TencentDB for DBbrain

Operation Guide

Product Documentation

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

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.

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

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`

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/*`

	`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`

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

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.

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.

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.

Exception Alarms

Last updated：2024-07-31 11:14:57

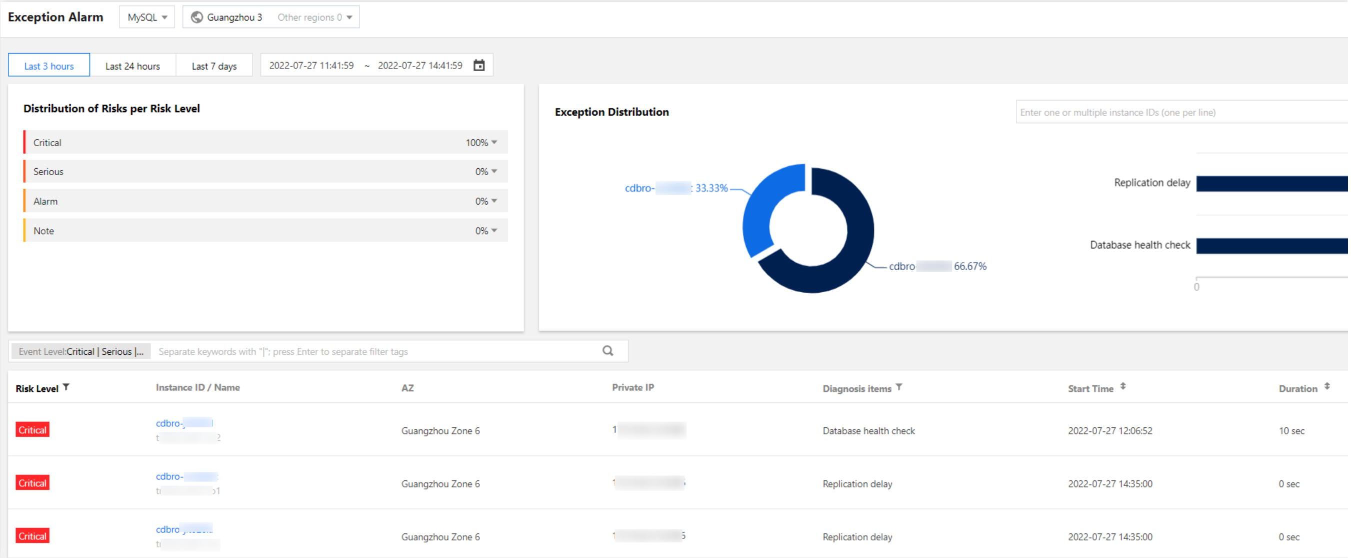
The exception alarm page displays the information overview of exception alarms (exceptions detected by "24/7 Exception Diagnosis") generated by database instances connected to DBbrain under your account.

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.



Distribution of Risks per Risk Level

Critical	100%
Serious	0%
Alarm	0%
Note	0%

Risk Level	Instance ID / Name	AZ	Private IP	Diagnosis items	Start Time	Duration
Critical	cdbro-[redacted] t[redacted]2	Guangzhou Zone 6	1[redacted]	Database health check	2022-07-27 12:06:52	10 sec
Critical	cdbro-[redacted] t[redacted]1	Guangzhou Zone 6	1[redacted]	Replication delay	2022-07-27 14:35:00	0 sec
Critical	cdbro[redacted] t[redacted]	Guangzhou Zone 6	1[redacted]5	Replication delay	2022-07-27 14:35:00	0 sec

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.

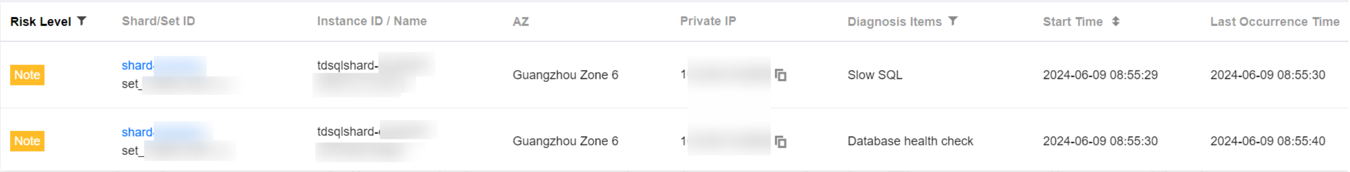
The list displays fields such as risk level, instance ID/name, diagnosis items, start time, last occurrence time, and operation. For different selected database types, the list shows different fields. Refer to the actual display.

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.

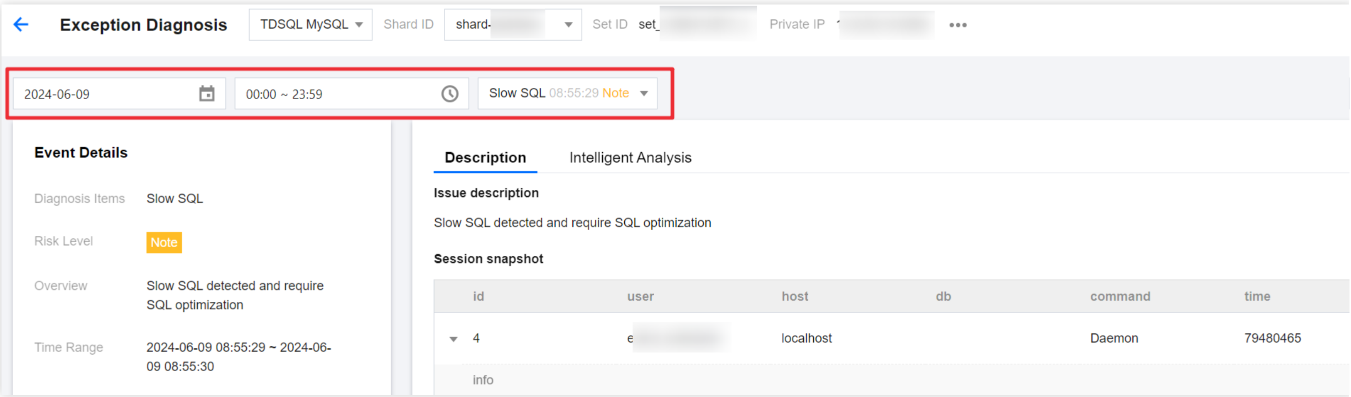
Viewing Exception Alarm Details

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.



Risk Level	Shard/Set ID	Instance ID / Name	AZ	Private IP	Diagnosis Items	Start Time	Last Occurrence Time
Note	shard[redacted] set_[redacted]	tdsqlshard-[redacted]	Guangzhou Zone 6	1[redacted]	Slow SQL	2024-06-09 08:55:29	2024-06-09 08:55:30
Note	shard[redacted] set_[redacted]	tdsqlshard-[redacted]	Guangzhou Zone 6	1[redacted]	Database health check	2024-06-09 08:55:30	2024-06-09 08:55:40

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.



Event Details

Diagnosis Items	Slow SQL
Risk Level	Note
Overview	Slow SQL detected and require SQL optimization
Time Range	2024-06-09 08:55:29 ~ 2024-06-09 08:55:30

Issue description

Slow SQL detected and require SQL optimization

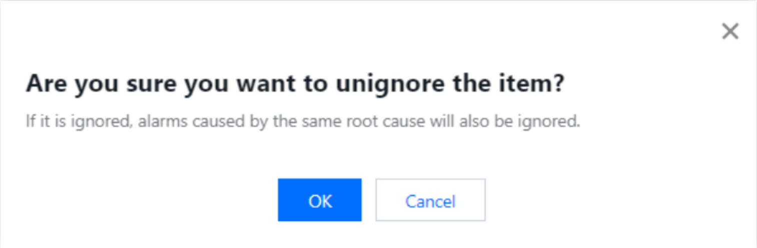
Session snapshot

id	user	host	db	command	time
4	e[redacted]	localhost		Daemon	79480465
info					

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.



Are you sure you want to unignore the item?

If it is ignored, alarms caused by the same root cause will also be ignored.

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.

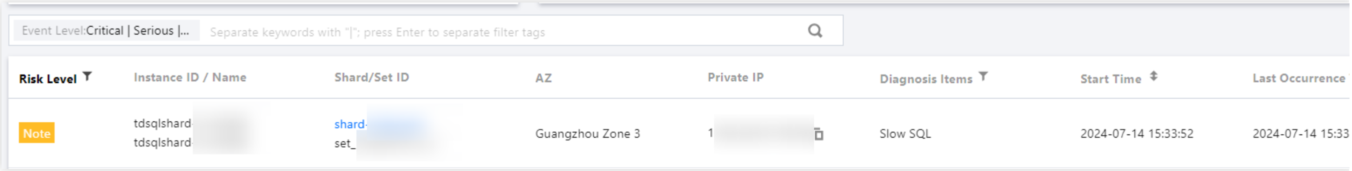
Exporting the Exception Alarm List

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.



Event Level:Critical | Serious |... Separate keywords with "|"; press Enter to separate filter tags

Risk Level	Instance ID / Name	Shard/Set ID	AZ	Private IP	Diagnosis Items	Start Time	Last Occurrence
Note	tdsqlshard [redacted] tdsqlshard [redacted]	shard- [redacted] set_ [redacted]	Guangzhou Zone 3	1 [redacted]	Slow SQL	2024-07-14 15:33:52	2024-07-14 15:33

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.

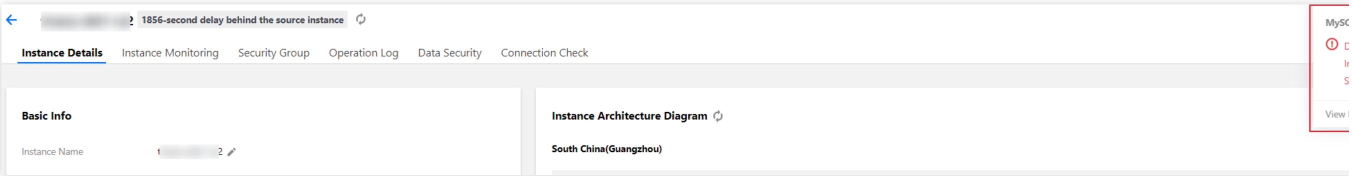
This document takes logging in to the MySQL console as an example to view alarms.

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.



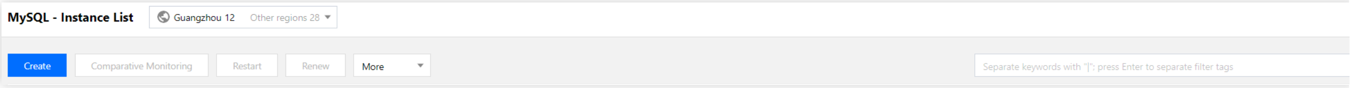
[redacted] 2 1856-second delay behind the source instance

Instance Name 1 [redacted] 2

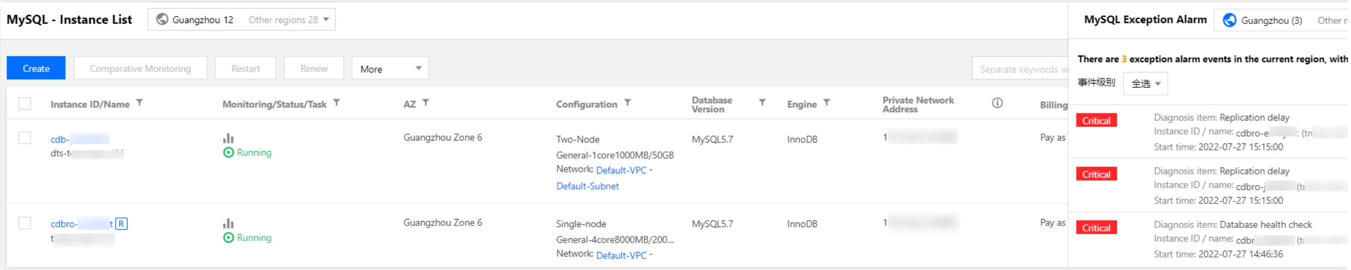
Option 2

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.



MySQL - Instance List Guangzhou 12 Other regions 28

□	Instance ID/Name	Monitoring/Status/Task	AZ	Configuration	Database Version	Engine	Private Network Address	Billing
□	cdb-[redacted] dts-t[redacted]	Running	Guangzhou Zone 6	Two-Node General-1core1000MB/50GB Network: Default-VPC - Default-Subnet	MySQL5.7	InnoDB	1[redacted]	Pay as
□	cdbro-[redacted]t R t[redacted]	Running	Guangzhou Zone 6	Single-node General-4core8000MB/200... Network: Default-VPC -	MySQL5.7	InnoDB	1[redacted]	Pay as

MySQL Exception Alarm Guangzhou (3) Other r

There are 3 exception alarm events in the current region, with

事件级别 全选

Critical
Diagnosis item: Replication delay
Instance ID / name: cdbro-e[redacted] (tr[redacted]
Start time: 2022-07-27 15:15:00

Critical
Diagnosis item: Replication delay
Instance ID / name: cdbro-j[redacted] (tr[redacted]
Start time: 2022-07-27 15:15:00

Critical
Diagnosis item: Database health check
Instance ID / name: cdbr[redacted] (t[redacted]
Start time: 2022-07-27 14:46:36

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:

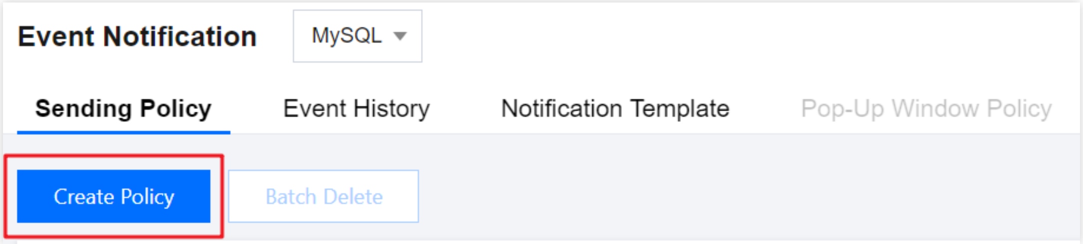
DBbrain event notification is fundamentally different from TCOP alarms. TCOP metric alarm feature monitors specific metrics and notifies users the corresponding metric alarm when the metrics reach the monitoring threshold. On the other hand, DBbrain event notification informs users about the diagnostic 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.

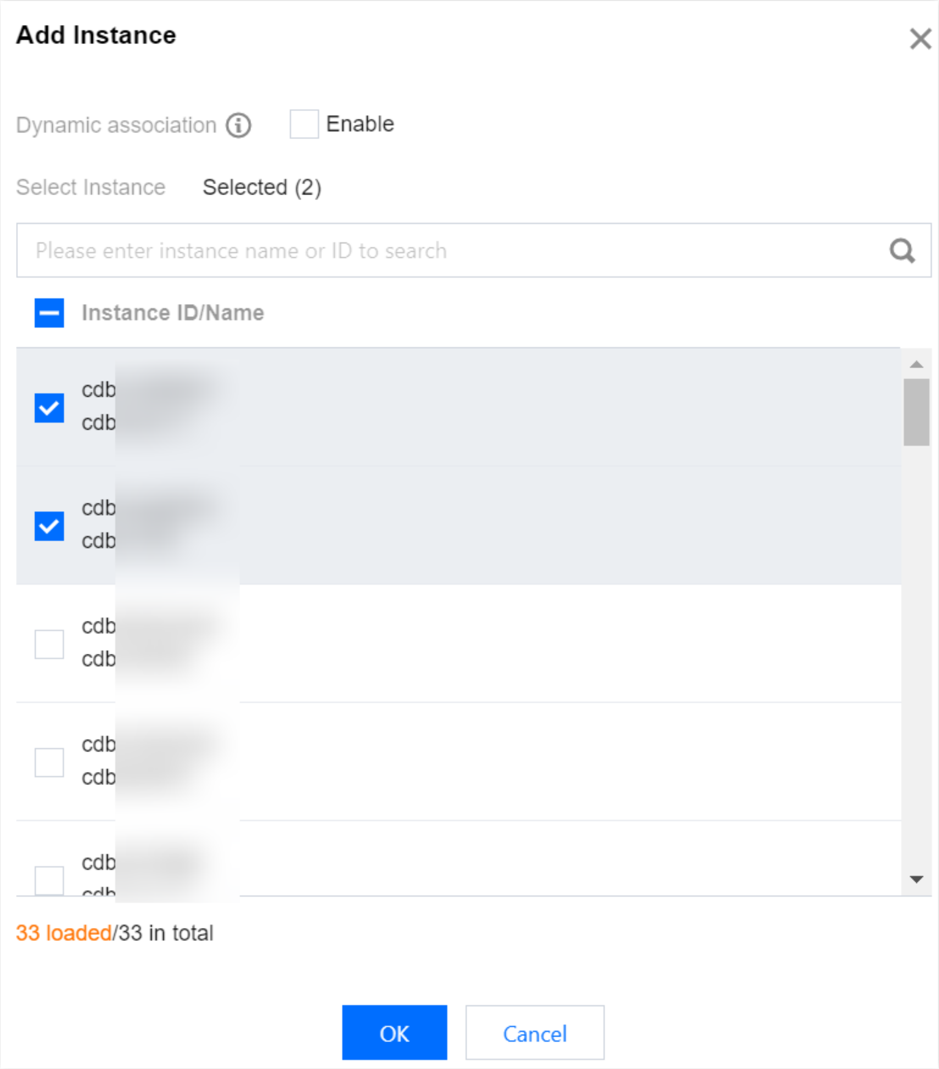
4.1 Select the database type.

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** .



Add Instance

Dynamic association ⓘ □ Enable

Select Instance Selected (2)

Please enter instance name or ID to search

☐ Instance ID/Name

☑ cdb
cdb

☑ cdb
cdb

□ cdb
cdb

□ cdb
cdb

□ cdb
cdb

33 loaded/33 in total

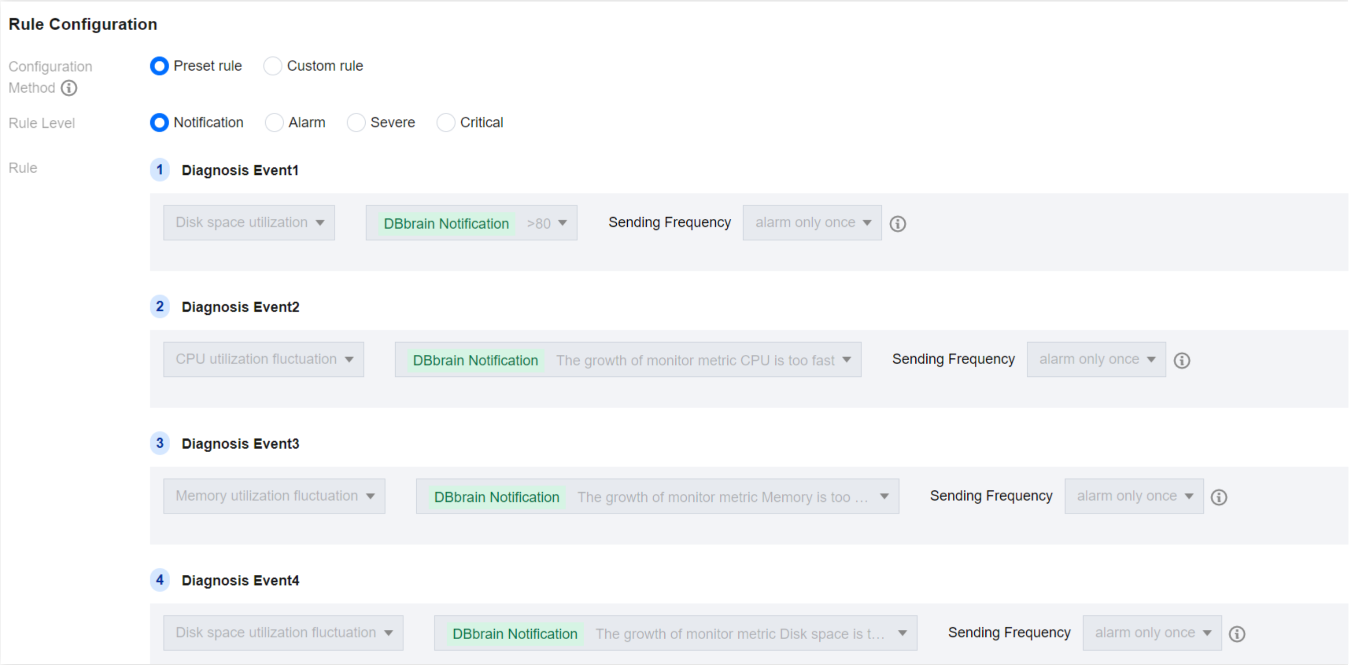
Select whether to enable dynamic association of instances: After dynamic association is enabled, all instances will be automatically selected for you. If there are newly added instances under your account, they will be dynamically loaded into this policy configuration.

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.



Rule Configuration

Configuration Method: ● Preset rule ○ Custom rule

Rule Level: ● Notification ○ Alarm ○ Severe ○ Critical

Rule

1 Diagnosis Event1

Disk space utilization | DBbrain Notification >80 | Sending Frequency alarm only once

2 Diagnosis Event2

CPU utilization fluctuation | DBbrain Notification The growth of monitor metric CPU is too fast | Sending Frequency alarm only once

3 Diagnosis Event3

Memory utilization fluctuation | DBbrain Notification The growth of monitor metric Memory is too ... | Sending Frequency alarm only once

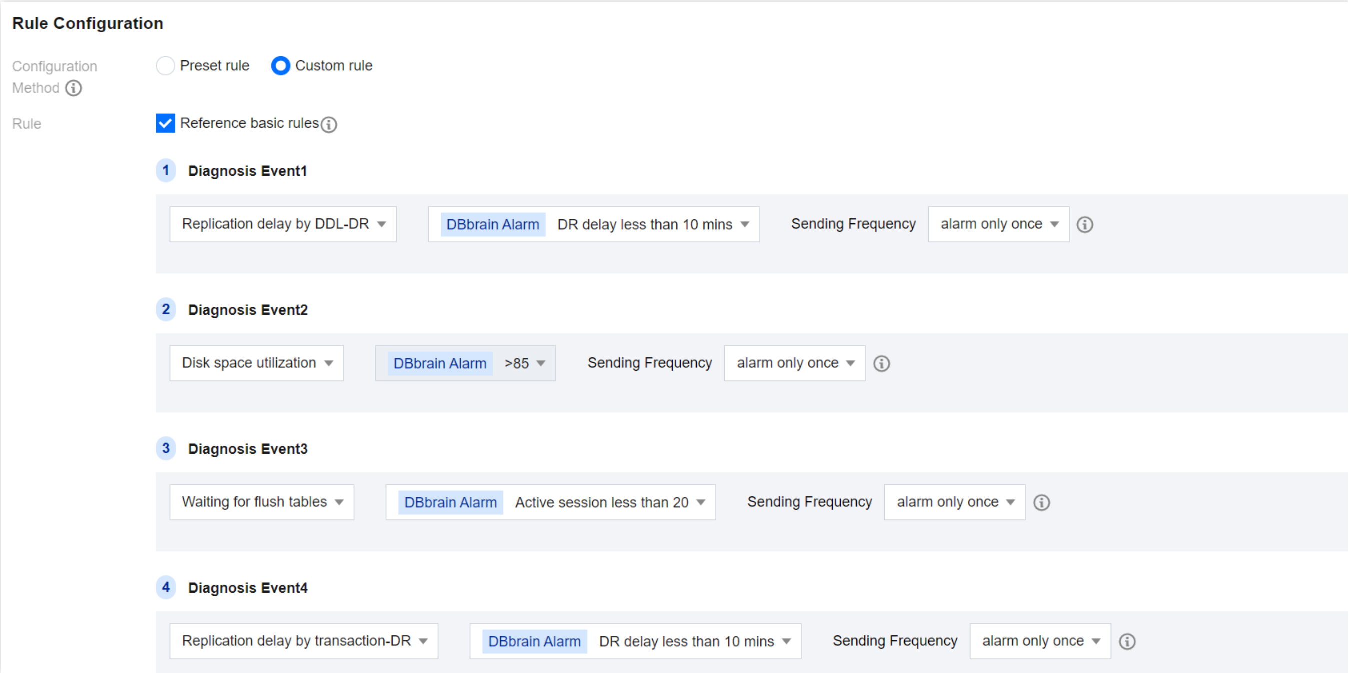
4 Diagnosis Event4

Disk space utilization fluctuation | DBbrain Notification The growth of monitor metric Disk space is t... | Sending Frequency alarm only once

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.



Rule Configuration

Configuration Method: ○ Preset rule ● Custom rule

Rule: ☑ Reference basic rules

1 Diagnosis Event1

Replication delay by DDL-DR | DBbrain Alarm DR delay less than 10 mins | Sending Frequency alarm only once

2 Diagnosis Event2

Disk space utilization | DBbrain Alarm >85 | Sending Frequency alarm only once

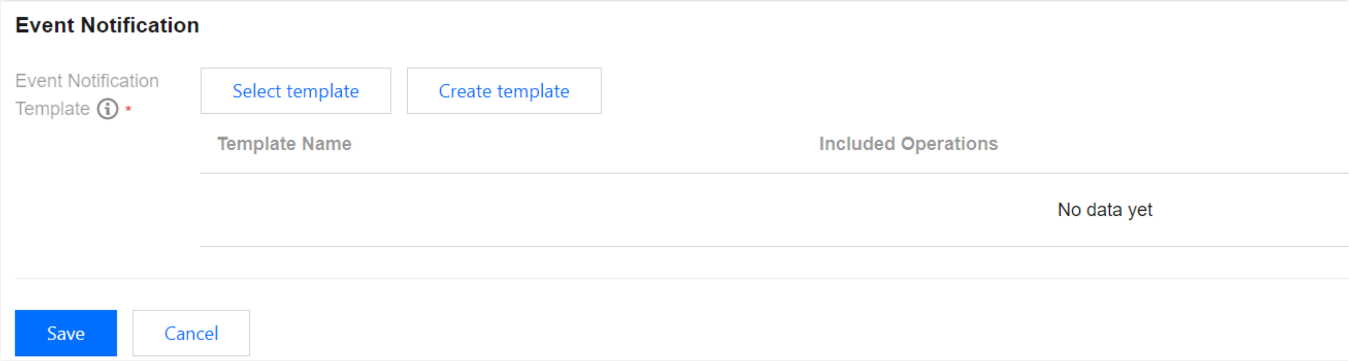
3 Diagnosis Event3

Waiting for flush tables | DBbrain Alarm Active session less than 20 | Sending Frequency alarm only once

4 Diagnosis Event4

Replication delay by transaction-DR | DBbrain Alarm DR delay less than 10 mins | Sending Frequency alarm only once

4.5 Event notification configuration.


Event Notification

Event Notification Template *

Select template Create template

Template Name	Included Operations
	No data yet

Save Cancel

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

Selecting Template

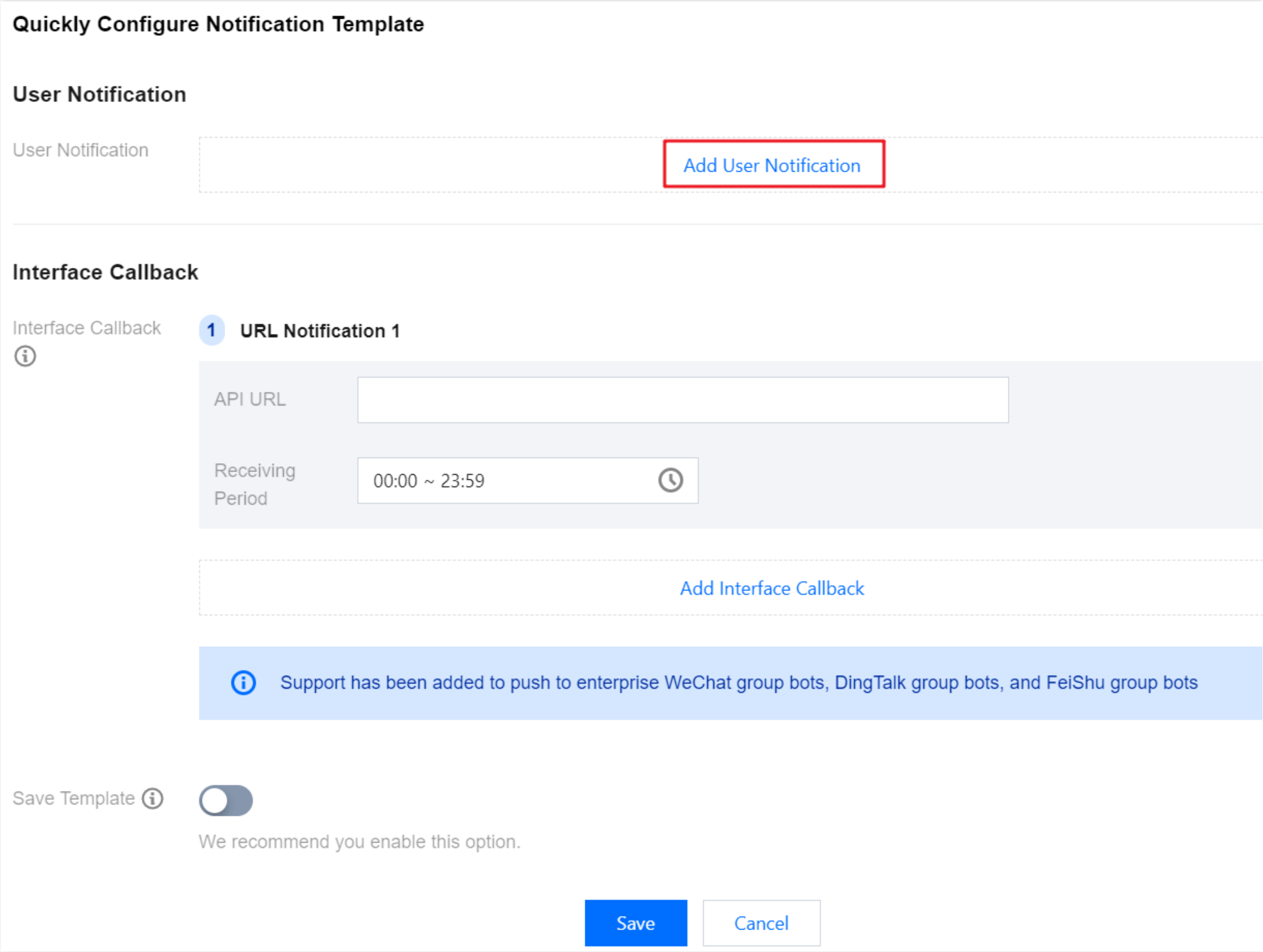
Click **Select template** , and in the pop-up dialog box, select the template name, and then click **OK** . This requires a configured notification template on the **Event Notification** - **Notification Template** page. For detailed operations, see managing notification templates.

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** .



Quickly Configure Notification Template

User Notification

User Notification: Add User Notification

Interface Callback

Interface Callback: 1 URL Notification 1

API URL:

Receiving Period: 00:00 ~ 23:59

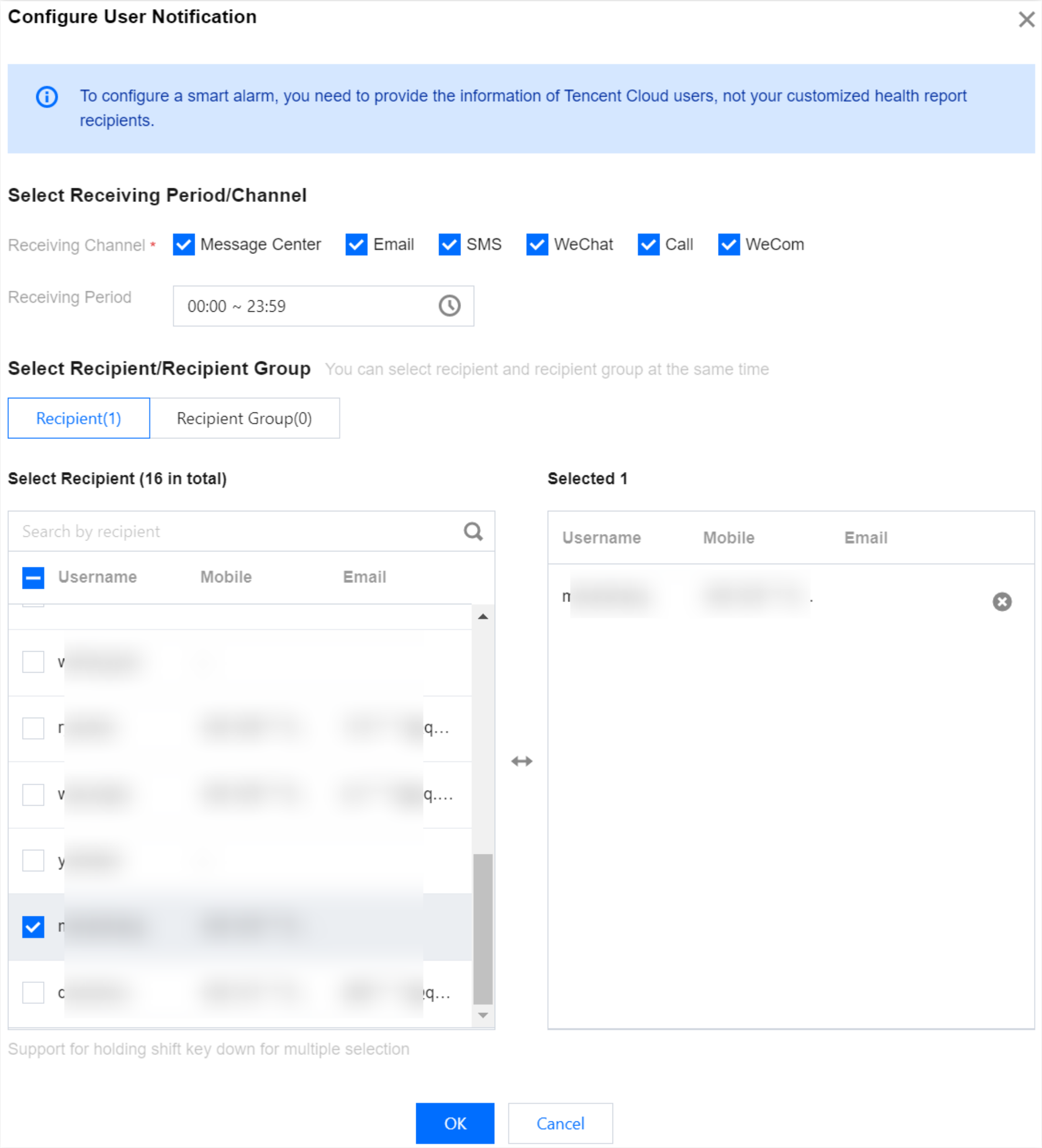
Add Interface Callback

Support has been added to push to enterprise WeChat group bots, DingTalk group bots, and FeiShu group bots

Save Template

We recommend you enable this option.

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** .



Configure User Notification

To configure a smart alarm, you need to provide the information of Tencent Cloud users, not your customized health report recipients.

Select Receiving Period/Channel

Receiving Channel * ☑ Message Center ☑ Email ☑ SMS ☑ WeChat ☑ Call ☑ WeCom

Receiving Period 00:00 ~ 23:59

Select Recipient/Recipient Group You can select recipient and recipient group at the same time

Recipient(1) | Recipient Group(0)

Select Recipient (16 in total)

Search by recipient

	Username	Mobile	Email
☐	v[redacted]	[redacted]	
☐	r[redacted]	[redacted]	[redacted]q...
☐	v[redacted]	[redacted]	[redacted]q....
☐	y[redacted]	[redacted]	
☑	r[redacted]	[redacted]	
☐	c[redacted]	[redacted]	[redacted]q...

Support for holding shift key down for multiple selection

Selected 1

Username	Mobile	Email
n[redacted]	[redacted]	.

OK Cancel

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.

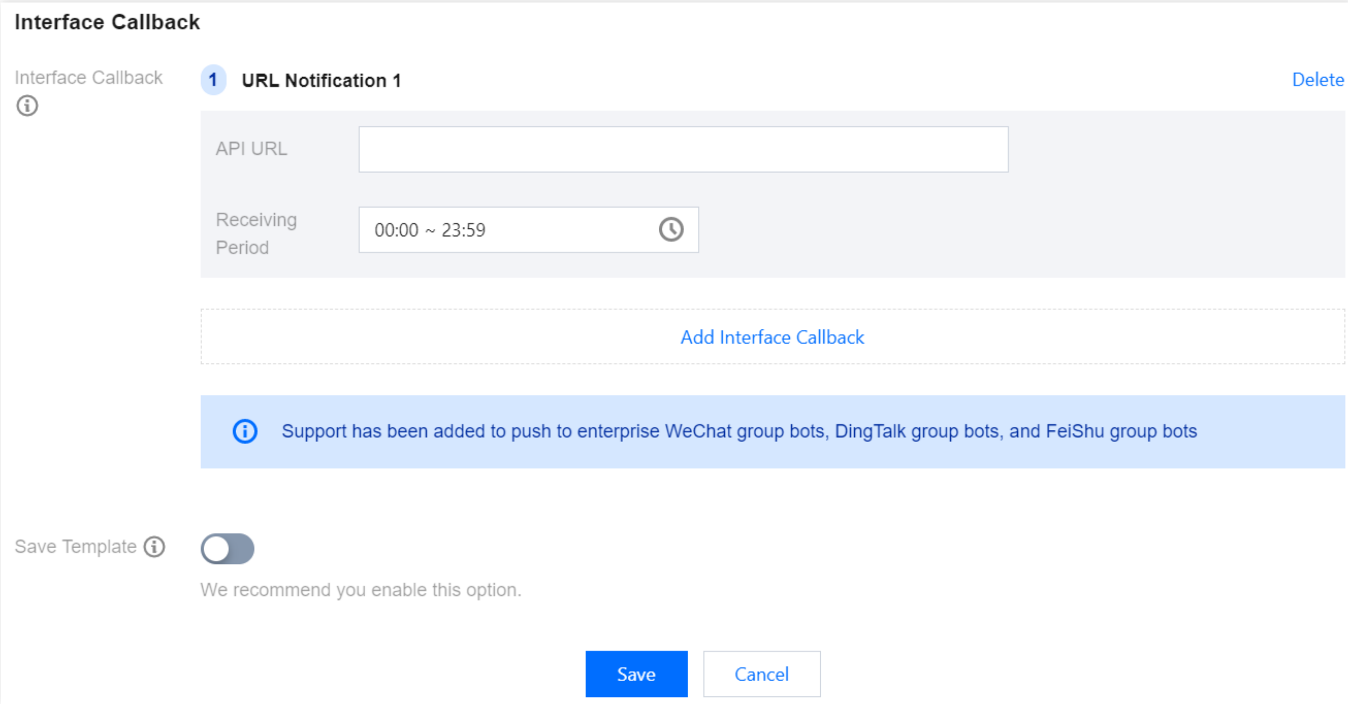
Note:

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.



Interface Callback

Interface Callback ⓘ

1 URL Notification 1 Delete

API URL

Receiving Period 00:00 ~ 23:59

Add Interface Callback

ⓘ Support has been added to push to enterprise WeChat group bots, DingTalk group bots, and FeiShu group bots

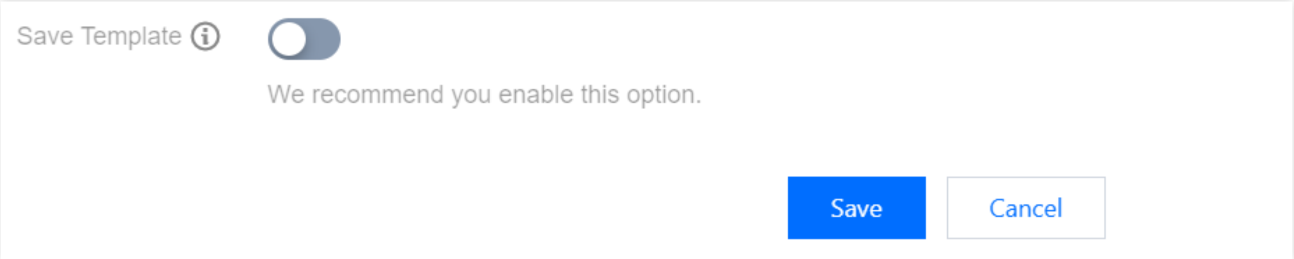
Save Template ⓘ

We recommend you enable this option.

Save Cancel

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** .



Save Template ⓘ

We recommend you enable this option.

Save Cancel

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.

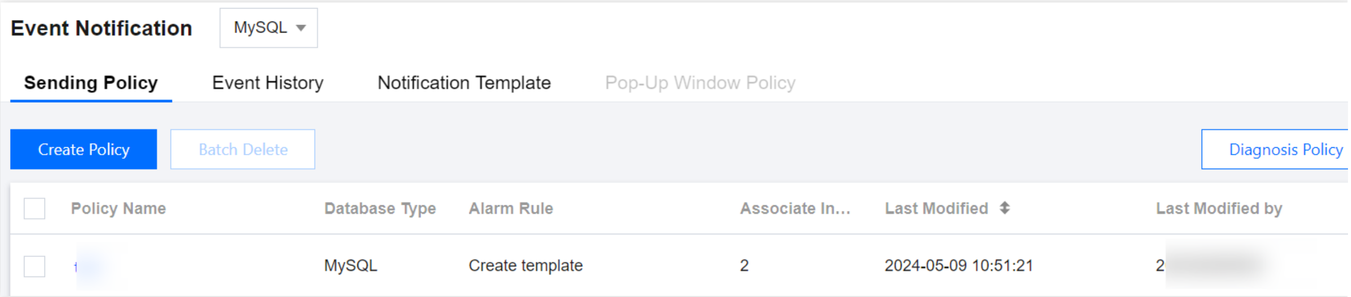
If successfully saved, the policy list will display the name of the newly created policy and the policy will be enabled by default.

Managing an Event Notification Policy

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

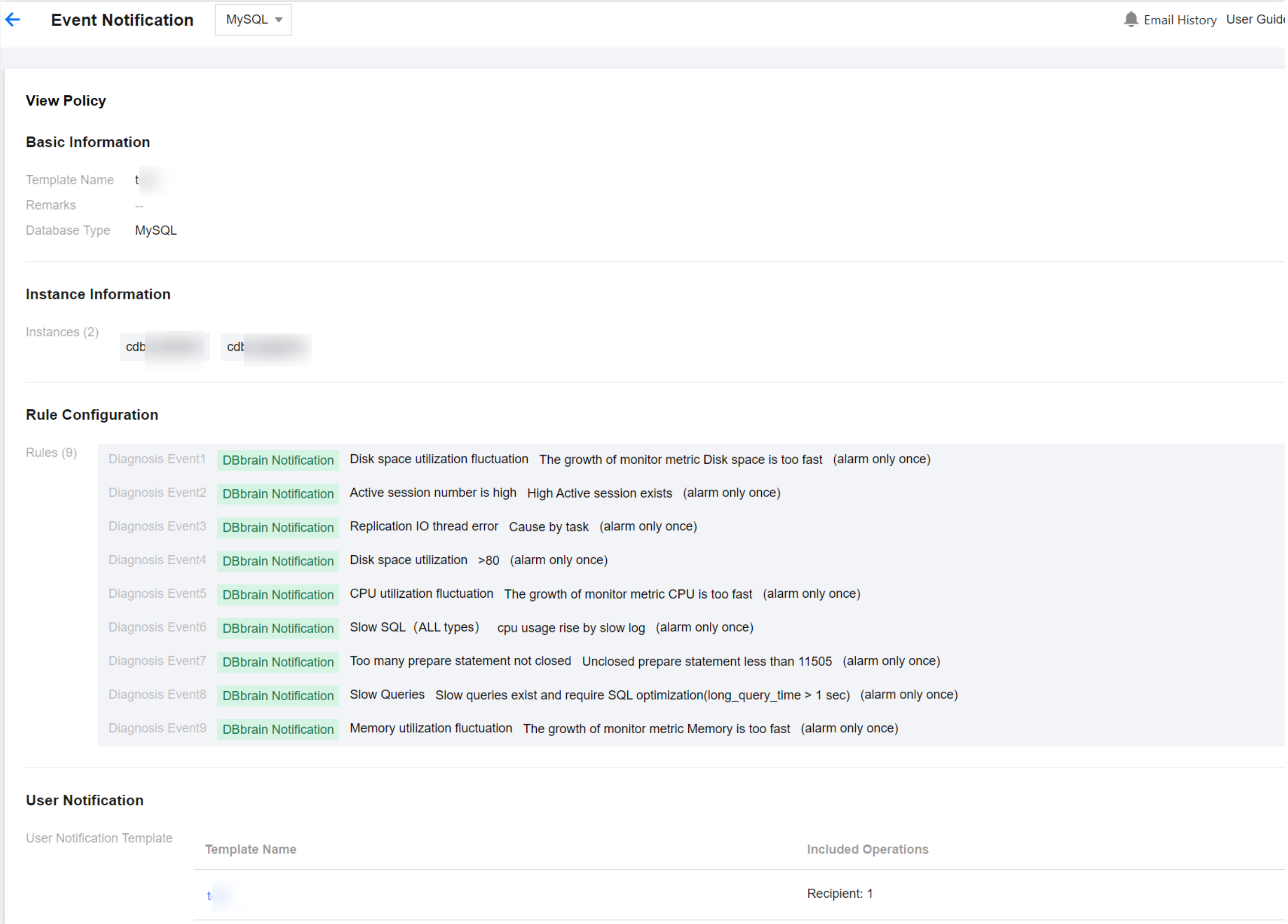
Viewing Policy Details

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.



