysis (Hot Key Analysis)
MongoDB Performance Optimization
Exception Diagnosis
Performance Trends
Slow SQL Analysis
Space Analysis
MongoStatus
MongoTop
Real-Time Session
SQL Throttling
Index Recommendation
Best Practices
Full-Link Analysis
Enabling Full-Link Analysis
Detail Query
SQL Analysis

Operation Guide
Cloud Access Management
Overview

Last updated：2022-09-01 18:34:46

Issues

If you have multiple users managing different Tencent Cloud services such as CVM, VPC, and TencentDB, and they all share your Tencent Cloud account access key, you may face the following problems:
The risk of your key being compromised is high since multiple users are sharing it.
Your users might introduce security risks from maloperations due to the lack of user access control.

Solution

You can avoid the above problems by allowing different users to manage different services through sub-accounts. By default, sub-accounts don't have permissions to use Tencent Cloud services or resources. Therefore, you need to create policies to grant them different permissions.
Cloud Access Management (CAM) is a web-based Tencent Cloud service that helps you securely manage and control access permissions of your Tencent Cloud resources. Using CAM, you can create, manage, and terminate users (groups), and control the Tencent Cloud resources that can be used by the specified user through identity and policy management.
When using CAM, you can associate a policy with a user or user group to allow or forbid them to use specified resources to complete specified tasks. For more information on CAM policies, see Element Reference.
If you do not need to manage the access permissions to DBbrain resources for sub-accounts, you can skip this chapter. Skipping this chapter will not affect your understanding and usage of other parts in the documentation.

Getting started

A CAM policy must authorize or deny the use of one or more DBbrain operations. At the same time, it must specify the resources that can be used for the operations (which can be all resources or partial resources for certain operations). A policy can also include the conditions set for the manipulated resources.
Note:
We recommend you manage DBbrain resources and authorize DBbrain operations through CAM policies. Although the user experience does not change for existing users who are granted permissions by project, we do not recommend you continue to manage resources and authorize operations in a project-based manner.

©2013-2025 Tencent Cloud International Pte. Ltd.

Currently, DBbrain does not support setting conditions for policies.

Task	Document
Quickly authorize a sub-user	Authorizing a sub-user
Learn more about the basic policy structure	Policy syntax
Define operations in a policy	DBbrain operations
Define resources in a policy	Resources that can be manipulated by DBbrain
View supported resource-level permissions	Authorizable Resource Types

©2013-2025 Tencent Cloud International Pte. Ltd.

Authorization Policy Syntax

Last updated：2022-09-01 18:34:46

Authorizing a Sub-User

1. Log in to the CAM Console with the root account, select the target sub-user in the user list, and click **Authorize**.
2. In the pop-up dialog box, select a preset policy and click **OK** to complete the authorization.

`QcloudDBBRAINFullAccess` (DBbrain full real and write access permission): an associated user can use all features provided by DBbrain, including viewing and creating tasks such as SQL insight task, health report, and compliance security report.

`QcloudDBBRAINReadOnlyAccess` (DBbrain read-only access permission): an associated user can only view DBbrain pages and cannot create tasks.

Policy Syntax

CAM policy:

```
{
         "version":"2.0",
         "statement":
         [
            {
               "effect":"effect",
               "action":["action"],
               "resource":["resource"],
                "condition": {"key":{"value"}}
            }
        ]
}
```

version is required. Currently, only "2.0" is allowed.

statement describes the details of one or more permissions. It contains a permission or permission set of multiple other elements such as `effect`, `action`, `resource`, and `condition`. One policy has only one `statement`.

effect describes whether the statement result is "allow" or "explicit deny". This element is required.

action describes the allowed or denied operation. An operation can be an API (prefixed with "cdb:"). This element is required.

resource describes the objects the statement covers. A resource is described in a six-segment format. Detailed resource definitions vary by product. This element is required.
condition describes the condition for the policy to take effect. A condition consists of an operator, operation key, and operation value. A condition value may contain information such as time and IP address. Some services allow you to specify additional values in a condition. This element is required.

DBbrain Operations

In a DBbrain policy statement, you can specify any API operation from any service that supports DBbrain. APIs prefixed with `dbbrain:` should be used for DBbrain, such as `dbbrain:DescribeSlowLogTopSqls` or `dbbrain:DescribeSlowLogTimeSeriesStats`.
To specify multiple operations in a single statement, separate them with commas as shown below:

```
"action":["dbbrain:action1","dbbrain:action2"]
```

You can also specify multiple operations by using a wildcard. For example, you can specify all the names of operations beginning with "Describe" as shown below:

```
"action":["dbbrain:Describe*"]
```

If you want to specify all operations in DBbrain, use the "*" wildcard as shown below:

```
"action":["dbbrain:*"]
```

Resources that can be Manipulated by DBbrain

Each CAM policy statement has its own resources. DBbrain allows you to operate on TencentDB resources. TencentDB resources generally have following format:

```
qcs:project_id:service_type:region:account:resource
```

project_id describes the project information and is only used to enable compatibility with legacy CAM logic. It can be left empty.
service_type describes the product's abbreviation, such as `cdb`.
region describes the region information, such as `ap-guangzhou`.
account is the root account of the resource owner, such as `uin/653339763`.
resource describes the detailed resource information of each product, such as `instanceId/instance_id1` or `instanceId/*`.
For example, you can specify a resource for a specific instance (cdb-k05xdcta) in a statement as shown below:

```
"resource":[ "qcs::cdb:ap-guangzhou:uin/653339763:instanceId/cdb-k05xdcta"]
```

You can also use the wildcard "*" to specify a resource for all instances that belong to a specific account as shown below:

```
"resource":[ "qcs::cdb:ap-guangzhou:uin/653339763:instanceId/*"]
```

If you want to specify all resources or if a specific API operation does not support resource-level permission control, you can use the wildcard "*" in the `resource` element as shown below:

```
"resource": ["*"]
```

To specify multiple resources in a single command, separate them with commas. Below is an example where two resources are specified:

```
"resource":["resource1","resource2"]
```

The table below describes the resources that can be used by TencentDB and the corresponding resource description methods, where words prefixed with $ are placeholders, `project` refers to a project ID, `region` refers to a region, and `account` refers to an account ID.

Resource	Resource Description Method in Authorization Policy
Instance	`qcs::cdb:$region:$account:instanceId/$instanceId`

Authorizable Resource Types

Last updated：2021-08-10 15:35:36

Resource-level permission is used to specify which resources a user can manipulate. DBbrain supports certain resource-level permissions. This means that for the TencentDB operations that support resource-level permission, you can control when a user is allowed to perform operations or what resources the user can use. The following table describes the types of resources that can be authorized in CAM.

Resource Type	Resource Description Method in the Authorization Policy
TencentDB instance resources	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`

The table below lists the DBbrain API operations that currently support resource-level permission control as well as the resources and condition keys supported by each operation. When specifying a resource path, you can use the "*" wildcard in the path.

> Any DBbrain API operation not listed in the table does not support resource-level permission. For such an operation, you can still authorize a user to perform it, but you must specify `*` as the resource element in the policy statement.

API Operation	Resource Path
DescribeSlowLogTopSqls	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
DescribeSlowLogTimeSeriesStats	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
DescribeDBDiagHistory	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
DescribeDBDiagEvent	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
CreateAuditLogStatsTask	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
DescribeAuditLogStatsTasks	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`

©2013-2025 Tencent Cloud International Pte. Ltd.

DescribeAuditLogSeriesForSqlTime	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
DescribeAuditLogTopSqls	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
DescribeAuditLogMetricRatio	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
DeleteAuditLogStatsTask	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
DescribeDBSpaceStatus	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
DescribeTopSpaceTables	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
DescribeDBPerfTimeSeries	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
DescribeSqlExplain	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
CreateDiagUserInstances	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
DeleteDiagUserInstances	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
DescribeProcessList	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
CreateDBDiagReportTask	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
DescribeDBDiagReportTasks	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
DescribeDBDiagReport	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
DeleteDBDiagReportTasks	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
DescribeSqlAdvice	`qcs::cdb:$region:$account:instanceId/*`

©2013-2025 Tencent Cloud International Pte. Ltd.

	`qcs::cdb:$region:$account:instanceId/$instanceId`
DescribeHealthScoreTimeSeries	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
DescribeHealthScore	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
CreateDBDiagReportUrl	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`

©2013-2025 Tencent Cloud International Pte. Ltd.

Instance Overview

Last updated：2022-08-13 20:18:46

The instance overview page displays the summary of your instances, which is customizable. You can view information such as task execution, region distribution, real-time performance, and health assessment of all connected instances.

Note:

Currently, instance overview is supported only for TencentDB for MySQL (excluding basic single-node instances), TDSQL-C for MySQL, TencentDB for Redis, self-built MySQL, and TencentDB for MongoDB.

Log in to the DBbrain console, select **Instance Overview** on the left sidebar, and select a database on the right. You can view **Real-Time** and **Historical** data of all regions or a specific region.

Recommended Features

The navigation bar at the top highlights popular features recommended by DBbrain. You can quickly access the details of the corresponding feature.

Self-built instance access

The self-built database instance access page displays the number of self-built database instances that access the DBbrain service through the Agent and direct connection under the current account. You can click **Quick Access** to redirect to the self-built database instance access page.

Custom Settings

DBbrain provides custom settings. Click **Custom Settings** to enter the instance management page, select the instances to be displayed, and configure them. For more information, see Instance Management.

Exception Alarming

DBbrain's 24/7 exception diagnosis module can detect problems in database instances in real time and provide optimization plans accordingly. This module displays the total number of exception alarms in the last 3 hours and in the last 24 hours. You can click to access the exception alarm page and view more details.

Health Rankings

©2013-2025 Tencent Cloud International Pte. Ltd.

DBbrain periodically performs health checks on all instances and scores them accordingly. On this page, you can view the health scores (current and historical) of all instances. You can click an instance to access the exception diagnosis page and view more details.

Monitoring Status Rankings

The resource consumption rankings of selected monitoring metrics are displayed. You can click an instance to view details about its exception diagnosis.
MySQL metrics: CPU, memory, disk utilization, TPS, QPS, number of slow queries, connected threads, and running threads.
TDSQL-C metrics: CPU, memory, storage utilization, TPS, QPS, number of slow queries, connected threads, and running threads.
Redis metrics: CPU utilization, memory utilization, connection utilization, inbound traffic utilization, outbound traffic utilization, and read request hit rate.

©2013-2025 Tencent Cloud International Pte. Ltd.

Instance Management

Last updated：2022-08-22 18:07:22

The instance management feature displays the information of the TencentDB instances supporting DBbrain so that you can conveniently manage database instances.
For TencentDB databases, this feature mainly displays the basic information of database instances (instance name/ID, status, etc.) and their groups, exception alarms, health scores, and operations.
For self-built databases, this feature mainly displays the basic information of database instances (instance name/ID, status, etc.), exception alarms, health scores, monitoring data collection, slow log collection, access mode, agent status, instance status, accounts, and operations.
Note:
Currently, this feature is supported only for TencentDB for MySQL (excluding basic single-node instances), TencentDB for Redis, TDSQL-C for MySQL, self-built MySQL, and TencentDB for MongoDB.

Management List

TencentDB databases

Log in to the DBbrain console and select **Instance Management** on the left sidebar. On the displayed page, select a TencentDB database at the top.
The instance management list shows the basic information of database instances, exception alarms, health scores, and operations. In the search box above the list, you can filter, aggregate, and search data by field.
Status: This column displays whether database inspection or instance overview is enabled for an instance. To modify the status of an instance, click the **Edit** icon in the **Status** column; to modify the status of multiple instances at a time, select the instances in the list and click **Custom Settings** at the top. You can filter data by status.
Health Score: This column displays the instance health score (the higher the score, the healthier the instance) rated during periodic health checks. You can sort data by health score.
Exception Alarms: This column displays the number of exceptions of an instance detected by "24/7 Exception Diagnosis". You can click the number in the column to view more details and sort data by the number.
Group: In this column, click the **Edit** icon to select the default group or create a group for an instance. You can also select one or multiple instances in the list and click **Manage Group** at the top to switch them to another group or add them to a new group.
Note:
Grouping is not supported for TDSQL-C for MySQL currently.
Operation: In this column, you can click **Performance Optimization** to enter the corresponding feature details page and view the instance status.

©2013-2025 Tencent Cloud International Pte. Ltd.

Self-built database

Log in to the DBbrain console and select **Instance Management** on the left sidebar. On the displayed page, select a self-built database at the top.

The instance management list shows the basic information, exception alarms, health scores, monitoring data collection status, slow log collection status, access mode, Agent status, instance status, accounts, and operations of database instances. In the search box above the list, you can filter, aggregate, and search for data by field.

Status: This column displays whether database inspection or instance overview is enabled for an instance. To modify the status of an instance, click the **Edit** icon in the **Status** column; to modify the status of multiple instances at a time, select the instances in the list and click **Custom Settings** at the top. You can filter data by status.

Health Score: This column displays the instance health score (the higher the score, the healthier the instance) rated during periodic health checks. You can sort data by health score.

Exception Alarms: This column displays the number of exceptions of an instance detected by "24/7 Exception Diagnosis". You can click the number in the column to view more details and sort data by the number.

Configuration: This column displays the configuration of the database, including number of CPU cores, memory size, and disk size. The configuration is assigned by the server to the self-built database. DBbrain will configure the computing performance based on the entered values.

Monitoring and Collection: This column displays whether the monitoring feature of DBbrain is enabled to collect the database performance data. The switch is toggled on by default and cannot be toggled off.

Slow Log Collection: This column displays whether slow log collection is enabled. After the switch is toggled on, DBbrain will monitor the database's slow log status. Before this feature is enabled, you need to check whether the slow log collection permission is enabled.

Note:

Self-built database instances that access the service through direct connection do not support slow log collection.

Network Type: This column displays the network type of a connected self-built database instance, including private network and public network.

Access Mode: This column displays the access method of a self-built database instance, including direct access and agent access.

Agent Status: This column displays the real-time status of the Agent for a self-built database instance that accesses the service through the Agent. It helps you detect Agent exceptions promptly.

Instance Status: This column displays the real-time status of a database instance, so you can promptly detect its exceptions.

Account: This column displays the database account that is authorized to access the DBbrain service. You can click **Change Database Account** in the **Operation** column to change the authorized account.

Operation:

Click **Performance Optimization** to enter the corresponding feature details page and view the instance status.

Select **More** > **Cancel Access** to remove the self-built database instance that accesses the DBbrain service.

Select **More** > **Change Database Account** to change the database account authorized to access the DBbrain service.

Select **More** > **Manage Agent** to view the basic information of the agent, including the agent's server IP, port, version, and status.

Note:

If an exception occurs on the Agent, click **Reconnect** in the **Operation** column next to the Agent status to restart the Agent. You can also click **Manual Restart Guide** in the top-right corner of the Agent management page to view how to manually restart the Agent on the server.

Custom Settings

DBbrain provides the custom settings feature. You can customize the settings about which instances to be displayed on the instance overview, database inspection or security governance page based on your needs.

1. Log in to the DBbrain console and select **Instance Management** on the left sidebar. On the displayed page, select a database at the top.
2. In the list, select one or multiple instances and click **Custom Settings**.
3. In the pop-up window, enable or disable database inspection or instance overview. You can click **View Details** to view the basic information of the selected instance.

Monitoring and Alarms
Database Inspection

Last updated：2022-08-13 20:23:22

Database inspection is used to automate full instance health checks regularly. You can also set up custom inspections based on your own needs to help troubleshoot potential instance issues and provide solutions.

Note:

Currently, database inspection is supported only for TencentDB for MySQL (excluding basic single-node instances), TDSQL-C for MySQL, self-built MySQL, TencentDB for Redis, and TencentDB for MongoDB.

Log in to the DBbrain console, select **Monitoring and Alarming** > **Database Inspection** on the left sidebar, and select a database type at the top.

Database inspection list

Note:

Health report email push is not supported for self-built databases currently.

The database inspection list displays a summary of inspection information generated by the database instance, such as basic instance information, health level, the number of slow queries, and the number of big tables.

You can select last 1 day, last 3 days, last 7 days or any time period to view the full instance inspection information.

You can also perform fuzzy search by instance ID, health level, etc.

The "Health Level" column displays the health level obtained through regular health inspections, including healthy, sub-healthy, dangerous, and high-risk.

Click **Export** on the top-right corner to export the report information of full instance inspection.

Click **Email Settings** on the top-right corner to configure the email information for receiving the health report generated by the database inspection. For more information, see Health Report Email Push.

Click **View** in the **Operation** column to view or download the health report of the instance.

Click **Email** in the **Operation** column or click **Batch Send** after selecting multiple database inspection records to email the health reports to the specified recipient. For more information, see Health Report Email Push.

Click **Deduction Details** in the **Operation** column to view the reason for deduction of health level, including name, category, max severity, occurrences, and deduction details. For detailed description of diagnosis items, see Exception Alarms.

Custom settings

DBbrain provides custom settings. Click **Custom Settings** to enter the instance management page and set the instance to be displayed. For more information, see Instance Management.

Quickly enabling/disabling inspection for all instances

DBbrain supports enabling/disabling inspection for all instances with just one click. The inspection for all instances is disabled by default. You can toggle on **Full instance inspection disabled** to enable inspection for all instances.

©2013-2025 Tencent Cloud International Pte. Ltd.

Exception Alarms

Last updated：2024-07-31 11:14:57

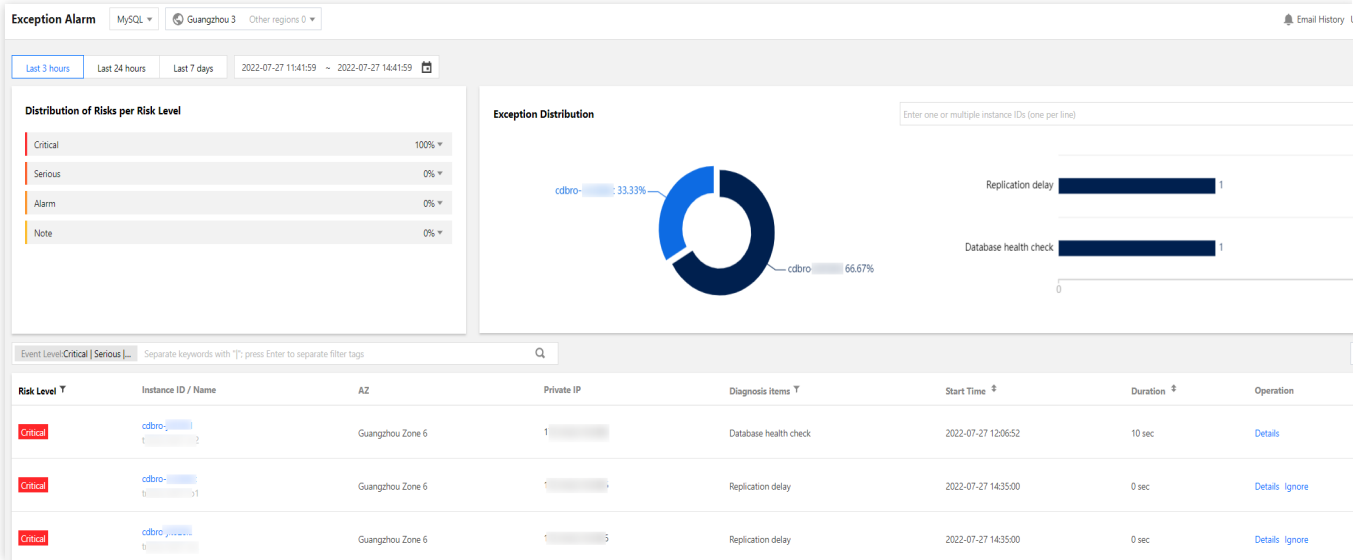
The exception alarm page displays the information overview of exception alarms (exceptions detected by "24/7 Exception Diagnosis") generated by database instances connected to DBbrain under your account.

Note:

Currently, exception alarm is supported only for TencentDB for MySQL (excluding basic single-node instances), TDSQL-C for MySQL, TencentDB for MariaDB, TDSQL for MySQL, TencentDB for Redis, TencentDB for MongoDB, and self-built MySQL.

Viewing an Exception Alarm

1. Log in to the DBbrain console.
2. In the left sidebar, choose **Monitoring & Alarm** > **Exception Alarm**.
3. On the top of the page, select the database type and region.
4. Select the time range for viewing alarms. Supported options are the last 3 hours, last 24 hours, last 7 days, and a custom time range.
5. View exception alarms.



View distribution of risks per risk level:

Displays the proportion of alarms at each risk level (including note, alarm, serious and critical). Click a specific proportion number to view the involved diagnosis items and the number of alarms. Click a specific diagnosis item's row in the alarm list to show the list of that diagnosis item.

View exception distribution:

In the pie chart, you can view the proportion of exception alarms for each instance. Click the instance name to view the

diagnosis items and the number of alarms related to each instance. It also supports filtering by instance ID to view the alarm proportion and involved diagnosis items and alarm quantity for one or more instances.

Click the right-side diagnosis items bar chart, and the alarm list will show the list of that diagnosis item.

View the exception alarms list:

You can filter alarms by instance name, instance ID, private IP address, and diagnosis items.

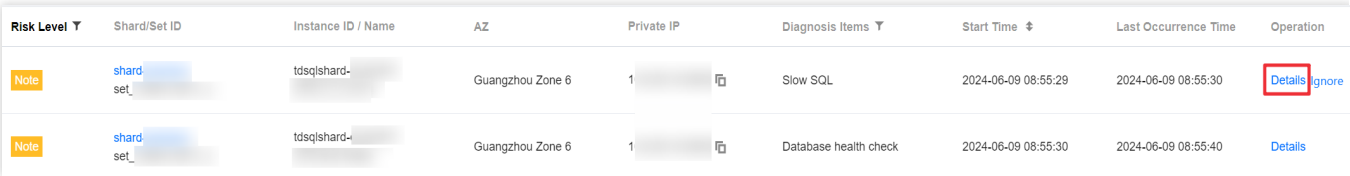
The list displays fields such as risk level, instance ID/name, diagnosis items, start time, last occurrence time, and operation. For different selected database types, the list shows different fields. Refer to the actual display.

Both risk level and diagnosis items in the list support filtering. The action bar supports viewing alarm details, ignoring, and unignoring alarms.

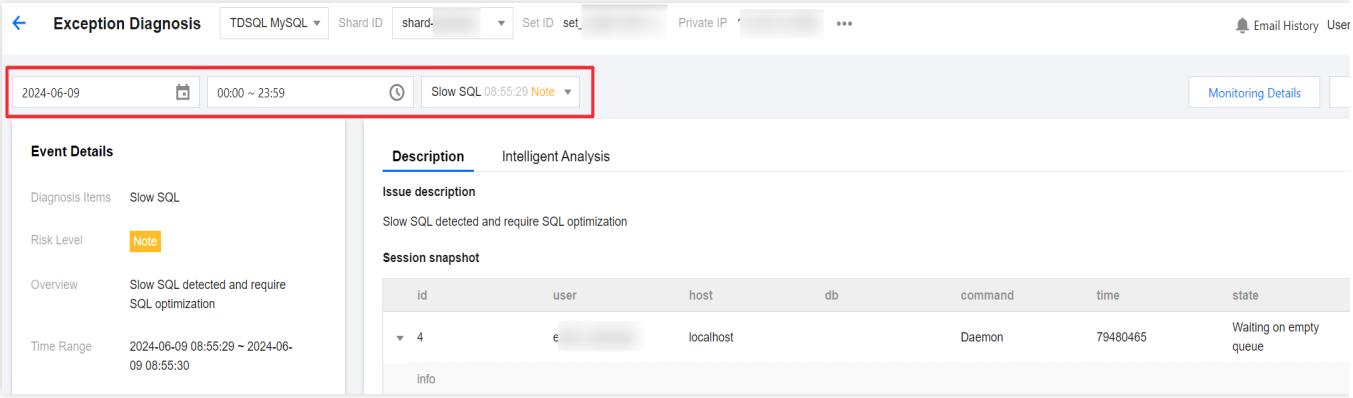
On the top right corner of the page, click **collapse chart** to collapse the risk level distribution and exception alarm distribution, showing only the exception alarms list.

Viewing Exception Alarm Details

1. Log in to the DBbrain console.
2. In the left sidebar, choose **Monitoring & Alarm** > **Exception Alarm**.
3. On the top of the page, select the database type and region.
4. Select the time range for viewing alarms. Supported options are the last 3 hours, last 24 hours, last 7 days, and a custom time range.
5. In the exception alarm list's **Operation** bar, click **Details** to enter the alarm details page, where you can view the alarm details corresponding to the instance.



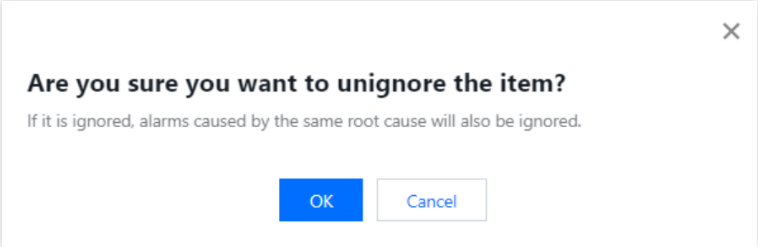
You can select the time range and diagnosis item type to view the alarm details. The content displayed in the alarm details varies according to the diagnosis item.



Ignoring/Unignoring an Alarm

You can ignore or unignore exception alarms that are not generated by **health inspections** to better filter exception alarms.

1. Log in to the DBbrain console.
2. In the left sidebar, choose **Monitoring & Alarm** > **Exception Alarm**.
3. On the top of the page, select the database type and region.
4. Select the time range for viewing alarms. Supported options are the last 3 hours, last 24 hours, last 7 days, and a custom time range.
5. In the exception alarm list's **Operation** bar, click **Ignore** to select **Ignore this item** or **Ignore this type**, and then click **OK**. You can also ignore alarms on the alarm details page.



Ignore this alarm: Only ignores this specific alarm.

Ignore this type: Once it is configured, exception alarms caused by the same root cause will also be ignored.

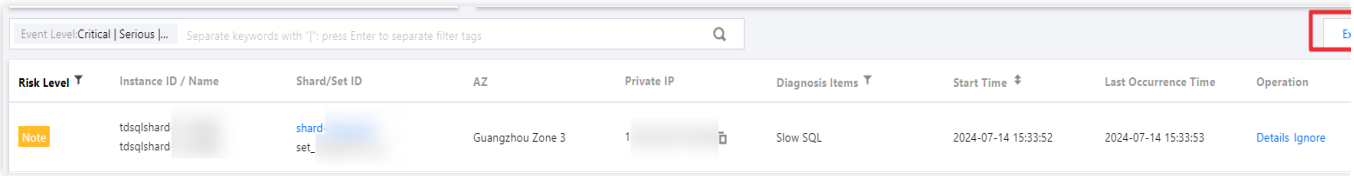
Alarms that have been ignored will be grayed out.

In the exception alarm list's **Operation** bar, click **Unignore** to unignore.

Exporting the Exception Alarm List

1. Log in to the DBbrain console.
2. In the left sidebar, choose **Monitoring & Alarm** > **Exception Alarm**.
3. On the top of the page, select the database type and region.

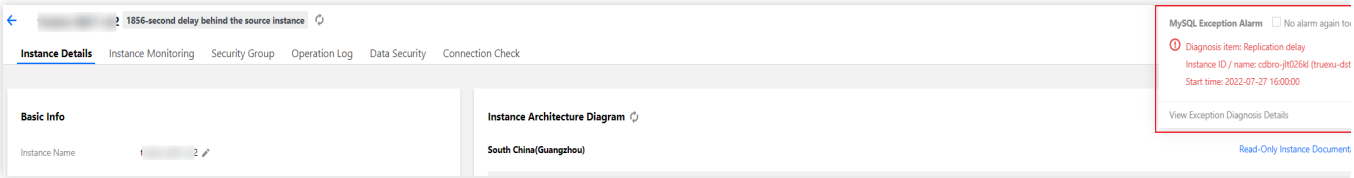
4. Select the time range for viewing alarms. Supported options are the last 3 hours, last 24 hours, last 7 days, and a custom time range.
5. At the top of the exception alarm list, click **Export** to export the exception alarm list in .csv format.
Up to 10,000 pieces of alarm data can be exported.



Viewing an Alarm from a Database

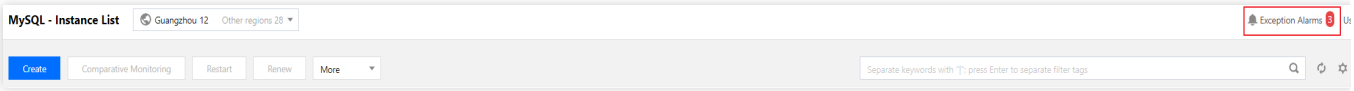
Option 1

Log in to the supported database console. If an instance has an exceptional diagnosis issue at the current time, a pop-up window will be pushed in real time in the top right corner of the console to notify you. The message contains the instance ID/name, diagnosis item, and start time, allowing you to quickly understand the instance's diagnostic problems.
This document takes logging in to the MySQL console as an example to view alarms.
1. Log in to the MySQL console.
2. View exception alarms in the pop-up window on the top right of the page.
Click **View Exception Diagnosis Details** in the message notification to view the specific diagnostic details and optimization suggestion for the instance.
If you check **No alarm again today** in the message notification, when an exception diagnostic problem occurs in a database instance under your account, no exception alarm messages will be pushed to you in a pop-up window.

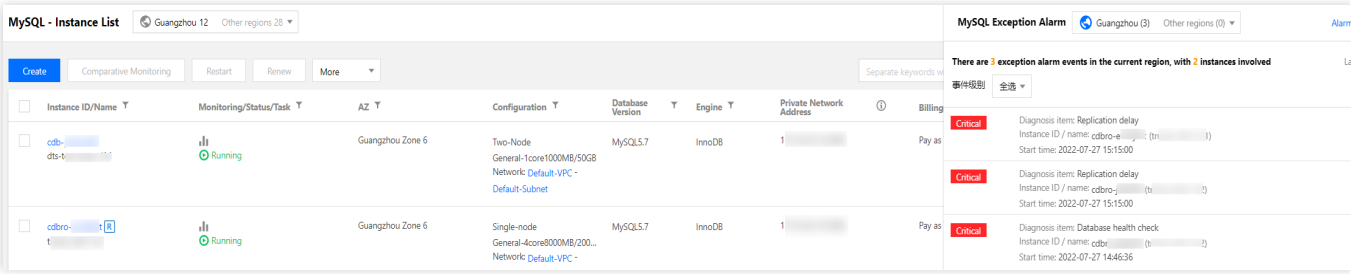


Option 2

1. Log in to the MySQL console.
2. In the left sidebar, choose **Instance List** , **Task List** , **Parameter Templates** , **Recycle Bin** , or **Placement Group**. The number of exception alarms is displayed in the top right corner. Click **Exception Alarms** to expand the list of historical exception alarm messages.



In the unfolded list of historical exception alarm messages, you can view all pushed historical exception alarm messages. You can view them by region, and filter them by alarm level. You can also click a message to view the diagnostic details of the exception alarm event.



Event Notification

Last updated：2024-07-31 10:35:18

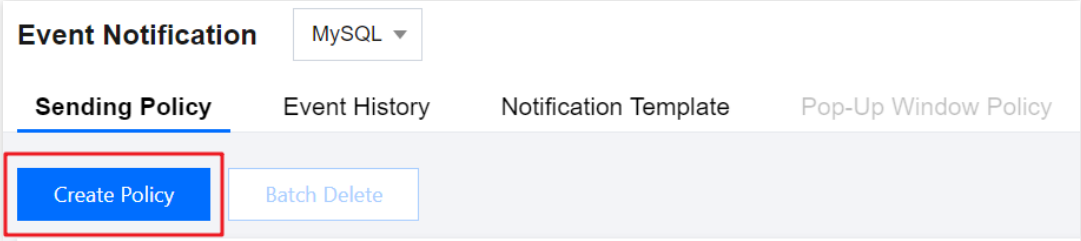
The event notification feature sends the diagnostic results of the DBbrain 24/7 exception diagnosis module to users through channels (currently supporting SMS, telephone, WeChat, WeCom, Email, Message Center) or through webhooks (currently supporting WeCom group bot webhook, DingTalk group bot webhook, Lark group bot webhook) to the respective WeCom groups, DingTalk groups, and Lark groups.
Users can configure diagnosis items, notification events, channels, and recipients according to their needs.

Note:

DBbrain event notification is fundamentally different from TCOP alarms. TCOP metric alarm feature monitors specific metrics and notifies users the corresponding metric alarm when the metrics reach the monitoring threshold. On the other hand, DBbrain event notification informs users about the diagnostic results from the DBbrain exception diagnosis module. To receive notifications based on exact values of the metrics, use the TCOP alarm system.
If you have previously created TCOP [DBbrain exception monitoring event], there will be conflicts with this system's exception events. After you create this one, it is recommended to delete the TCOP [DBbrain exception monitoring event]. Otherwise, you will receive multiple event notifications.
Currently, the event notification feature is only available for TencentDB for MySQL, TDSQL-C for MySQL, TencentDB for Redis, and TencentDB for MongoDB.

Creating an Event Notification Policy

1. Log in to the DBbrain console.
2. In the left sidebar, choose **Monitoring & Alarm** - **Event Notification** .
3. On the top of the page, select the database type. Select the **Sending Policy** tab and click **Create Policy** .



4. Configure the policy according to the interface prompts.

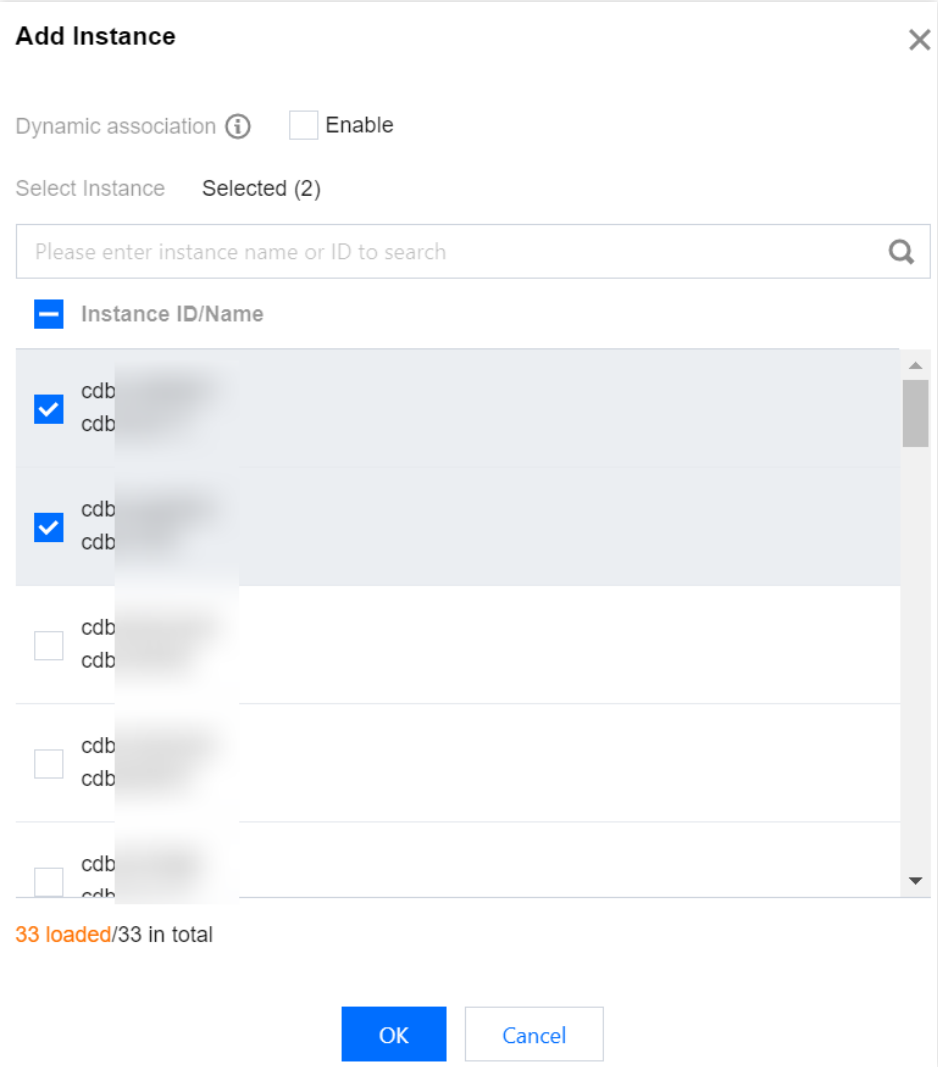
4.1 Select the database type.

4.2 Configure basic information.

Policy name: Required. Enter the policy name. Naming rule: It can contain Chinese characters, letters, digits, and the symbols ()_-(), but cannot start with an underscore. It should be no more than 60 characters in length.

4.3 Associate instances.

Click **Select Instance** . In the pop-up window, select the instance and click **OK** .



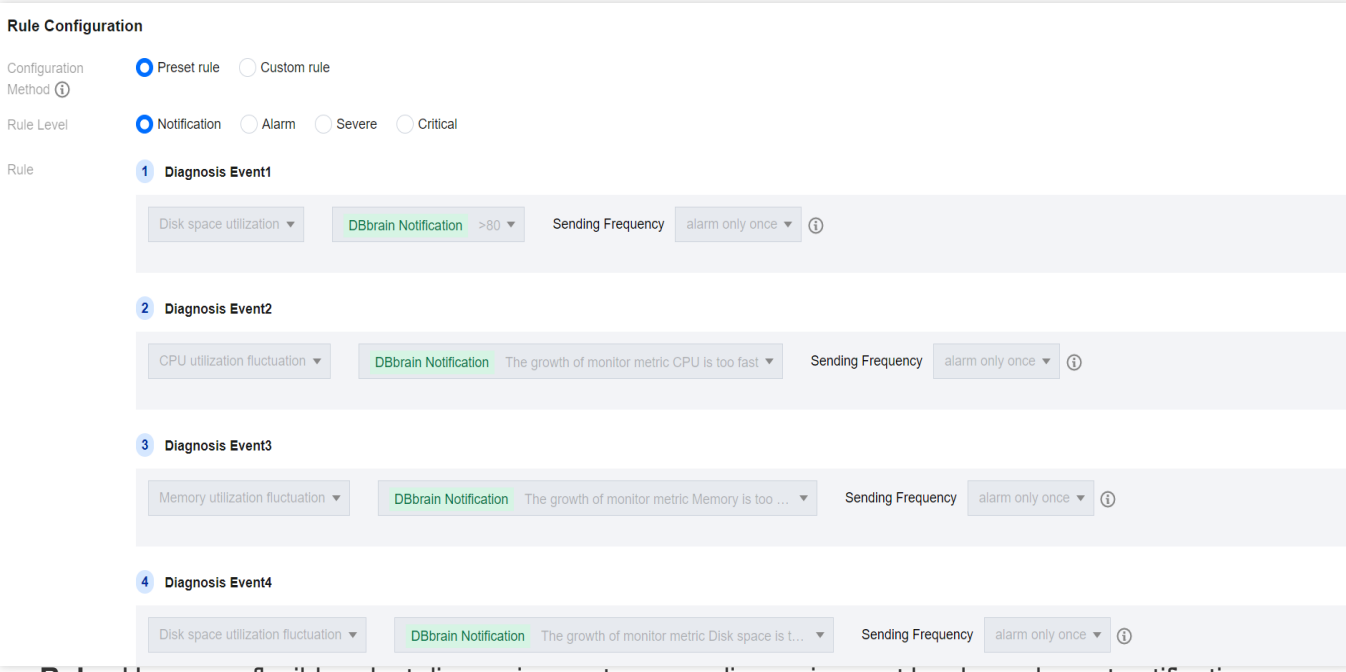
Select whether to enable dynamic association of instances: After dynamic association is enabled, all instances will be automatically selected for you. If there are newly added instances under your account, they will be dynamically loaded into this policy configuration.

Instances to be manually associated: Supports selecting one or multiple instances.

4.4 Rule configuration.

Rule configuration includes two methods: Preset rule and custom rule.

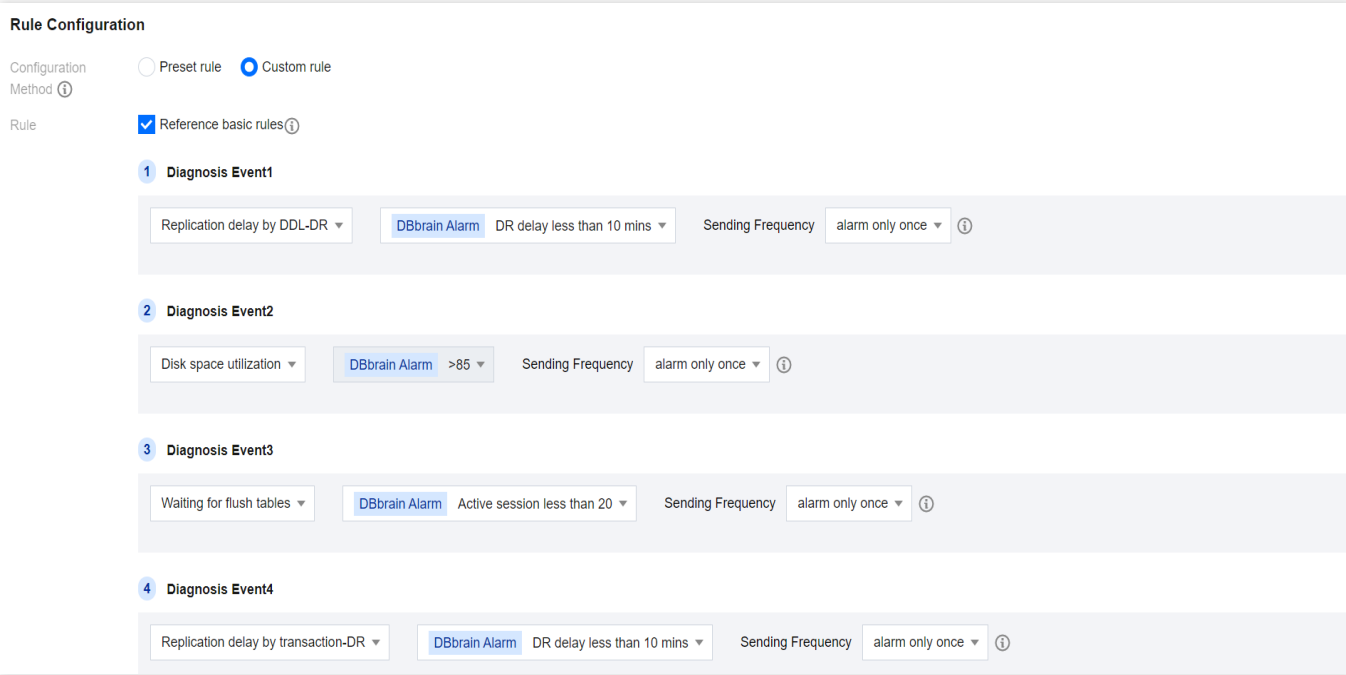
Preset rule : DBbrain provides four levels (notification, alarm, severe, and critical), each containing corresponding diagnosis event content. Users can select any one of the four levels, but the content cannot be modified.



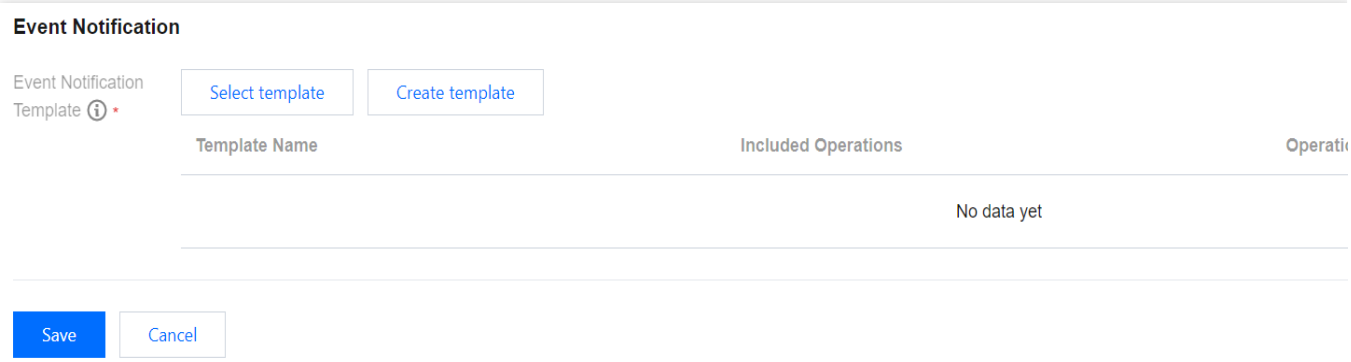
Custom Rule : Users can flexibly select diagnosis event names, diagnosis event levels, and event notification sending frequencies according to their needs.

For custom rules, **Reference basic rules** is checked by default. It allows users to freely modify diagnosis event names, diagnosis event levels, and event notification sending frequencies based on the base rules. If base rules are not needed, the **Reference basic rules** option can be unchecked.

Additionally, it supports clicking **Add Metric** to continue adding diagnosis events. It also supports clicking **Delete** to remove diagnosis events.



4.5 Event notification configuration.



Notification templates include selecting existing notification templates and quick configuration of notification templates.

Selecting Template

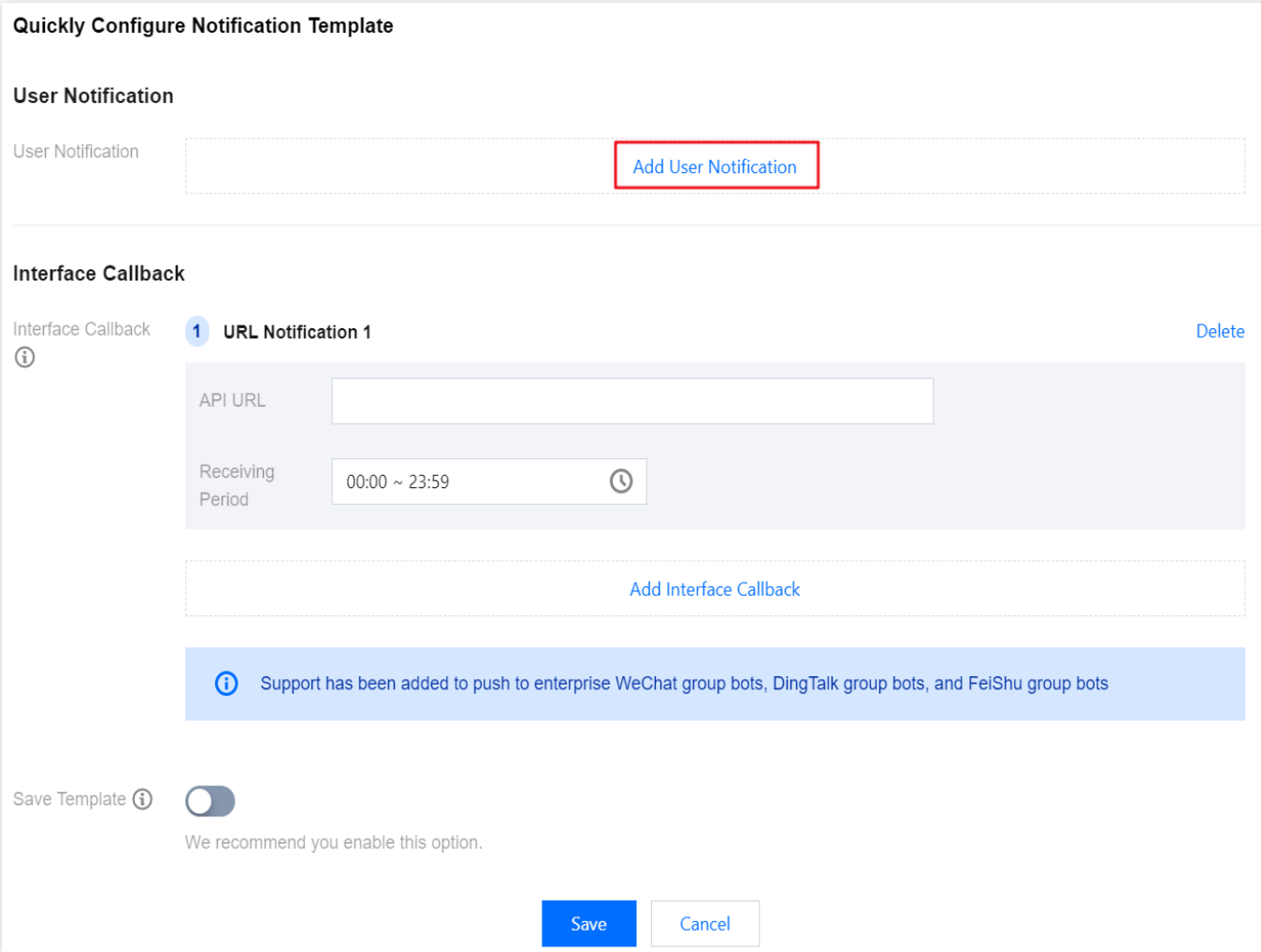
Click **Select template** , and in the pop-up dialog box, select the template name, and then click **OK** . This requires a configured notification template on the **Event Notification** - **Notification Template** page. For detailed operations, see managing notification templates.

Quick Configuration

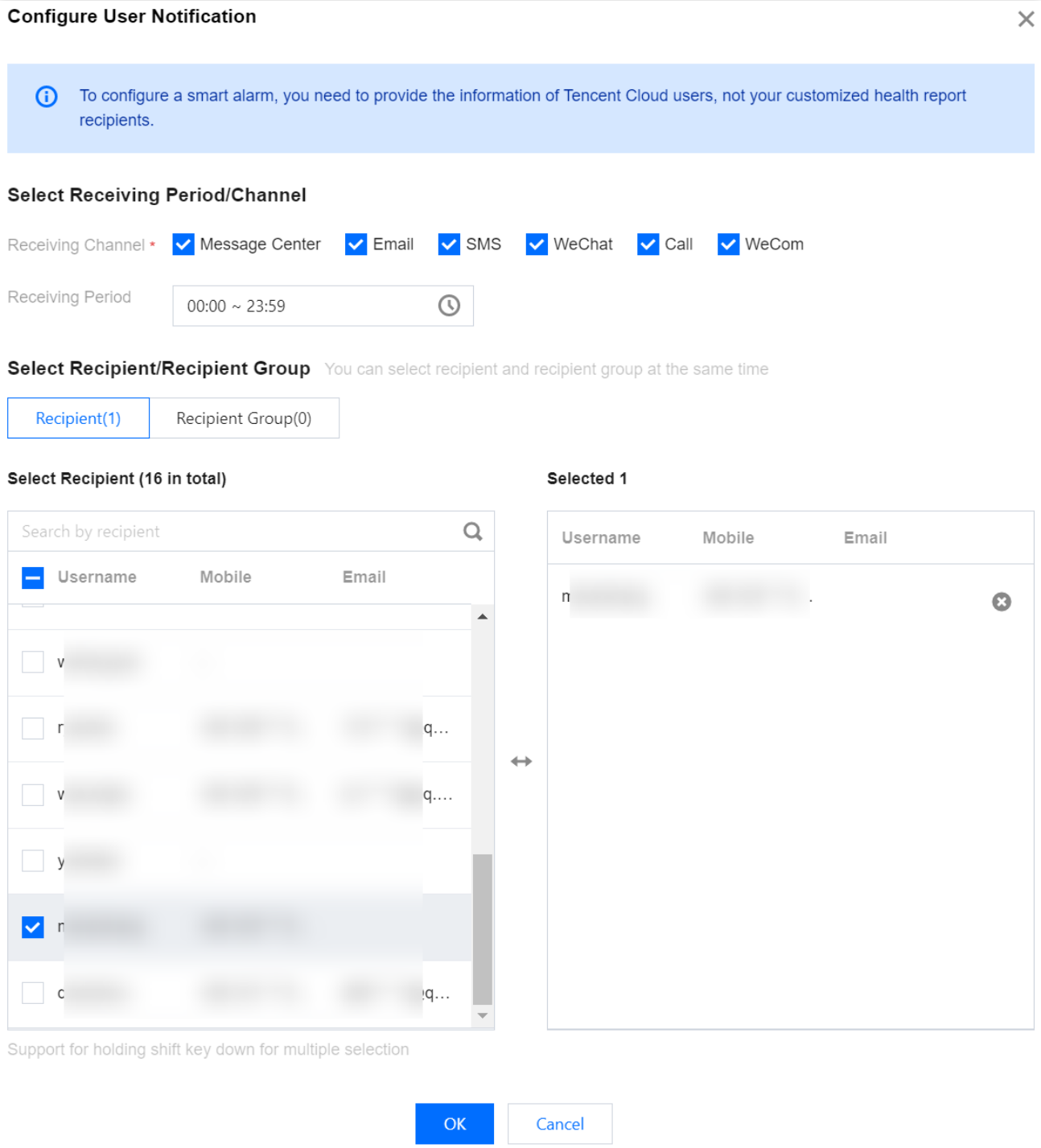
4.5.1.1 Click **Create template.**

4.5.1.2 Configure user notification.

4.5.1.2.1 In the **Quickly Configure Notification Template** dialog box, click **Add User Notification** .



4.5.1.2.2 In the pop-up **Configure User Notification** dialog box, select the receiving channel, receiving period, and recipient details, and then click **OK** .



If you want to continue adding, click **Add User Notification** . You can configure up to 5 sets of user notifications. The added user notifications support editing and deleting.

4.5.1.3 Configure webhook URL.

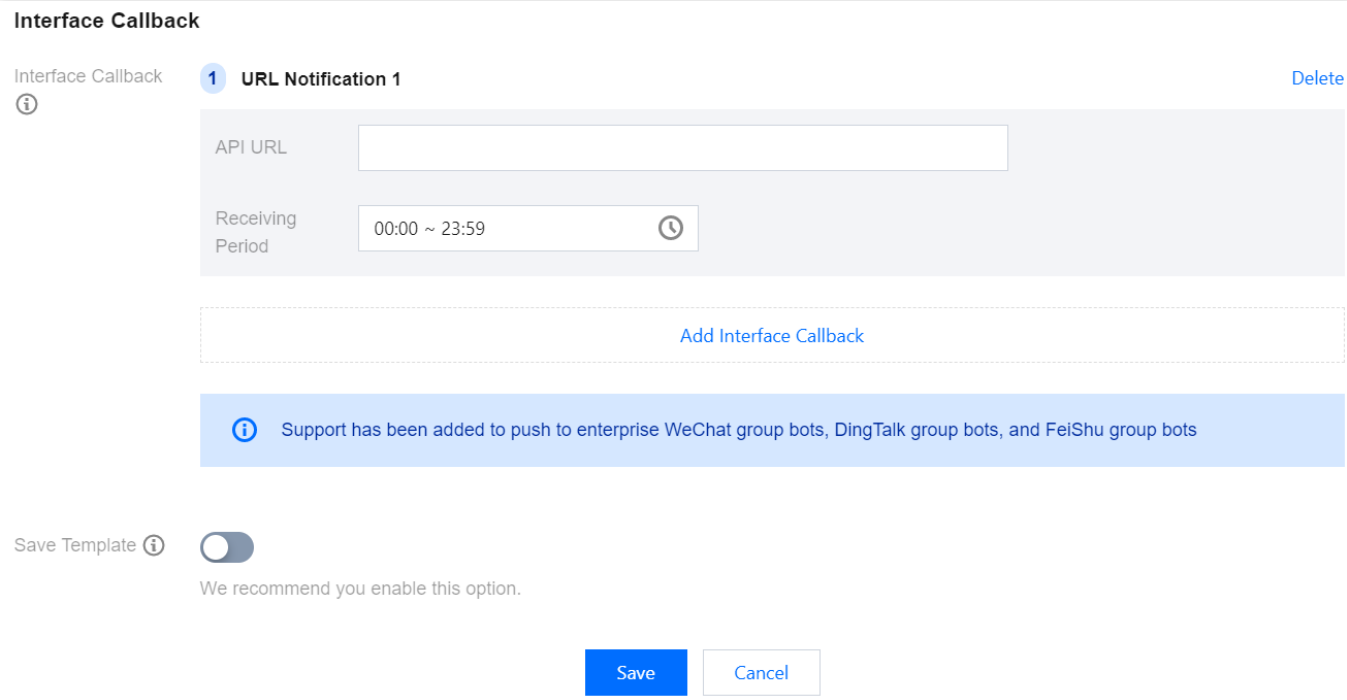
Note:

You can fill in the public-network-accessible WeCom group bot webhook, DingTalk group bot webhook, and Lark group bot webhook. DBbrain event notifications will promptly push alarm information to the corresponding WeCom groups, DingTalk groups, and Lark groups.

If the alarm push fails, it will retry up to 3 times, with a timeout waiting of 1 second for each push request.

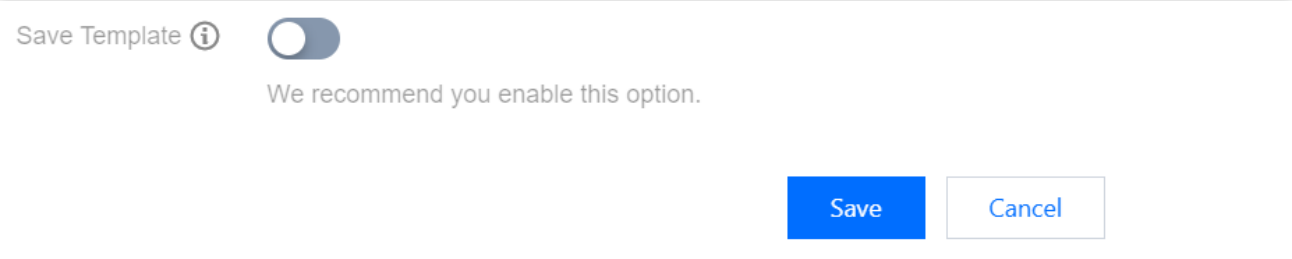
Each bot has message sending limits. For example, the WeCom group bots can send up to 20 messages per minute. Messages exceeding this limit will be discarded. See the official documentation for the limits of DingTalk and Lark.

In the **Interface Callback** area, enter the webhook API URL and select the notification receiving period.



If you need to configure multiple webhook URLs, click **Add Interface Callback** . You can configure up to 5 webhook URLs. The added URL notifications support URL modification and deletion.