□	Policy Name	Database Type	Alarm Rule	Associate In...	Last Modified	Last Modified by
□	t[redacted]	MySQL	Create template	2	2024-05-09 10:51:21	2[redacted]

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



View Policy

Basic Information

Template Name t[redacted]
Remarks --
Database Type MySQL

Instance Information

Instances (2) cdb[redacted] cdb[redacted]

Rule Configuration

Rules (9)

Diagnosis Event1	DBbrain Notification	Disk space utilization fluctuation	The growth of monitor metric Disk space is too fast	(alarm only once)
Diagnosis Event2	DBbrain Notification	Active session number is high	High Active session exists	(alarm only once)
Diagnosis Event3	DBbrain Notification	Replication IO thread error	Cause by task	(alarm only once)
Diagnosis Event4	DBbrain Notification	Disk space utilization	>80	(alarm only once)
Diagnosis Event5	DBbrain Notification	CPU utilization fluctuation	The growth of monitor metric CPU is too fast	(alarm only once)
Diagnosis Event6	DBbrain Notification	Slow SQL (ALL types)	cpu usage rise by slow log	(alarm only once)
Diagnosis Event7	DBbrain Notification	Too many prepare statement not closed	Unclosed prepare statement less than 11505	(alarm only once)
Diagnosis Event8	DBbrain Notification	Slow Queries	Slow queries exist and require SQL optimization(long_query_time > 1 sec)	(alarm only once)
Diagnosis Event9	DBbrain Notification	Memory utilization fluctuation	The growth of monitor metric Memory is too fast	(alarm only once)

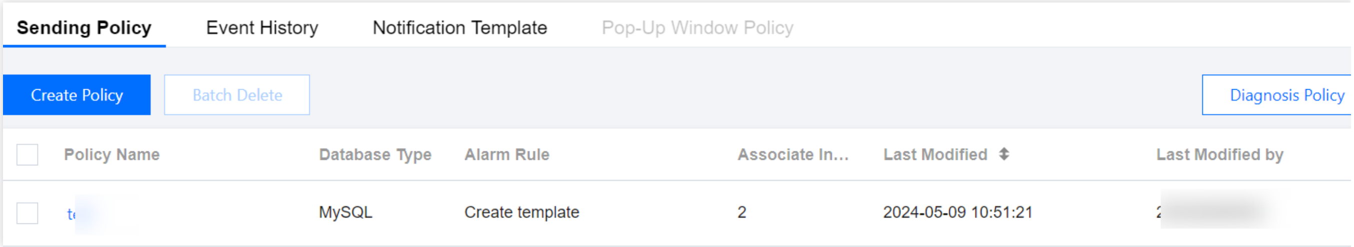
User Notification

User Notification Template

Template Name	Included Operations
t[redacted]	Recipient: 1

Disabling or Enabling Policies

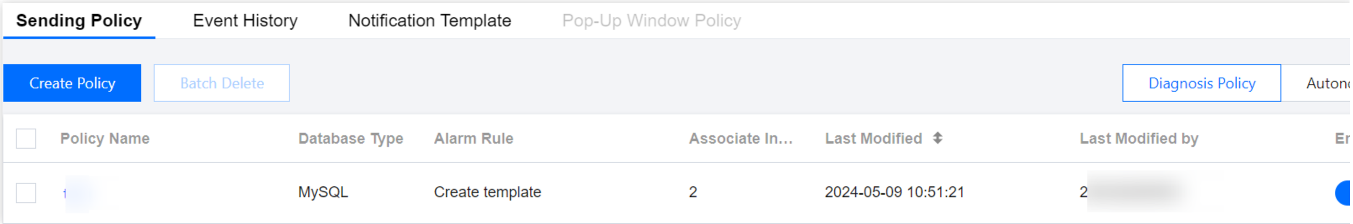
By default, a policy is enabled after created. If you need to disable this policy, turn off the switch in the **Enable** column in the policy list of the **Sending Policy** tab. To enable it again, simply turn on the switch back.



□	Policy Name	Database Type	Alarm Rule	Associate In...	Last Modified	Last Modified by
□	[redacted]	MySQL	Create template	2	2024-05-09 10:51:21	[redacted]

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.



□	Policy Name	Database Type	Alarm Rule	Associate In...	Last Modified	Last Modified by	En
□	[redacted]	MySQL	Create template	2	2024-05-09 10:51:21	[redacted]	●

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.

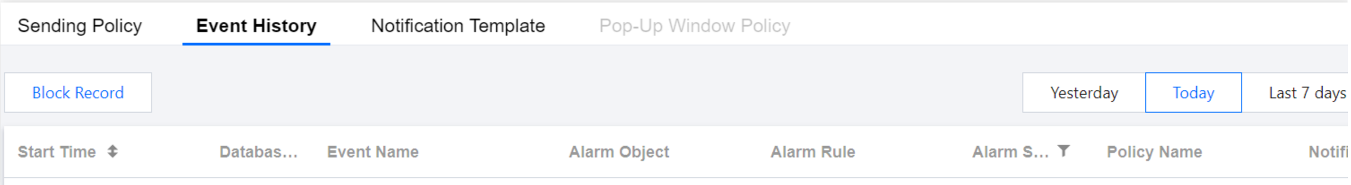
Single deletion: In the **Sending Policy** tab, find the name of the policy to be deleted, and in the corresponding **Operation** bar, click **Delete** .

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** .

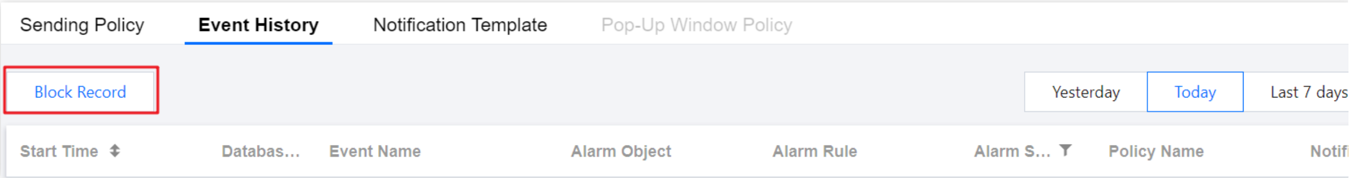
3. On the top of the page, select database type. Select the **Event History** tab, and on the top right of the page, select the time range to view the history of event notifications. The default interface displays the events of the day.



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.

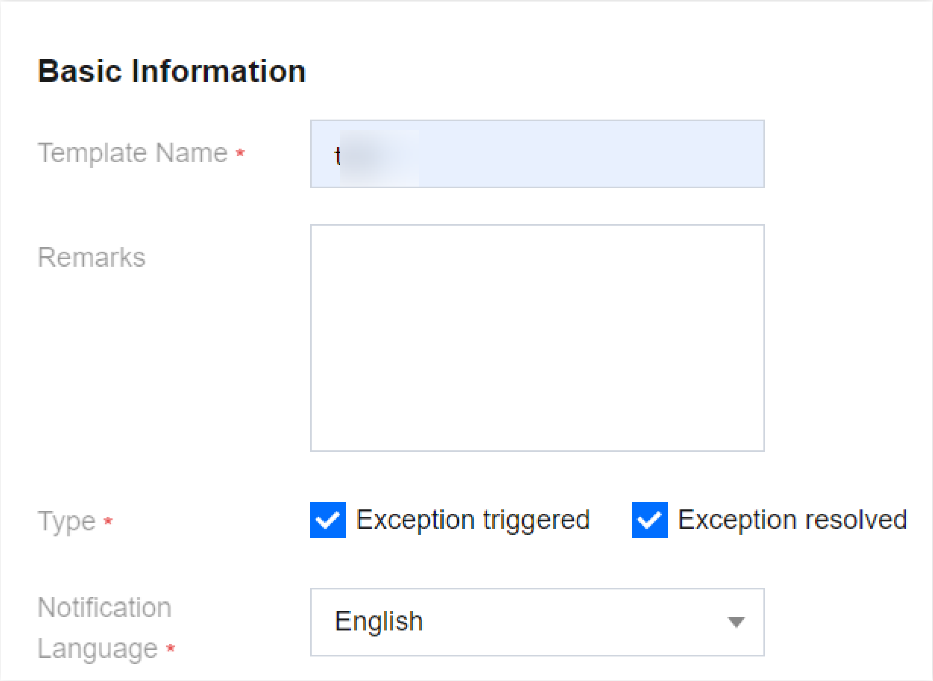
The notification template supports the following two methods:

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.

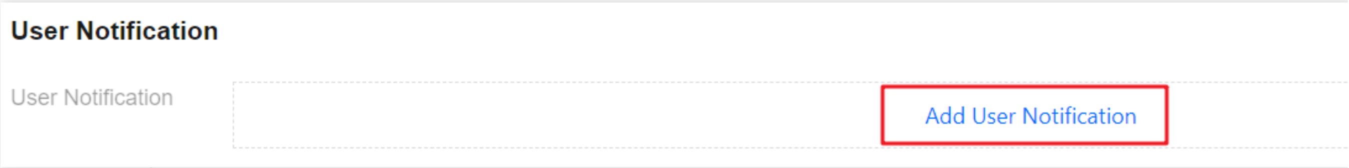


Basic Information

Template Name *	t
Remarks	
Type *	☑ Exception triggered ☑ Exception resolved
Notification Language *	English ▾

5. Configure user notifications.

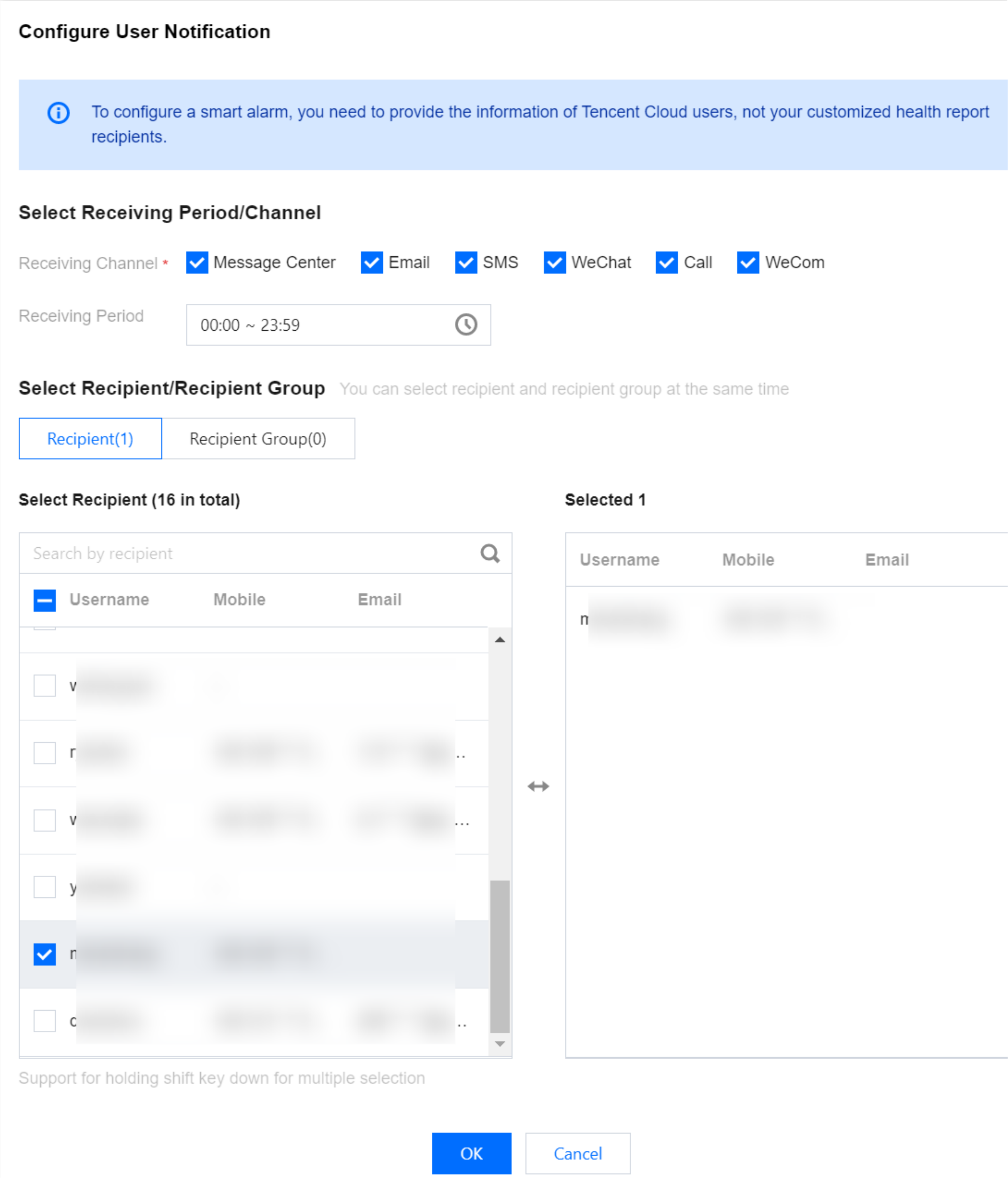
5.1 In the **User Notification** section, click **Add User Notification** .



User Notification

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** .



Configure User Notification

ⓘ To configure a smart alarm, you need to provide the information of Tencent Cloud users, not your customized health report recipients.

Select Receiving Period/Channel

Receiving Channel * ☑ Message Center ☑ Email ☑ SMS ☑ WeChat ☑ Call ☑ WeCom

Receiving Period: 00:00 ~ 23:59

Select Recipient/Recipient Group You can select recipient and recipient group at the same time

Recipient(1) | Recipient Group(0)

Select Recipient (16 in total)

Search by recipient

	Username	Mobile	Email
□	[redacted]	[redacted]	[redacted]
□	[redacted]	[redacted]	[redacted]
□	[redacted]	[redacted]	[redacted]
□	[redacted]	[redacted]	[redacted]
☑	[redacted]	[redacted]	[redacted]
□	[redacted]	[redacted]	[redacted]

Selected 1

Username	Mobile	Email
[redacted]	[redacted]	

Support for holding shift key down for multiple selection

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.

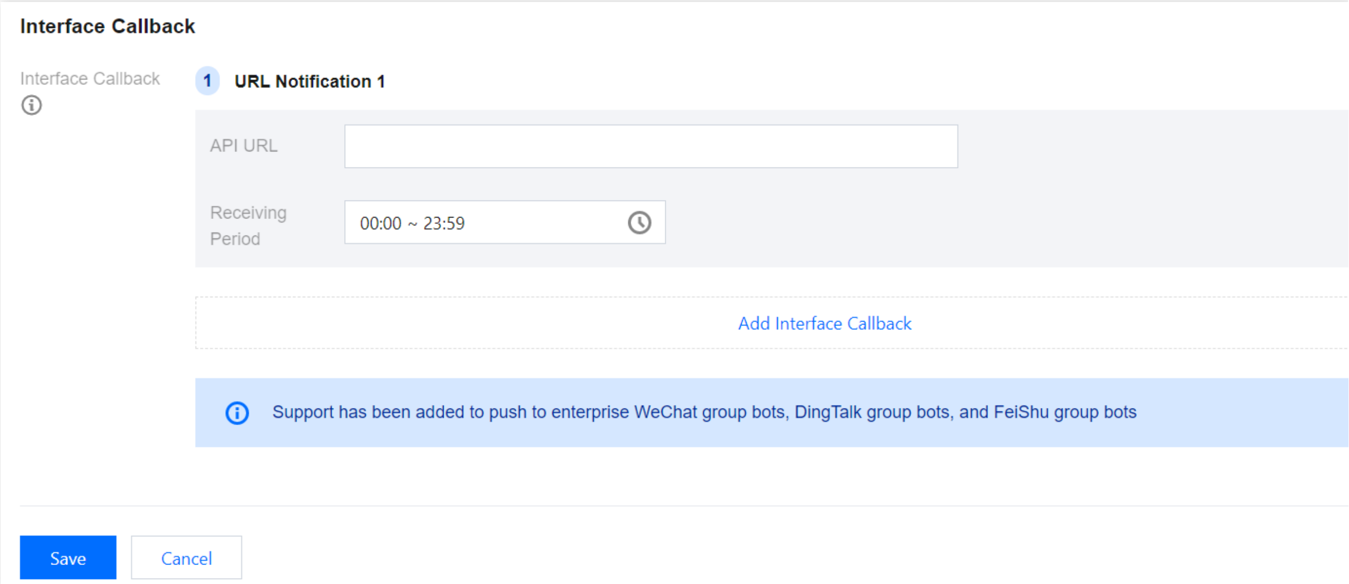
Note:

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.



Interface Callback

Interface Callback ⓘ

1 URL Notification 1

API URL:

Receiving Period: 00:00 ~ 23:59

Add Interface Callback

ⓘ Support has been added to push to enterprise WeChat group bots, DingTalk group bots, and FeiShu group bots

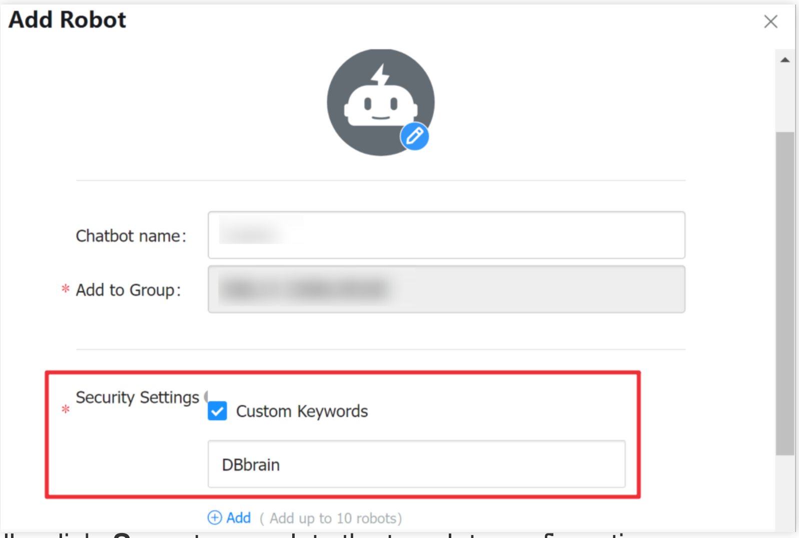
Save Cancel

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.



Add Robot

Chatbot name: [redacted]

* Add to Group: [redacted]

* Security Settings (☑ Custom Keywords

DBbrain

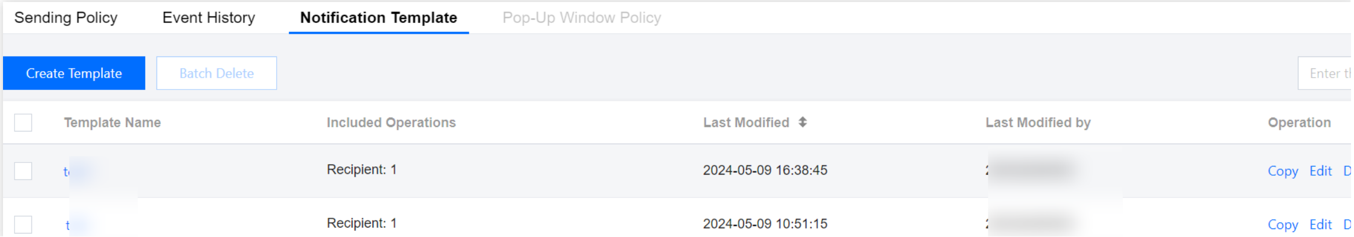
⊕ Add (Add up to 10 robots)

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.



□	Template Name	Included Operations	Last Modified	Last Modified by	Operation
□	t[redacted]	Recipient: 1	2024-05-09 16:38:45	2[redacted]	Copy Edit D
□	t[redacted]	Recipient: 1	2024-05-09 10:51:15	2[redacted]	Copy Edit D

Intelligent Monitoring (Monitoring Dashboard)

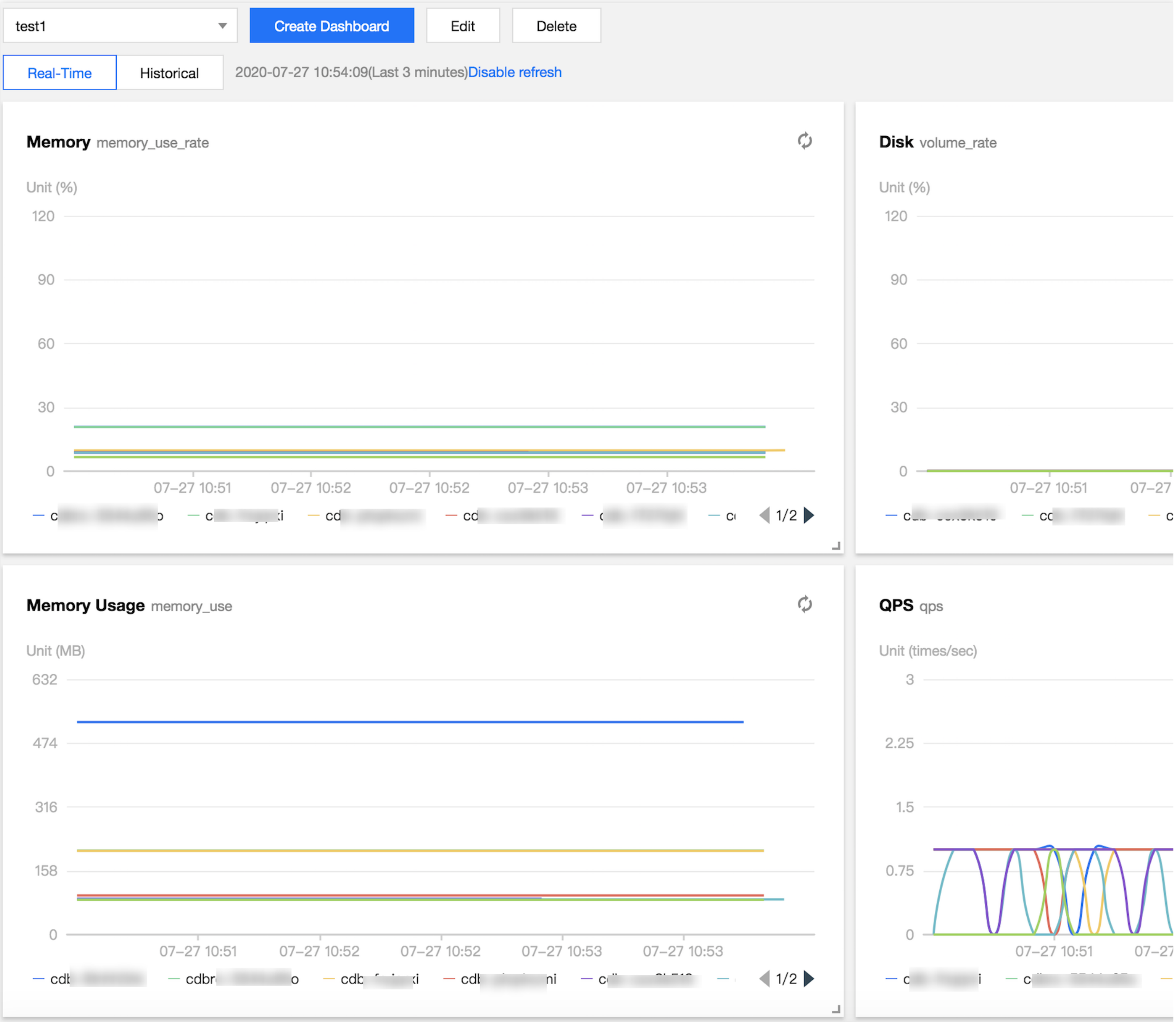
Last updated：2022-09-01 18:34:46

Feature Description

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

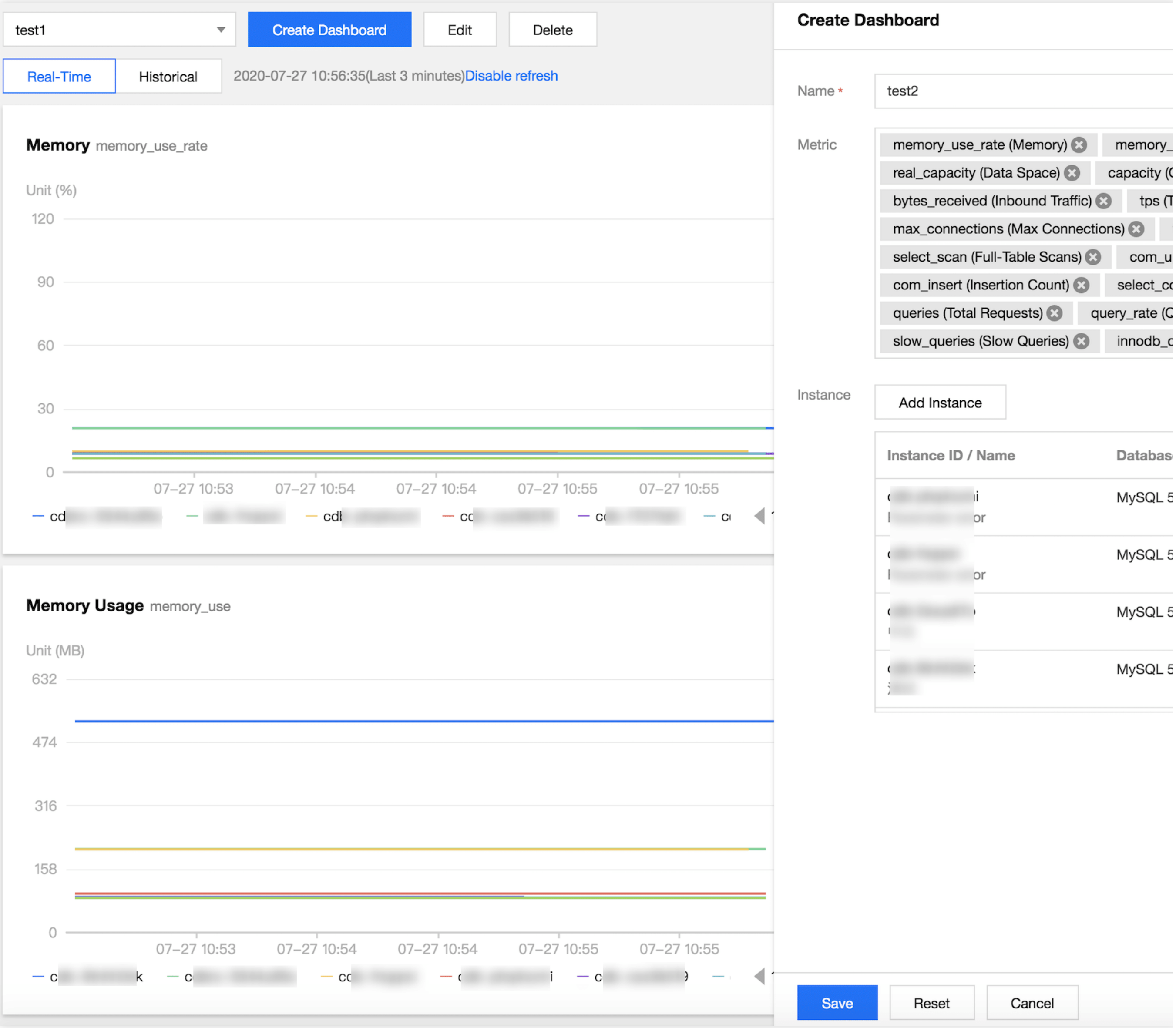
Note:

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**.



Create Dashboard

Name * test2

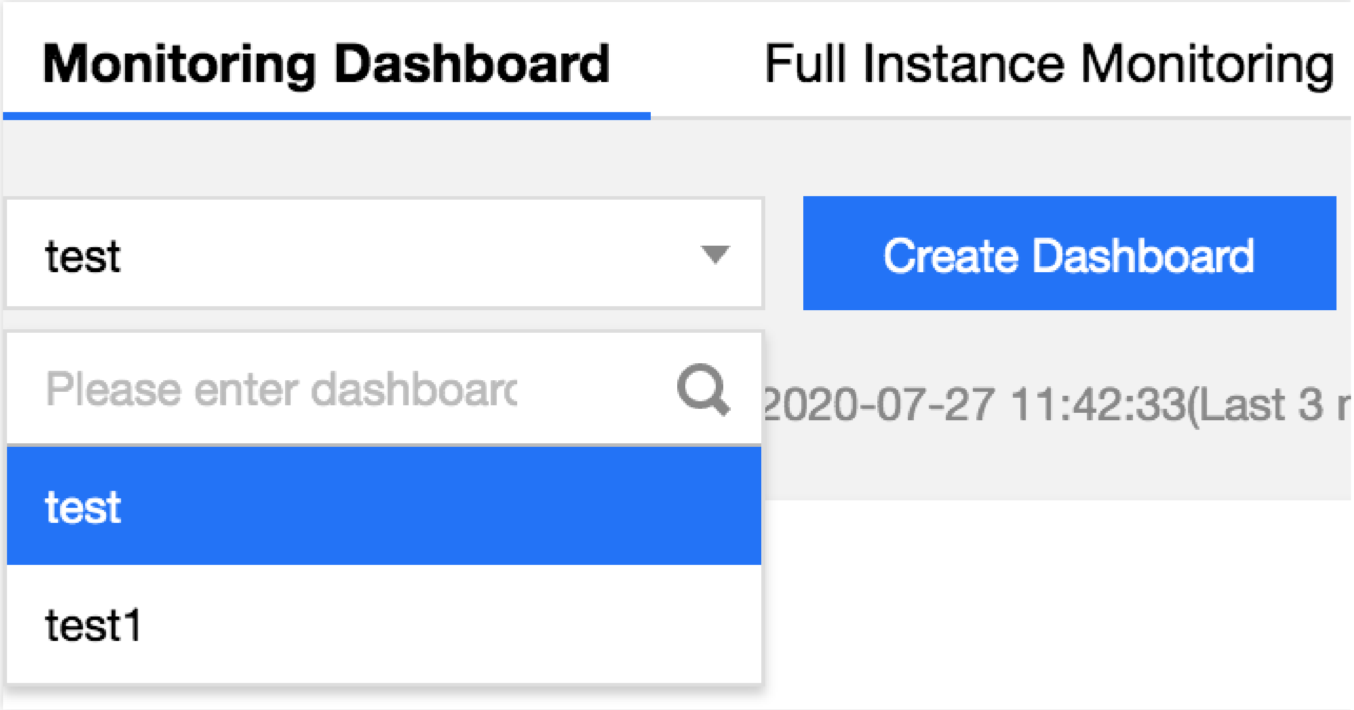
Metric: memory_use_rate (Memory), memory_, real_capacity (Data Space), capacity (, bytes_received (Inbound Traffic), tps (T, max_connections (Max Connections), select_scan (Full-Table Scans), com_u, com_insert (Insertion Count), select_c, queries (Total Requests), query_rate (C, slow_queries (Slow Queries), innodb_c

Instance

Instance ID / Name	Databas
	MySQL 5
	MySQL 5
	MySQL 5
	MySQL 5

Finding/Editing/Deleting a Dashboard

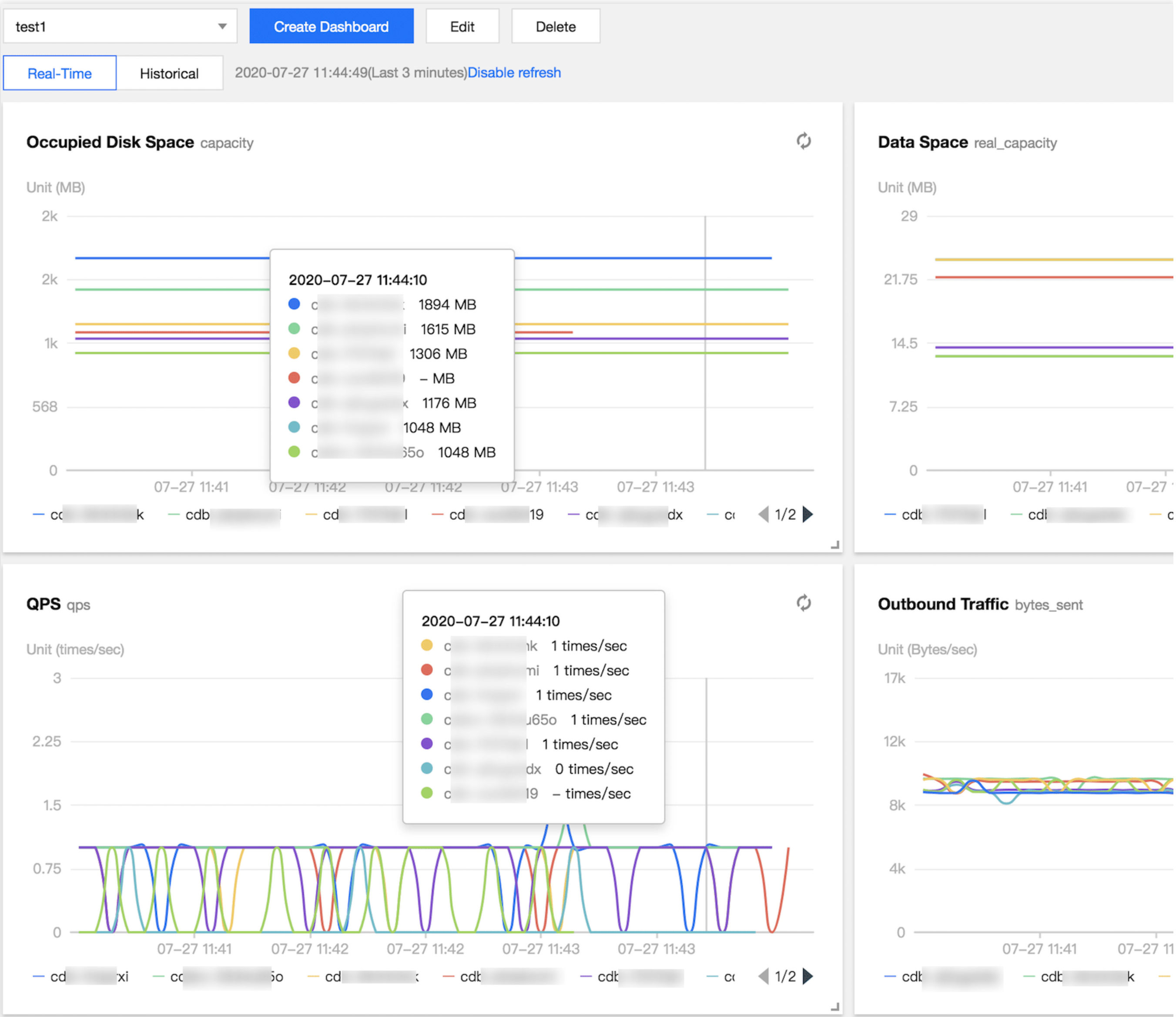
Log in to the DBbrain console and select **Monitoring and Alarming** > **Intelligent Monitoring** on the left sidebar. On the displayed page, select a database type and an instance at the top, and select the **Monitoring Dashboard** tab.
Click the dashboard name drop-down list to switch between different 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**.



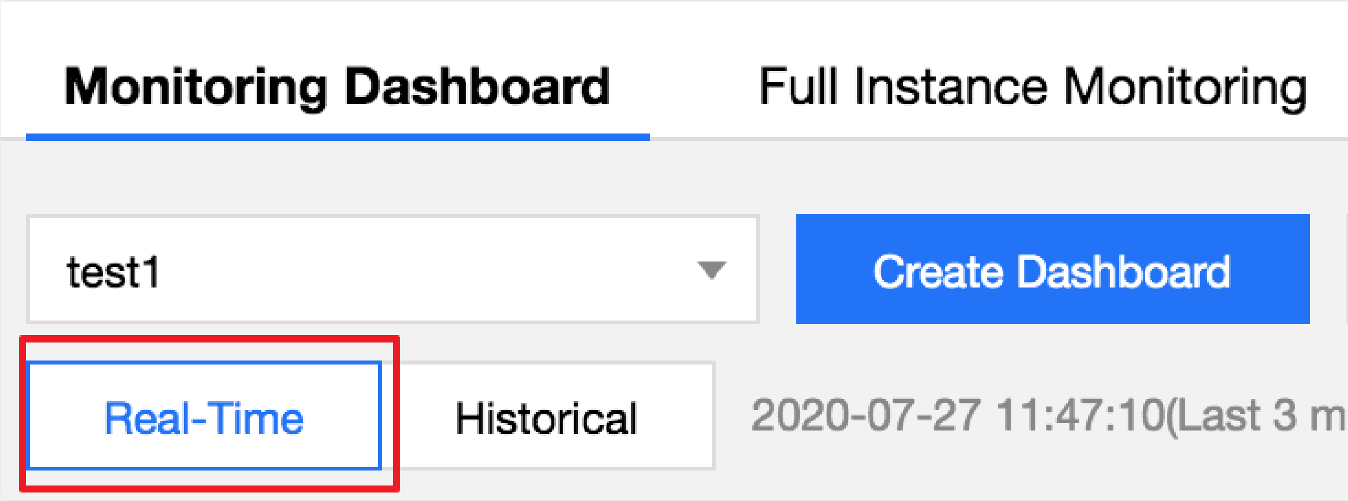
Switching between one-column and two-column modes

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.

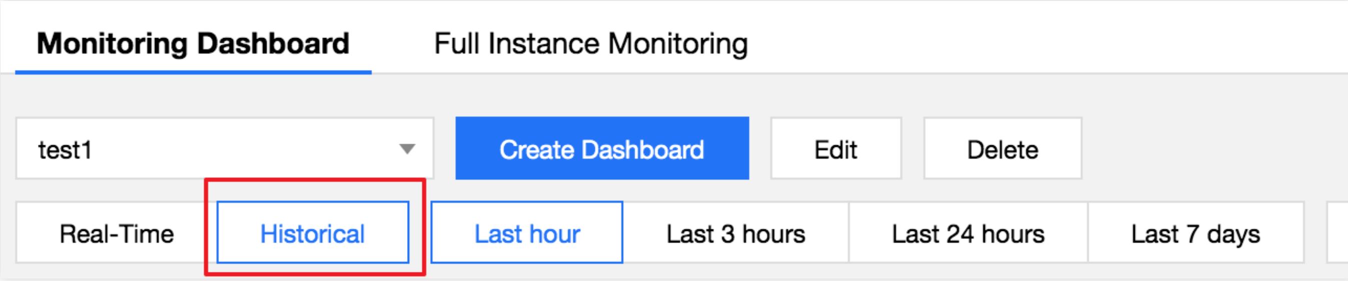
Switching between real-time and historical views

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

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

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

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

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

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	✓	×

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	✓	✓
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	✓	✓
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	✓	×

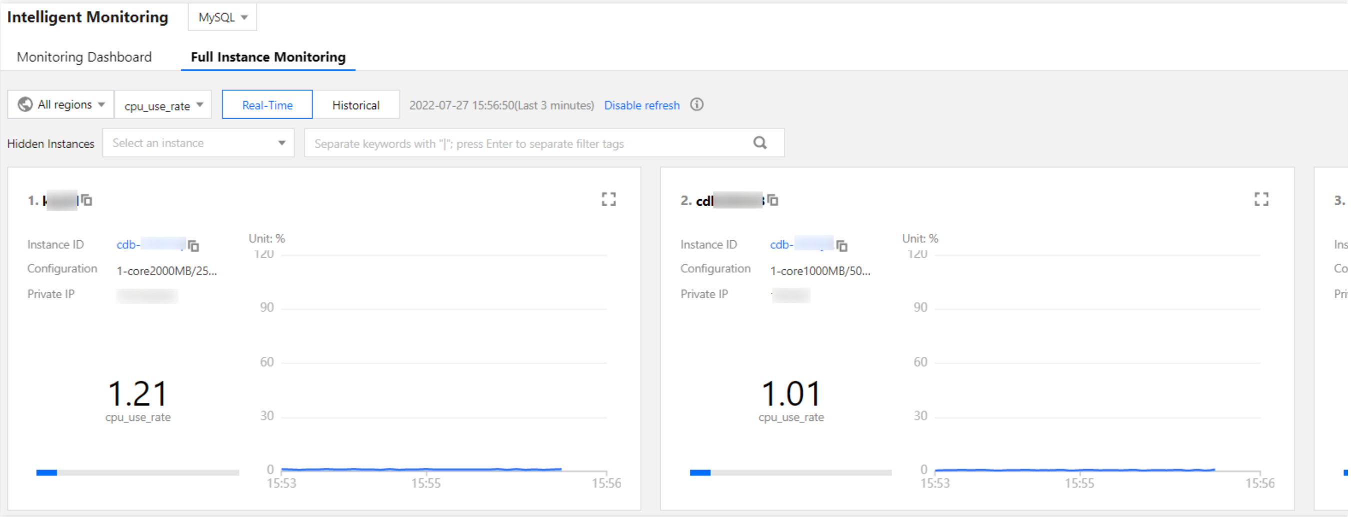
Intelligent Monitoring (Full Instance Monitoring)

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.



2022-07-27 15:56:50(Last 3 minutes)

Separate keywords with "|"; press Enter to separate filter tags

1. [redacted]

Instance ID cdb-[redacted]

Configuration 1-core2000MB/25...

Private IP [redacted]

1.21 cpu_use_rate

2. cd[redacted]

Instance ID cdb-[redacted]

Configuration 1-core1000MB/50...

Private IP [redacted]

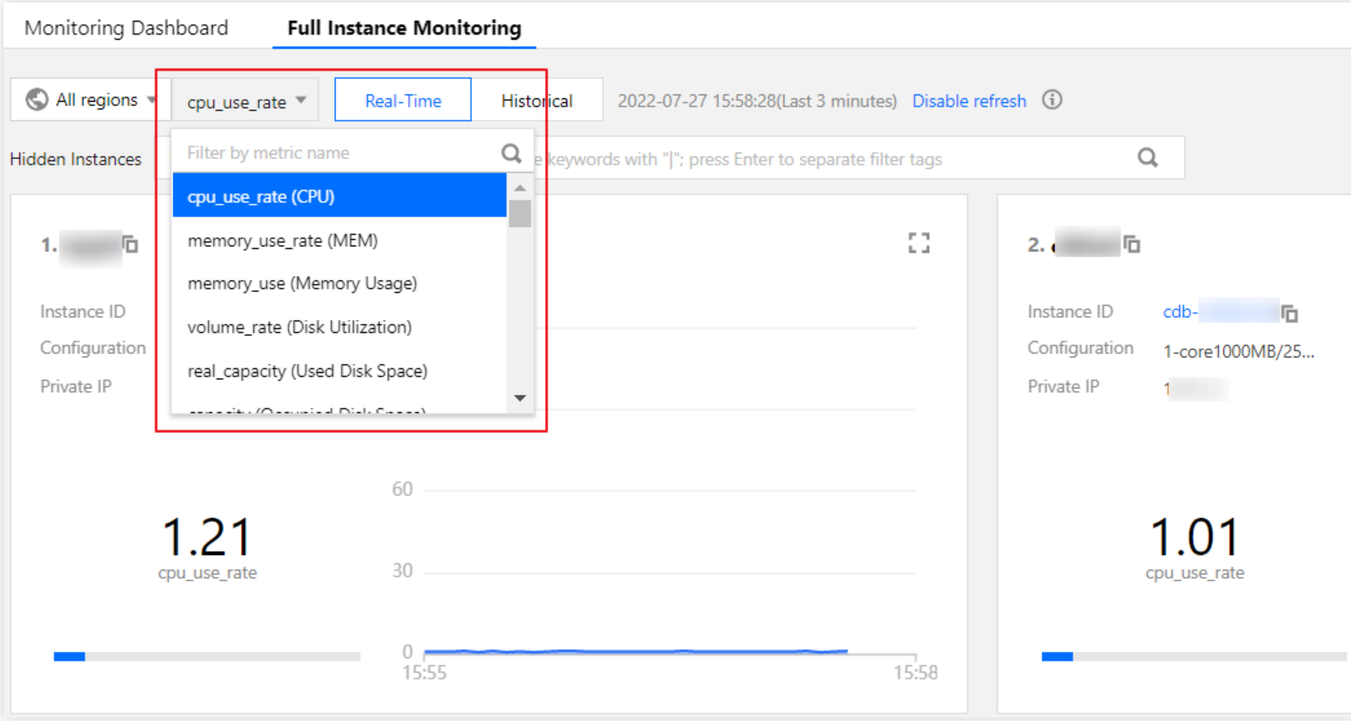
1.01 cpu_use_rate

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.