4.5.1.4 In the **Quickly Configure Notification Template** dialog box, confirm the user notification information. Select whether to save this as a notification template, and click **Save** .



If saving is enabled, the template will be displayed on the **Event Notification - Notification Template** page and can be directly referenced next time after it is saved.

If saving is not enabled, the user notification will be a one-time configuration, and the configured user information will not be viewable when users view policy details later.

5. After the policy configuration is completed, click **Save** on the bottom of the page.

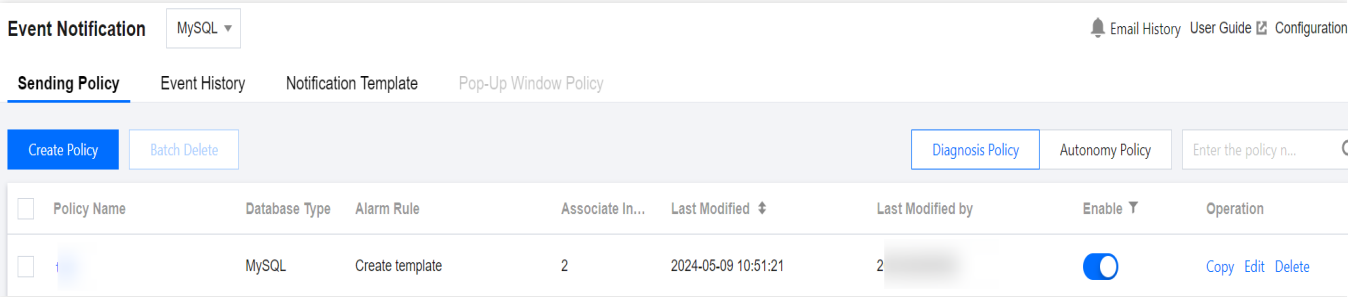
If successfully saved, the policy list will display the name of the newly created policy and the policy will be enabled by default.

Managing an Event Notification Policy

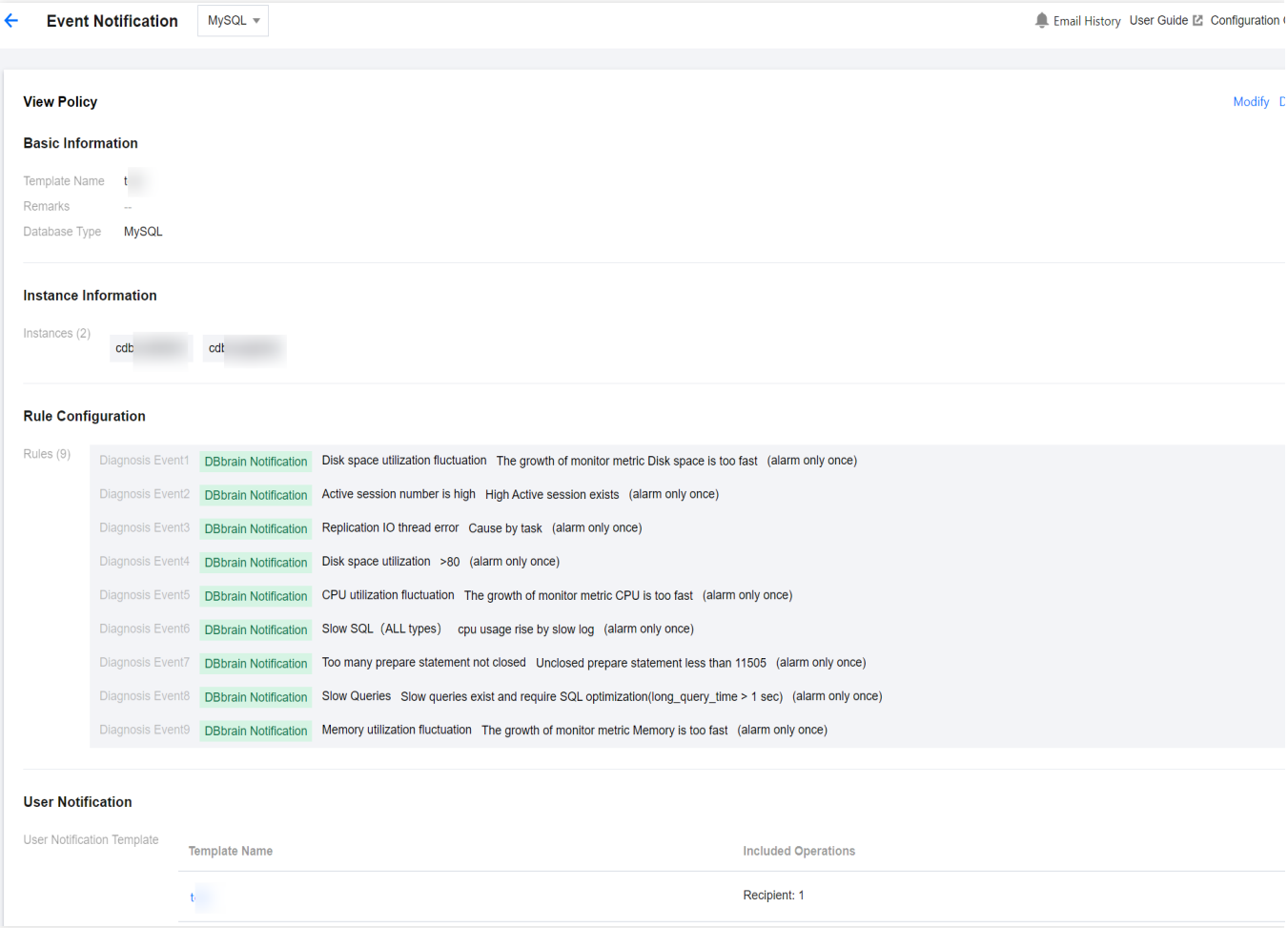
Supports viewing policy details, disabling or enabling policies, copying policies, editing policies, and deleting policies.

Viewing Policy Details

1. Log in to the DBbrain console.
2. In the left sidebar, choose **Monitoring & Alarm** - **Event Notification** .
3. On the top of the page, select the database type. Select the **Sending Policy** tab to view the configured policies.

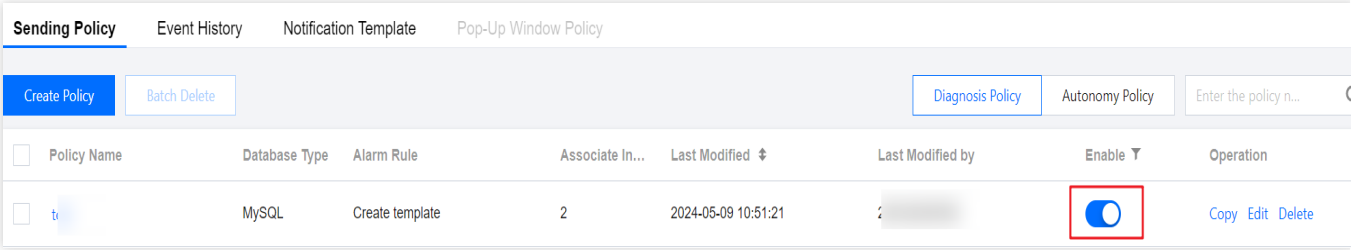


4. Click the specified policy name to view policy details.



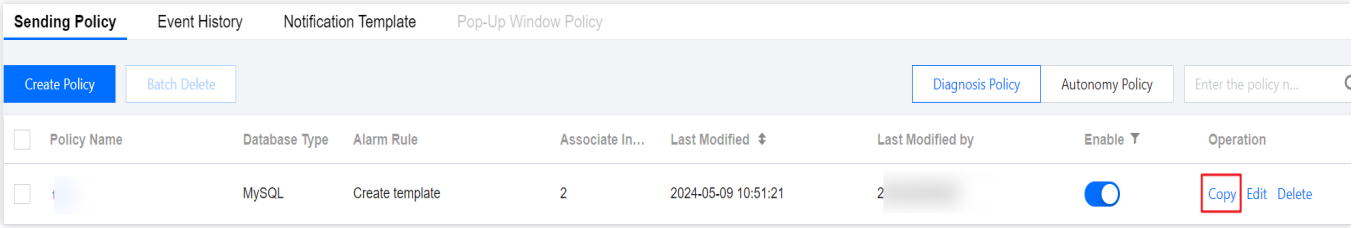
Disabling or Enabling Policies

By default, a policy is enabled after created. If you need to disable this policy, turn off the switch in the **Enable** column in the policy list of the **Sending Policy** tab. To enable it again, simply turn on the switch back.



Copying a Policy

If you need to create a new policy based on an existing one, click **Copy** in the **Operation** column in the policy list of the **Sending Policy** tab. You can then set the policy name and freely modify the original policy configuration information.



Editing a Policy

When you need to modify an already configured policy, the following two methods are supported:

In the **Sending Policy** tab, click the policy name you want to edit and then click **Modify** on the top of the policy details page.

In the **Sending Policy** tab, find the policy name you want to edit and click **Edit** in the corresponding **Operation** bar.

Deleting a Policy or Policies

When you no longer need one or more policies, you can select single deletion or batch deletion.

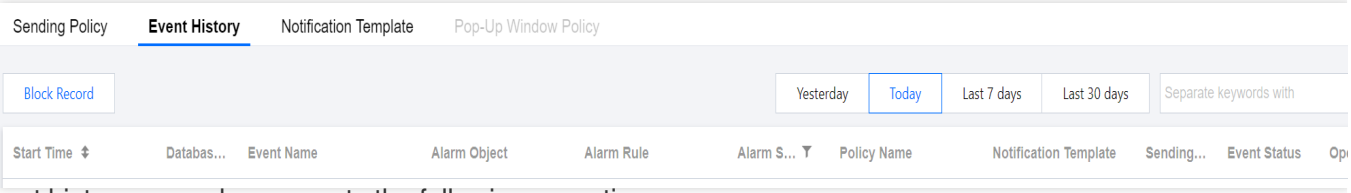
Single deletion: In the **Sending Policy** tab, find the name of the policy to be deleted, and in the corresponding **Operation** bar, click **Delete** .

Batch deletion: In the **Sending Policy** tab, select one or more policies to be deleted, and on the top of the page, click **Batch Delete** .

Viewing the Event History

1. Log in to the DBbrain console.
2. In the left sidebar, choose **Monitoring & Alarm** - **Event Notification** .

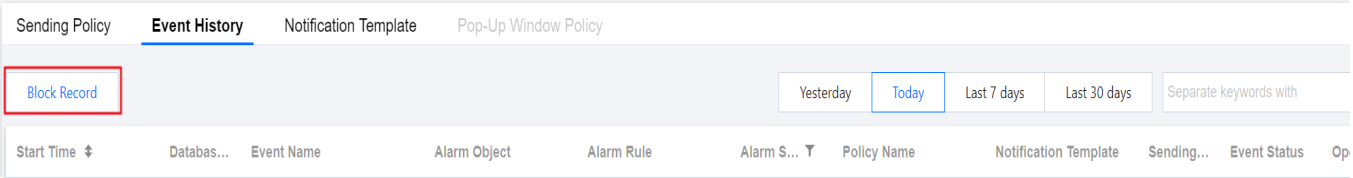
3. On the top of the page, select database type. Select the **Event History** tab, and on the top right of the page, select the time range to view the history of event notifications. The default interface displays the events of the day.



The event history page also supports the following operations:

Temporary block event : For events that are being triggered and continuously notified, users can temporarily block them by clicking **Hide** in the action bar corresponding to the event history. The maximum duration for a single block is 24 hours.

All blocked records can be viewed by clicking **Block Record** on the top left of the page.



Navigate to event details page : In the event history list, click the event name.

Navigate to event policy details page : In the event history list, click the policy name.

Managing a Notification Template

After you create a notification template, you can directly bind it when creating a policy.

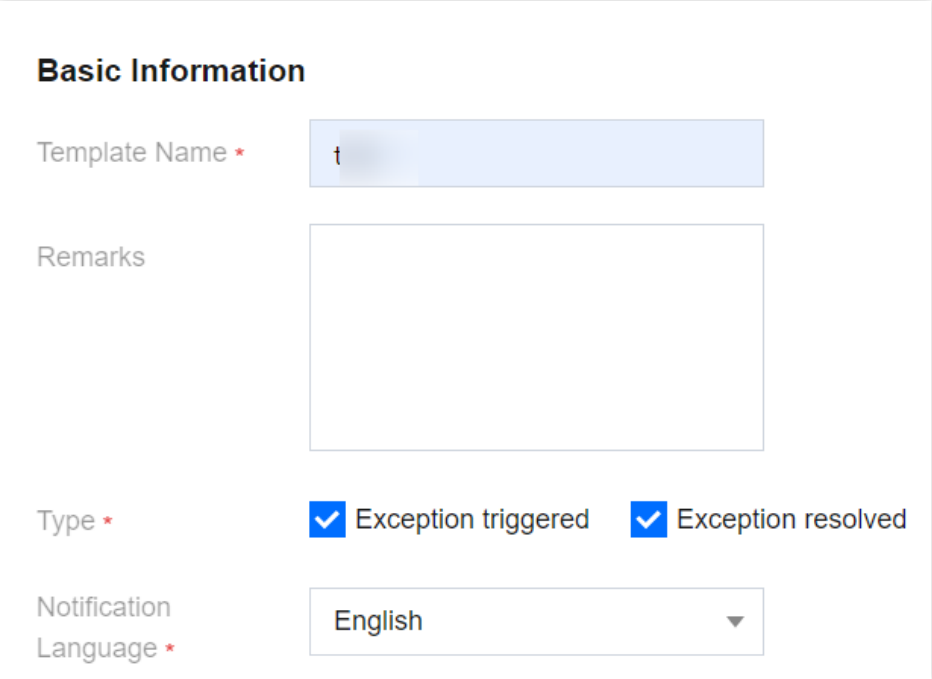
The notification template supports the following two methods:

Configure user notifications: Configure the notification time, channel, and recipients.

Configure the webhook URL: Set the public-network-accessible WeCom group bot webhook, DingTalk group bot webhook, and Lark group bot webhook. DBbrain event notifications will promptly push alarm information to the corresponding WeCom groups, DingTalk groups, and Lark groups.

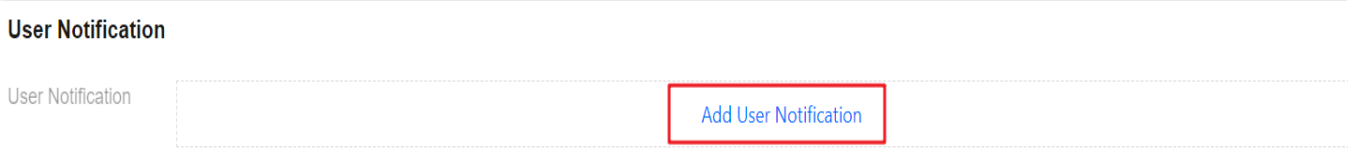
Creating a Notification Template

1. Log in to the DBbrain console.
2. In the left sidebar, choose **Monitoring & Alarm** - **Event Notification** .
3. On the top of the page, select the database type. Select the **Notification Template** tab, and click **Create Template** to open the template configuration page.
4. Configure the template name, and select the notification type and notification language.

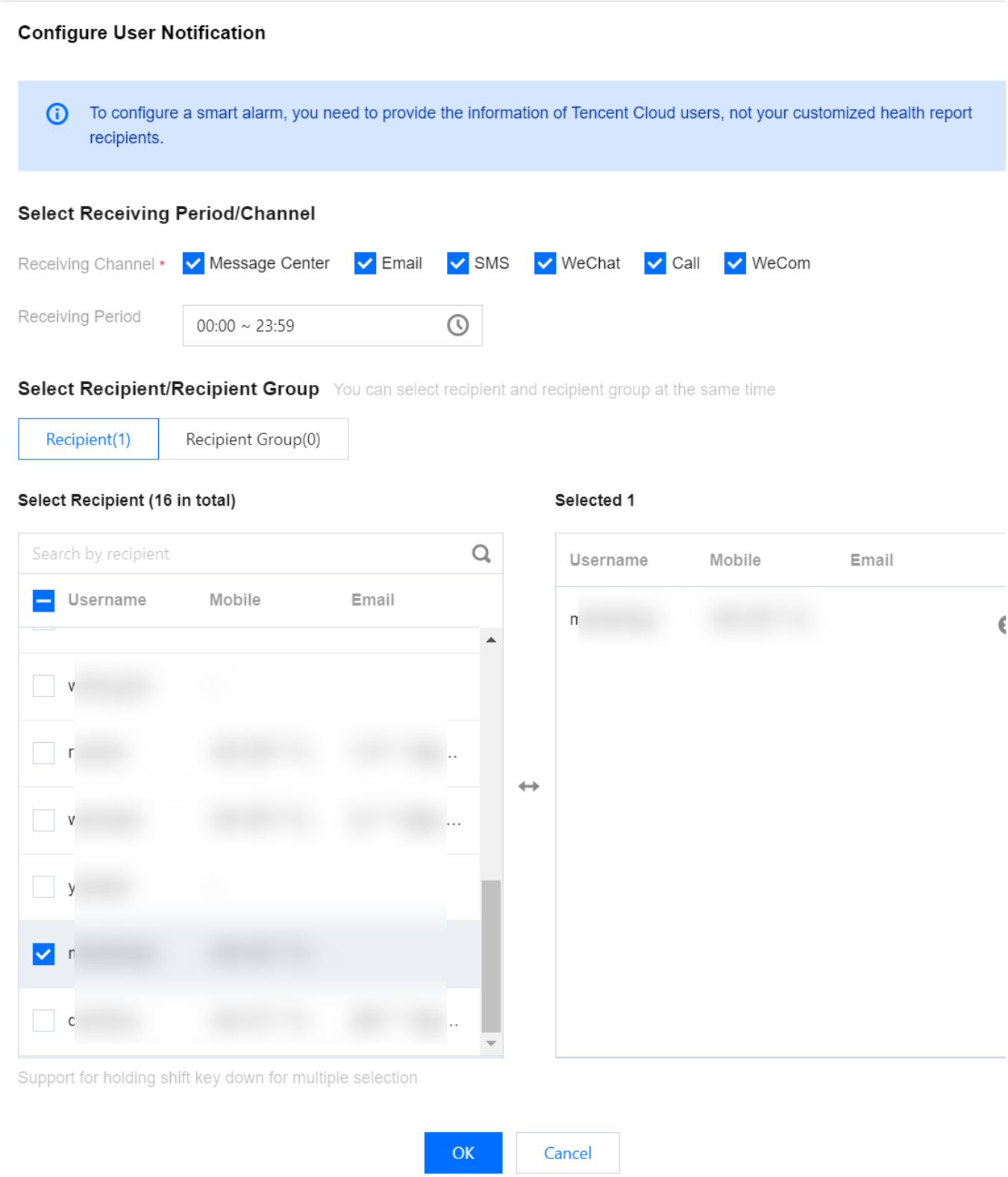


5. Configure user notifications.

5.1 In the **User Notification** section, click **Add User Notification** .



5.2 In the pop-up **Configure User Notification** dialog box, select the receiving method, receiving period, and recipient details, then click **OK** .



If you want to continue adding, click **Add User Notification** . You can configure up to 5 sets of user notifications. The added user notifications support editing and deleting.

6. Configure webhook URL.

Note:

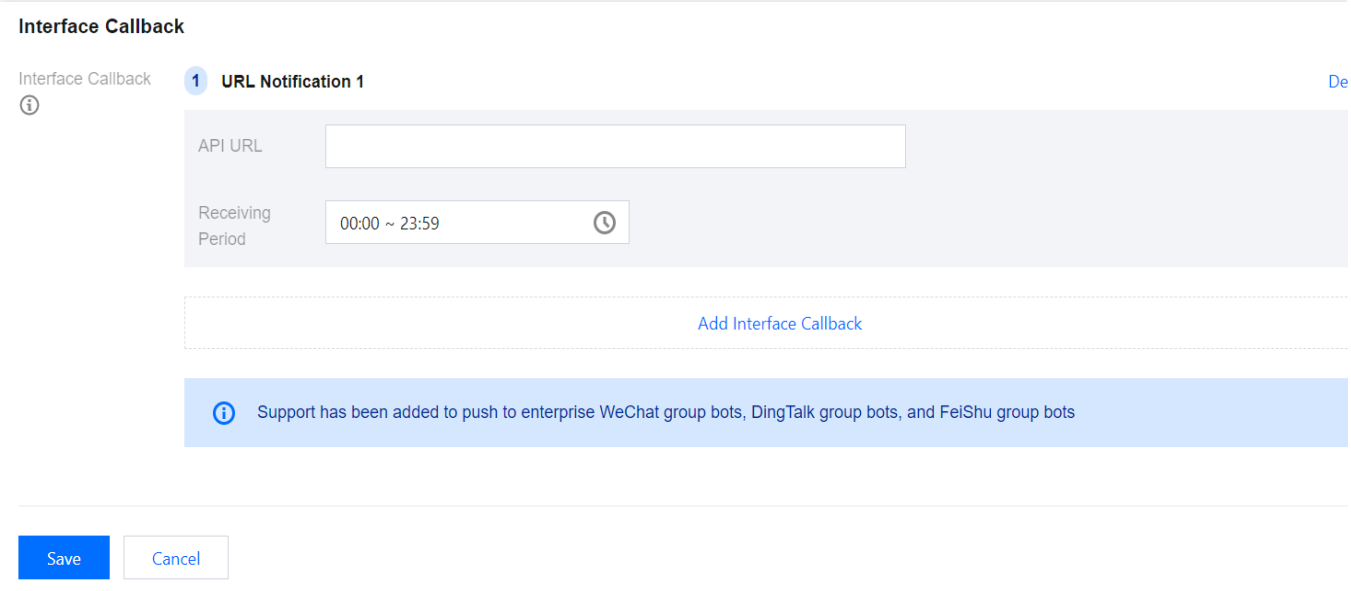
You can fill in the public-network-accessible WeCom group bot webhook, DingTalk group bot webhook, and Lark group bot webhook. DBbrain event notifications will promptly push alarm information to the corresponding WeCom

groups, DingTalk groups, and Lark groups.

If the alarm push fails, it will retry up to 3 times, with a timeout waiting of 1 second for each push request.

Each bot has message sending limits. For example, the WeCom group bots can send up to 20 messages per minute. Messages exceeding this limit will be discarded. See the official documentation for the limits of DingTalk and Lark.

In the **Interface Callback** area, enter the webhook API URL and select the notification receiving period.

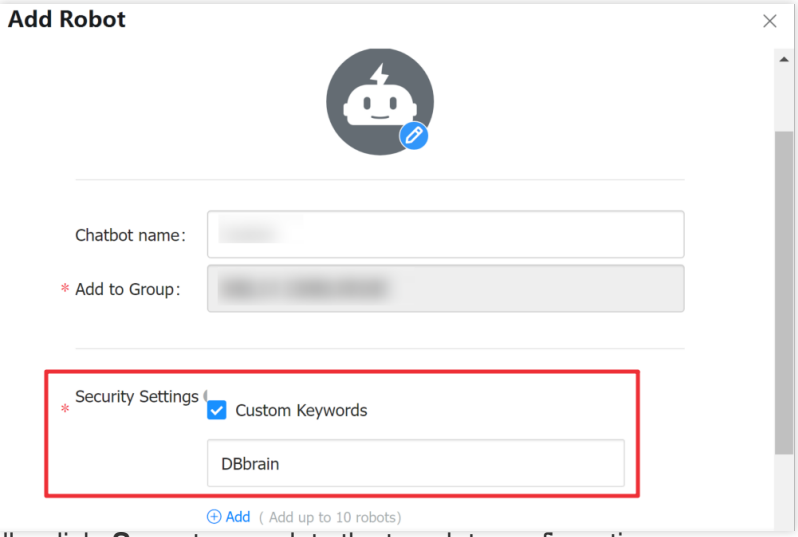


If you need to configure multiple webhook URLs, click **Add Interface Callback** . You can configure up to 5 webhook URLs. The added URL notifications support URL modification and deletion.

Note:

If the push verification fails, check for the following issues:

1. Incorrect URL: Provide the correct URL.
2. Security settings not enabled for reception service: Add the keyword DBbrain to the security settings. Example: In the DingTalk group bot webhook security settings, check the custom keyword option and enter DBbrain.

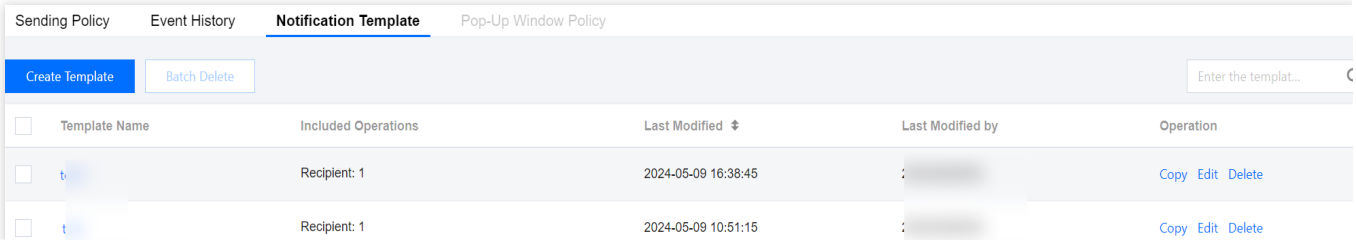


7. Finally, click **Save** to complete the template configuration.

Viewing/Copying/Editing/Deleting a Notification Template

You can click the template name to view the template details.

In the **Operation** column of the selected template, you can click **Copy**, **Edit**, or **Delete** to copy, edit, or delete the template.



Intelligent Monitoring (Monitoring Dashboard)

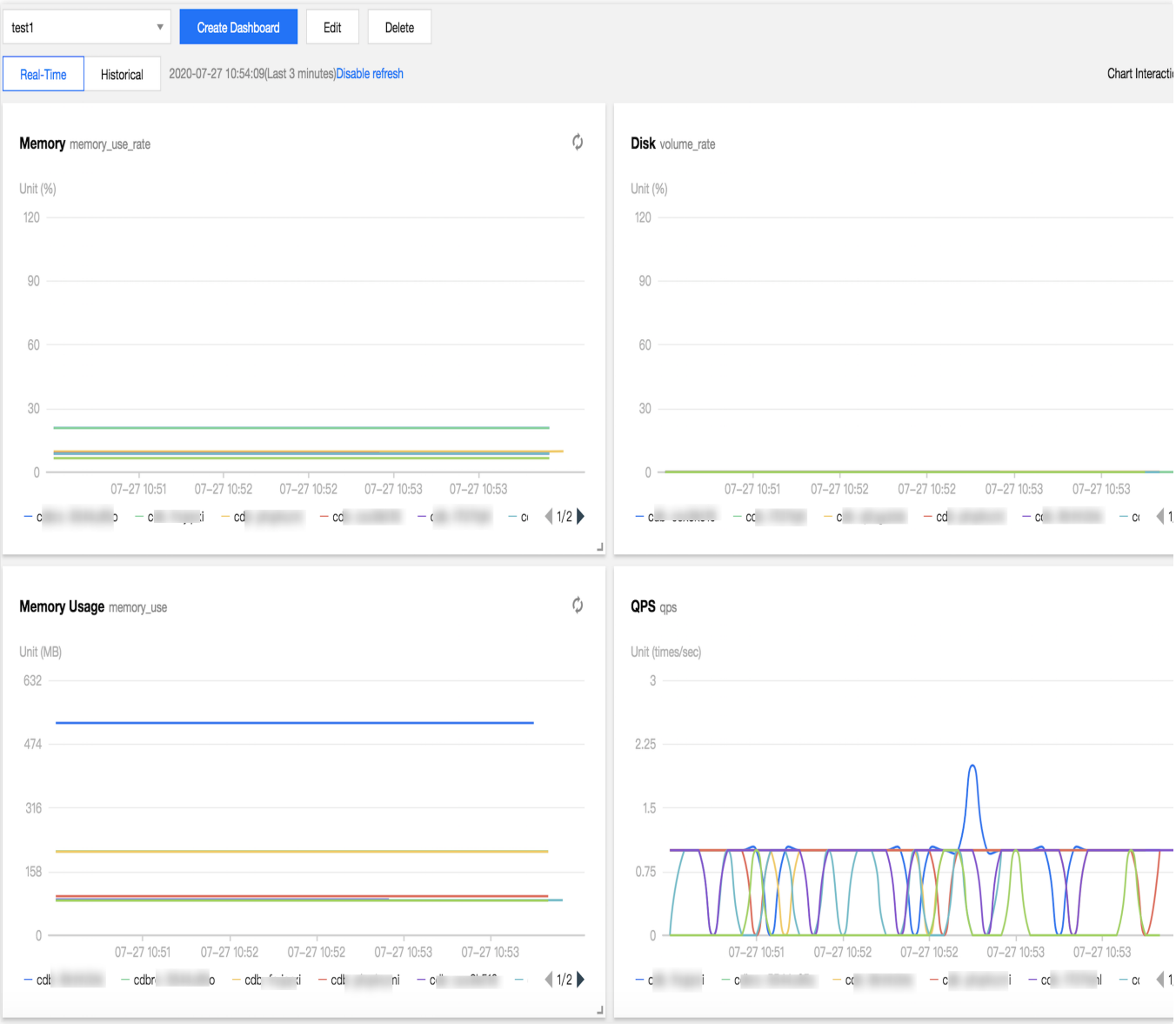
Last updated：2022-09-01 18:34:46

Feature Description

DBbrain allows you to customize the monitoring dashboard and link, compare, and view the monitoring data of multiple instances and metrics.

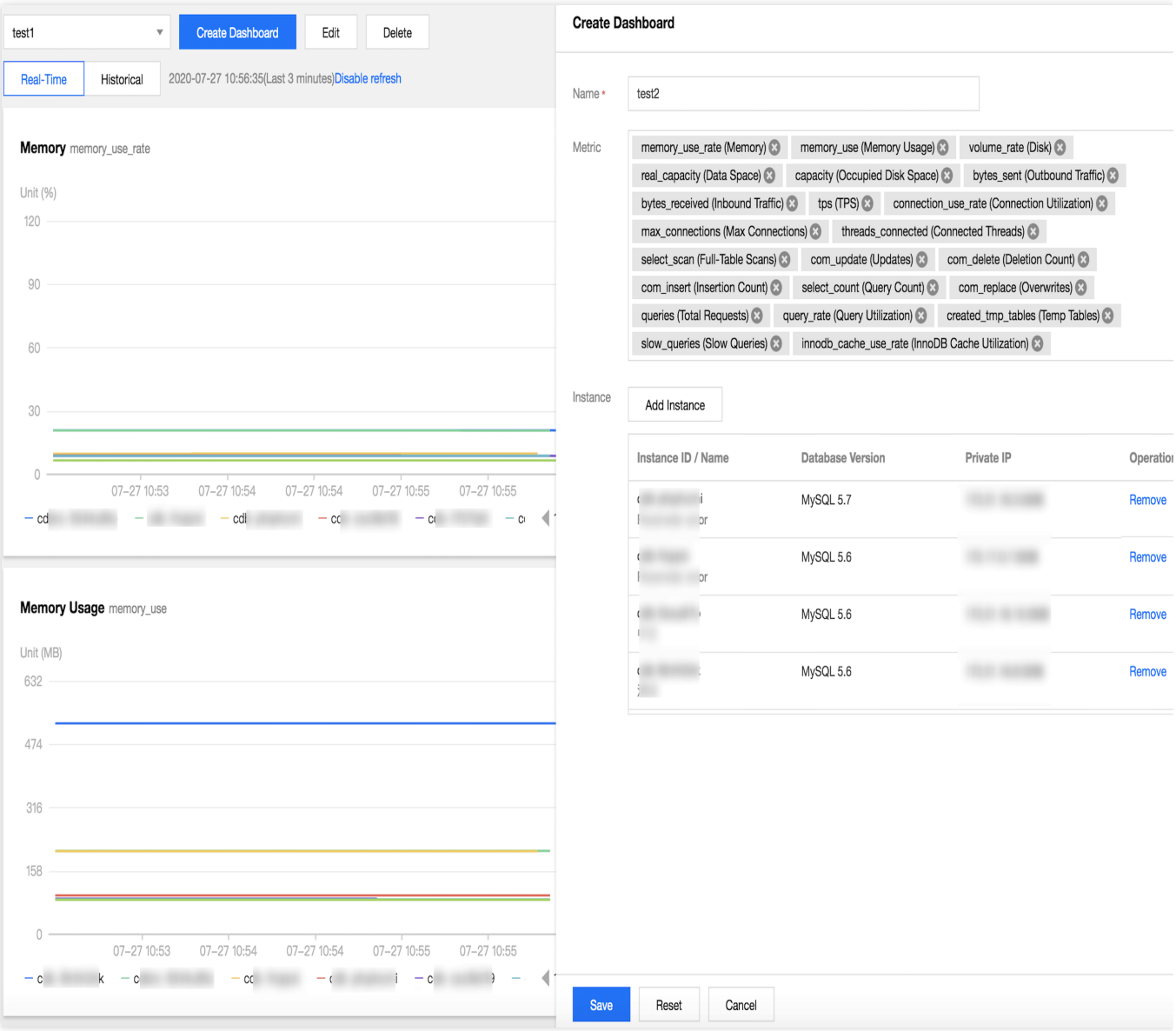
Note:

Currently, the monitoring dashboard feature is supported for TencentDB for MySQL (excluding basic single-node instances), TDSQL-C for MySQL, self-built MySQL, TencentDB for Redis, and TencentDB for MongoDB.



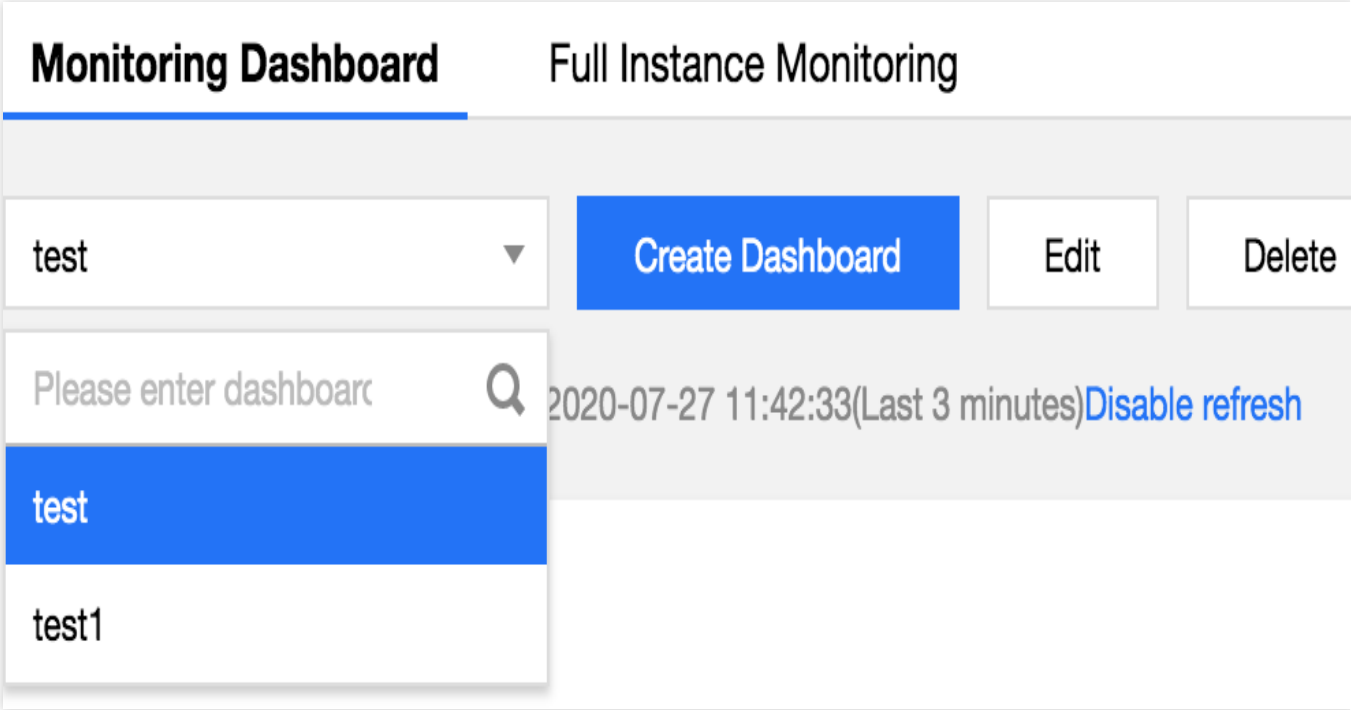
Creating a Dashboard

1. Log in to the DBbrain console and select **Monitoring and Alarming** > **Intelligent Monitoring** on the left sidebar. On the displayed page, select a database type and an instance at the top, and select the **Monitoring Dashboard** tab.
2. Click **Create Dashboard**, enter the dashboard name, select the monitoring metrics for comparison, add an instance, and click **Save**.



Finding/Editing/Deleting a Dashboard

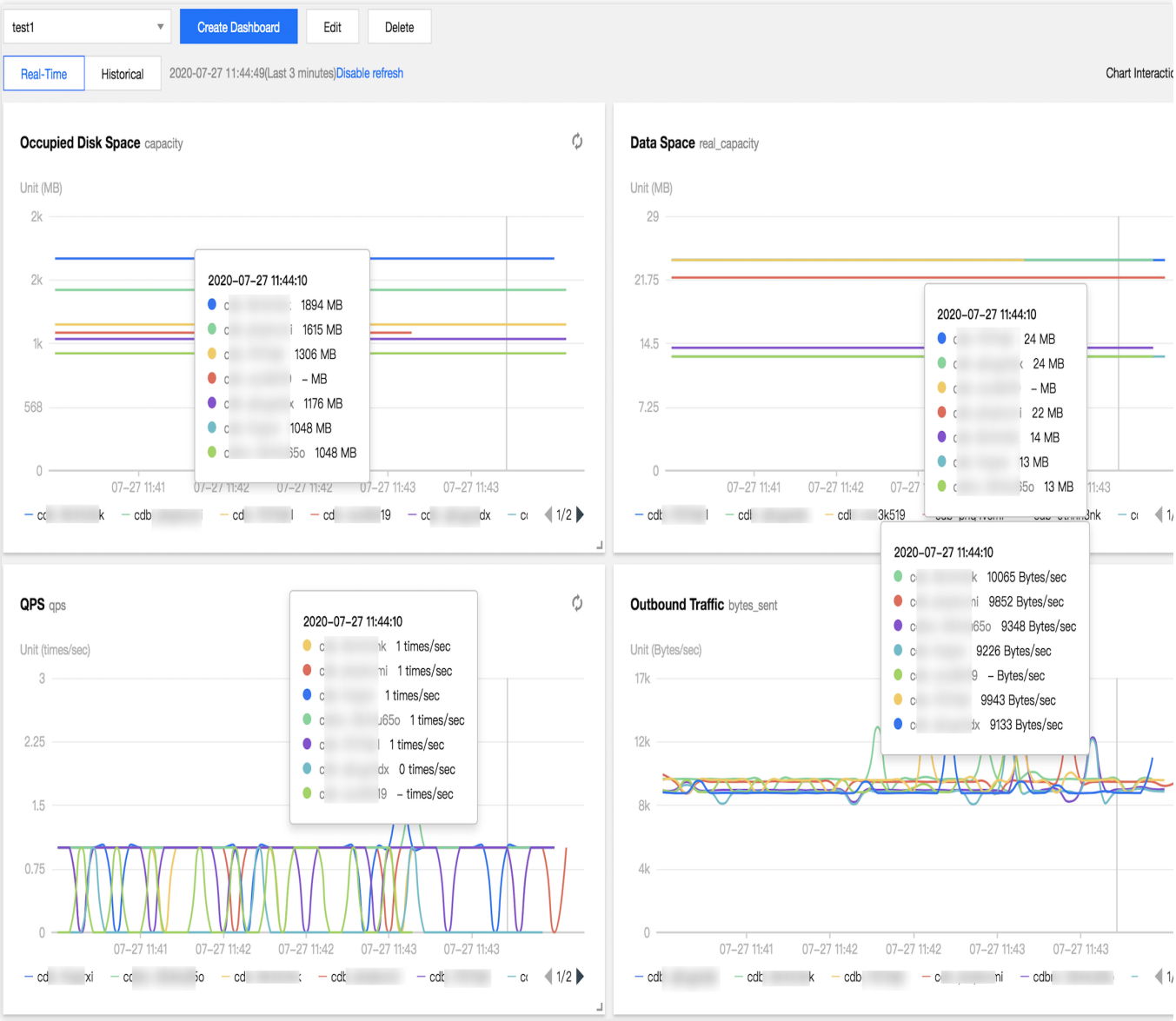
Log in to the DBbrain console and select **Monitoring and Alarming** > **Intelligent Monitoring** on the left sidebar. On the displayed page, select a database type and an instance at the top, and select the **Monitoring Dashboard** tab.
Click the dashboard name drop-down list to switch between different monitoring dashboards.
After selecting a dashboard, click **Edit** to modify its monitoring metrics and instance.
Click **Delete** to delete the current dashboard.



Viewing Dashboard Details

Enabling chart interaction

1. Log in to the DBbrain console and select **Monitoring and Alarming** > **Intelligent Monitoring** on the left sidebar. On the displayed page, select a database type and an instance at the top, and select the **Monitoring Dashboard** tab.
2. Toggle on **Chart Interaction** on the right to link and compare the monitoring views of multiple instances or metrics. When you hover over a data point in any monitoring view, the data at the same time point will be displayed in other monitoring views. Click the data point to pin it for display. To unpin it, click **Deselect the Time Point**.



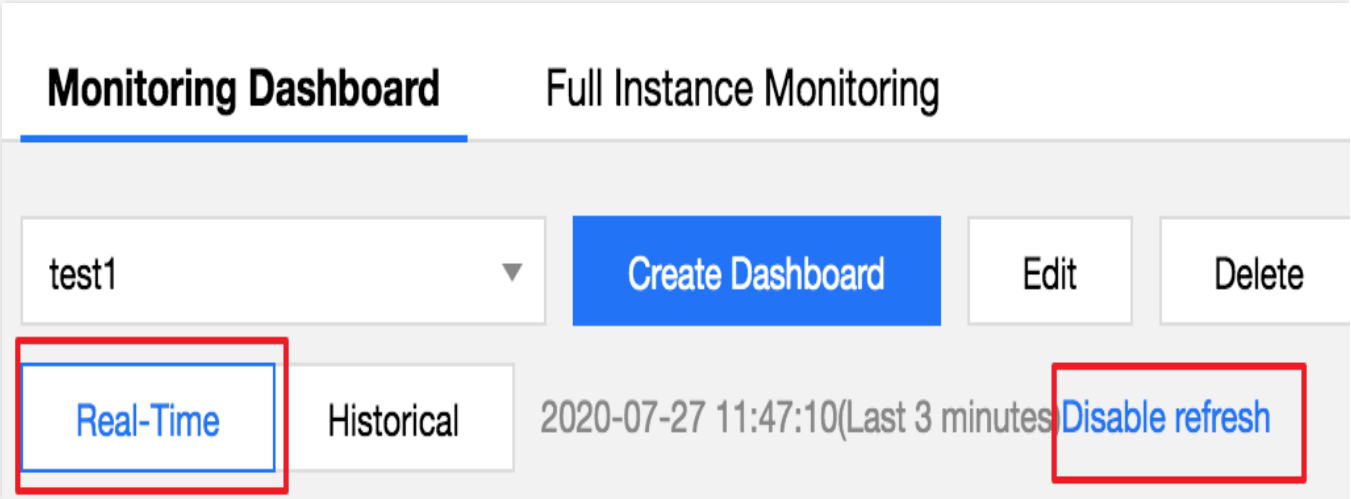
Switching between one-column and two-column modes

1. Click the button on the right of **Chart Interaction** in the top-right corner to switch.
2. Click the border of a monitoring view to drag it to the desired position.

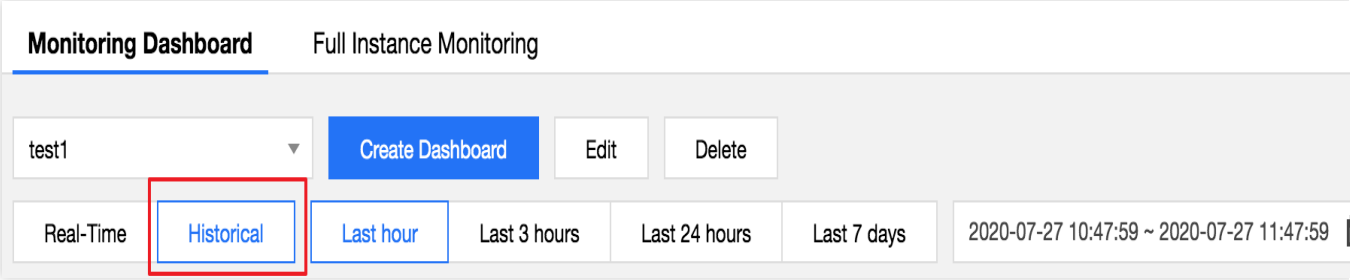
Switching between real-time and historical views

Click **Real-Time** or **Historical** to view the real-time or historical monitoring view.

The real-time monitoring view displays the performance metric comparison of the instance in the last three minutes and is automatically refreshed by default. You can click **Disable refresh** to stop refreshing the monitoring data in real time.



In the historical monitoring view, you can select a time range (**Last hour**, **Last 3 hours**, **Last 24 hours**, **Last 7 days**, or a custom time range) to display the monitoring dashboard in the selected time range.



Monitoring Metrics

DBbrain (TencentDB for MySQL)

In DBbrain, the custom monitoring dashboard for TencentDB for MySQL currently supports the following monitoring metrics:

Monitoring Metric	Description
cpu_use_rate	CPU Utilization
memory_use_rate	Memory Utilization
memory_use	Memory Usage
volume_rate	Disk Utilization
real_capacity	Used Disk Space
capacity	Occupied Disk Space
bytes_sent	Outbound Traffic

bytes_received	Inbound Traffic
qps	QPS
tps	TPS
connection_use_rate	Connection Utilization
max_connections	Max Connections
threads_connected	Connected Threads
slow_queries	Slow Queries
select_scan	Full-Table Scans
select_count	Queries
com_update	Updates
com_delete	Deletions
com_insert	Insertions
com_replace	Overwrites
queries	Total Requests
query_rate	Query Utilization
created_tmp_tables	Temp Tables
table_locks_waited	Table Locks Awaited
innodb_cache_hit_rate	InnoDB Cache Hit Rate
innodb_cache_use_rate	InnoDB Cache Utilization
innodb_os_file_reads	InnoDB Disk Reads
innodb_os_file_writes	InnoDB Disk Writes
innodb_os_fsyncs	InnoDB fsync Count
innodb_num_open_files	InnoDB Opened Tables
key_cache_hit_rate	MyISAM Cache Hit Rate
key_cache_use_rate	MyISAM Cache Utilization

©2013-2025 Tencent Cloud International Pte. Ltd.

com_commit	Submissions
com_rollback	Rollbacks
threads_created	Created Threads
created_tmp_disk_tables	Temp Disk Tables
threads_running	Running Threads
created_tmp_files	Temp Files
handler_read_rnd_next	Requests of Reading Next Row
handler_rollback	Internal Rollbacks
handler_commit	Internal Submissions
innodb_buffer_pool_pages_free	InnoDB Empty Pages
innodb_buffer_pool_pages_total	Total InnoDB Pages
innodb_buffer_pool_read_requests	InnoDB Logical Reads
innodb_buffer_pool_reads	InnoDB Physical Reads
innodb_data_read	InnoDB Reads
innodb_data_reads	Total InnoDB Reads
innodb_data_written	InnoDB Writes
innodb_data_writes	Total InnoDB Writes
innodb_rows_deleted	InnoDB Rows Deleted
innodb_rows_inserted	InnoDB Rows Inserted
innodb_rows_updated	InnoDB Rows Updated
innodb_rows_read	InnoDB Rows Read
innodb_row_lock_time_avg	Average InnoDB Row Lock Acquiring Time
innodb_row_lock_waits	InnoDB Row Lock Waits
key_blocks_unused	Unused Blocks in Key Cache
key_blocks_used	Used Blocks in Key Cache

©2013-2025 Tencent Cloud International Pte. Ltd.

key_read_requests	Data Blocks Read by Key Cache
key_reads	Data Blocks Read by Disks
key_write_requests	Data Blocks Written into Key Cache
key_writes	Data Blocks Written into Disks
opened_tables	Opened Tables
table_locks_immediate	Table Locks Released Immediately
open_files	Total Opened Files
log_capacity	Log Space
slave_io_running	IO Thread Status
slave_sql_running	SQL Thread Status
master_slave_sync_distance	Source-Replica Delay Distance
seconds_behind_master	Source-Replica Delay Time

DBbrain (TDSQL-C for MySQL)

In DBbrain, the custom monitoring dashboard for TDSQL-C for MySQL currently supports the following monitoring metrics.

Monitoring Metric	Description
cpu_use_rate	CPU Utilization
memory_use_rate	Memory Utilization
memory_use	Memory Usage
volume_rate	Storage Utilization
real_capacity	Used Storage Space
qcache_hits	Cache Hits
qcache_hit_rate	Cache Hit Rate
capacity	Total Storage Space
bytes_sent	Outbound Traffic

©2013-2025 Tencent Cloud International Pte. Ltd.

bytes_received	Inbound Traffic
queries	QPS
com_commit	TPS
max_connections	Max Connections
threads_connected	Connected Threads
slow_queries	Slow Queries
select_scan	Full-Table Scans
select_count	Queries
com_update	Updates
com_delete	Deletions
com_insert	Insertions
com_replace	Overwrites
created_tmp_tables	Temp Tables
innodb_cache_hit_rate	InnoDB Cache Hit Rate
innodb_cache_use_rate	InnoDB Cache Utilization
threads_created	Created Threads
threads_running	Running Threads
handler_rollback	Rolled-Back Transactions per Second
innodb_buffer_pool_read_requests	InnoDB Logical Reads
handler_commit	Committed Transactions per Second
innodb_buffer_pool_write_requests	InnoDB Logic Write
innodb_rows_deleted	InnoDB Rows Deleted
innodb_rows_updated	InnoDB Rows Updated
innodb_rows_inserted	InnoDB Rows Inserted
innodb_rows_read	InnoDB Rows Read

©2013-2025 Tencent Cloud International Pte. Ltd.

log_capacity	Log Space
replicate_lag	Replica Instance Delay in Redo Log Based Replication
replicate_lsn_lag	Redo Log LSN Difference between Source and Replica Instances
replicate_status	Replication Status of Replica Instance

DBbrain (TencentDB for Redis)

In DBbrain, the custom monitoring dashboard for TencentDB for Redis currently supports the following monitoring metrics:

Monitoring Metric	Description
cmd_big_value	Big Value Request
cmd_err	Execution Error
cmd_hits	Read Request Hit
cmd_hits_ratio	Read Request Hit Rate
%cmd_key_count	Key Requests
cmd_mget	Mget Requests
cmd_miss	Read Request Miss
cmd_other	Other Requests
cmd_read	Read Request
cmd_slow	Slow Query
cmd_write	Write Request
commands	Total Requests
connections	Connections
connections_util	Connection Utilization
%cpu_max_util	Max Node CPU Utilization
%cpu_util	CPU Utilization
%evicted	Evicted Keys

©2013-2025 Tencent Cloud International Pte. Ltd.

expired	Expired Keys
in_bandwidth_util	Inbound Traffic Utilization
%in_flow	Inbound Traffic
MBit/sin_flow_limit	Inbound Traffic Throttling Trigger
keys	Total Keys
latency_max	Max Execution Latency
mslatency_other	Avg Latency of Other Commands
mslatency_avg	Avg Execution Latency
mslatency_read	Avg Read Latency
mslatency_write	Avg Write Latency
msmem_max_util	Max Node MEM Utilization
%mem_used	Memory Usage
MBmem_util	Memory Utilization
%out_bandwidth_util	Outbound Traffic Utilization
%out_flow	Outbound Traffic
MBit/sout_flow_limit	Outbound Traffic Throttling Trigger
latency_p99	P99 Execution Latency

DBbrain (self-built MySQL)

In DBbrain, the custom monitoring dashboard for self-built MySQL currently supports the following monitoring metrics:

Monitoring Metric	Description	Agent Access	Direct Access
cpu_use_rate	CPU Utilization	✓	×
memory_use_rate	Memory Utilization	✓	×
memory_use	Memory Usage	✓	×
volume_rate	Disk Utilization	✓	×
real_capacity	Used Disk Space	✓	×

©2013-2025 Tencent Cloud International Pte. Ltd.

capacity	Occupied Disk Space	✓	×
bytes_sent	Outbound Traffic	✓	✓
bytes_received	Inbound Traffic	✓	✓
qps	QPS	✓	✓
tps	TPS	✓	✓
connection_use_rate	Connection Utilization	✓	✓
max_connections	Max Connections	✓	✓
threads_connected	Connected Threads	✓	✓
slow_queries	Slow Queries	✓	✓
select_scan	Full-Table Scans	✓	✓
select_count	Queries	✓	✓
com_update	Updates	✓	✓
com_delete	Deletions	✓	✓
com_insert	Insertions	✓	✓
com_replace	Overwrites	✓	✓
queries	Total Requests	✓	✓
query_rate	Query Utilization	✓	✓
created_tmp_tables	Temp Tables	✓	✓
table_locks_waited	Table Locks Awaited	✓	✓
innodb_cache_hit_rate	InnoDB Cache Hit Rate	✓	✓
innodb_cache_use_rate	InnoDB Cache Utilization	✓	✓
innodb_os_file_reads	InnoDB Disk Reads	✓	✓
innodb_os_file_writes	InnoDB Disk Writes	✓	✓
innodb_os_fsyncs	InnoDB fsync Count	✓	✓
innodb_num_open_files	InnoDB Opened Tables	✓	✓

©2013-2025 Tencent Cloud International Pte. Ltd.

key_cache_hit_rate	MyISAM Cache Hit Rate	✓	✓
key_cache_use_rate	MyISAM Cache Utilization	✓	✓
com_commit	Submissions	✓	✓
com_rollback	Rollbacks	✓	✓
threads_created	Created Threads	✓	✓
created_tmp_disk_tables	Temp Disk Tables	✓	✓
threads_running	Running Threads	✓	✓
created_tmp_files	Temp Files	✓	✓
handler_read_rnd_next	Requests of Reading Next Row	✓	✓
handler_rollback	Internal Rollbacks	✓	✓
handler_commit	Internal Submissions	✓	✓
innodb_buffer_pool_pages_free	InnoDB Empty Pages	✓	✓
innodb_buffer_pool_pages_total	Total InnoDB Pages	✓	✓
innodb_buffer_pool_read_requests	InnoDB Logical Reads	✓	✓
innodb_buffer_pool_reads	InnoDB Physical Reads	✓	✓
innodb_data_read	InnoDB Reads	✓	✓
innodb_data_reads	Total InnoDB Reads	✓	✓
innodb_data_written	InnoDB Writes	✓	✓
innodb_data_writes	Total InnoDB Writes	✓	✓
innodb_rows_deleted	InnoDB Rows Deleted	✓	✓
innodb_rows_inserted	InnoDB Rows Inserted	✓	✓
innodb_rows_updated	InnoDB Rows Updated	✓	✓
innodb_rows_read	InnoDB Rows Read	✓	✓
innodb_row_lock_time_avg	Average InnoDB Row Lock Acquiring Time	✓	✓
innodb_row_lock_waits	InnoDB Row Lock Waits	✓	✓

©2013-2025 Tencent Cloud International Pte. Ltd.

key_blocks_unused	Unused Blocks in Key Cache	✓	✓
key_blocks_used	Used Blocks in Key Cache	✓	✓
key_read_requests	Data Blocks Read by Key Cache	✓	✓
key_reads	Data Blocks Read by Disks	✓	✓
key_write_requests	Data Blocks Written into Key Cache	✓	✓
key_writes	Data Blocks Written into Disks	✓	✓
opened_tables	Opened Tables	✓	✓
table_locks_immediate	Table Locks Released Immediately	✓	✓
open_files	Total Opened Files	✓	✓
log_capacity	Log Space	✓	×

©2013-2025 Tencent Cloud International Pte. Ltd.

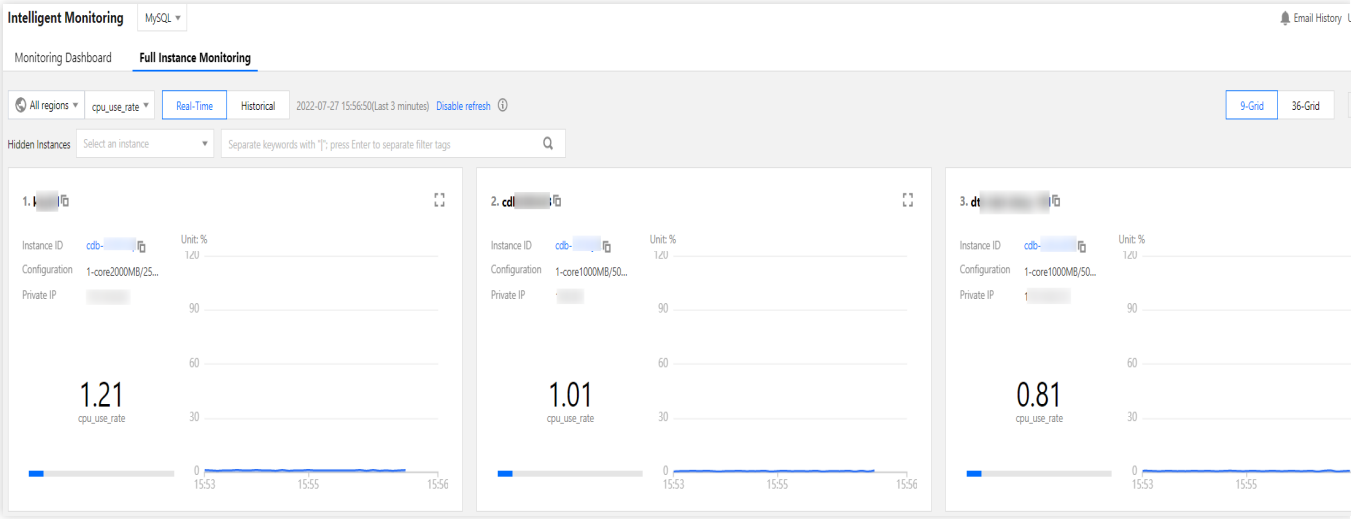
Intelligent Monitoring (Full Instance Monitoring)

Last updated：2022-09-19 22:34:38

The full instance monitoring page gives you an overview of the database monitoring metrics of all instances. The unified monitoring view displays the horizontal view of single monitoring metrics of all instances, allowing you to view and detect database exceptions and providing you with a new macro view on monitoring information.