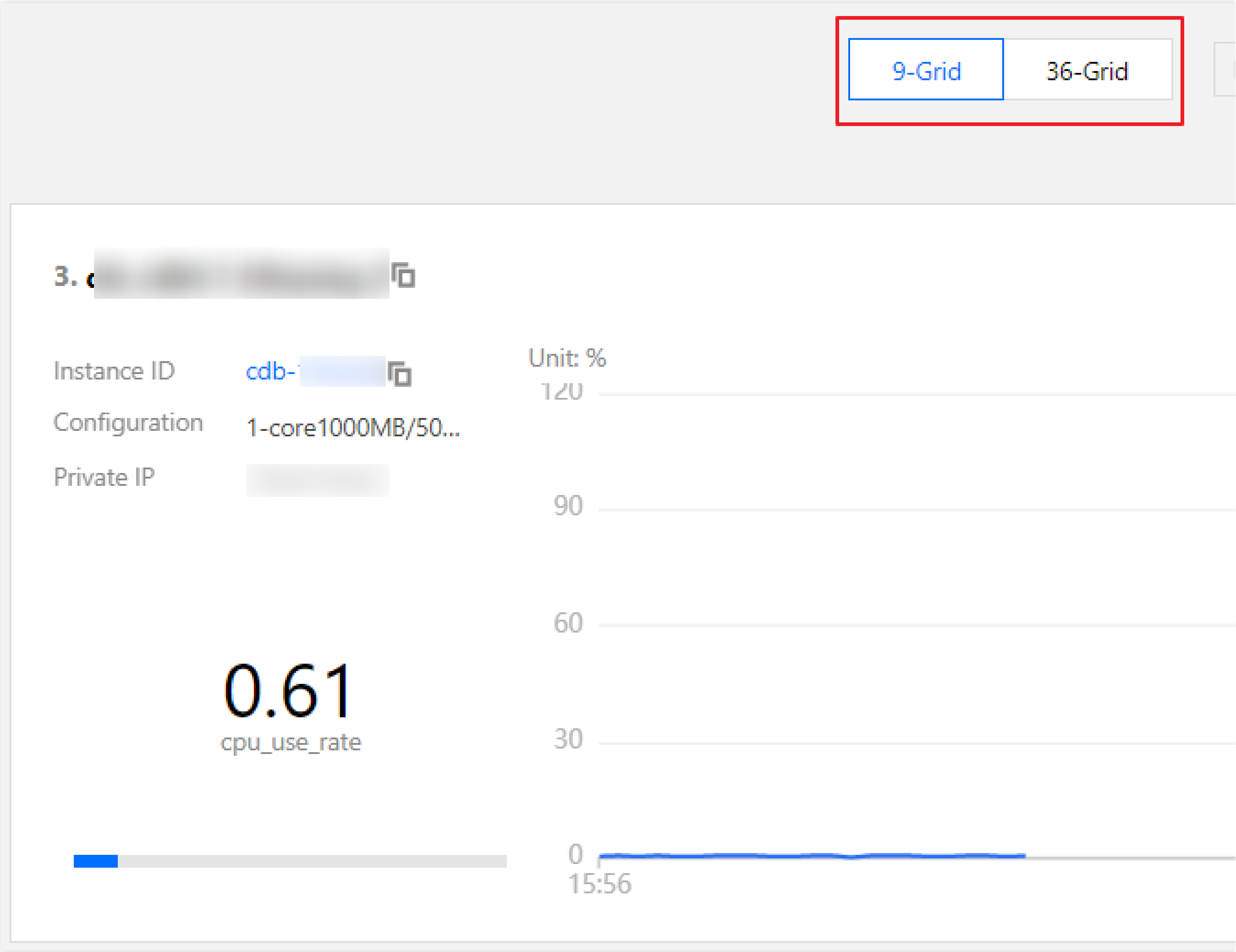
Searching for an instance

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

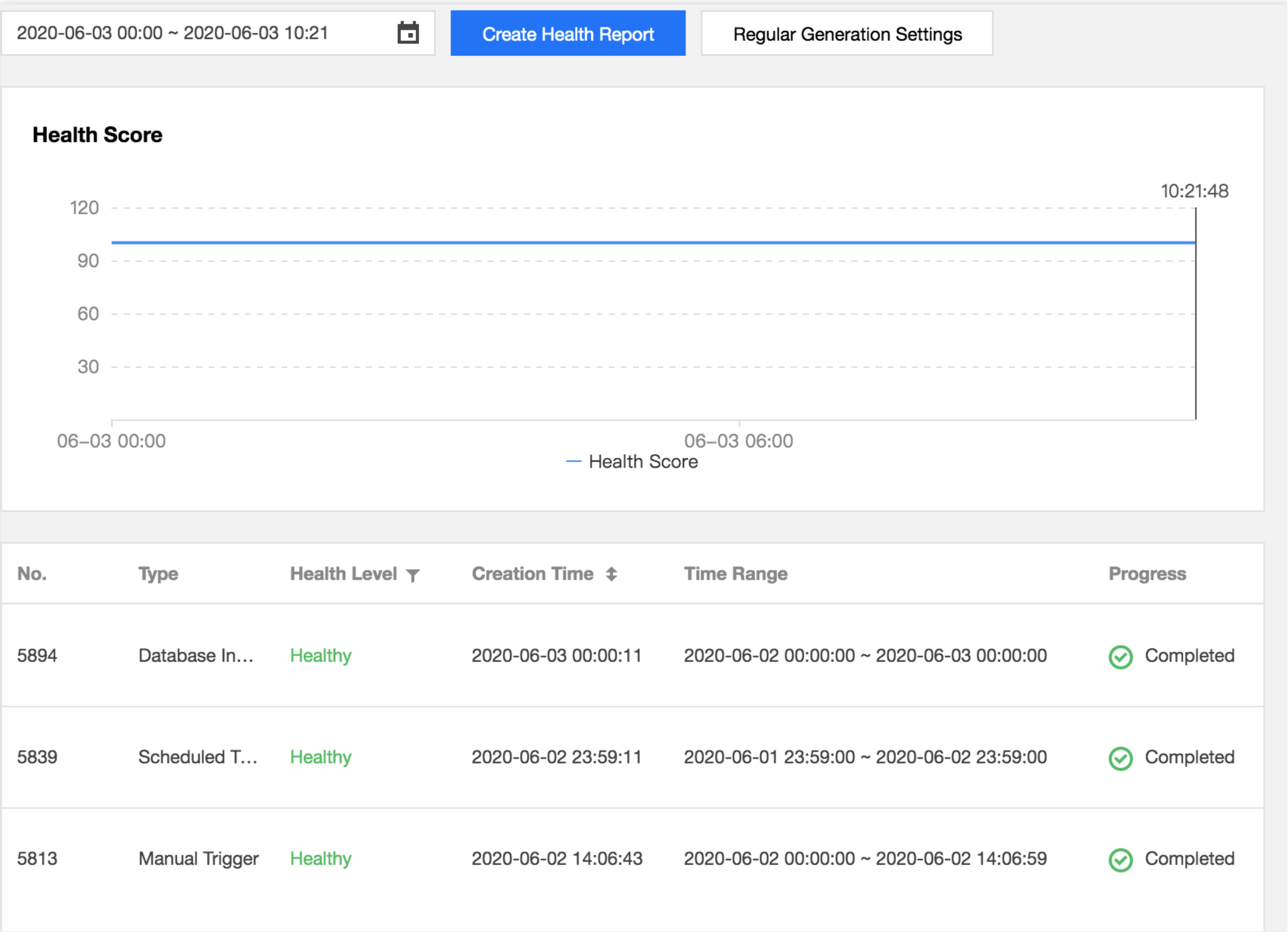
Log in to the DBbrain console and select **Performance Optimization** on the left sidebar. On the displayed page, select a database at the top and select the **Health Report** tab. You can view the health score trends and the problem overview for the specified time period.

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.



No.	Type	Health Level	Creation Time	Time Range	Progress
5894	Database In...	Healthy	2020-06-03 00:00:11	2020-06-02 00:00:00 ~ 2020-06-03 00:00:00	✓ Completed
5839	Scheduled T...	Healthy	2020-06-02 23:59:11	2020-06-01 23:59:00 ~ 2020-06-02 23:59:00	✓ Completed
5813	Manual Trigger	Healthy	2020-06-02 14:06:43	2020-06-02 00:00:00 ~ 2020-06-02 14:06:59	✓ Completed

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.

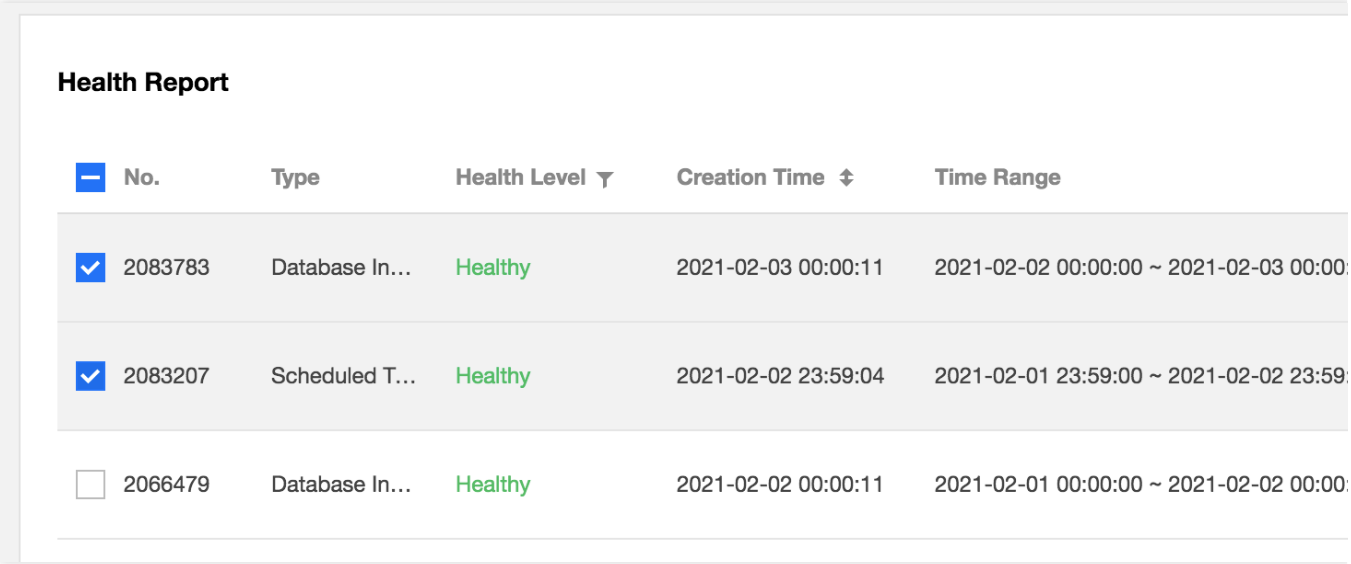
The **Type** column displays how the report is generated, including being generated manually, as scheduled, or in an database inspection.

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.

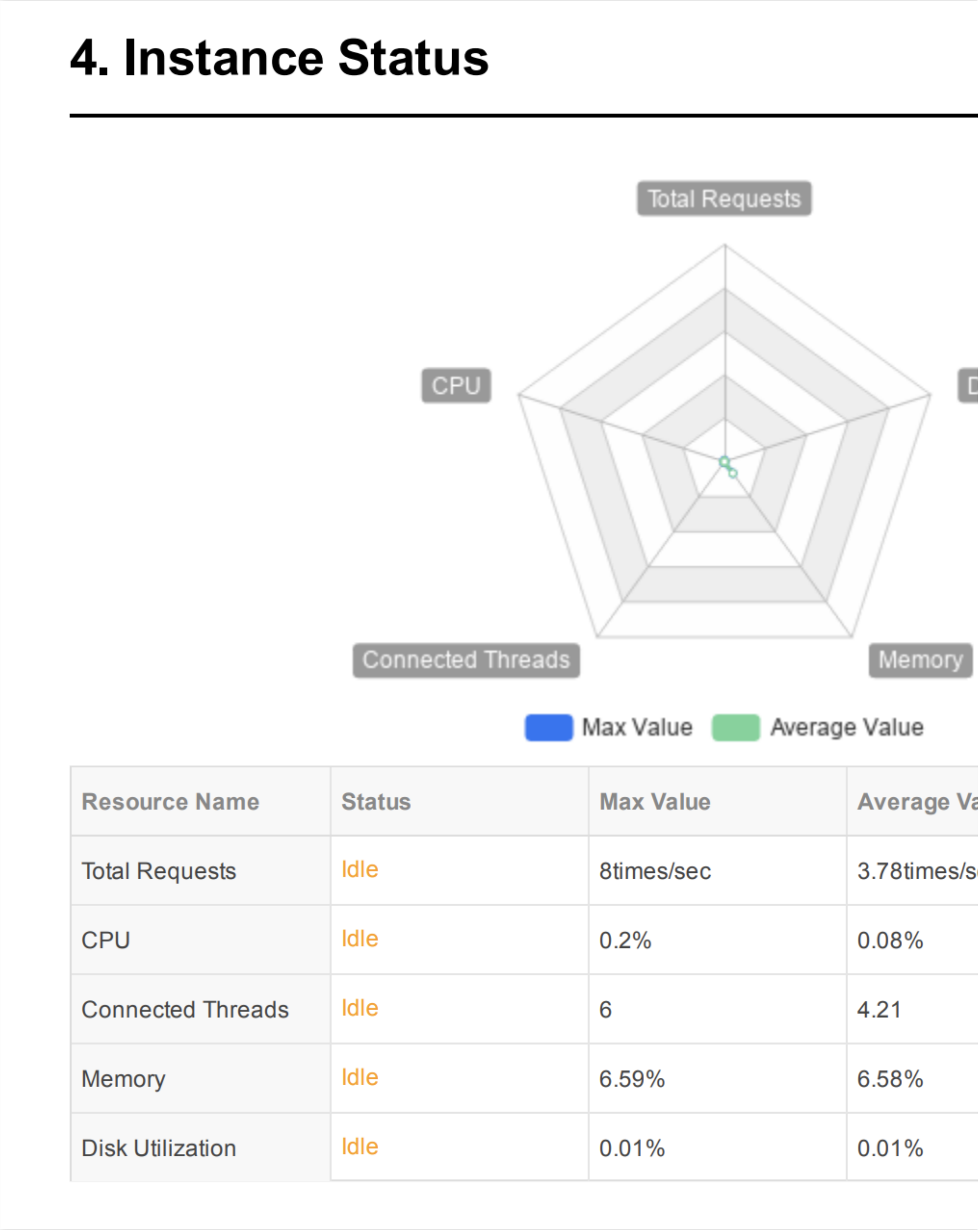


Health Report

	No.	Type	Health Level	Creation Time	Time Range
☑	2083783	Database In...	Healthy	2021-02-03 00:00:11	2021-02-02 00:00:00 ~ 2021-02-03 00:00:
☑	2083207	Scheduled T...	Healthy	2021-02-02 23:59:04	2021-02-01 23:59:00 ~ 2021-02-02 23:59:
□	2066479	Database In...	Healthy	2021-02-02 00:00:11	2021-02-01 00:00:00 ~ 2021-02-02 00:00:

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.



4. Instance Status

Resource Name	Status	Max Value	Average Va
Total Requests	Idle	8times/sec	3.78times/s
CPU	Idle	0.2%	0.08%
Connected Threads	Idle	6	4.21
Memory	Idle	6.59%	6.58%
Disk Utilization	Idle	0.01%	0.01%

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

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**.

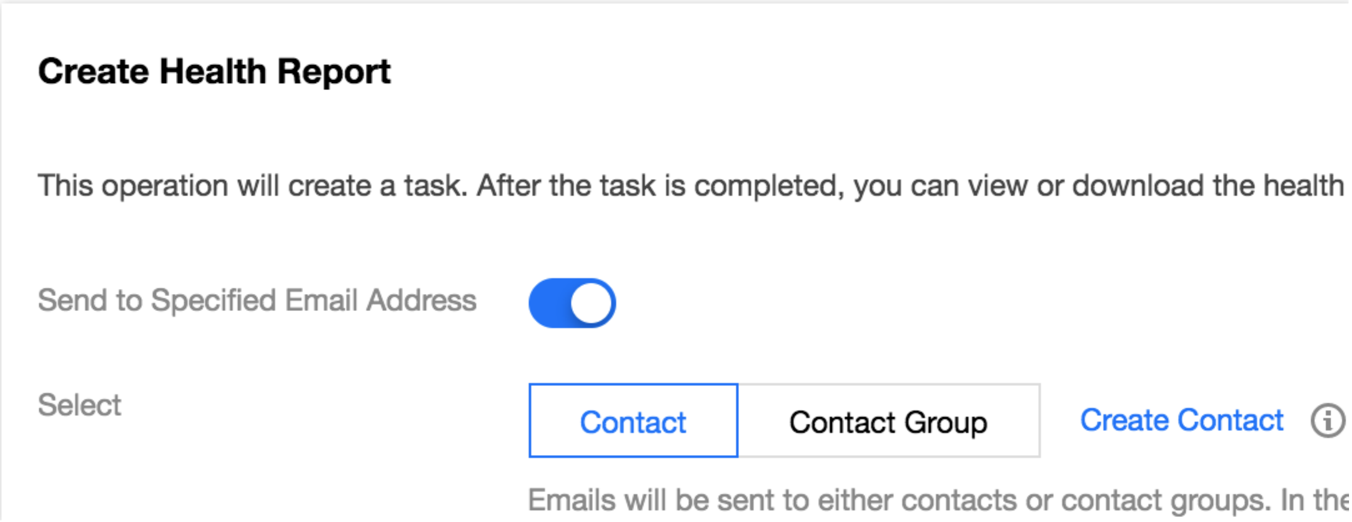
2. In the pop-up window, enable **Send to Specified Email Address**, select **Contact** or **Contact Group**, and click **Confirm**. The generated health reports will be sent to the email of the specified contact or contact group.

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.

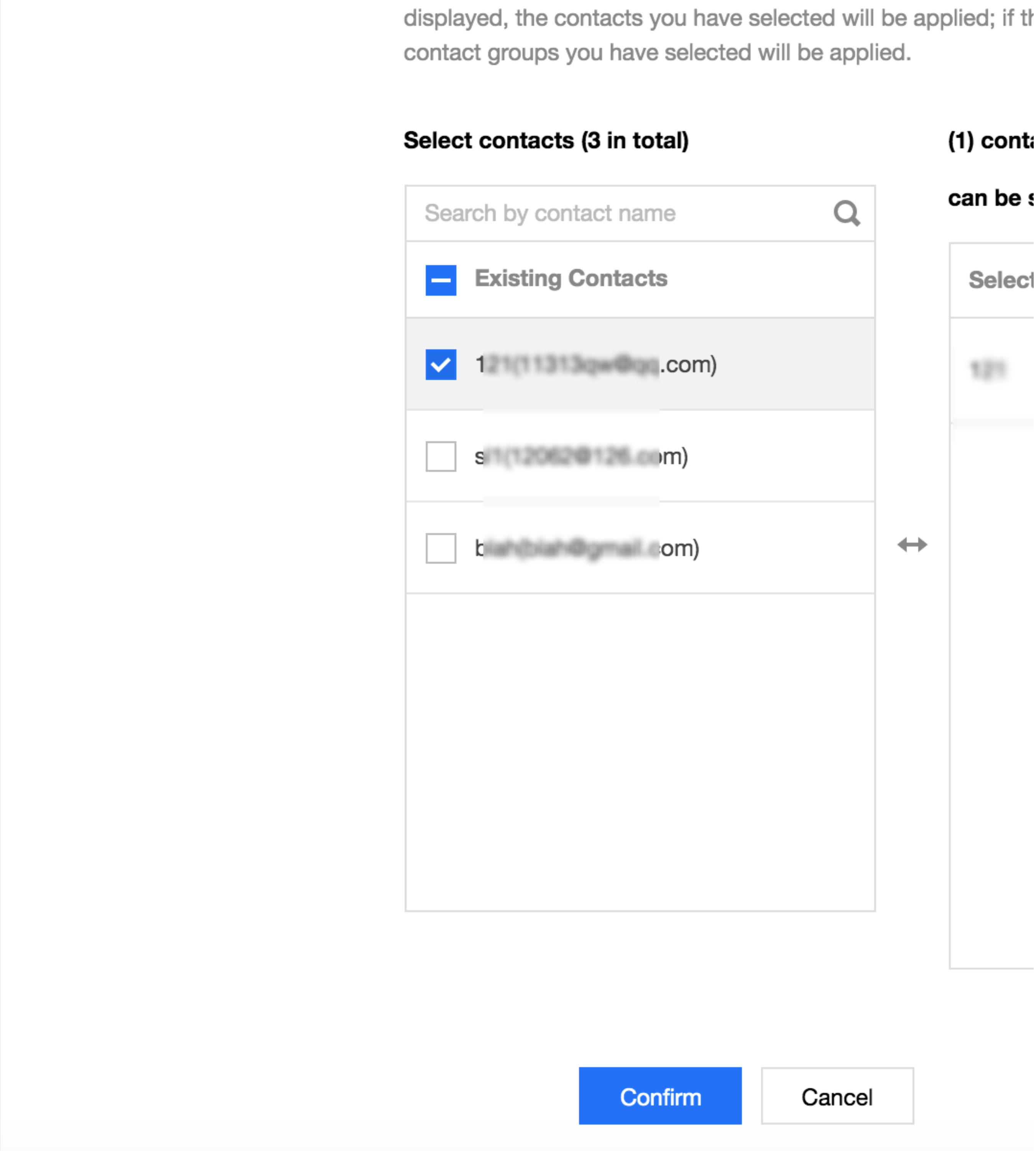


Create Health Report

This operation will create a task. After the task is completed, you can view or download the health

Send to Specified Email Address

Emails will be sent to either contacts or contact groups. In the



displayed, the contacts you have selected will be applied; if th
contact groups you have selected will be applied.

Select contacts (3 in total)

(1) cont
can be s

Search by contact name

☐ Existing Contacts

☑ 1[redacted].com)

☐ s[redacted]m)

☐ b[redacted]om)

Sending Health Report Generated by Scheduled Tasks via Email

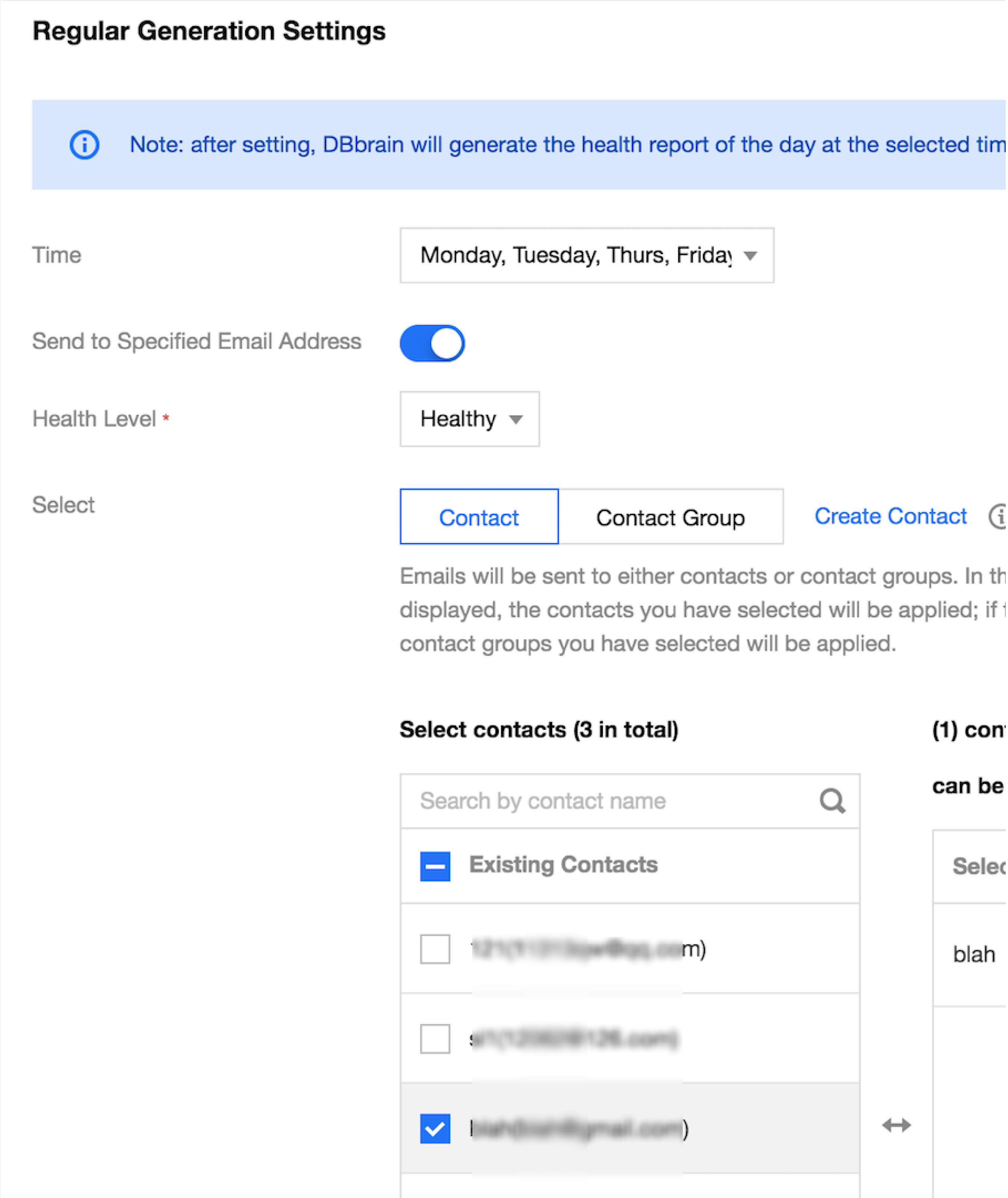
1. Log in to the DBbrain console, and select **Performance Optimization** on the left sidebar. On the displayed page, select a database type at the top, select the **Health Report** tab, and click **Regular Generation Settings**.
2. In the pop-up window, configure the time to generate the health report, 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.



Regular Generation Settings

Note: after setting, DBbrain will generate the health report of the day at the selected tim

Time: Monday, Tuesday, Thurs, Friday

Send to Specified Email Address: on

Health Level *: Healthy

Select: Contact | Contact Group | Create Contact

Emails will be sent to either contacts or contact groups. In th displayed, the contacts you have selected will be applied; if t contact groups you have selected will be applied.

Select contacts (3 in total)

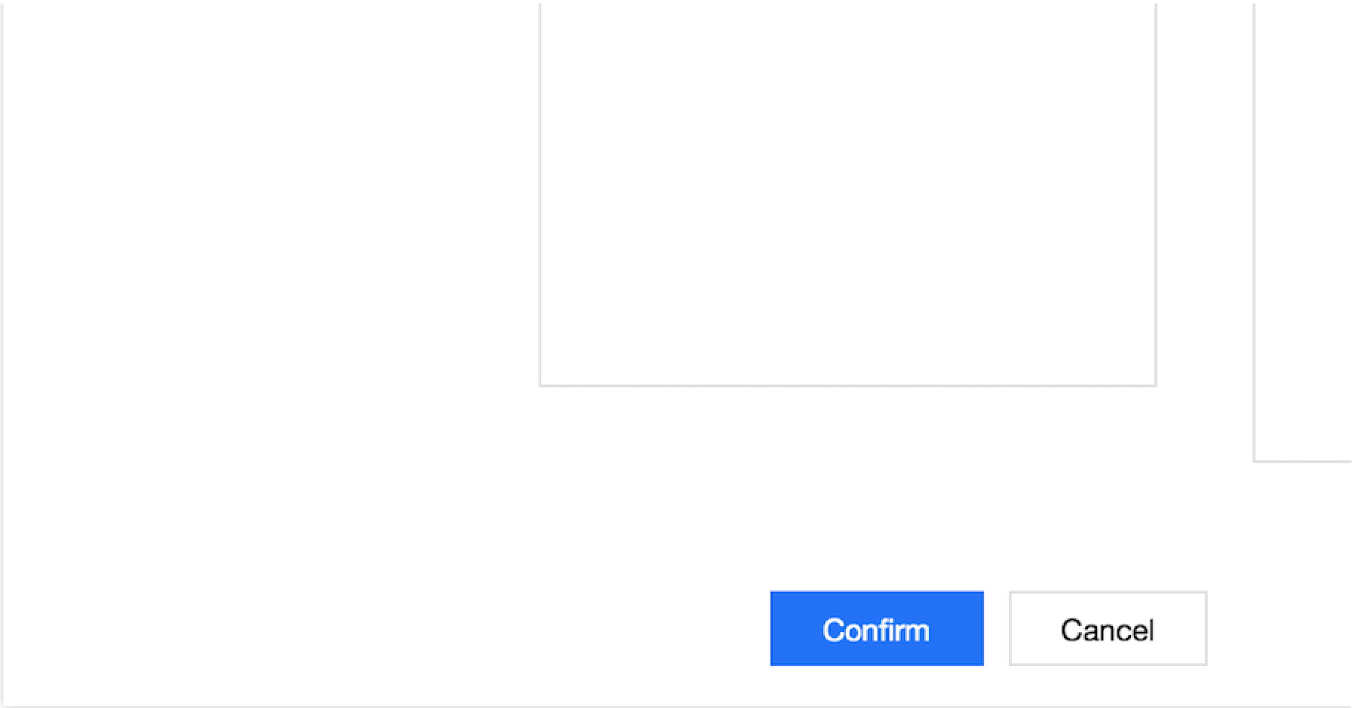
Search by contact name

- ☐ Existing Contacts
- ☐ [redacted]
- ☐ [redacted]
- ☑ [redacted]

(1) con can be

Selec

blah



Sending Health Report Generated by Database Inspection via Email

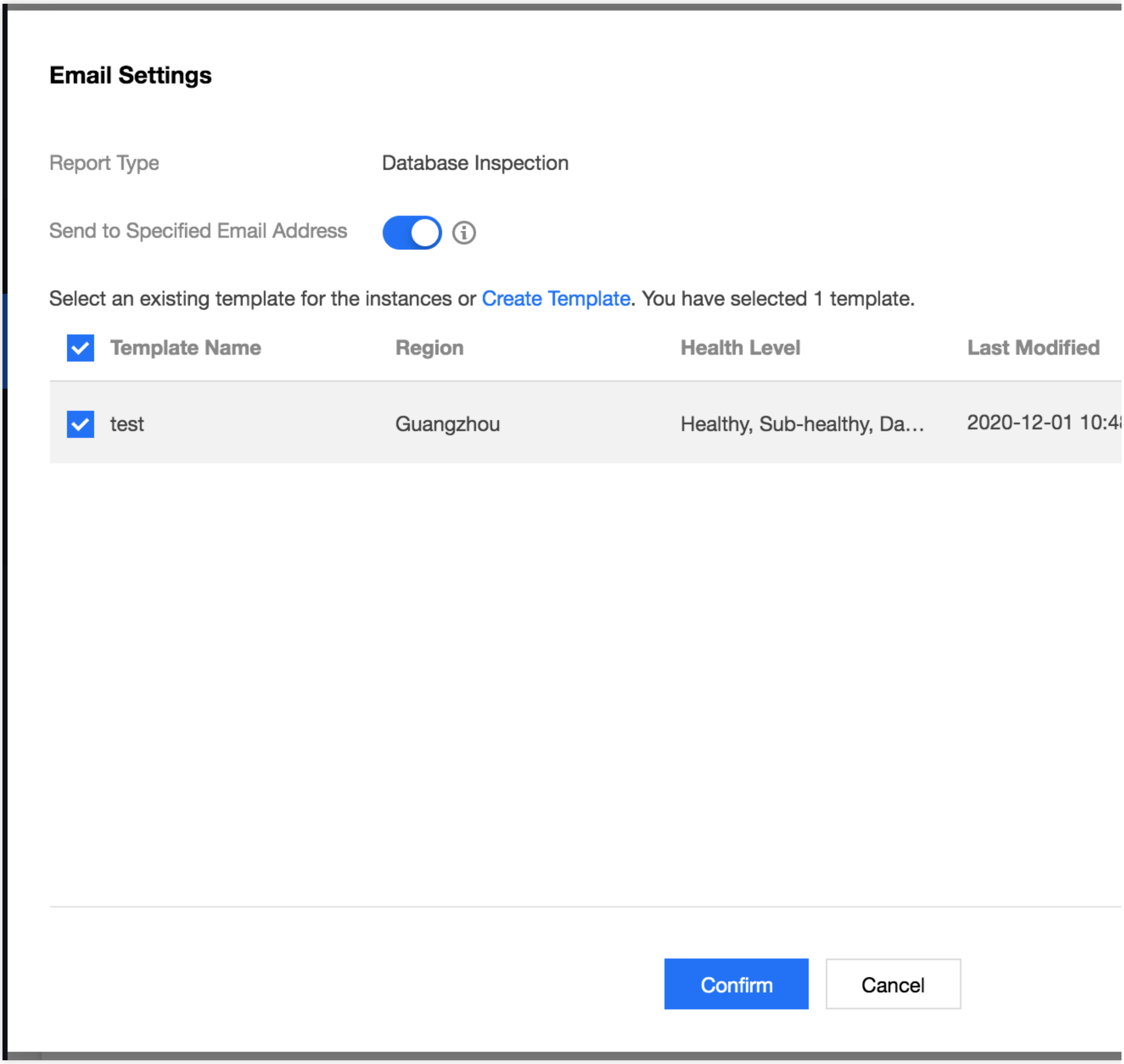
1. Log in to the DBbrain console, select **Monitoring & Alarm** > **Database Inspection** on the left sidebar, and click **Email Settings** on the top-right corner.
2. In the pop-up window, enable **Send to Specified Email Address**, select an existing email template or create a template, and click **Confirm**. 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.



Email Settings

Report Type Database Inspection

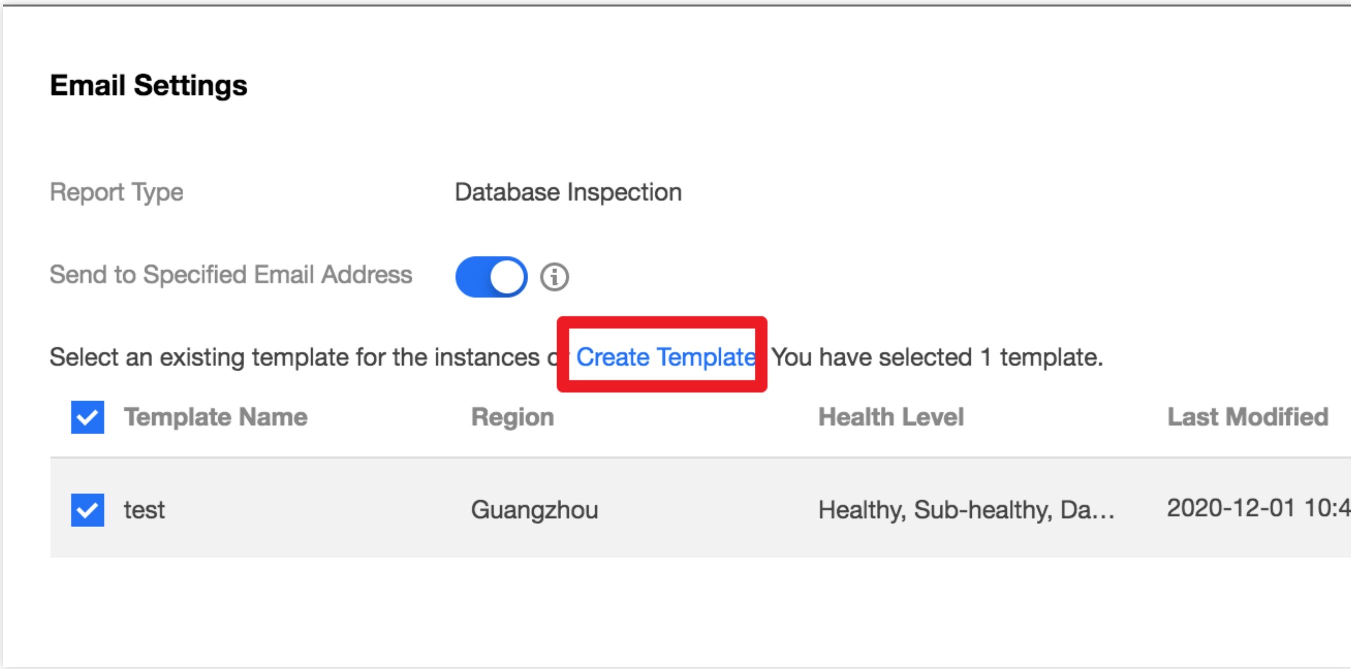
Send to Specified Email Address

Select an existing template for the instances or Create Template. You have selected 1 template.

☑ Template Name	Region	Health Level	Last Modified
☑ test	Guangzhou	Healthy, Sub-healthy, Da…	2020-12-01 10:4

Creating a template

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



Email Settings

Report Type Database Inspection

Send to Specified Email Address

Select an existing template for the instances c Create Template You have selected 1 template.

☑ Template Name	Region	Health Level	Last Modified
☑ test	Guangzhou	Healthy, Sub-healthy, Da...	2020-12-01 10:4

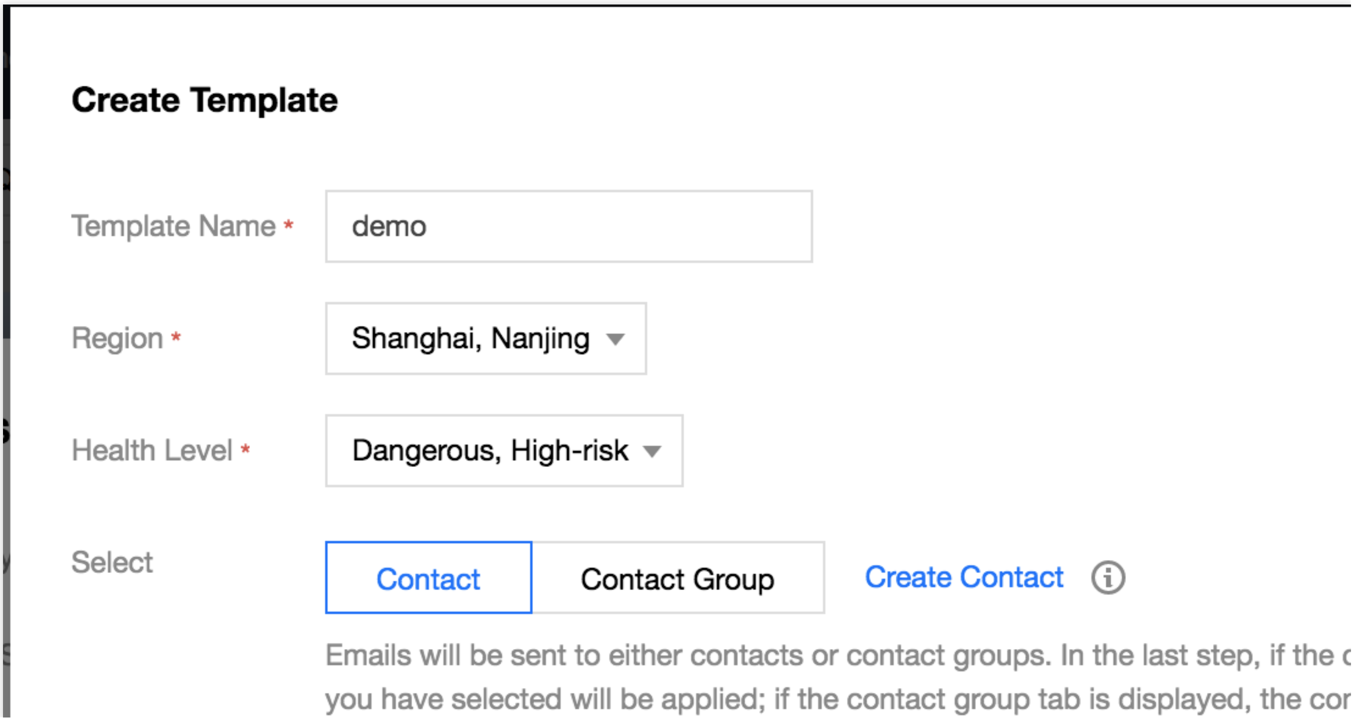
2. In the pop-up window, enter the template name, region, and health level, select contact or contact group, and click **Confirm**.

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.



Create Template

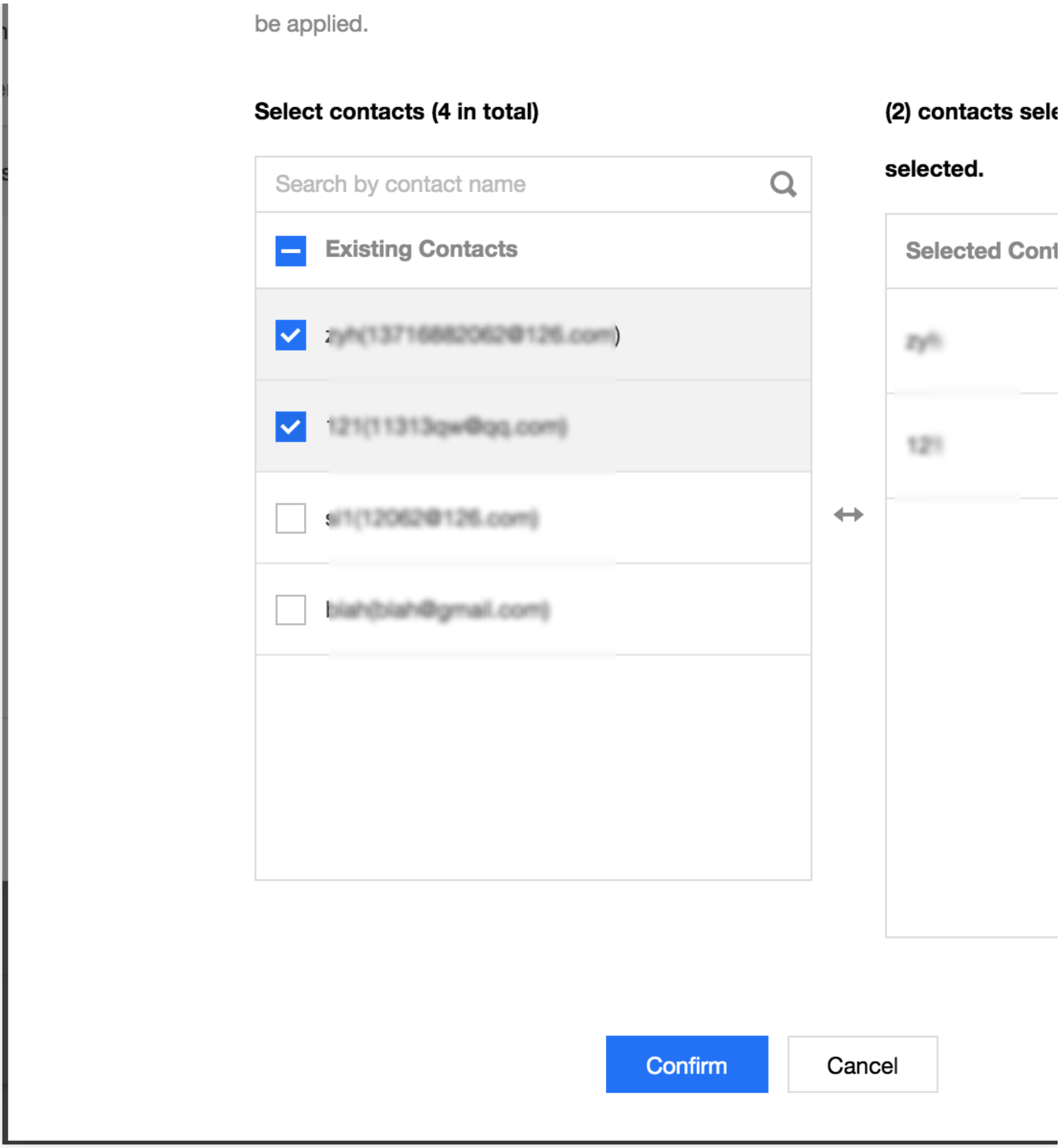
Template Name * demo

Region * Shanghai, Nanjing

Health Level * Dangerous, High-risk

Select Contact Contact Group Create Contact

Emails will be sent to either contacts or contact groups. In the last step, if the c
you have selected will be applied; if the contact group tab is displayed, the cor


be applied.

Select contacts (4 in total)

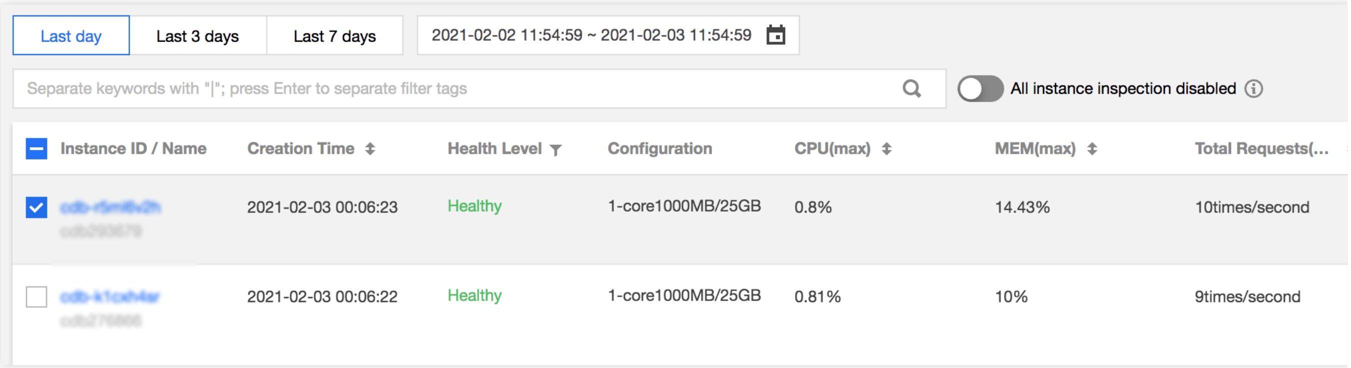
(2) contacts sele

selected.

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.