Note:

Currently, full instance monitoring is supported only for TencentDB for MySQL (excluding basic single-node instances), TDSQL-C for MySQL, self-built MySQL, TencentDB for Redis, and TencentDB for MongoDB.

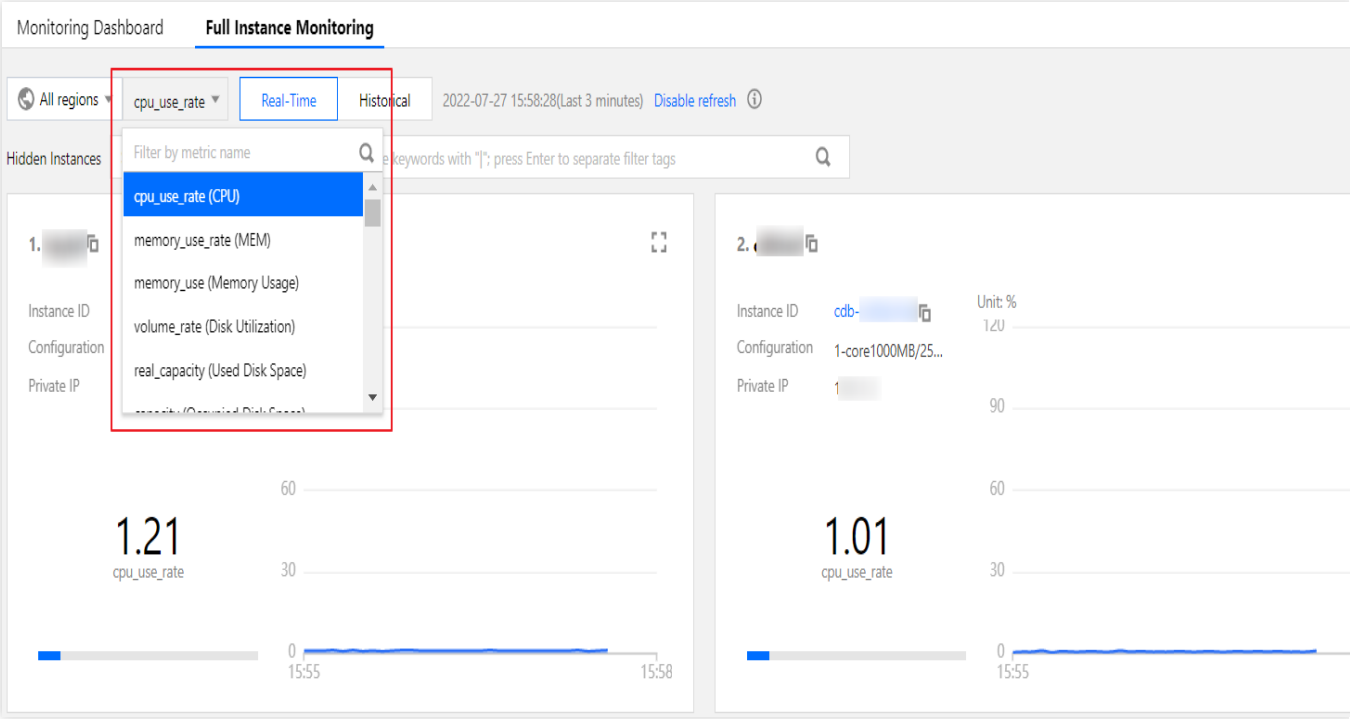


Switching the region

1. Log in to the DBbrain console and select **Monitoring and Alarming** > **Intelligent Monitoring** on the left sidebar. On the displayed page, select a database at the top and select the **Full Instance Monitoring** tab.
2. The full instance monitoring page displays the database instances in all regions by default. You can filter instances by region in the drop-down list at the top.

Switching the monitoring metric

1. Log in to the DBbrain console and select **Monitoring and Alarming** > **Intelligent Monitoring** on the left sidebar. On the displayed page, select a database at the top and select the **Full Instance Monitoring** tab.
2. You can filter and select a metric in the drop-down list at the top. All monitoring metrics of TencentDB for MySQL and TDSQL-C as well as the monitoring metrics of your self-built database are supported. The information of the selected monitoring metric is displayed and sorted by metric value on this tab.



Viewing the real-time/historical monitoring data

1. Log in to the DBbrain console and select **Monitoring and Alarming** > **Intelligent Monitoring** on the left sidebar. On the displayed page, select a database at the top and select the **Full Instance Monitoring** tab.
2. You can view real-time and historical monitoring information on the full instance monitoring page. In historical monitoring information, the maximum value and its occurrence time of the selected metric in the specified time period will be displayed.

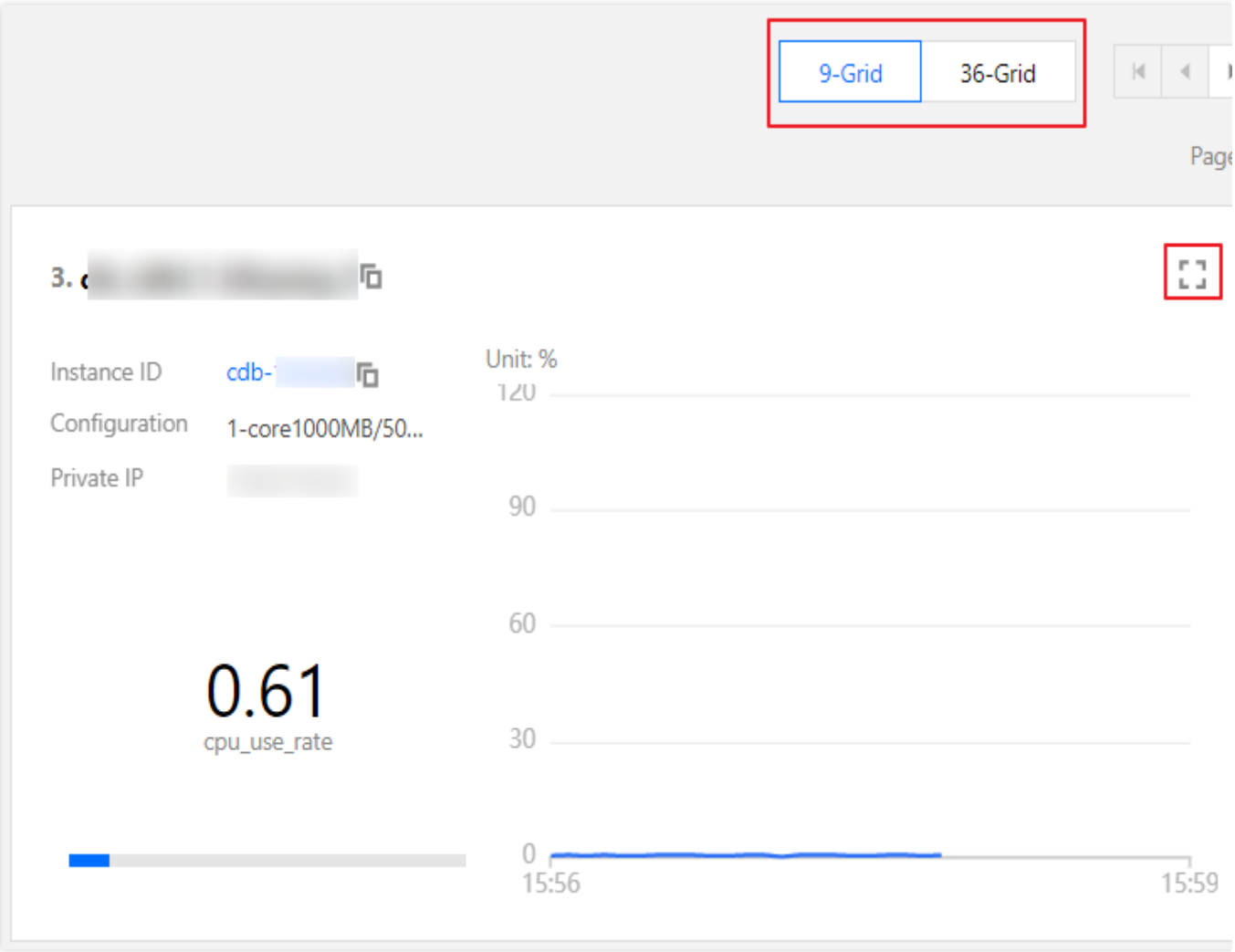
Searching for an instance

1. Log in to the DBbrain console and select **Monitoring and Alarming** > **Intelligent Monitoring** on the left sidebar. On the displayed page, select a database at the top and select the **Full Instance Monitoring** tab.
2. On this tab, you can search instances. If you select TencentDB for MySQL, fuzzy search by instance ID/name or private IP is supported; if you select TDSQL-C, fuzzy search by cluster ID/name, instance ID/name, or access point

address is supported; if you select self-built MySQL database, fuzzy search by instance ID, instance name, or IP address is supported.

Note:

Click the **i** icon on the right in the search box to view the help document for instance search.



Switching the grid view

1. Log in to the DBbrain console and select **Monitoring and Alarming** > **Intelligent Monitoring** on the left sidebar. On the displayed page, select a database at the top and select the **Full Instance Monitoring** tab.
2. You can switch between 9-grid view and 36-grid view. We recommend you use the **36-grid view** if the number of instances is high, as it provides a broader view and you can see the fluctuations of monitoring metrics more clearly. Click the **Unfold** icon in the top-right corner of the block of an instance to view its information and metric trend details.

Health Report
Health Report Management

Last updated：2021-08-13 15:15:10

The health report feature can routinely perform health checks on database instances and output the corresponding health reports for the specified time period, which helps you gain in-depth insights into the database instance health, failures, and potential risks and provides professional optimization suggestions for your reference.

Note:

Currently, health report is supported only for TencentDB for MySQL (excluding Basic single-node instances), TencentDB for Redis, TDSQL-C for MySQL and self-built MySQL databases.

Health report email push is not supported for self-built databases currently.

Creating Health Report

Log in to the DBbrain console and select **Performance Optimization** on the left sidebar. On the displayed page, select a database at the top and select the **Health Report** tab. You can view the health score trends and the problem overview for the specified time period.

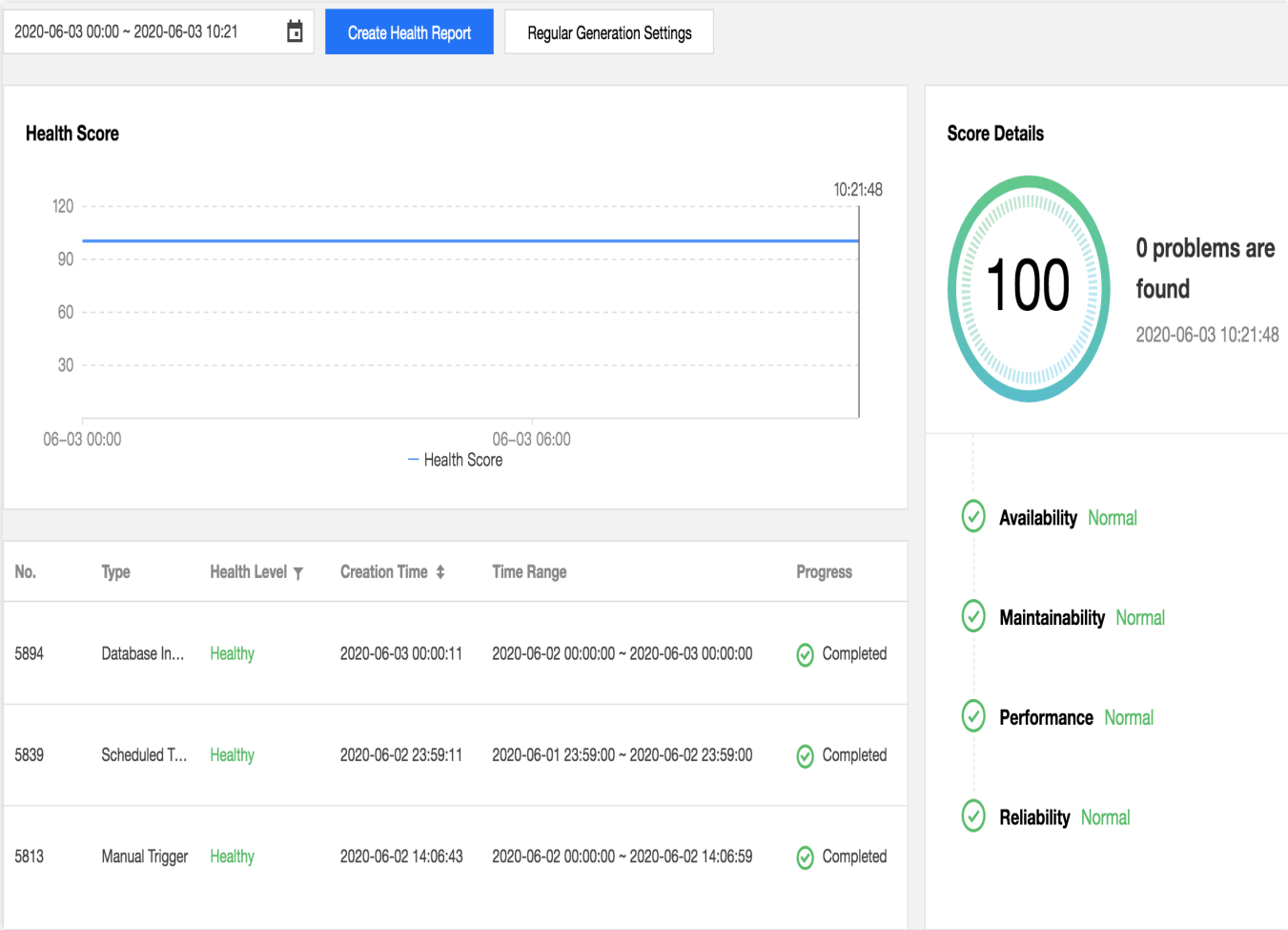
Click **Create Health Report** to create a task. After the task is completed, you can view or download the health report for the specified time period. For more information on how to send health reports to a recipient via email, please see Sending Health Report Generated by Manual Trigger via Email.

Note:

The time period of the health report is the same as that selected on the left.

Click **Regular Generation Settings** to configure the time period for automatically generating health reports. For more information on how to send health reports to a recipient via email, please see Sending Health Report Generated by Scheduled Tasks via Email.

©2013-2025 Tencent Cloud International Pte. Ltd.



Score details

In the **Score Details** section, you can view instance score details for database availability, maintainability, performance, and reliability. For more information, please see Exception Alarms.

Viewing/Downloading Health Report

In the task list, the type, health level, creation time, starting and ending time, progress, and operations of each health report task are displayed.

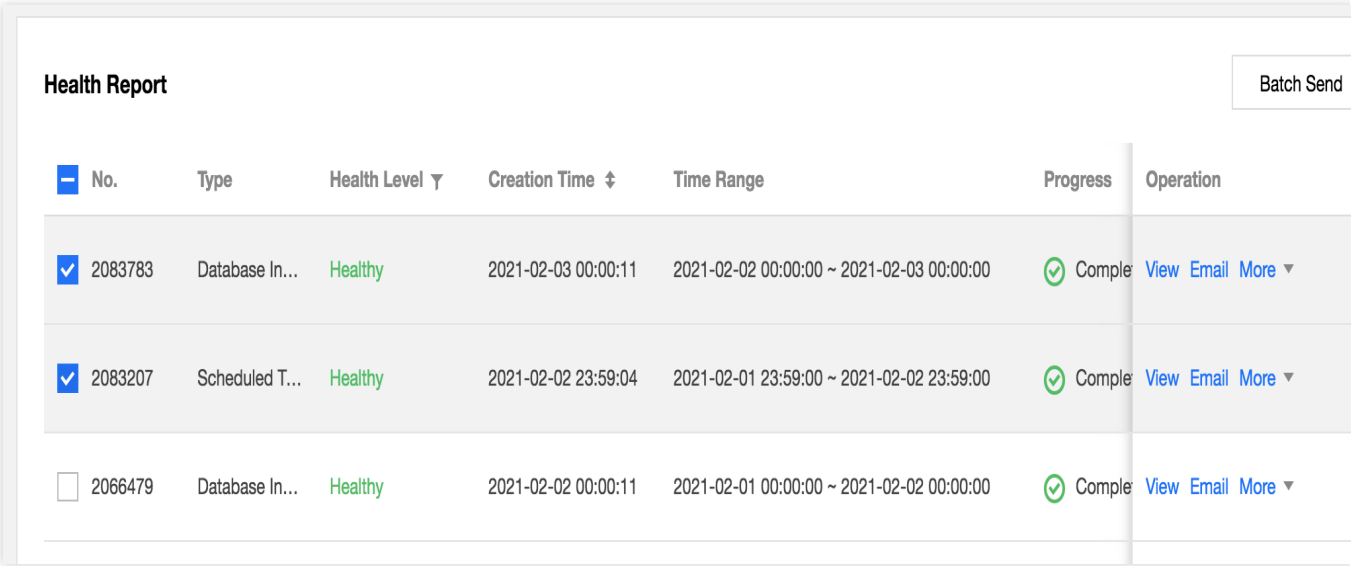
The **Type** column displays how the report is generated, including being generated manually, as scheduled, or in an database inspection.

The **Health Level** column displays the health level obtained through diagnoses, including healthy, suboptimal, risky, and critical.

You can click **View Report** in the **Operation** column to view the health report details and download the report as a PDF file.

You can click **Email** in the **Operation** column or click **Batch Send** after selecting multiple health report records to send the health reports to the mailbox of the specified contact. For more information, please see Sending Historical Health Report via Email.

You can click **More** > **Deduction Details** in the **Operation** column to view the reason for the deduction of health report task scores.
You can select **More** > **Delete** in the **Operation** column to delete the health report task.

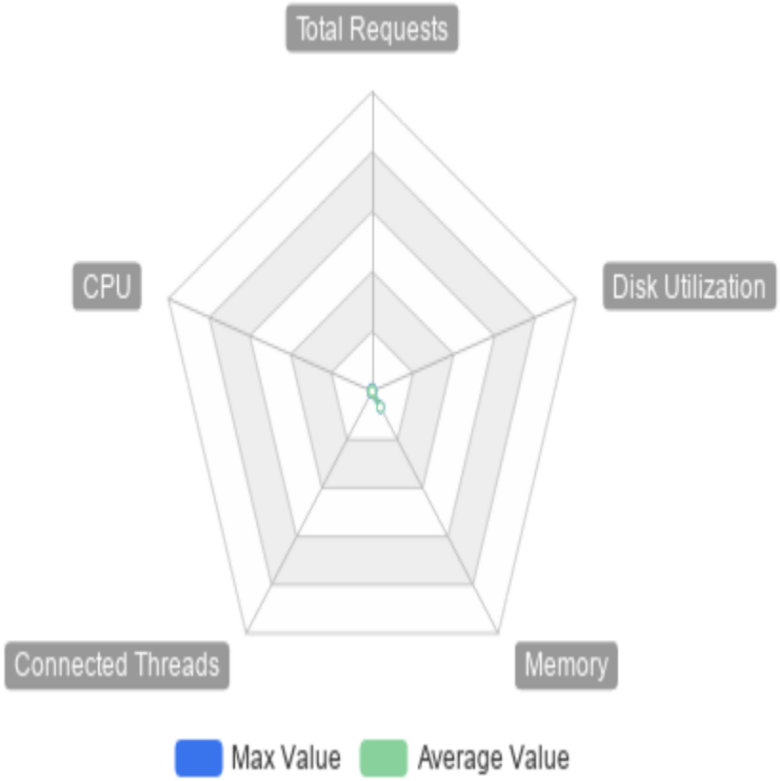


Reading Health Report

A health report displays DBbrain's evaluation of the overall operation conditions of the selected database instance in the specified time period. Items in the report includes the database's existing problems, an analysis of existing problems, and corresponding suggestions, helping you gain a comprehensive understanding of the overall operation status of the selected instance and coordinate relevant personnel to troubleshoot issues.
A report mainly contains the following sections: overview, basic information, health, instance status, exception diagnosis, slow SQL analysis, big table analysis, and performance curve.

©2013-2025 Tencent Cloud International Pte. Ltd.

4. Instance Status



Resource Name	Status	Max Value	Average Value	Reference Value
Total Requests	Idle	8times/sec	3.78times/sec	800 ~ 1000
CPU	Idle	0.2%	0.08%	20 ~ 60
Connected Threads	Idle	6	4.21	400 ~ 1000
Memory	Idle	6.59%	6.58%	60 ~ 80
Disk Utilization	Idle	0.01%	0.01%	20 ~ 80

Appendix

Reported exception level definitions

No.	Type	Description
1	Fatal	The value is 1
2	Severe	The value is 2
3	Warning	The value is 3
4	Notice	The value is 4
5	Healthy	The value is 5

Reported health level definitions

No.	Type	Description
1	Healthy	Score ≥ 95
2	Suboptimal	80 ≤ score < 95
3	Risky	60 ≤ score < 80
4	Critical	Score < 60

©2013-2025 Tencent Cloud International Pte. Ltd.

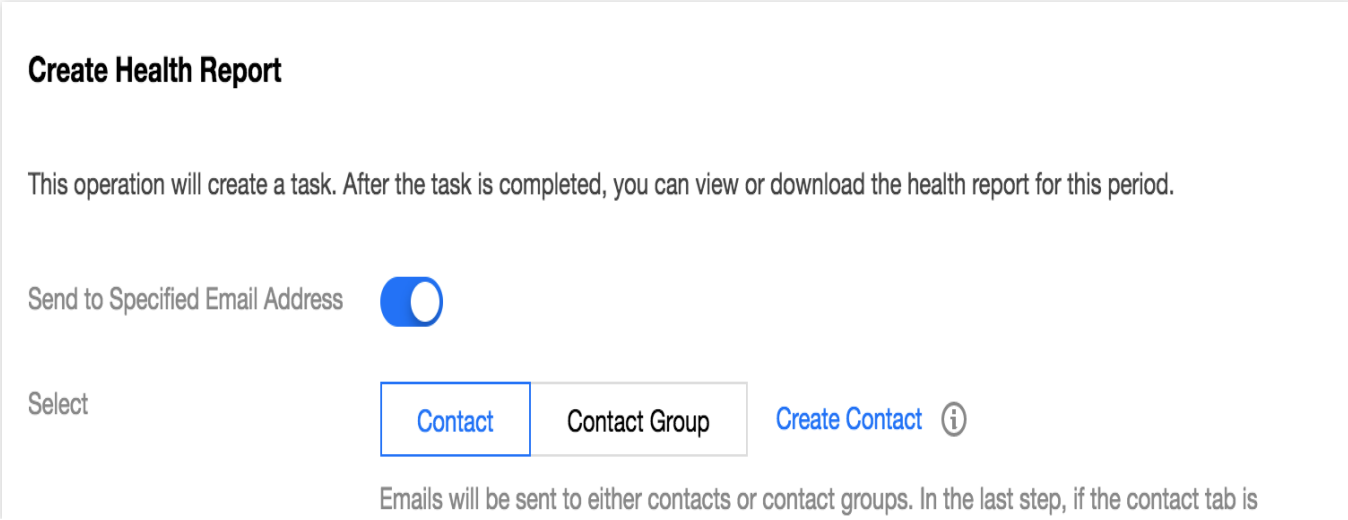
Health Report Email Push

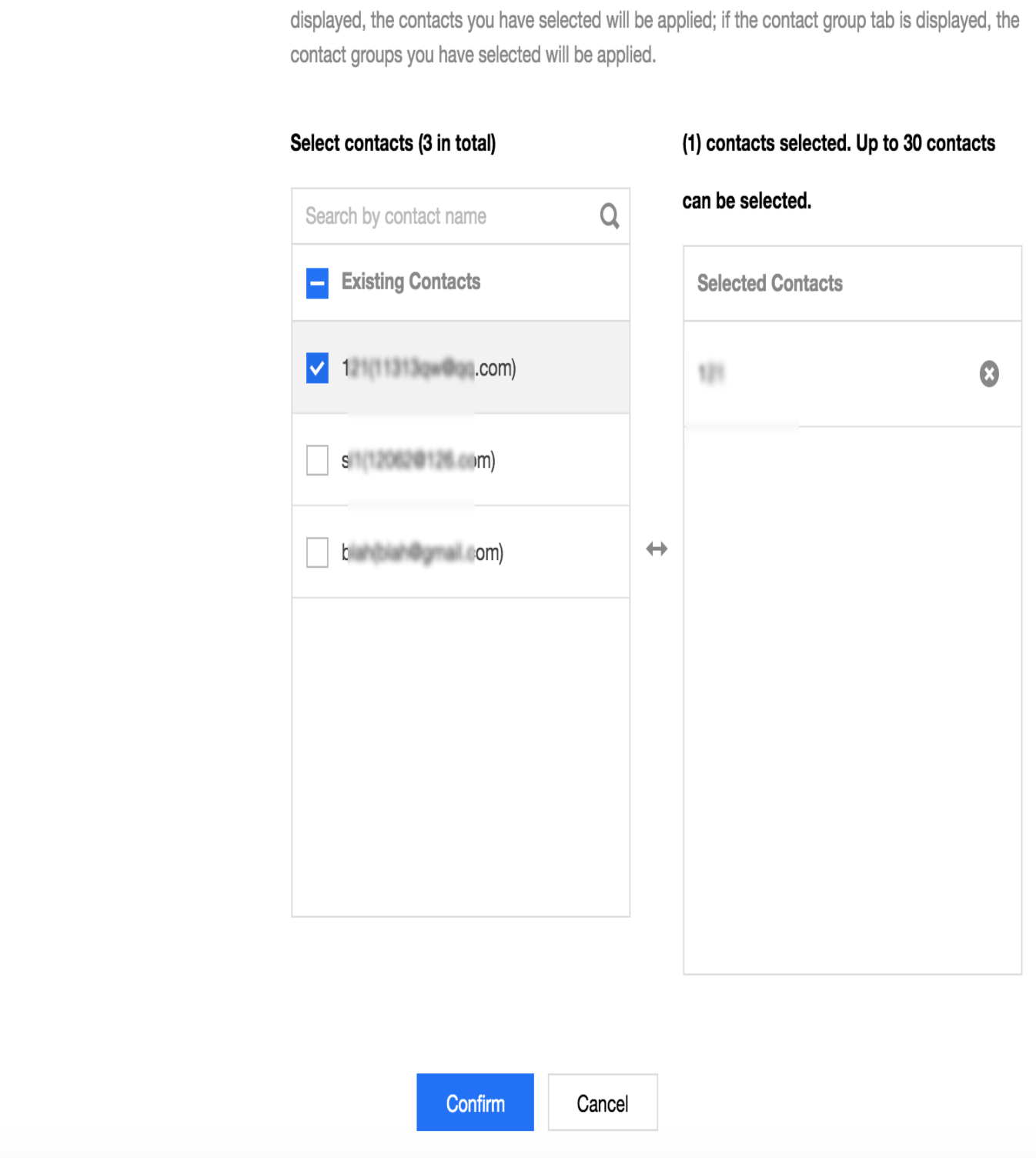
Last updated：2021-04-02 16:46:33

DBbrain supports the feature of health report email push. Users can easily know about the health status of the database instance without logging in to the console.
To help more relevant business personnel know about the health status of the database instance in time, users can also customize the health reports and the recipients to send to as needed.
Currently, the health reports are generated by three methods: manual trigger, scheduled tasks and database inspection. All of these reports can be sent via email. Users can send the reports to the specified recipient's email once the reports are created, or select the historical reports to send to the specified recipient's email.
Note:
Currently, health report email push is supported only for TencentDB for MySQL (excluding the basic single-node instance).

Sending Health Report Generated by Manual Trigger via Email

1. Log in to the DBbrain console, select **Performance Optimization** on the left sidebar. On the displayed page, select a database type at the top, select the **Health Report** tab, and click **Create Health Report**.
2. In the pop-up window, enable **Send to Specified Email Address**, select **Contact** or **Contact Group**, and click **Confirm**. The generated health reports will be sent to the email of the specified contact or contact group.
Note:
You can only select either contact or contact group.
You can send the health report to up to 30 contacts at a time.
The health report will be sent based on the selected email address. To avoid email block, please add the email allowlist policy in advance: dbbrain@qcloudmail.com.





Sending Health Report Generated by Scheduled Tasks via Email

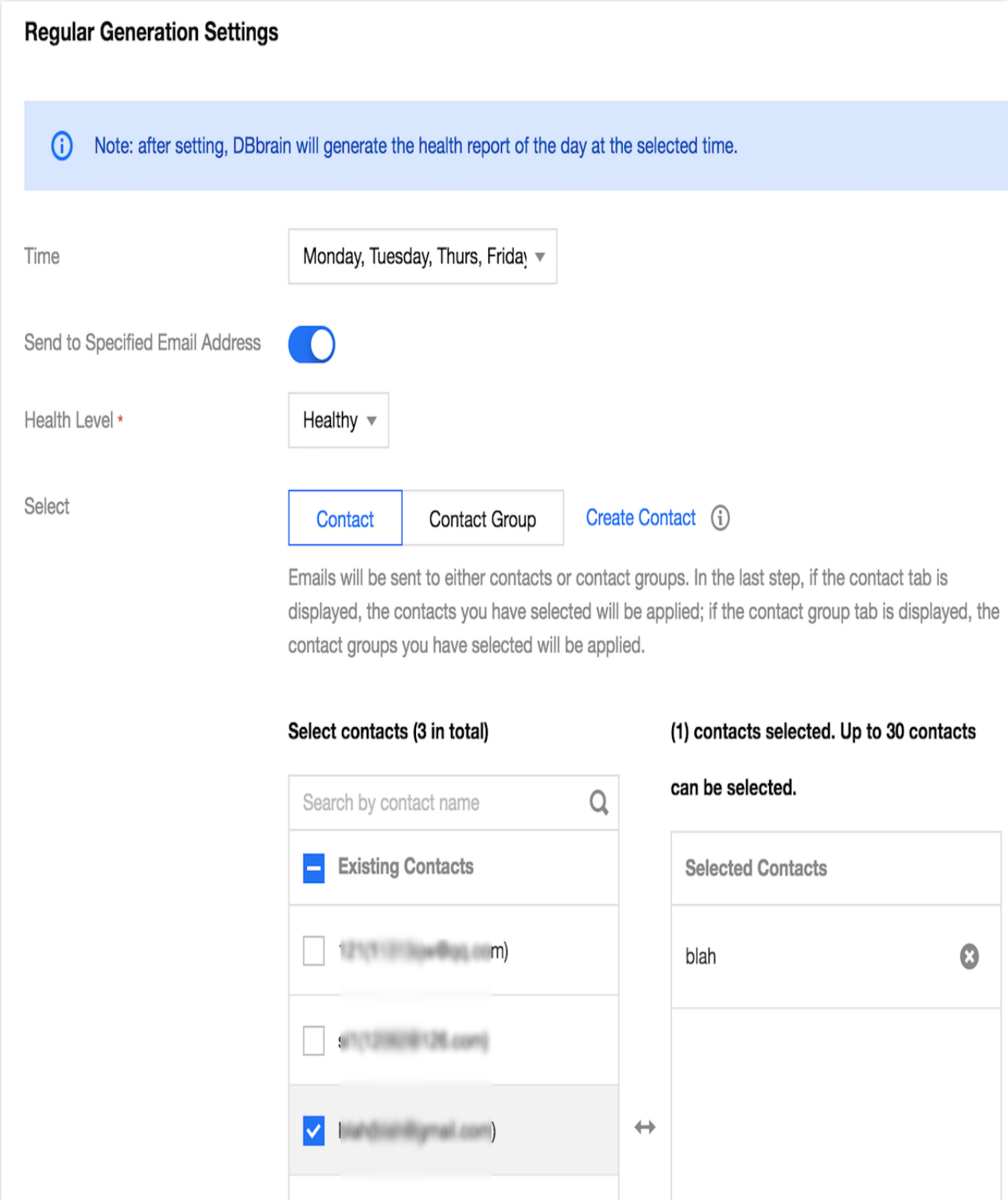
1. Log in to the DBbrain console, and select **Performance Optimization** on the left sidebar. On the displayed page, select a database type at the top, select the **Health Report** tab, and click **Regular Generation Settings**.
2. In the pop-up window, configure the time to generate the health report, enable **Send to Specified Email Address**, select the **Health Level**, select **Contact** or **Contact Group**, and click **Confirm**. The generated health reports will be regularly sent to the email of the specified contact or contact group.

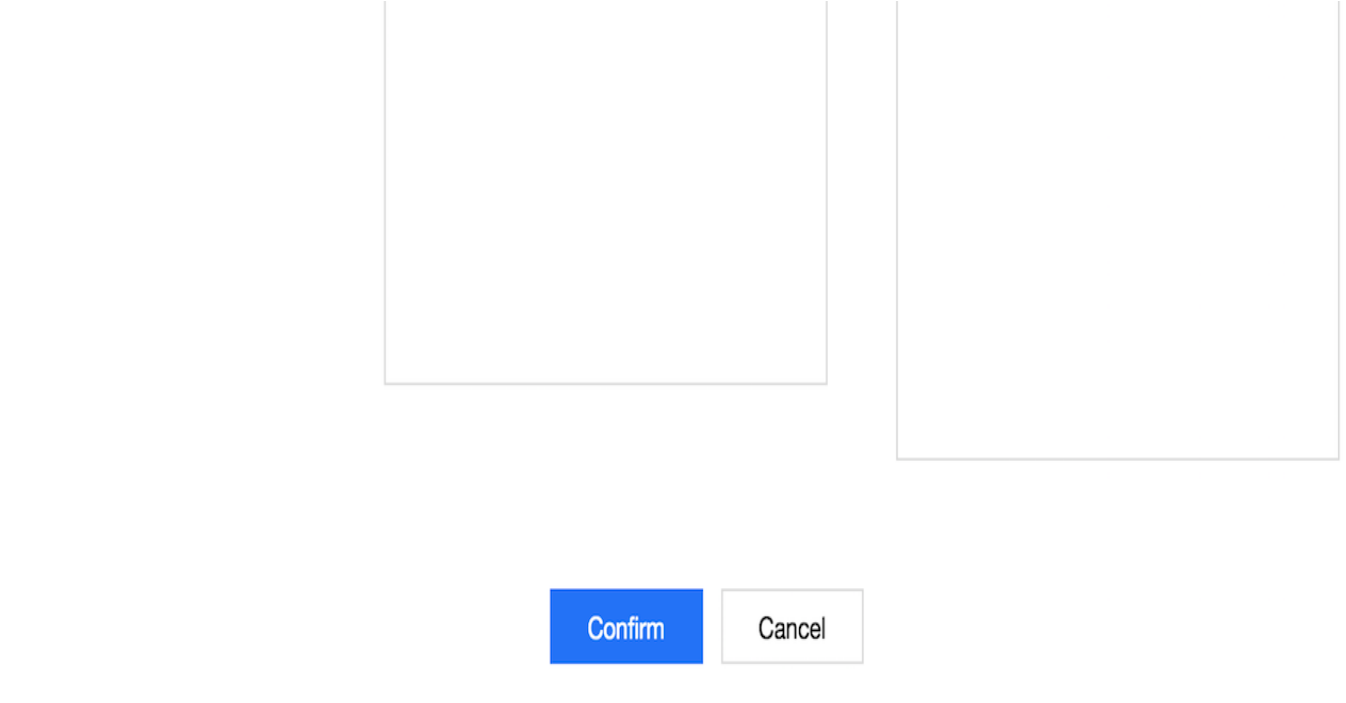
Note:

You can only select either contact or contact group.

In **Regular Generation Settings**, you can send the health report to up to 30 contacts at a time.

The health report will be sent based on the selected email address. To avoid email block, please add the email allowlist policy in advance: dbbrain@qcloudmail.com.





Sending Health Report Generated by Database Inspection via Email

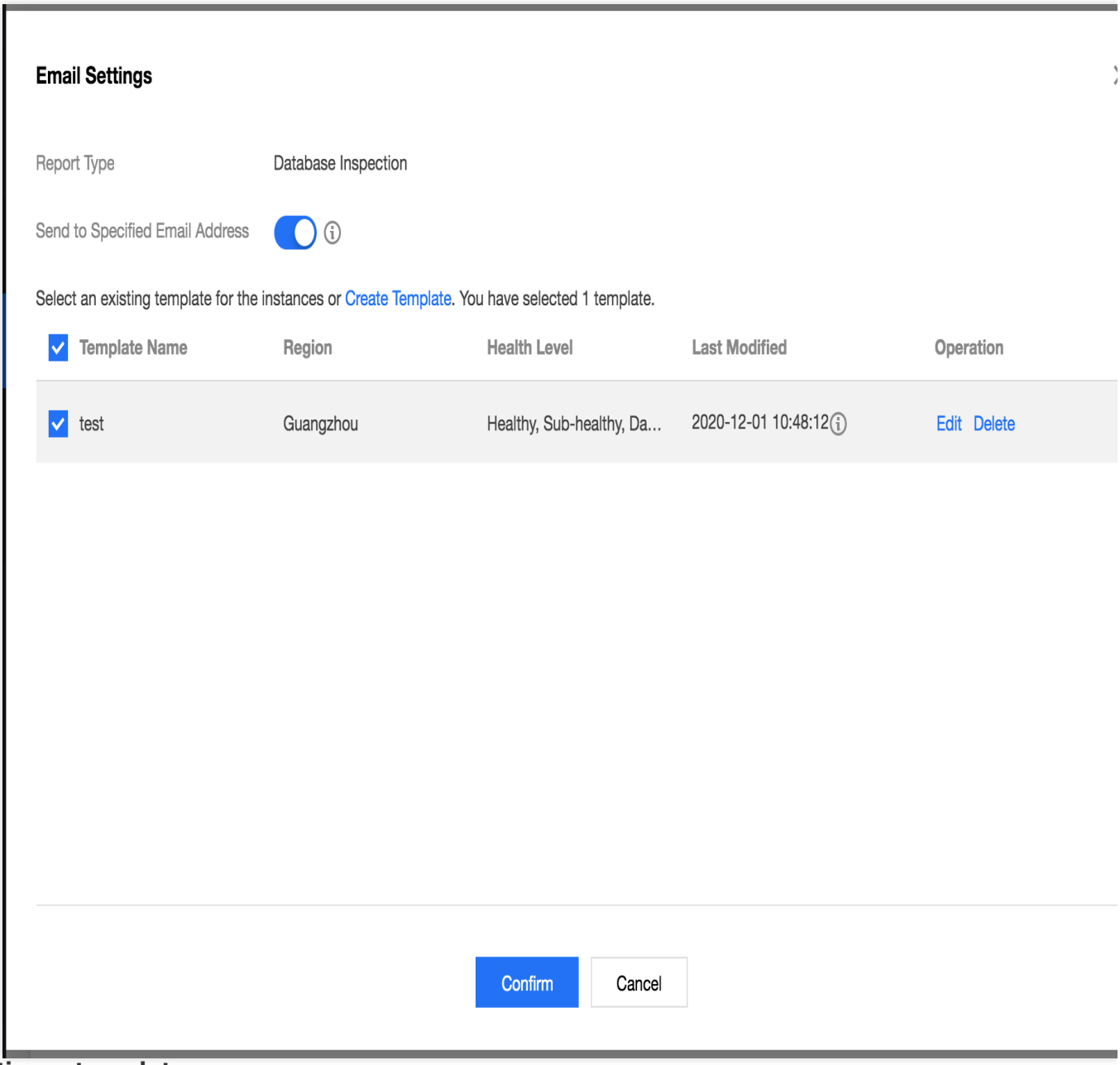
1. Log in to the DBbrain console, select **Monitoring & Alarm** > **Database Inspection** on the left sidebar, and click **Email Settings** on the top-right corner.
2. In the pop-up window, enable **Send to Specified Email Address**, select an existing email template or create a template, and click **Confirm**. The health report generated by the database inspection will be sent to the email of the contact or contact group specified in the selected template.

Note:

You can select up to 5 database inspection email templates at a time.

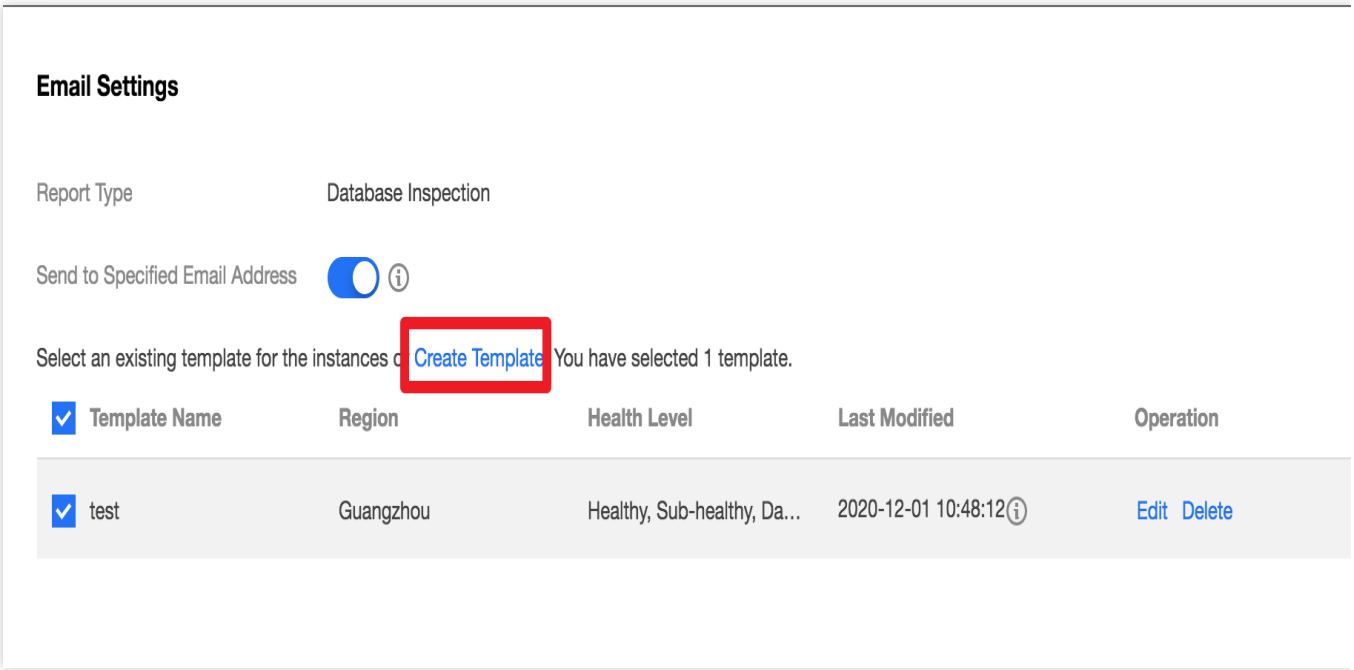
The **Last Modified** column displays the information of the last editor of the template, and the health report of the instance will be sent based on the instance permissions of the last editor.

The health report will be sent based on the selected email address. To avoid email block, please add the email allowlist policy in advance: dbbrain@qcloudmail.com.



Creating a template

1. In the pop-up window of **Email Settings** in Database Inspection, click **Create Template**.



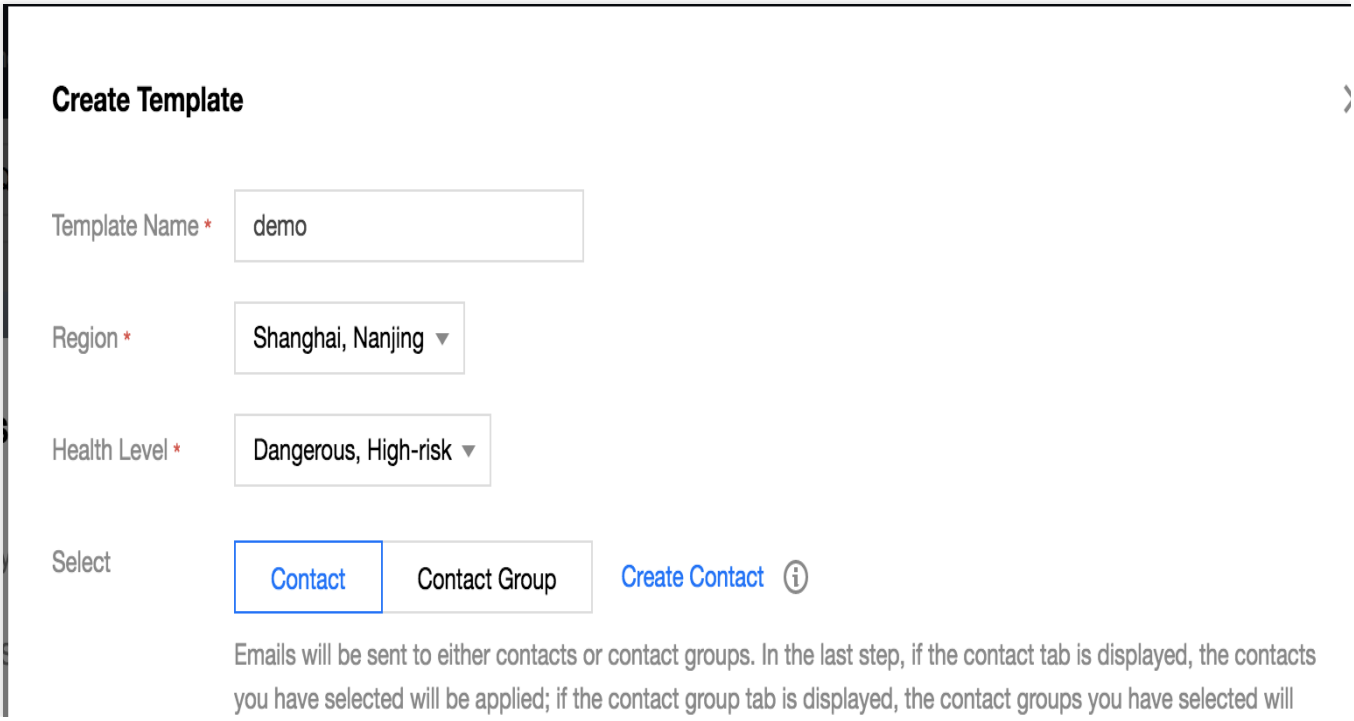
2. In the pop-up window, enter the template name, region, and health level, select contact or contact group, and click **Confirm**.

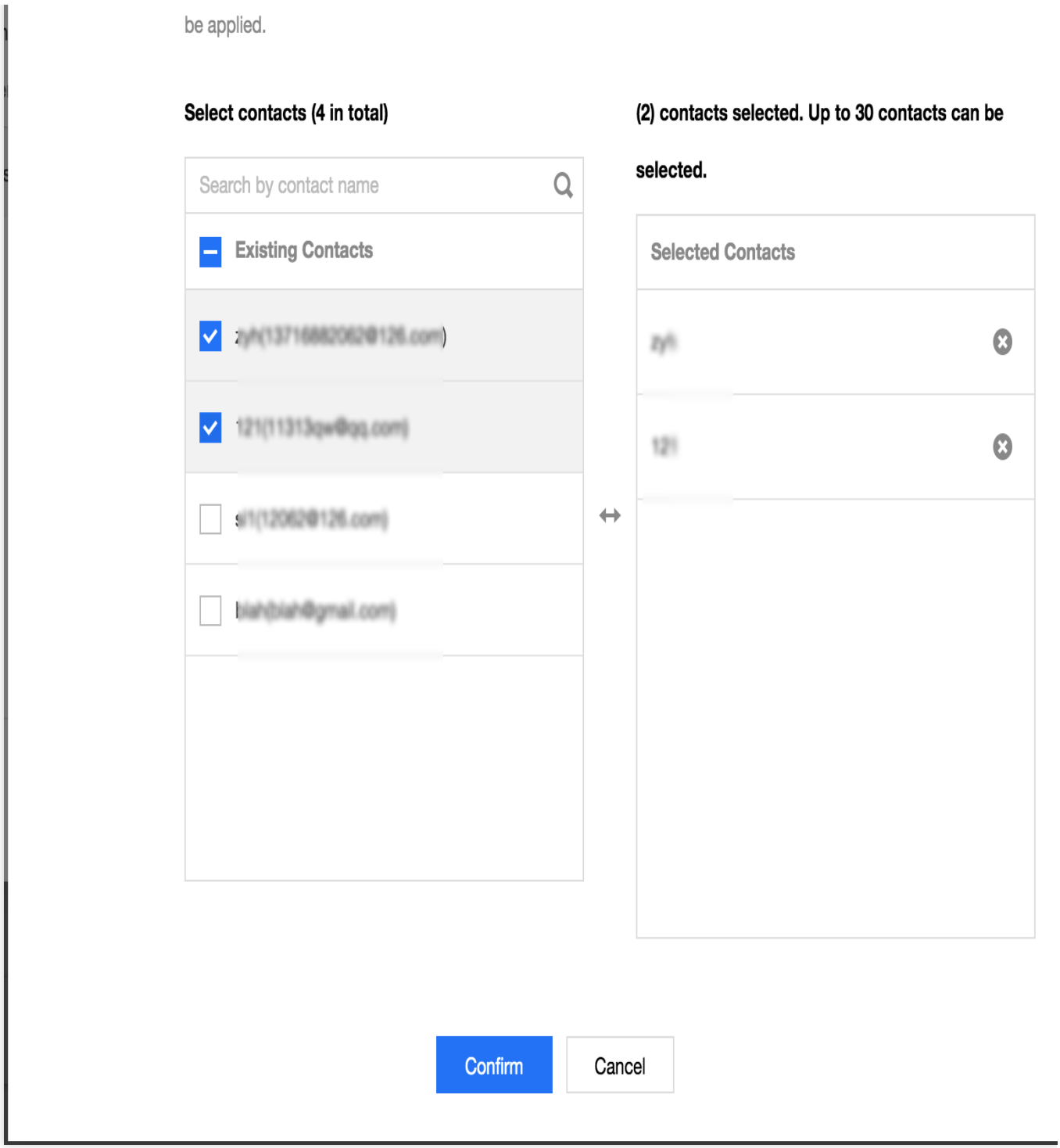
Note:

After the region and health level are set, the generated health report will be sent according to the selected region and health level.

The health report is sent on the premise that the instance in the region has enabled the database inspection.

The health report will be sent based on the selected email address. To avoid email block, please add the email allowlist policy in advance: dbbrain@qcloudmail.com.

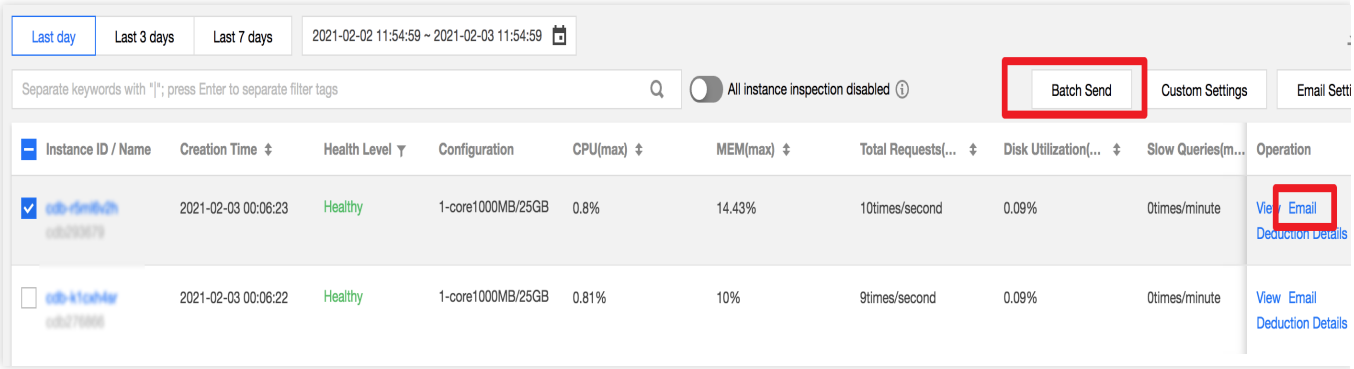




Sending Historical Health Report via Email

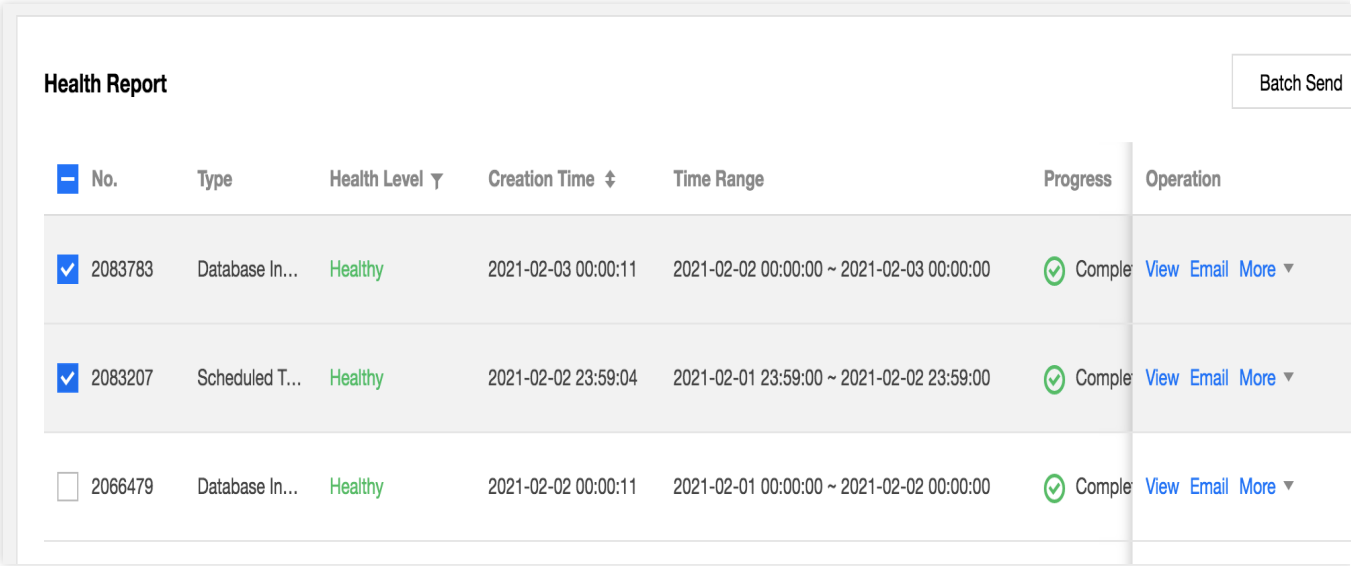
Sending email in database inspection page

1. Log in to the DBbrain console, select **Monitoring & Alarm** > **Database Inspection** on the left sidebar.
2. In the database inspection list, click **Email** in the **Operation** column of an inspection record, or select multiple inspection records and click **Batch Send**.
3. In the pop-up window, select the contact or contact group, and click **Confirm**. The generated health report will be sent to the email of the selected contact or contact group.



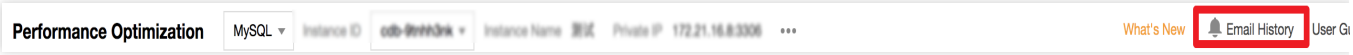
Sending email in health report page

1. Log in to the DBbrain console, select **Performance Optimization** on the left sidebar. On the displayed page, select a database type at the top, select the **Health Report** tab.
2. In the health report list, click **Email** in the **Operation** column of a report, or select multiple health reports and click **Batch Send**.
3. In the pop-up window, select the contact or contact group, and click **Confirm**. The selected health report will be sent to the email of the specified contact or contact group.

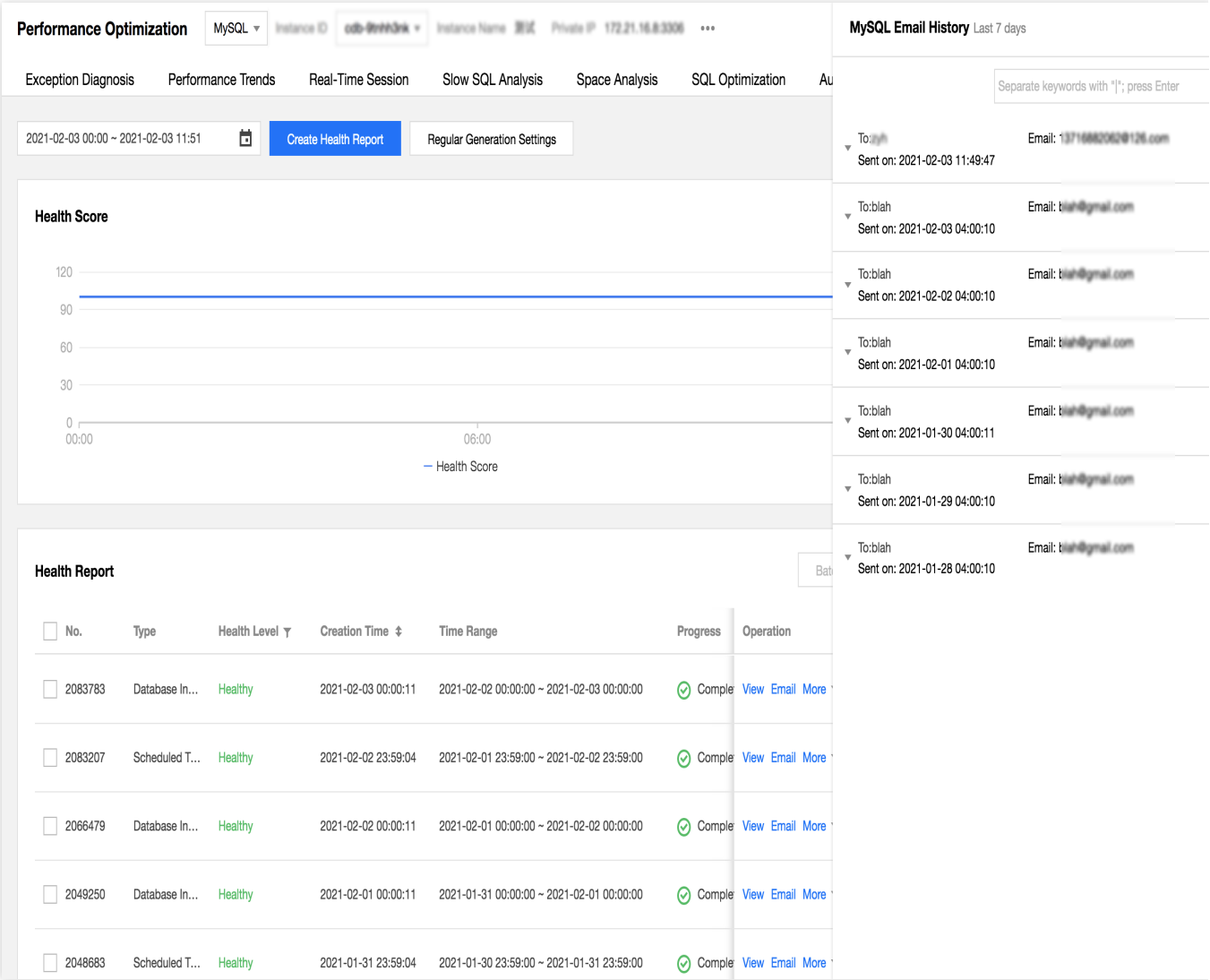


Email History

1. Log in to the DBbrain console, click **Email History** in the top-right corner to view email histories of the sent health reports.



2. In the email history, you can view the recipient, instance basic information, and email sending status.
The recipient information includes recipient, email, and sending time. When the recipient is a contact group, click **Number of Contacts** to view the details of the contacts in the group.
The instance basic information includes instance ID/name and report time range.
The email sending status includes all succeeded, partially succeeded, and all failed. When the status is partially succeeded or all failed, please check whether the email address is correct.



Email Content

After the email of the health report is sent successfully, user will receive an email which involves the instance ID, instance name, health level, type, time range, operation, etc. Click **View** in the **Operation** column, and you can directly download the PDF file of the health report for this instance in the email.

Note:

The health report is valid for 3 days. Please download it before it expires.

Contact Management

Last updated：2022-07-31 17:26:10

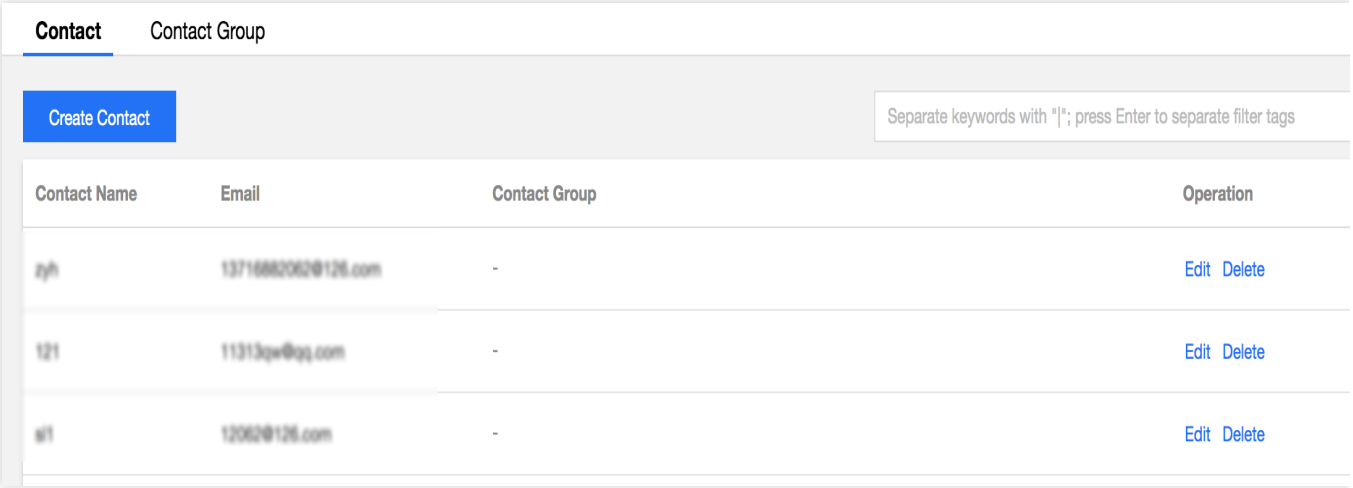
Contact management is used to centrally manage and set the recipients and recipient groups of the health report email push, and supports the management of contacts and contact groups.
Log in to the DBbrain console, click **Performance Optimization** on the left sidebar, select **Health Report** tab, and click **Contact Management**.

Contact

The Contact tab is used to manage and set the email recipients. Click **Create Contact**, enter the contact name, email, select the contact group (optional), and click **Confirm**.
In the contact list, it displays the contact basic information, including: contact name, email address, contact group and operation. You can edit and delete the created contacts, and also can query contacts based on the contact name, email address, and contact group.

Note:

Once a contact is deleted, the contact will no longer receive associated health reports.



Contact group

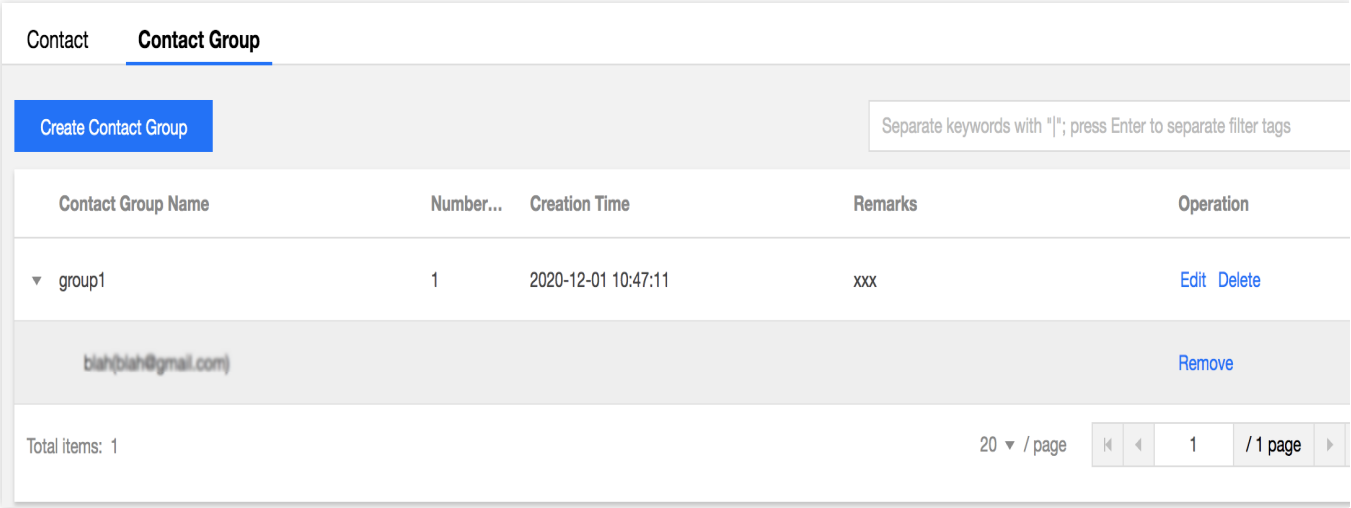
The Contact Group tab is used to manage and set the email recipient groups. Click **Create Contact Group**, enter the contact group name and remarks (optional), and click **Confirm**.

In the contact group list, it displays the basic information of the contact group, including the contact group name, the number of contacts, creation time, remarks and operation. You can edit and delete the created contact group.

Click the icon in front of the group name to view the details of contacts in this group. You can remove the contact from the group and query a contact group by the contact group name.

Note:

A contact group can contain up to 10 contacts.



MySQL/TDSQL-C for MySQL Performance Optimization
Exception Diagnosis

Last updated：2022-08-16 15:55:36

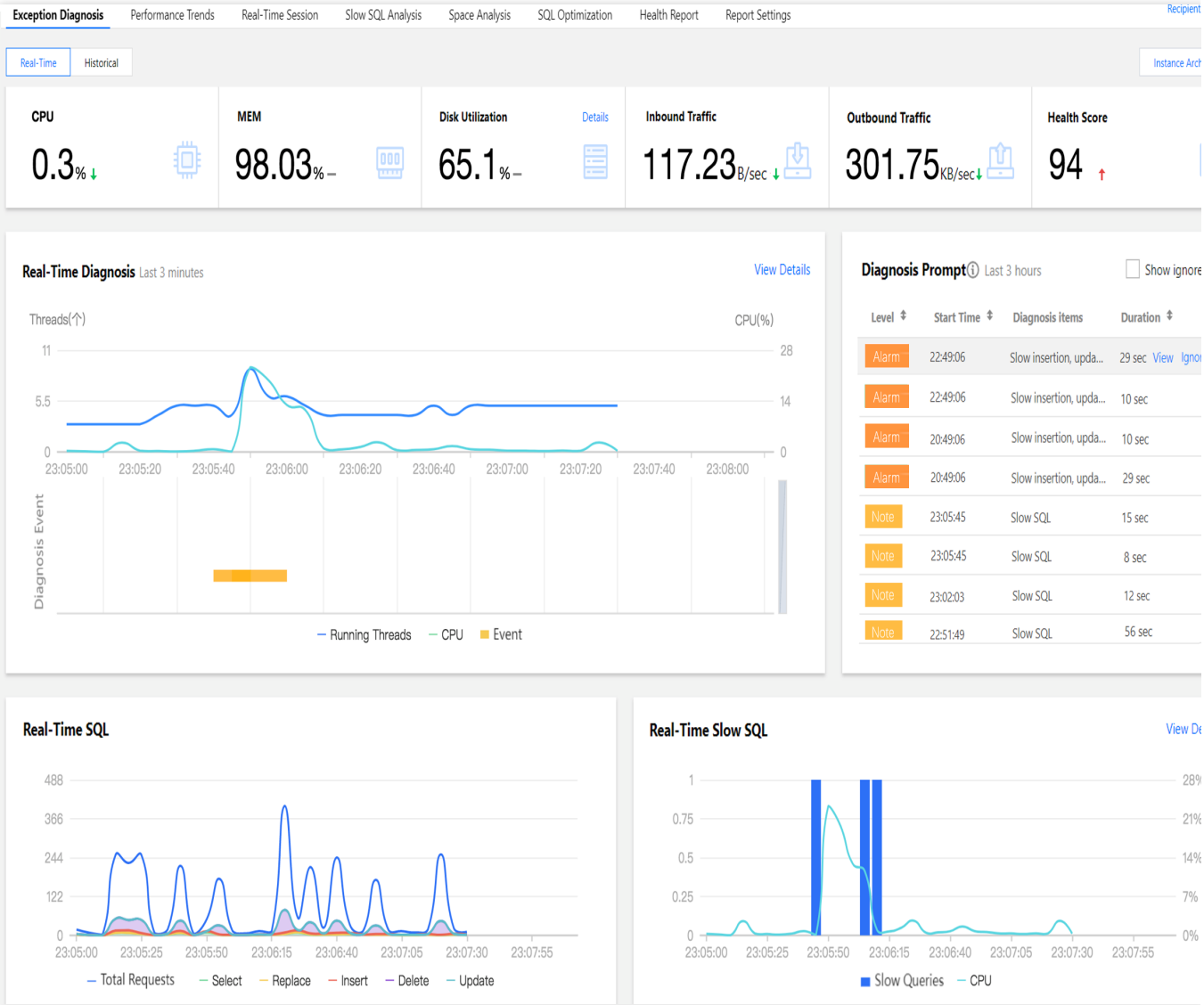
Feature Description

The exception diagnosis feature provides you with real-time performance monitoring, health inspections, and failure diagnosis and optimization, so that you can intuitively know the real-time operation status of database instances, locate newly appeared performance exceptions in real time, and optimize the system based on the optimization suggestions. Exception diagnosis provides real-time and historical view modes.

Overview

Log in to the DBbrain console and select **Performance Optimization** on the left sidebar. On the displayed page, select a database type and an instance at the top, and select the **Exception Diagnosis** tab.

©2013-2025 Tencent Cloud International Pte. Ltd.



Viewing Monitoring Information

The **Exception Diagnosis** tab displays **CPU Utilization**, **Memory Utilization**, **Inbound Traffic**, **Outbound Traffic**, and **Health Score**. To view details on disk utilization, click **Details** in the top-right corner. AI-based health scores can reflect the actual status of your databases.

Viewing Diagnosis Information

The **Real-Time Diagnosis** or **Diagnosis Records** section displays the current instance's real-time or historical information about the number of running threads, CPU utilization, and diagnosis events.

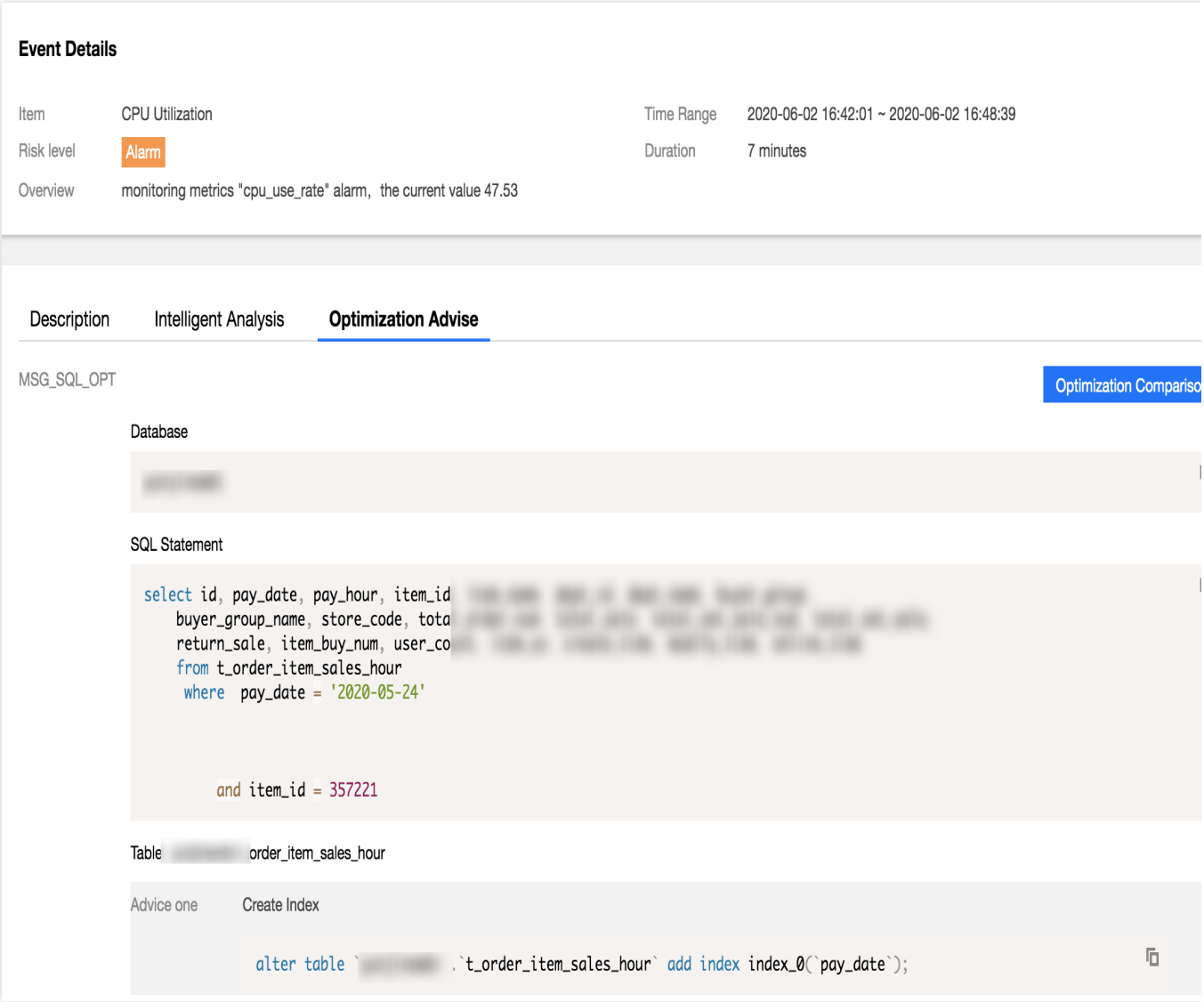
The **Diagnosis Prompt** section displays the overview information of diagnosis event history, including **Level** (**Healthy**, **Note**, **Alarm**, **Serious**, or **Critical**), **Start Time**, **Diagnosis items**, and **Duration**. DBbrain performs health inspections on the instance once every ten minutes.

Viewing diagnosis details

1. Log in to the DBbrain console and select **Performance Optimization** on the left sidebar. On the displayed page, select a database type and an instance at the top, and select the **Exception Diagnosis** tab.
2. In the **Real-Time Diagnosis** or **Diagnosis Records** section, select a time range and zoom in the view by using the mouse wheel. You can click **View Details** or click an item in the **Diagnosis Prompt** section to enter the **Diagnosis Details** page.
3. Click a diagnosis event in the view to display the event details.

Event Details: Include the **Diagnosis items**, **Time Range**, **Risk Level**, **Duration**, and **Overview**.
Description: Includes problem snapshots and performance trends of the exception or health inspection event.
Intelligent Analysis: Analyzes the root cause of the performance exception to help you locate the specific operation.
Optimization Suggestion: Provides optimization suggestions, including but not limited to SQL optimization (index and rewrite), resource configuration optimization, and parameter fine-tuning.

©2013-2025 Tencent Cloud International Pte. Ltd.



Click **Optimization Comparison** on the **Optimization Suggestion** tab. In the pop-up window, you can view the SQL statement's execution plan, index advice, table structure, and performance before and after SQL optimization. The performance of an optimized SQL statement is estimated based on the analysis of the statistics of database tables related to the statement, the OPTIMIZER_SWITCH configuration, and the index selectivity. A chart is used to visually show the decrease in the performance. You can also compare the execution plans before and after SQL optimization to further verify the optimization results.

Ignoring/Unignoring an alarm

You can click **Ignore** to ignore an alarm. Then, other diagnosis item alarms of the instance generated by the same root cause will also be ignored. Ignored alarms will be grayed out.

Note:

Only diagnosis item alarms that are not generated by health inspections can be ignored or unignored.

You can click **Unignore** to unignore an ignored alarm. Then, other diagnosis item alarms of the instance generated by the same root cause will also be unignored. Ignored diagnosis items are not displayed by default.

1. Log in to the DBbrain console and select **Performance Optimization** on the left sidebar. On the displayed page, select a database type and an instance at the top, and select the **Exception Diagnosis** tab.
2. In the **Diagnosis Prompt** section, hover over an alarm to display the **Ignore** button and click it. You can click **Ignore** or **Unignore** on the row of an alarm to ignore or unignore it and other alarms generated by the same root cause.
Or, go to the **Event Details** page and click **Ignore** or **Unignore** in the top-right corner.

Viewing SQL and Slow SQL Information

The **Real-Time SQL** or **Historical SQL** section displays the overall information and distribution of requests made to the instance, including the trends of total requests as well as SELECT, REPLACE, INSERT, DELETE, and UPDATE requests.
The **Real-Time Slow SQL** or **Historical Slow SQL** section displays the trends of slow SQL statements (slow logs) and CPU utilization. You can click **View Details** in the top-right corner to enter the **Slow SQL Analysis** page and view analysis details.
Note:
For self-built database instances accessed directly, as server resource monitoring metrics cannot be collected, some features may not be displayed. We recommend you access such instances through the Agent.

©2013-2025 Tencent Cloud International Pte. Ltd.

Performance Trends

Last updated：2022-08-16 16:08:18

Feature description

DBbrain's performance trends feature not only supports the selection of multiple performance metrics such as key metrics, all metrics, and custom metrics, but also supports multiple ways to view performance trends, such as fine-grained view of one single performance metric trend, as well as link comparison view and time comparison view of multiple performance metric trends.

Supported performance metrics

TencentDB for MySQL

Category	Subcategory	Metric
Resource Monitoring	CPU	CPU
	Memory	Memory
		Memory Usage
	Storage Space	Disk Utilization
		Occupied Disk Space
	Traffic	Outbound Traffic
		Inbound Traffic
MySQL Server	TPS/QPS	TPS/QPS
	Connection	Max Connections
		Connected Threads
		Running Threads
		Created Threads
	Requests	Select
		Update

©2013-2025 Tencent Cloud International Pte. Ltd.

		Delete
		Insert
		Replace
		Total Requests
	Slow Query	Slow Queries
		Full-Table Scans
InnoDB Engine	InnoDB Buffer Pool Pages	InnoDB Empty Pages
		Total InnoDB Pages
		InnoDB Logical Reads
		InnoDB Physical Reads
	Read/Written InnoDB Data	InnoDB Reads
		InnoDB Writes
	InnoDB Data Reads/Writes	Total InnoDB Reads
		Total InnoDB Writes
	InnoDB Row Operations	InnoDB Rows Deleted
		InnoDB Rows Inserted
		InnoDB Rows Updated
		InnoDB Rows Read
	InnoDB Row Lock	InnoDB Row Lock Waits
		Average InnoDB Row Lock Acquiring Time
MySQL Replication	Replication Status	Source-Replica Delay Distance
		Source-Replica Delay Time
	Replication Delay	IO Thread Status
		SQL Thread Status

Self-built MySQL

Monitoring Metric			Agent Access	Direct Access
Resource Monitoring	CPU	CPU	✓	×
	Memory	Memory	✓	×
		Memory Usage	✓	×
	Storage Space	Storage Utilization	✓	×
		Used Storage Space	✓	×
	Traffic	Outbound Traffic	✓	✓
		Inbound Traffic	✓	✓
MySQL Server	TPS/QPS	TPS/QPS	✓	✓
	Connection	Max Connections	✓	✓
		Connected Threads	✓	✓
		Running Threads	✓	✓
		Created Threads	✓	✓
	Requests	Select	✓	✓
		Update	✓	✓
		Delete	✓	✓
		Insert	✓	✓
		Replace	✓	✓
		Total Requests	✓	✓
	Slow Query	Slow Queries	✓	✓
		Full-Table Scans	✓	✓
InnoDB Engine	InnoDB Buffer Pool Pages	InnoDB Empty Pages	✓	✓
		Total InnoDB Pages	✓	✓
		InnoDB Logical Reads	✓	✓
		InnoDB Physical Reads	✓	✓

©2013-2025 Tencent Cloud International Pte. Ltd.

	Read/Written InnoDB Data	InnoDB Reads	✓	✓
		InnoDB Writes	✓	✓
	InnoDB Data Reads/Writes	Total InnoDB Reads	✓	✓
		Total InnoDB Writes	✓	✓
	InnoDB Row Operations	InnoDB Rows Deleted	✓	✓
		InnoDB Rows Inserted	✓	✓
		InnoDB Rows Updated	✓	✓
		InnoDB Rows Read	✓	✓
	InnoDB Row Lock	InnoDB Row Lock Waits	✓	✓
		Average InnoDB Row Lock Acquiring Time	✓	✓

TDSQL-C for MySQL

Category	Subcategory	Metric
Resource Monitoring	CPU	CPU
	Memory	Memory
		Memory Usage
	Storage Space	Storage Utilization
		Used Storage Space
	Traffic	Outbound Traffic
		Inbound Traffic
MySQL Server	TPS/QPS	TPS/QPS
	Connection	Max Connections
		Connected Threads
		Running Threads
		Created Threads

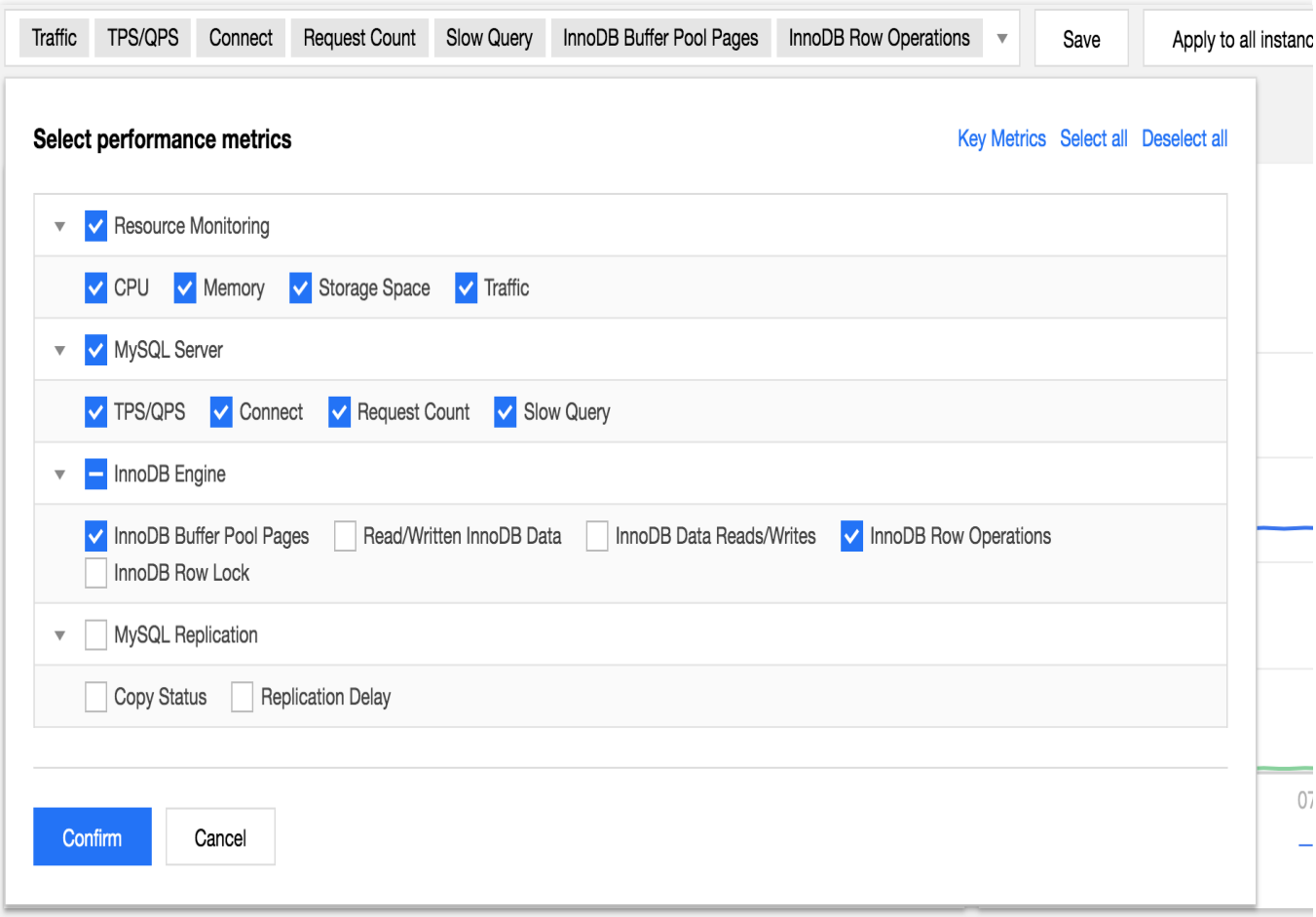
	Requests	Select
		Update
		Delete
		Insert
		Replace
		Total Requests
	Slow Query	Slow Queries
		Full-Table Scans
InnoDB Engine	InnoDB Row Operations	InnoDB Rows Deleted
		InnoDB Rows Inserted
		InnoDB Rows Updated
		InnoDB Rows Read
	InnoDB Buffer Pool Pages	InnoDB Logical Reads
		InnoDB Logical Writes
MySQL Replication	Replication Status	Replication Status of Replica Instance
	Replication Delay	Redo Log LSN Difference between Source and Replica Instances
		Replica Instance Delay in Redo Log Based Replication

Viewing performance trend metrics

1. Log in to the DBbrain console and select **Performance Optimization** on the left sidebar. On the displayed page, select a database type and an instance at the top, and select the **Performance Trends** tab.
2. On the **Performance Trends** tab, select specific performance metrics or select **Key Metrics**, **Select All**, or **Deselect All** in the top-right corner, and click **Save**.

Note:

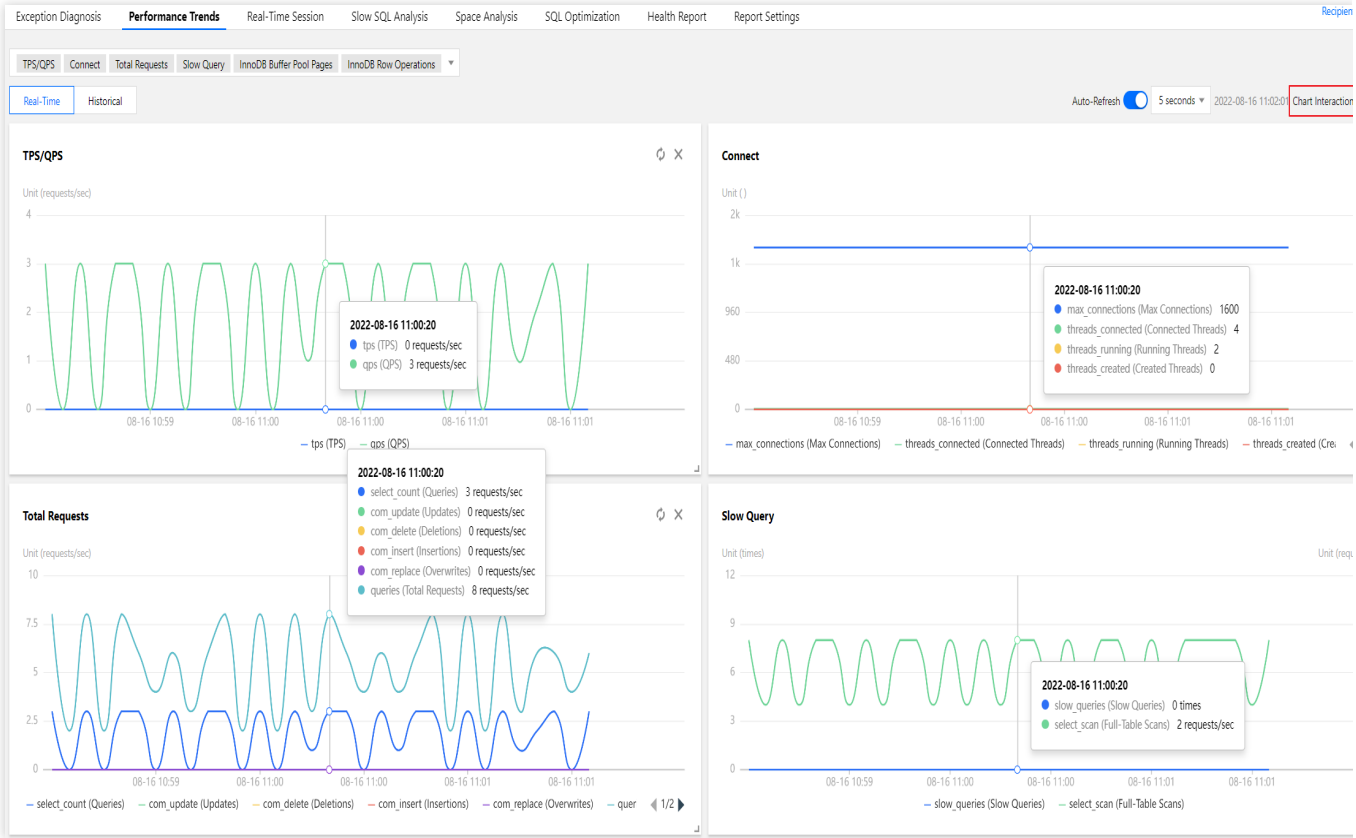
Click **Save** to apply the selected metrics to the current database instance, or click **Save and Apply to All Instances** to apply the selected metrics to all database instances.



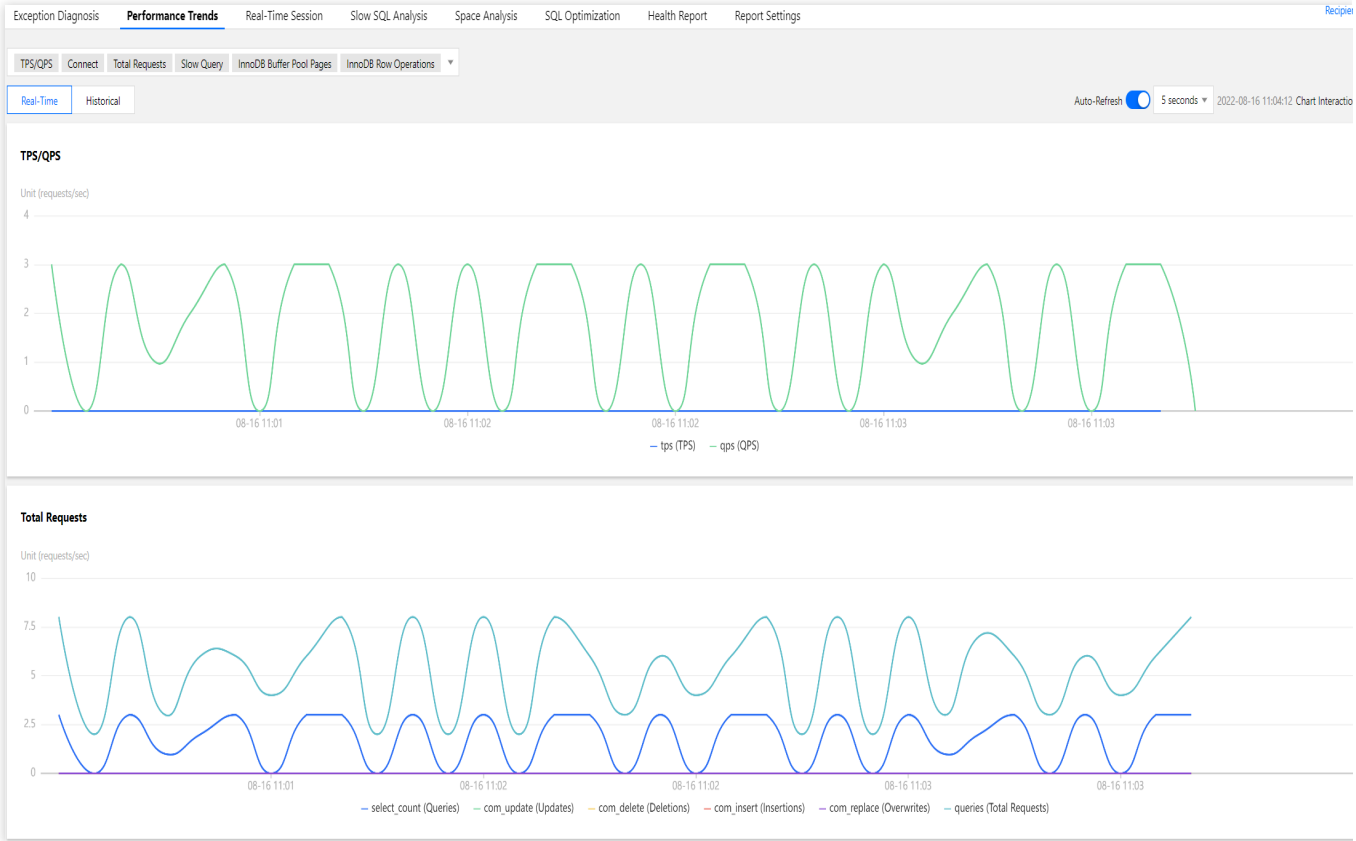
3. View metrics.

Chart interaction: Click **Chart Interaction** on the right to link and compare the monitoring views of multiple instances or metrics.

When you hover over a data point in any monitoring view, the data at the same time point will be displayed in other monitoring views. Click the data point to pin it for display. To unpin it, click **Deselect the Time Point**.

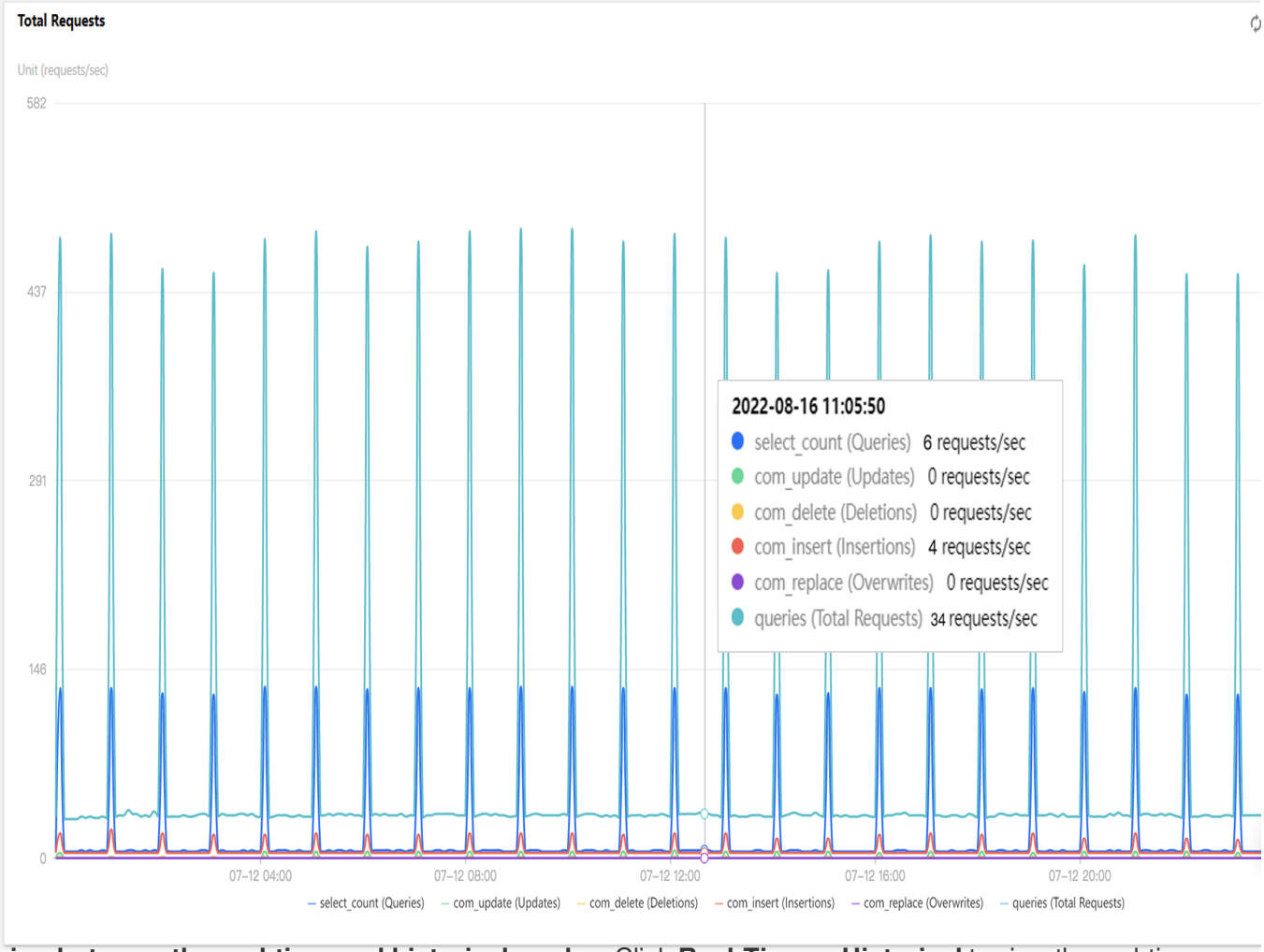


Switching between the one-column and two-column modes: Click the button on the right of **Chart Interaction** in the top-right corner to switch.



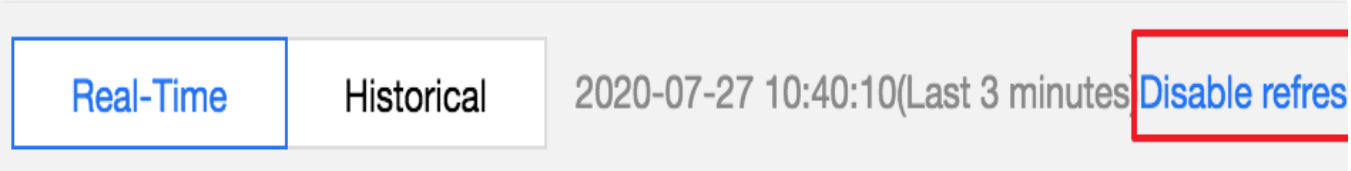
Dragging a monitoring view: Click the border of a monitoring view to drag it to the desired position.

Zooming in a monitoring view: Drag the icon in the bottom-right corner of a monitoring view to zoom it in for fine-grained display of the trend of one single performance metric.

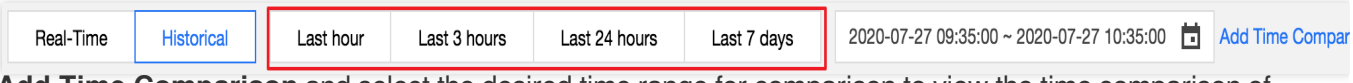


Switching between the real-time and historical modes: Click **Real-Time** or **Historical** to view the real-time or historical performance trends.

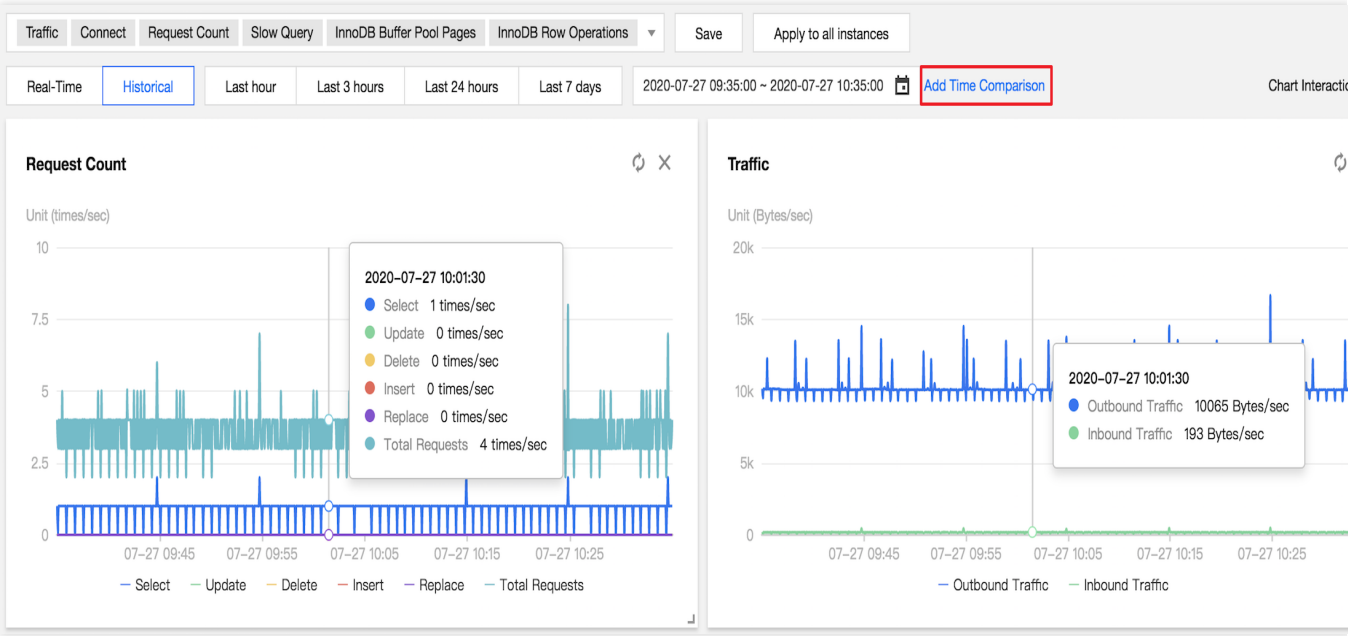
The real-time performance trends view displays the performance trends of the instance and is automatically refreshed by default. You can click **Disable refresh** to stop refreshing the trends in real time.



In the historical performance trends view, you can select a time range (**Last hour**, **Last 3 hours**, **Last 24 hours**, **Last 7 days**, or a custom time range) to display the performance trends over the selected time range.



Click **Add Time Comparison** and select the desired time range for comparison to view the time comparison of multiple performance metric trends.


Traffic
Connect
Request Count
Slow Query
InnoDB Buffer Pool Pages
InnoDB Row Operations
Save
Apply to all instances
Real-Time
Historical
Last hour
Last 3 hours
Last 24 hours
Last 7 days
2020-07-27 09:35:00 ~ 2020-07-27 10:35:00
Add Time Comparison
Chart Interactio
Request Count
Unit (times/sec)
10
7.5
5
2.5
0
2020-07-27 10:01:30
Select 1 times/sec
Update 0 times/sec
Delete 0 times/sec
Insert 0 times/sec
Replace 0 times/sec
Total Requests 4 times/sec
07-27 09:45
07-27 09:55
07-27 10:05
07-27 10:15
07-27 10:25
Select
Update
Delete
Insert
Replace
Total Requests
Traffic
Unit (Bytes/sec)
20k
15k
10k
5k
0
2020-07-27 10:01:30
Outbound Traffic 10065 Bytes/sec
Inbound Traffic 193 Bytes/sec
07-27 09:45
07-27 09:55
07-27 10:05
07-27 10:15
07-27 10:25
Outbound Traffic
Inbound Traffic

Real-Time Session

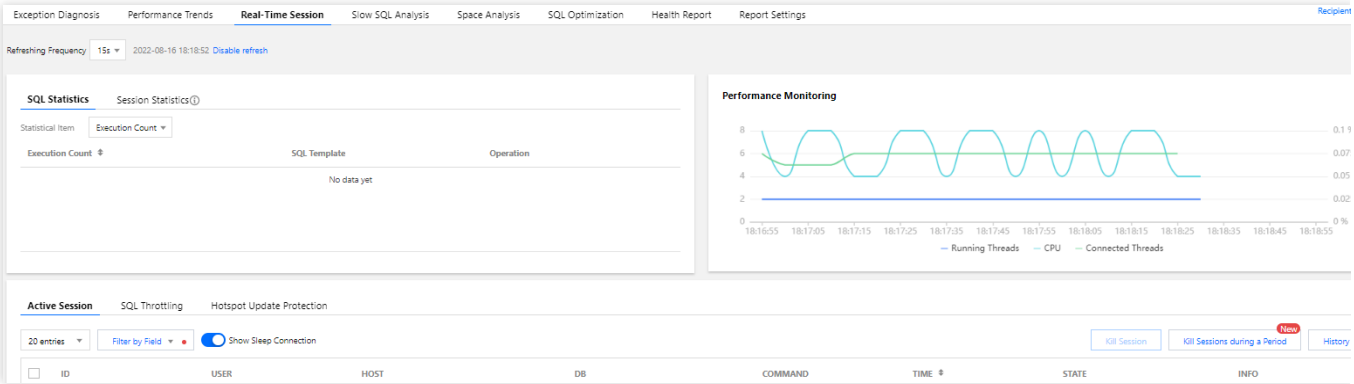
Last updated：2022-08-16 18:34:06

Feature description

You can use DBbrain's real-time session feature to view the real-time session information of your instance, including **Performance Monitoring**, **Connection Monitoring**, **Active Session**, **SQL Throttling**, and **Hotspot Update Protection**.

SQL statistics/Session statistics/Performance monitoring

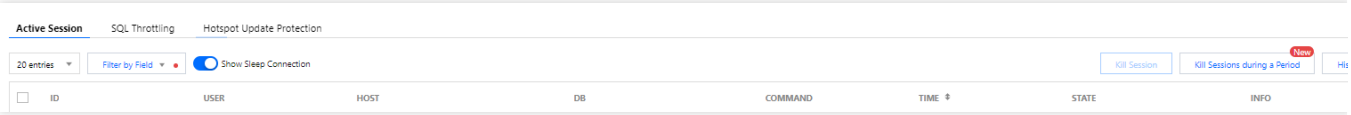
Log in to the DBbrain console and select **Performance Optimization** on the left sidebar. On the displayed page, select a database type and an instance at the top, and select the **Real-Time Session** tab.
The **Refreshing Frequency** is **15s** by default and can be modified as needed. You can also disable refresh.



Active session

On the **Active Session** tab, you can set the limit, filter by field, and enable or disable **Show Sleep Connection**.
You can set the limit to 20, 50, or 100.
Filter by Field supports filtering by **ID**, **USER**, **HOST**, **STATE**, **DB**, **COMMAND**, **INFO**, and **TIME** fields.
You can filter threads by **All**, **Not Sleep**, or **Others** (including Binlog Dump, Change user, Close stmt, Connect, Connect Out, Create DB, Daemon, Debug, Delayed insert, Drop DB, Error, Execute, Fetch, Field List, Init DB, Kill, Long Data, Ping, Prepare, Processlist, Query, Quit, Refresh, Register Slave, Reset stmt, Set option, Shutdown, Sleep, Statistics, Table Dump, and Time).
You can also enable **Show Sleep Connection**.

©2013-2025 Tencent Cloud International Pte. Ltd.



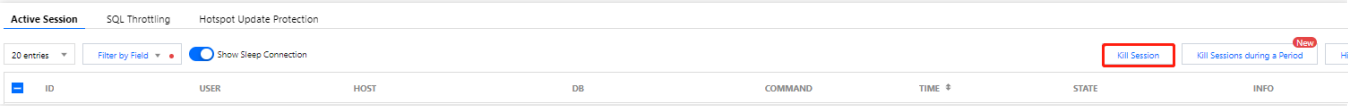
Killing sessions

DBbrain allows you to kill sessions for easier session management.

Kill current sessions

Select target sessions and click **Kill Session**.

You can kill 1–100 sessions at a time.



Kill sessions during a period

DBbrain offers the feature of killing sessions during a period. You can set the conditions for killing sessions, so that when the conditions are met, sessions will be killed automatically.

1. Task Settings.

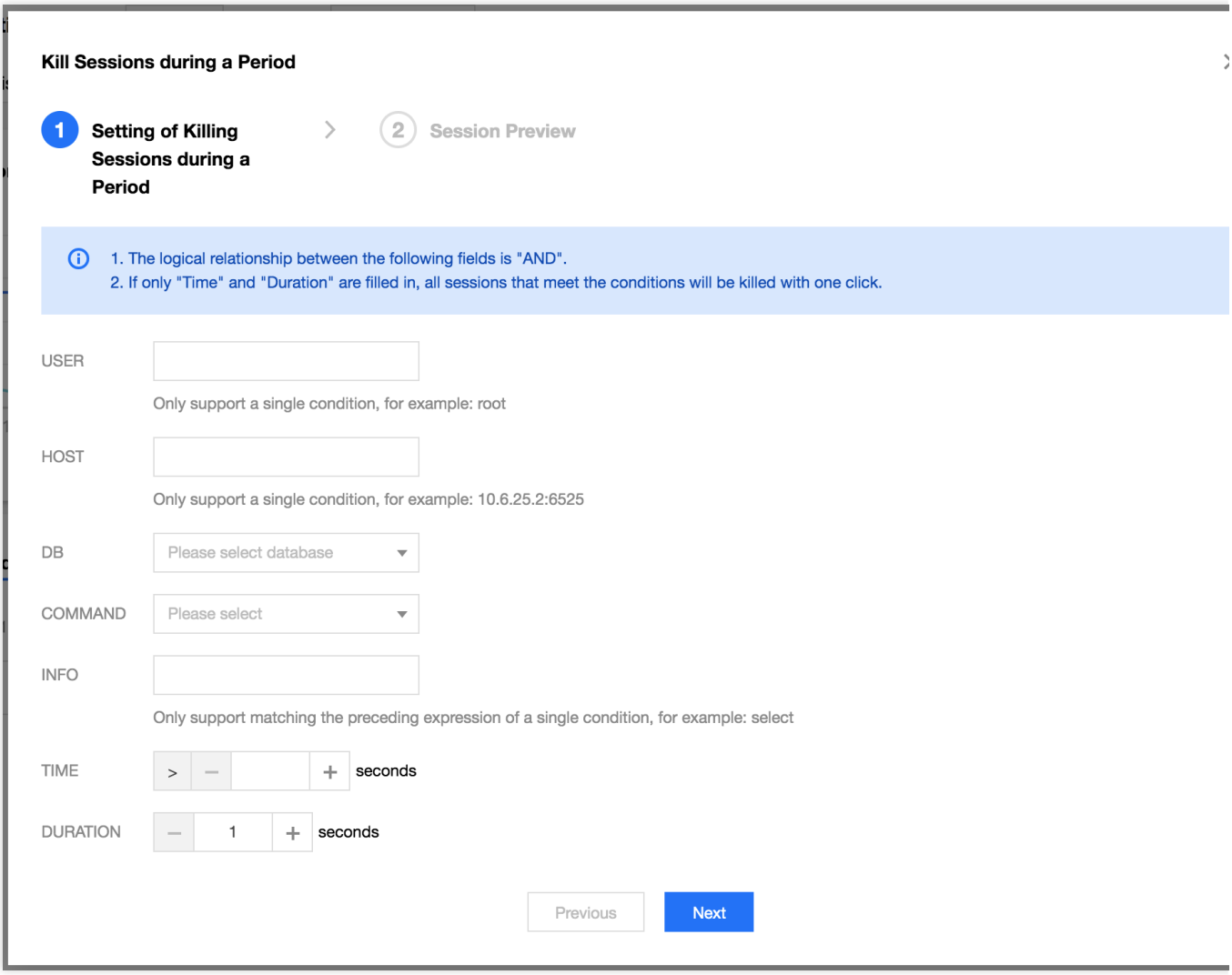
Set the conditions for killing sessions during a period (including **USER**, **HOST**, **DB**, **COMMAND**, **INFO**, and **TIME**) and set the **Execution Mode**.

Note:

You can set one or more filter conditions which are evaluated using the logical AND operator. Then, all sessions that meet the conditions except system connections will be killed.

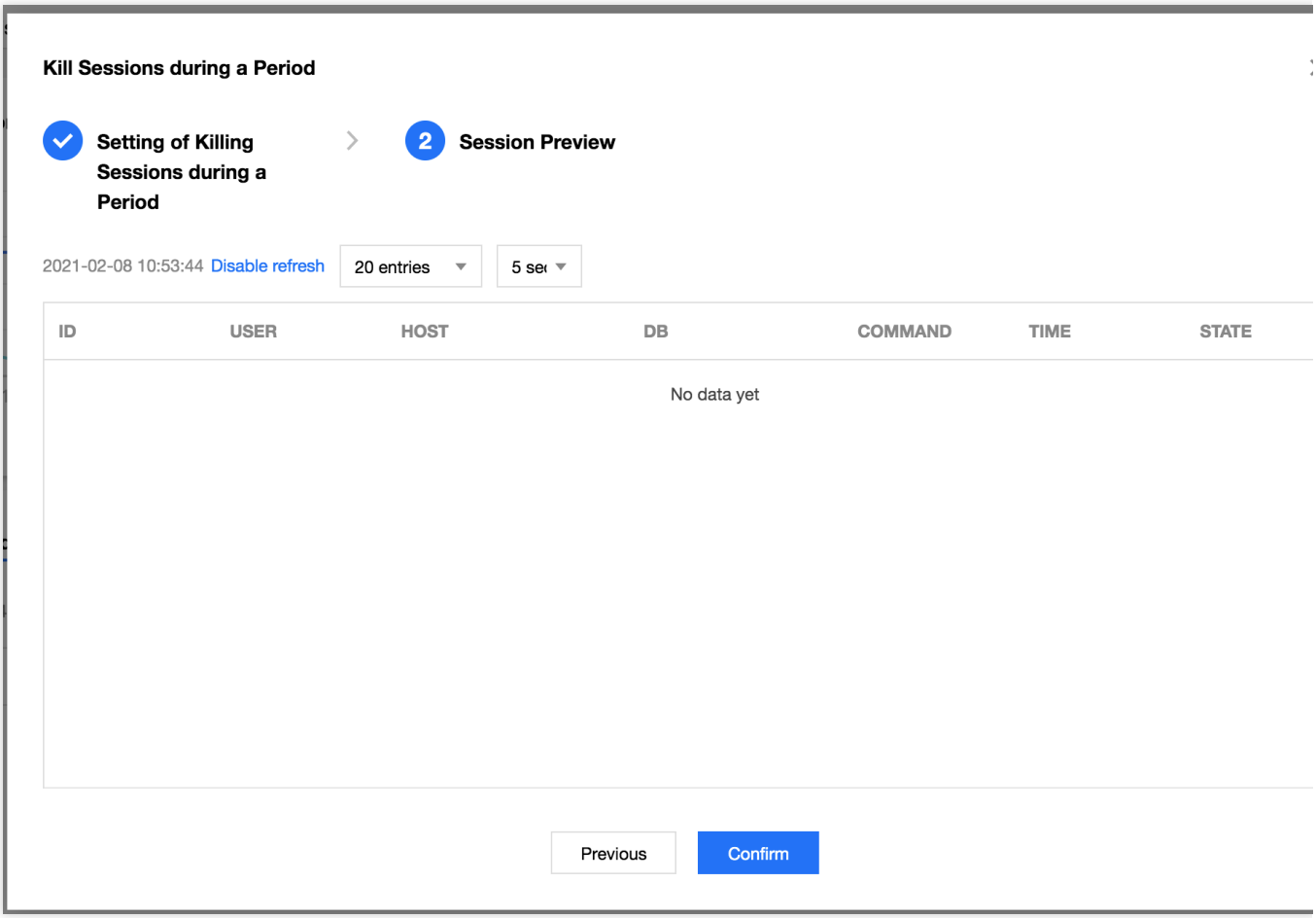
If only **Time** and **Duration** are set, all sessions that meet the conditions will be killed quickly.

©2013-2025 Tencent Cloud International Pte. Ltd.



2. Session Preview.

After setting the task, you can preview the sessions to be killed in the **Session Preview** section. After killing sessions during a period is enabled, the generated sessions that meet the conditions will be automatically killed.



3. Task Details.

After setting the task, click **Details** in the top-right corner to view the details of the sessions killed during a period.

View the history of killed sessions

DBbrain provides the feature of viewing the history of killed sessions. To use this feature, click **History**.

SQL throttling

DBbrain supports the SQL throttling feature to ensure service availability. You can create SQL throttling tasks to control the database requests and SQL concurrency by setting the **SQL Type**, **Max Concurrency**, **Throttling Duration**, and **SQL Keyword**. Multiple tasks do not conflict with each other.