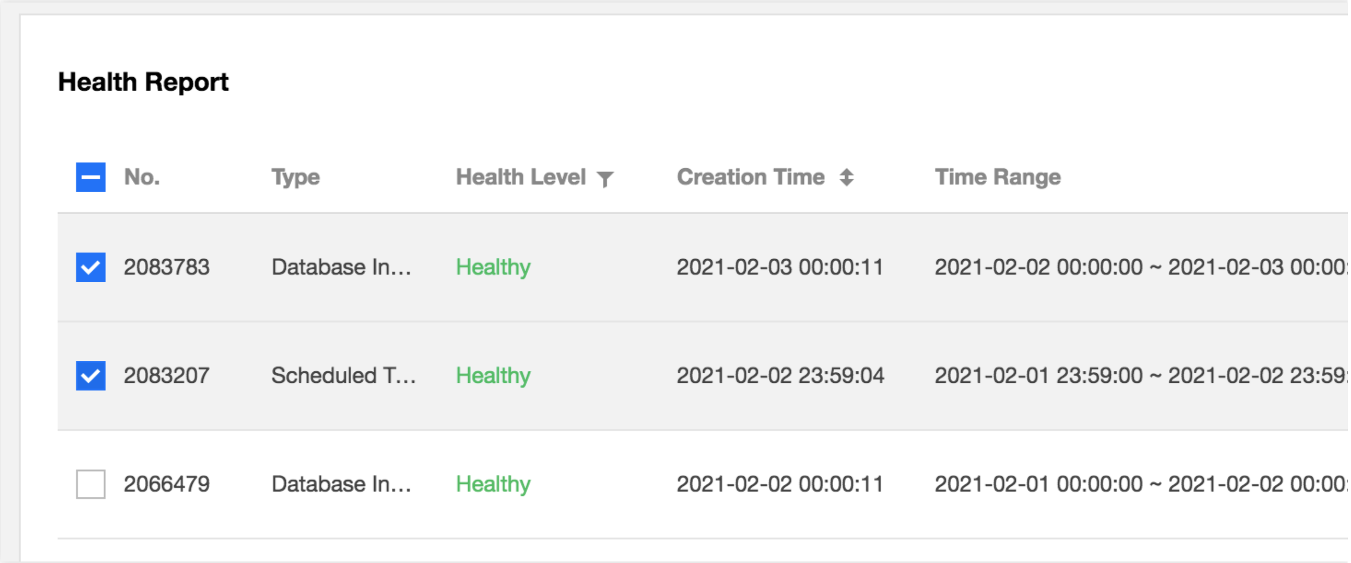


Last day
Last 3 days
Last 7 days
2021-02-02 11:54:59 ~ 2021-02-03 11:54:59
Separate keywords with "|"; press Enter to separate filter tags
All instance inspection disabled

	Instance ID / Name	Creation Time	Health Level	Configuration	CPU(max)	MEM(max)	Total Requests(...
☑	[redacted]	2021-02-03 00:06:23	Healthy	1-core1000MB/25GB	0.8%	14.43%	10times/second
□	[redacted]	2021-02-03 00:06:22	Healthy	1-core1000MB/25GB	0.81%	10%	9times/second

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.

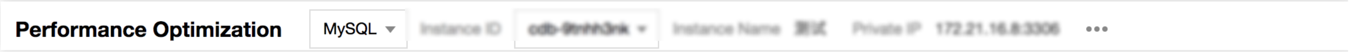


Health Report

	No.	Type	Health Level	Creation Time	Time Range
☑	2083783	Database In...	Healthy	2021-02-03 00:00:11	2021-02-02 00:00:00 ~ 2021-02-03 00:00
☑	2083207	Scheduled T...	Healthy	2021-02-02 23:59:04	2021-02-01 23:59:00 ~ 2021-02-02 23:59
□	2066479	Database In...	Healthy	2021-02-02 00:00:11	2021-02-01 00:00:00 ~ 2021-02-02 00:00

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.

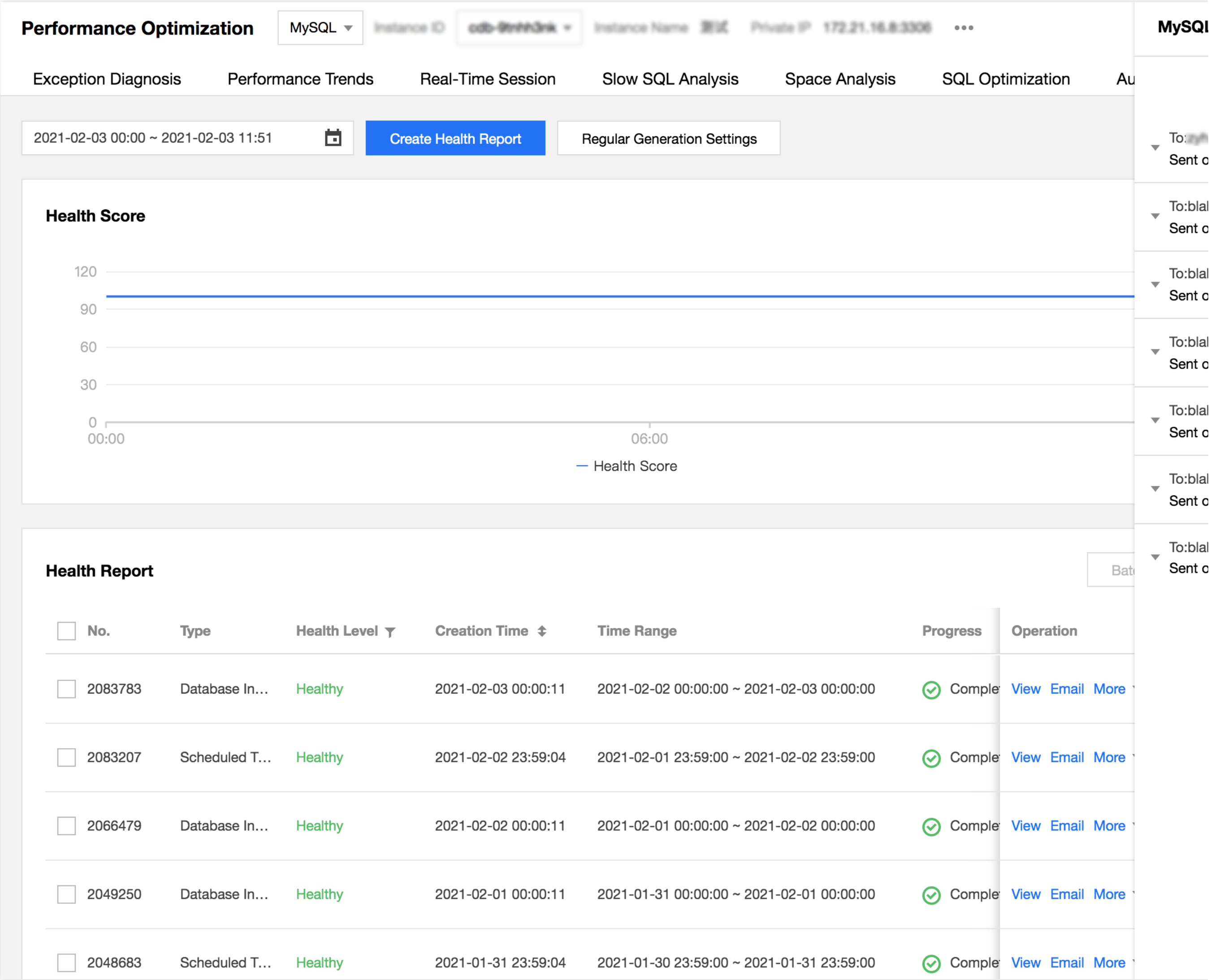


2. In the email history, you can view the recipient, instance basic information, and email sending status.

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.



□	No.	Type	Health Level	Creation Time	Time Range	Progress	Operation
□	2083783	Database In...	Healthy	2021-02-03 00:00:11	2021-02-02 00:00:00 ~ 2021-02-03 00:00:00	✓ Comple	View Email More
□	2083207	Scheduled T...	Healthy	2021-02-02 23:59:04	2021-02-01 23:59:00 ~ 2021-02-02 23:59:00	✓ Comple	View Email More
□	2066479	Database In...	Healthy	2021-02-02 00:00:11	2021-02-01 00:00:00 ~ 2021-02-02 00:00:00	✓ Comple	View Email More
□	2049250	Database In...	Healthy	2021-02-01 00:00:11	2021-01-31 00:00:00 ~ 2021-02-01 00:00:00	✓ Comple	View Email More
□	2048683	Scheduled T...	Healthy	2021-01-31 23:59:04	2021-01-30 23:59:00 ~ 2021-01-31 23:59:00	✓ Comple	View Email More

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:

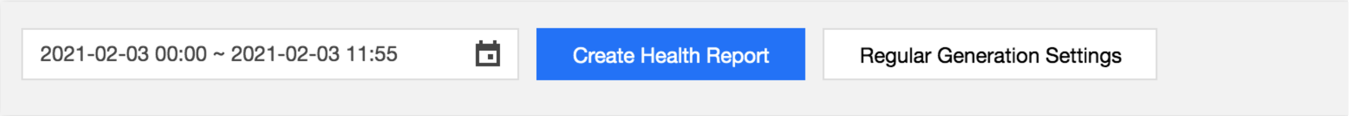
The health report is valid for 3 days. Please download it before it expires.

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**.



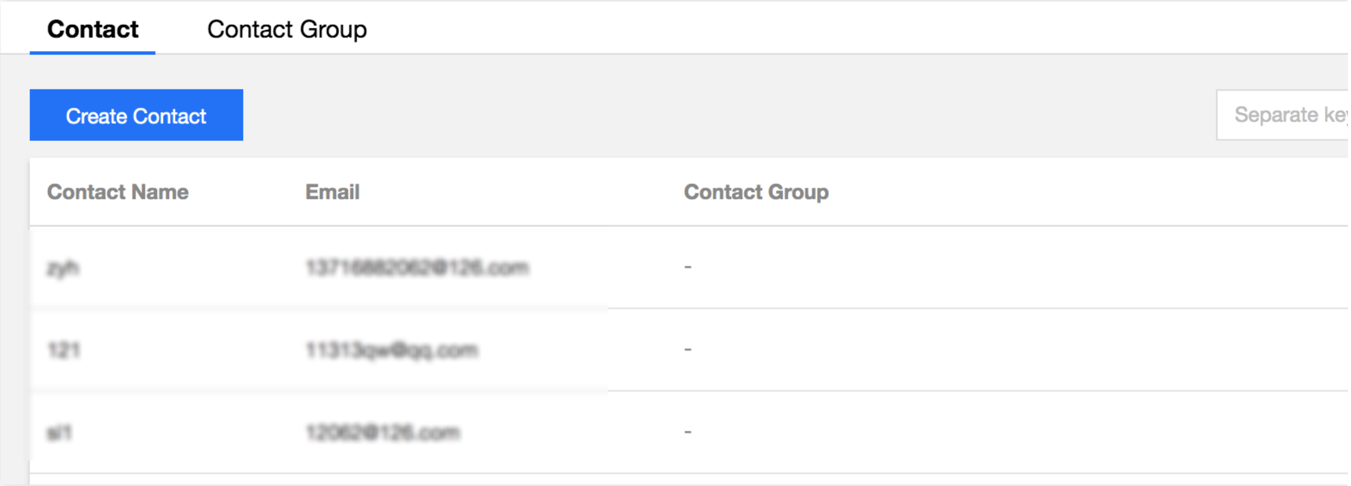
Contact

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 Name	Email	Contact Group
[redacted]	[redacted]	-
[redacted]	[redacted]	-
[redacted]	[redacted]	-

Contact group

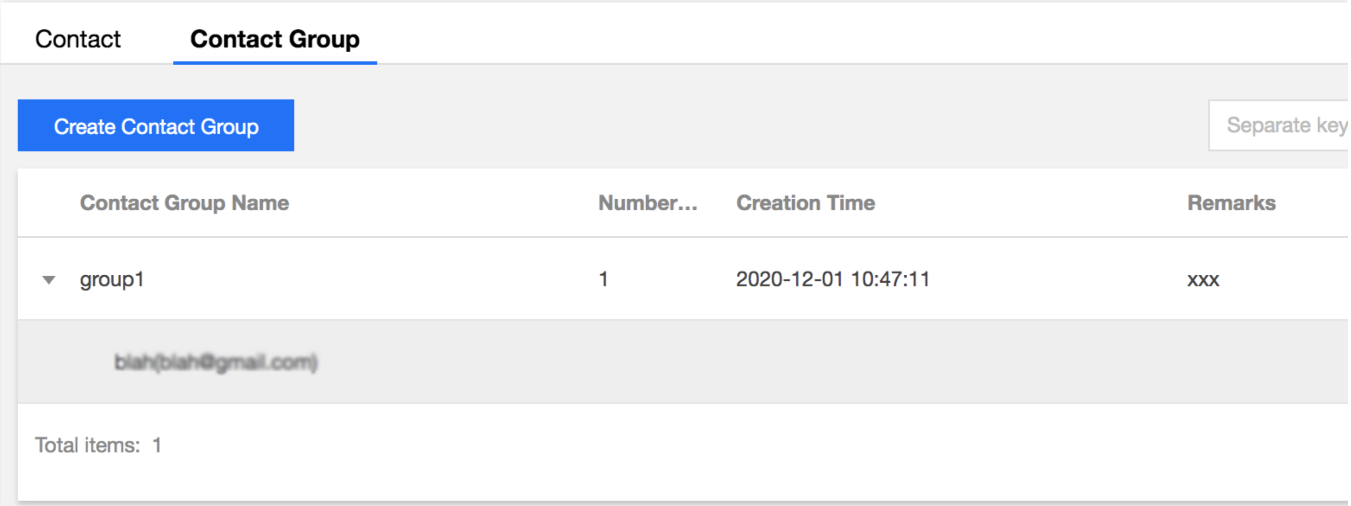
The Contact Group tab is used to manage and set the email recipient groups. Click **Create Contact Group**, enter the contact group name and remarks (optional), and click **Confirm**.

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.



	Contact Group Name	Number...	Creation Time	Remarks
▾	group1	1	2020-12-01 10:47:11	xxx
	blah(blah@gmail.com)			

Total items: 1

MySQL/TDSQL-C for MySQL Performance Optimization

Exception Diagnosis

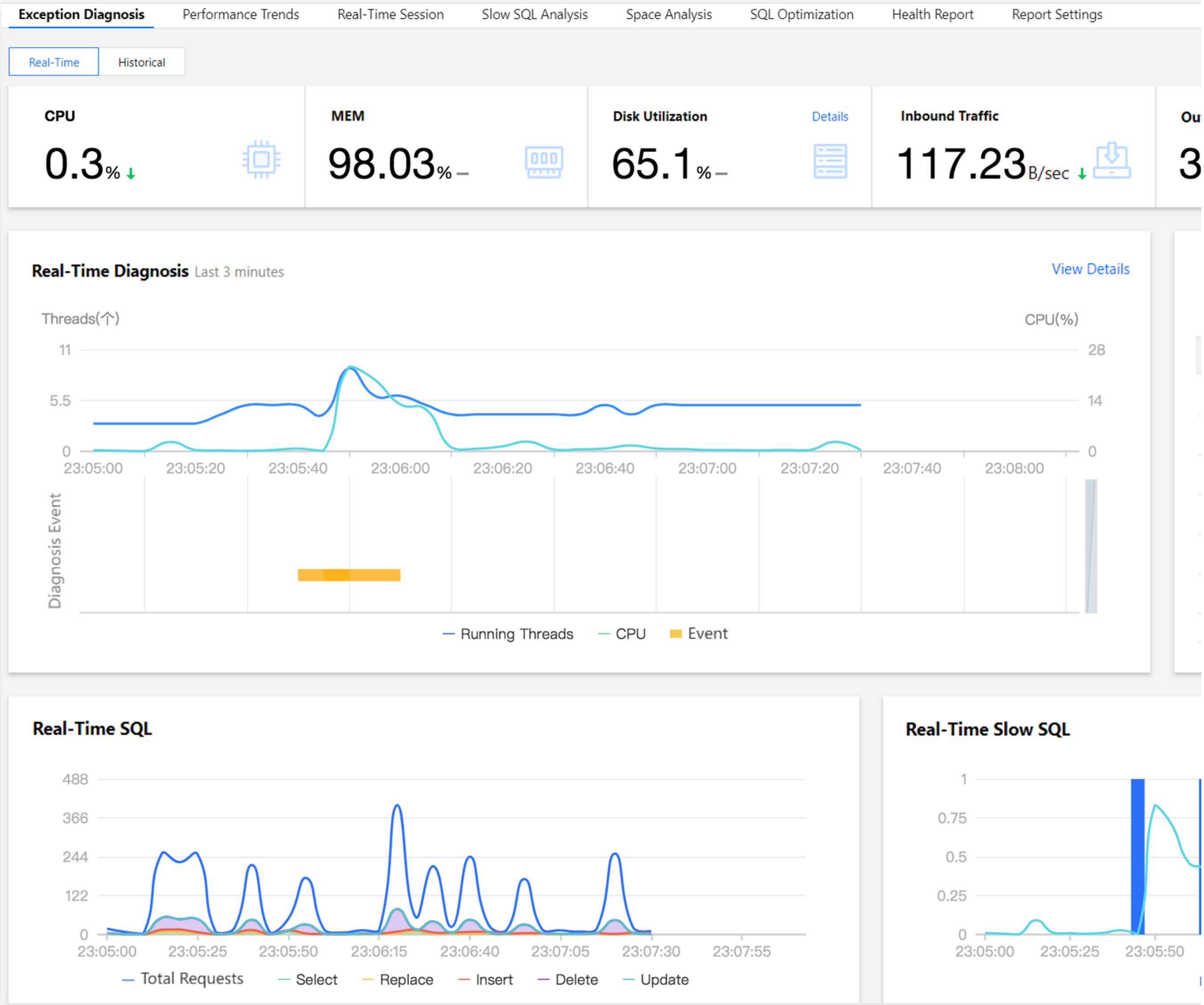
Last updated：2022-08-16 15:55:36

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.



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

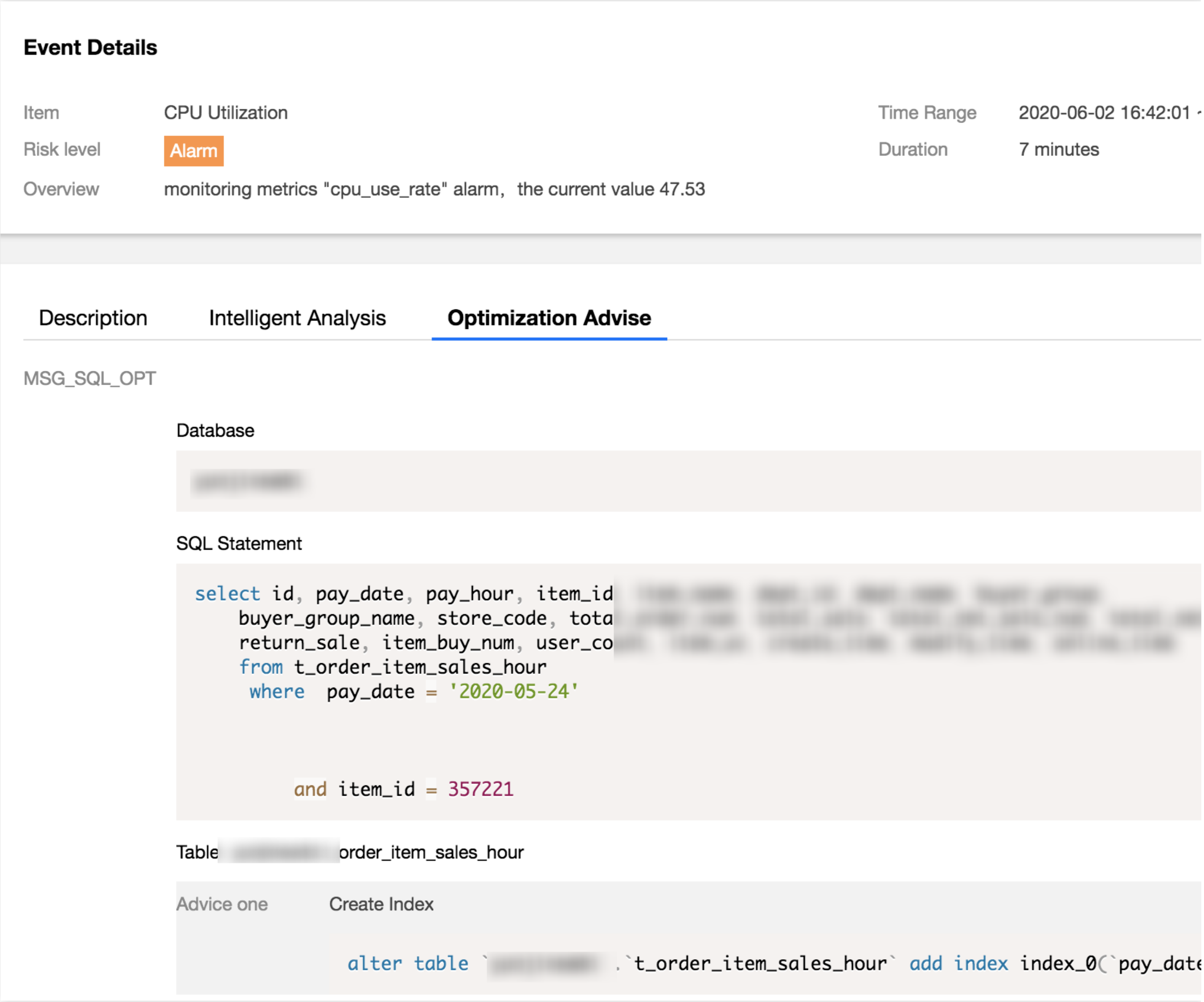
The **Real-Time Diagnosis** or **Diagnosis Records** section displays the current instance's real-time or historical information about the number of running threads, CPU utilization, and diagnosis events.

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.



Event Details

Item	CPU Utilization	Time Range	2020-06-02 16:42:01 -
Risk level	Alarm	Duration	7 minutes
Overview	monitoring metrics "cpu_use_rate" alarm， the current value 47.53		

MSG_SQL_OPT

Database

[redacted]

SQL Statement

```
select id, pay_date, pay_hour, item_id [redacted]
    buyer_group_name, store_code, tota
    return_sale, item_buy_num, user_co
    from t_order_item_sales_hour
     where  pay_date = '2020-05-24'


         and item_id = 357221
```

Table [redacted] order_item_sales_hour

Advice one Create Index

```
alter table `[redacted] .`t_order_item_sales_hour` add index index_0(`pay_date
```

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.

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

		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	✓	✓

	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

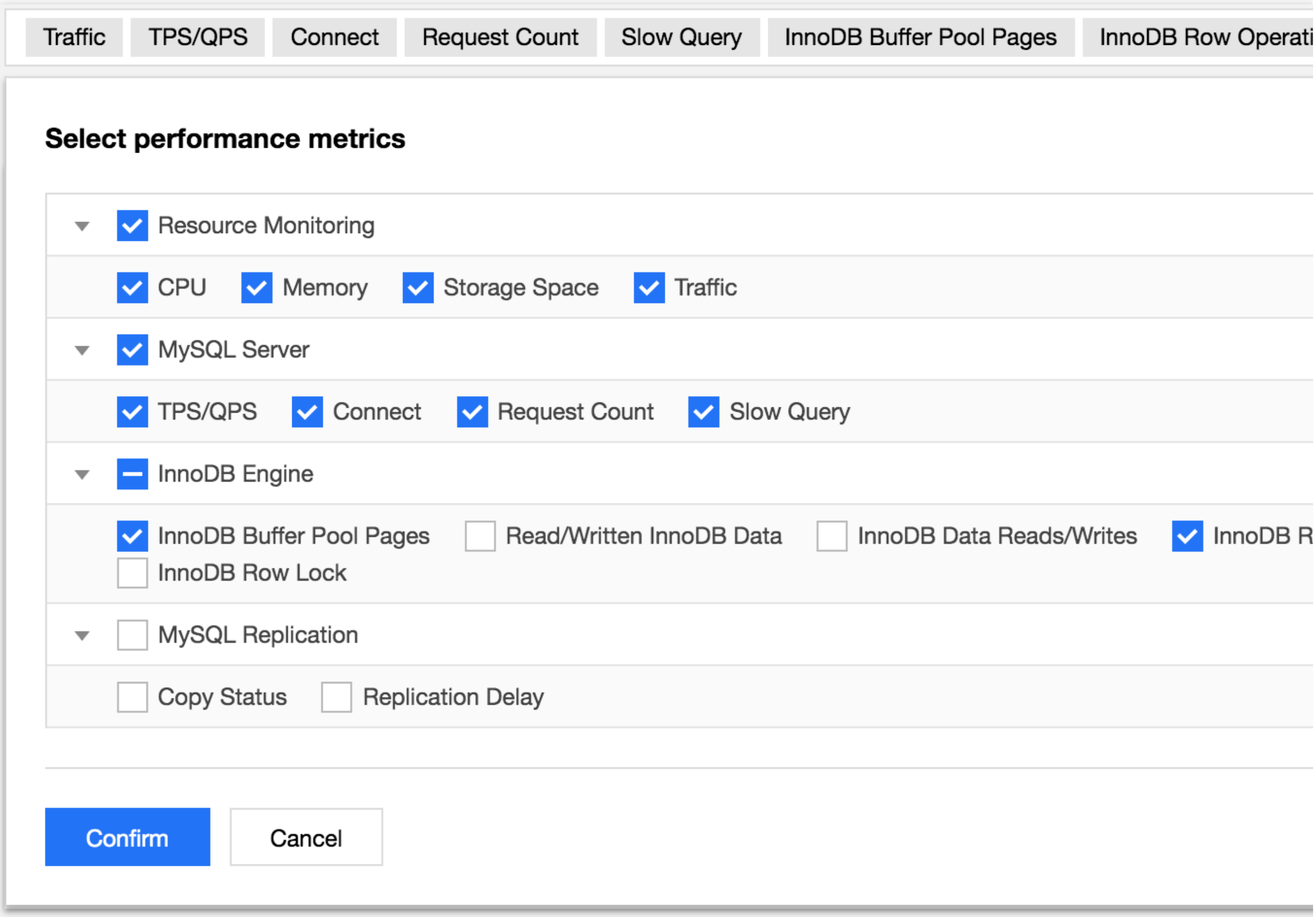
	Requests	Select
		Update
		Delete
		Insert
		Replace
		Total Requests
	Slow Query	Slow Queries
		Full-Table Scans
InnoDB Engine	InnoDB Row Operations	InnoDB Rows Deleted
		InnoDB Rows Inserted
		InnoDB Rows Updated
		InnoDB Rows Read
	InnoDB Buffer Pool Pages	InnoDB Logical Reads
		InnoDB Logical Writes
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.



Select performance metrics

▾ ☑ Resource Monitoring

☑ CPU ☑ Memory ☑ Storage Space ☑ Traffic

▾ ☑ MySQL Server

☑ TPS/QPS ☑ Connect ☑ Request Count ☑ Slow Query

▾ ⊟ InnoDB Engine

☑ InnoDB Buffer Pool Pages □ Read/Written InnoDB Data □ InnoDB Data Reads/Writes ☑ InnoDB R
□ InnoDB Row Lock

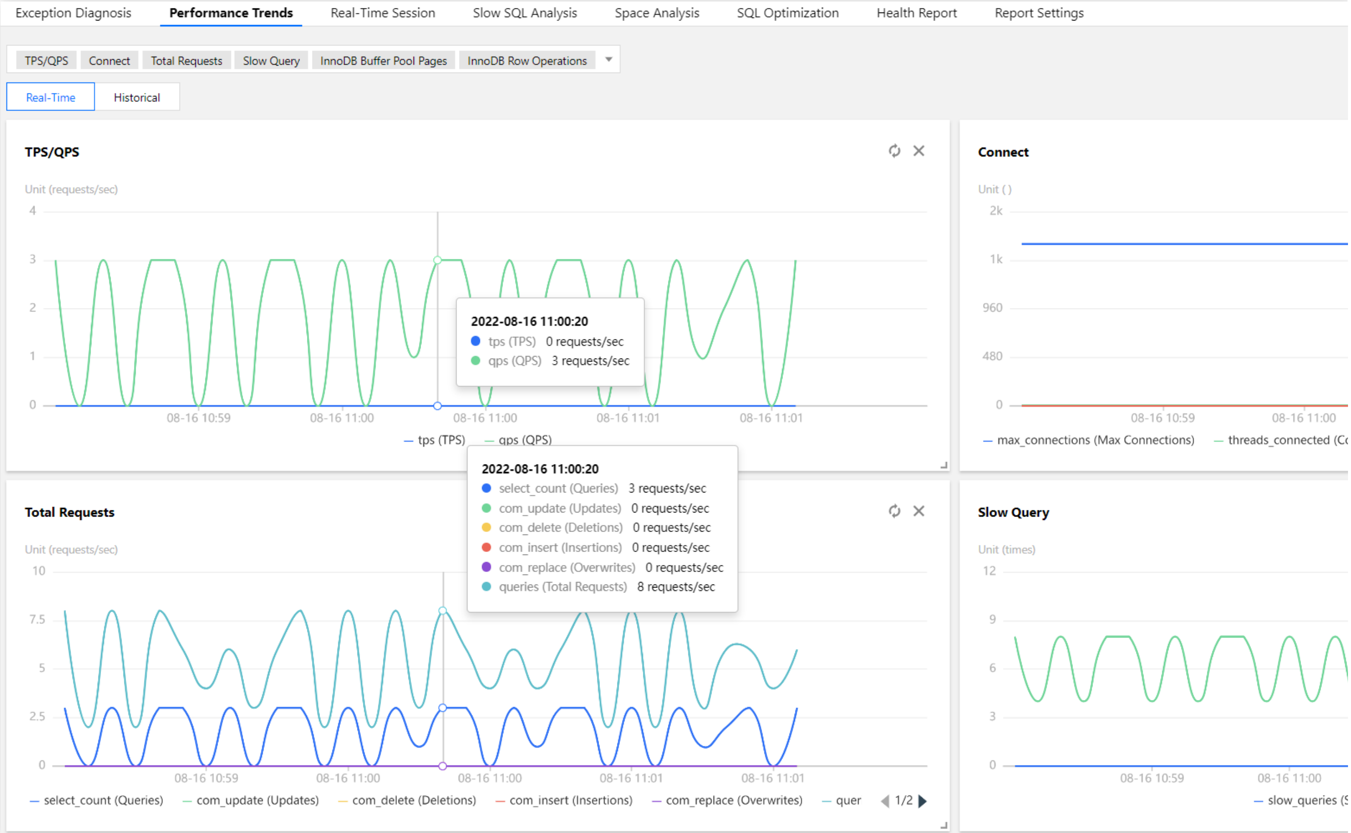
▾ □ MySQL Replication

□ Copy Status □ Replication Delay

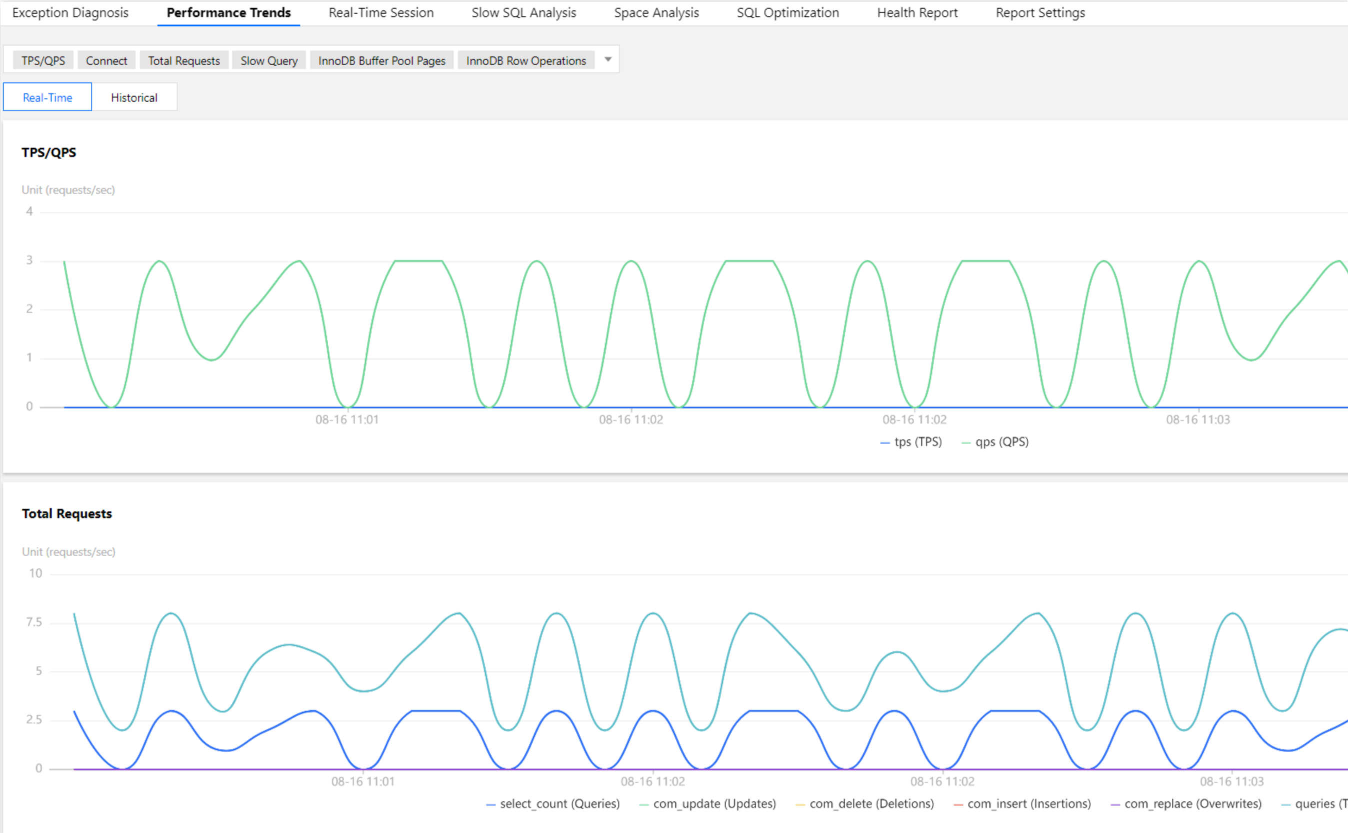
3. View metrics.

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**.

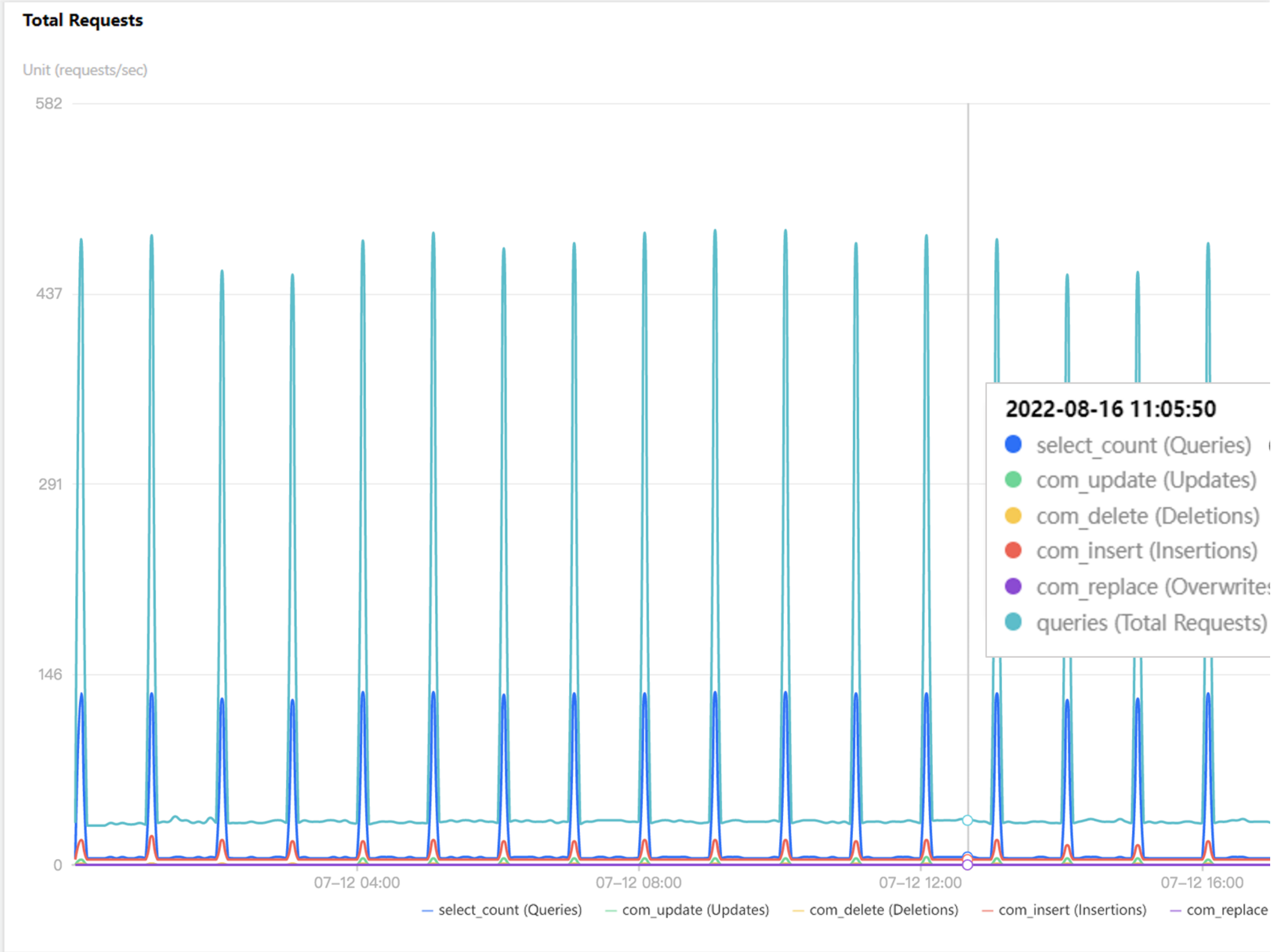


Switching between the one-column and two-column modes: Click the button on the right of **Chart Interaction** in the top-right corner to switch.



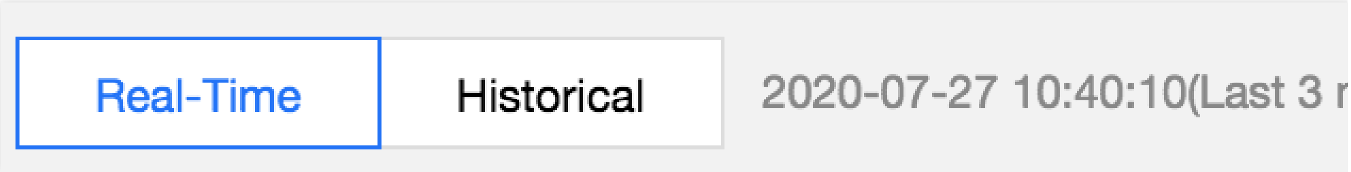
Dragging a monitoring view: Click the border of a monitoring view to drag it to the desired position.

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.

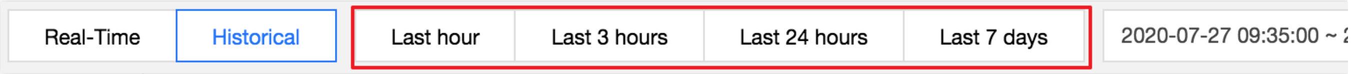


Switching between the real-time and historical modes: Click **Real-Time** or **Historical** to view the real-time or historical performance trends.

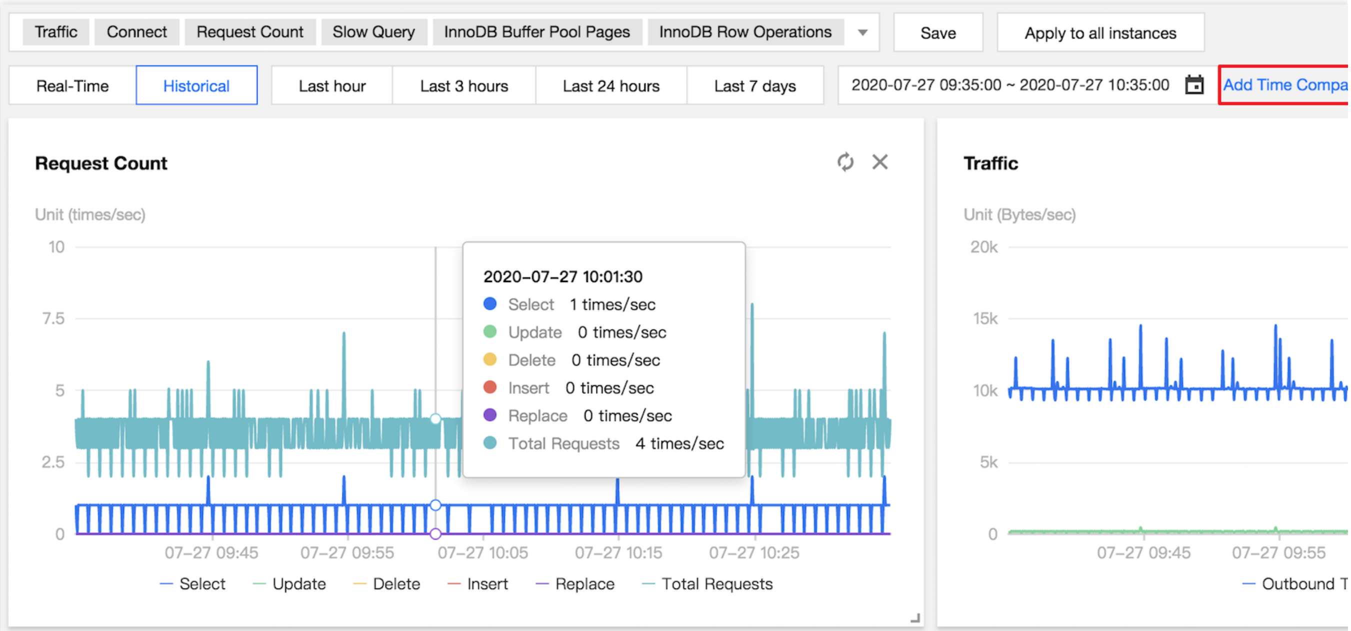
The real-time performance trends view displays the performance trends of the instance and is automatically refreshed by default. You can click **Disable refresh** to stop refreshing the trends in real time.



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.



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

Real-Time Session

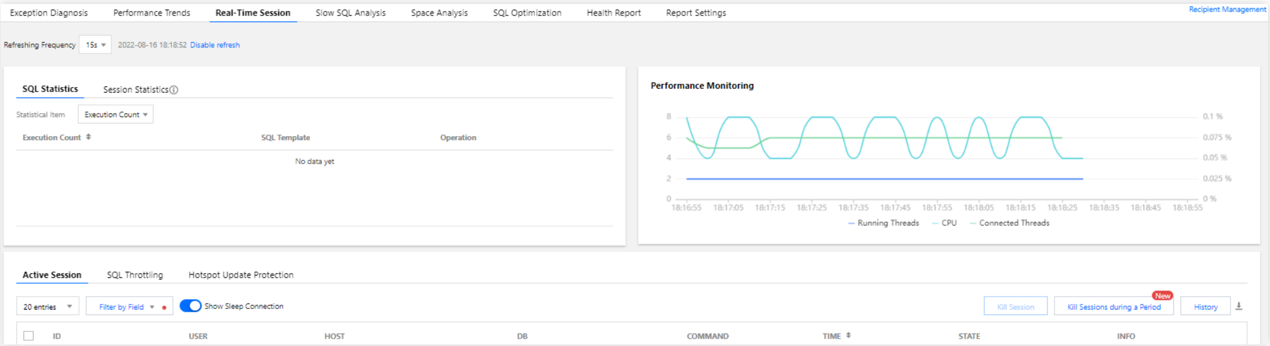
Last updated：2022-08-16 18:34:06

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.



Refreshing Frequency 15s 2022-08-16 18:18:52

Statistical Item Execution Count

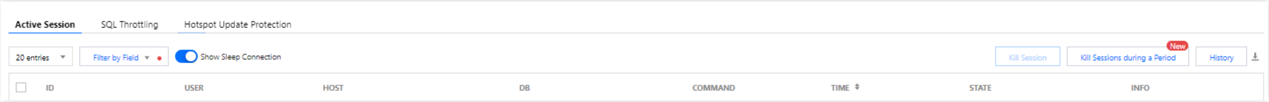
Execution Count	SQL Template	Operation
	No data yet	

20 entries Filter by Field ● Show Sleep Connection

□	ID	USER	HOST	DB	COMMAND	TIME	STATE	INFO

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**.



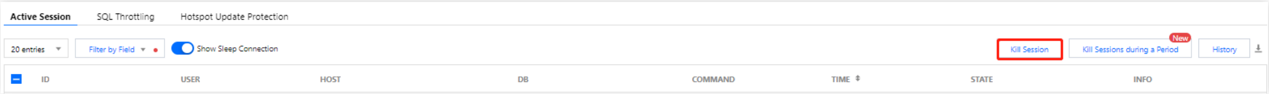
Killing sessions

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.

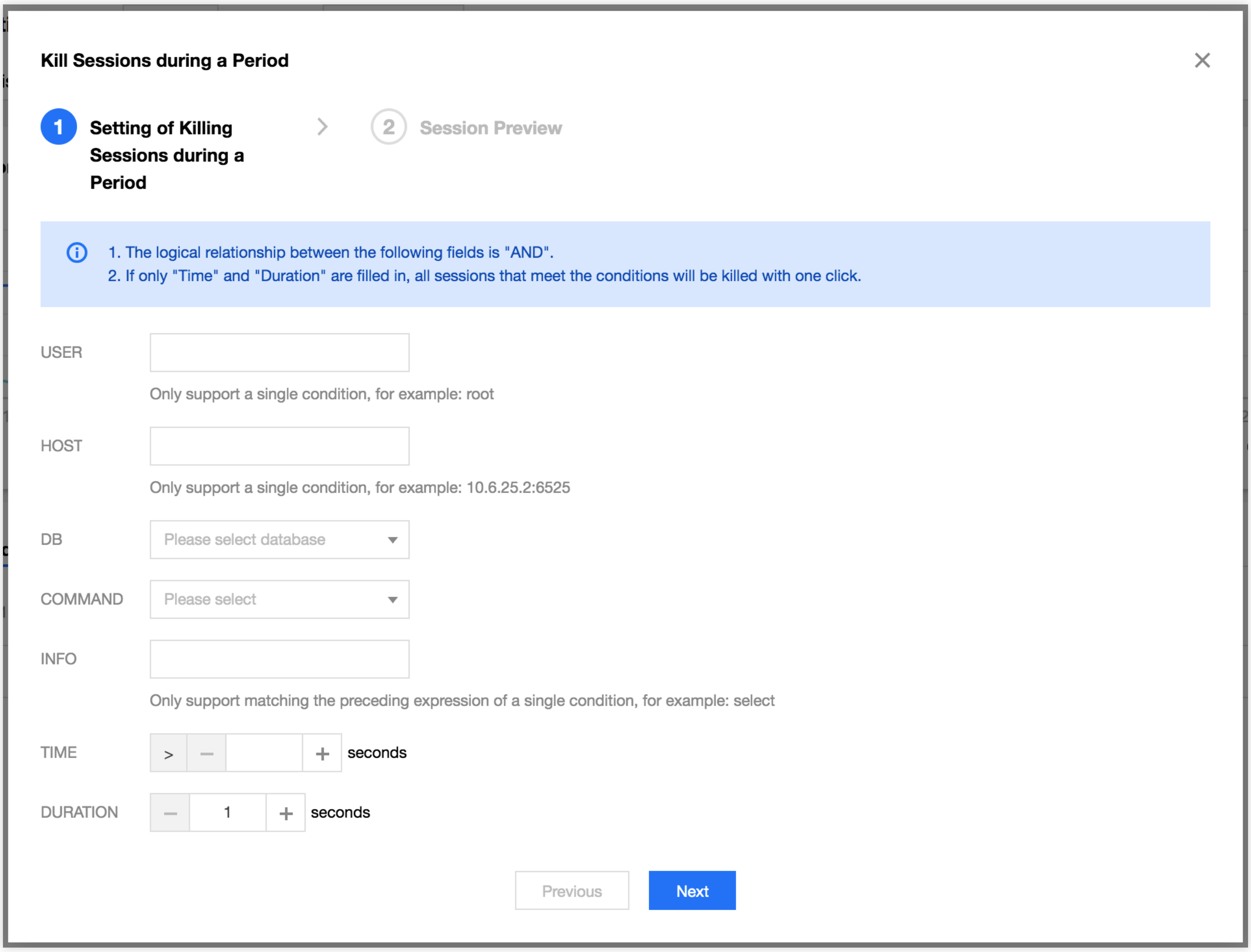
1. Task Settings.