Note:

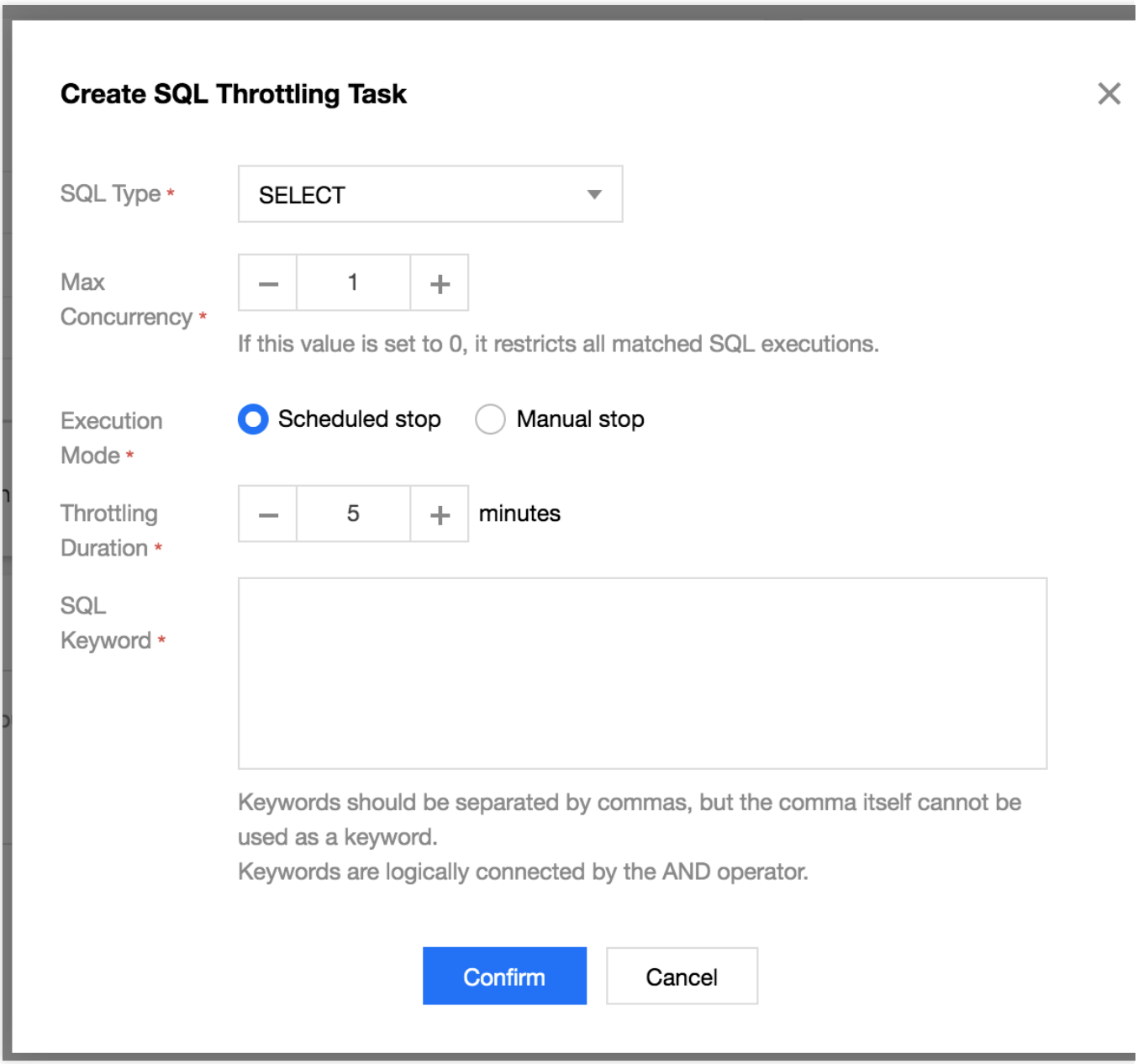
SQL throttling is supported only for TencentDB for MySQL (excluding the Basic Edition).

To create a SQL throttling task, you need to log in to the database account first.

If SQL throttling prevents a SQL statement from being executed, the error message `SQL rejected by CDB_SQL_FILTER` will be displayed.

SQL Type: Select **SELECT**, **UPDATE**, **DELETE**, **INSERT**, or **REPLACE**.

Max Concurrency: Set the maximum number of concurrent SQL executions. If the number of concurrent SQL executions containing specified keywords reaches this value, the SQL throttling policy will be triggered. If this value is set to 0, it restricts all matched SQL executions.
Execution Mode: Select **Scheduled stop** or **Manual stop**.
Throttling Duration: If you select **Scheduled stop**, you need to set how long the SQL throttling task runs.
SQL Keyword: Set the keywords. SQL statements containing the specified keywords will be restricted. Multiple keywords should be separated by comma and are evaluated by using the logical `AND` operator. Comma cannot be used as a keyword.



On the **SQL Throttling** tab, the list displays the **SQL Type**, **Status**, **Keyword**, **Start Time**, **Remaining Time**, **Max**

Concurrency, and **Operation**.

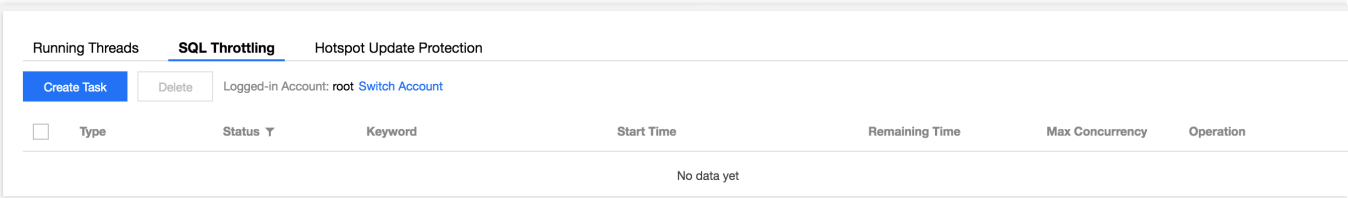
Click **Details** in the **Operation** column to view SQL throttling details.

After a SQL throttling task is enabled, it will remain in the **Running** status until its remaining time decreases to zero.

You can click **Disable** in the **Operation** column to disable the task, and its status will change to **Terminated**.

After a SQL throttling task is enabled, its status will change to **Terminated** once its remaining time decreases to zero.

Click **Delete** in the **Operation** column to delete a SQL throttling task in the **Terminated** or **Completed** status.



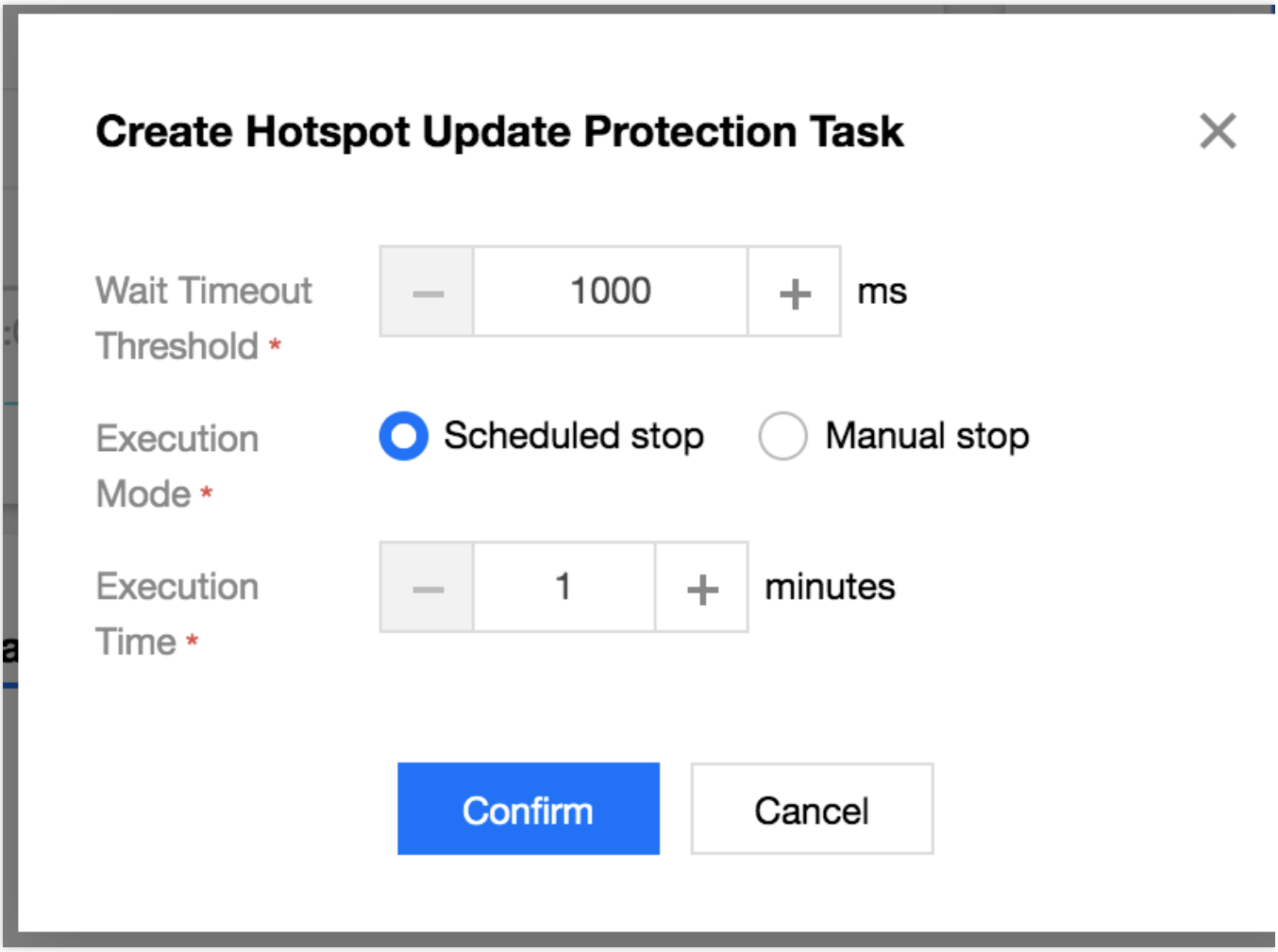
Hotspot update protection

DBbrain provides the hotspot update protection feature. According to the statement queuing mechanism, this feature queues the statements with the same conflict in the memory queue. It reduces the overheads of lock conflict and improves the database performance in high concurrency scenarios.

Note:

Hotspot update protection is supported only for TencentDB for MySQL (excluding the Basic Edition).

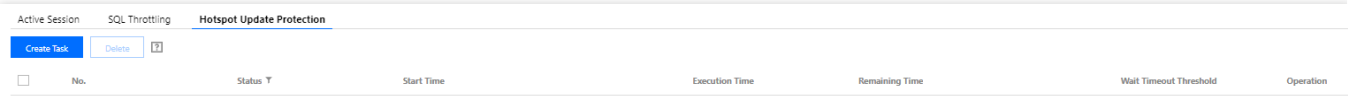
On the **Hotspot Update Protection** tab, click **Create Task** to create a hotspot update protection task. You can set the **Wait Timeout Threshold** and **Execution Mode** (**Scheduled stop** or **Manual stop**). If you select **Scheduled stop**, you can set the **Execution Time**.



On the **Hotspot Update Protection** tab, the list displays the **Status**, **Start Time**, **Execution Time**, **Remaining Time**, **Wait Timeout Threshold**, and **Operation**.

For a task in the **Running** status, click **Disable** in the **Operation** column to terminate it.

For a task in the **Terminated** or **Completed** status, click **Delete** in the **Operation** column to delete it.



Slow SQL Analysis

Last updated：2022-08-16 18:44:21

Feature description

The slow SQL analysis feature calculates, samples, and aggregates records and execution information (source information, number of executions, execution duration, result set, scan set, etc.) of slow SQL statements in the instance. This feature analyzes the performance of slow SQL statements based on the execution plan, comprehensive resource usage, sizes of scan and result sets, and index usage rationality of the aggregated SQL statements and provides optimization suggestions.

Note:

Before you use slow log analysis for self-built databases accessed through the Agent, you need to check whether slow log collection is enabled at https://console.intl.cloud.tencent.com/dbbrain/instance?product=dbbrain-mysql.

Self-built database instances that access the service directly do not support slow log analysis.

Viewing slow SQL analysis

1. Log in to the DBbrain console and select **Performance Optimization** on the left sidebar. On the displayed page, select a database type and an instance at the top, and select the **Slow SQL Analysis** tab.

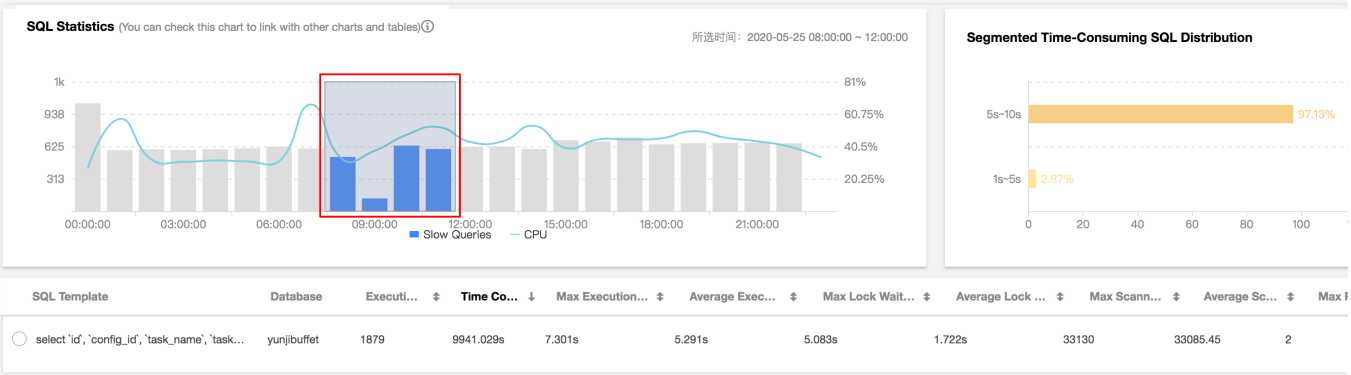
Note:

The **SQL Statistics** section displays the number of slow queries and CPU utilization of the instance. You can adjust the time range to view slow SQL statements. If the instance has slow SQL statements, the quantity and occurrence points in time will be displayed in the view.

2. You can click a single time range or drag to select multiple time ranges for slow queries in the **SQL Statistics** section, and the aggregated SQL template and execution information (including the number of executions, total execution duration, scanned rows, and returned rows) will be displayed below. Each column of data can be sorted in ascending or descending order. The consumed time distribution section on the right displays the overall consumed time distribution of SQL statements in the selected time range.

You can quickly set the time dimension for statistics collection to **Last 5 minutes**, **Last 10 minutes**, **Last hour**, **Last 3 hours**, **Last 24 hours**, or **Last 3 days**.

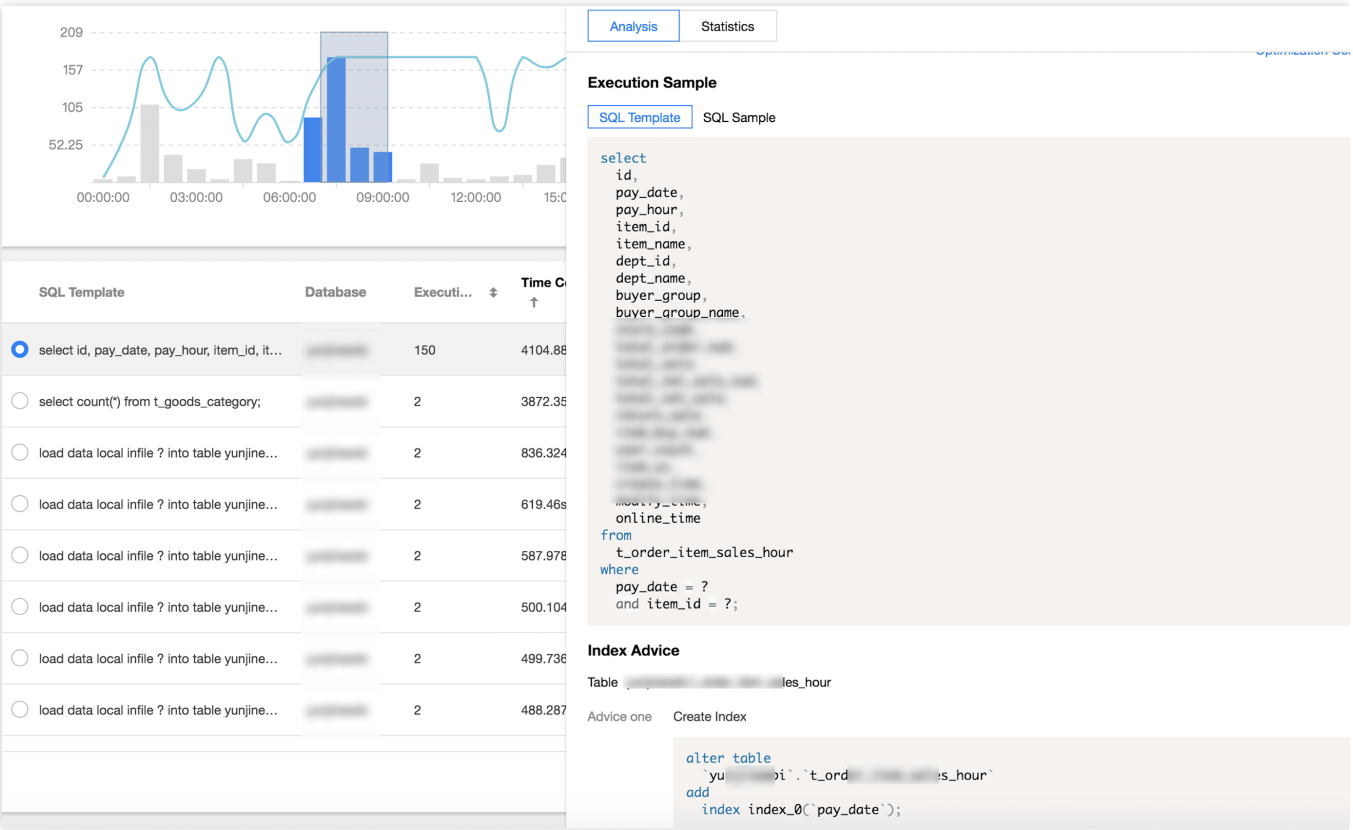
©2013-2025 Tencent Cloud International Pte. Ltd.



3. Click an aggregated SQL template in the **SQL Template** column as shown in the red box above, and specific SQL analysis and statistics will be displayed on the right.

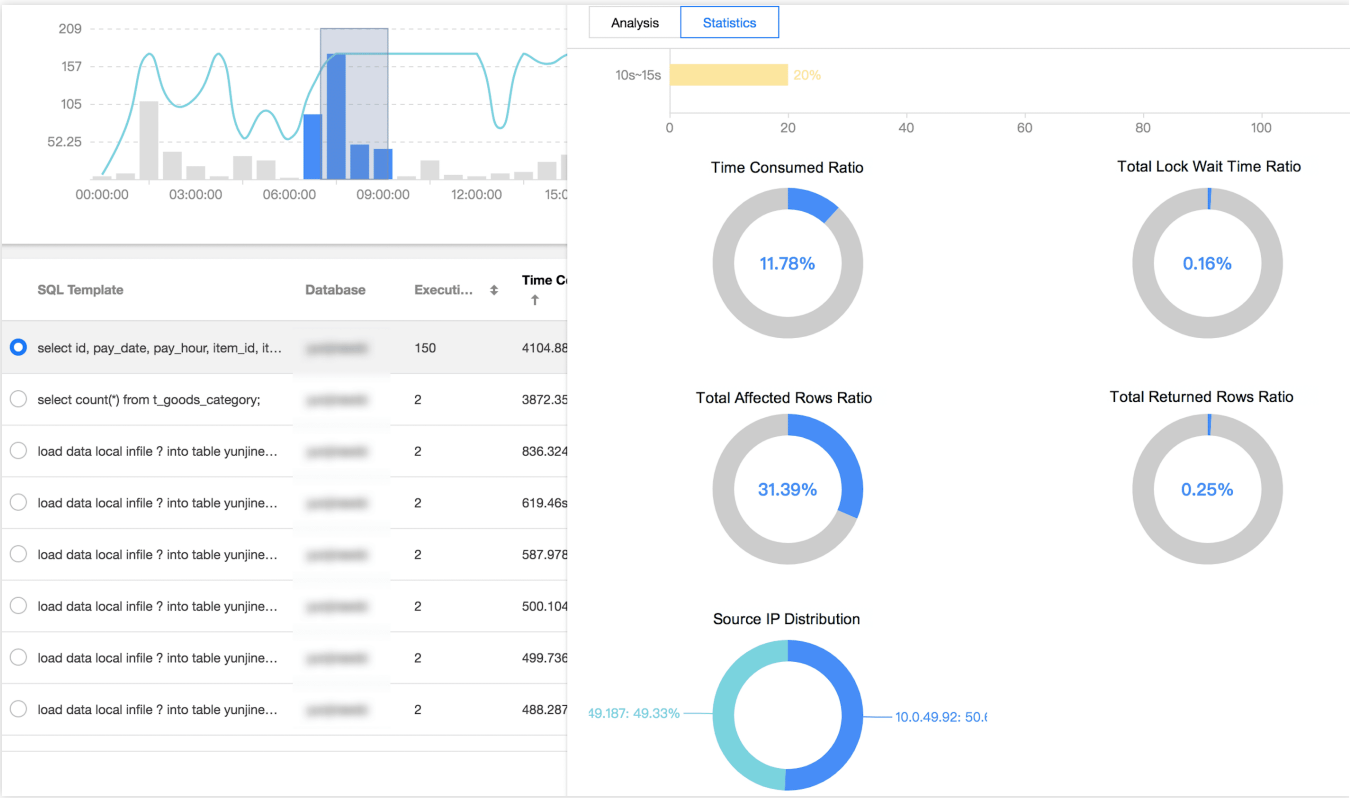
On the **Analysis** tab, you can view the complete SQL template, SQL sample, optimization suggestion, and description. You can optimize your SQL statement based on the expert suggestions provided by DBbrain to improve the statement quality and reduce the delay.

The **Analysis** > **Execution Plan** tab provides the visual analysis result. The visual chart can be zoomed in or out or displayed in full screen mode. Click a number or icon in the chart to view further details.



On the **Statistics** tab, you can perform cross-sectional analysis on the root cause of a slow SQL statement based on the total lock wait time ratio, total affected rows ratio, and total returned rows ratio in the statistics report, and then

optimize the statement accordingly. You can also view the execution duration distribution of the specified type of aggregated SQL statements and the proportions of access source IPs.



On the **Details** tab, you can view the SQL execution details.

4. Export the slow SQL data.

Click **Export** on the right of the SQL list to export the data of slow SQL analysis in CSV format for easier viewing.

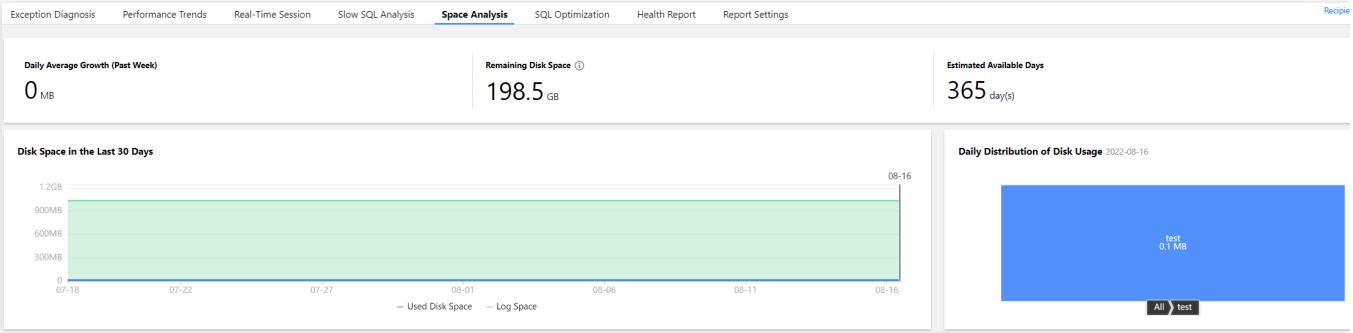
Space Analysis

Last updated：2024-01-04 15:50:15

With DBbrain's space analysis feature, you can view the instance space utilization, including the sizes of data and logs, the daily increase in space utilization, the estimated number of available days, and the space used by the tables and databases of the instance.

Disk space

Log in to the DBbrain console and select **Performance Optimization** on the left sidebar. On the displayed page, select a database type and an instance at the top, and select the **Space Analysis** tab.
On the **Space Analysis** tab, you can view the daily average growth in the past week, remaining disk space, estimated available days, daily distribution of disk usage, and disk space trend in the last 30 days.
For TencentDB for MySQL, the remaining disk space = purchased disk space - data space.
For TDSQL-C for MySQL, the remaining storage space = maximum storage space - data space.
For self-built MySQL, the remaining disk space = disk space - data space.
For TencentDB for MongoDB, the remaining disk space = maximum storage space - data space.

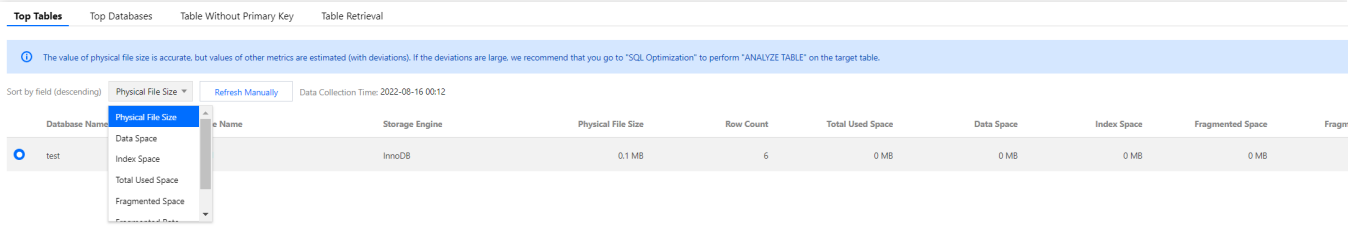


Top tablespace

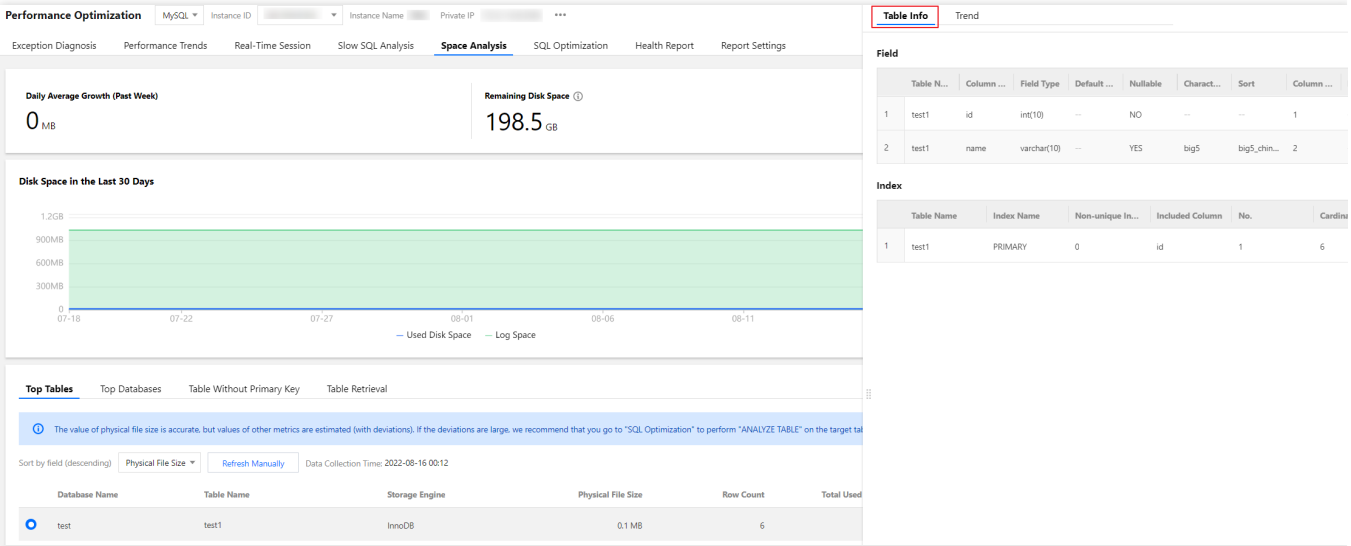
Note:
You can manually refresh the top tables/databases data. Data is collected once a day by default. When the information is inaccurate due to the large gap between the data collection time and the current time, you can click **Refresh Manually** to collect and analyze the real-time data of top tables/databases. Note that there may be a slight delay when the instance has many databases and tables or when the access pressure is high.

©2013-2025 Tencent Cloud International Pte. Ltd.

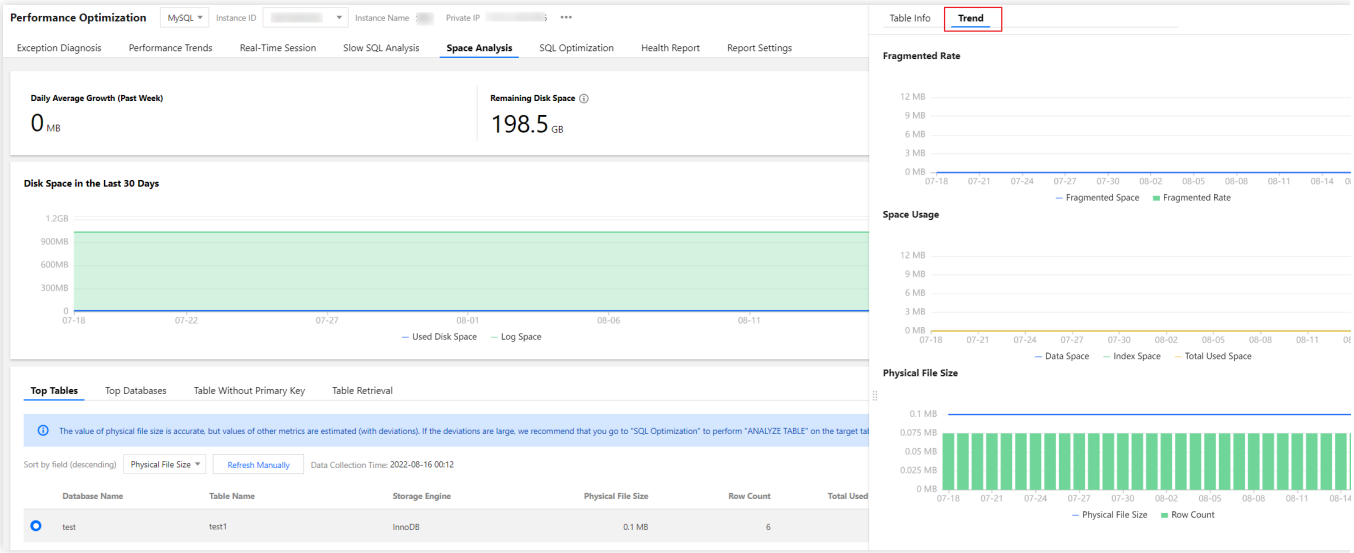
The **Top Tablespace** section shows the details of the tables that have relatively high space usage. The table list in the section contains columns such as **Storage Engine**, **Physical File Size**, **Row Count**, **Total Used Space**, **Data Space**, **Index Space**, **Fragmented Space**, and **Fragmented Rate**. Each column of data can be sorted in descending order, and the real-time data can be refreshed manually. You can view the disk space usage details in this section and perform optimization promptly.



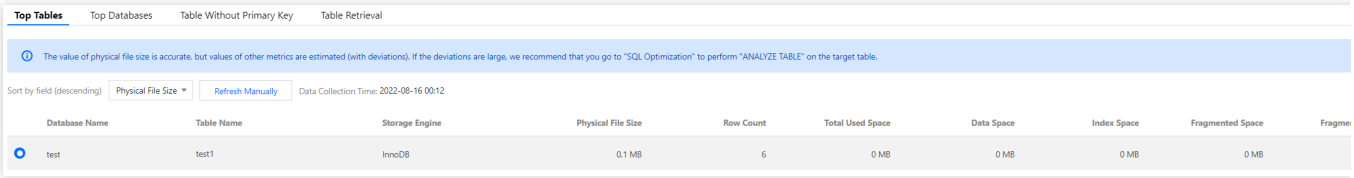
The **Top Tables** section displays data by table. In the table list, you can click the row of a table to view its field and index information. The field information includes the **Table Name**, **Column Name**, **Field Type**, **Default Value**, **Nullable**, **Character Set**, **Sort**, **Column Position**, and **Remarks**. The index information includes the **Table Name**, **Index Name**, **Unique Index**, **Included Column**, **No.**, and **Cardinality**.



In the table list, click the row of a table to view the space usage trend, including the trends of the **Physical File Size**, **Space Usage** (**Data Space**, **Index Space**, and **Total Used Space**), and **Fragmented Rate**.

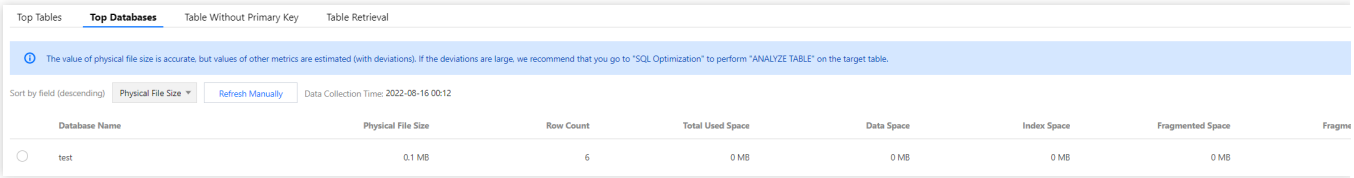


Click the download icon in the top-right corner to download the data of top tables in CSV format.

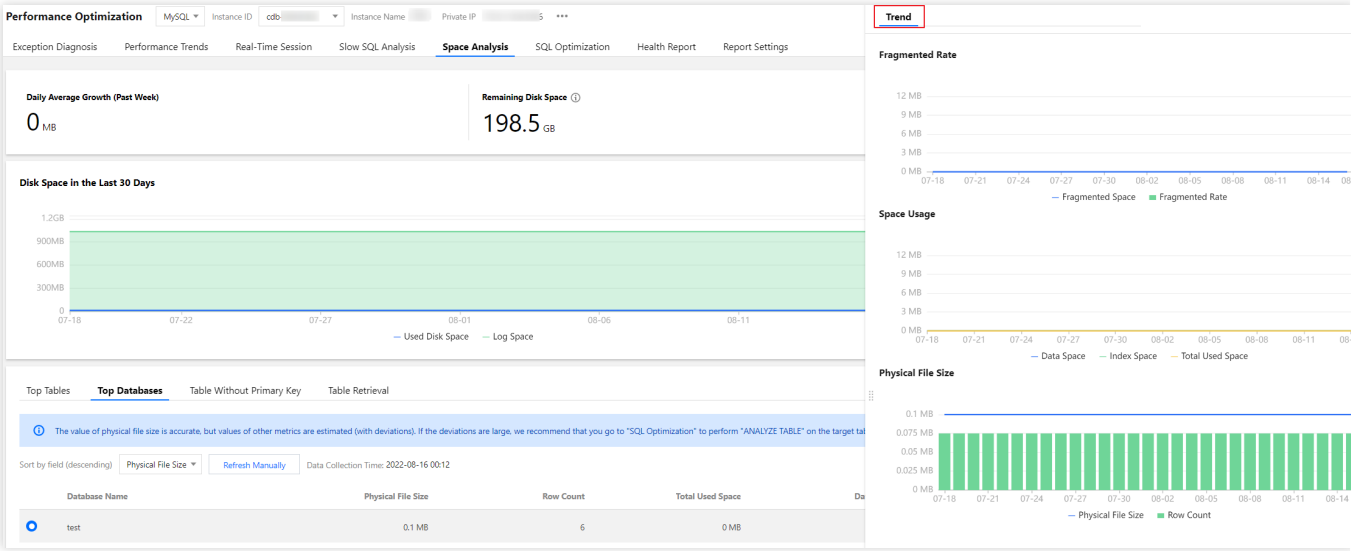


Top databases

The **Top Databases** section shows the details of the databases that have relatively high space usage. The database list in the section contains columns such as the **Physical File Size**, **Row Count**, **Total Used Space**, **Data Space**, **Index Space**, **Fragmented Space**, and **Fragmented Rate**. Each column of data can be sorted in descending order. You can view the disk space usage details in this section and perform optimization promptly.



The **Top Databases** section displays data by database. In the database list, click the row of a database to view the space usage trend, including the trends of the **Physical File Size**, **Space Usage** (**Data Space**, **Index Space**, and **Total Used Space**), and **Fragmented Rate**.



Click the download icon in the top-right corner to download the data of top databases in CSV format.

Tables without a primary key

The **Table Without Primary Key** section displays the information of tables that lack a primary key in the current instance. Such tables have potential risks and will affect the instance's read/write performance, sync efficiency, etc. We recommend you process them timely and add primary keys suitable for your business scenario.

The list of tables without a primary key supports two refreshing methods: regular scan (once per day) and manual refresh. You can click a table in the list to view its field and index information.

Click the download icon in the top-right corner to download the data of tables without a primary key in CSV format.

SQL Optimization

Last updated：2022-08-15 14:25:00

Feature description

The SQL optimization feature enables you to optimize SQL statements in just a few clicks and provides the corresponding execution plan interpretation and optimization suggestion. It is suitable for scenarios such as slow SQL statement optimization, pre-release code review, and self-check.

This feature provides expert suggestions about SQL optimization and supports many database management features, including viewing database table structures or executing/modifying SQL statements in the console. It helps you optimize all aspects of SQL statements and allows you to manipulate databases in the same way as you do in a database client tool.

You can manually enter SQL statements and analyze them to get their performance evaluation results and optimization suggestions.

The visual execution plan feature is added to help you understand the entire SQL statement execution process and details. In this way, you can easily get a grasp of your statement performance overheads.

Note:

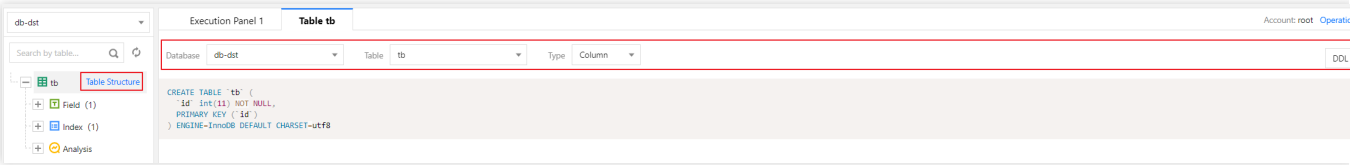
Currently, SQL optimization is supported only for TencentDB for MySQL (excluding basic single-node instances), TDSQL-C for MySQL, and self-built MySQL databases.

Optimizer execution

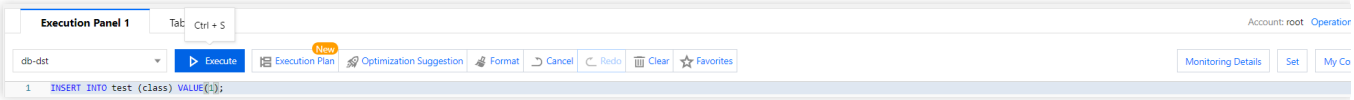
1. Log in to the DBbrain console and select **Performance Optimization** on the left sidebar. On the displayed page, select a database type and an instance at the top, and select the **SQL Optimization** tab.
2. On the **SQL Optimization** tab, you can view the information of database tables, SQL statements, and SQL execution.

The left section displays databases, tables, fields, and index names. You can filter databases by database name and click **Table Structure** next to a table to view its details.

The right section displays SQL details. You can filter data by database, table, or type and view data in the **Table** or **DDL** mode.



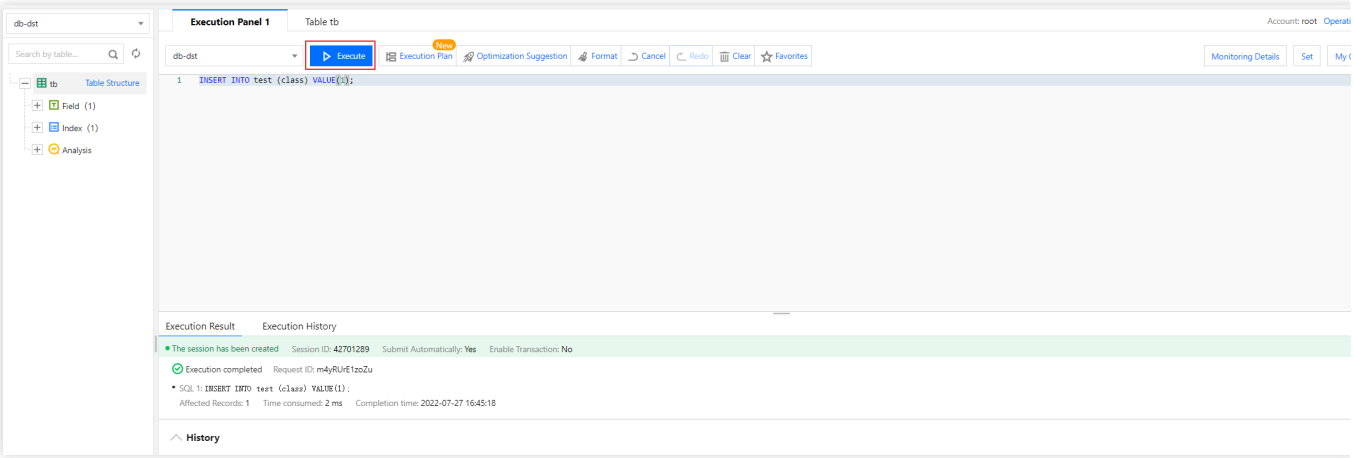
3. On the execution panel, you can enter or paste a SQL statement to execute it, format it, or view its execution plan and optimization suggestion. You can also clear it or cancel or redo your operations.
Each operation can be controlled with a keyboard shortcut, which can be viewed by hovering over the corresponding button.



Click **Execute** to execute the entered SQL statement. You can also view the **Execution Result** and **Execution History** or clear the record of the execution result.

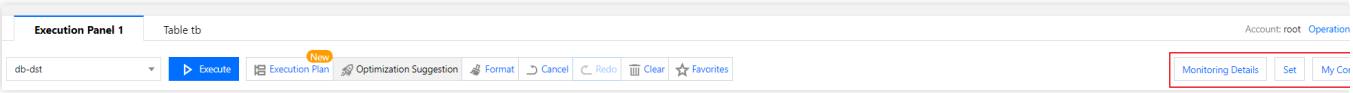
Note:

You can only view the SQL execution plan if you are not logged in. To perform operations such as SQL optimization, log in to the target database instance first.



Click **Execution History** to view the SQL execution history. You can also switch to view the history of the current session or all sessions.
Click **Execution Plan** to view the SQL execution plan details and optimization suggestion. For more information, see Visual execution plan.
Click **Format** to format the selected SQL statement as shown below:
Click **Optimization Suggestion** to view the optimization suggestion for the SQL statement.
In the **Optimization Comparison** window, you can view the SQL statement's execution plan, index advice, rewriting advice, table structure, and performance before and after optimization.
The performance of an optimized SQL statement is estimated based on the analysis of the statistics of database tables related to the statement, the OPTIMIZER_SWITCH configuration, and the index selectivity. A chart is used to visually show the decrease in the performance. You can also compare the execution plans before and after SQL optimization to further verify the optimization results.

4. On the right of the execution panel, you can view the monitoring details, set the SQL query conditions, and view historical commands.



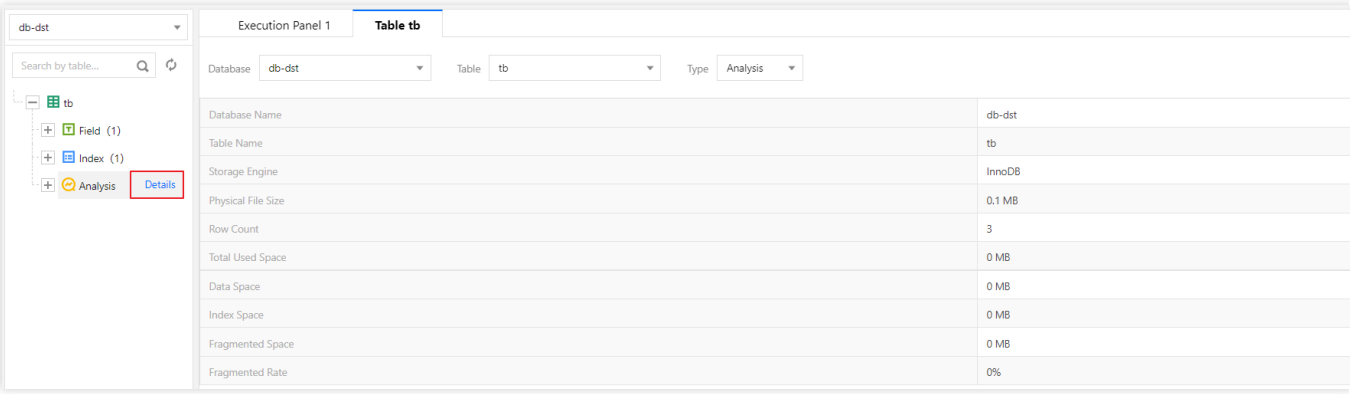
Click **Monitoring Details** on the right to view the monitoring information of the database instance.

Click **Set** on the right to set specific query conditions, including the **Execution Timeout Period** and **Max Returned Rows**.

Click **My Commands** on the right to view your favorites and system Ops SQL templates, including parameter/metric, user, information_schema, and other templates. These templates help you execute common Ops SQL statements easily and quickly.

5. View the table analysis data.

Select the target table on the left and click the **Analysis** tab to view the table analysis data on the right.



Visual execution plan

1. Log in to the DBbrain console and select **Performance Optimization** on the left sidebar. On the displayed page, select a database type and an instance at the top, and select the **SQL Optimization** tab.

2. On the **SQL Optimization** tab, you can see the button bar on the execution panel.

3. The first button on the right of the **Execute** button can start the **visual execution plan** feature.

Enter or select the SQL statement on which you want to perform a visual analysis on the execution panel.

Click **Execution Plan** to display the visual execution plan effect.

Click the small button in the base table block to view the structure of the table.

Click the small button in the step block to get the SQL information of the step.

Click each information block to get the node details of the step, which may vary by node.

The statement execution plan helps you better understand which steps generate temp tables or file sorting.
Different index types are reflected by different colors based on the performance.
Depending on the complexity of your statements, different visual matrix effects will be displayed. If there is too much content in the visual graphical area, you can use the scaling icons to freely adjust the displayed area or enable the full screen mode.

©2013-2025 Tencent Cloud International Pte. Ltd.

Deadlock Visualization

Last updated：2022-08-11 15:45:29

Background

In a database system, when multiple processes concurrently access the same data, the locking mechanism can ensure that the data is only accessed by one process at a time, ensuring data integrity and consistency. Because of resource preemption during execution, locking may cause a deadlock in which two or more processes wait for each other.
There are many types of deadlocks, and the entire lock system is very complex. In InnoDB's lock system, there are table-level locks and row-level locks, depending on their granularity. Row locks include gap locks, insert intention locks, and next-key locks. They are divided into exclusive locks and shared locks according to their mode. Some of these locks are compatible, while some are incompatible. In addition, both the isolation level and data access method affect the scope of locking and the types of lock.
Viewing deadlock logs has traditionally been used to locate deadlocks, which is inefficient and requires database lock system expertise.

Feature Description

DBbrain offers an all-new deadlock visualization feature to intelligently diagnose and analyze database deadlocks and help you use better SQL statements to eliminate unreasonable locking. This effectively reduces slow queries, improves the resource utilization, and prevents deadlocks.
Visual topology: The topology of a deadlock is displayed to visually reproduce the deadlock situation and reflect the information of and wait relationship between transactions.
Lock information display: Click a lock in the visual chart to view the scope of locking, locked data, etc.
SQL information display: Executions are inferred through SQL parsing to help avoid deadlocks.

Operation Entry

1. Log in to the DBbrain console and select **Performance Optimization** on the left sidebar. On the displayed page, select a database type and an instance at the top, and select the **Exception Diagnosis** tab.
2. In the **Diagnosis Prompt** list, if a diagnosis item is **Deadlock**, click it to enter the deadlock analysis and visualization page.

©2013-2025 Tencent Cloud International Pte. Ltd.

Visual Topology

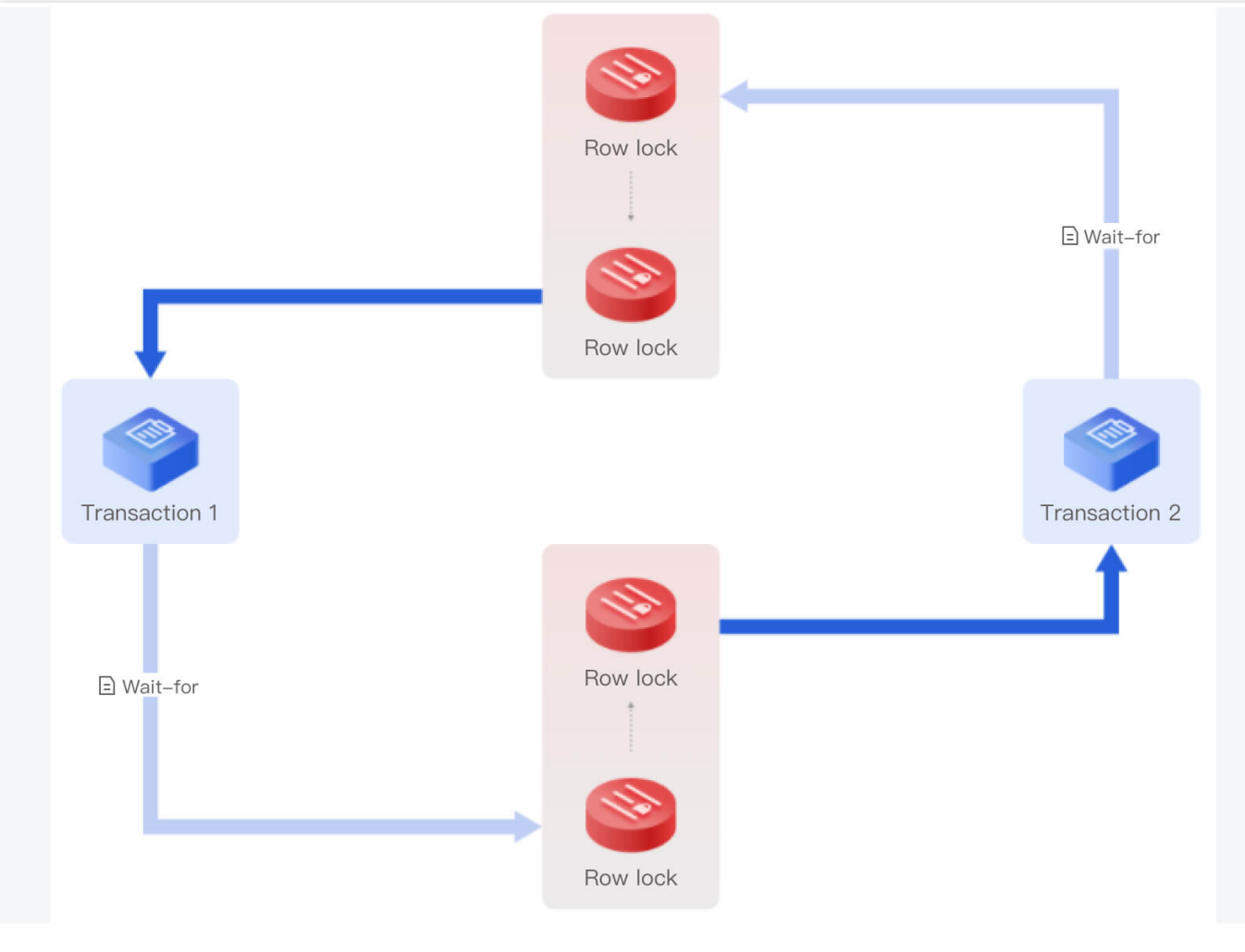
Traditionally, deadlocks are located by viewing deadlock logs, and the information about the last deadlock in InnoDB can be viewed through `SHOW ENGINE INNODB STATUS`. The logs show the SQL statements and transaction IDs but not `lock_mode X waiting` and `hex 80000007` data (particularly the relationship between the locks involved in the deadlock situation). They can be efficiently analyzed and located only if you have a good knowledge of database locking systems and deadlock logs.

```
*** (1) TRANSACTION:
TRANSACTION 1741848, ACTIVE 1 sec starting index read
mysql tables in use 1, locked 1
LOCK WAIT 2 lock struct(s), heap size 1136, 1 row lock(s)
MySQL thread id 12, OS thread handle 123145410191360, query id 154 localhost 127.0.0.1 root updating
DELETE FROM dept_manager where num = 7
*** (1) WAITING FOR THIS LOCK TO BE GRANTED:RECORD LOCKS space id 383 page no 4 n bits 80 index num of table
`employees`.`dept_manager` trx id 1741848 lock_mode X waiting
Record lock, heap no 6 PHYSICAL RECORD: n_fields 2; compact format; info bits 32
0: len 4; hex 80000007; asc ;;
1: len 4; hex 800003f0; asc ;;
```

DBbrain visually displays the topology of a deadlock, with transactions and locks as points (to display the requesting and holding relationships between transactions and locks), and with the conflicts between locks as lines (to form a cycle). The following illustrates the visual topologies of various deadlocks.

Sample 1. A deadlock between two transactions

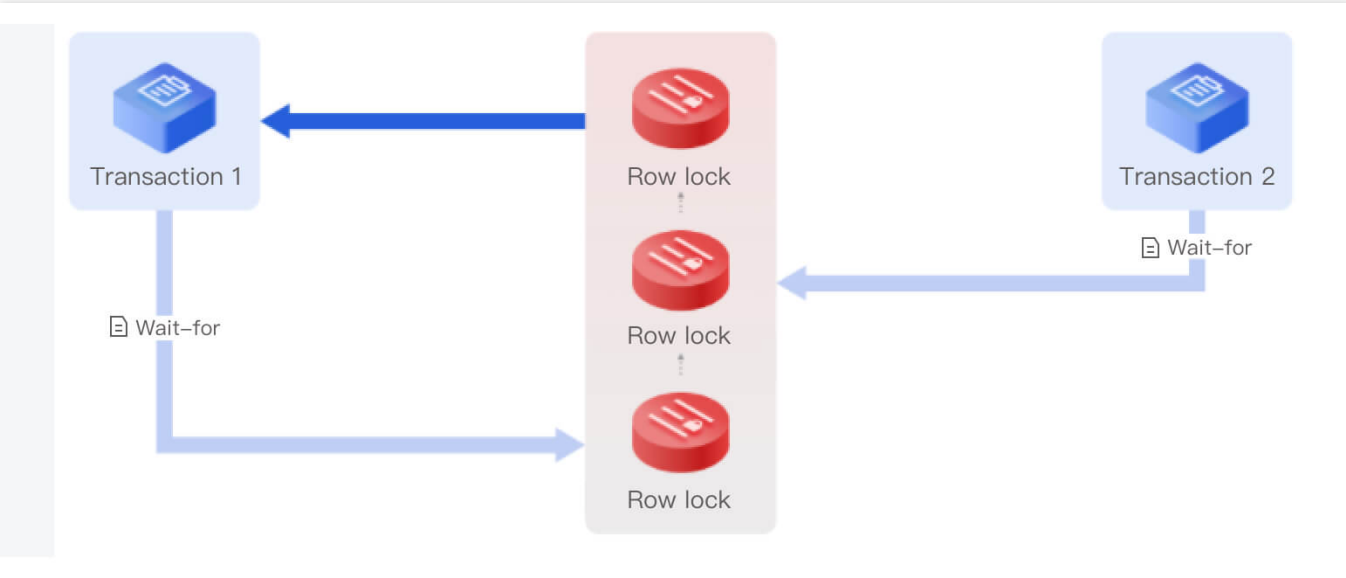
Transaction 1 and transaction 2 each hold a lock (represented by a dark blue line) and request a lock (represented by a light blue line). The lock held by transaction 1 blocks the lock requested by transaction 2, and the lock requested by transaction 1 is blocked by the lock held by transaction 2, causing a deadlock.
Incompatible and conflicting locks that are placed on the same record are connected by a dotted line.



Sample 2. A deadlock caused while waiting for unlocking

In MySQL, even if a waited unlock has not been acquired successfully (i.e., in waiting status), it can still block other lock requests. This is different from locks in operating systems.

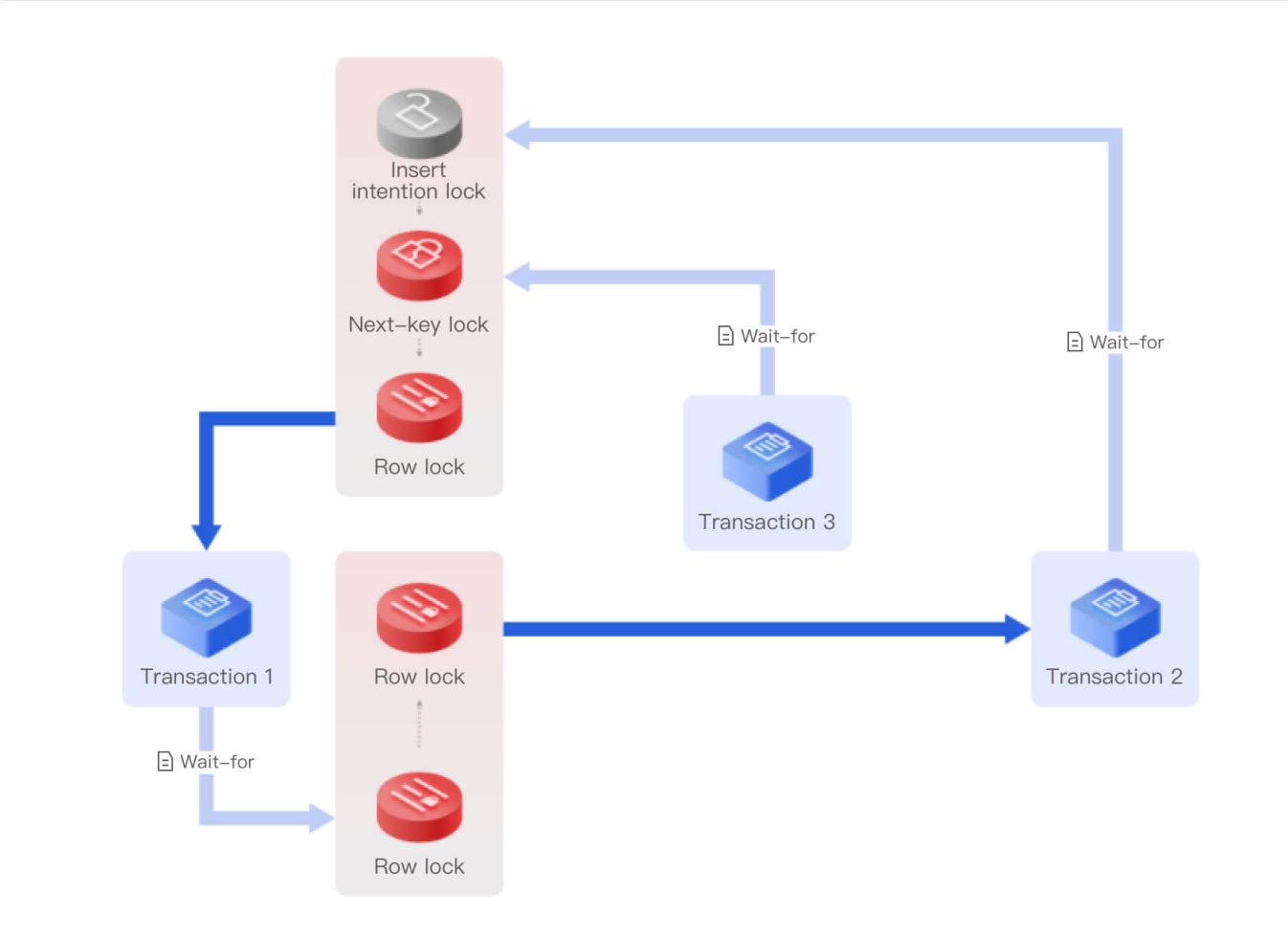
As shown below, the row lock requested by transaction 2 (represented by a light blue line) is blocked by the row lock held by transaction 1 (represented by a dark blue line), and the row lock requested by transaction 2 blocks the row lock requested by transaction 1, causing a deadlock.



Sample 3. A deadlock among three transactions

The next-key lock requested by transaction 3 is blocked by the row lock held by transaction 1 (represented by a dark blue line), the row lock requested by transaction 1 (represented by a light blue line) is blocked by the row lock held by transaction 2 (represented by a dark blue line), and the insert intention lock requested by transaction 2 (represented by a light blue line) is blocked by the row lock held by transaction 1, causing a deadlock among the three transactions.

©2013-2025 Tencent Cloud International Pte. Ltd.

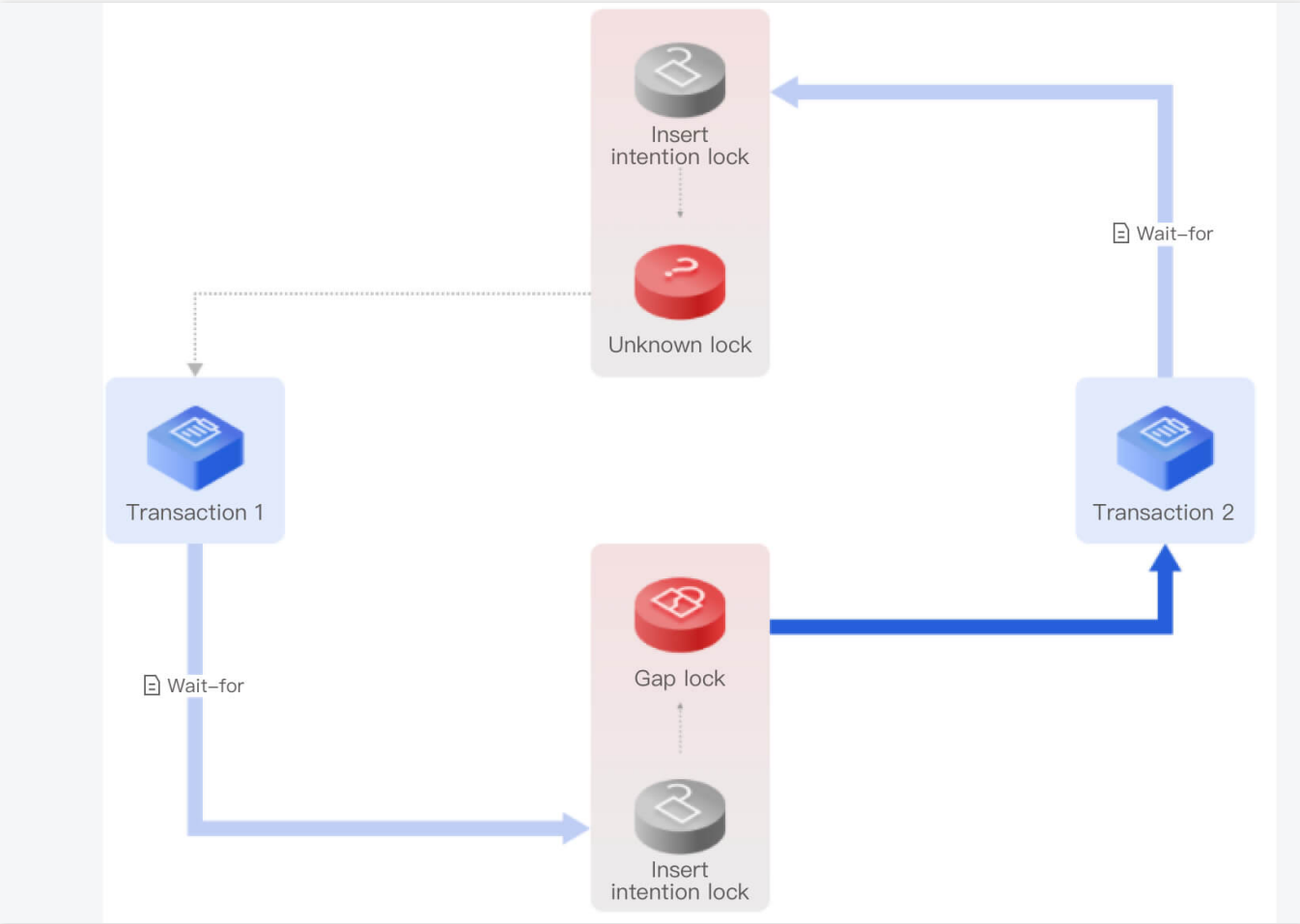


Sample 4. An "unknown lock" (for MySQL 5.6 and 5.7)

Samples 1, 2, and 3 use MySQL 8.0 as an example, which offers complete deadlock logs. If a transaction encounters a conflict when requesting a lock, it will check for a deadlock by looking for a cycle in the wait-for graph.

MySQL 5.6 and 5.7 only use the depth-first search method to search for cycles, but do not record cycles. Therefore, the deadlock logs record only the first and last transactions and are incomplete.

To address this problem, DBbrain introduces the concept of "unknown lock" to make the cycle complete. An unknown lock refers to a lock that should be here as inferred but we don't know what lock it is. There should be a path between the unknown lock and transaction 1, and there may be another transaction along this path, which is therefore represented by a dotted line.

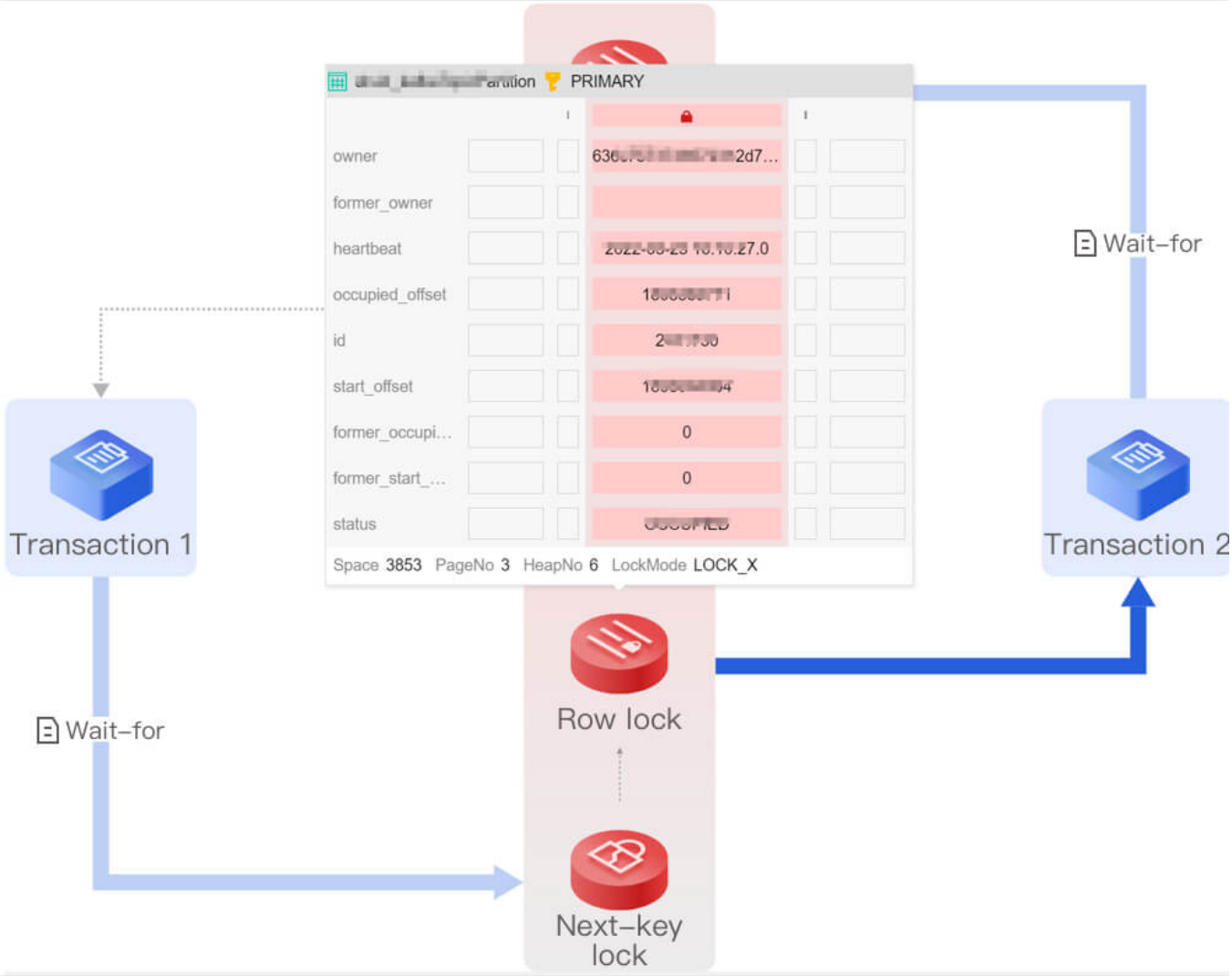


Lock Information Display

A deadlock log displays the lock information, including the lock mode (exclusive or shared lock), waiting status, and lock type (such as row lock, gap lock, next-key lock, or insert intention lock). A record lock is a lock on one or multiple records, for which the deadlock log records the physical address, including the `space` , `page no` , and `heap no` , schema, index, and other information. The recorded data is displayed by using a list, but only hexadecimal strings are printed, which are unreadable.

```
RECORD LOCKS space id 11 page no 4 n bits 120 index PRIMARY of table `employees`.`test` trx id 13331 lock_mode X locks rec bu
not gap
Record lock, heap no 26 PHYSICAL RECORD: n_fields 7; compact format; info bits 128
 0: len 4; hex 8001adc6; asc     ;;
 1: len 4; hex 64303031; asc d001;;
 2: len 6; hex 00000000337e; asc     3~;;
 3: len 7; hex 0200000167024c; asc     g L;;
 4: len 3; hex 8f8221; asc   !;;
 5: len 3; hex 8f8f41; asc   A;;
 6: len 4; hex 8001adc6; asc     ;;
```

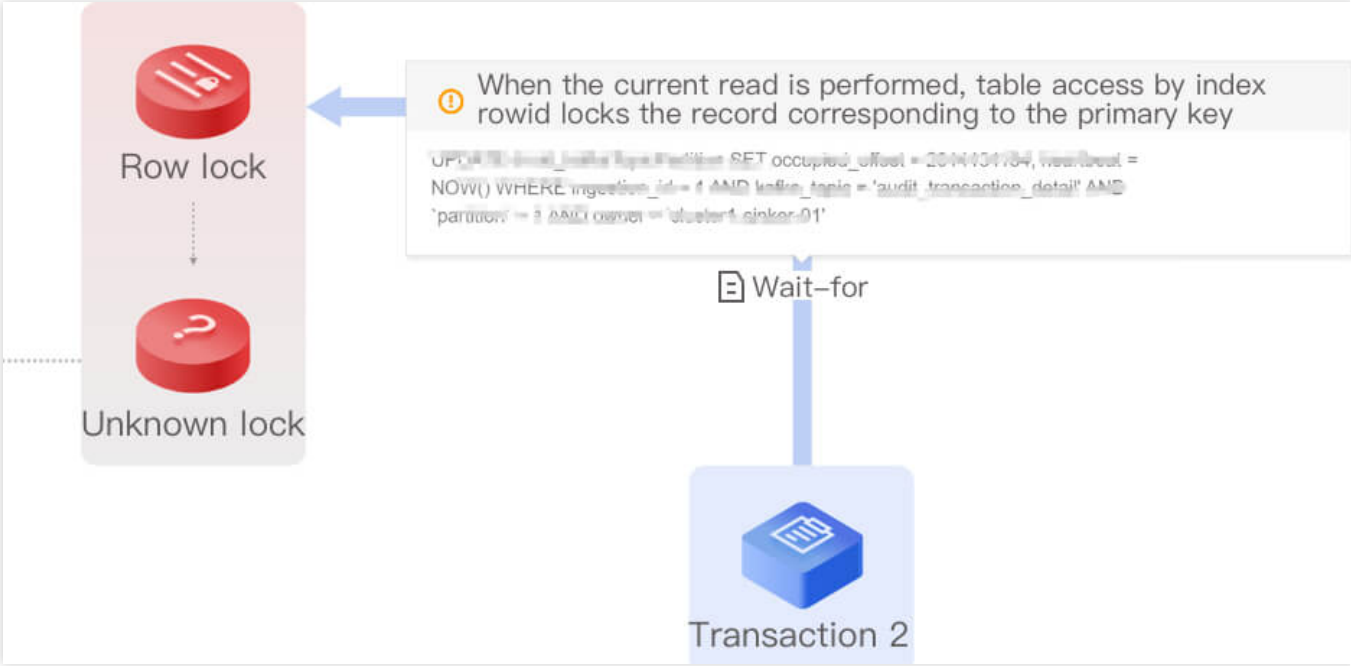
DBbrain displays the lock information, including the scope of locked data, locked row records, and locked gap. Click a lock in the visual chart to view the scope of data, gap locked by the lock, and other information. Click a transaction to view its details.



SQL Information Display

In addition to locating the deadlock situation and relevant information, DBbrain also helps you perform further smart diagnosis. It displays the SQL statement executed when a deadlock occurred in the chart and adds comments to explain what happened when the SQL statement was executed, as well as the rule that MySQL used to place the lock. This helps you optimize your business and SQL statements more quickly and easily.

In the visual chart, click **Waits for** to view the detailed information of the SQL statement.



Event Notification

Last updated：2022-08-11 15:45:29

For more information, see Event Notification.

©2013-2025 Tencent Cloud International Pte. Ltd.

Best Practices

Last updated：2022-08-11 15:45:29

Fixing High CPU Utilization on MySQL Instance
Fixing Lock Conflict on MySQL Instance

©2013-2025 Tencent Cloud International Pte. Ltd.

Redis Performance Optimization Exception Diagnosis

Last updated：2025-04-17 17:38:42

The exception diagnosis feature provides you with real-time performance monitoring, health inspections, and failure diagnosis, so that you can intuitively know the real-time operation status of database instances, locate newly appeared performance exceptions in real time.

Overview

Viewing Diagnosis Information

1. Log in to the DBbrain Console.
2. In the left sidebar, choose **Performance Optimization**.
3. Select the corresponding database type and instance ID at the top, and select the **Exception Diagnosis** tab.
4. On the right side of the page, select to view real-time or historical diagnosis information.

5. View the health score trend chart, diagnosed exception events, and instance architecture diagram within the selected timeline.

View health score trend chart

Click any time point on the trend chart to display the health score.

View diagnosis event bar chart

Hover over the diagnosis event bar chart to display information such as risk level, overview, and start/end time. Click the bar chart to enter the Event Details page to view information including event details, on-site descriptions, intelligent analysis, and optimization suggestions. For more information on viewing event details, see Exception Alarms.

View real-time data for health score and instance architecture diagram.

Health score: Real-time data will be displayed for health score, CPU utilization, memory utilization, connection utilization, read request hit rate, inbound traffic utilization, and outbound traffic utilization.

Click **Details** under the health score to enter the Health Report page and view the health score, score details, and health report.

Instance architecture diagram: Displays the Proxy and node architecture of the instance, including the location of nodes triggering alerts. Hover over the corresponding node or Proxy to display average metrics for the selected node.

Viewing Diagnosis Prompts

The diagnosis event levels are categorized as healthy, notice, warning, severe, and critical. DBbrain performs regular health checks on the instance every 10 minutes.

1. Log in to the DBbrain Console.
2. In the left sidebar, choose **Performance Optimization**.
3. Select the corresponding database type and instance ID at the top, and select the **Exception Diagnosis** tab.
4. On the right side of the page, select to view real-time or historical diagnosis information.

Real-Time: Select real-time to display the risk distribution and diagnosis details for the last three hours.

Historical: Select history to display the risk distribution and diagnosis details for the selected time period.

5. View the diagnosis prompts for the selected time range.

View diagnosis event details

In the **Diagnosis Details**, click the row of a specific event alarm or hover over the event alarm and click **View** to enter the Event Details page and view the event details.

Event details mainly include event details, on-site descriptions, intelligent analysis, and optimization suggestions. The event details displayed vary depending on the diagnosis type. Refer to the actual display.

Event Details: They include the diagnosis item, start/end time, risk level, and overview.

Description: They include problem snapshots and performance trends of the exception or health inspection events.

Ignore/Unignore alarms

In the **Diagnosis Details**, hover over the event alarm and click **Ignore** to select **Ignore this item** or **Ignore this type**, and click **OK**. You can also ignore alarms on the Event Details page.

Note:

This feature is only for exception alarms with diagnosis items other than "Health Inspection."

Ignore This: It means you can only ignore this alarm.

Ignore This Type: It means you can ignore exception alarms generated from the same root cause.

Ignored diagnosis events will be grayed out. To unignore, you can also click **Unignore**.

Detailed Description of Diagnosis Items

Diagnosis items related to intelligent diagnosis are categorized into four types: performance, availability, reliability, and maintainability. Each diagnosis item belongs to one category only.

Name of Diagnosis Items	Type of Diagnosis Items	Note:	Risk Level Classification
Node CPU Utilization	Performance	Node CPU utilization is too high.	Critical: node CPU utilization ≥ 95 Serious: 95＜node CPU utilization ≥ 90 Alarm: 90＜node CPU utilization ≥ 80 Note: 80＜node CPU utilization ≥ 60
Node Memory Utilization	Performance	Node memory utilization is too high.	Critical: node memory utilization ≥ 95 Serious: 95＜node memory utilization ≥ 90 Alarm: 90＜node memory utilization ≥ 80 Note: 80＜node memory utilization ≥ 60
Node Connection Utilization	Performance	Node connection utilization is too high.	Critical: Node Connection Utilization Rate ≥ 95 Serious: 95＜node connection utilization ≥ 90 Alarm: 90＜node connection utilization ≥ 80 Note: 80＜node connection utilization ≥ 60
Proxy Connection Utilization	Performance	Proxy connection utilization is too high.	Critical: proxy connection utilization ≥ 95 Serious: 95＜proxy connection utilization ≥ 90 Alarm: 90＜proxy connection utilization ≥ 80 Note: 80＜proxy connection utilization ≥ 60
Proxy Inflow Utilization	Performance	Proxy inbound traffic usage is too high.	Critical: Proxy inbound traffic usage ≥ 1536 Serious: 1536＜proxy inbound traffic usage ≥ 1228.8 Alarm: 1228.8＜proxy inbound traffic usage ≥ 1024 Note: 1024＜proxy inbound traffic usage ≥ 800
Proxy Outflow Utilization	Performance	Proxy outbound traffic usage is too high.	Critical: proxy outbound traffic usage ≥ 1536 Serious: 1536＜proxy outbound traffic usage ≥ 1228.8 Alarm: 1228.8＜proxy outbound traffic usage ≥ 1024 Note: 1024＜proxy outbound traffic usage ≥ 800
Proxy Inflow	Performance	Proxy Inbound Traffic	Critical

Limit Occur		Throttling	
Proxy Outflow Limit Occur	Performance	Proxy Inbound Traffic Throttling	Critical
Error Command	Maintainability	There are error commands detected.	Alarm
Risk	Maintainability	There are high-risk commands detected.	Alarm
Connectivity Health Check	Availability	Database connection error, unable to connect to the database instance.	Critical

©2013-2025 Tencent Cloud International Pte. Ltd.

Performance Trends

Last updated：2025-04-17 17:41:10

Real-time performance trend monitoring tracks key performance metrics for Redis database instances, Redis nodes, and Proxy nodes, including CPU, memory, key information, network usage, network utilization, requests, and responses. Monitoring data is collected at a granularity of seconds, dynamically displaying changes in various metrics graphically and statistically summarizing maximum, minimum, and average values in table form. It supports multi-node metric comparisons, time-range comparison analysis, and flexible drag-and-zoom monitoring views.
With its powerful data statistical analysis capabilities, diverse display options, and high real-time performance, it meets various Ops and troubleshooting scenarios for database instances, helping Ops personnel quickly grasp the overall performance status of databases and prevent risks promptly.

Supported Performance Metrics

DBbrain currently supports monitoring the following Tencent Cloud Redis database performance metrics:

Category	Category Child Items	Chinese Metric Name	English Metric Name	Unit	M
Instance	CPU	CPU Utilization	cpu_util	%	Av ut
		Max Node CPU Utilization	cpu_max_util	%	M ut no re in
	Memory Info	Memory Usage	mem_used	MB	M ac in ar
		Memory Utilization	mem_util	%	Th ac m ap
		Max Node MEM Utilization	mem_max_util	%	M m ut no

					re in
	Latency	Avg Execution Latency	latency_avg	millisecond	Th ex fro th
		Max Execution Latency	latency_max	millisecond	Th ex fro th
		P99 Execution Latency	latency_p99	millisecond	Th ex fro th
		Avg Read Latency	latency_read	millisecond	Th ex of co th Re
		Avg Write Latency	latency_write	millisecond	Th ex of co th Re
		Avg Latency of Other Commands	latency_other	millisecond	Th ex of ot ar co th Re
	Key Info	Total keys	keys	pcs	Th of ar lev
		Expired Keys	expired	pcs	Th

					ke wi wi cc th ou cc
		Evicted Keys	evicted	pcs	Th ke wi wi cc th ou cc
	Network Usage	Connections	connections	pcs	Th TC to
		Inbound Traffic	in_flow	Mb/second	Pr inl
		Outbound Traffic	out_flow	Mb/second	Pr ou
	Network Utilization	Connection Utilization	connections_util	%	Th ac cc to cc
		Inbound Traffic Utilization	in_bandwidth_util	%	Th ac pr inl th tra
		Outbound Traffic Utilization	out_bandwidth_util	%	Th ac pr ou th tra

		Inbound Traffic Throttling Trigger	in_flow_limit	times	Th tin inl ex ma ba
		Outbound Traffic Throttling Trigger	out_flow_limit	times	Th tin ou ex ma ba
	Request	Total Requests	commands	times/second	Av ut
		Read Request	cmd_read	times/second	M ut no re ins
		Write Request	cmd_write	times/second	Th tin wr ex se
		Other Requests	cmd_other	times/second	Th tin co ex se re co
		Big Value Request	cmd_big_value	times/second	Th tin co ex se re ex
		Key Requests	cmd_key_count	pcs/second	Th ke

					se
		Mget Requests	cmd_cmget	times/second	Th re ex se M
	Response	Slow Query	cmd_slow	times	Th tin co ex is th sl co
		Read Request Hit	cmd_hits	times	Th re ex co th ke m by co
		Read Request Miss	cmd_miss	times	Th re ke ex co th ke m by co
		Read Request Hit Rate	cmd_hits_ratio	%	Ke + m ca W is nu

	Execution Error	Execution Error	cmd_err	times/second	Th tin cc ex oc nc cc pa
Redis Node	CPU	CPU Utilization	cpu_util	%	Av ut
	Memory Info	Memory Usage	mem_used	MB	M ac in ar
		Memory Utilization	mem_util	%	Th ac m ap
	Key Info	Total Keys	keys	pcs	Th of ar le
		Expired Keys	expired	pcs	Th ke wi wi cc th ou cc
		Evicted Keys	evicted	pcs	Th ke wi wi cc th ou cc
	Replication Delay	Replication Delay	repl_delay	B	Th cc

					le re th
	Network Usage	Connections	connections	pcs	Th co th no
		Connection Utilization	connections_util	%	No ut
	Request	Total Requests	commands	times/second	Q tin co ex
		Read Request	cmd_read	times/second	Th tin re ex
		Write Request	cmd_write	times/second	Th tin wr ex
		Other Requests	cmd_other	times/second	Th tin co ex th wr
	Response	Slow Query	cmd_slow	times	Th tin co ex is th sl co
		Read Request Hit	cmd_hits	times	Th re ex co

					th ke m by cc
		Read Request Miss	cmd_miss	times	Th re ke ex cc th ke m by cc
		Read Request Hit Rate	cmd_hits_ratio	%	Ke + m th Ca
Proxy node	CPU	CPU Utilization	cpu_util	%	Pr ut
	Traffic	Inbound Traffic	in_flow	Mb/second	Pr inl
		Outbound Traffic	out_flow	Mb/second	Pr ou
	Request	Total Requests	proxy_commands	times/second	Th cc ex pr
		Key Requests	cmd_key_count	pcs/second	Th ke th
		Mget Request	cmd_mget	times/second	Th tin M ex
		Execution	cmd_err	times/second	Th

		Error			tin cc ex su ex cc pa
		Big Value Requests	cmd_big_value	times/second	Th tin re ex th se
	Network Usage	Connections	connections	pcs	Th TC to
		Connections per Sec	client_connections_received_per_second	times/second	Th TC es se
		Disconnections per Sec	client_connections_closed_per_second	times/second	Th TC di se
		Abnomal Disconnections per Sec	client_connections_aborted_per_second	times/second	Th ab di pe
	Network Utilization	Connection Utilization	connections_util	%	Th ac cc to cc
		Inbound Traffic Utilization	in_bandwidth_util	%	Th ac pr inl th tra

		Outbound Traffic Utilization	out_bandwidth_util	%	Th ac pr ou th tra
		Inbound Traffic Throttling Trigger	in_flow_limit	times	Th tin inl tri th
		Outbound Traffic Throttling Trigger	out_flow_limit	times	Th tin ou tri th
	Latency	Avg Execution latency	latency_avg	millisecond	Th ex fro th
		Max Execution Latency	latency_max	millisecond	Th ex fro th
		P99 Execution Latency	latency_p99	millisecond	Th ex fro th
		Avg Read Latency	latency_read	millisecond	Th ex of co th Re
		Avg Write Latency	latency_write	millisecond	Th ex of co th Re

		Avg Latency of Other Commands	latency_other	millisecond	Th ex of ot ar co th Re

Viewing Performance Trend

Step 1: Selecting Monitoring Dimensions

1. Log in to the DBbrain Console.
2. In the left sidebar, choose **Performance Optimization**.
3. Select the Redis database type and instance ID at the top, and select the **Performance Trends** tab.

4. Select the monitoring dimension you want to view.

Monitoring dimensions support Redis instance monitoring, Redis node monitoring, and Proxy node monitoring.
Instance: Displays the monitoring view of the entire instance, supporting real-time and historical data.
Redis Node: Supports viewing monitoring metrics for a single node and comparing the trend of metrics among nodes. For performance trend comparisons across multiple Redis nodes, see Multi-node Performance Comparison.
Proxy Node: Displays comparative trends of related metrics across Proxy nodes. When selecting the Proxy node dimension, you can select between Aggregate and Node View modes for real-time and historical data.
In the **Aggregate view** mode, all Proxy node information is displayed. Select specific metrics in the upper left corner, and view single metric information for all nodes. Click **Details** to enter the **Node view**.

Node view mode, all monitoring metric information of a single node is displayed.

Step 2: Selecting Monitoring Metrics

Click the metrics dropdown list, check Performance metrics, or use the quick-select options in the upper right corner to select all or none. After selecting metrics, click **Save**.
Note:
Different monitoring dimensions have different monitoring metrics. For details, see Supported Performance Metrics.
Click **Save** to apply the metrics to the currently selected instance.
Click **Save and Apply to All Instances** to apply the selected metrics to all database instances.

Step 3: Viewing Performance Trend Monitoring View

View the performance trend monitoring view based on the viewing conditions set in the above steps.
The performance trend monitoring view can also be viewed in the following ways.

Switch Between Real-time or Historical Views

Click **Real-Time** or **Historical** to view the corresponding real-time performance trend and historical performance trend.
Real-time performance trend view: Users can view the performance trend of the instance. Auto-refresh is enabled by default, with a refresh frequency of 5 seconds. You can select 5 seconds, 10 seconds, 15 seconds, or turn off auto-refresh.
Historical performance trend view: By selecting different time periods, the performance trend monitoring view of the selected period will be displayed. You can switch between the past 1 hour, past 3 hours, past 24 hours, past 7 days, and custom time periods.

Click **Add Time Comparison**, and select the desired comparison time period to view the time comparison of multiple performance metrics.

Multi-node Performance Metric Comparison

Currently, only multi-node performance metric comparisons for Redis node monitoring dimensions are supported.
1. At the top of the page, click **Multi-Node Performance Comparison**.

2. In the pop-up window on the right, click **Create Multi-Node Performance Comparison Task**, select the type, monitoring time, and monitoring item, and click **OK**.

3. In the pop-up window on the right, click **View**.

4. View the generated full node comparison chart.

Enabling Chart Linkage

Click the **Chart Interaction** button on the right side of the page to view associated comparisons across multiple monitoring metrics. Hover over any data point in a monitoring chart to display the data at the same time in other monitoring views. Click to fix the data display, and click **Deselect the Time Point** to cancel.

Displaying Monitoring Metric Data in Chart Format

On the right side of the page, click **Show Statistics** to display the maximum, minimum, and average statistics of each monitoring metric in table format below the global metrics monitoring chart.

©2013-2025 Tencent Cloud International Pte. Ltd.

At the top right of the single metric monitoring chart, click

to display the maximum, minimum, and average statistics of that monitoring metric in table format.

Custom Monitoring Metric Comparison Analysis

At the top right of any monitoring view, click

to add other types of monitoring metrics for comparison view and analysis.

Switching between Single-Column and Double-Column Display Modes for the Monitoring View

Click

on the right of chart linkage in the top right to switch between single-column mode and double-column mode. Refer to the diagram below for single-column mode.

Dragging and Dropping Monitoring View Freely

You can drag and drop the monitoring views freely. Click the border of a monitoring view with the mouse pointer to drag it. You can adjust the order of the monitoring views flexibly according to the Ops scenario, making it easy for efficient viewing and analysis.

Fine-Grained View of Monitoring Chart

Drag the mouse pointer to select the time period to zoom in on a single monitoring chart for a fine-grained trend display. Click **Reset** to restore.

Zooming in Monitoring View

Drag the icon at the bottom right of any monitoring chart to stretch and zoom in the image freely, making it easier to display the trend chart of the metrics more clearly.

©2013-2025 Tencent Cloud International Pte. Ltd.

Real-Time Session

Last updated：2025-04-17 17:42:04

The real-time session feature focuses on two key metrics: the CPU utilization of the database instance's Proxy node and the number of client connections. It dynamically displays the trends of these metrics while continuously tracking data such as database sessions, access sources, and active connections. Additionally, it supports killing sessions for the current Proxy or all Proxy nodes.
Through real-time sessions, Ops and management personnel can quickly identify the CPU utilization of the current session and efficiently locate logic issues about database session connections that are difficult to detect manually.

Viewing Real-Time Session Statistics

1. Log in to the DBbrain Console.
2. In the left sidebar, choose **Performance Optimization.**
3. At the top of the page, select Database Type as Redis, specify instance ID, and select **Real-Time Session** tab.
4. From the dropdown list in the upper left corner of the **Performance Monitoring** trend chart, you can select the **CPU Utilization** trend chart or **Connections** trend chart to select the **Proxy ID** to analyze.
Select the page refresh rate. Options include 5 seconds, 15 seconds, and 30 seconds, with a default value of **5 seconds**. You can also stop refreshing.

5. View detailed real-time session data.
In the **Performance Monitoring** area, you can view the number of connections and CPU utilization trends of the selected Proxy node.

In the **Session Statistics** area, you can view the statistics of the current access sources, total connections, and active connections of the selected proxy node.

Killing Sessions

When you need to clear the client connections of the current Proxy or all Proxies of the selected Redis instance, restart, or clear unfinished tasks, use the Kill Session feature to forcibly close all client connections.
This operation is typically used in emergencies, such as long-term blocking or deadlocks, or when a system restart is required to recover data.
Note:

©2013-2025 Tencent Cloud International Pte. Ltd.

Killing sessions will interrupt ongoing operations and may cause data loss. Use with caution. Before using, back up data and assess risks.

Killing Current Proxy

At the top of the page, click **Kill current Proxy**. In the pop-up dialog box, click **OK.**

Killing All Proxies

At the top of the page, click **Kill all Proxies**. In the pop-up dialog box, click**OK.**

©2013-2025 Tencent Cloud International Pte. Ltd.

Slow Log Analysis

Last updated：2025-04-17 17:34:05

Redis's slow log analysis is different from MySQL and TDSQL-C. Redis slow logs are recorded separately for the instance and Proxy dimensions.
In the instance (Redis database instance) dimension, you can view detailed information such as CPU utilization, the number of slow queries, segmented time consumption statistics of logs, and a comprehensive list of slow logs.
In the Proxy (middleware cluster node) dimension, you can view the Proxy's slow log statistics, segmented time consumption statistics, and detailed slow log lists.

Directions

1. Log in to the DBbrain Console.
2. In the left sidebar, choose **Performance Optimization**.
3. At the top of the page, select Database Type as Redis, specify instance ID, and select **Slow Log Analysis** tab.
4. Select to view instance-level or Proxy node slow logs.
Click **Instance** to view the instance dimension's slow log statistics trend chart.
Click **Proxy Node**, and select the Proxy ID to analyze from the dropdown list. You can select the Proxy ID to analyze based on the trend chart of CPU utilization or slow log quantity change.

5. At the top of the page, select the time period. Options include today, last 5 minutes, last 10 minutes, last 1 hour, last 3 hours, last 24 hours, last 3 days, and custom time periods.
6. View the statistical trend chart of slow logs, segmented time consumption of slow logs, and slow log list data.
Slow Log Statistics: Click a single time range or drag to select multiple time ranges in the slow log statistics chart to view the slow log statistics in the corresponding time ranges.
Slow Query Statistics: This section displays the distribution of the overall time consumption within the selected period. The horizontal axis represents the percentage of slow logs, and the vertical axis shows the statistical period. Hover over a specific period to view the percentage of slow logs for that time.
Slow Log List: Click to view the analysis and statistics details.
If this instance has slow SQL in the selected time range, the slow log statistics will display the time points and the number of slow SQL statements in blue bar charts. Click the bar chart to view all corresponding slow SQL information (aggregated by templates) in the slow log list below. On the right, the segmented time consumption statistics will display the time consumption distribution of SQL statements in the corresponding period.

7. (Optional) In the upper right corner of the page, click **Monitoring Details** to view detailed monitoring metrics.

©2013-2025 Tencent Cloud International Pte. Ltd.

7.1 Select monitoring metrics. For better visualization, limit the selected metric units to two or fewer. For a description of the optional metrics, see Supported Monitoring Metrics.
7.2 Select a time range, with options including the last 1 hour, last 3 hours, last 24 hours, last 7 days, and custom time periods.

Click **Add Time Comparison**, enter the comparison period, and compare the monitoring metrics between two different time periods.

8. In the slow log list, click the aggregated command template, or click **View** in the row where the command template resides. A popup on the right displays the specific analysis and statistical data of the SQL.

On the **Analysis** page, you can view command templates, sample commands, optimization suggestions, and descriptions.
On the **Statistics** page, you can conduct a horizontal analysis of the specific reasons for the slow SQL using metrics such as total time consumption percentage, time distribution, and the access percentage of source IP addresses (only shown for proxy nodes), and make corresponding optimizations.
9. Export slow log analysis data.
On the right side of the slow log list, click

to export the slow log analysis data in .csv format to your local system for easy review.

Memory Analysis (Big Key Analysis)

Last updated：2025-04-17 17:43:06

In TencentDB for Redis cluster mode, if slot sharding is uneven, data and query skew may occur. Some Redis nodes with Large Keys may occupy more memory and network interface resources, causing Redis congestion.
Memory analysis primarily focuses on analyzing Large Keys stored in the database. It dynamically displays changes in the instance's memory utilization trends and provides real-time statistics for the top 100 Large Keys in terms of memory usage, element count, length, and expiration time. This helps Ops personnel quickly identify Large Keys, split or expire them, and optimize database performance promptly, avoiding service performance degradation, memory shortages, and potential business disruptions caused by Large Keys.

Memory Analysis Usage Instructions

Memory analysis provides **Periodic Big Key Analysis** and **Ad Hoc Analysis of Big Key**:
Periodic Big Key Analysis: When using Large Key Analysis for the first time, enable the Large Key Analysis feature on the **Instance Management** page. Once the feature is enabled, the system will automatically initiate a Large Key Analysis task the next day, with the results displayed in **Memory Analysis** > **Big Key Analysis**. Subsequently, a routine analysis task will be performed daily, and the data results will be updated accordingly.
For detailed steps on enabling the Large Key Analysis feature and view analysis results, see Enabling Large Key Analysis and Viewing Large Key Analysis Results.
Ad Hoc Analysis of Big Key: After a real-time Large Key Analysis task is created, a backup is immediately generated to collect the latest data. The analysis results can be viewed on the **Ad Hoc Analysis of Big Key** tab task list or on the **Big Key Analysis** tab. The results are retained for 30 days by default.
If the Large Key Analysis feature is not enabled on the **Instance Management** page before the real-time Large Key Analysis task is created, the data under the **Big Key Analysis** tab will be displayed for the first time.
For detailed steps on creating real-time Large Key Analysis tasks and viewing Large Key Analysis results, see Creating Real-Time Large Key Analysis Tasks and Viewing Large Key Analysis Results.

Limits

Redis instances with storage exceeding 100 GB do not support **Periodic Big Key Analysis**. However, memory analysis can be conducted by creating **Ad Hoc Analysis of Big Key** tasks.

Enabling Large Key Analysis (**Periodic Large Key Analysis**)

1. Log in to the DBbrain Console.
2. Enable the Large Key Analysis feature.

Enabling Instance Large Key Analysis on the Instance Analysis Page

2.1.1 In the left sidebar, select **Instance Management**, and select the Redis instance.

2.1.2 Enable the Large Key Analysis feature using one of the following three methods.

Method 1: Select the instances for which you want to enable Large Key Analysis in the instance list, and click **batch setting** at the top-left corner of the page.

Method 2: In the **Status** column of the target instance, click

.

Method 3: In the **Operation** column of the target instance, click **Configuration**.

Enabling Instance Large Key Analysis on the Memory Analysis Page

2.1.1. In the left sidebar, select **Performance Optimization**.

2.1.3 Select **Memory Analysis** > **Big Key Analysis**, and select the Redis Data Type and Instance ID.

2.1.4 Click **regular analysis setting** in the top-right corner of the page.

3. In the pop-up dialog box, enable **Top 100 Big Key Regular Analysis**, **Separators**, and click **OK**.

Note:

After you enable **Top 100 Big Key Regular Analysis**, the **Performance Optimization** > **Memory Analysis** > **Big Key Analysis** tab will display analysis results in three dimensions: **Top 100 Big Keys (by MEM Usage)**, **Top 100 Big Keys (by Element Quantity)**, and **Top 100 Key Prefixes**.

After a delimiter is specified,**Top 100 Key Prefixes** statistics are based on the key prefixes split by the specified delimiter and sorted by memory usage.

Creating Real-Time Large Key Analysis Task

1. Log in to the DBbrain Console.
2. In the left sidebar, choose **Performance Optimization.**
3. At the top of the page, select a Redis instance.
4. Select **Memory Analysis** tab, and select **Ad Hoc Analysis of Big Key**.

5. Click **Create Task**, select a delimiter and shard ID in the pop-up dialog box, and click **OK**.

You can click **View All Nodes** in the **Operation** column to view all node IDs.

Once the task is created, DBbrain will automatically generate a backup and perform automated analysis.

6. In the task list, when the task progress reaches **100%**, click **View** in the **Operation** column to view the analysis results in a pop-up on the right.

The task analysis results display the **Top 100 Big Keys (by MEM Usage)**, **Top 100 Big Keys (by Element Quantity)**, and **Top 100 Key Prefixes** in three dimensions, and allow viewing results from both instance and shard dimensions.

Note:

The analysis results generated by the real-time Large Key Analysis task are also available in the **Large Key Analysis** tab. For more details, see Viewing Large Key Analysis Results.

The **Operation** column in the task list also supports the following operations:

Download task analysis results: Click **Download** to export the Top 100 Large Key Analysis results in .csv format.

Delete real-time Big Key Analysis tasks:

Single deletion: Click **Delete**, and in the pop-up dialog box, click **OK**.

Batch deletion: Select tasks in the task list, click **Delete** at the top of the list, and in the pop-up dialog box, click **OK**.

Viewing the Large Key Analysis Result

1. Log in to the DBbrain Console.
2. In the left sidebar, choose **Performance Optimization.**
3. At the top of the page, select a Redis instance.
4. Select **Memory Analysis** tab, then select **Big Key Analysis** tab.
5. View the Large Key Analysis results, including a MEM Utilization (Last 30 Days) trend chart and Top 100 Big Key statistics.

Note:

In the **MEM Utilization (Last 30 Days)** trend chart, the memory utilization trends of the instance over the last 30 days are displayed by default. Click a specific date on the horizontal axis to fix the timeline. The Top 100 Large Key list will then dynamically show the Large Key information for that day, allowing you to quickly identify keys consuming high memory on that date.

MEM Utilization (Last 30 Days)

Supports viewing historical memory usage over 30 days by instance or shard (for multi-shard instances only).

Select a specific time range on the timeline to zoom in and view memory usage trends for that period.

Top 100 Big Keys

In the **Data Type** dropdown menu, select a data storage type to view Top 100 Large Key information, including memory usage, element count, maximum element length, average element length, and expiration time.

Top 100 Big Keys (by MEM Usage): Lists the Top 100 Large Keys ranked by memory usage in descending order.

Top 100 Big Keys (by Element Quantity): Lists the Top 100 Large Keys ranked by element count in descending order.

Top 100 Key Prefixes: Lists the Top 100 Key Prefixes ranked by memory usage in descending order.

©2013-2025 Tencent Cloud International Pte. Ltd.

Latency Analysis

Last updated：2025-04-17 17:43:50

The Redis latency analysis feature helps you understand the database latency in real time. Through latency analysis, you can quickly view the total requests, CPU usage, and history of the current instance and locate time-consuming commands, time-consuming command execution time, overall latency distribution, and access command hits.

Viewing the latency analysis result

1. Log in to the DBbrain Console.
2. In the left sidebar, choose **Performance Optimization**.
3. At the top of the page, select the corresponding database and instance ID, then select **Latency Analysis** > **Latency Analysis** tab.

4. On the Latency Analysis page, you can switch between real-time and historical views to check latency analysis data.
Real-time view: Displays the analysis results at each time point in real-time.
Historical view: Displays the analysis results in the last 30 minutes, last 6 hours, last 24 hours, or a custom time range.

The latency distribution displays the percentage of data across different latency ranges, allowing users to quickly perceive the overall latency of their business operations. The Command area displays time-consuming commands and the the number of hits.

©2013-2025 Tencent Cloud International Pte. Ltd.

Latency Analysis (Command Word Analysis)

Last updated：2025-04-17 17:44:55

In addition to the analysis of big keys and hot keys, DBbrain also provides Redis command word analysis to help you better understand the current conditions of your database.

Viewing command word analysis

1. Log in to the DBbrain Console.
2. In the left sidebar, choose **Performance Optimization**.
3. At the top of the page, select the corresponding database and instance ID, then select **Latency Analysis** > **Command Word Analysis** tab.

4. On the Command Word Analysis page, select a real-time or historical time range.
Real-time view: Displays the analysis results at each time point in real-time.
Historical view: Displays the analysis results in the last 30 minutes, last 6 hours, last 24 hours, or a custom time range.

5. Select a command type, click **OK** to view the command word analysis.

©2013-2025 Tencent Cloud International Pte. Ltd.

Latency Analysis (Hot Key Analysis)

Last updated：2025-04-17 17:45:45

In Redis, frequently accessed keys are called hot keys. When a Redis database receives a lot of requests to access a hot key, the traffic gets too concentrated and reaches the upper limit of the physical ENI, which will cause problems or even downtime of the Redis service.
With DBbrain's hot key analysis feature, you can find hot keys quickly to optimize the service accordingly.

Viewing Hot Key Analysis

1. Log in to the DBbrain Console.
2. In the left sidebar, choose **Performance Optimization**.
3. At the top of the page, select the corresponding database and instance ID, then select **Latency Analysis** > **Hot Key Analysis** tab.

4. On the Hot Key Analysis page, select the data type and time period. The time period can be switched between real-time and historical views to analyze the data.
Data type: Supports All, string, list, set, hash, sortedset, and stream.
Real-time view: Displays the analysis results at each time point in real-time.
Historical view: Displays the analysis results in the last hour, last 3 hours, last 24 hours, last 7 days, or a custom time range.

Select the Redis Node to be analyzed and the List Item Limit (supporting a limit of 50 items, and a limit of 100 items), and view the analysis data.

©2013-2025 Tencent Cloud International Pte. Ltd.

MongoDB Performance Optimization Exception Diagnosis

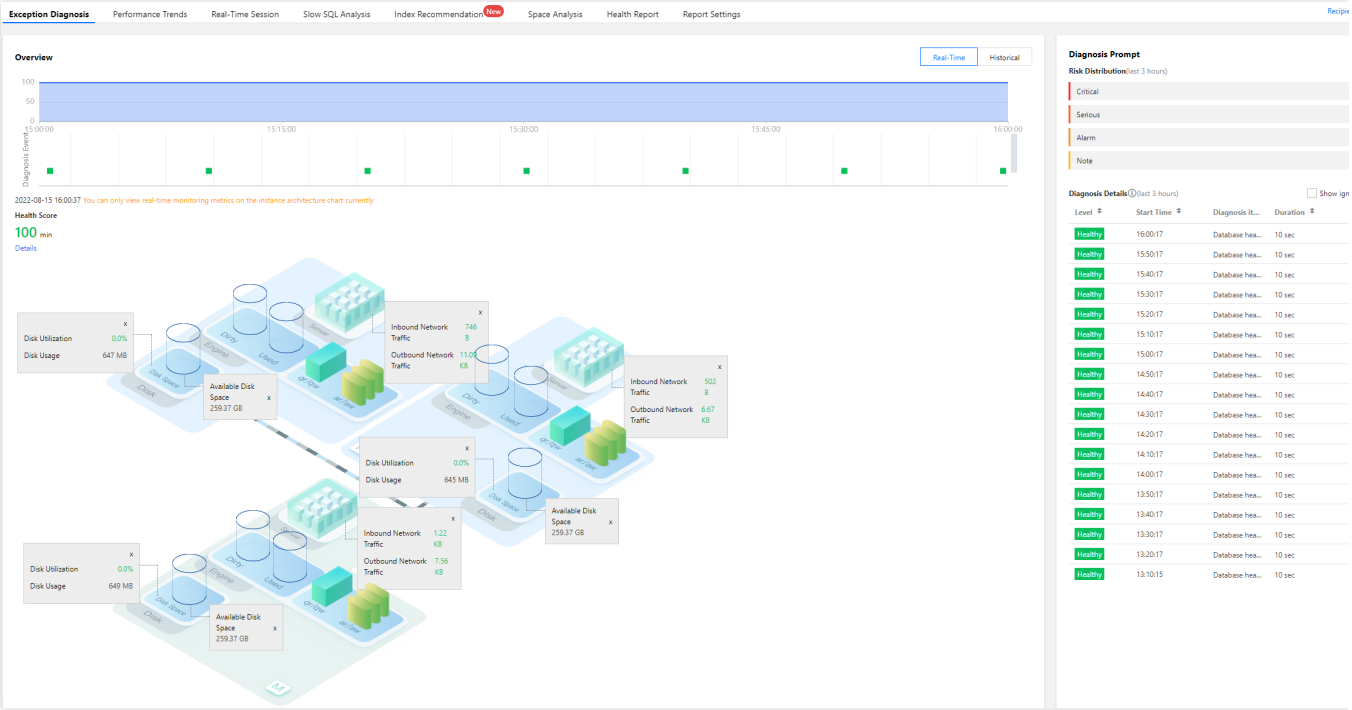
Last updated：2022-08-15 16:23:27

Feature description

The exception diagnosis feature provides you with real-time performance monitoring, health inspections, and failure diagnosis, so that you can intuitively know the real-time operation status of database instances, locate newly appeared performance exceptions in real time.

Overview

Log in to the DBbrain console and select **Performance Optimization** on the left sidebar. On the displayed page, select a database type and an instance at the top, and select the **Exception Diagnosis** tab.



Viewing the monitoring overview

The **Overview** section displays the database's overall health score, exception diagnosis event timeline, topology, and other information.
At the top of the **Overview** section, you can select **Real-Time** or **Historical** to view corresponding statistics.
On the timeline of **Diagnosis Event**, you can view the occurrence time point of each diagnosis event. Hover over the timeline and scroll up or down the mouse wheel to zoom it in or out.
The **Health Score** section displays the instance's **CPU Utilization**, **Memory Utilization**, **Connection Utilization**, and **Read Request Hit Rate**. AI-based health scores can reflect the actual status of your databases.

Viewing diagnosis information

Diagnosis events are displayed in the following risk levels: **Healthy**, **Note**, **Alarm**, **Serious**, and **Critical**. DBbrain performs health inspections on the instance once every ten minutes.
1. The **Diagnosis Prompt** section displays the **Distribution of Risks per Risk Level** of events.
2. In the **Diagnosis Details** list, click an event to enter the **Event Details** page.
3. In **Event Details**, view the **Description** of the event.
Event Details: Include the **Diagnosis items**, **Time Range**, **Risk Level**, **Duration**, and **Overview**.
Description: Includes problem snapshots and performance trends of the exception or health inspection event.
4. Ignore/Unignore an alarm.
You can click **Ignore** to ignore an alarm. Then, other diagnosis item alarms of the instance generated by the same root cause will also be ignored. Ignored alarms will be grayed out.
Note:
Only diagnosis item alarms that are not generated by health inspections can be ignored or unignored.
You can click **Unignore** to unignore an ignored alarm. Then, other diagnosis item alarms of the instance generated by the same root cause will also be unignored. Ignored diagnosis items are not displayed by default.
In the **Diagnosis Prompt** section, hover over an alarm to display the **Ignore** button and click it. You can click **Ignore** or **Unignore** on the row of an alarm to ignore or unignore it and other alarms generated by the same root cause.
Or, go to the **Event Details** page and click **Ignore** or **Unignore** in the top-right corner.

Viewing SQL and slow SQL information

The **Real-Time SQL** or **Historical SQL** section displays the overall information and distribution of the number of requests made to the instance, including `aggregate`, `command`, `count`, `delete`, `getMore`, `insert`, `read`, and `update` requests.
The **Real-Time Slow SQL** or **Historical Slow SQL** section displays the trends of slow SQL requests and CPU utilization.

Performance Trends

Last updated：2022-08-13 16:25:49

Feature description

The performance trends feature provides the following real-time monitoring information of your MongoDB database to help you locate time-consuming commands and their execution time and overall latency distribution.

Resource Monitoring: **CPU**, **MEM**, **Storage Space**, and **Traffic**.

Request Statistics: **Request Latency Distribution**, **Request Type Distribution**, **Request Type with 10-50 ms Latency**, **Request Type with 50-100 ms Latency**, **Request Type with over 100 ms Latency**, **TTL Request Statistics**, **Active Sessions**, and **Request Latency**.

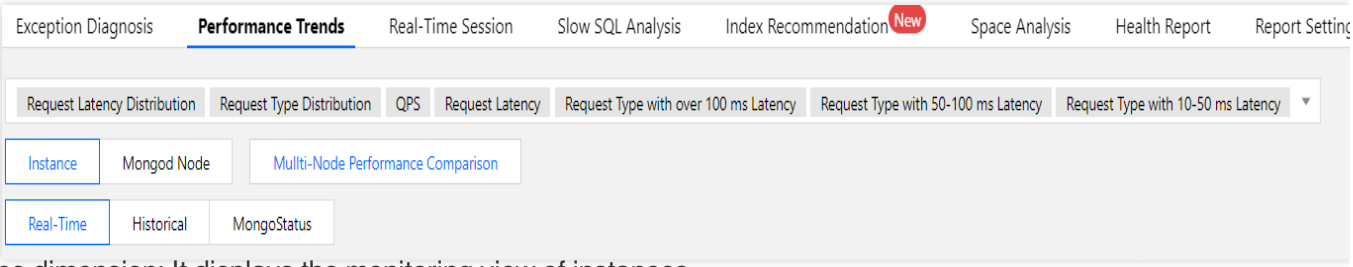
MongoDB Primary-Secondary Replication: **Secondary Node Replication Delay** and **Oplog Retention Period**.

Storage Engine: **Cache**, **qr**/**qw**, and **ar**/**aw**.

Viewing performance trends

1. Log in to the DBbrain console and select **Performance Optimization** on the left sidebar. On the displayed page, select a database type and an instance at the top, and select the **Performance Trends** tab.
2. Set the monitoring dimension and metrics of performance trends.

Monitoring dimension: You can select instance monitoring or node monitoring.



Instance dimension: It displays the monitoring view of instances.

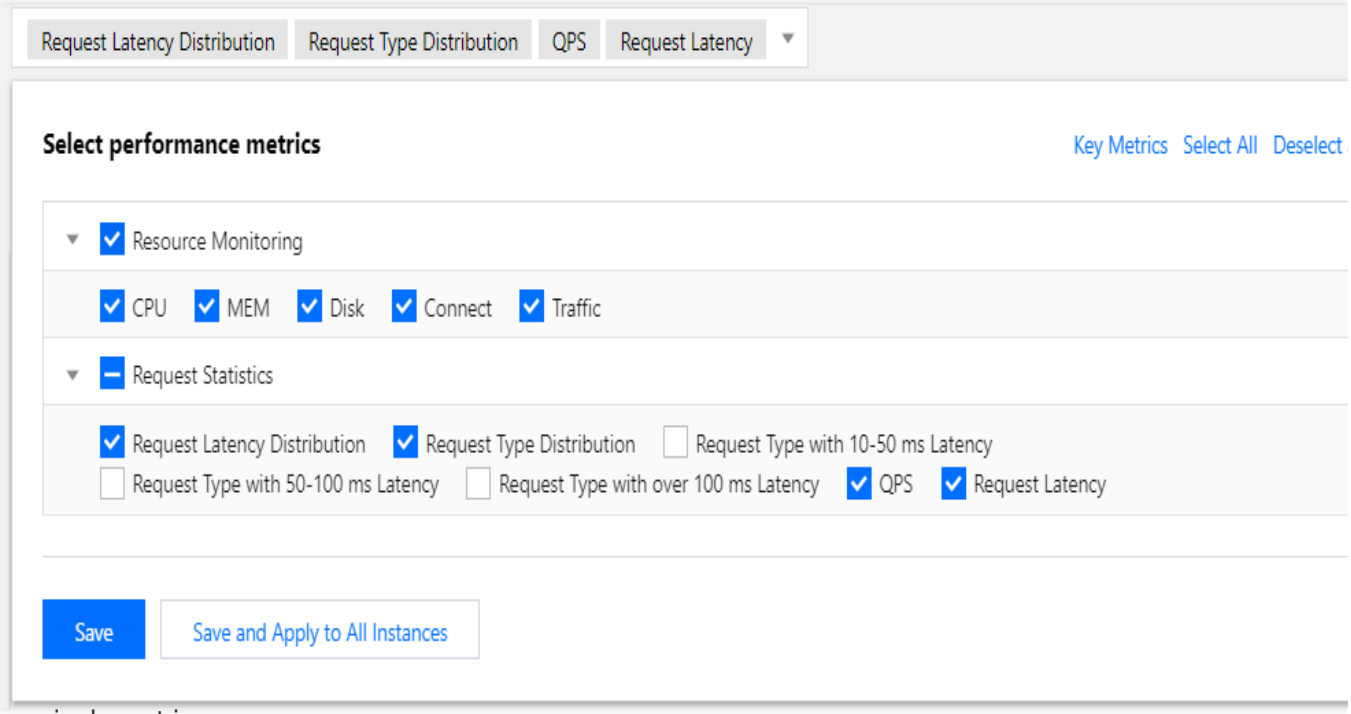
Node dimension: It displays the comparison trend views of relevant metrics on each MongoDB node.

Metric categories: Include **CPU**, **MEM**, **Disk**, **Connection**, **Traffic**, and **Request Statistics**.