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.



Kill Sessions during a Period

1 Setting of Killing Sessions during a Period > 2 Session Preview

1. The logical relationship between the following fields is "AND".
2. If only "Time" and "Duration" are filled in, all sessions that meet the conditions will be killed with one click.

USER

Only support a single condition, for example: root

HOST

Only support a single condition, for example: 10.6.25.2:6525

DB: Please select database

COMMAND: Please select

INFO

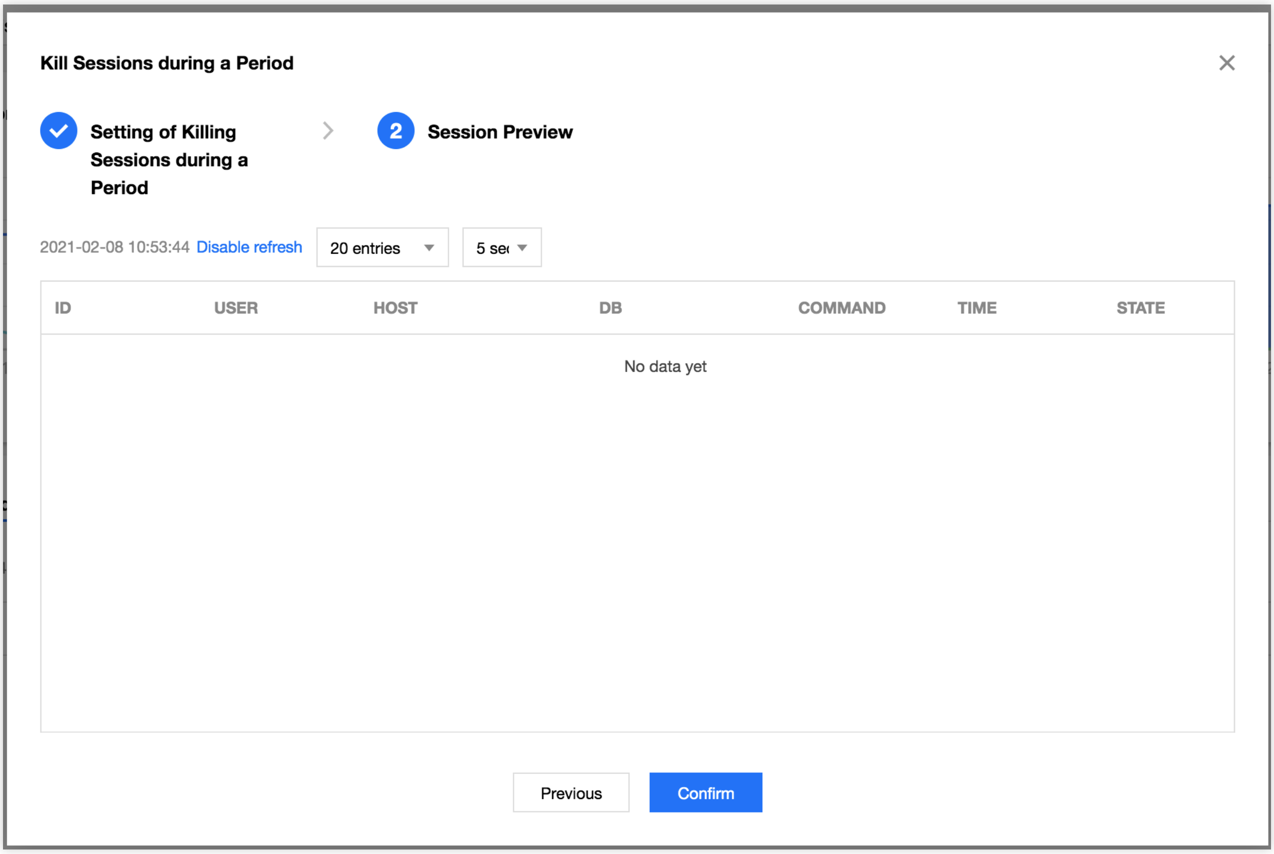
Only support matching the preceding expression of a single condition, for example: select

TIME: > seconds

DURATION: 1 seconds

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.



Kill Sessions during a Period

✓ Setting of Killing Sessions during a Period > 2 Session Preview

2021-02-08 10:53:44 Disable refresh | 20 entries | 5 sec

ID	USER	HOST	DB	COMMAND	TIME	STATE
No data yet						

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.

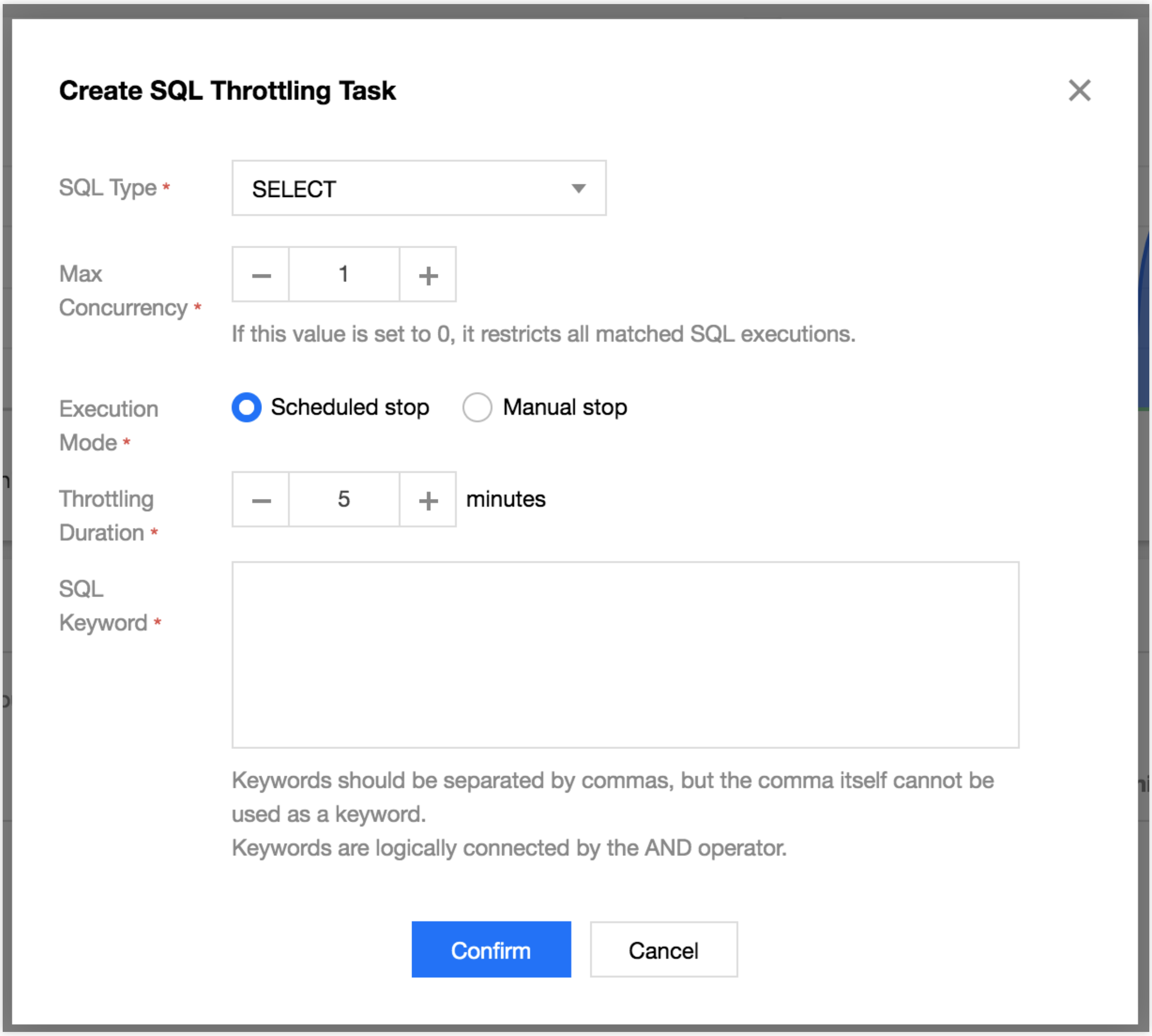
SQL Type: Select **SELECT**, **UPDATE**, **DELETE**, **INSERT**, or **REPLACE**.

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.



Create SQL Throttling Task

SQL Type *: SELECT

Max Concurrency *: 1

If this value is set to 0, it restricts all matched SQL executions.

Execution Mode *: ● Scheduled stop ○ Manual stop

Throttling Duration *: 5 minutes

SQL Keyword *:

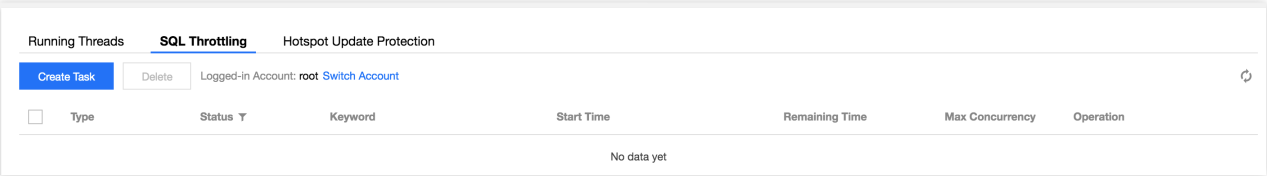
Keywords should be separated by commas, but the comma itself cannot be used as a keyword.
Keywords are logically connected by the AND operator.

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.


Running Threads
SQL Throttling
Hotspot Update Protection
Create Task
Delete
Logged-in Account: root Switch Account

□	Type	Status	Keyword	Start Time	Remaining Time	Max Concurrency	Operation
			No data yet				

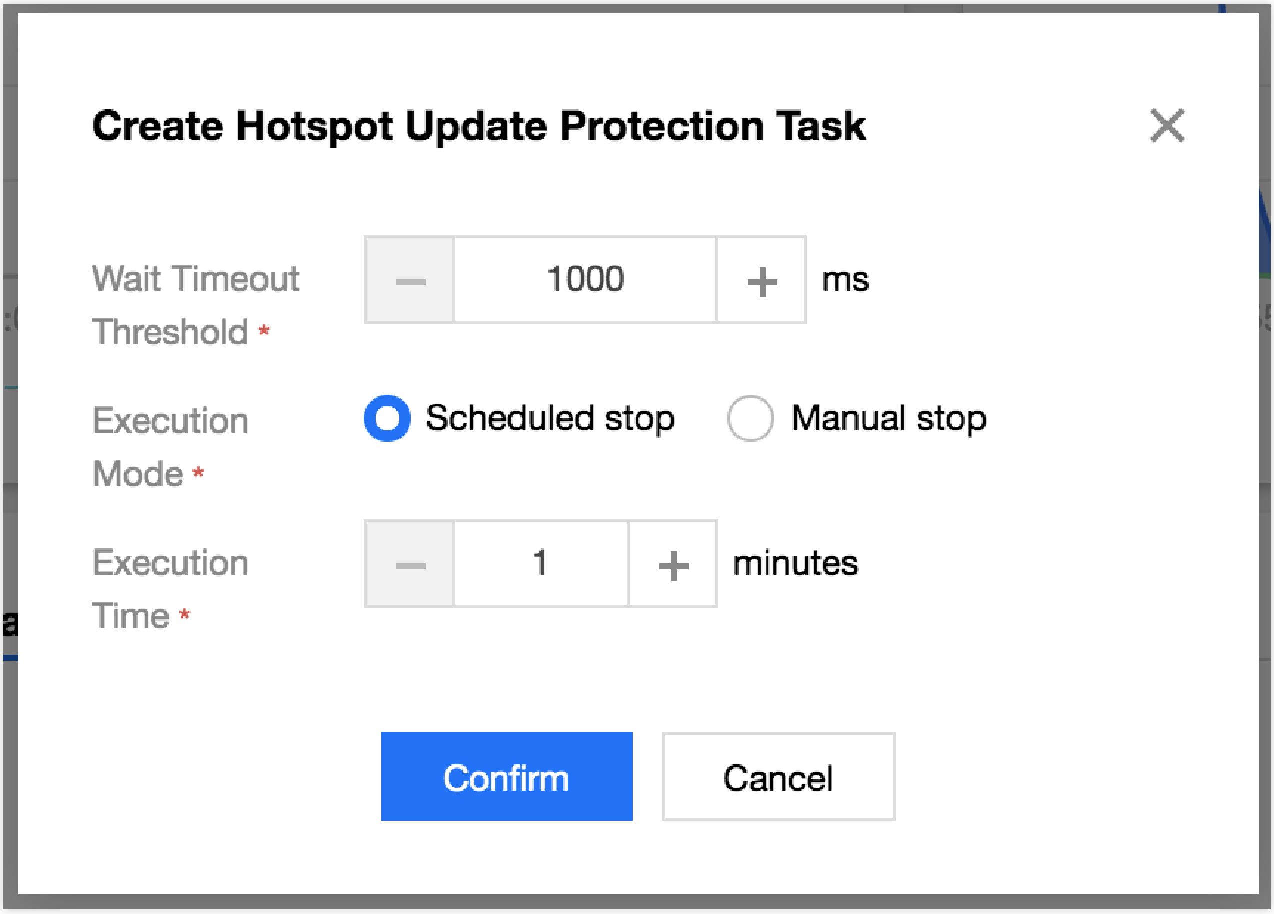
Hotspot update protection

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**.



Create Hotspot Update Protection Task

Wait Timeout Threshold *: 1000 ms

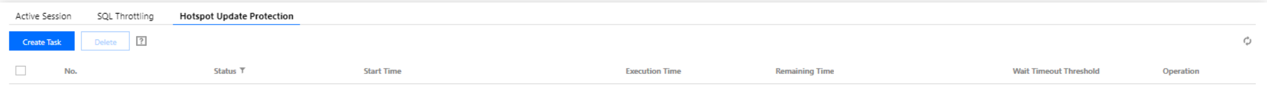
Execution Mode *: ● Scheduled stop ○ Manual stop

Execution Time *: 1 minutes

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.



□	No.	Status	Start Time	Execution Time	Remaining Time	Wait Timeout Threshold	Operation

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

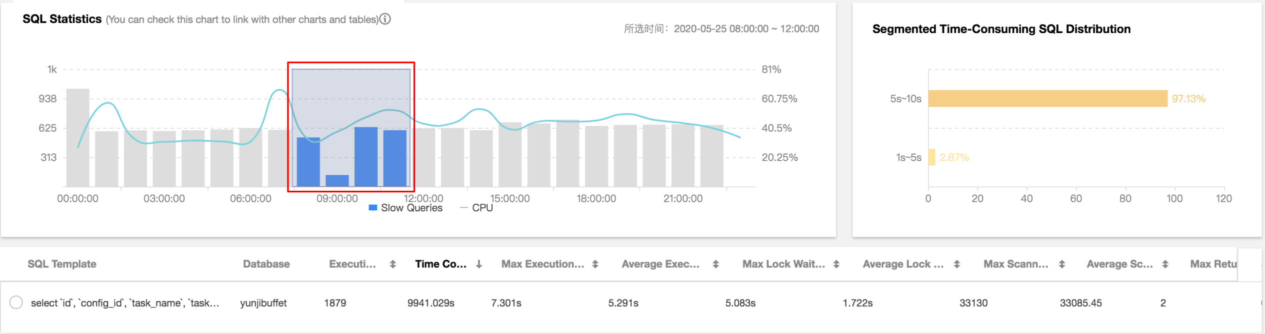
1. Log in to the DBbrain console and select **Performance Optimization** on the left sidebar. On the displayed page, select a database type and an instance at the top, and select the **Slow SQL Analysis** tab.

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**.

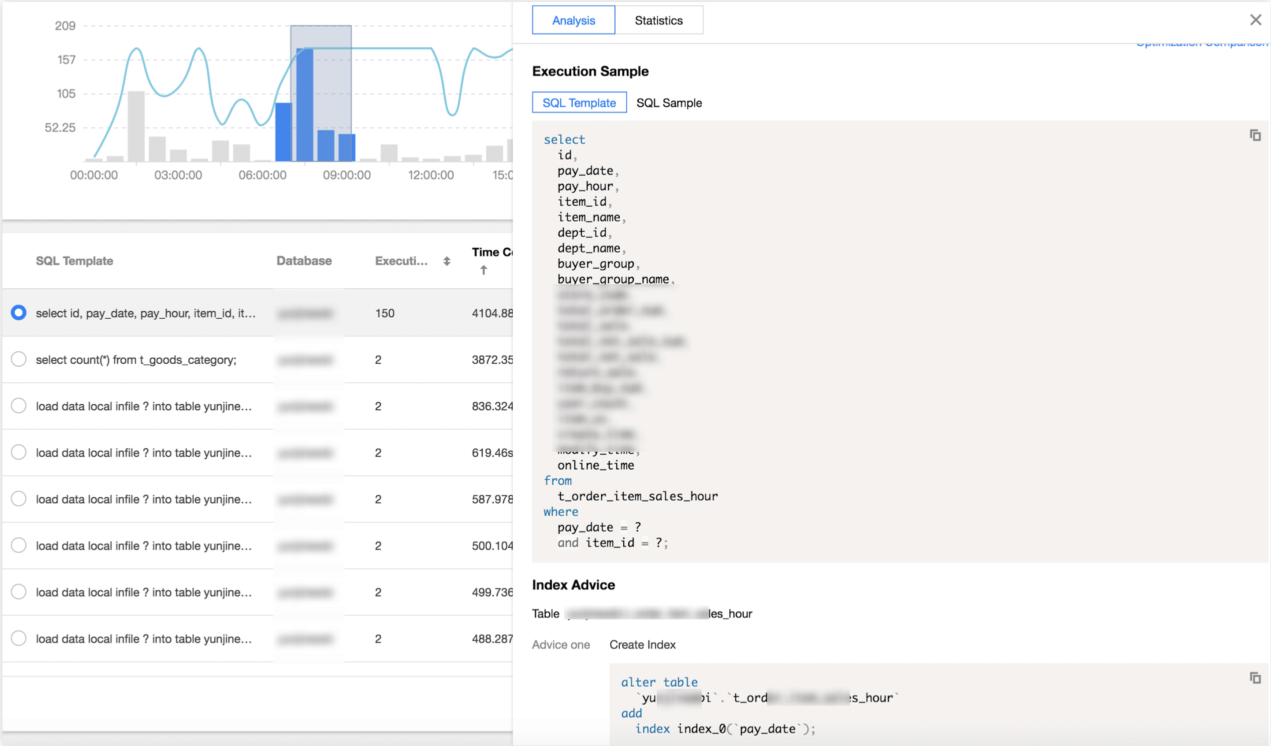

SQL Statistics (You can check this chart to link with other charts and tables)ⓘ
所选时间：2020-05-25 08:00:00 ~ 12:00:00
1k
938
625
313
81%
60.75%
40.5%
20.25%
00:00:00
03:00:00
06:00:00
09:00:00
12:00:00
15:00:00
18:00:00
21:00:00
Slow Queries
CPU
Segmented Time-Consuming SQL Distribution
5s~10s
1s~5s
97.13%
2.87%
0
20
40
60
80
100
120

SQL Template	Database	Executi...	Time Co...	Max Execution...	Average Exec...	Max Lock Wait...	Average Lock ...	Max Scann...	Average Sc...	Max Retu
○ select `id`, `config_id`, `task_name`, `task...	yunjibuffet	1879	9941.029s	7.301s	5.291s	5.083s	1.722s	33130	33085.45	2

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.



SQL Template	Database	Executi...	Time C...
● select id, pay_date, pay_hour, item_id, it...	[redacted]	150	4104.88
○ select count(*) from t_goods_category;	[redacted]	2	3872.35
○ load data local infile ? into table yunjine...	[redacted]	2	836.324
○ load data local infile ? into table yunjine...	[redacted]	2	619.46s
○ load data local infile ? into table yunjine...	[redacted]	2	587.978
○ load data local infile ? into table yunjine...	[redacted]	2	500.104
○ load data local infile ? into table yunjine...	[redacted]	2	499.736
○ load data local infile ? into table yunjine...	[redacted]	2	488.287

Execution Sample

```
select
  id,
  pay_date,
  pay_hour,
  item_id,
  item_name,
  dept_id,
  dept_name,
  buyer_group,
  buyer_group_name,
  [redacted]
  online_time
from
  t_order_item_sales_hour
where
  pay_date = ?
  and item_id = ?;
```

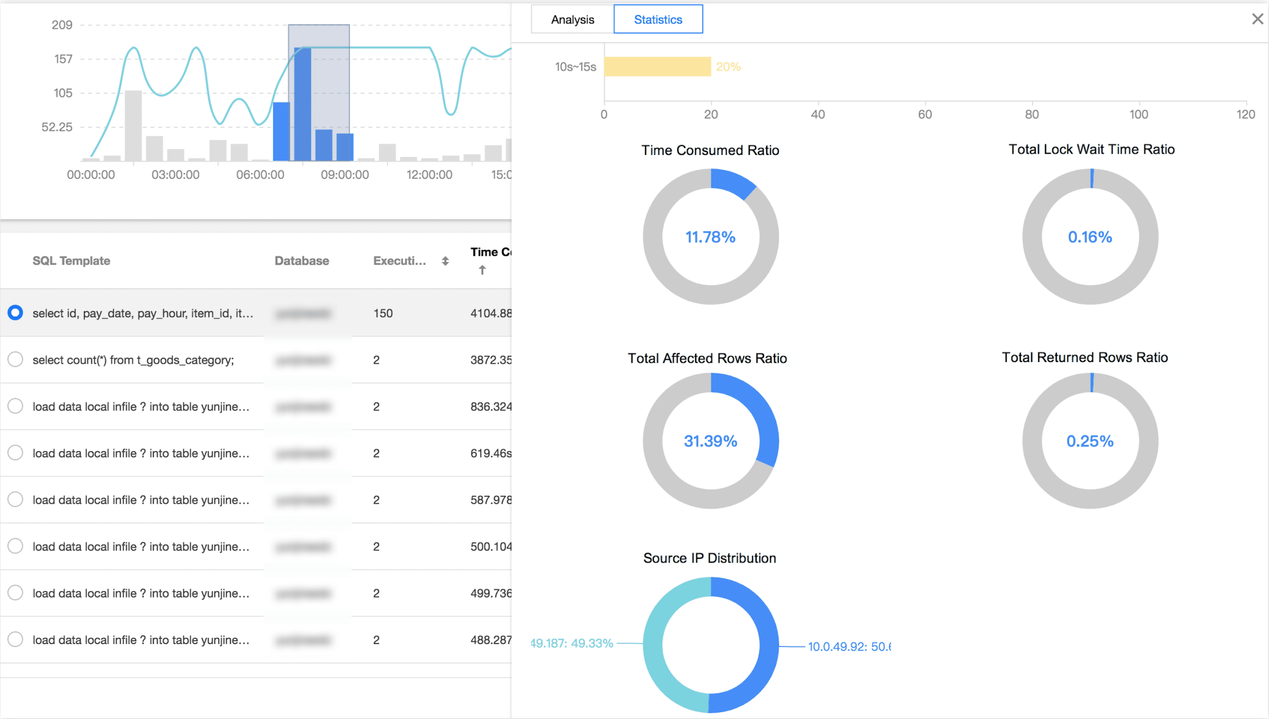
Index Advice

Table [redacted]les_hour

Advice one   Create Index

```
alter table
  `yu[redacted]i`.`t_ord[redacted]s_hour`
add
  index index_0(`pay_date`);
```

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.



	SQL Template	Database	Executi...	Time C
●	select id, pay_date, pay_hour, item_id, it…	[redacted]	150	4104.88
○	select count(*) from t_goods_category;	[redacted]	2	3872.35
○	load data local infile ? into table yunjine…	[redacted]	2	836.324
○	load data local infile ? into table yunjine…	[redacted]	2	619.46s
○	load data local infile ? into table yunjine…	[redacted]	2	587.978
○	load data local infile ? into table yunjine…	[redacted]	2	500.104
○	load data local infile ? into table yunjine…	[redacted]	2	499.736
○	load data local infile ? into table yunjine…	[redacted]	2	488.287

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.

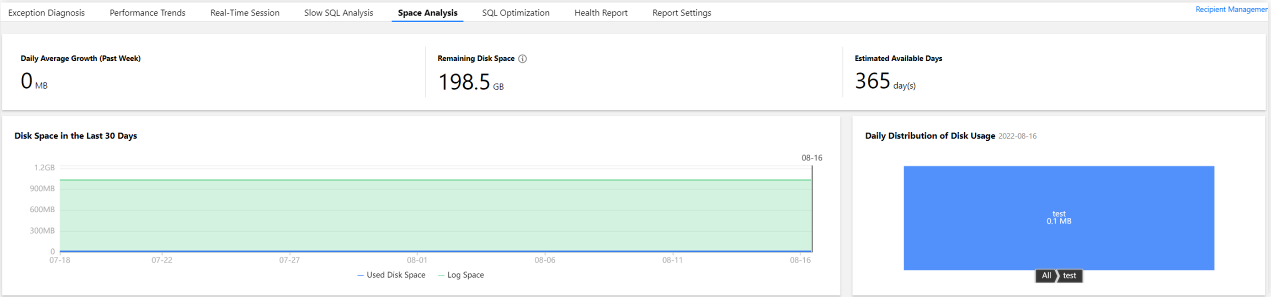
Space Analysis

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.



Daily Average Growth (Past Week)
0 MB

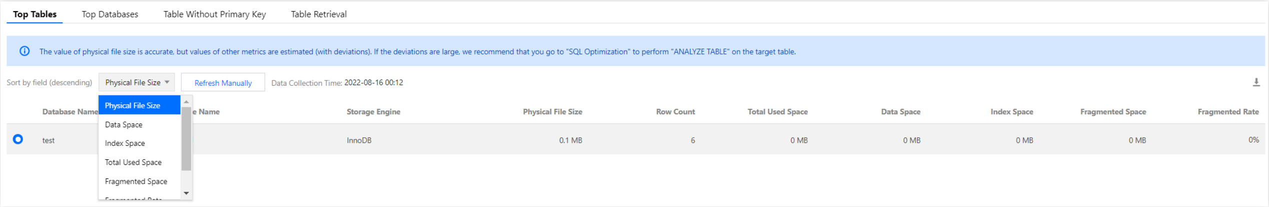
Remaining Disk Space
198.5 GB

Estimated Available Days
365 day(s)

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.
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.



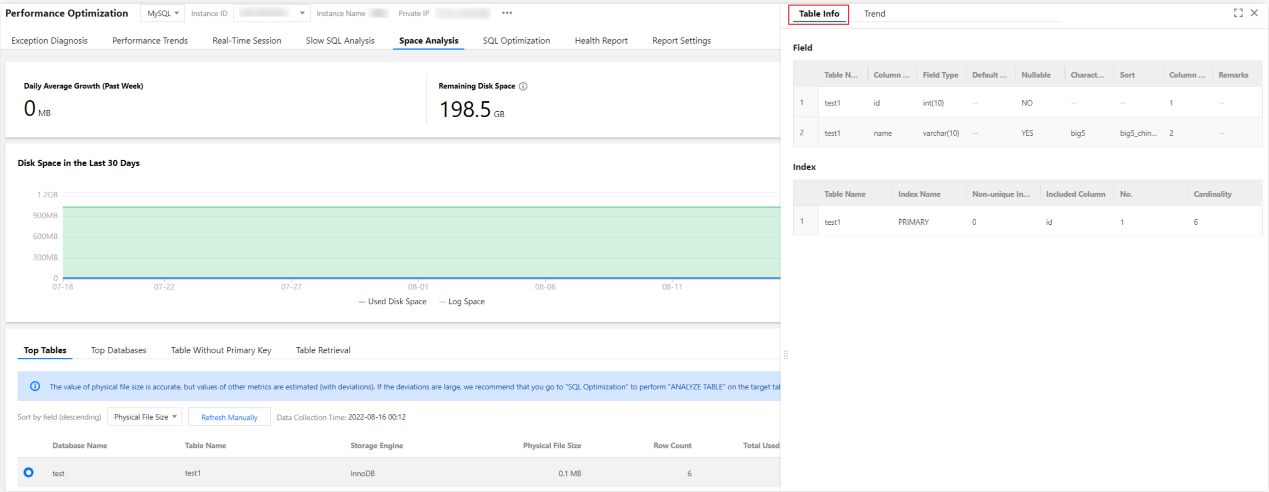
The value of physical file size is accurate, but values of other metrics are estimated (with deviations). If the deviations are large, we recommend that you go to "SQL Optimization" to perform "ANALYZE TABLE" on the target table.

Sort by field (descending): Physical File Size

Data Collection Time: 2022-08-16 00:12

	Database Name	e Name	Storage Engine	Physical File Size	Row Count	Total Used Space	Data Space	Index Space	Fragmented Space	Fragmented Rate
○	test		InnoDB	0.1 MB	6	0 MB	0 MB	0 MB	0 MB	0%

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**.



Daily Average Growth (Past Week): 0 MB

Remaining Disk Space: 198.5 GB

The value of physical file size is accurate, but values of other metrics are estimated (with deviations). If the deviations are large, we recommend that you go to "SQL Optimization" to perform "ANALYZE TABLE" on the target tal

Sort by field (descending): Physical File Size

Data Collection Time: 2022-08-16 00:12

	Database Name	Table Name	Storage Engine	Physical File Size	Row Count	Total Used
○	test	test1	InnoDB	0.1 MB	6	

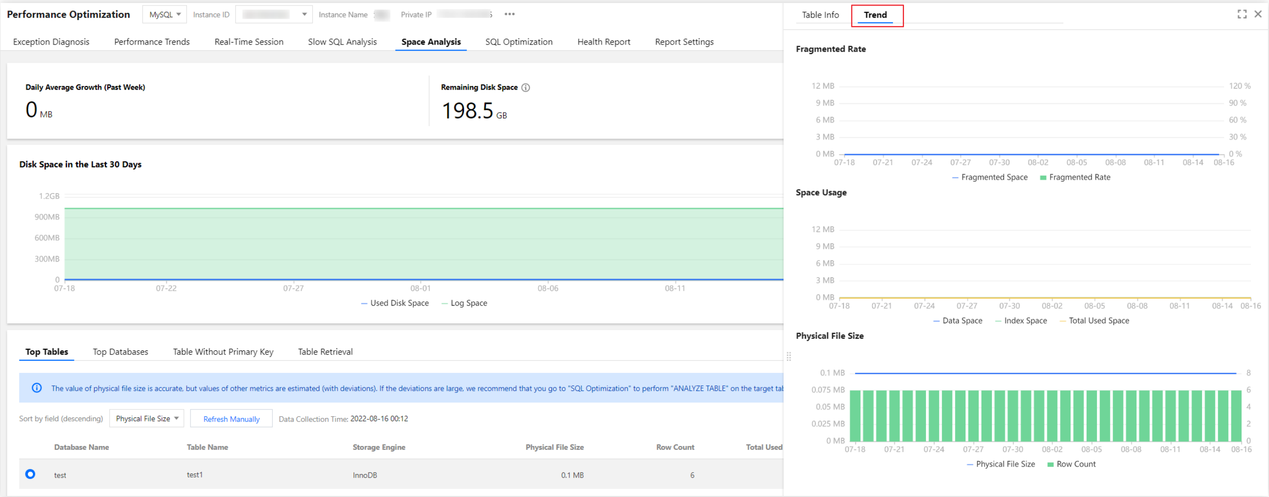
Field

	Table N...	Column ...	Field Type	Default ...	Nullable	Charact...	Sort	Column ...	Remarks
1	test1	id	int(10)	--	NO	--	--	1	--
2	test1	name	varchar(10)	--	YES	big5	big5_chin...	2	--

Index

	Table Name	Index Name	Non-unique In...	Included Column	No.	Cardinality
1	test1	PRIMARY	0	id	1	6

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**.



Performance Optimization MySQL ▾ Instance ID [redacted] ▾ Instance Name [redacted] Private IP [redacted] •••

Exception Diagnosis | Performance Trends | Real-Time Session | Slow SQL Analysis | **Space Analysis** | SQL Optimization | Health Report | Report Settings

Daily Average Growth (Past Week)
0 MB

Remaining Disk Space ⓘ
198.5 GB

Disk Space in the Last 30 Days

Top Tables | Top Databases | Table Without Primary Key | Table Retrieval

ⓘ The value of physical file size is accurate, but values of other metrics are estimated (with deviations). If the deviations are large, we recommend that you go to "SQL Optimization" to perform "ANALYZE TABLE" on the target tal

Sort by field (descending) Physical File Size ▾ Refresh Manually Data Collection Time: 2022-08-16 00:12

	Database Name	Table Name	Storage Engine	Physical File Size	Row Count	Total Used
●	test	test1	InnoDB	0.1 MB	6	

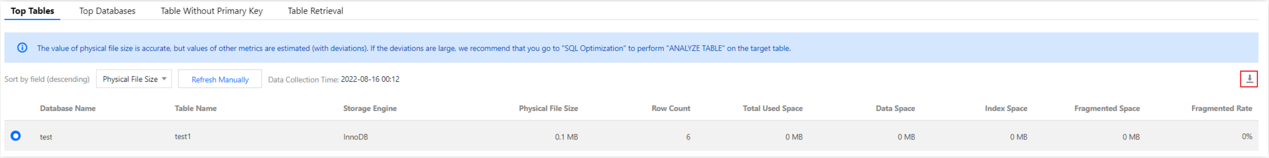
Table Info | **Trend**

Fragmented Rate

Space Usage

Physical File Size

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



Top Tables | Top Databases | Table Without Primary Key | Table Retrieval

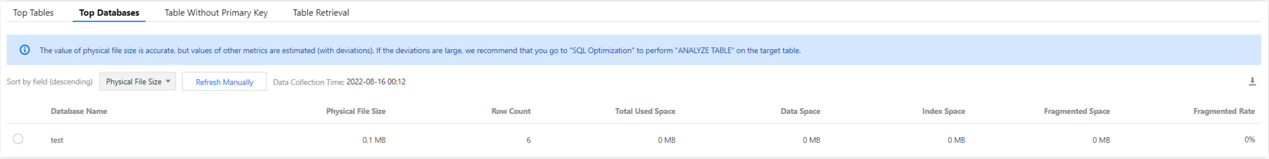
ⓘ The value of physical file size is accurate, but values of other metrics are estimated (with deviations). If the deviations are large, we recommend that you go to "SQL Optimization" to perform "ANALYZE TABLE" on the target table.

Sort by field (descending) Physical File Size ▾ Refresh Manually Data Collection Time: 2022-08-16 00:12

	Database Name	Table Name	Storage Engine	Physical File Size	Row Count	Total Used Space	Data Space	Index Space	Fragmented Space	Fragmented Rate
●	test	test1	InnoDB	0.1 MB	6	0 MB	0 MB	0 MB	0 MB	0%

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.



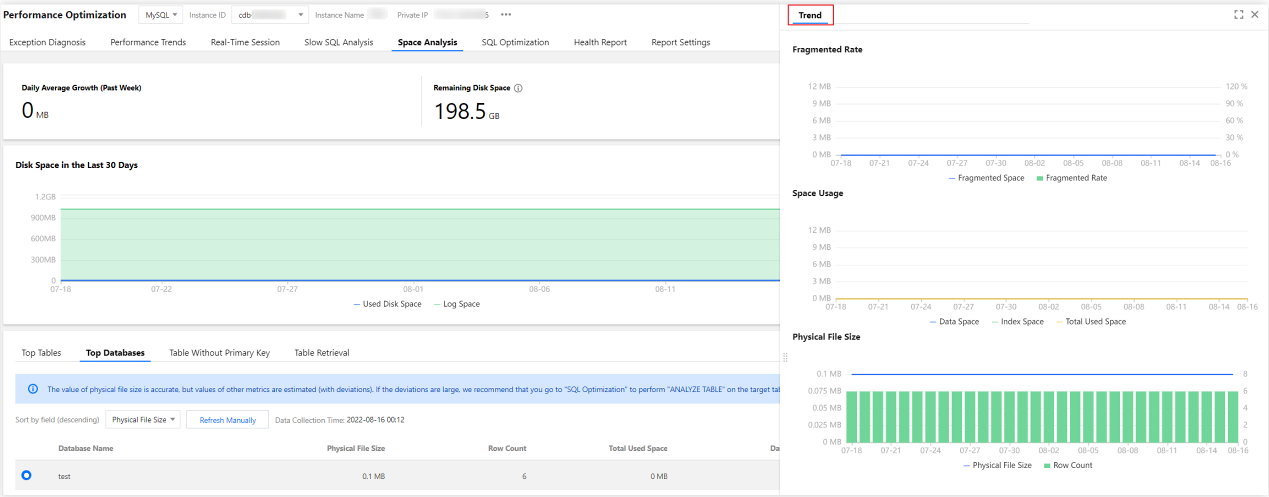
Top Tables | **Top Databases** | Table Without Primary Key | Table Retrieval

ⓘ The value of physical file size is accurate, but values of other metrics are estimated (with deviations). If the deviations are large, we recommend that you go to "SQL Optimization" to perform "ANALYZE TABLE" on the target table.

Sort by field (descending) Physical File Size ▾ Refresh Manually Data Collection Time: 2022-08-16 00:12

	Database Name	Physical File Size	Row Count	Total Used Space	Data Space	Index Space	Fragmented Space	Fragmented Rate
○	test	0.1 MB	6	0 MB	0 MB	0 MB	0 MB	0%

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**.



Daily Average Growth (Past Week): 0 MB

Remaining Disk Space: 198.5 GB

The value of physical file size is accurate, but values of other metrics are estimated (with deviations). If the deviations are large, we recommend that you go to "SQL Optimization" to perform "ANALYZE TABLE" on the target tal

Sort by field (descending): Physical File Size

Data Collection Time: 2022-08-16 00:12

	Database Name	Physical File Size	Row Count	Total Used Space
●	test	0.1 MB	6	0 MB

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:

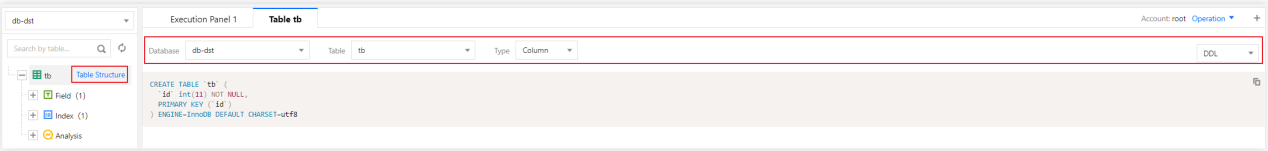
Currently, SQL optimization is supported only for TencentDB for MySQL (excluding basic single-node instances), TDSQL-C for MySQL, and self-built MySQL databases.

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.

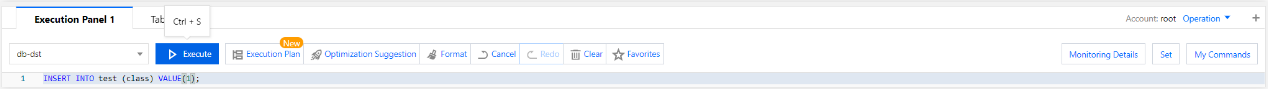


Database: db-dst
Table: tb
Type: Column
DDL

```
CREATE TABLE `tb` (
  `id` int(11) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
```

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.

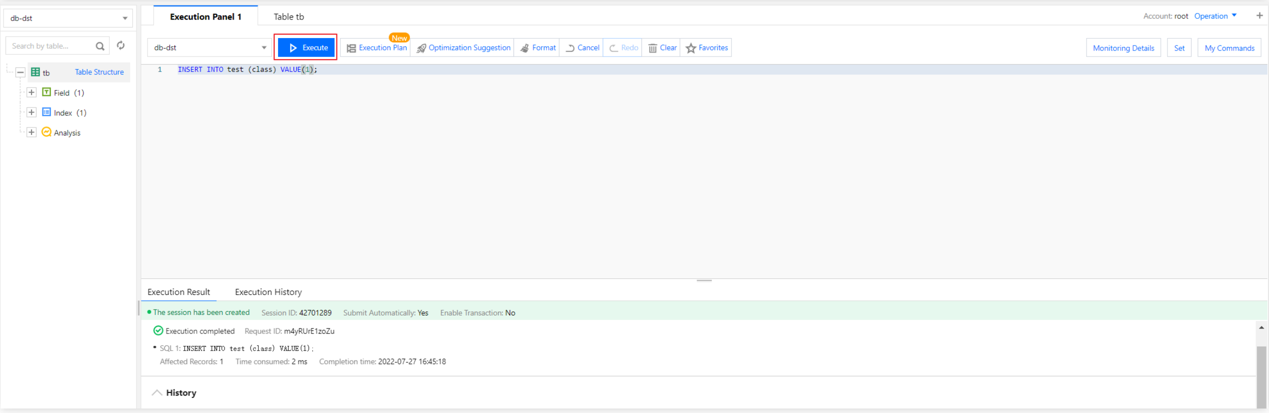


```
INSERT INTO test (class) VALUE(1);
```

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.



```
INSERT INTO test (class) VALUE(1);
```

● The session has been created Session ID: 42701289 Submit Automatically: Yes Enable Transaction: No

Execution completed Request ID: m4yRUrE1zoZu

- SQL 1: INSERT INTO test (class) VALUE(1);
  Affected Records: 1 Time consumed: 2 ms Completion time: 2022-07-27 16:45:18

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.

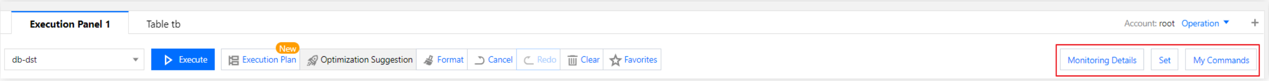
Click **Format** to format the selected SQL statement as shown below:

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.



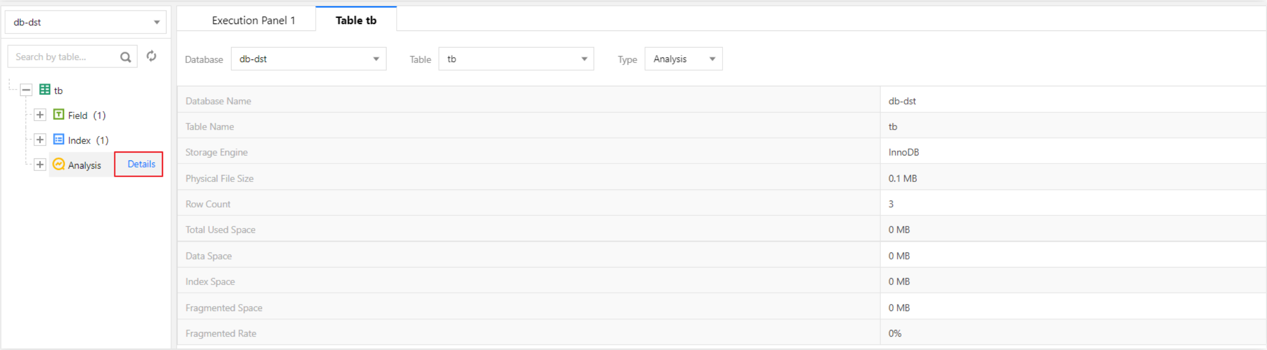
Click **Monitoring Details** on the right to view the monitoring information of the database instance.

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.



Database Name	db-dst
Table Name	tb
Storage Engine	InnoDB
Physical File Size	0.1 MB
Row Count	3
Total Used Space	0 MB
Data Space	0 MB
Index Space	0 MB
Fragmented Space	0 MB
Fragmented Rate	0%

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.

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.