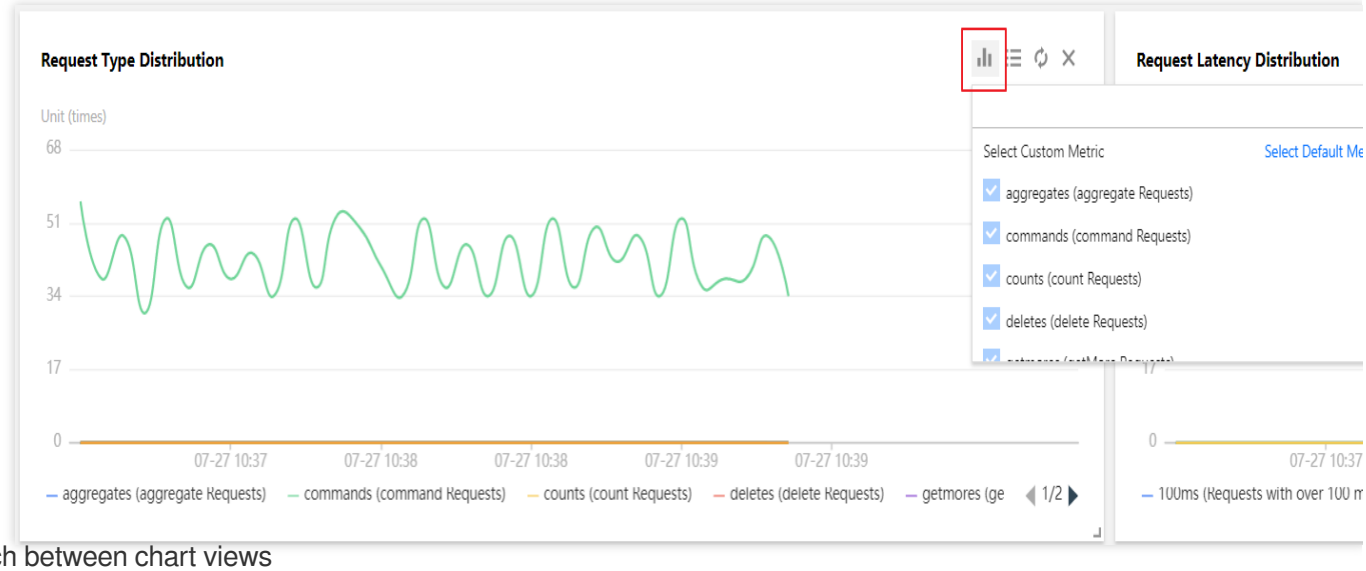
Select performance metrics: You can select all metrics, custom metrics, and various views.

Filter global metrics

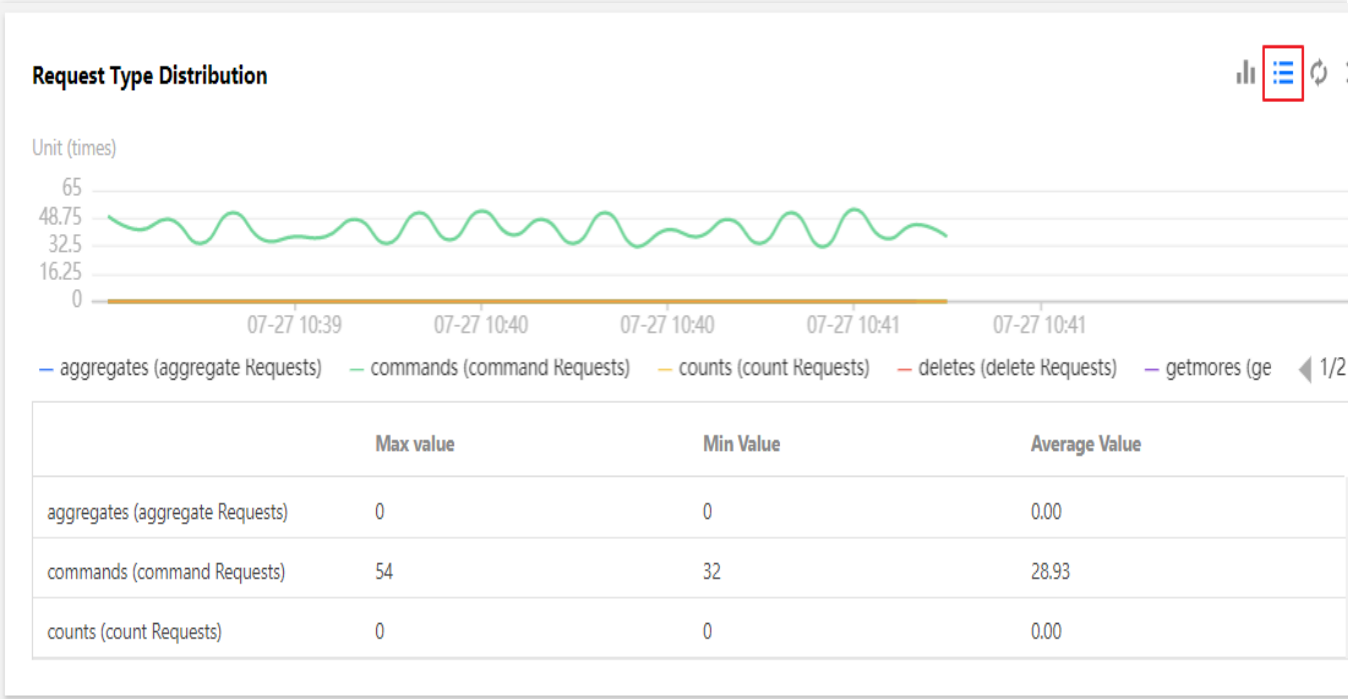
©2013-2025 Tencent Cloud International Pte. Ltd.



Filter one single metric



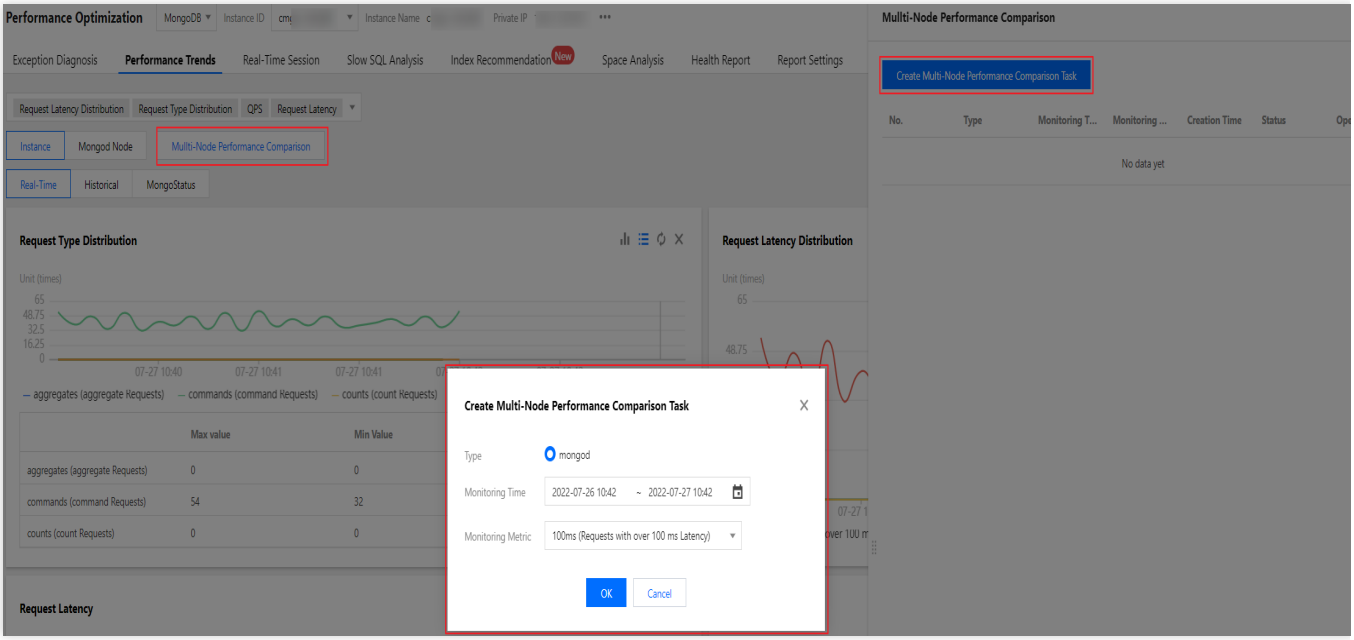
Switch between chart views



3. Switch between the **Real-Time** and **Historical** views.

DBbrain allows you to switch between real-time and historical data. Based on the selected time view, different granularities are provided. Single metric view and comparison view are also available.

Customize the multi-node comparison chart.

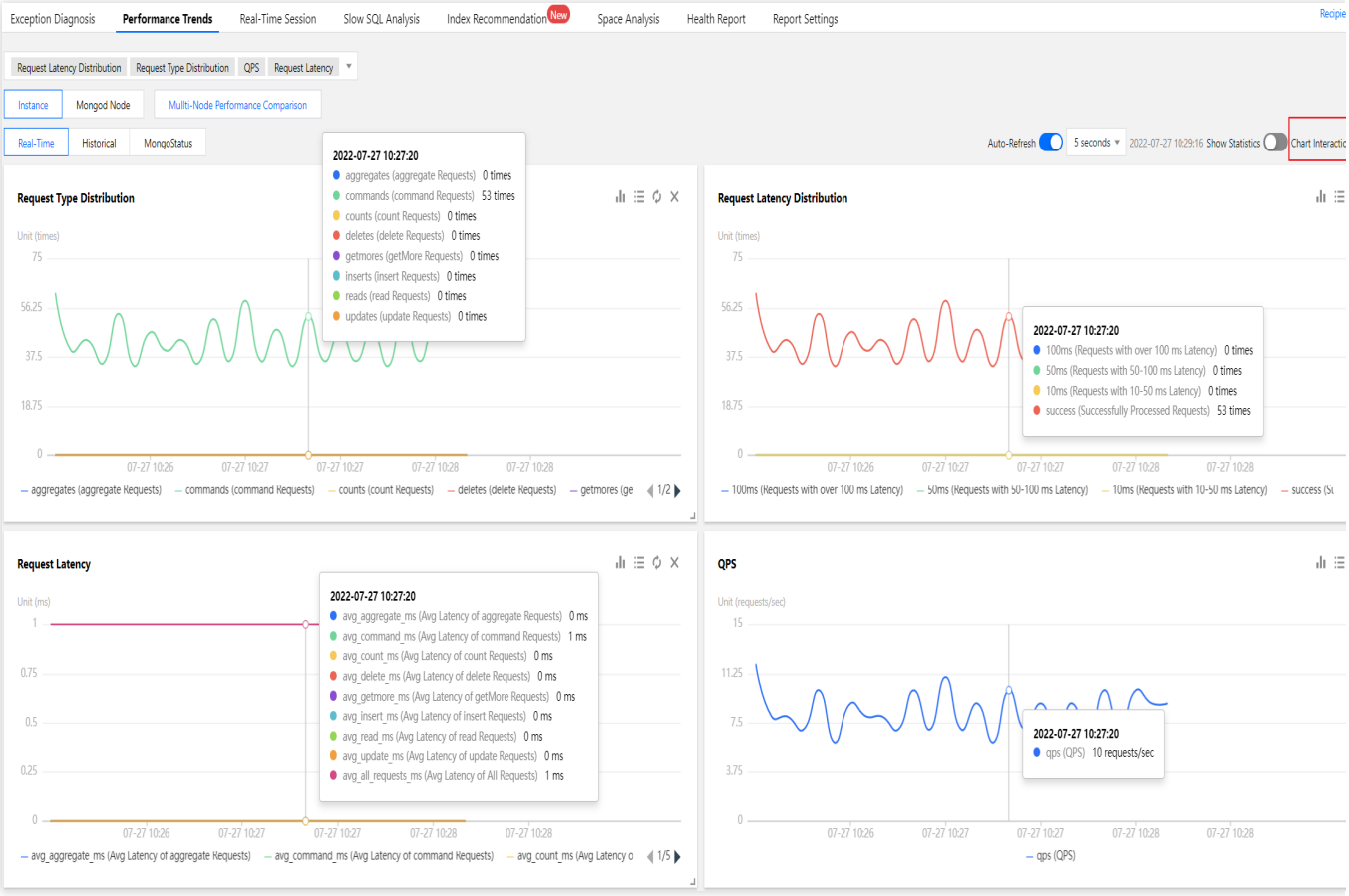


4. Enable chart interaction.

For one single instance, node, or proxy, you can view relevant metric trend comparison, add custom metrics, and view the performance metric trend comparison by time.

After you enable chart interaction, when you hover over a data point in any monitoring view, the data at the same time

point will be displayed in other monitoring views. Click the data point to pin it for display. To unpin it, click **Deselect the Time Point**.



5. Switch between the one-column and two-column modes, drag a monitoring view, or zoom in a monitoring view.

Switching between the one-column and two-column modes: Click the button on the right of **Chart Interaction** in the top-right corner to switch.

Dragging a monitoring view: Click the border of a monitoring view to drag it to the desired position.

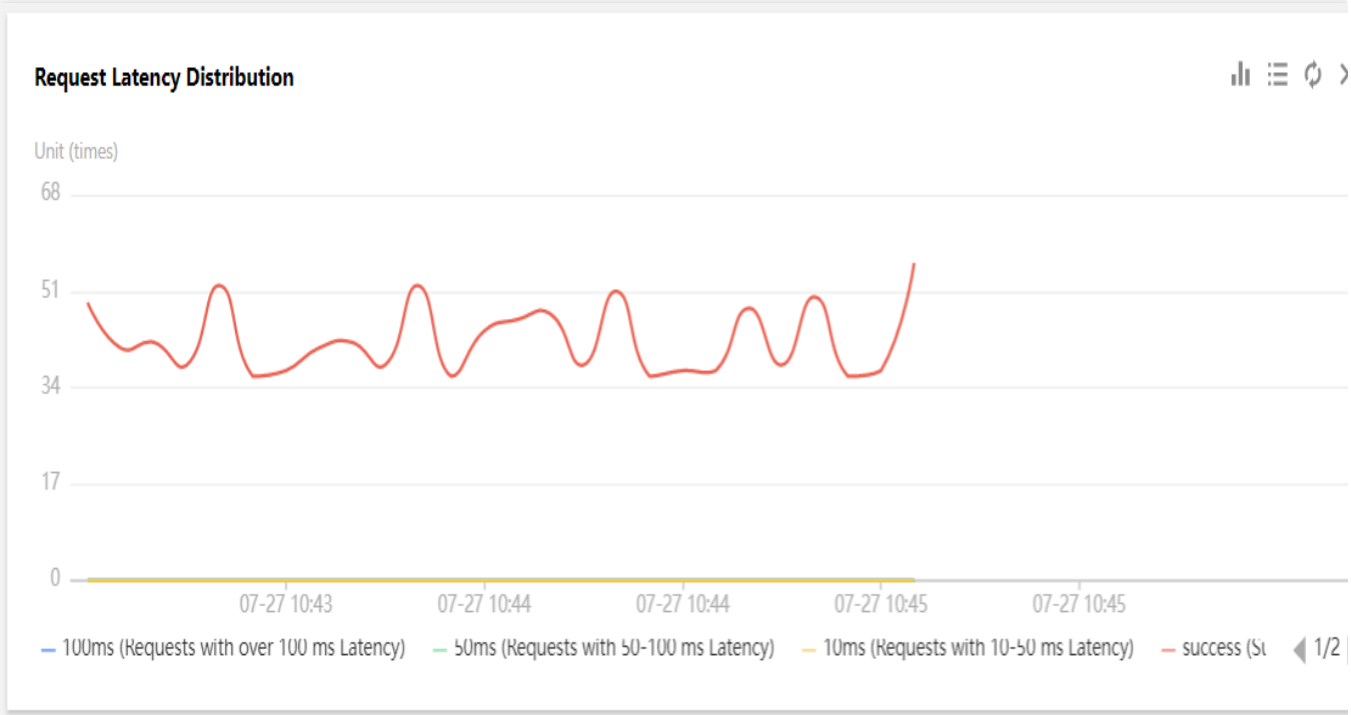
Zooming in a monitoring view: Drag the icon in the bottom-right corner of a monitoring view to zoom it in for fine-grained display of the trend of one single performance metric.

6. Check the status of the MongoDB node. For more information, see MongoStatus and MongoTop.

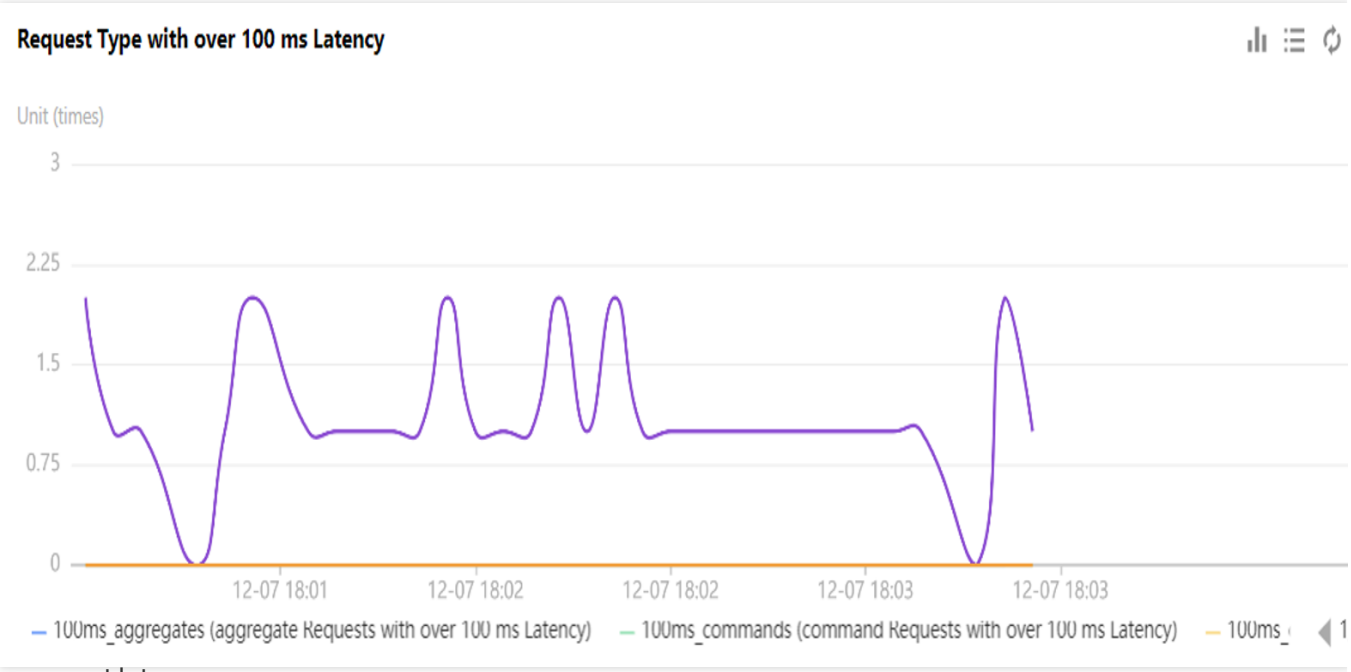
7. View the data of latency analysis.

Below is a sample performance trends query result. Click a data point in the chart to display the metric details.

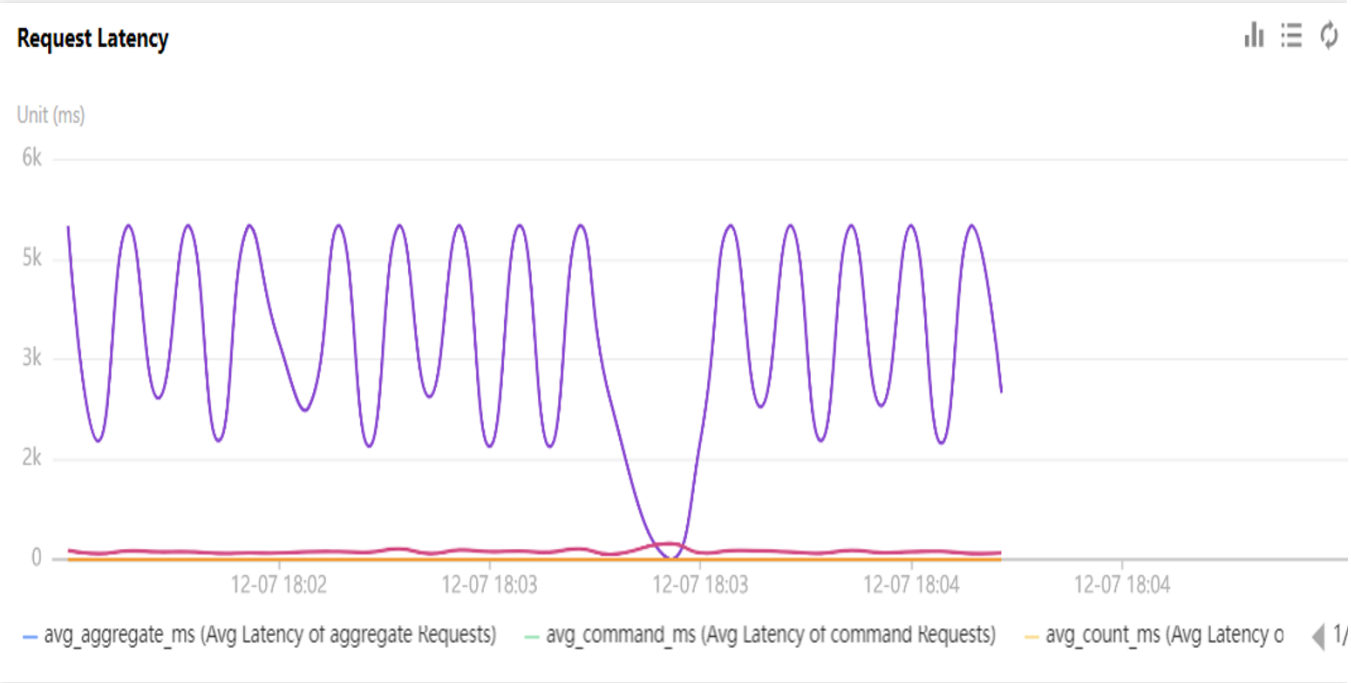
Sample request latency distribution:



Sample distribution of the request type with over 100 ms latency:



Sample request latency:


Request Latency
Unit (ms)
6k
5k
3k
2k
0
12-07 18:02
12-07 18:03
12-07 18:03
12-07 18:04
12-07 18:04
avg_aggregate_ms (Avg Latency of aggregate Requests)
avg_command_ms (Avg Latency of command Requests)
avg_count_ms (Avg Latency o

Slow SQL Analysis

Last updated：2025-04-14 21:22:29

The slow SQL analysis feature calculates, samples, and aggregates slow SQL statements from three dimensions: instance, mongod node, and mongos node. It aggregates slow SQL statistics based on command template and command space, including the number of executions, average execution time, resource consumption (such as CPU), sizes of scan, and result sets. It also supports viewing slow SQL details to help users quickly locate and resolve slow SQL issues.

Viewing Slow SQL Statistics

1. Log in to the DBbrain Console.
2. In the left sidebar, choose **Performance Optimization**, then select the **Slow SQL Analysis** tab.
3. At the top of the page, select the MongoDB database type and the instance ID to be analyzed.

4. Select the **Statistics** tab.

5. Select the view dimension.
Instance dimension: Click **Instance** to view the slow log statistics trend chart, segmented time consumption statistics of slow logs, and slow log list for the instance dimension.
Mongod node dimension: Click **Mongod Node**, select **Full Node Analysis** or single node name to view the slow log statistics trend chart, segmented time consumption statistics of slow logs, and slow log list for the mongod node dimension.
Mongos node dimension: Click **Mongos Node**, select **Full Node Analysis** or single node name to view the slow log statistics trend chart, segmented time consumption statistics of slow logs, and slow log list for the mongos node dimension.
6. Select a time range. You can select the current day, last 5 minutes, last 10 minutes, last 1 hour, last 3 hours, last 24 hours, last 3 days, or a custom period (up to 30 days ago and with a maximum span of 3 days).
7. View the slow log statistics, segmented time consumption statistics of slow logs, and slow log list.
Slow log statistics trend chart
Slow log statistics focus on the number of slow queries and the cluster's maximum CPU utilization. It can quickly identify CPU usage during periods with a large number of slow queries in the selected time range. This avoids high CPU utilization caused by excessively slow queries, preventing computer lag or no response.
In the trend chart, click the time period with slow logs (which means the histogram), and the view will display the time points and number of slow SQLs generated.

©2013-2025 Tencent Cloud International Pte. Ltd.

Click a certain time period with slow logs or drag the mouse pointer to select multiple time periods. The segmented time consumption statistics and slow log list will display data information for the selected period in a linked display.

Segmented time consumption statistics of slow logs

It displays the overall time consumption distribution of slow logs in the selected period.

Slow log list

It displays the slow logs in the selected time period based on command space aggregation.

7.1.1 Filter slow logs by namespace.

It supports inclusion or exclusion. After selecting, you can enter one or more namespaces. If you select **Include**, the relationship between multiple namespaces is OR, and if you select **Exclude**, the relationship is AND.

7.1.2 View the slow log list. By default, it is sorted in descending order by overall time consumption.

Click a specific aggregated slow log to view the log statistics and details in the pop-up panel on the right.

Statistics tab: It displays the total time consumption, the proportion of scanned lines, the number of scanned lines and the time distribution of this type statement in the selected period.

Details tab: It displays the detailed SQL execution information. For specific operations, see View Slow SQL Details.

Additionally, the slow log list supports the following operations:

Copy or view command templates: Hover over the command template row, and click **Copy** or **View**.

Click the list header to sort parameters in ascending or descending order: It includes execution count, total time, average execution time (s), average scan rows, maximum scan rows, average index scan rows, maximum index scan rows, average return rows, and maximum return rows.

7.1.3 Export slow log list.

In the upper right corner of the slow log list, click

to export all the slow log statistics displayed in the current list. The export format is .csv.

Viewing Slow SQL Details

1. Log in to the DBbrain Console.
2. In the left sidebar, choose **Performance Optimization**, then select the **Slow SQL Analysis** tab.
3. At the top of the page, select the MongoDB database type and the instance ID to be analyzed.
4. Select the **Details** tab.

5. Select a time range. You can select the current day, last 5 minutes, last 10 minutes, last 1 hour, last 3 hours, last 24 hours, last 3 days, or a custom period (up to 30 days ago and with a maximum span of 3 days).
6. (Optional) Filter slow SQL statements by time consumption or namespace.
Time Consumed: It can be greater or less than a specific execution time, or execution time within a particular interval.

Namespace: It can be selected to include or exclude one or more namespaces. If you select **Include**, the relationship between multiple namespaces is OR, and if you select **Exclude**, the relationship is AND.

7. View slow SQL details. By default, the order is sorted in descending order by execution time.

Additionally, the slow SQL list supports the following operations:
Copy or view SQL statements: Hover over the SQL statements, and click **Copy** or **View**.
Click the list header to sort parameters in ascending or descending order: It includes execution time, scanned indexes, returned rows, and scanned rows.
8. Export slow SQL detail data.
Above the slow SQL list, click

to export all slow SQL details of the current list. The export format is .csv.

Related Operations

If you need to view monitoring metrics details on the **Slow SQL Analysis** page or compare the monitoring metrics of two time periods, you can click **Monitoring Details** at the top right of the page.

Select monitoring metrics and the time range, and view the monitoring metric trend chart and list information.

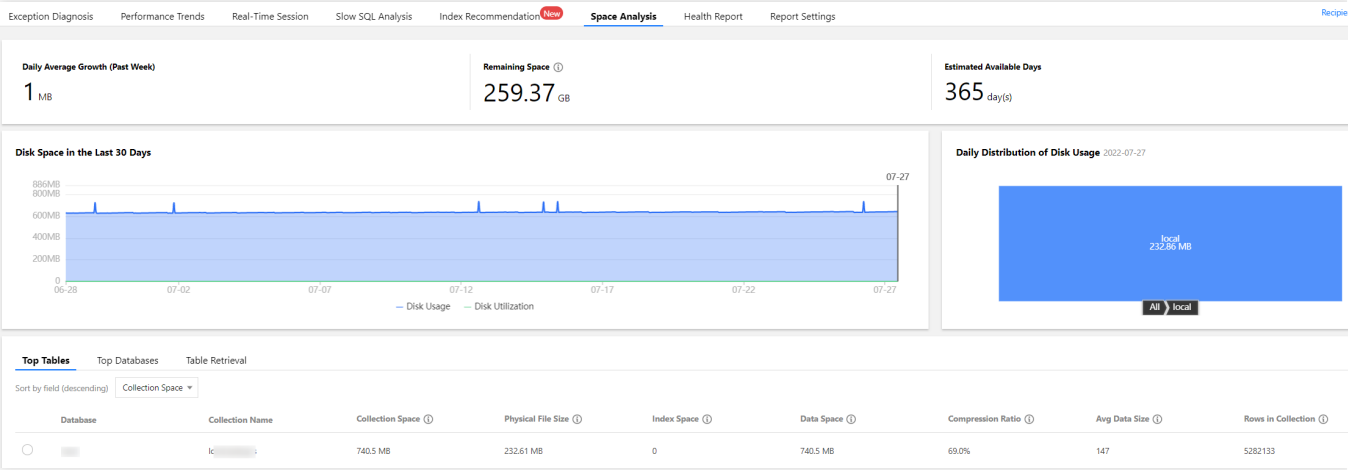
Space Analysis

Last updated：2022-08-13 16:25:49

Feature description

With DBbrain's space analysis feature, you can view the instance space utilization, including the sizes of data and logs, the daily increase in space utilization, the estimated number of available days, and the space used by the tables and databases of the instance.

Overview



Directions

1. Log in to the DBbrain console and select **Performance Optimization** on the left sidebar. On the displayed page, select a database type and an instance at the top, and select the **Space Analysis** tab.
2. Check the disk space.

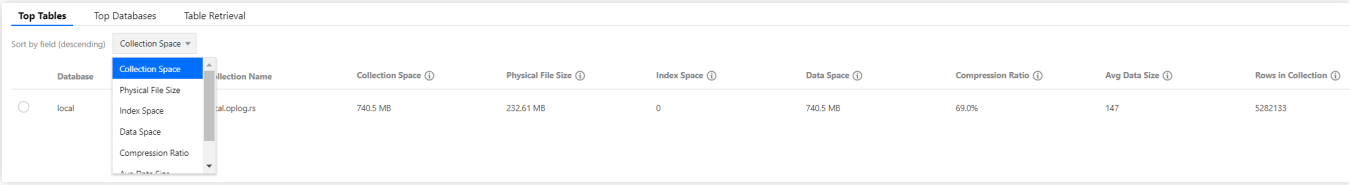
In the upper part of the **Space Analysis** tab, you can view the daily average growth in the past week, remaining disk space, estimated available days, daily distribution of disk usage, and disk space trend in the last 30 days.

For TencentDB for MongoDB, the remaining disk space = purchased disk space - data space.

3. View top tables.

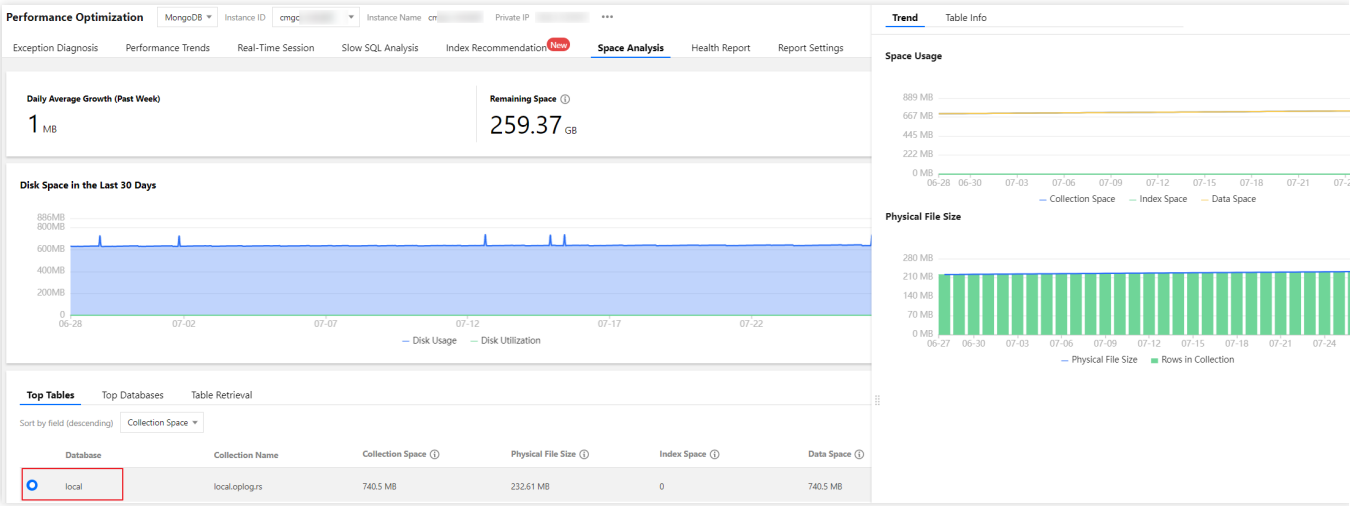
The **Top Tables** section shows the details of the tables that have relatively high space usage. The table list in the section contains columns such as the **Collection Name**, **Collection Space**, **Physical File Size**, **Index Space**,

Data Space, **Compression Ratio**, **Avg Data Size**, and **Rows in Collection**. The tables can be sorted by specified field in descending order. You can view the disk space usage details in this section and perform optimization promptly.



Select a table to further view its **Trend** and **Table Info** on the **Space Analysis** tab.

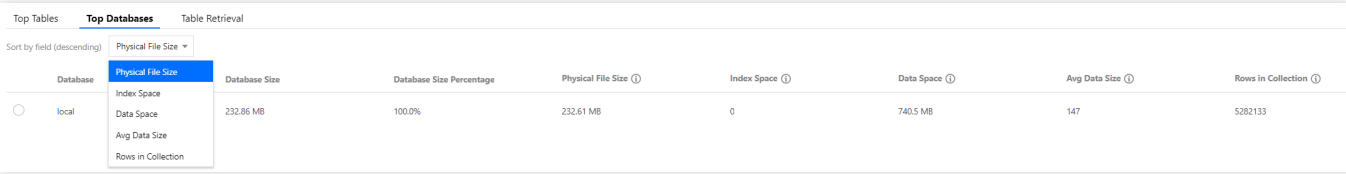
The **Trend** section displays the trends of the **Collection Space**, **Index Space**, and **Data Space** as well as the statistics of the **Physical File Size** and **Rows in Collection**.



The **Table Info** section allows you to locate an index and its details. This makes it easy for you to quickly locate data with a high space usage.

4. View top databases.

The **Top Databases** section shows the details of the databases that have relatively high space usage. The database list in the section contains columns such as the **Physical File Size**, **Index Space**, **Data Space**, **Avg Data Size**, and **Rows in Collection**. The databases can be sorted by specified field in descending order. You can view the disk space usage details in this section and perform optimization promptly.



Select a database to view its statistical trends.

5. View the table retrieval.

Enter a database name and a collection name to view their space statistics.

6. Download the space analysis data.

On the **Top Tables** and **Top Databases** tabs, click the download icon in the top-right corner to download the data in CSV format.

MongoStatus

Last updated：2022-08-13 16:25:49

Overview

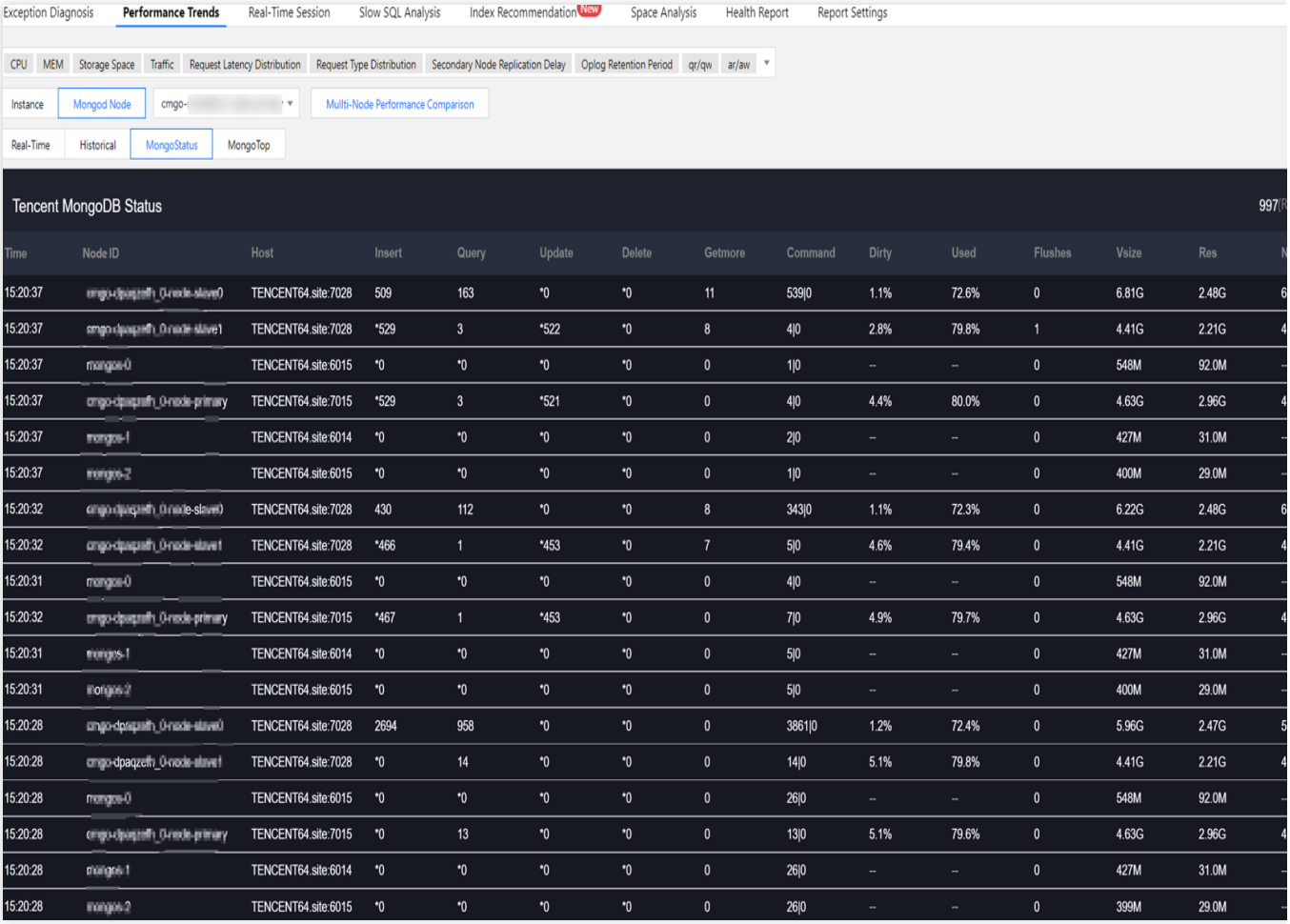
To facilitate daily database Ops, DBbrain provides the TencentDB for MongoDB MongoStatus tool. This tool monitors the MongoDB status at the instance or node level by checking the traffic and storage engine in real time.

Directions

Instance-level MongoDB status

1. Log in to the DBbrain console and select **Performance Optimization** on the left sidebar. On the displayed page, select a database type and an instance at the top, and select the **Performance Trends** tab.
2. Select **Instance** > **MongoStatus**.
3. Click **Pause** in the top-right corner to pause and view the data.

©2013-2025 Tencent Cloud International Pte. Ltd.



Time	Node ID	Host	Insert	Query	Update	Delete	Getmore	Command	Dirty	Used	Flushes	Vsize	Res
15:20:37	[illegible]	TENCENT64.site:7028	509	163	*0	*0	11	539|0	1.1%	72.6%	0	6.81G	2.48G
15:20:37	[illegible]	TENCENT64.site:7028	*529	3	*522	*0	8	4|0	2.8%	79.8%	1	4.41G	2.21G
15:20:37	[illegible]	TENCENT64.site:6015	*0	*0	*0	*0	0	1|0	--	--	0	548M	92.0M
15:20:37	[illegible]	TENCENT64.site:7015	*529	3	*521	*0	0	4|0	4.4%	80.0%	0	4.63G	2.96G
15:20:37	[illegible]	TENCENT64.site:6014	*0	*0	*0	*0	0	2|0	--	--	0	427M	31.0M
15:20:37	[illegible]	TENCENT64.site:6015	*0	*0	*0	*0	0	1|0	--	--	0	400M	29.0M
15:20:32	[illegible]	TENCENT64.site:7028	430	112	*0	*0	8	343|0	1.1%	72.3%	0	6.22G	2.48G
15:20:32	[illegible]	TENCENT64.site:7028	*466	1	*453	*0	7	5|0	4.6%	79.4%	0	4.41G	2.21G
15:20:31	[illegible]	TENCENT64.site:6015	*0	*0	*0	*0	0	4|0	--	--	0	548M	92.0M
15:20:32	[illegible]	TENCENT64.site:7015	*467	1	*453	*0	0	7|0	4.9%	79.7%	0	4.63G	2.96G
15:20:31	[illegible]	TENCENT64.site:6014	*0	*0	*0	*0	0	5|0	--	--	0	427M	31.0M
15:20:31	[illegible]	TENCENT64.site:6015	*0	*0	*0	*0	0	5|0	--	--	0	400M	29.0M
15:20:28	[illegible]	TENCENT64.site:7028	2694	958	*0	*0	0	3861|0	1.2%	72.4%	0	5.96G	2.47G
15:20:28	[illegible]	TENCENT64.site:7028	*0	14	*0	*0	0	14|0	5.1%	79.8%	0	4.41G	2.21G
15:20:28	[illegible]	TENCENT64.site:6015	*0	*0	*0	*0	0	26|0	--	--	0	548M	92.0M
15:20:28	[illegible]	TENCENT64.site:7015	*0	13	*0	*0	0	13|0	5.1%	79.6%	0	4.63G	2.96G
15:20:28	[illegible]	TENCENT64.site:6014	*0	*0	*0	*0	0	26|0	--	--	0	427M	31.0M
15:20:28	[illegible]	TENCENT64.site:6015	*0	*0	*0	*0	0	26|0	--	--	0	399M	29.0M

Node-level MongoDB status

1. Log in to the DBbrain console and select **Performance Optimization** on the left sidebar. On the displayed page, select a database type and an instance at the top, and select the **Performance Trends** tab.
2. Select **Mongod Node** > **MongoStatus**.
3. Select a node in the drop-down list.
4. Click **Pause** in the top-right corner to pause and view the data.

MongoStatus monitoring metrics

MongoStatus fields are as described below:

Monitoring Field	Description	Impact on Performance and Optimization
host	Node address	-
insert	Number of insertions per second	If the value of this field stays high, you can perform optimization based

		on the analysis of `dirty` and `used` .
query	Number of query requests per second	Check the index and make sure that the index exists.
update	Number of updates per second	1. Check the index and make sure that the index exists. 2. 2. If the value of this field stays high, you can perform optimization based on the analysis of dirty and used.
delete	Number of deletions per second	1. Check the index and make sure that the index exists. 2. If the value of this field stays high, you can perform optimization based on the analysis of dirty and used.
getmore	Number of getMore requests per second	-
command	Number of commands per second	-
dirty	Proportion of dirty data cached in the storage engine	If the value of this field stays high (above 20% by default), we recommend you increase the value of `threads_max` in the storage engine.
used	Proportion of the used cache of the storage engine	If the value of this field stays high (above 95% by default), we recommend you increase the value of `threads_max` in the storage engine.
flushes	Number of flushes per second	-
vsize	Amount of virtual memory used by processes	-
res	Amount of resident memory used by processes	-
qrw	Information of the waiting read/write queue on the client	If the value of this field stays above 0 and the value of `arw` stays close to 128, requests are queuing.
arw	Information of the active read/write queue on the client	-
net_in	Inbound traffic	-

net_out	Outbound traffic	-
conn	Number of connections	-
set	Replica set name	-
repl	Source-replica status	-
time	Monitoring time point	-

©2013-2025 Tencent Cloud International Pte. Ltd.

MongoTop

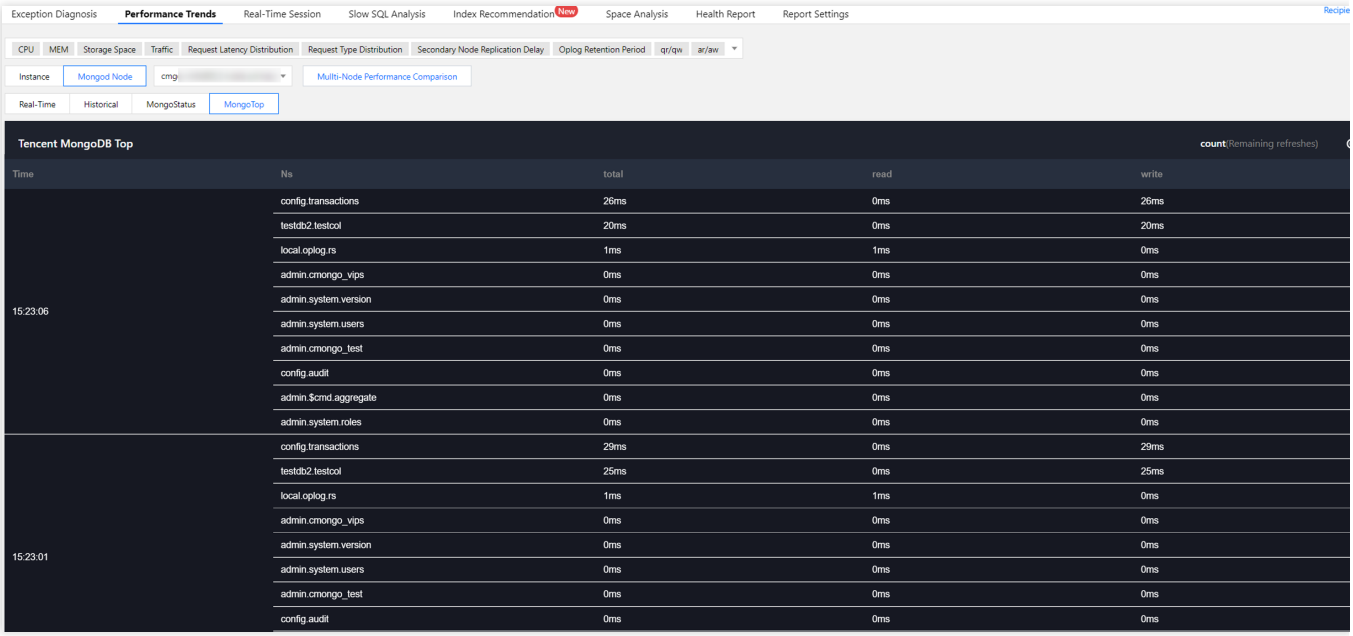
Last updated：2022-08-13 16:25:49

Overview

To facilitate daily database Ops, DBbrain provides the TencentDB for MongoDB MongoTop tool. Like MongoDB's official tool, this tool allows you to view the monitoring data of top tables at the node level in real time.

Directions

1. Log in to the DBbrain console and select **Performance Optimization** on the left sidebar. On the displayed page, select a database type and an instance at the top, and select the **Performance Trends** tab.
2. Select **Mongod Node** > **MongoTop**.
3. Select a node in the drop-down list.
4. Click **Pause** in the top-right corner to pause and view the data.



Time	Ns	total	read	write
15:23:06	config.transactions	26ms	0ms	26ms
	testdb2.testcol	20ms	0ms	20ms
	local.oplog.rs	1ms	1ms	0ms
	admin.cmongo_vips	0ms	0ms	0ms
	admin.system.version	0ms	0ms	0ms
	admin.system.users	0ms	0ms	0ms
	admin.cmongo_test	0ms	0ms	0ms
	config.audit	0ms	0ms	0ms
	admin.$cmd.aggregate	0ms	0ms	0ms
	admin.system.roles	0ms	0ms	0ms
15:23:01	config.transactions	29ms	0ms	29ms
	testdb2.testcol	25ms	0ms	25ms
	local.oplog.rs	1ms	1ms	0ms
	admin.cmongo_vips	0ms	0ms	0ms
	admin.system.version	0ms	0ms	0ms
	admin.system.users	0ms	0ms	0ms
	admin.cmongo_test	0ms	0ms	0ms
	config.audit	0ms	0ms	0ms

MongoTop table monitoring fields

MongoTop fields are as described below:

Time:

Current time of the database.

ns:

Namespace of the database.

total:

The total time mongod spent in the namespace.

read:

The time spent by mongod performing read operations in the namespace.

write:

The time spent by mongod performing write operations in the namespace.

©2013-2025 Tencent Cloud International Pte. Ltd.

Real-Time Session

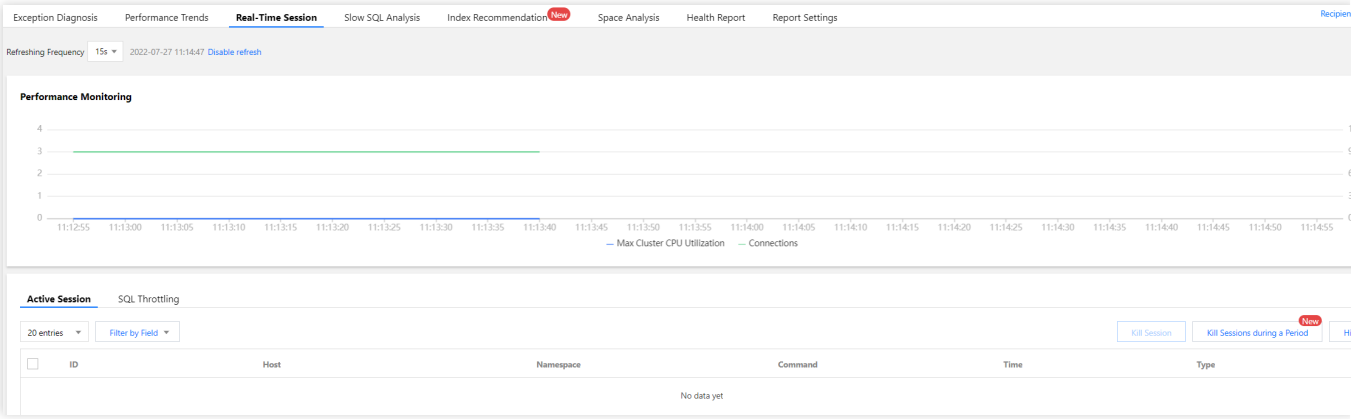
Last updated：2022-08-13 16:25:50

Feature description

You can use DBbrain's real-time session feature to view the real-time session information of your instance, including **Performance Monitoring**, **Connection Monitoring**, and **Active Session**.

Performance monitoring

Log in to the DBbrain console and select **Performance Optimization** on the left sidebar. On the displayed page, select a database type and an instance at the top, and select the **Real-Time Session** tab.
The **Refreshing Frequency** is **15s** by default and can be set as needed. You can also disable refresh.



Active session

On the **Active Session** tab, you can set the limit, filter by field, and enable or disable **Show Sleep Connection**.
You can set the limit to 20, 50, or 100.
Filter by Field supports filtering by **ID**, **HOST**, **Namespace**, **Type**, and **TIME** fields.
You can filter active sessions by **All** or **Others** (including `update`, `insert`, `query`, `getMore`, `remove`, `killcursors`, `command`, `compressed`, and none).

Killing sessions

DBbrain allows you to kill sessions for easier session management.

Kill current sessions

Select target sessions and click **Kill Session**.

You can kill 1–100 sessions at a time.

Kill sessions during a period

DBbrain offers the feature of killing sessions during a period. You can set the conditions for killing sessions, so that when the conditions are met, sessions will be killed automatically.

1. Task Settings.

Set the conditions for killing sessions during a period (including **HOST**, **Namespace**, **Type**, and **TIME**) and set the **Execution Mode**.

Note:

You can set one or more filter conditions which are evaluated using the logical AND operator.

If only **Time** and **Duration** are set, all sessions that meet the conditions will be killed quickly.

2. Session Preview.

After setting the task, you can preview the sessions to be killed in the **Session Preview** section. After killing sessions during a period is enabled, the generated sessions that meet the conditions will be automatically killed.

3. Task Details.

After setting the task, click **Details** in the top-right corner to view the details of the sessions killed during a period.

View the history of killed sessions

DBbrain provides the feature of viewing the history of killed sessions. To use this feature, click **History**.

SQL throttling

For more information, see SQL Throttling.

SQL Throttling

Last updated：2022-08-15 15:47:25

Feature description

The SQL throttling feature is suitable for scenarios involving high CPU utilization caused by high traffic. You can create SQL throttling tasks to control the database requests and SQL concurrency by setting the **SQL Type**, **Max Concurrency**, **Throttling Duration**, and **SQL Keyword**.

Note:

SQL throttling is supported only for TencentDB for MongoDB 4.0. To upgrade to this version, submit a ticket.

If SQL throttling prevents a SQL statement from being executed, the error message `SQL rejected by CDB_SQL_FILTER` will be displayed.

Creating a SQL throttling task

1. Log in to the DBbrain console and select **Performance Optimization** on the left sidebar. On the displayed page, select a database type and an instance at the top, and select the **Real-Time Session** tab to view the **SQL Throttling** module.

2. Create a SQL throttling task.

To create a SQL throttling task, you need to log in to your database first.

SQL Type: Select **Find**, **Insert**, **Update**, or **Delete**.

Max Concurrency: Set the maximum number of concurrent SQL executions. If the number of concurrent SQL executions containing specified keywords reaches this value, the SQL throttling policy will be triggered. If this value is set to 0, it restricts all matched SQL executions.

Execution Mode: Select **Scheduled stop** or **Manual stop**.

Throttling Duration: If you select **Scheduled stop**, you need to set how long the SQL throttling task runs.

SQL Keyword: Set the keywords. SQL statements containing the specified keywords will be restricted. Multiple keywords should be separated by comma and are evaluated by using the logical `AND` operator. Comma cannot be used as a keyword.

3. View the status and details of the SQL throttling task.

Click **Details** in the **Operation** column to view SQL throttling details.

After a SQL throttling task is enabled, it will remain in the **Running** status until its remaining time decreases to zero.

You can click **Disable** in the **Operation** column to disable the task, and its status will change to **Terminated**.

After a SQL throttling task is enabled, its status will change to **Terminated** once its remaining time decreases to zero.

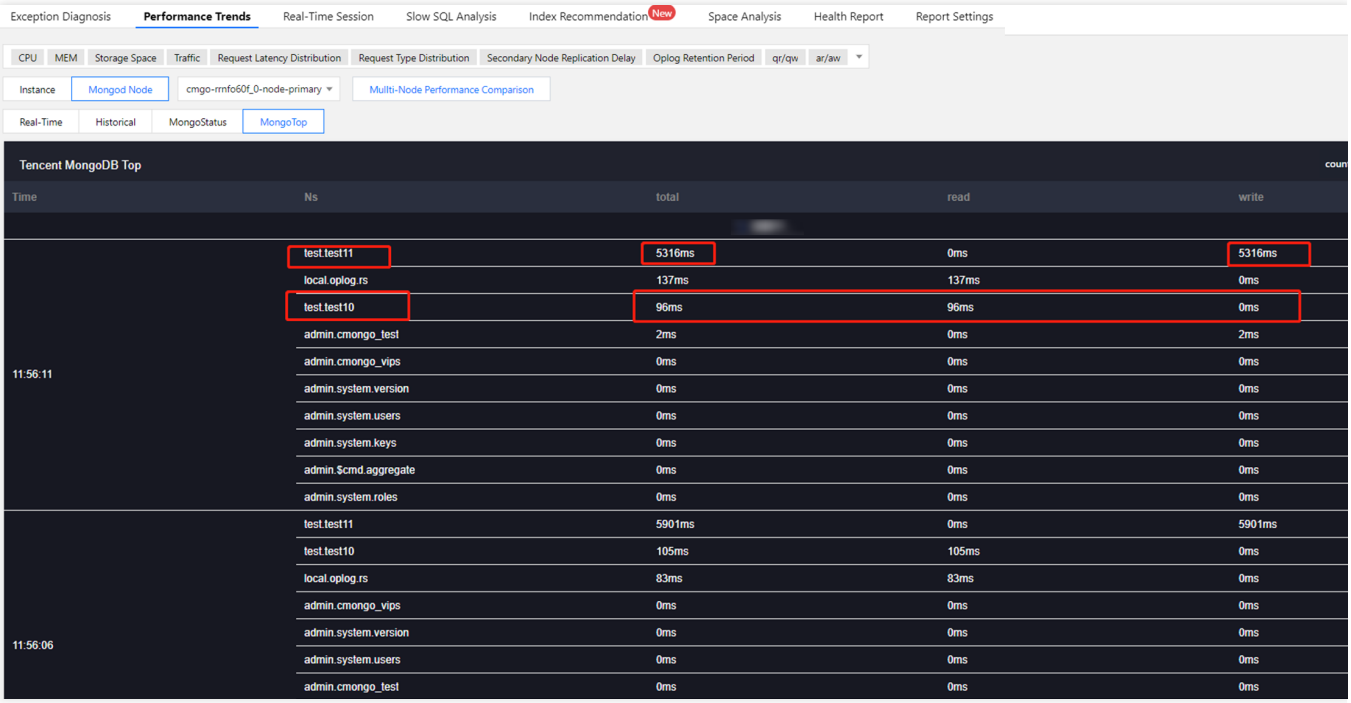
Click **Delete** in the **Operation** column to delete a SQL throttling task in the **Terminated** or **Completed** status.

©2013-2025 Tencent Cloud International Pte. Ltd.