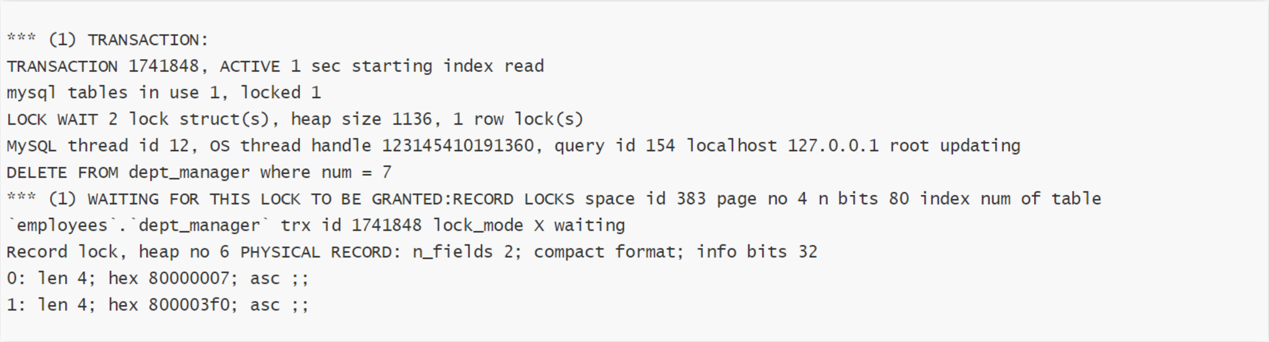
SQL information display: Executions are inferred through SQL parsing to help avoid deadlocks.

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.

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.

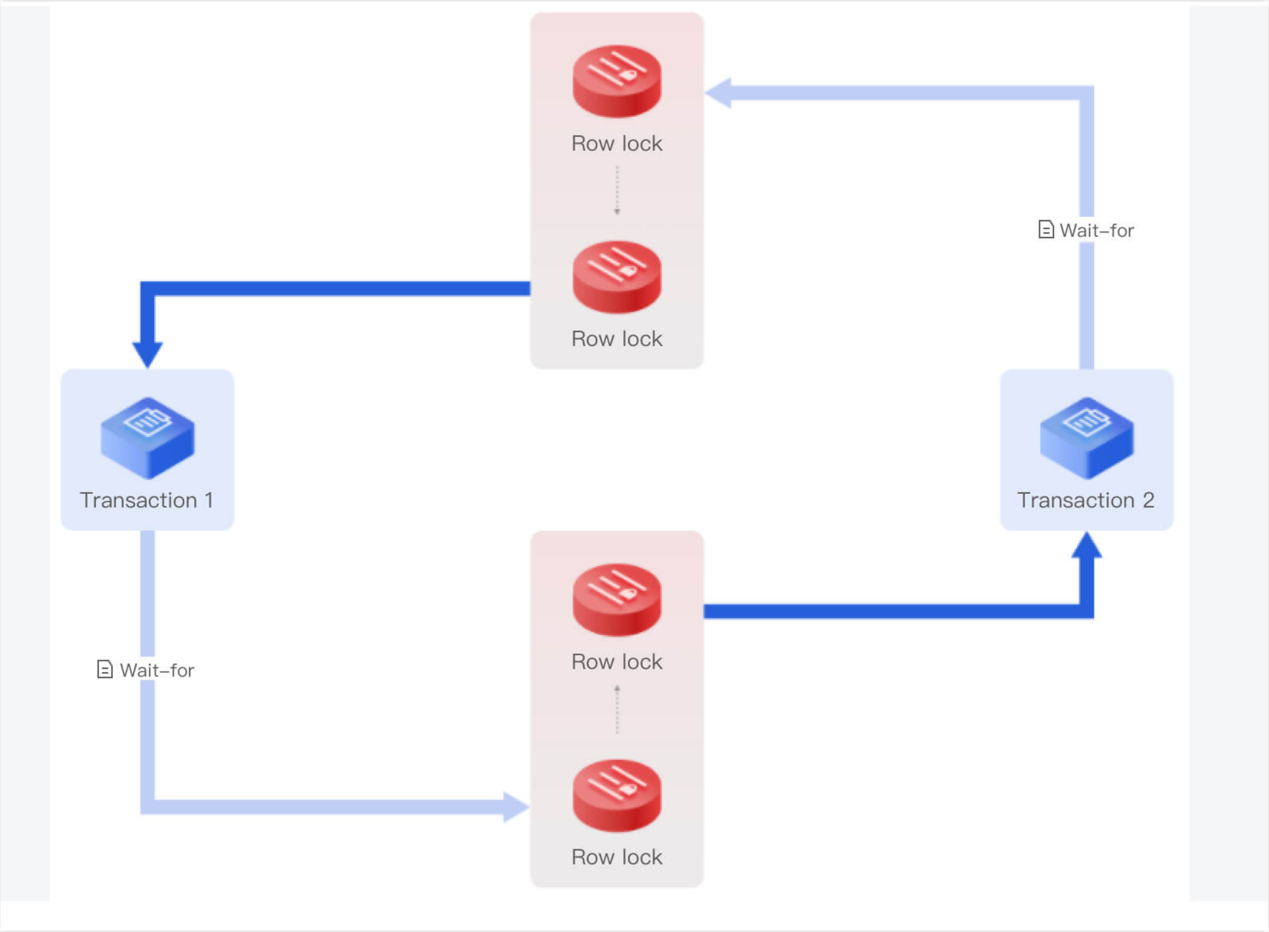


```
*** (1) TRANSACTION:
TRANSACTION 1741848, ACTIVE 1 sec starting index read
mysql tables in use 1, locked 1
LOCK WAIT 2 lock struct(s), heap size 1136, 1 row lock(s)
MySQL thread id 12, OS thread handle 123145410191360, query id 154 localhost 127.0.0.1 root updating
DELETE FROM dept_manager where 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

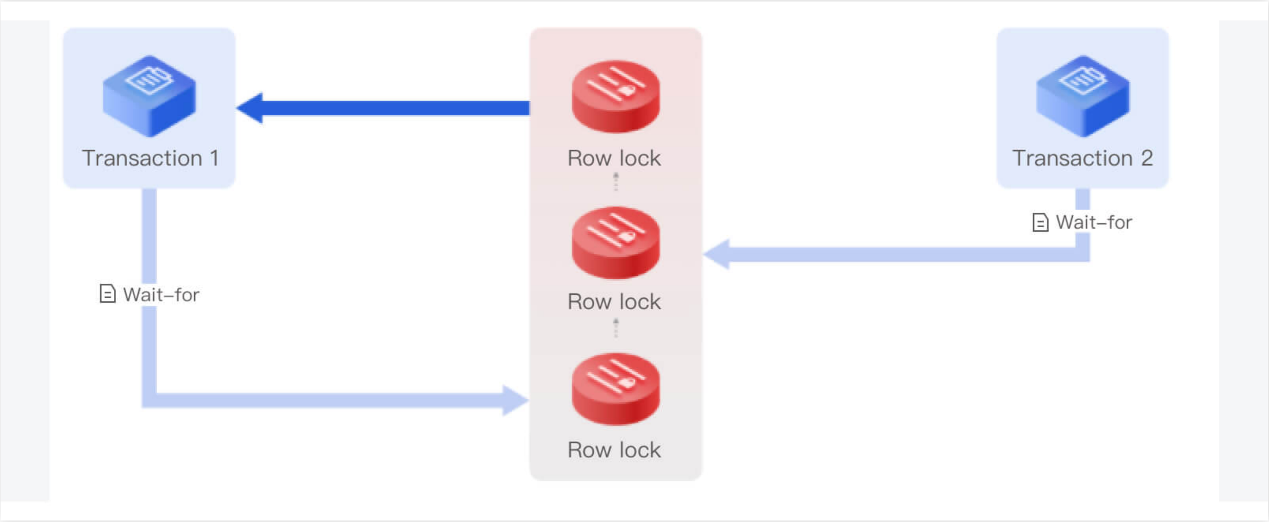
Transaction 1 and transaction 2 each hold a lock (represented by a dark blue line) and request a lock (represented by a light blue line). The lock held by transaction 1 blocks the lock requested by transaction 2, and the lock requested by transaction 1 is blocked by the lock held by transaction 2, causing 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

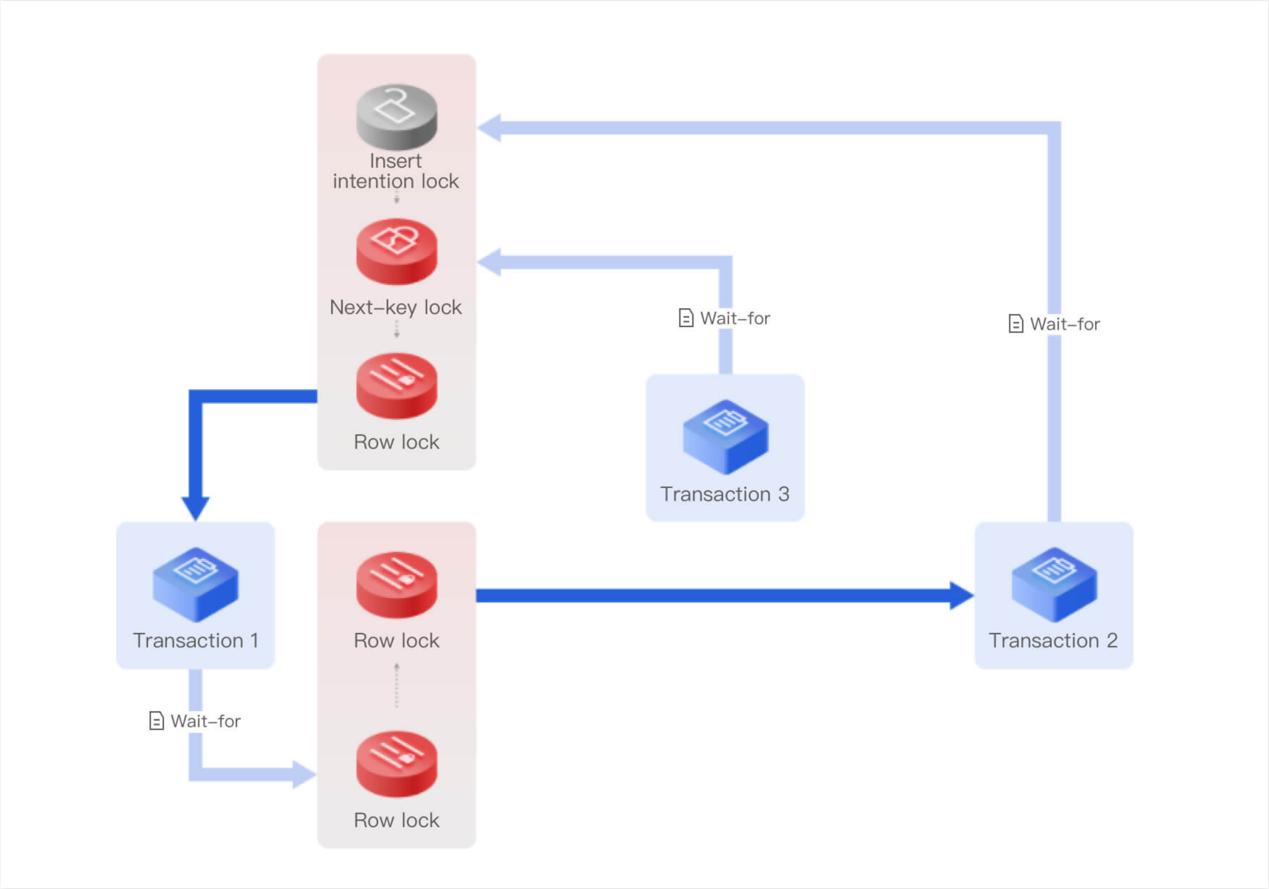
In MySQL, even if a waited unlock has not been acquired successfully (i.e., in waiting status), it can still block other lock requests. This is different from locks in operating systems.

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.

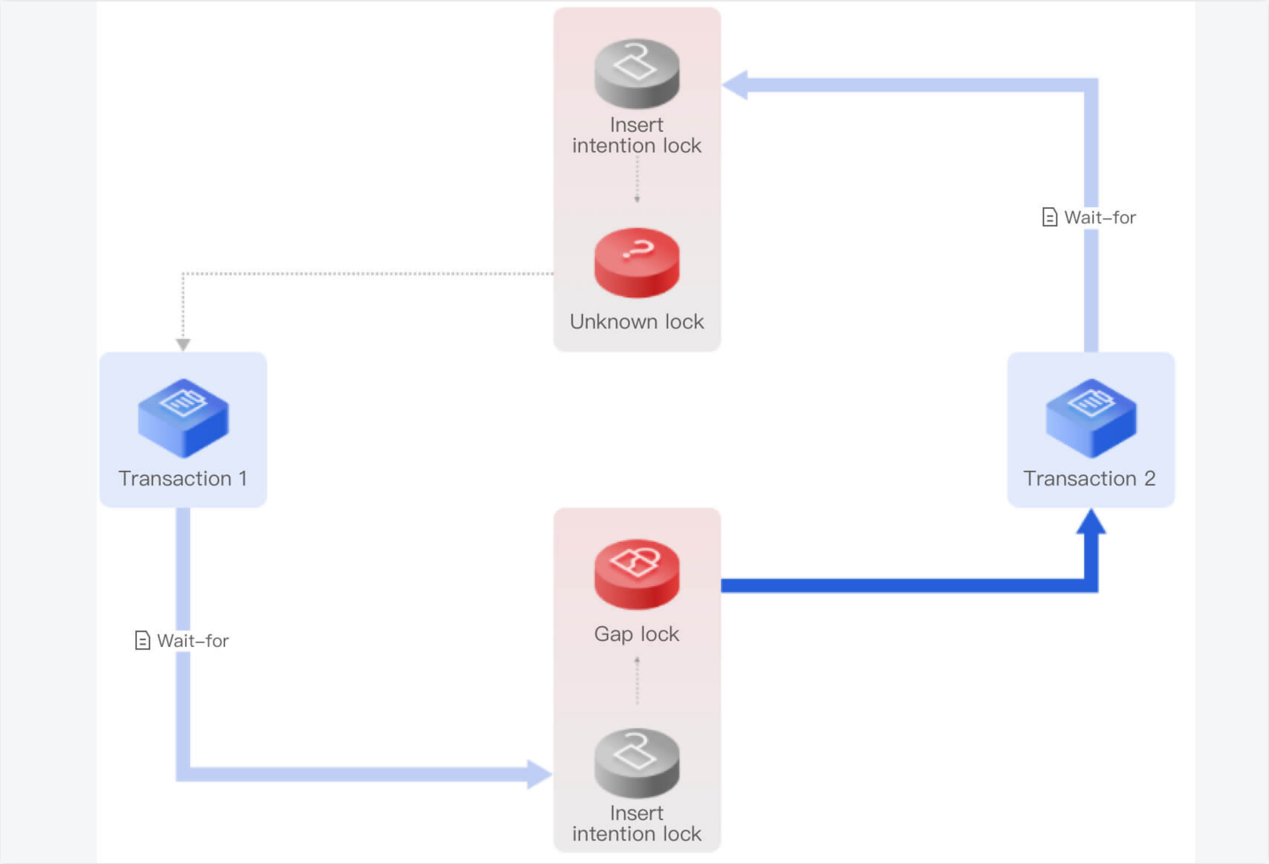


Sample 4. An "unknown lock" (for MySQL 5.6 and 5.7)

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.

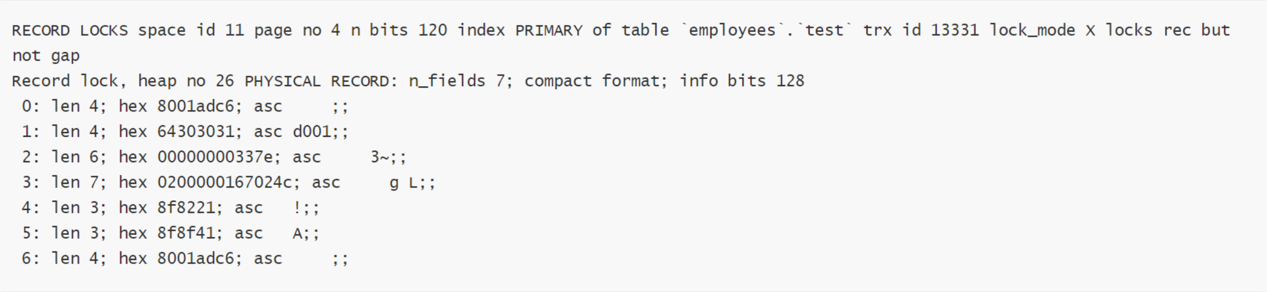
MySQL 5.6 and 5.7 only use the depth-first search method to search for cycles, but do not record cycles. Therefore, the deadlock logs record only the first and last transactions and are incomplete.

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.



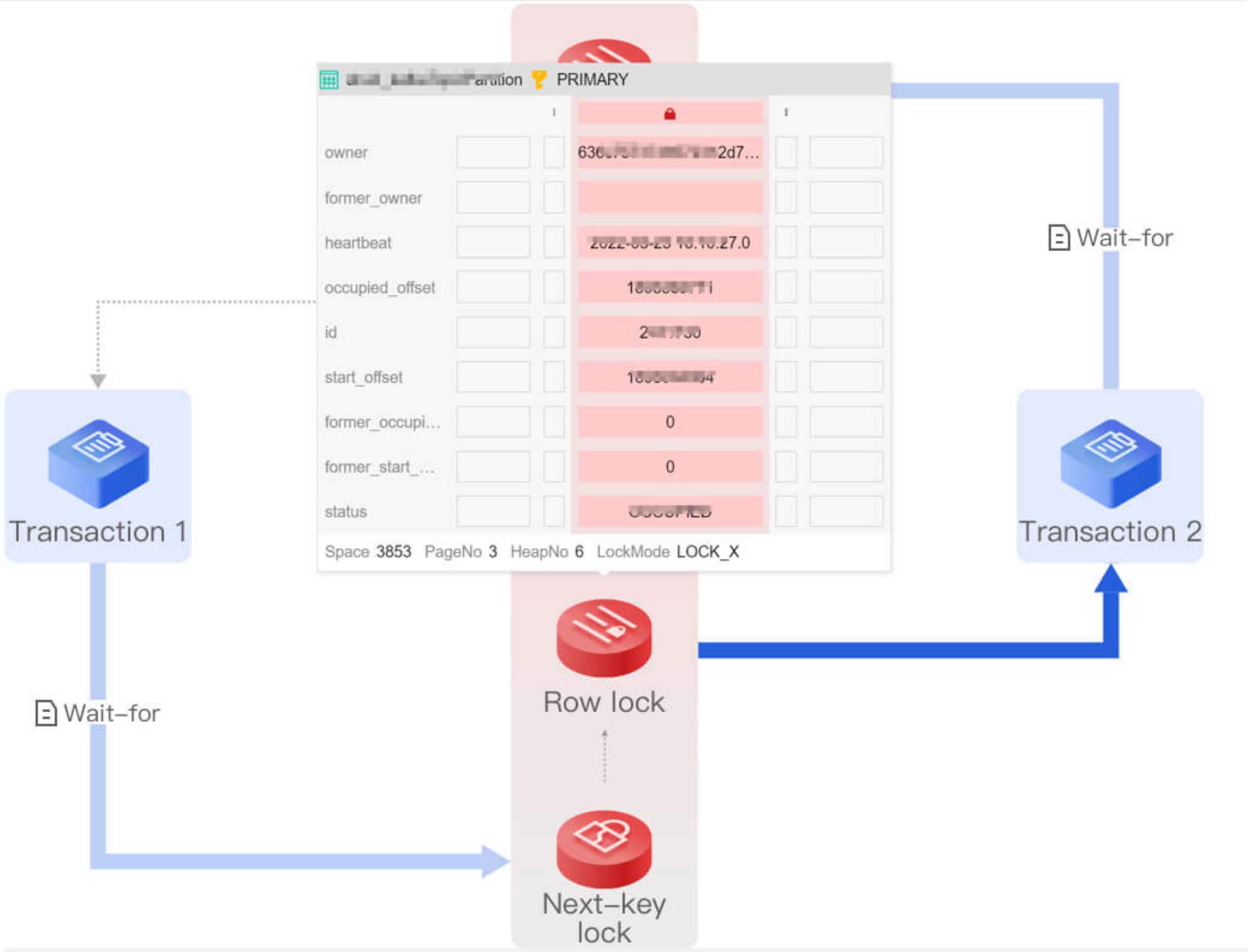
Lock Information Display

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 but
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.



[redacted]artition PRIMARY

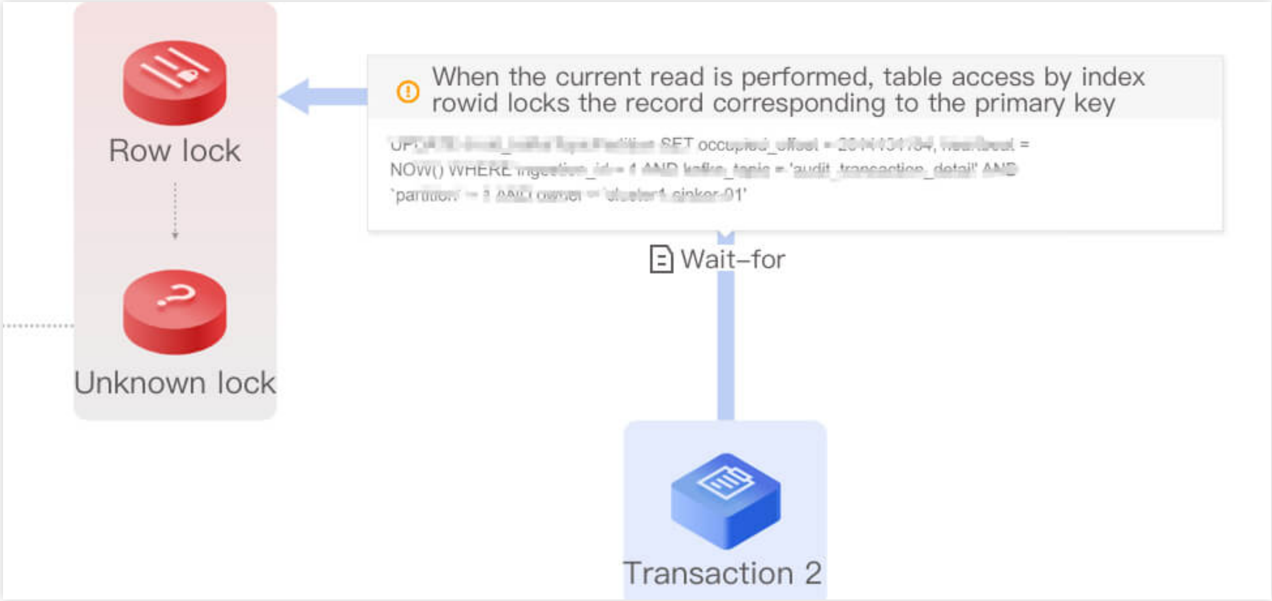
			🔒		
owner			[redacted]		
former_owner					
heartbeat			[redacted]		
occupied_offset			[redacted]		
id			[redacted]		
start_offset			[redacted]		
former_occupi...			0		
former_start_...			0		
status			[redacted]		

Space 3853 PageNo 3 HeapNo 6 LockMode LOCK_X

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.



When the current read is performed, table access by index rowid locks the record corresponding to the primary key

```
UP[redacted] SET occupied_offset = [redacted], [redacted] =
NOW() WHERE [redacted] AND [redacted] = 'audit_transaction_detail' AND
`partition` = [redacted] AND owner = [redacted]
```

Event Notification

Last updated：2022-08-11 15:45:29

For more information, see Event Notification.

Best Practices

Last updated：2022-08-11 15:45:29

Redis Performance Optimization Exception Diagnosis

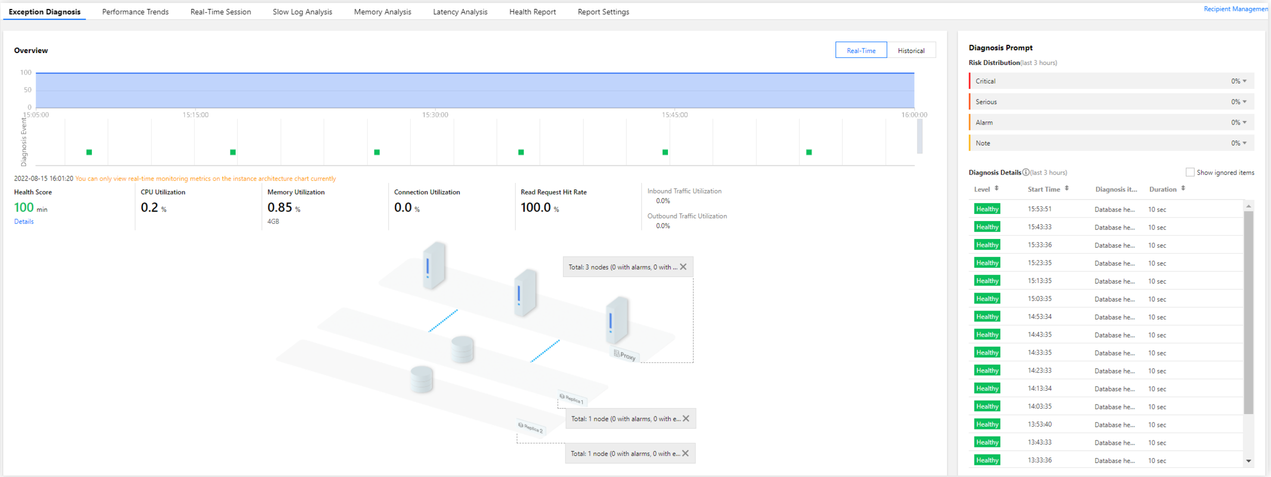
Last updated：2022-08-15 16:19:47

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.



2022-08-15 16:01:20 You can only view real-time monitoring metrics on the instance architecture chart currently

Health Score 100 min Details

CPU Utilization 0.2 %

Memory Utilization 0.85 % 4GB

Connection Utilization 0.0 %

Read Request Hit Rate 100.0 %

Inbound Traffic Utilization 0.0%

Outbound Traffic Utilization 0.0%

Total: 3 nodes (0 with alarms, 0 with ...

Total: 1 node (0 with alarms, 0 with e...

Total: 1 node (0 with alarms, 0 with e...

Diagnosis Prompt

Risk Distribution(last 3 hours)

Critical	0%
Serious	0%
Alarm	0%
Note	0%

Diagnosis Details(last 3 hours) □ Show ignored items

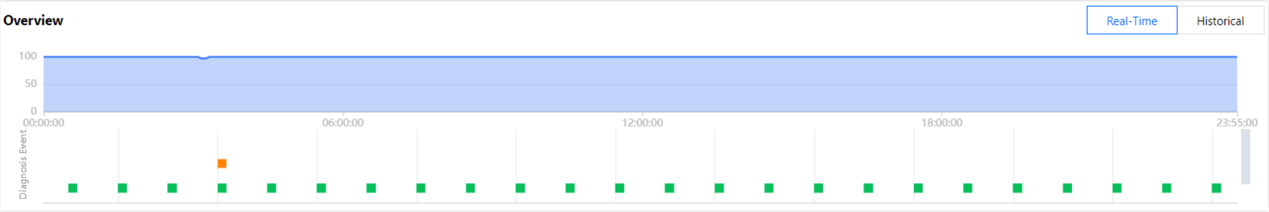
Level	Start Time	Diagnosis it...	Duration
Healthy	15:53:51	Database he...	10 sec
Healthy	15:43:33	Database he...	10 sec
Healthy	15:33:36	Database he...	10 sec
Healthy	15:23:35	Database he...	10 sec
Healthy	15:13:35	Database he...	10 sec
Healthy	15:03:35	Database he...	10 sec
Healthy	14:53:34	Database he...	10 sec
Healthy	14:43:35	Database he...	10 sec
Healthy	14:33:35	Database he...	10 sec
Healthy	14:23:33	Database he...	10 sec
Healthy	14:13:34	Database he...	10 sec
Healthy	14:03:35	Database he...	10 sec
Healthy	13:53:40	Database he...	10 sec
Healthy	13:43:33	Database he...	10 sec
Healthy	13:33:36	Database he...	10 sec

Viewing the monitoring overview

The **Overview** section displays the database's overall health score, exception diagnosis event timeline, topology, and other information.

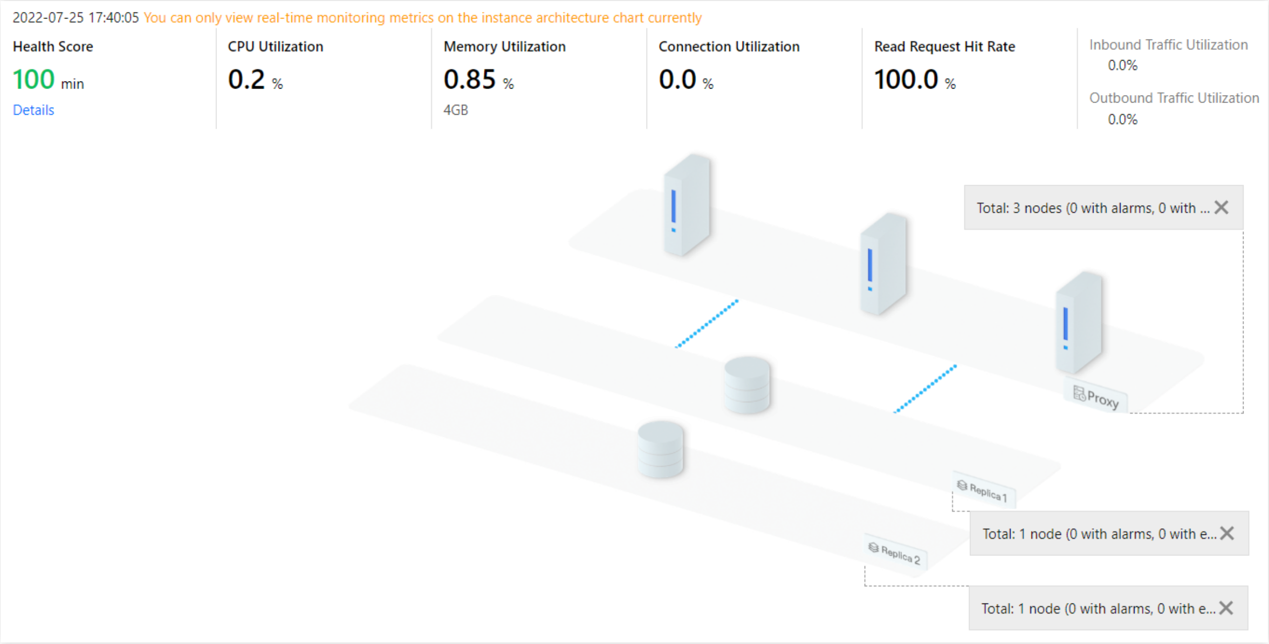
At the top of the **Overview** section, you can select **Real-Time** or **Historical** to view corresponding statistics.

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.

The topology displays the node structure of the instance and the location of the nodes where alarms occurred. Hover over a node to view its metric statistics.



2022-07-25 17:40:05 You can only view real-time monitoring metrics on the instance architecture chart currently

Health Score 100 min Details

CPU Utilization 0.2 %

Memory Utilization 0.85 % 4GB

Connection Utilization 0.0 %

Read Request Hit Rate 100.0 %

Inbound Traffic Utilization 0.0%

Outbound Traffic Utilization 0.0%

Total: 3 nodes (0 with alarms, 0 with ...

Total: 1 node (0 with alarms, 0 with e...

Total: 1 node (0 with alarms, 0 with e...

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.

Performance Trends

Last updated：2022-08-05 11:13:15

Feature description

You can select multiple performance metrics for Redis performance trends. Specifically, you can switch instances (Redis database instances), Redis nodes (between nodes, such as between node A and node B), and proxy nodes (middleware cluster nodes). You can also select performance metrics, such as real-time/historical views, monitoring granularities, single metric view/comparison view, and multiple views and comparison views of instances, Redis nodes, and proxy nodes.

Supported performance metrics

DBbrain currently supports monitoring the following performance metrics of TencentDB for Redis:

Category	Subcategory	Metric
Resource Monitoring	CPU	CPU
	Memory	Memory
		Memory Usage
	Storage Space	Storage Utilization
		Used Storage Space
	Traffic	Outbound Traffic
		Inbound Traffic
Redis	Key Information	Total Keys, Expired Keys, Evicted Keys
	Memory	Memory Usage
		Memory Utilization
	Replication Delay	Replication Delay
	Network Usage	Network Usage
	Request	Total Requests

		Read Request
		Write Request
		Other Requests
	Response	Slow Queries
		Read Request Hit
		Read Request Miss
		Read Request Hit Rate
proxy	CPU	CPU Utilization
	Traffic	Inbound Traffic
		Outbound Traffic
	Request	Total Requests
		Key Requests
		Mget Requests
		Execution Error
		Big Value Request
	Network Usage	Connections
		Connections per Sec
		Disconnections per Sec
		Abnormal Disconnections per Sec
	Network Utilization	Connection Utilization
		Inbound Traffic Utilization
		Outbound Traffic Utilization
		Inbound Traffic Throttling Trigger
		Outbound Traffic Throttling Trigger
		Avg Execution Latency

	Latency	Max Execution Latency
		P99 Execution Latency
		Avg Read Latency
		Avg Write Latency
		Avg Latency of Other Commands

Viewing performance trends

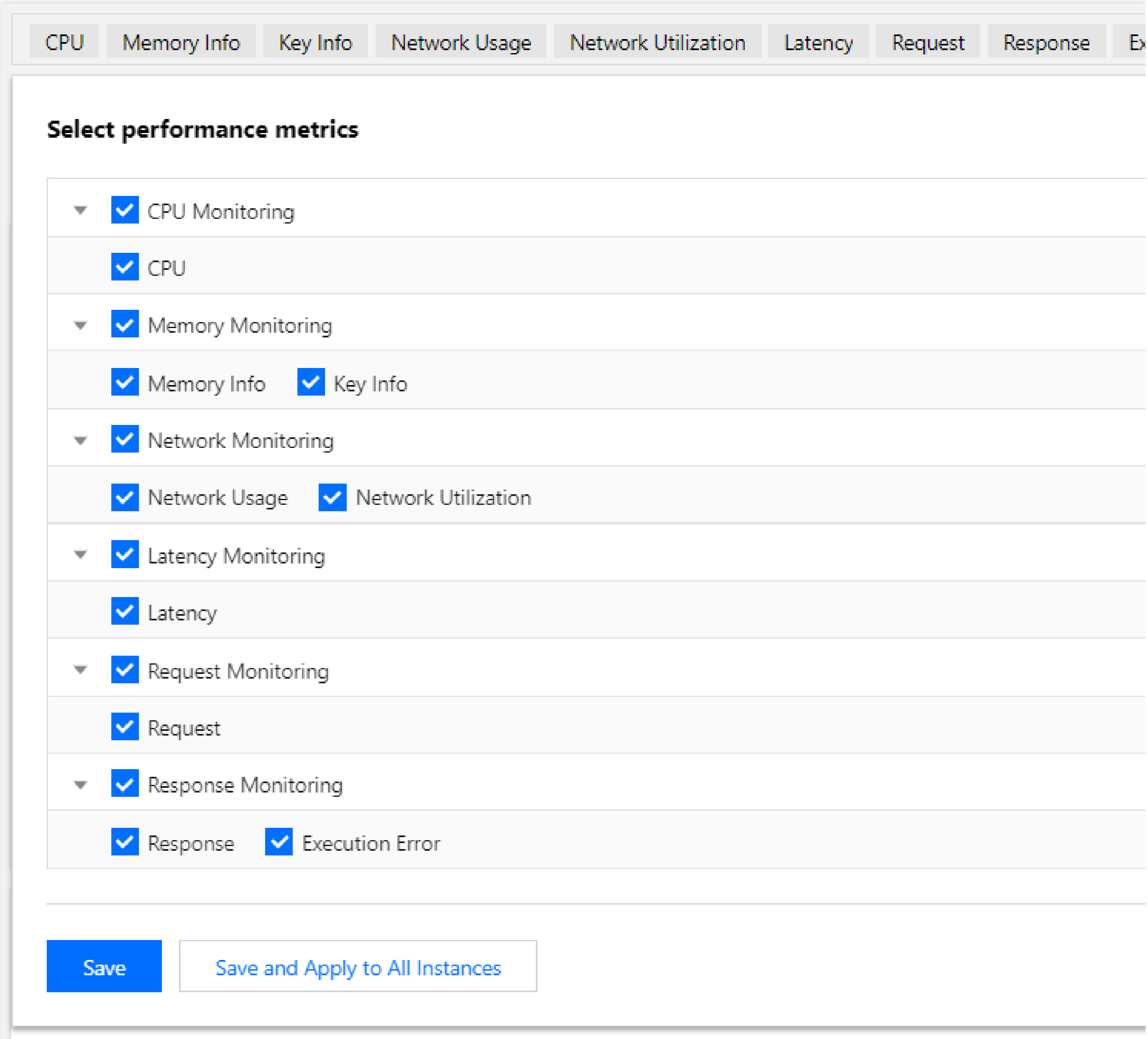
1. Log in to the DBbrain console and select **Performance Optimization** on the left sidebar. On the displayed page, select a database type and an instance at the top, and select the **Performance Trends** tab.

2. Set monitoring metrics.

Metric categories: Include CPU, memory, network, latency, request, and response.

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

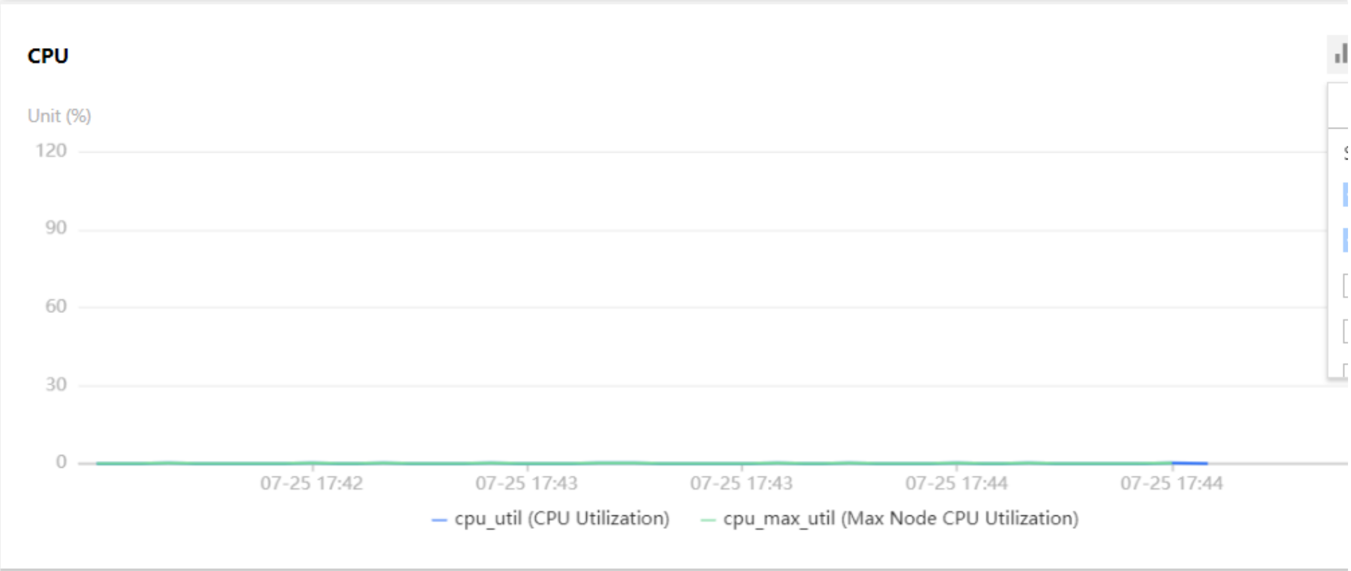
Filter global metrics



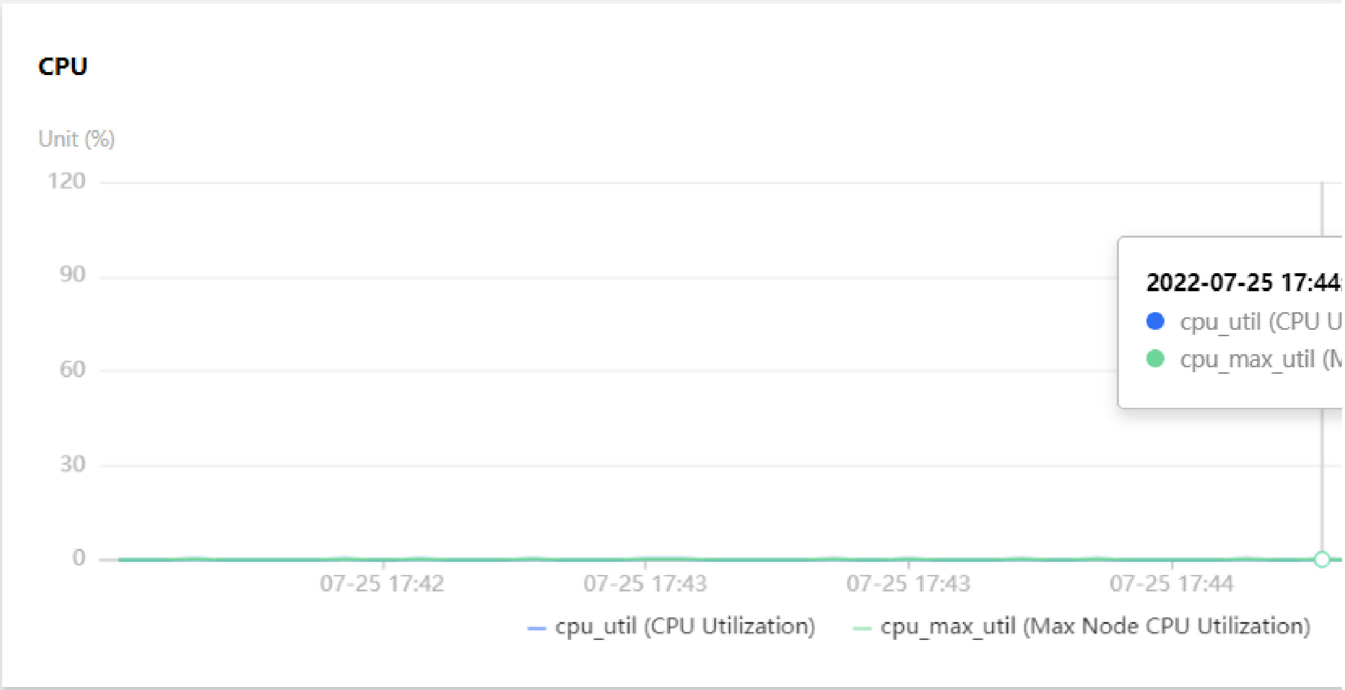
Select performance metrics

- ☑ CPU Monitoring
  - ☑ CPU
- ☑ Memory Monitoring
  - ☑ Memory Info ☑ Key Info
- ☑ Network Monitoring
  - ☑ Network Usage ☑ Network Utilization
- ☑ Latency Monitoring
  - ☑ Latency
- ☑ Request Monitoring
  - ☑ Request
- ☑ Response Monitoring
  - ☑ Response ☑ Execution Error

Filter one single metric



Switch between chart views



3. Set the monitoring dimension.

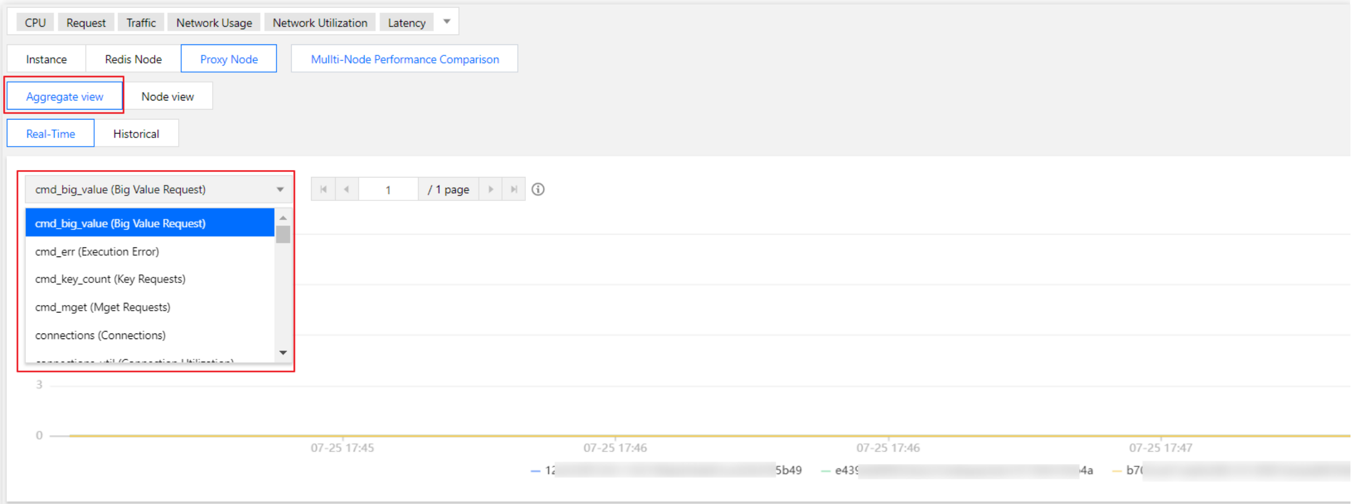
You can set the monitoring dimension to **Instance**, **Redis Node**, or **Proxy Node**.

Instance: It displays the monitoring view of the entire instance.

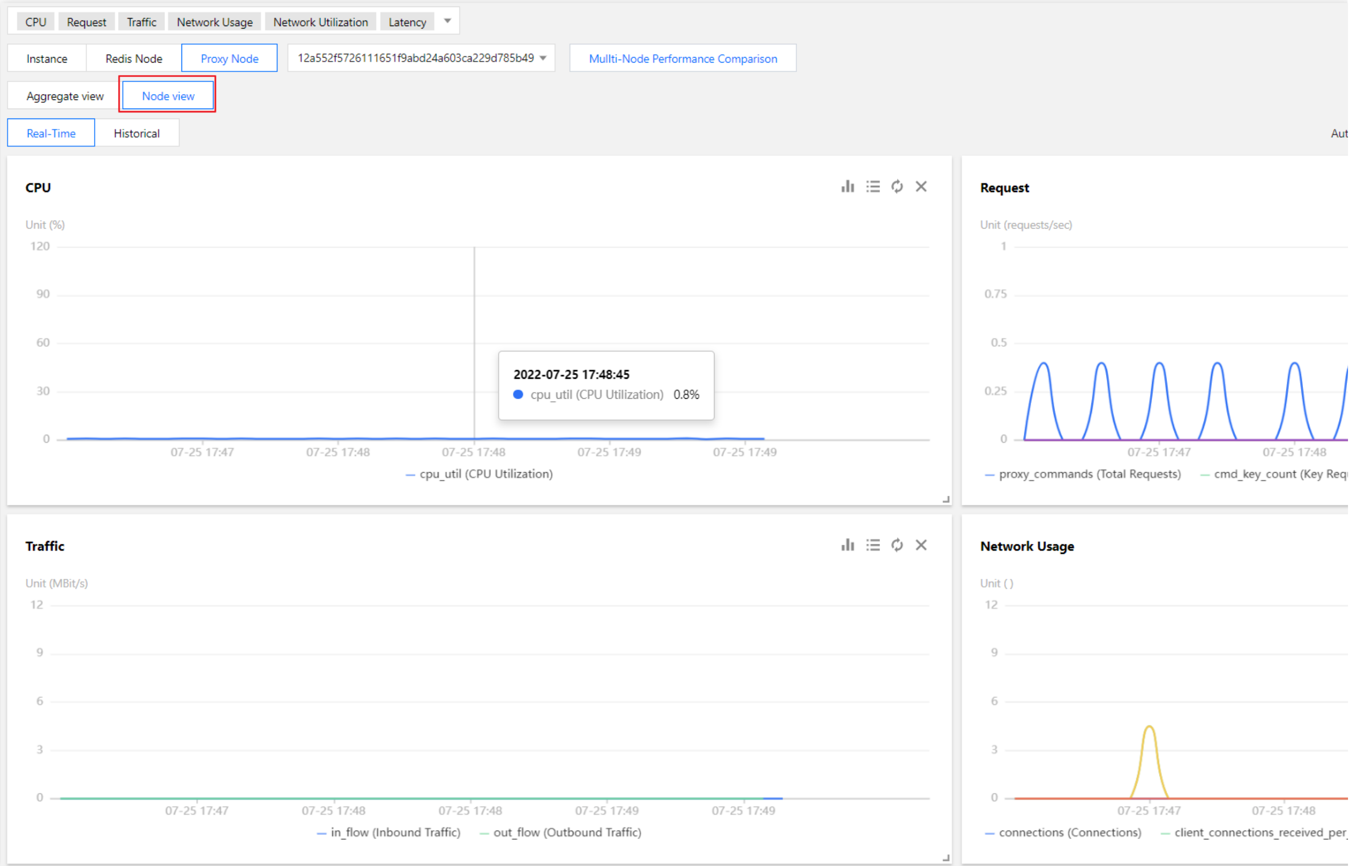
Redis Node: It displays the comparison trend views of relevant metrics on each Redis node.

Proxy Node: It displays the comparison trend views of relevant metrics on each proxy node.If you select **Proxy Node**, you can select the **Aggregate view** or **Node view**.

In **Aggregate view** mode, the information of all proxy nodes is displayed. You need to select a metric in the top-left corner to display the single-metric information of all nodes. Click **Details** next to each metric to enter the **Node view**.



In **Node view** mode, the information of all monitoring metrics of a node is displayed.

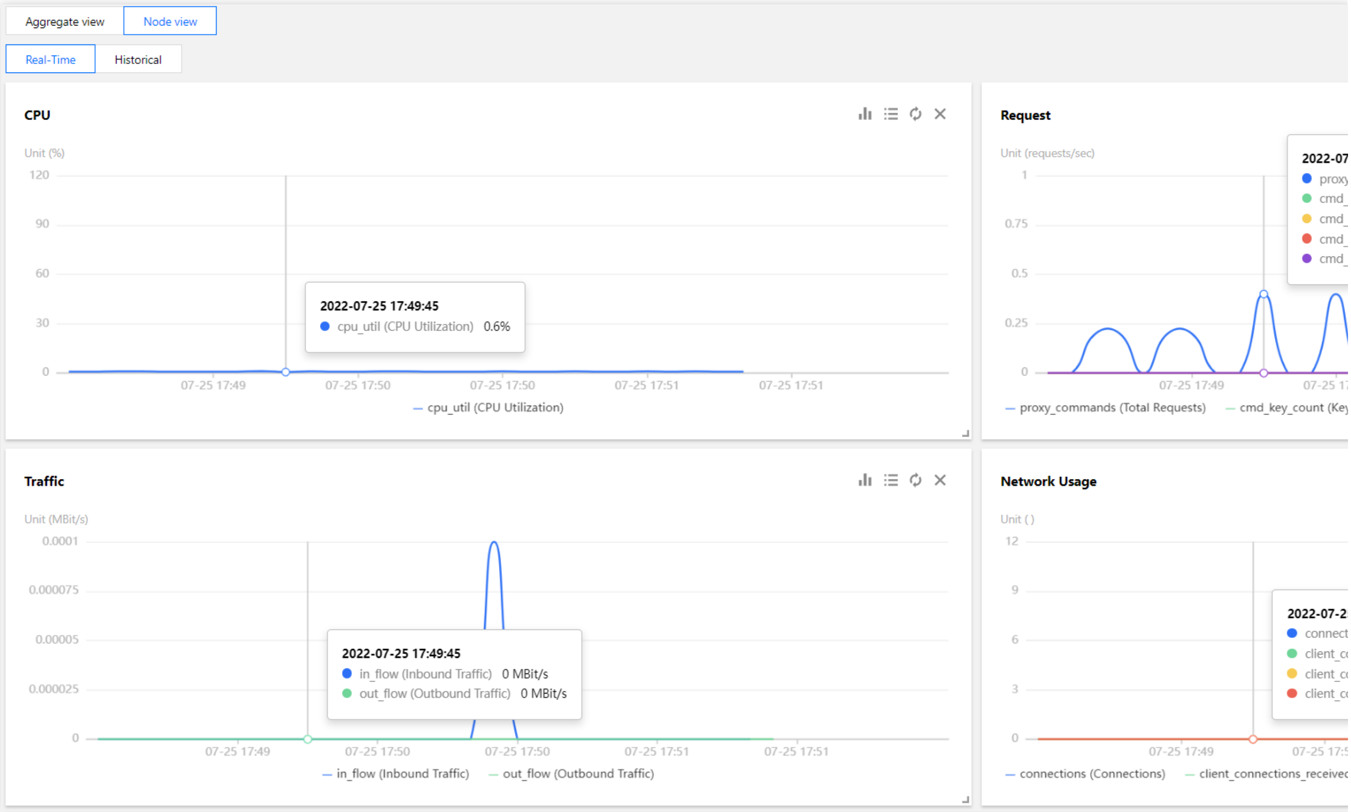


4. Switch between the **Real-Time** and **Historical** views.

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

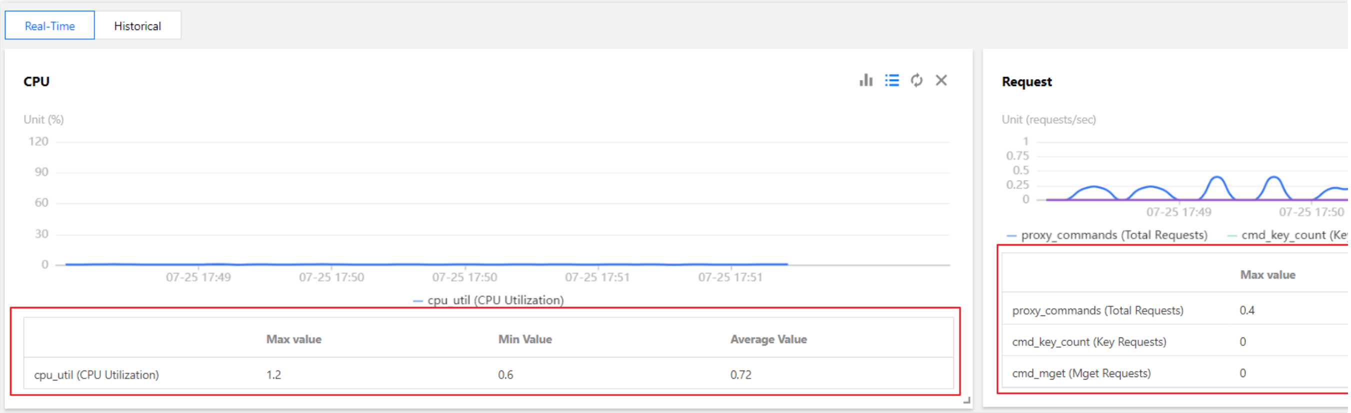
5. Enable chart interaction.For one single instance, node, or proxy, you can view relevant metric trend comparison, add custom metrics, and view the performance metric trend comparison by time.

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**.



6. Show statistics.

Metric monitoring charts support displaying specific monitoring data simultaneously. After **Show Statistics** is enabled, the table data will be displayed below each monitoring chart.

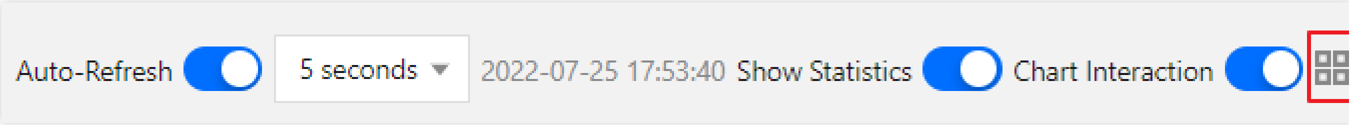


	Max value	Min Value	Average Value
cpu_util (CPU Utilization)	1.2	0.6	0.72

	Max value
proxy_commands (Total Requests)	0.4
cmd_key_count (Key Requests)	0
cmd_mget (Mget Requests)	0

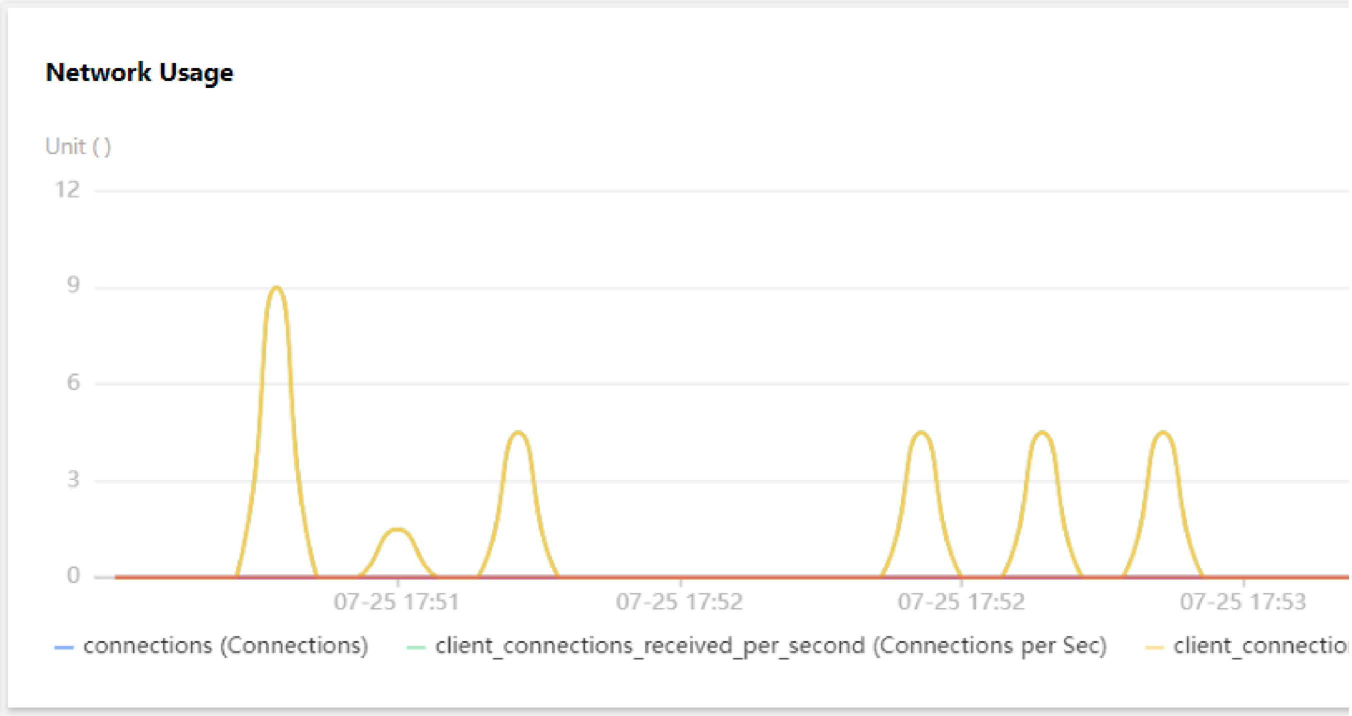
7. Switch between the one-column and two-column modes, drag a monitoring view, or zoom in a monitoring view.

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.

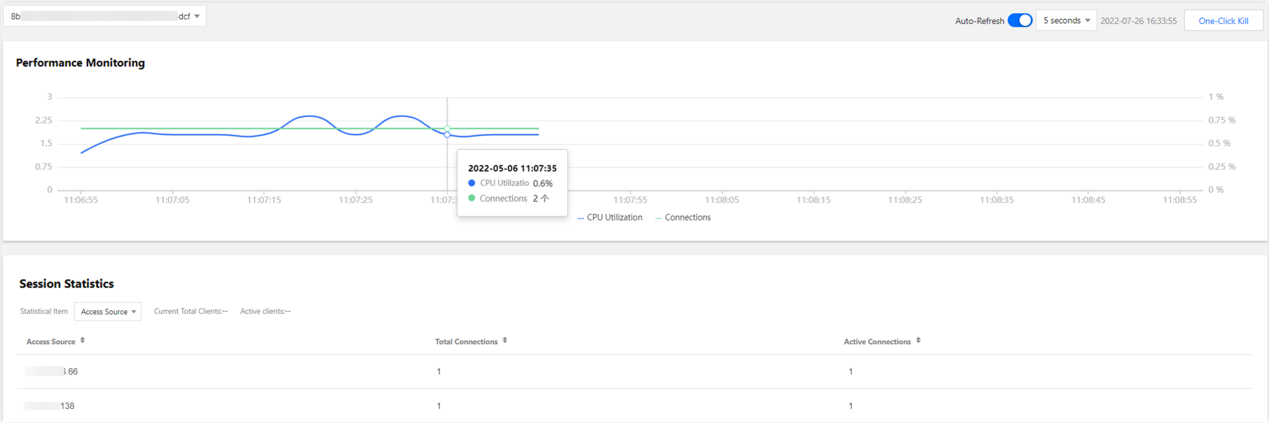


Real-Time Session

Last updated：2022-08-05 11:13:15

Feature description

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



8b[redacted]dcf ▾

Auto-Refresh: on, 5 seconds ▾, 2022-07-26 16:33:55

Performance Monitoring

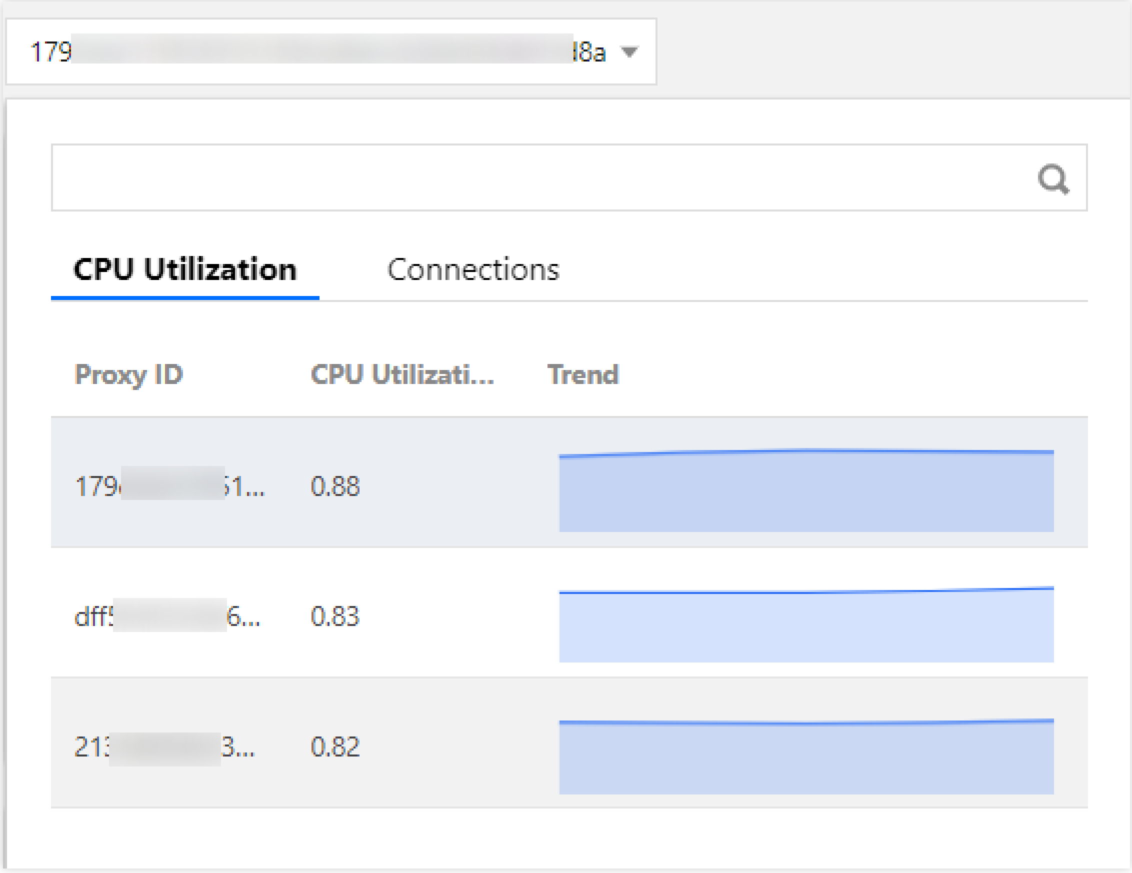
Session Statistics

Statistical Item: Access Source ▾ Current Total Clients:-- Active clients:--

Access Source	Total Connections	Active Connections
[redacted].66	1	1
[redacted]138	1	1

Performance monitoring/Session statistics

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.
2. Select the **Proxy ID** in the top-left corner to display the information of **Performance Monitoring** and **Session Statistics**.<br>

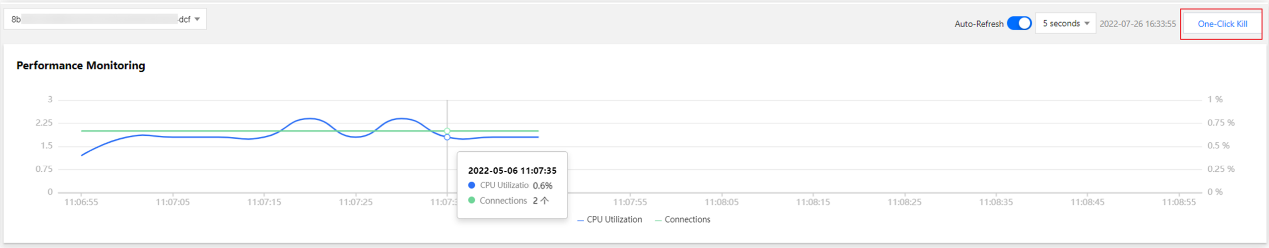


179[redacted]l8a

Proxy ID	CPU Utilizati...	Trend
179[redacted]1...	0.88	
dff[redacted]6...	0.83	
21[redacted]3...	0.82	

One-click kill

Click **One-Click Kill** to kill all sessions.



8b[redacted]dcf

Auto-Refresh 5 seconds 2022-07-26 16:33:55

Slow Log Analysis

Last updated：2022-08-05 11:13:15

Overview

Slow log analysis in Redis is different from that in MySQL and TDSQL-C and counts the slow logs in two dimensions: instance and proxy.

In the instance (Redis database instance) dimension, you can clearly view the CPU utilization, number of slow queries, consumed time statistics by log segment, and information of the entire slow log list.

In the proxy (middleware cluster node) dimension, you can view the proxy's slow log statistics, consumed time statistics by segment, and details of the slow log list.

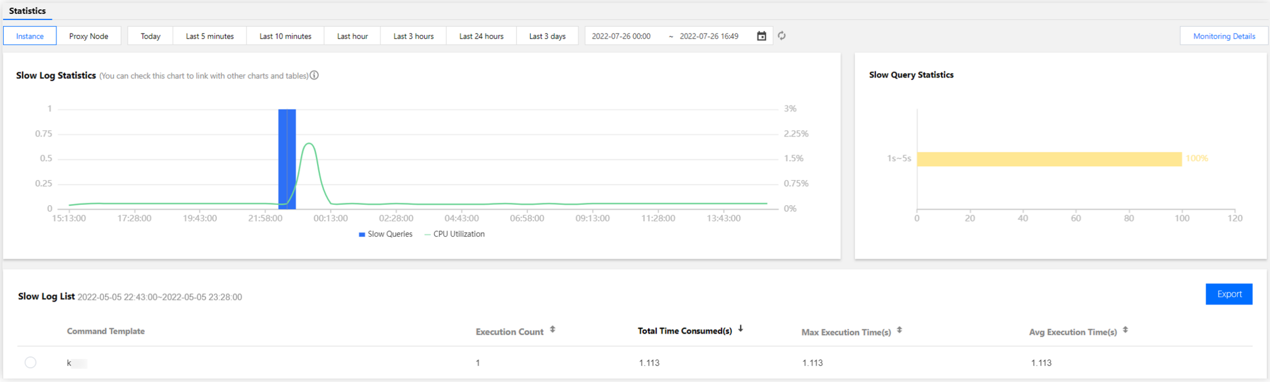
Directions

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 **Slow Log Analysis** tab.

2. On the **Slow Log Analysis** tab, you can view instance-level and proxy-level slow logs.

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**.

Instance-level slow log:



Slow Log List 2022-05-05 22:43:00~2022-05-05 23:28:00

	Command Template	Execution Count	Total Time Consumed(s)	Max Execution Time(s)	Avg Execution Time(s)
○	k	1	1.113	1.113	1.113

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 overall consumed time distribution of slow logs in the selected time range. The horizontal axis is the proportion of slow logs, and the vertical axis is the statistical period. When the cursor is over a certain statistical period, the proportion of slow logs in this period will be displayed.

Slow Log List: Click a slow log to view its analysis and statistics details.

Proxy-level slow log.

3. Perform chart interaction for logs.

In the **Slow Log Statistics** module, click the time point you want to locate, and the information of the slow log generated at the time point and specific time consumed will be synchronously displayed.

4. On the **Slow Log Analysis** tab, click **Monitoring Details** in the top-right corner to add multiple time ranges or monitoring metrics for comparison.

5. In the **Slow Log List** below, click an aggregated command template, and specific command analysis and statistics will be displayed on the right.

On the **Analysis** tab, you can view:

Command template.

Command sample.

Optimization suggestion and description.

On the **Statistics** tab, you can view:

Execution duration distribution ranges of the specified type of aggregated commands.

Access distribution and proportion of source IPs (available for proxy nodes but not Redis nodes).

6. Export the data of slow log analysis.

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

Memory Analysis (Big Key Analysis)

Last updated：2022-08-05 11:13:15

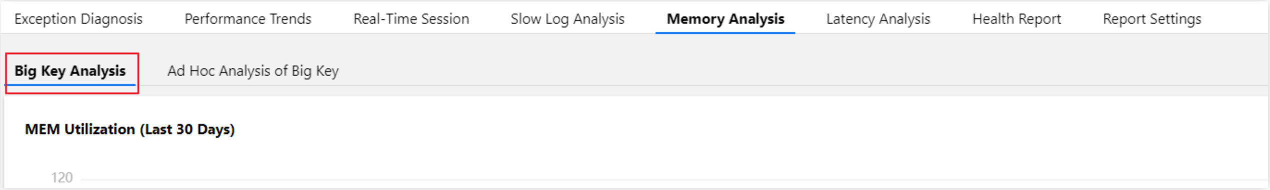
DBbrain supports big key analysis for Redis, which can quickly find big keys on instances and dynamically display the statistical analysis results of the top 100 big keys. This helps you locate memory usage by big keys, element information, and expiration time, so as to avoid service performance decline and insufficient memory issues caused by big keys.

Note:

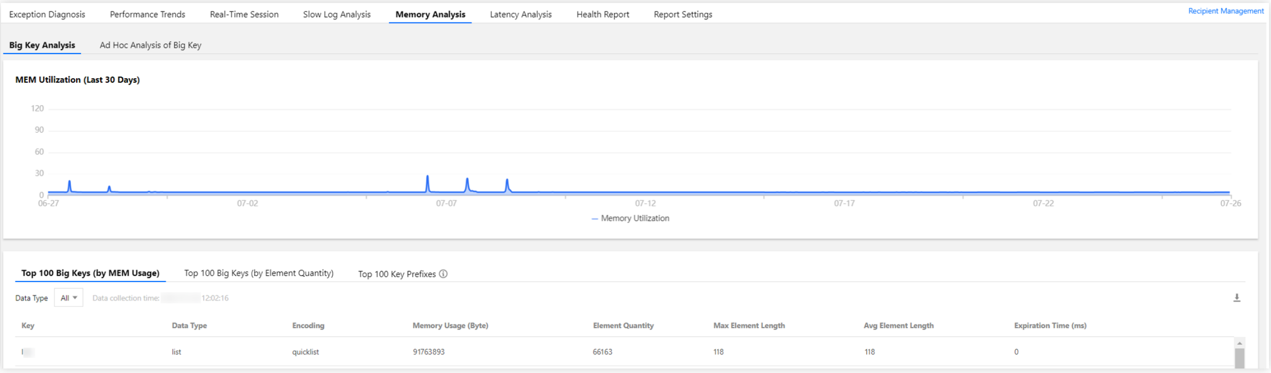
Currently, big key inspection is not supported for overlarge Redis clusters or Redis clusters with too many shards. You can perform an additional analysis by creating an **Ad Hoc Analysis of Big Key** task.

Viewing the big key analysis result

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 **Memory Analysis** tab.



2. Select a data type (based on memory, quantity, or prefix) below. The top 100 big keys of each data type are analyzed.



Data Type All ▾ Data collection time: [redacted] 12:02:16

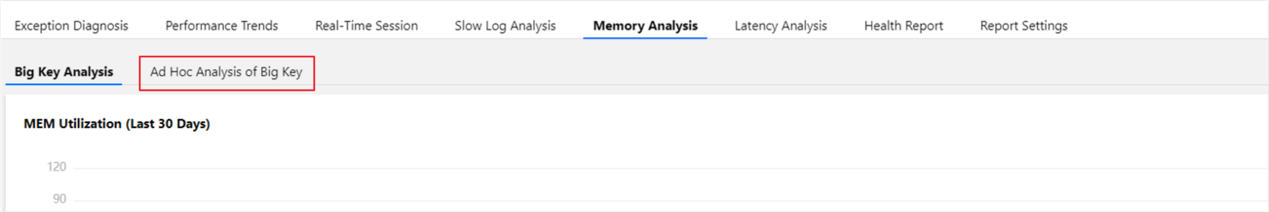
Key	Data Type	Encoding	Memory Usage (Byte)	Element Quantity	Max Element Length	Avg Element Length	Expiration Time (ms)
[redacted]	list	quicklist	91763893	66163	118	118	0

3. In the **MEM Utilization (Last 30 Days)** module, drag the timeline to view the big key analysis/memory status in the last 30 days.

The trend displays the memory usage in the last 30 days. Click a day, and the time column will be fixed. At this time, the big keys on the corresponding day will be displayed in the list below.

Ad hoc analysis of big key

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 **Memory Analysis** tab to view the **Ad Hoc Analysis of Big Key** feature.



2. Click **Create Task**, and DBbrain will fetch the last backup of the database for automatic analysis. You can view the analysis progress on the progress bar in the task list.
3. After the analysis is completed, you can view the analysis result in the task list.

Latency Analysis

Last updated：2022-08-15 15:15:15

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