Use case and effect of SQL throttling

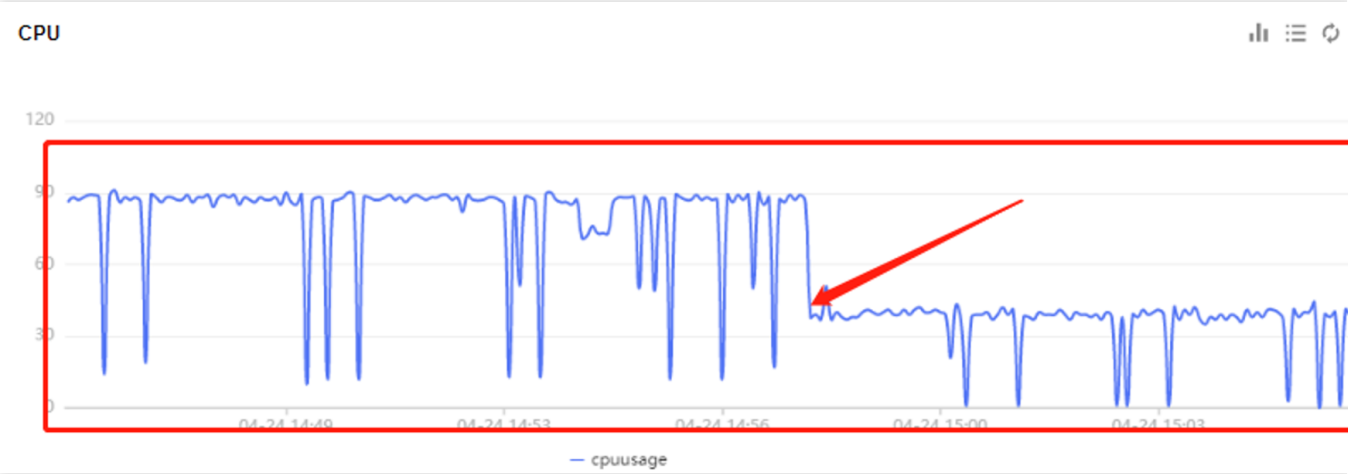
The database traffic was too high, resulting in a high CPU utilization.

1. The **MongoTop** tab in the console shows that the traffic of the `test.test11` table was too high. If the main business traffic was the read traffic to the `test.test10` table, then the traffic to the `test.test11` table was abnormal traffic.



Time	Ns	total	read	write
11:56:11	test.test11	5316ms	0ms	5316ms
	local.oplog.rs	137ms	137ms	0ms
	test.test10	96ms	96ms	0ms
	admin.cmongo_test	2ms	0ms	2ms
	admin.cmongo_vips	0ms	0ms	0ms
	admin.system.version	0ms	0ms	0ms
	admin.system.users	0ms	0ms	0ms
	admin.system.keys	0ms	0ms	0ms
	admin.$cmd.aggregate	0ms	0ms	0ms
	admin.system.roles	0ms	0ms	0ms
11:56:06	test.test11	5901ms	0ms	5901ms
	test.test10	105ms	105ms	0ms
	local.oplog.rs	83ms	83ms	0ms
	admin.cmongo_vips	0ms	0ms	0ms
	admin.system.version	0ms	0ms	0ms
	admin.system.users	0ms	0ms	0ms
	admin.cmongo_test	0ms	0ms	0ms

2. SQL throttling was enabled to throttle the traffic to the `test.test11` table.
3. As shown in the CPU performance trend chart below, CPU utilization dropped rapidly after throttling was enabled.



Index Recommendation

Last updated：2022-08-13 16:25:50

Feature description

Index optimization is an important part of database optimization. An optimal index can improve the query efficiency of the entire database. In view of the Ops characteristics of TencentDB for MongoDB, DBbrain offers the index recommendation feature to help you easily increase the global indexing efficiency of your instance.
After collecting and automatically analyzing slow logs in real time, the index recommendation feature proposes globally optimal indexes and rank them by their impact on the performance. An index that has a greater recommendation value will increase the performance more significantly. In addition, this feature also displays the slow queries and performance metrics associated with the recommended indexes, as well as invalid and duplicate indexes and their causes.
You only need to perform one operation based on the recommended indexes, and you can easily check the operation progress.

Enabling index recommendation

1. Log in to the DBbrain console and select **Performance Optimization** on the left sidebar. On the displayed page, select the target TencentDB for MongoDB instance at the top and select the **Index Recommendation** tab.
2. Read the note on data privacy risk and feature, indicate your consent, and click **Enable Now** as shown below.
Note:
When you enable index recommendation for the first time, all data may not be obtained immediately as the calculation starts from the current time point. Data will be complete after a period of time.
The index recommendation feature basically has no impact on the database performance; for example, in a 4-core 8GB MEM database, it consumes only 0.3 CPU cores after sampling for 10 minutes in a large table with 100 million data records.

Viewing recommended indexes

1. View the overall optimization level of the instance.
DBbrain assesses the index data of the source instance and presents one of four recommended SQL optimization levels: S > A > B > C. Level S indicates the optimal database performance, while level C indicates the worst database performance (the database requires urgent optimization).

©2013-2025 Tencent Cloud International Pte. Ltd.

2. View the recommended index sets.
DBbrain aggregates the recommendations based on the detected index data and sorts the indexes by recommendation value. A greater value indicates that the index set contains indexes that require urgent optimization, and their optimization will most significantly improve the database performance.
3. Click the name of an index set, and the recommendation details of indexes in it will be displayed on the right.
The **Recommended Indexes** tab displays indexes that need to be added as there are many slow queries. Similarly, indexes that have a greater recommendation value will more significantly enhance the performance after being added.
The **Invalid Indexes** tab displays indexes that are recommended to be deleted.

Adding a recommended index

1. On the **Recommended Indexes** tab, click an index, and the corresponding slow query analysis and records will be displayed on the right.
2. Click the icon in the red box as shown below to zoom in the slow query window for clearer information display. You can also download the slow query information.
3. In the **Auto-Generate Execution Statement** module, click **Create Index**.
To perform index operations, you need to log in to your database for authentication first.
4. You can select **Default** or **Specify options** as the creation method as needed, and DBbrain will automatically generate a creation statement accordingly.
5. You can view the index creation progress. You can also view the index set's operations in its **Operation Records**. In the operation list, you can view the historical addition or deletion details of indexes in the index set and terminate the indexes being processed.

Note:
To ensure the stability of your production database, when an index in the index set is being created or deleted, you cannot add or delete another index in the index set, and the system will report an error if you do so.

Deleting an invalid index based on recommendation

On the **Invalid** tab, view and delete invalid indexes. When your database contains an invalid index, the index recommendation system will display the reason for its invalidity and generate a deletion command. You can delete the index as prompted.

Viewing the index history and index adding effect

1. Click **History** on the right of **Recommended Index Sets** or click **View Details** below **Optimization Statistics** to view the historical index optimization records of the current instance.
2. After clicking **History**, click **Comparison** in the **Operation** column to view the effect before and after optimization.

©2013-2025 Tencent Cloud International Pte. Ltd.

Best Practices

Last updated：2025-04-14 21:12:24

For more information on how to fix the system exception caused by high CPU utilization in a TencentDB for MongoDB instance, see Fixing High CPU Utilization in MongoDB Instance.
For more information on how to fix the issue of a short retention period of node oplog in a TencentDB for MongoDB instance, see Fixing Short Node Oplog Retention Period in MongoDB Instance.

©2013-2025 Tencent Cloud International Pte. Ltd.

Full-Link Analysis
Enabling Full-Link Analysis

Last updated：2024-11-20 14:12:59

Full-link analysis collects, parses, and analyzes audit logs generated in the database system. Once full-link analysis feature is enabled, it assists users in monitoring database activities, identifying potential security issues, and supporting compliance requirements.

Prerequisites

To enable full-link analysis feature for MySQL, TDSQL-C for MySQL, MariaDB, TDSQL MySQL, and PostgreSQL instances, you should first enable the database audit service for each instance. For details on enabling the audit service, see the respective instructions for each database: Activate MySQL Audit Service, Activate TDSQL-C for MySQL Audit Service, Activate MariaDB Audit Service, Activate TDSQL MySQL Audit Service, and Activate PostgreSQL Audit Service.

Directions

1. Log in to the DBbrain Console.
2. Enable Full-Link Analysis.

Method I: Enabling Full-Link Analysis from the Instance Management Page

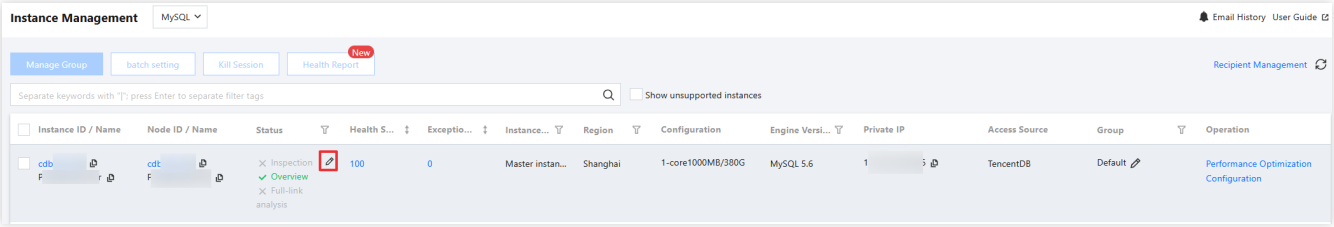
2.1. In the left sidebar, choose **Instance Management**, and select the database type.

2.2. Access the instance configuration page using either of the following methods.

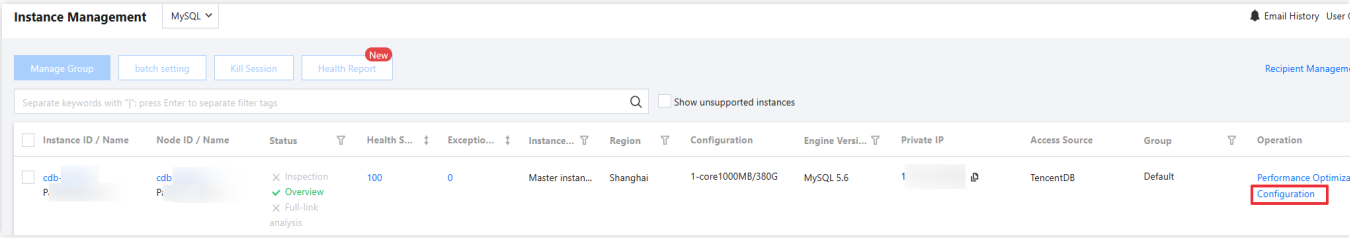
The display may vary by page. The following steps use the MySQL database as an example; see the actual page display for details.

Method I: In the **Status** column of the instance to be enabled, click

.



Method II: In the **Operation** column of the instance to be enabled, click **Configuration**.

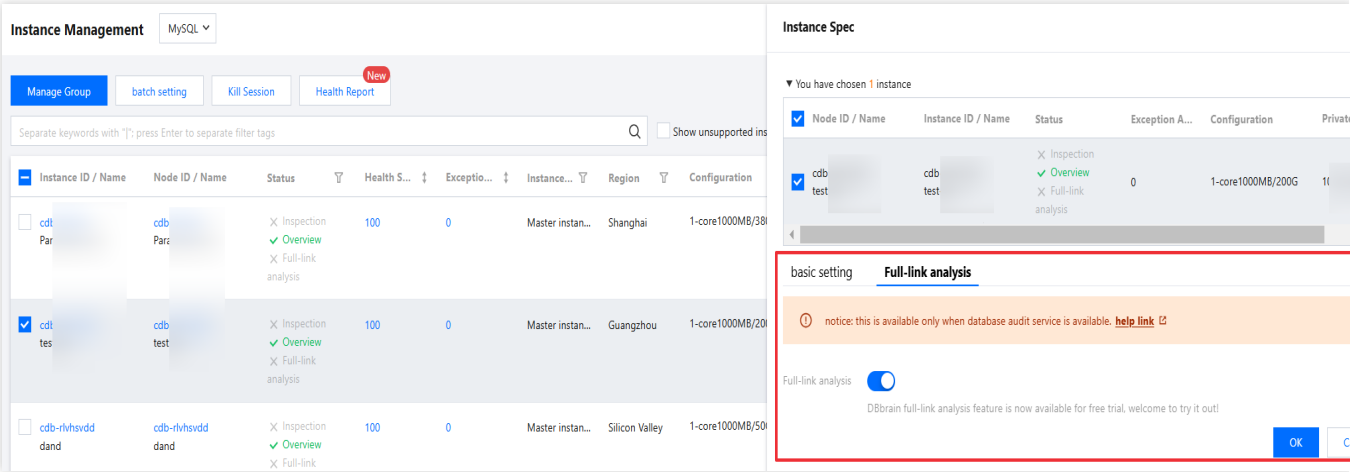


2.3. In the pop-up dialog box, review the selected instance information, select **Full-link analysis**, toggle the Full-Link Analysis button, and click **OK**.

Note:

After the full-link analysis feature is enabled, the **Full-link analysis** > **Details Inquiry** and **Full-link analysis** > **SQL Analysis** pages will display relevant statistical information. For detailed instructions on viewing SQL detail data and SQL analysis data, see Detail Query and SQL Analysis.

To disable the full-link analysis feature, follow the same steps as above. On the instance configuration page, toggle off the Full-Link Analysis button, and click **OK** .

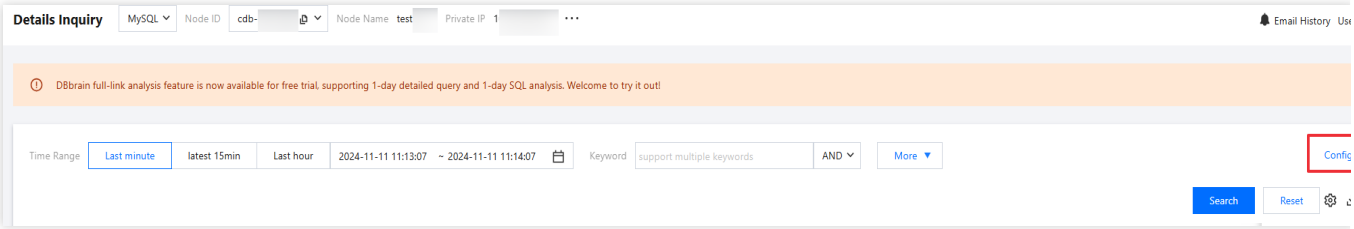


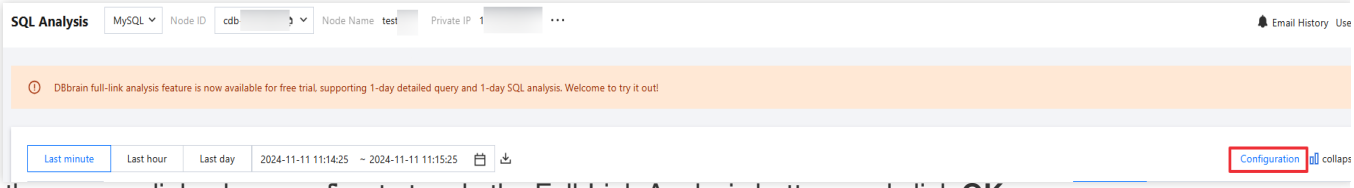
Method II: Enabling Full-Link Analysis from the Detail Query or SQL Analysis Page

2.1. In the left sidebar, select **Full-link analysis** > **Details Inquiry** or **Full-link analysis** > **SQL Analysis**.

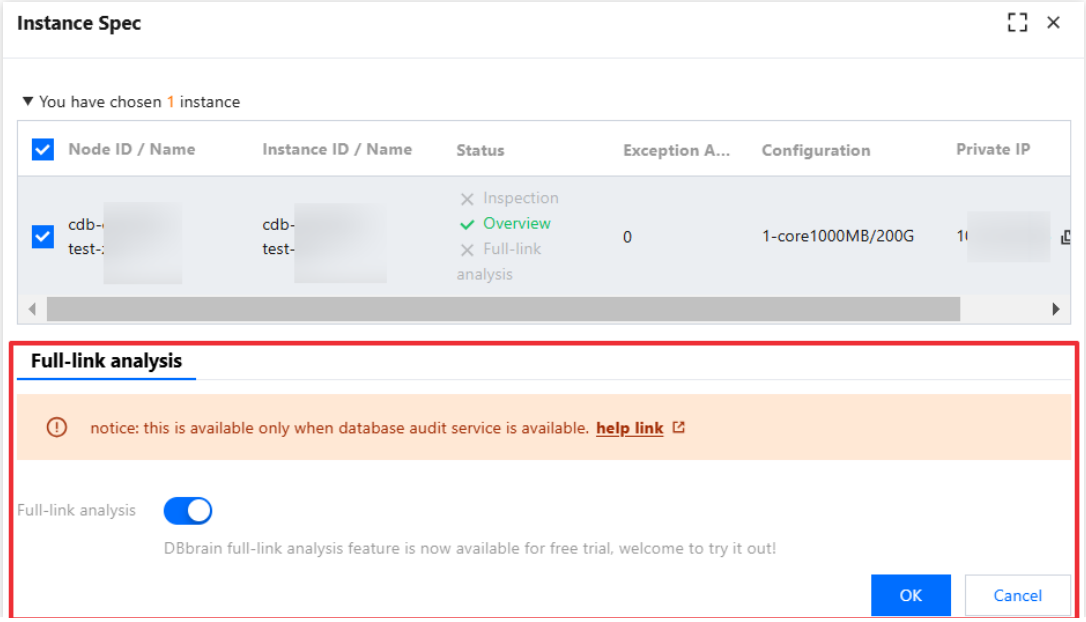
2.2. At the top of the page, select the database type, and then select node ID, instance ID, or cluster ID.

2.3. In the top right corner of the page, click **Configuration**.





2.4. In the pop-up dialog box, confirm to toggle the Full-Link Analysis button, and click **OK**.



Detail Query

Last updated：2024-11-20 14:19:36

The detail query page provides in-depth retrieval and analysis of SQL stored in the database. Through SQL detail query, users can filter, sort, and aggregate SQL statements, extracting necessary data from the database based on specific needs for further processing and analysis, thereby enabling targeted database performance optimization.
The detail query page allows users to view the SQL details list and SQL audit details, and download the SQL record list.

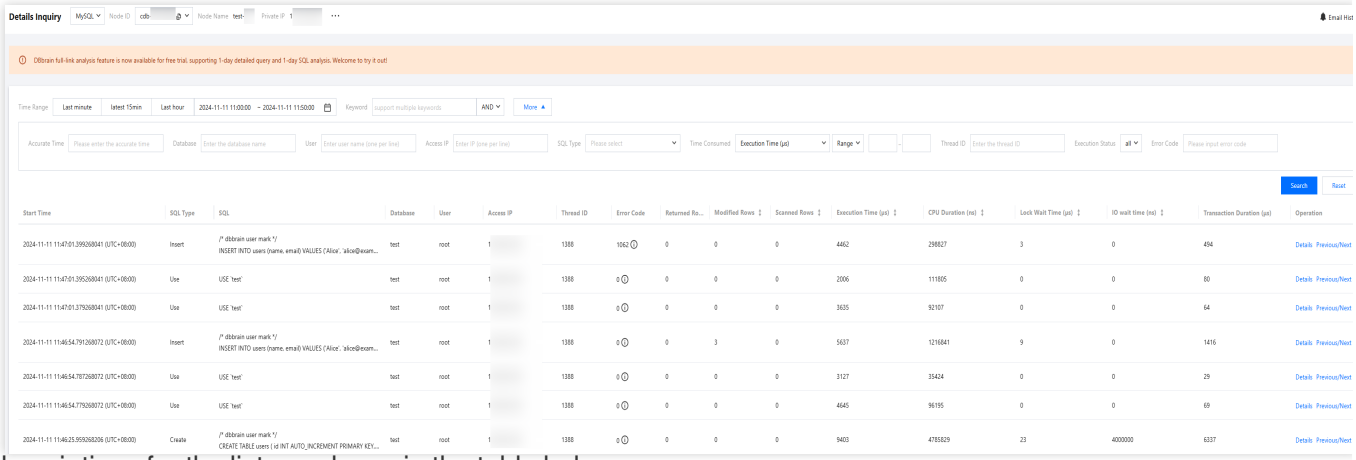
Prerequisites

Full-link analysis feature has been enabled for the instances. For detailed instructions, see Enabling End-to-End Analysis.

Viewing SQL Details List

1. Log in to the DBbrain Console.
2. In the left sidebar, choose **Full-link analysis** > **Details Inquiry**.
3. At the top of the page, select the database type, and select a node ID, instance ID, or cluster ID.
4. Filter SQL using the following options.
Time range: Select a time range, supporting options such as the last 1 minute, last 15 minutes, last 1 hour, or a custom time range.
Keywords: Enter one or more keywords for fuzzy matching, selecting **OR** or **AND**.
More fields: Click **More** at the top of the page to expand the available fields, and enter or select criteria in the respective fields. Setting multiple fields simultaneously is supported for precise SQL filtering.
5. View SQL details.
MySQL&TDSQL-C for MySQL
TDSQL MySQL&MariaDB
PostgreSQL

Viewing SQL Details



Field descriptions for the list are shown in the table below.

Field	Description
Start Time	The time when the SQL statement starts execution.
SQL Type	The type of SQL to which the executed statement belongs.
SQL	The executed SQL statement. Hover the mouse pointer over the SQL statement to display the copy and view buttons, allowing you to copy the SQL statement or view the complete SQL statement.
Database	The database where the SQL statement is executed.
User	The username initiating the SQL operation.
Access IP	The client IP address executing the SQL statement.
Thread ID	A unique identifier assigned to distinguish between different threads.
Error Code	Displays the corresponding error code if an error occurs while the SQL statement is executed. An error code of 0 indicates no error.
Returned Rows	The number of rows returned after the SQL statement is executed.
Modified Rows	The number of rows modified during an SQL update.
Scanned Rows	The number of rows searched in the SQL query.
Execution Time (μs)	The time spent to execute the SQL statement.
CPU Duration	The time spent by the SQL query to execute on the CPU.

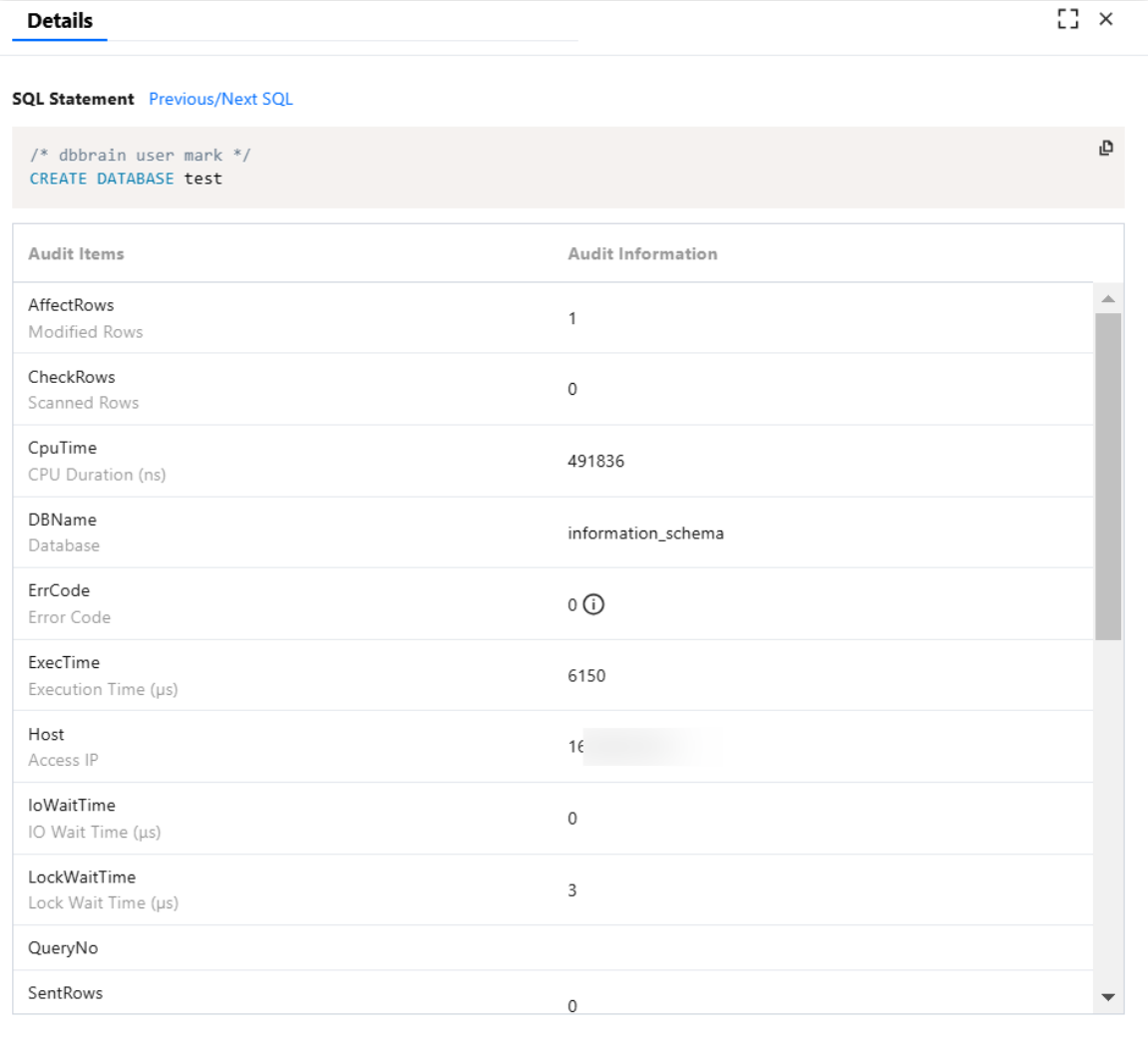
(ns)	
Lock Wait Time (μs)	The time a transaction spends waiting for another transaction to release a lock.
IO wait time (μs)	The time a thread spends waiting for an IO operation to complete.
Transaction Duration (μs)	The total time from the start of the transaction until it is committed or rolled back.
Operation	**Viewing Audit Details**: Click **Details**. For specific instructions on viewing audit details, see Viewing Audit Details. **Viewing Previous/Next SQL**: Click **Previous/Next SQL**. For specific instructions on viewing previous/next SQL, see Viewing Before and After SQL. Only MySQL supports viewing previous/next SQL.

Click

in the top right corner of the SQL template list to customize the list fields, then click **OK**.

Viewing Audit Details

1. In the SQL list, click **Details** in the **Operation** column for the target SQL.
2. On the audit details page, view audit items and audit information.



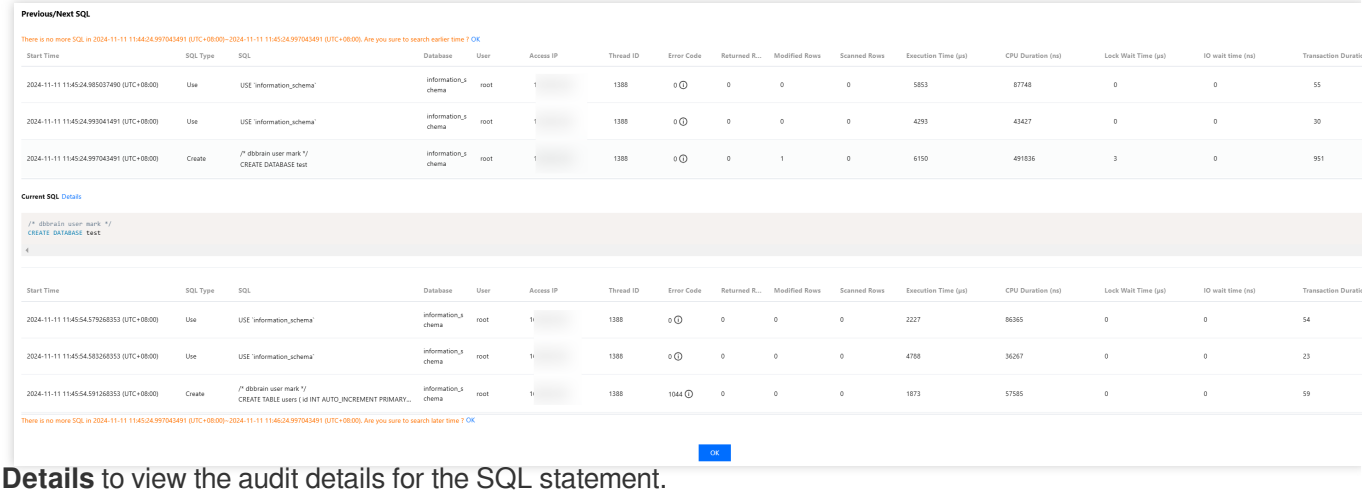
Click **Previous/Next SQL** to view the SQL statements executed before and after this SQL statement.

Viewing Previous/Next SQL Statements

Note:

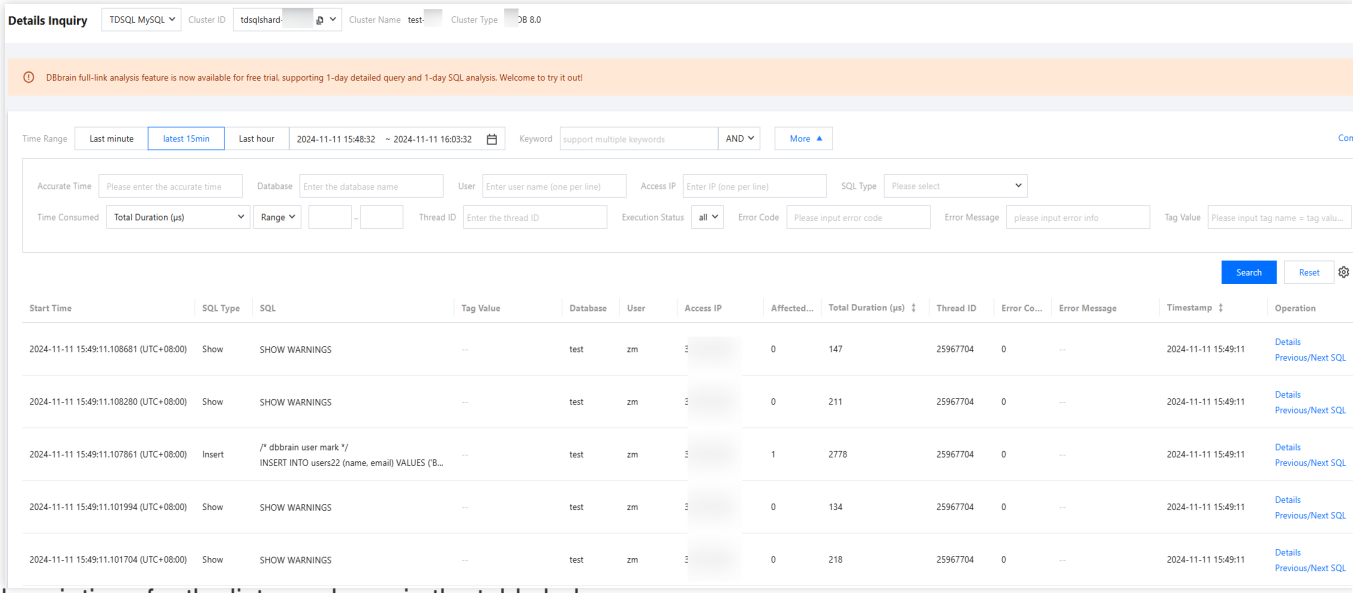
Only MySQL databases support viewing previous/next SQL statements.

1. In the SQL list, click **Previous/Next SQL** in the **Operation** column for the target SQL statement.
2. On the Previous/Next SQL page, view the SQL statements executed before and after the selected statement.



Click **Details** to view the audit details for the SQL statement.

Viewing SQL Details



Field descriptions for the list are shown in the table below.

Field	Description
Start Time	The time when the SQL statement starts execution.
SQL Type	The type of SQL to which the executed statement belongs.
SQL	The executed SQL statement. Hover the mouse pointer over the SQL statement to display the copy and view buttons, allowing you to copy the SQL statement or view the complete SQL statement.
Tag Value	A identifier added to SQL statements based on different business needs, allowing for the categorization and analysis of SQL statements by this identifier (tag value).
Database	The database where the SQL statement is executed.

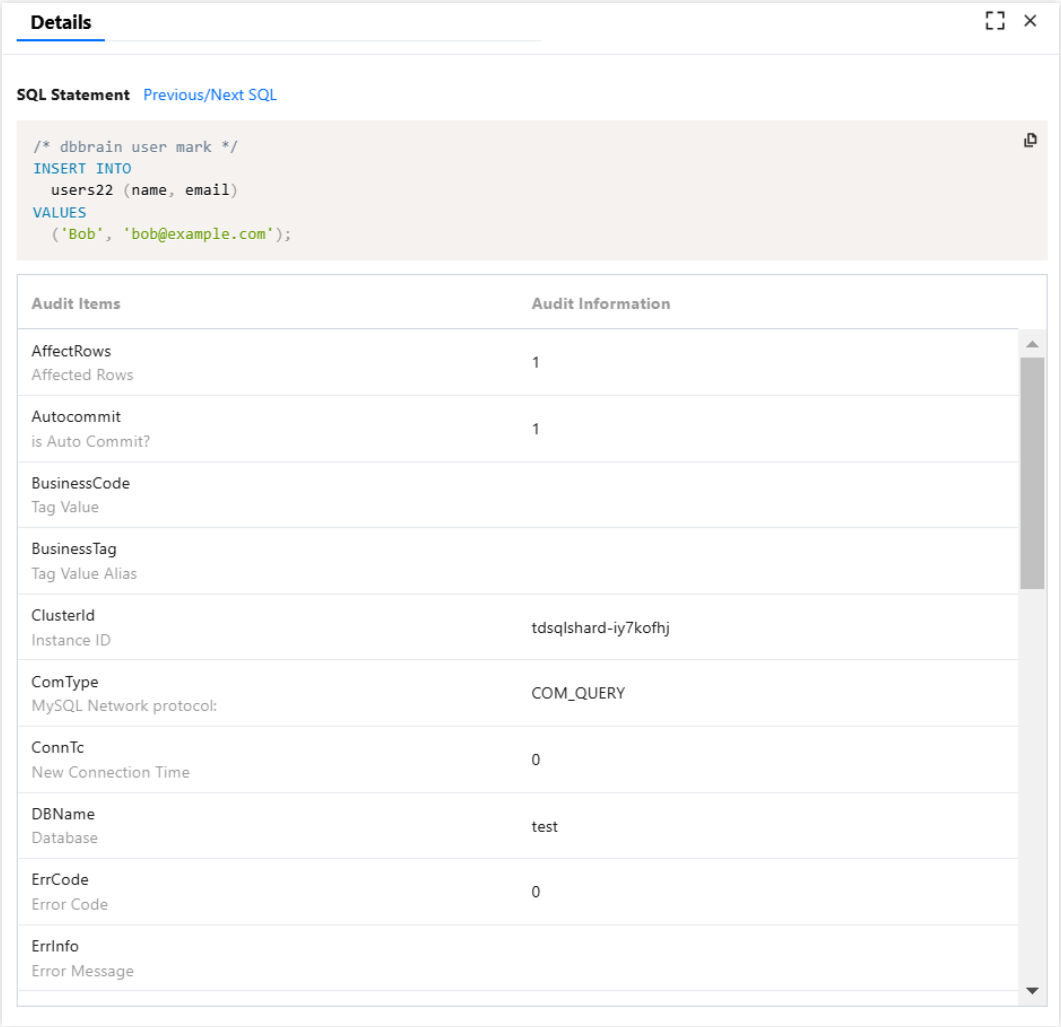
User	The username initiating the SQL operation.
Access IP	The client IP address executing the SQL statement.
Affected Rows	The number of record rows affected in an SQL operation.
Total Duration (μs)	The time spent to execute the SQL statement.
Thread ID	A unique identifier assigned to distinguish between different threads.
Error Code	Displays the corresponding error code if an error occurs while the SQL statement is executed. An error code of 0 indicates no error.
Error Message	Displays the corresponding error information if an error occurs during SQL statement execution.
Timestamp	Timestamp.
Operation	**Viewing Audit Details**: Click **Details**. For specific instructions on audit details, see Viewing Audit Details. **Viewing Previous/Next SQL**: Click **Previous/Next SQL**. For specific instructions on viewing previous/next SQL, see Viewing Before and After SQL.

Click

in the top right corner of the SQL template list to customize the list fields, then click **OK**.

Viewing Audit Details

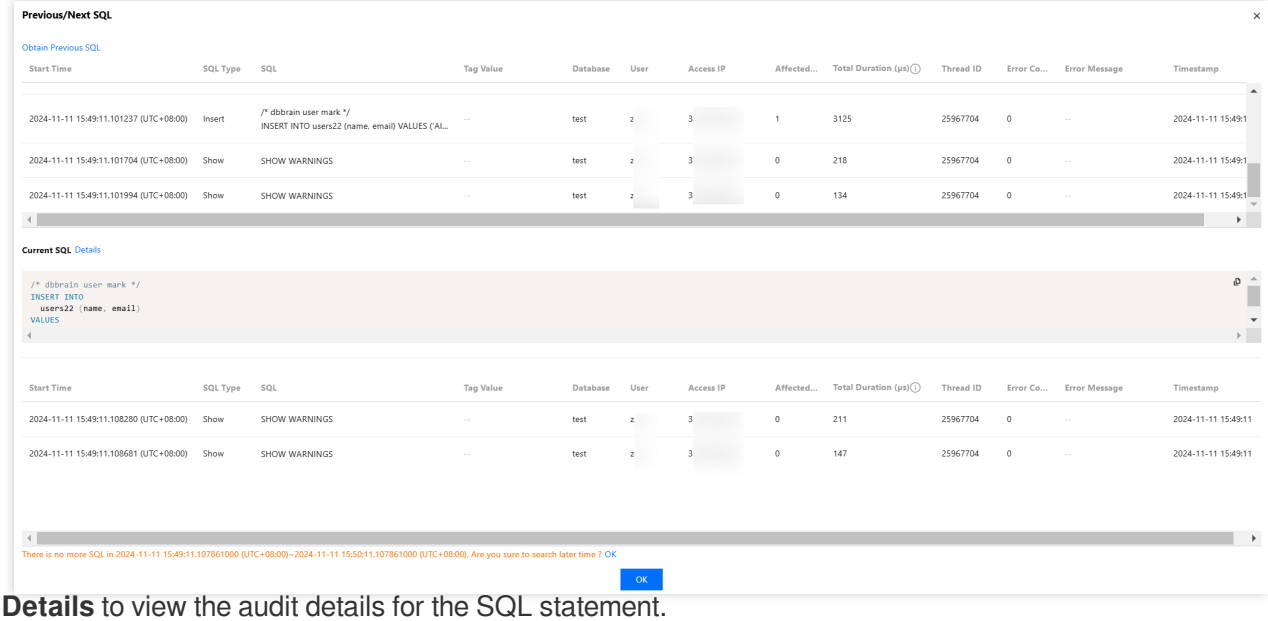
1. In the SQL list, click **Details** in the **Operation** column for the target SQL.
2. On the audit details page, view audit items and audit information.



Click **Previous/Next SQL** to view the SQL statements executed before and after this SQL statement.

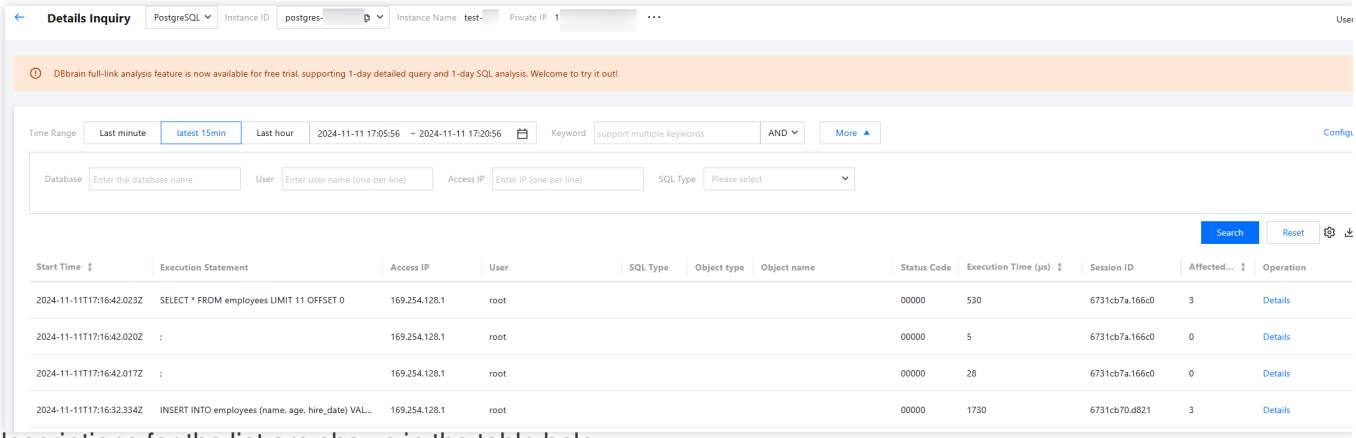
Viewing Previous/Next SQL Statements

1. In the SQL list, click **Previous/Next SQL** in the **Operation** column for the target SQL statement.
2. On the **Previous/Next SQL** page, view the SQL statements executed before and after the selected statement.



Click **Details** to view the audit details for the SQL statement.

Viewing SQL Details



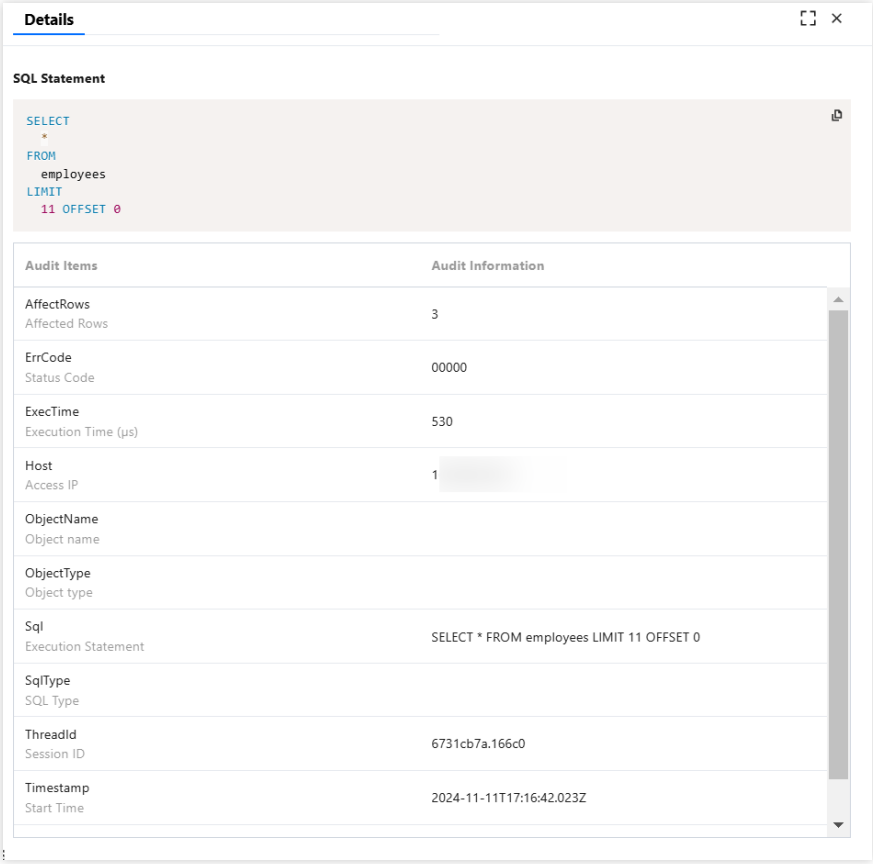
Field descriptions for the list are shown in the table below.

Field	Description
Start Time	The time when the SQL statement starts execution.
Execution Statement	The executed SQL statement. Hover the mouse pointer over the SQL statement to display the copy and view buttons, allowing you to copy the SQL statement or view the complete SQL statement.
Access IP	The client IP address executing the SQL statement.
User	The username initiating the SQL operation.
SQL Type	The type of SQL to which the executed statement belongs.
Object type	The object types include table, view, index, function, trigger, schema, and sequence.

Object name	The name of the object.
Status Code	When you execute an SQL command, different status codes may be returned. These status codes provide information about the execution result. Below are some common execution status codes: 00000: Indicates that the SQL command completed successfully. 01000: Indicates that an alarm was encountered during the execution of the SQL command. 02000: Indicates that the SQL command did not return any data. 03000: Indicates that the completion status of the SQL command is unknown. 08000: Indicates that an error occurred during the connection process. 09000: Indicates that an error occurred during the execution of the trigger. 0A000: Indicates that an unsupported feature was used.
Execution Time (ms)	The time spent to execute the SQL statement.
Session ID	Process ID.
Affected Rows	The number of rows affected after the SQL statement is executed.
Operation	**Viewing Audit Details**: Click **Details**. For specific instructions on audit details, see Viewing Audit Details.

Viewing Audit Details

1. In the SQL list, click **Details** in the **Operation** column for the target SQL.
2. On the audit details page, view audit items and audit information.



Downloading SQL Record List

1. Click in the top right corner of the SQL list. In the pop-up dialog box, click

to confirm and generate the download task.

2. Click

at the top of the list to view the file download progress. When the status displays as **Completed**, click **Copy URL** in the **Operation** column to download the file from the new page.

Note:

For tasks where the file is successfully generated, the file has been stored on the corresponding device.

The download file is available for 24 hours by default, so download it promptly.

The maximum download limit per task is 6 million records.

SQL Analysis

Last updated：2024-11-20 14:22:26

The SQL analysis page aggregates SQL template analysis, providing a global sorting of SQL templates based on their impact on various performance metrics within a specified period. It also enables real-time access to detailed SQL information within each template, helping users understand statistical data, data distribution, and outliers for better data comprehension and informed decision-making.

Note:

SQL analysis is supported only for MySQL, TDSQL-C for MySQL, MariaDB, and TDSQL MySQL instances.

Prerequisites

The full-link analysis feature has been enabled for the instances. For detailed instructions, see Enabling End-to-End Analysis.

Viewing SQL Performance Trend Chart and SQL Template List

1. Log in to the DBbrain Console.
2. In the left sidebar, choose **Full-link analysis** > **SQL Analysis**.
3. At the top of the page, select the database type, then select node ID, instance ID, or cluster ID.
4. At the top of the page, select a time range, including the last 1 minute, last 1 hour, last 1 day, or a custom time range.
5. View the SQL performance trend chart.

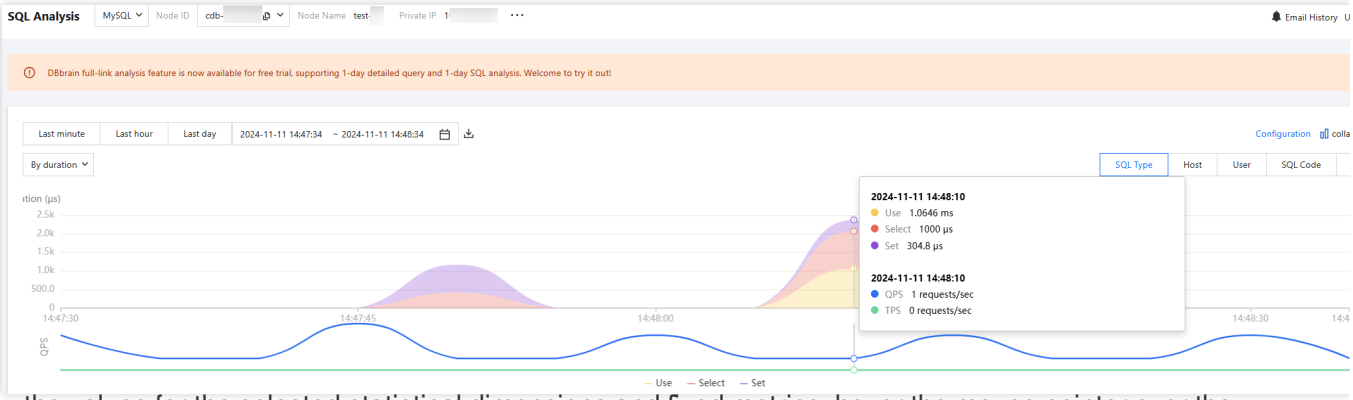
Select the horizontal axis metric in the trend chart, including options such as SQL Type, Host, User, SQL Code, and Time.

Select the vertical axis metric in the trend chart, including options for time consumption (SQL execution time) and count (number of SQL executions).

The vertical axis also displays QPS and TPS by default.

You can view the performance trend chart in the following methods.

©2013-2025 Tencent Cloud International Pte. Ltd.



To view the values for the selected statistical dimensions and fixed metrics, hover the mouse pointer over the performance trend chart.

To display finer granularity for the performance trend within a selected time period, click and drag over the desired time range on the chart. To exit, click **Reset** in the top right corner of the trend chart.

To download statistical data for the selected time period and dimension, click the

above the trend chart. The file will be downloaded in .csv format. This feature is only supported for MySQL and TDSQL-C for MySQL.

Note:

If you do not need to view the performance trend chart, click **collapse chart** in the top right corner of the page.

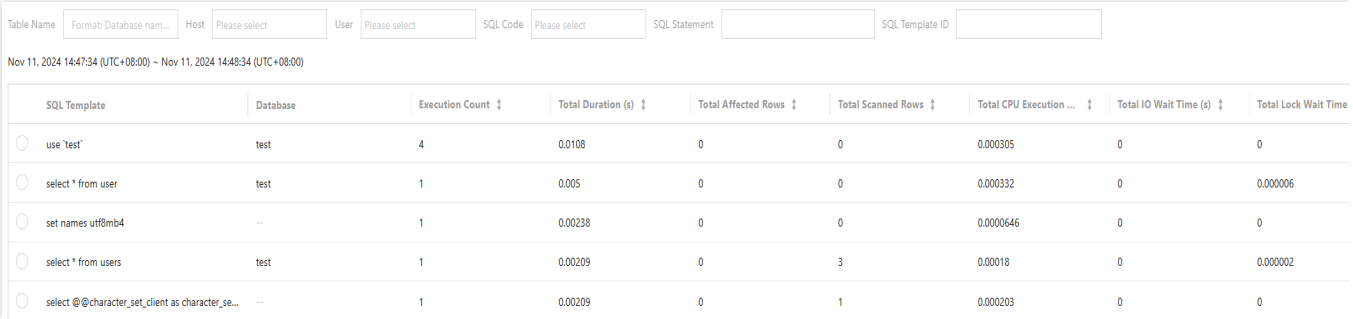
6. View the SQL template list.

6.1 (Optional) Click

in the top right corner of the SQL template list to customize the list fields, then click **OK**.

Filter the SQL template list by Table Name, Host, User, SQL Code, SQL Statement, or SQL Template ID.

6.2 View the SQL template list.



The SQL template list also supports the following operations:

Copy SQL template: Hover the mouse pointer over the SQL Template and click **Copy**.

View SQL template details: Click SQL Template or hover the mouse pointer over it, and click **View**. For detailed instructions, see View SQL Template Details.

Sort SQL templates in ascending or descending order based on a specific field: On the right side of the field marked with

, click

to sort the SQL templates in ascending or descending order.

Download SQL template list: Click

in the top right corner of the SQL template list to download a file in .csv format.

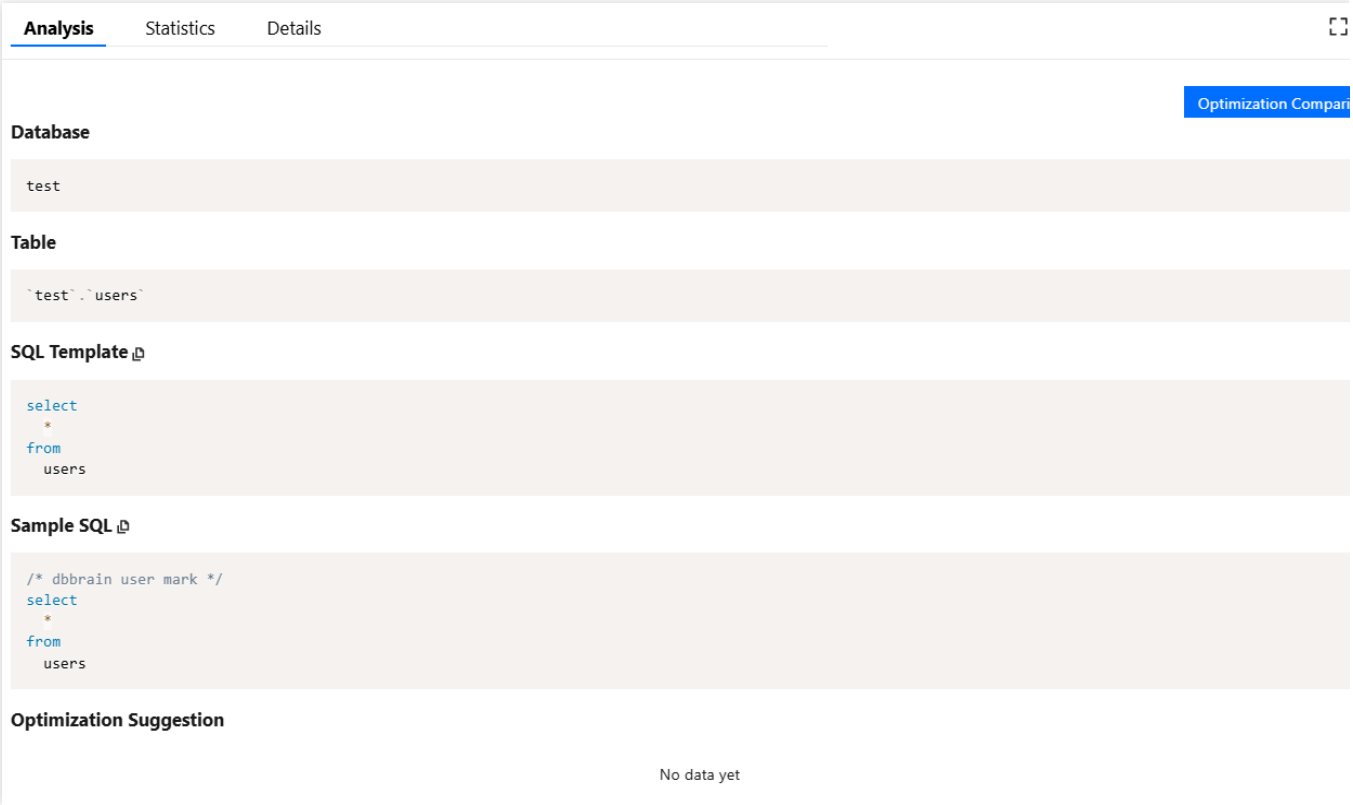
Viewing SQL Template Details

In the SQL template list, click an SQL template or hover the mouse pointer over it, and click **View**.

View the analysis view, statistical information, and detailed records for the selected SQL template.

Viewing SQL Template - Analysis Tab

Displays information related to the SQL template, including associated databases, tables, and template details.



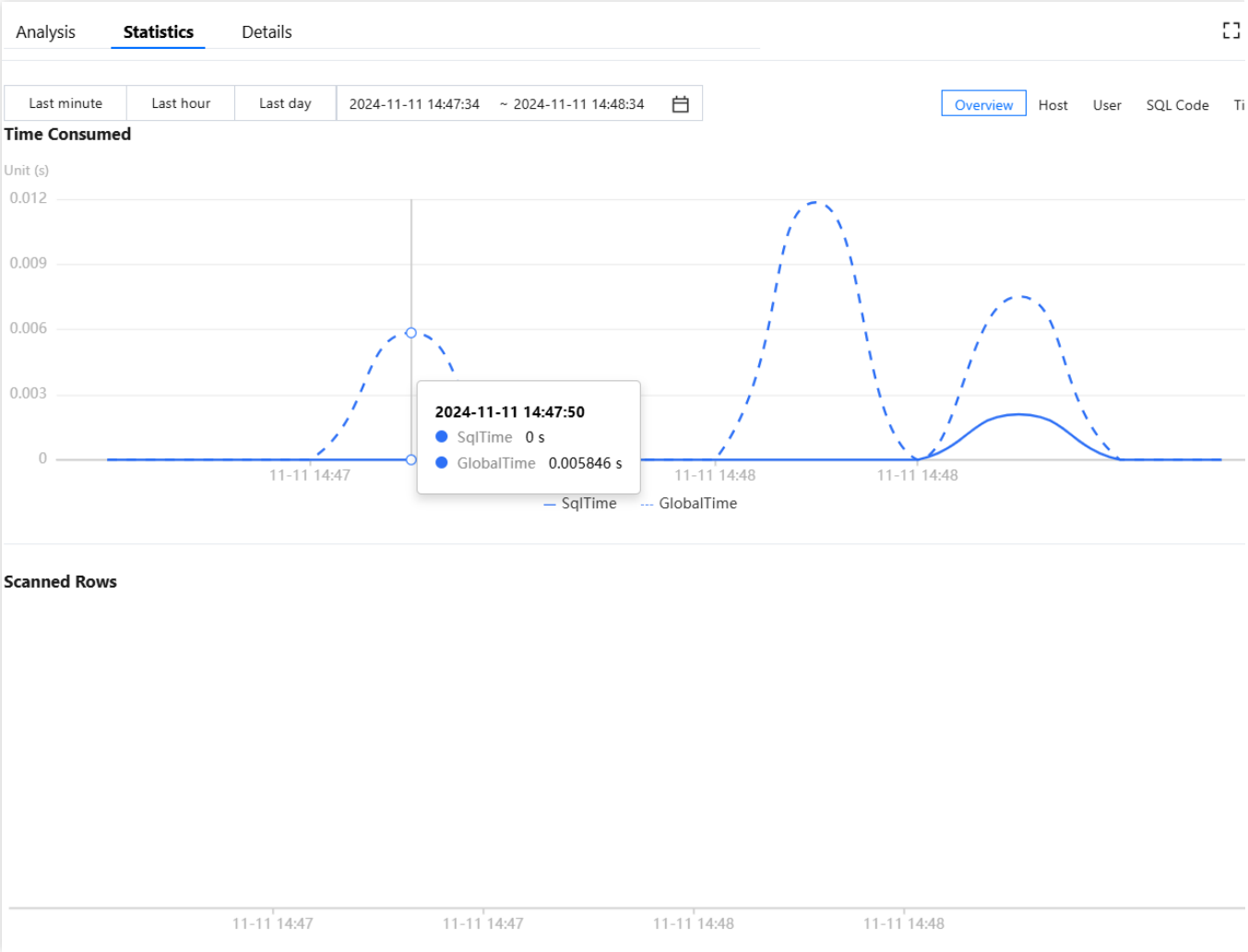
Viewing SQL Template - Statistics Tab

Displays performance trends and ring charts for SQL execution time and number of scanned rows.

Supported statistical parameters:

Select time range: You can view statistics for the last 1 minute, last 1 hour, last 1 day, or a custom time range.

Select performance statistical dimension: Options include Overview, Host, User, SQL Code, and Time.



Viewing SQL Template - Detail Tab

The Detail tab displays the specific results for the SQL template. For detailed instructions on viewing detailed SQL information, see Detail Query.