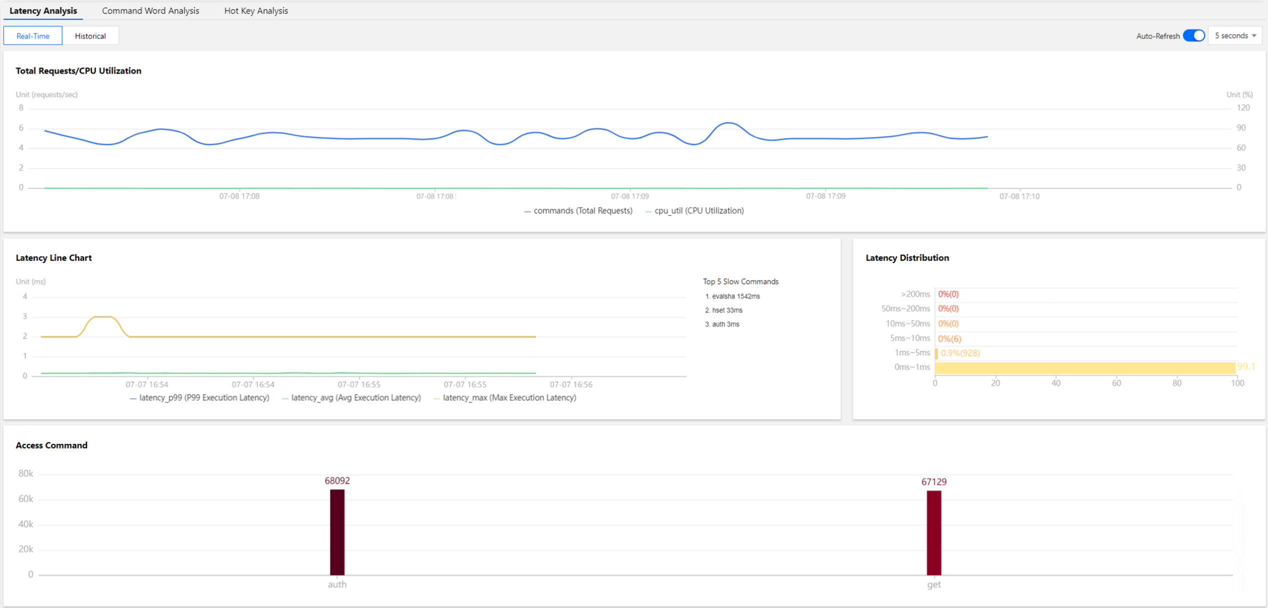
1. Log in to the DBbrain console and select **Performance Optimization** on the left sidebar. On the displayed page, select a database type and an instance at the top, and select the **Latency Analysis** tab.
2. On the **Latency Analysis** tab, you can view real-time/historical latency analysis and latency details.

Real-time/Historical analysis

Latency and latency distribution

The **Latency Distribution** section displays the percentages of different latency durations, giving you a quick overview of your overall business latency.

Time-consuming commands and number of hits



Auto-Refresh: 5 seconds

Top 5 Slow Commands
1. evalsha 1542ms
2. hset 33ms
3. auth 3ms

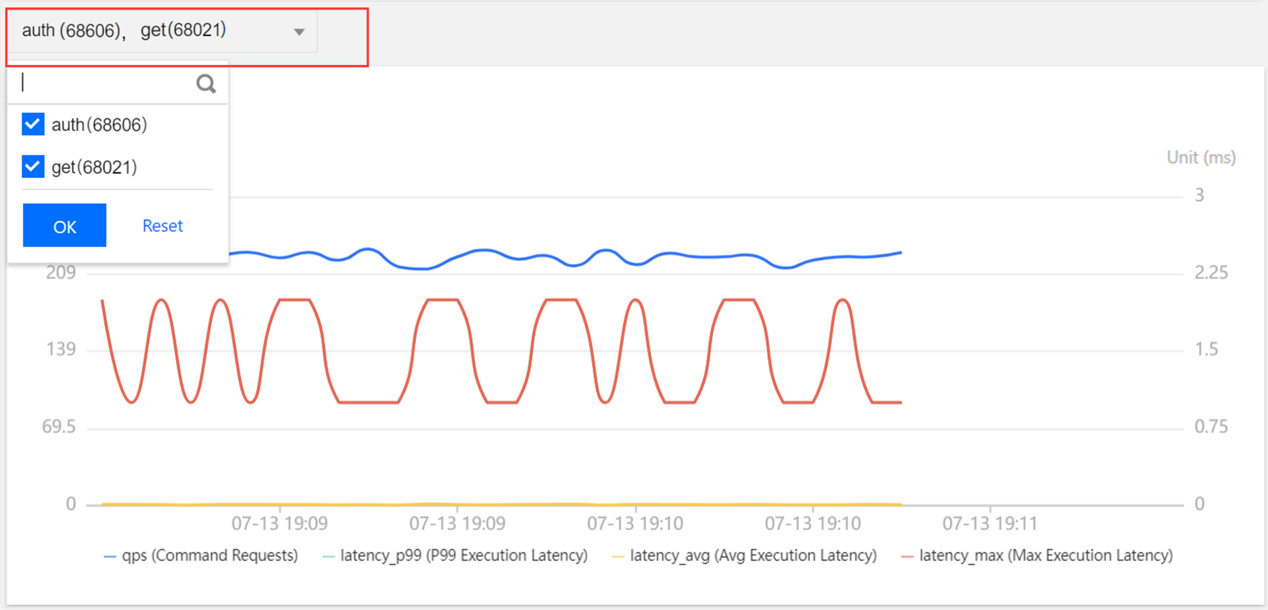
Latency Analysis (Command Word Analysis)

Last updated：2022-08-15 15:16:02

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 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 **Latency Analysis** > **Command Word Analysis** tab.
2. On the **Command Word Analysis** tab, you can view real-time and historical conditions.
3. You can filter command words in the top-left corner to view corresponding analysis results.



Latency Analysis (Hot Key Analysis)

Last updated：2022-08-05 11:13:15

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.

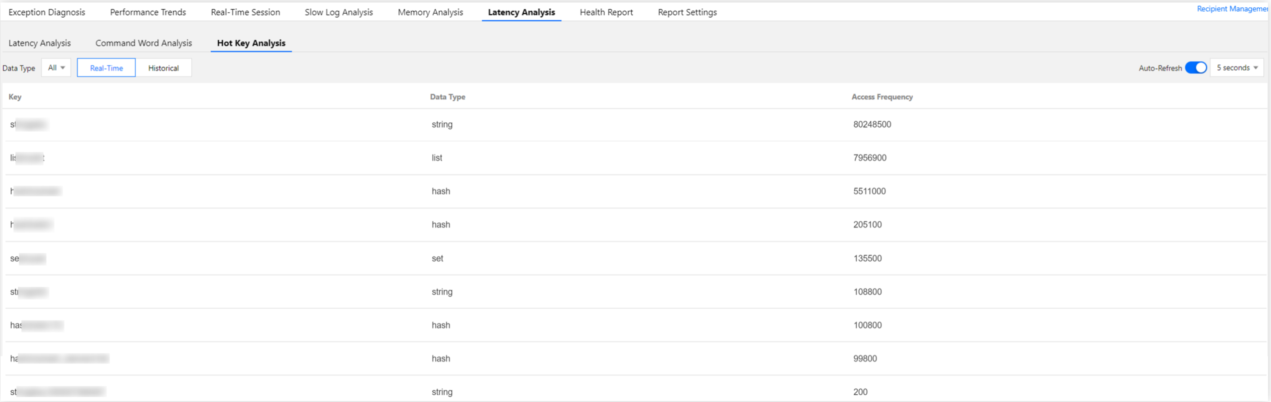
With DBbrain's hot key analysis feature, you can find hot keys quickly to optimize the service accordingly.

Viewing hot key analysis

1. Log in to the DBbrain console 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 **Latency Analysis** > **Hot Key Analysis** tab.
2. On the **Hot Key Analysis** tab, you can switch between the real-time and historical views.

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.



Data Type All ▾

Auto-Refresh 5 seconds ▾

Key	Data Type	Access Frequency
st[redacted]	string	80248500
lis[redacted]	list	7956900
h[redacted]	hash	5511000
h[redacted]	hash	205100
se[redacted]	set	135500
str[redacted]	string	108800
has[redacted]	hash	100800
ha[redacted]	hash	99800
st[redacted]	string	200

MongoDB Performance Optimization Exception Diagnosis

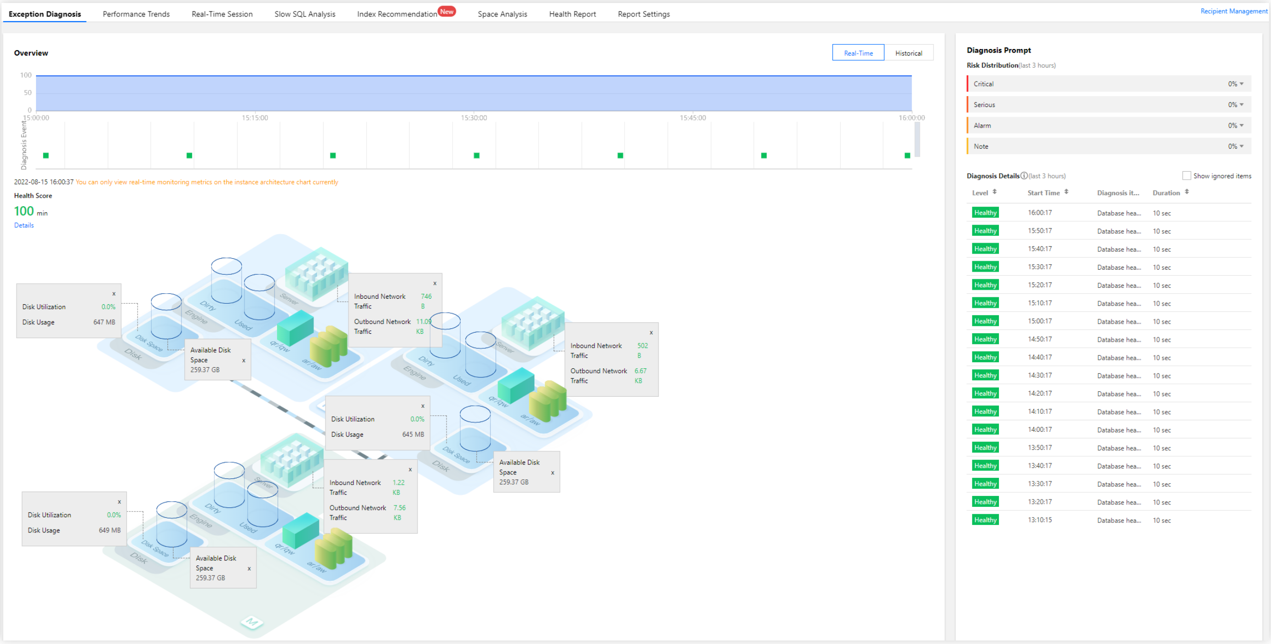
Last updated：2022-08-15 16:23:27

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.



2022-08-15 16:00:37 You can only view real-time monitoring metrics on the instance architecture chart currently

Health Score

100 min

Details

Disk Utilization 0.0%
Disk Usage 647 MB

Available Disk Space 259.37 GB

Inbound Network Traffic 746 B
Outbound Network Traffic 11.09 KB

Inbound Network Traffic 502 B
Outbound Network Traffic 6.67 KB

Disk Utilization 0.0%
Disk Usage 645 MB

Available Disk Space 259.37 GB

Inbound Network Traffic 1.22 KB
Outbound Network Traffic 7.56 KB

Disk Utilization 0.0%
Disk Usage 649 MB

Available Disk Space 259.37 GB

Diagnosis Prompt

Risk Distribution(last 3 hours)

Critical	0%
Serious	0%
Alarm	0%
Note	0%

Diagnosis Details(last 3 hours) □ Show ignored items

Level	Start Time	Diagnosis it...	Duration
Healthy	16:00:17	Database hea...	10 sec
Healthy	15:50:17	Database hea...	10 sec
Healthy	15:40:17	Database hea...	10 sec
Healthy	15:30:17	Database hea...	10 sec
Healthy	15:20:17	Database hea...	10 sec
Healthy	15:10:17	Database hea...	10 sec
Healthy	15:00:17	Database hea...	10 sec
Healthy	14:50:17	Database hea...	10 sec
Healthy	14:40:17	Database hea...	10 sec
Healthy	14:30:17	Database hea...	10 sec
Healthy	14:20:17	Database hea...	10 sec
Healthy	14:10:17	Database hea...	10 sec
Healthy	14:00:17	Database hea...	10 sec
Healthy	13:50:17	Database hea...	10 sec
Healthy	13:40:17	Database hea...	10 sec
Healthy	13:30:17	Database hea...	10 sec
Healthy	13:20:17	Database hea...	10 sec
Healthy	13:10:15	Database hea...	10 sec

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**.

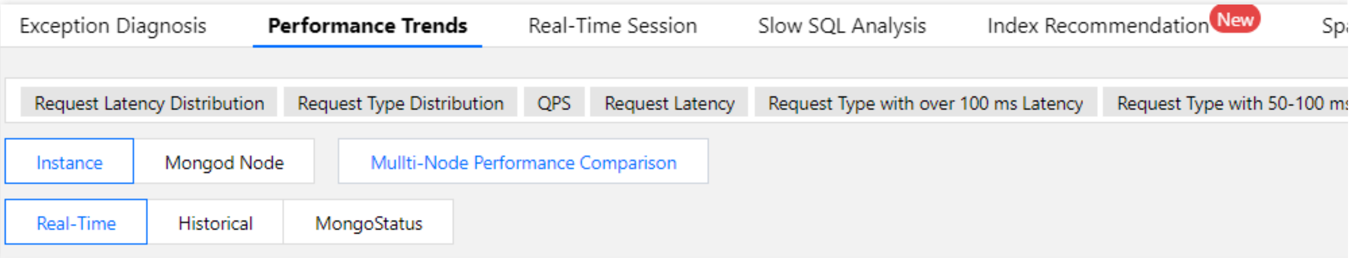
MongoDB Primary-Secondary Replication: **Secondary Node Replication Delay** and **Oplog Retention Period**.

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.



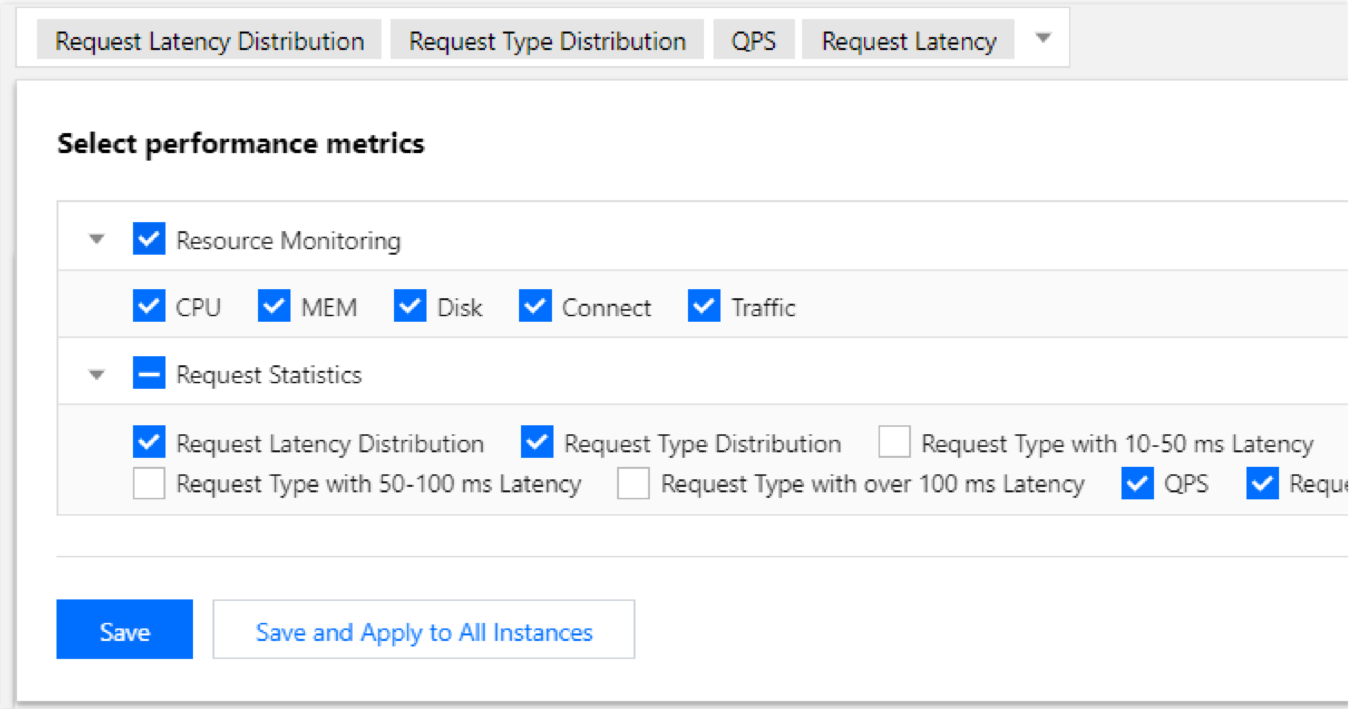
Instance dimension: It displays the monitoring view of instances.

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



Request Latency Distribution | Request Type Distribution | QPS | Request Latency

Select performance metrics

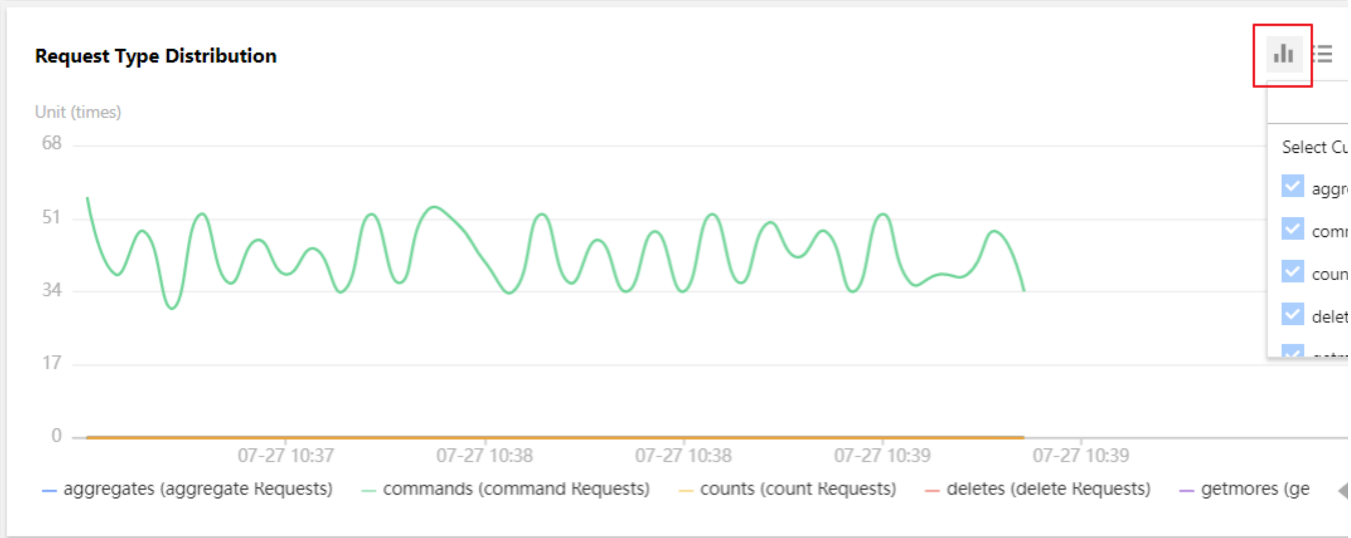
☑ Resource Monitoring

☑ CPU ☑ MEM ☑ Disk ☑ Connect ☑ Traffic

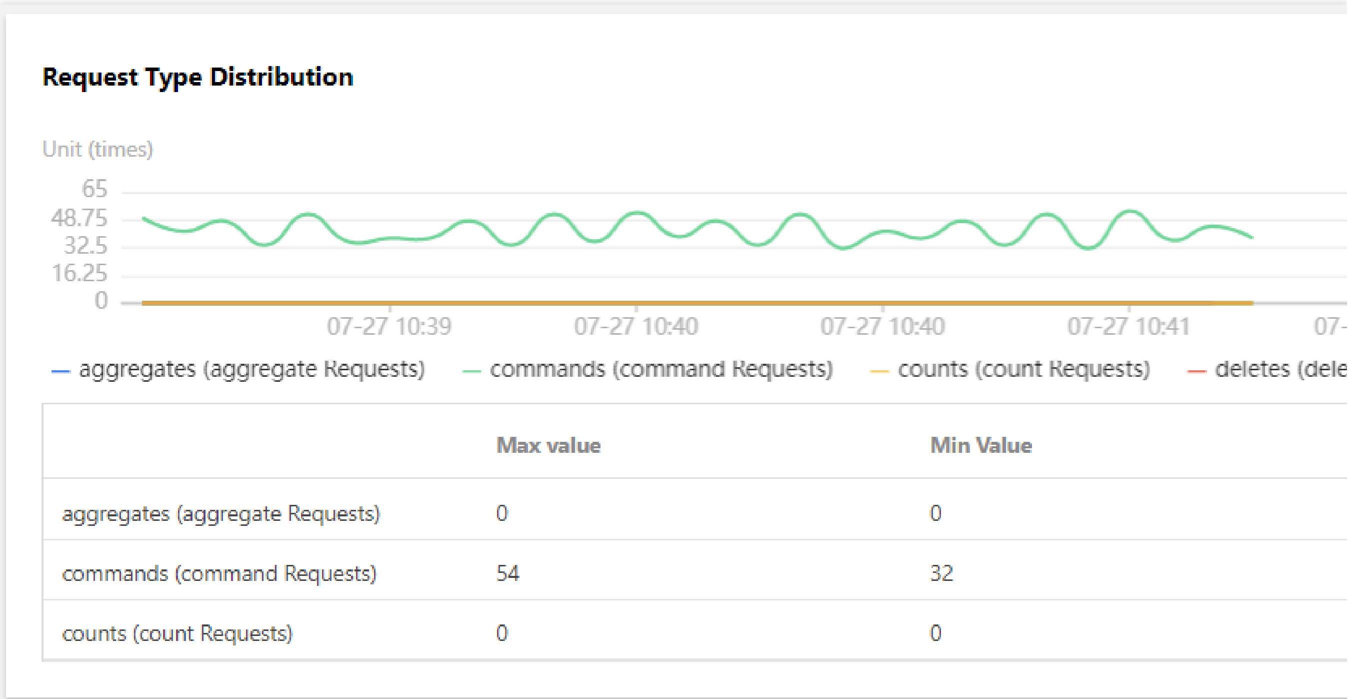
☐ Request Statistics

☑ 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 ☑ QPS ☑ Reque

Filter one single metric



Switch between chart views

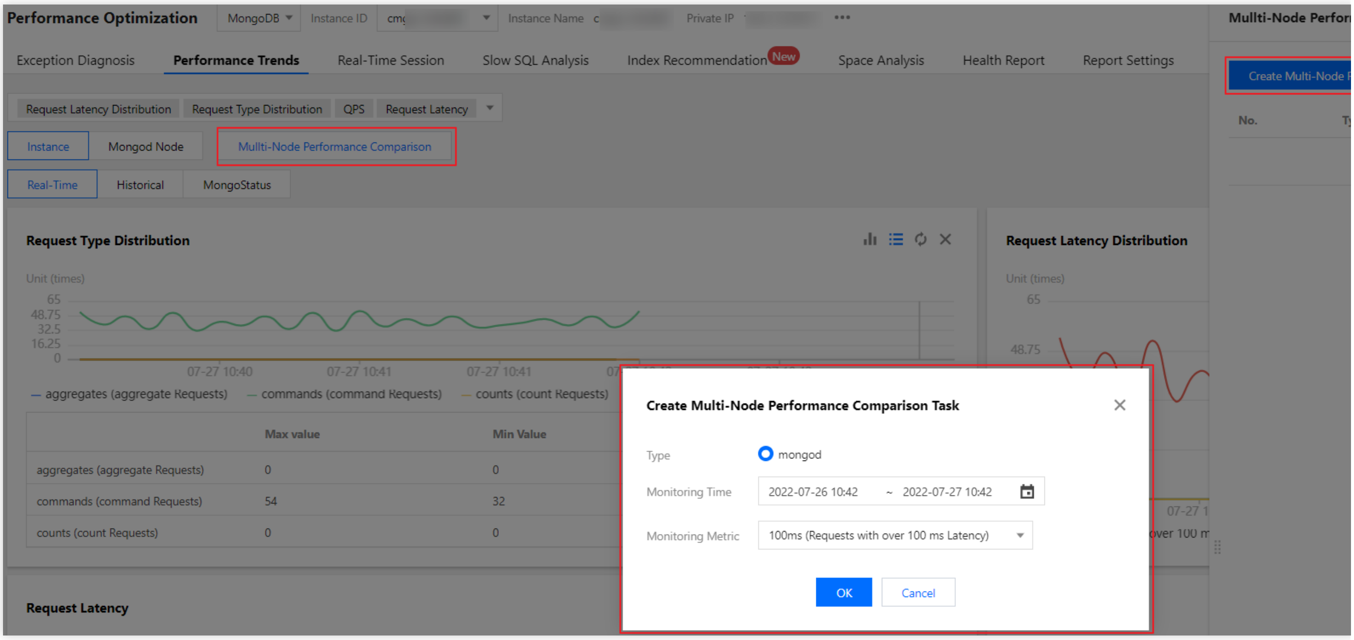

Request Type Distribution
Unit (times)
65
48.75
32.5
16.25
0
07-27 10:39
07-27 10:40
07-27 10:40
07-27 10:41
07-
— aggregates (aggregate Requests)
— commands (command Requests)
— counts (count Requests)
— deletes (dele

	Max value	Min Value
aggregates (aggregate Requests)	0	0
commands (command Requests)	54	32
counts (count Requests)	0	0

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.



	Max value	Min Value
aggregates (aggregate Requests)	0	0
commands (command Requests)	54	32
counts (count Requests)	0	0

Create Multi-Node Performance Comparison Task

Type: ● mongod

Monitoring Time: 2022-07-26 10:42 ~ 2022-07-27 10:42

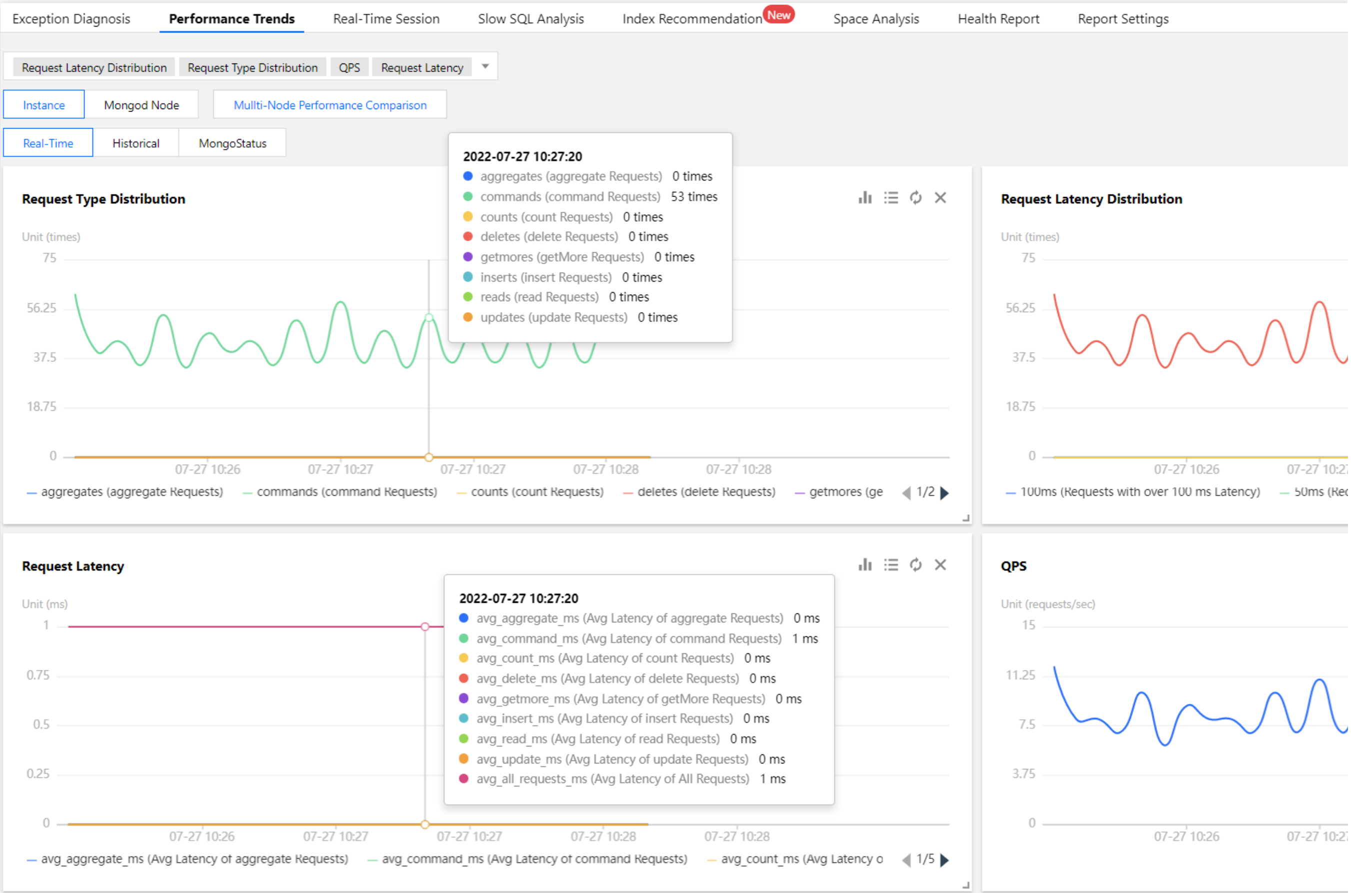
Monitoring Metric: 100ms (Requests with over 100 ms Latency)

OK Cancel

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**.



2022-07-27 10:27:20
- aggregates (aggregate Requests) 0 times
- commands (command Requests) 53 times
- counts (count Requests) 0 times
- deletes (delete Requests) 0 times
- getmores (getMore Requests) 0 times
- inserts (insert Requests) 0 times
- reads (read Requests) 0 times
- updates (update Requests) 0 times

2022-07-27 10:27:20
- avg_aggregate_ms (Avg Latency of aggregate Requests) 0 ms
- avg_command_ms (Avg Latency of command Requests) 1 ms
- avg_count_ms (Avg Latency of count Requests) 0 ms
- avg_delete_ms (Avg Latency of delete Requests) 0 ms
- avg_getmore_ms (Avg Latency of getMore Requests) 0 ms
- avg_insert_ms (Avg Latency of insert Requests) 0 ms
- avg_read_ms (Avg Latency of read Requests) 0 ms
- avg_update_ms (Avg Latency of update Requests) 0 ms
- avg_all_requests_ms (Avg Latency of All Requests) 1 ms

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.

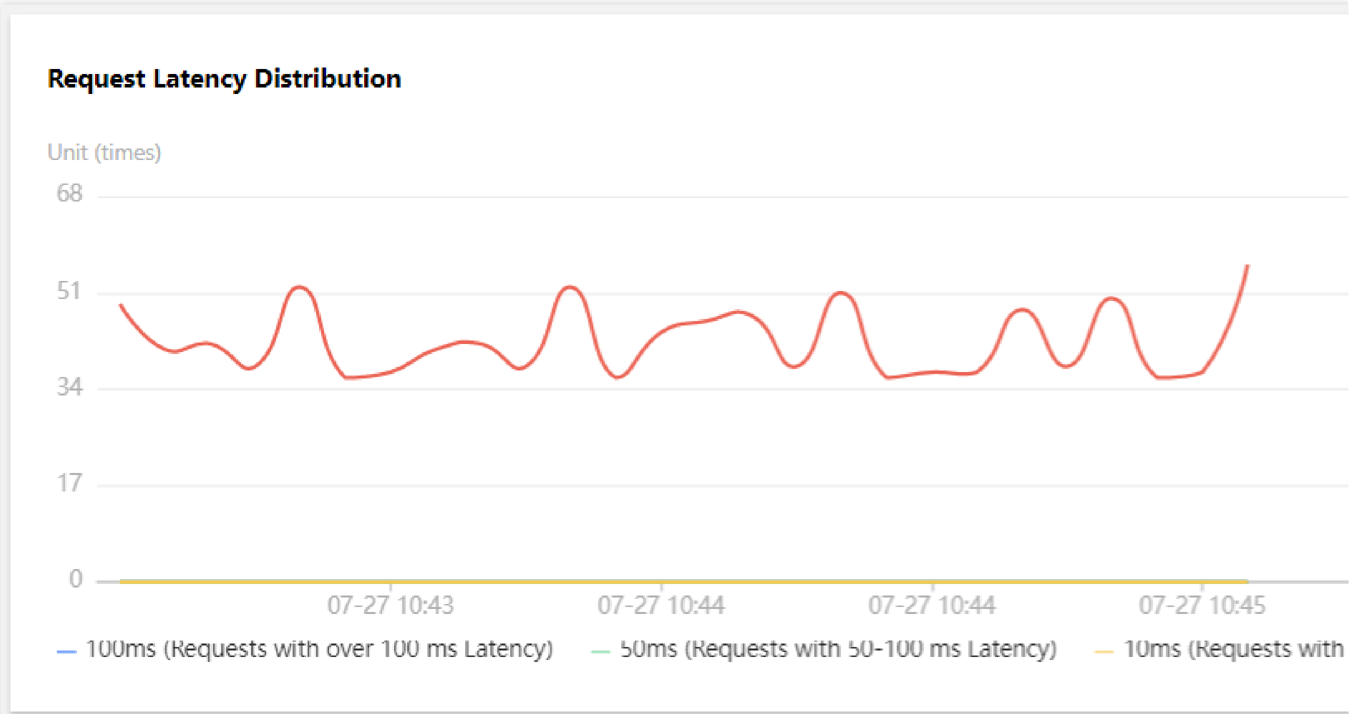
Zooming in a monitoring view: Drag the icon in the bottom-right corner of a monitoring view to zoom it in for fine-grained display of the trend of one single performance metric.

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

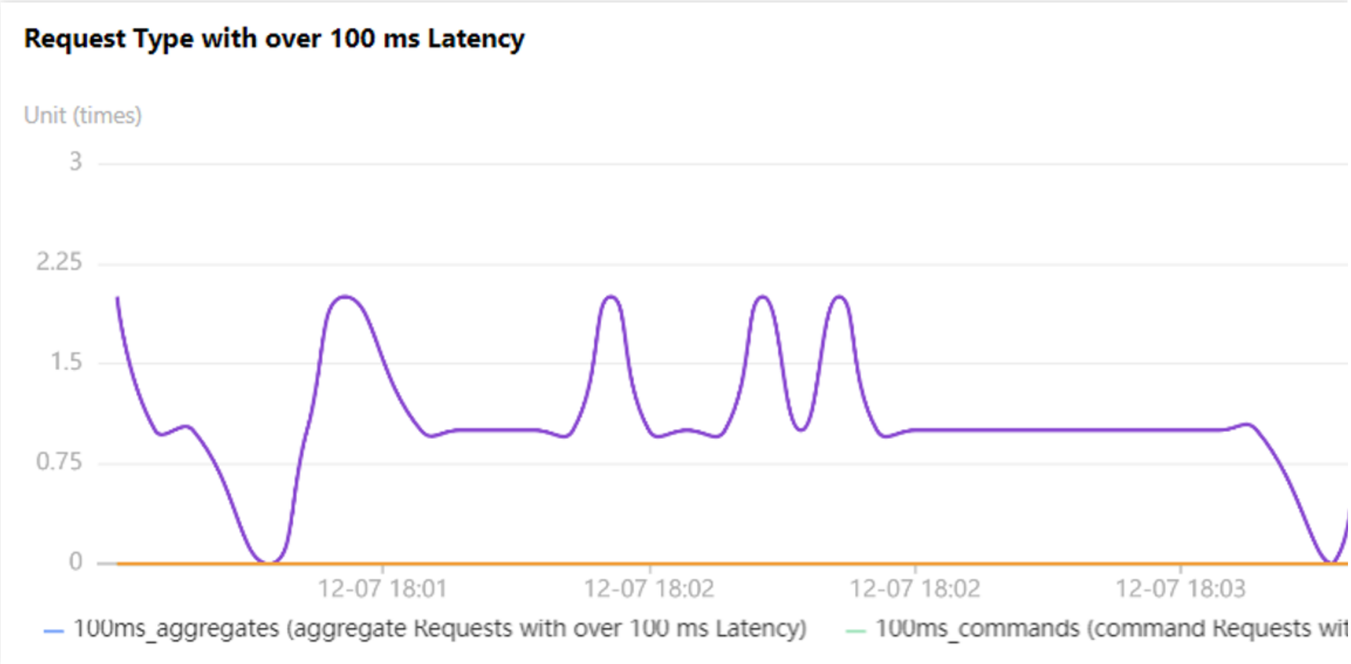
7. View the data of latency analysis.

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

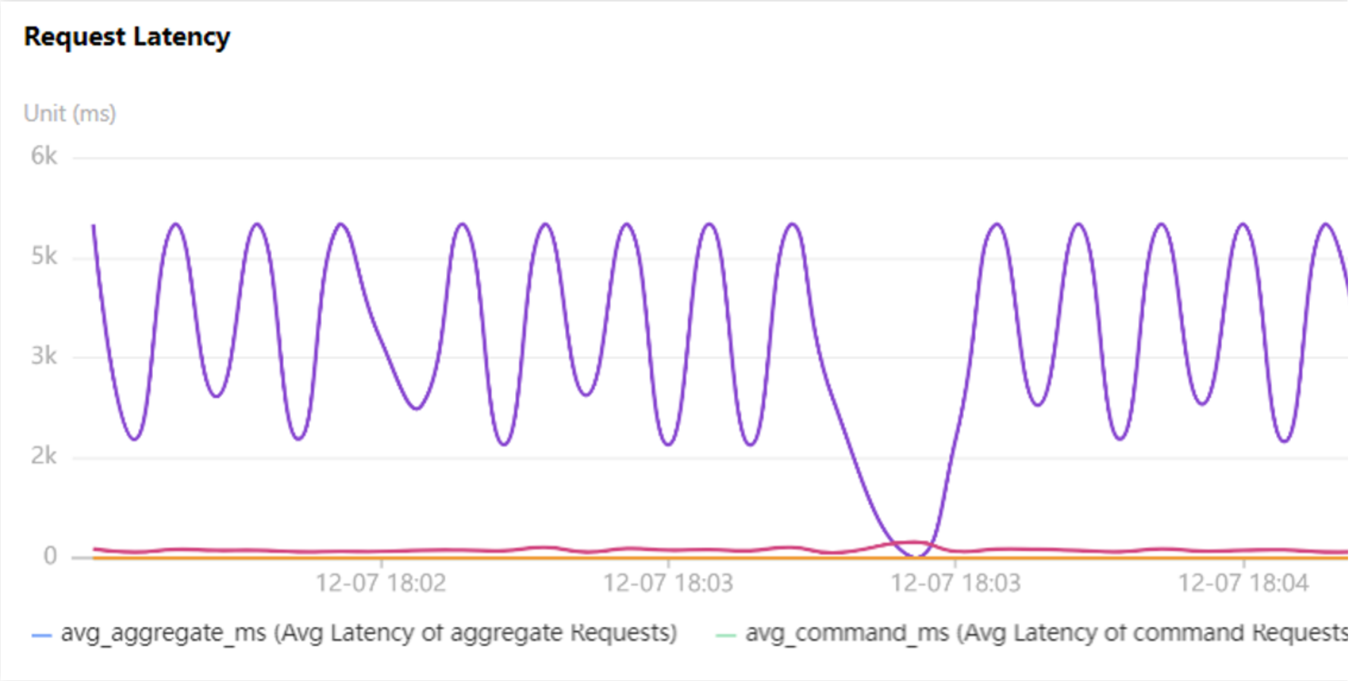
Sample request latency distribution:



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
avg_aggregate_ms (Avg Latency of aggregate Requests)
avg_command_ms (Avg Latency of command Requests

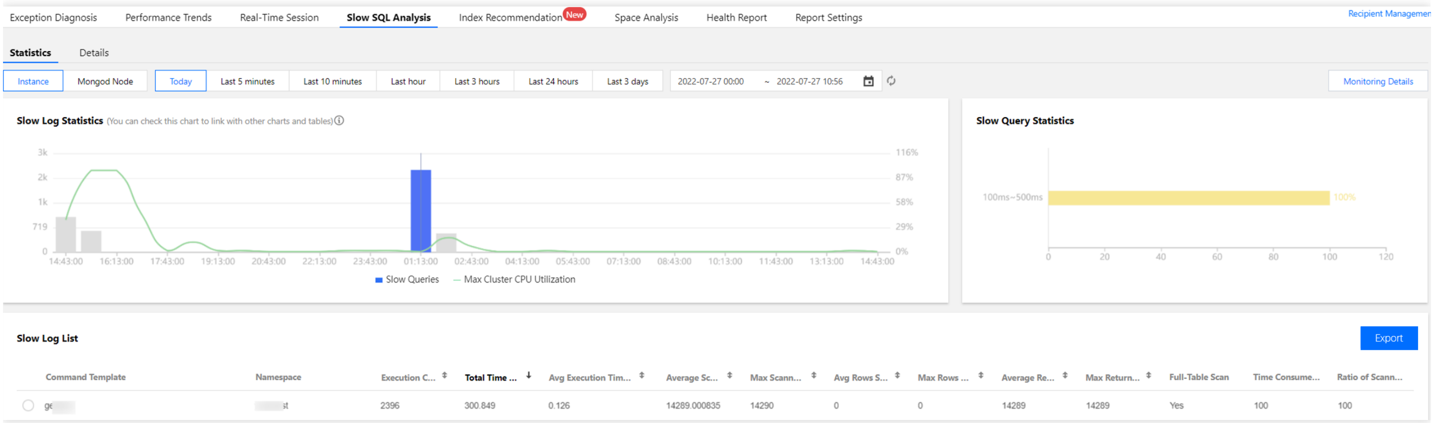
Slow SQL Analysis

Last updated：2022-08-13 16:25:49

Feature description

The slow log analysis feature calculates, samples, and aggregates records and execution information (source information, number of executions, execution time, result set, scan set, etc.) of slow logs in the instance.

Overview



Slow Log List

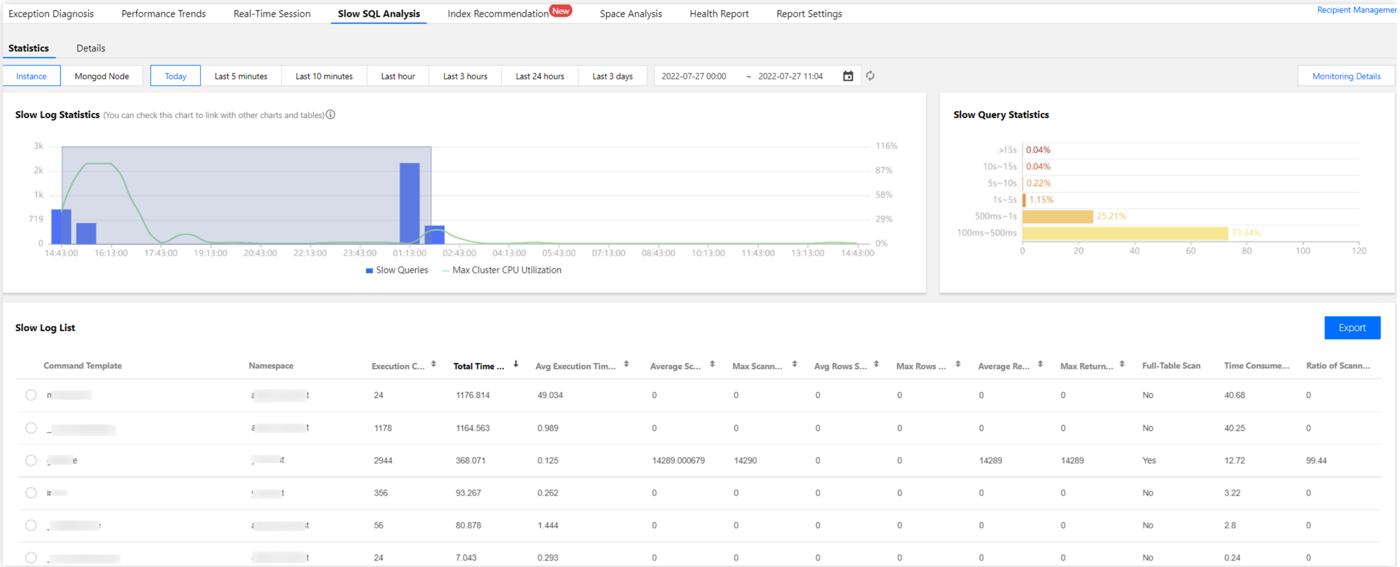
	Command Template	Namespace	Execution C...	Total Time ...	Avg Execution Tim...	Average Sc...	Max Scann...	Avg Rows S...	Max Rows ...	Average Re...	Max Return...	Full-Table Scan	Time Consume...	Ratio of Scann...
○	ge[redacted]	[redacted]st	2396	300.849	0.126	14289.000835	14290	0	0	14289	14289	Yes	100	100

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.
   The **Slow Log Statistics** section displays the **Slow Queries** and **Max Cluster 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.

- 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**.
- You can select **Instance** or **Mongod Node** as the statistical dimension.

2. You can click a single time range or drag to select multiple time ranges for slow queries in the **Slow Log Statistics** section, and the aggregated slow log 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 slow logs in the selected time range.



Slow Log List

	Command Template	Namespace	Execution C...	Total Time ...	Avg Execution Tim...	Average Sc...	Max Scann...	Avg Rows S...	Max Rows ...	Average Re...	Max Return...	Full-Table Scan	Time Consume...	Ratio of Scann...
○	[redacted]	[redacted]	24	1176.814	49.034	0	0	0	0	0	0	No	40.68	0
○	[redacted]	[redacted]	1178	1164.563	0.989	0	0	0	0	0	0	No	40.25	0
○	[redacted]	[redacted]	2944	368.071	0.125	14289.000679	14290	0	0	14289	14289	Yes	12.72	99.44
○	[redacted]	[redacted]	356	93.267	0.262	0	0	0	0	0	0	No	3.22	0
○	[redacted]	[redacted]	56	80.878	1.444	0	0	0	0	0	0	No	2.8	0
○	[redacted]	[redacted]	24	7.043	0.293	0	0	0	0	0	0	No	0.24	0

3. Click an aggregated slow log, and its statistics and details will be displayed on the right.

- On the **Statistics** tab, you can view the **Consumed Time Distribution**, **Time Consumed Ratio**, **Ratio of Scanned Rows**, and **Average Scanned Rows**.
- On the **Details** tab, you can view the details of the **Command Template**, including the **SQL Statement**, **Namespace**, **Execution Time**, **Scanned Indexes**, **Returned Rows**, and **Scanned Rows**. You can also filter logs by **Time Range**, **Namespace**, or **Time Consumed** to query the details of historical SQL statements.
  You can pull the **Details** tab to the left to expand it horizontally or view the enlarged statistical chart on the **Details** tab on the homepage.

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

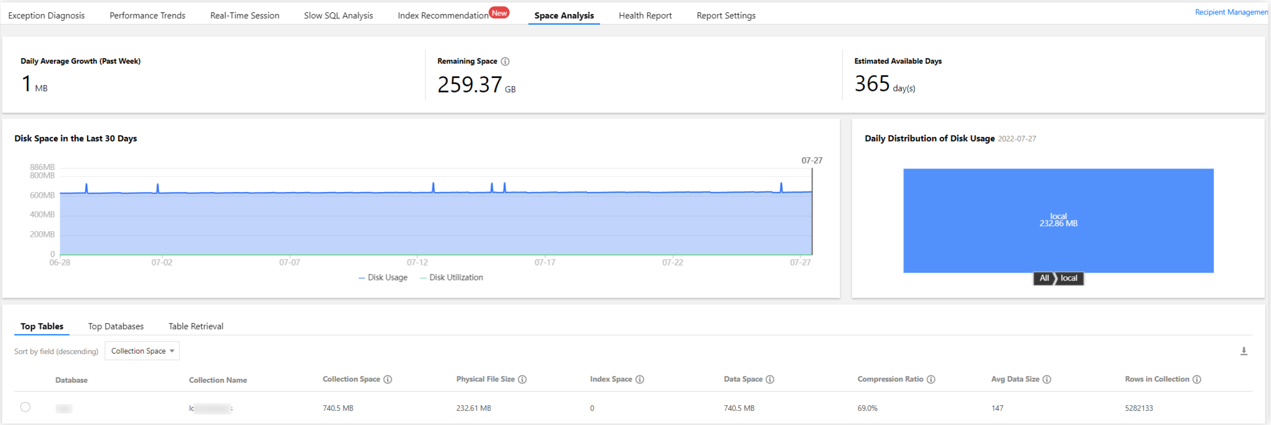
Space Analysis

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



Daily Average Growth (Past Week): 1 MB

Remaining Space: 259.37 GB

Estimated Available Days: 365 day(s)

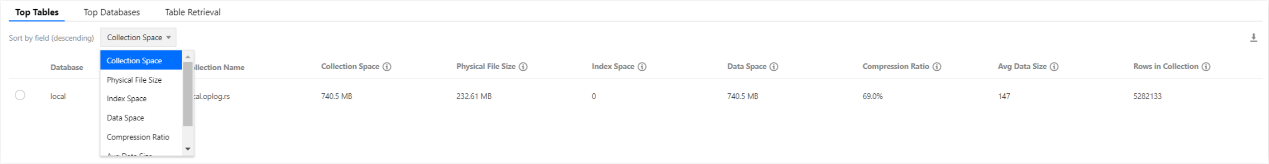
Sort by field (descending): Collection Space

	Database	Collection Name	Collection Space	Physical File Size	Index Space	Data Space	Compression Ratio	Avg Data Size	Rows in Collection
○	[redacted]	lc[redacted];	740.5 MB	232.61 MB	0	740.5 MB	69.0%	147	5282133

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.

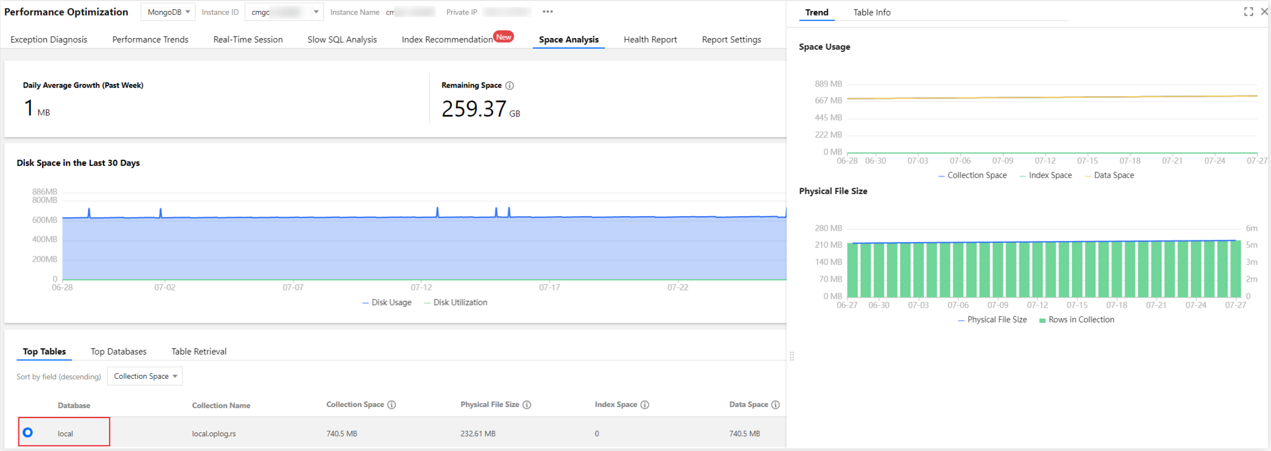


Sort by field (descending) Collection Space

	Database	llection Name	Collection Space ⓘ	Physical File Size ⓘ	Index Space ⓘ	Data Space ⓘ	Compression Ratio ⓘ	Avg Data Size ⓘ	Rows in Collection ⓘ
○	local	cal.oplog.rs	740.5 MB	232.61 MB	0	740.5 MB	69.0%	147	5282133

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**.



Instance ID cmgc[redacted] Instance Name cm[redacted] Private IP [redacted]

Daily Average Growth (Past Week)
1 MB

Remaining Space ⓘ
259.37 GB

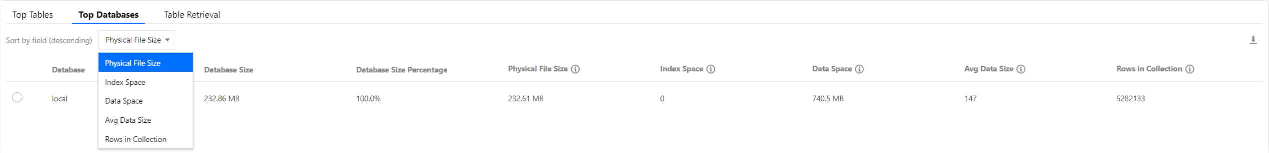
Sort by field (descending) Collection Space

	Database	Collection Name	Collection Space ⓘ	Physical File Size ⓘ	Index Space ⓘ	Data Space ⓘ
●	local	local.oplog.rs	740.5 MB	232.61 MB	0	740.5 MB

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.

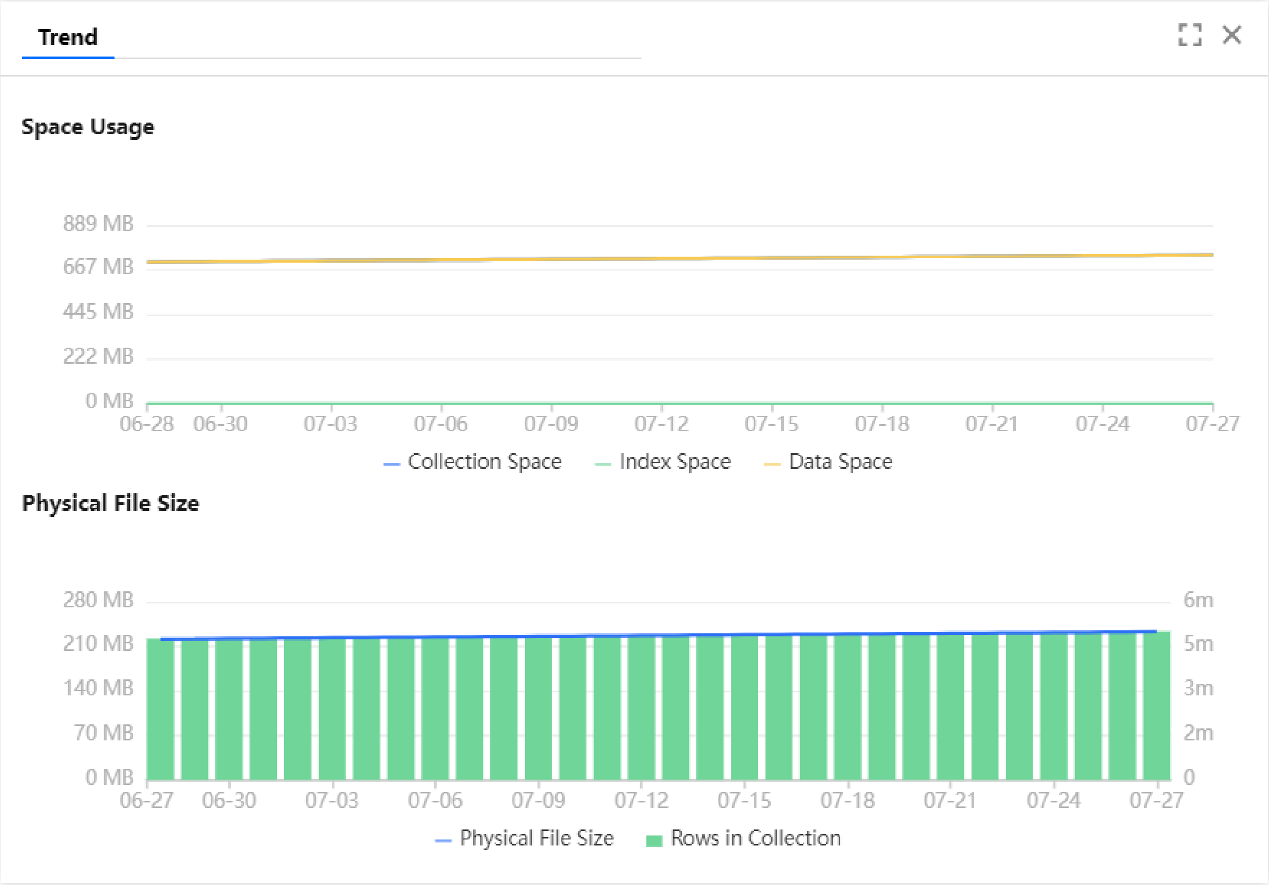
The **Top Databases** section shows the details of the databases that have relatively high space usage. The database list in the section contains columns such as the **Physical File Size**, **Index Space**, **Data Space**, **Avg Data Size**, and **Rows in Collection**. The databases can be sorted by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



Sort by field (descending) Physical File Size

	Database	Database Size	Database Size Percentage	Physical File Size ⓘ	Index Space ⓘ	Data Space ⓘ	Avg Data Size ⓘ	Rows in Collection ⓘ
○	local	232.86 MB	100.0%	232.61 MB	0	740.5 MB	147	5282133

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

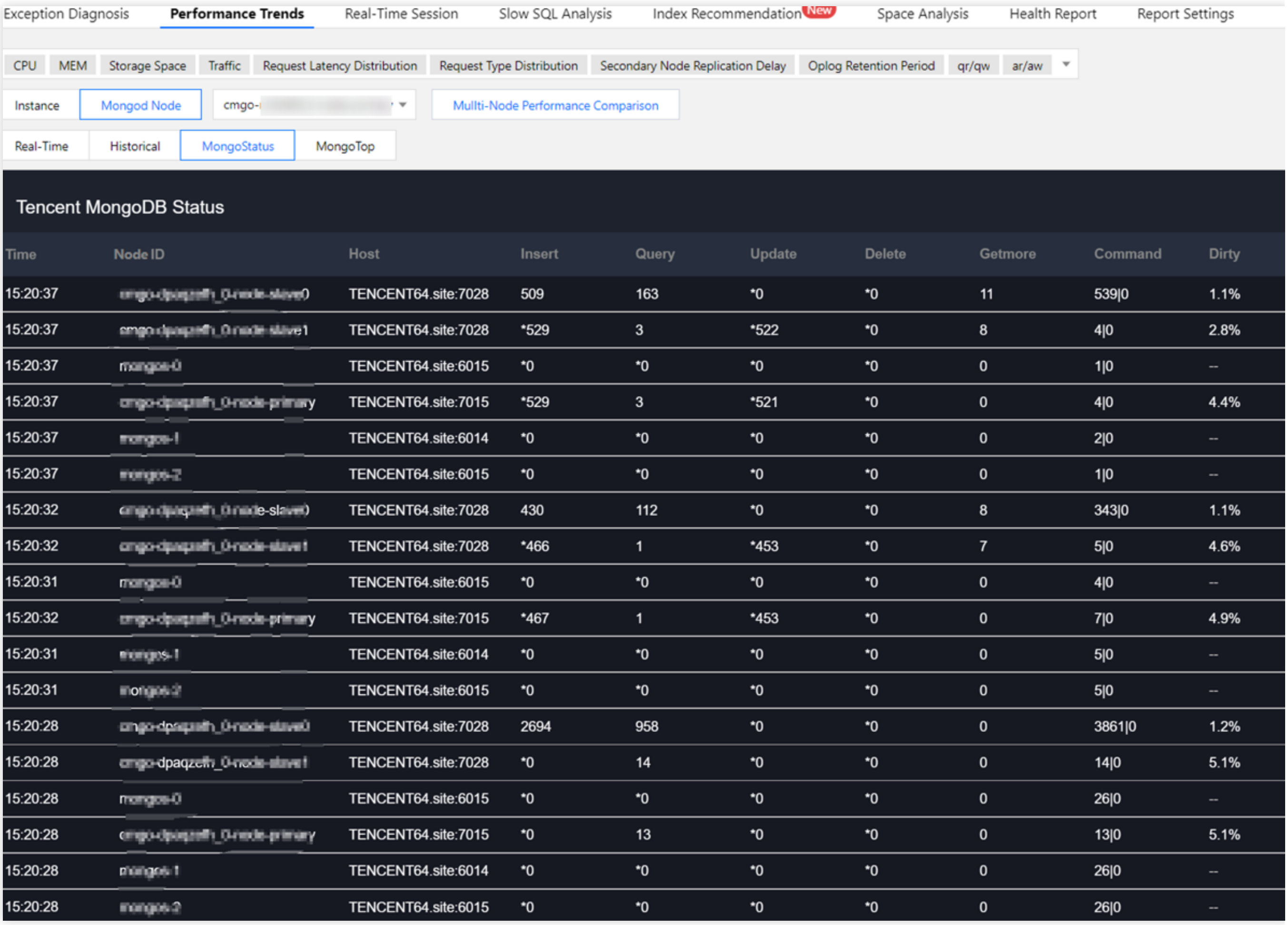
Overview

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.



Tencent MongoDB Status

Time	Node ID	Host	Insert	Query	Update	Delete	Getmore	Command	Dirty
15:20:37	[redacted]	TENCENT64.site:7028	509	163	*0	*0	11	539|0	1.1%
15:20:37	[redacted]	TENCENT64.site:7028	*529	3	*522	*0	8	4|0	2.8%
15:20:37	[redacted]	TENCENT64.site:6015	*0	*0	*0	*0	0	1|0	--
15:20:37	[redacted]	TENCENT64.site:7015	*529	3	*521	*0	0	4|0	4.4%
15:20:37	[redacted]	TENCENT64.site:6014	*0	*0	*0	*0	0	2|0	--
15:20:37	[redacted]	TENCENT64.site:6015	*0	*0	*0	*0	0	1|0	--
15:20:32	[redacted]	TENCENT64.site:7028	430	112	*0	*0	8	343|0	1.1%
15:20:32	[redacted]	TENCENT64.site:7028	*466	1	*453	*0	7	5|0	4.6%
15:20:31	[redacted]	TENCENT64.site:6015	*0	*0	*0	*0	0	4|0	--
15:20:32	[redacted]	TENCENT64.site:7015	*467	1	*453	*0	0	7|0	4.9%
15:20:31	[redacted]	TENCENT64.site:6014	*0	*0	*0	*0	0	5|0	--
15:20:31	[redacted]	TENCENT64.site:6015	*0	*0	*0	*0	0	5|0	--
15:20:28	[redacted]	TENCENT64.site:7028	2694	958	*0	*0	0	3861|0	1.2%
15:20:28	[redacted]	TENCENT64.site:7028	*0	14	*0	*0	0	14|0	5.1%
15:20:28	[redacted]	TENCENT64.site:6015	*0	*0	*0	*0	0	26|0	--
15:20:28	[redacted]	TENCENT64.site:7015	*0	13	*0	*0	0	13|0	5.1%
15:20:28	[redacted]	TENCENT64.site:6014	*0	*0	*0	*0	0	26|0	--
15:20:28	[redacted]	TENCENT64.site:6015	*0	*0	*0	*0	0	26|0	--

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	-

MongoTop

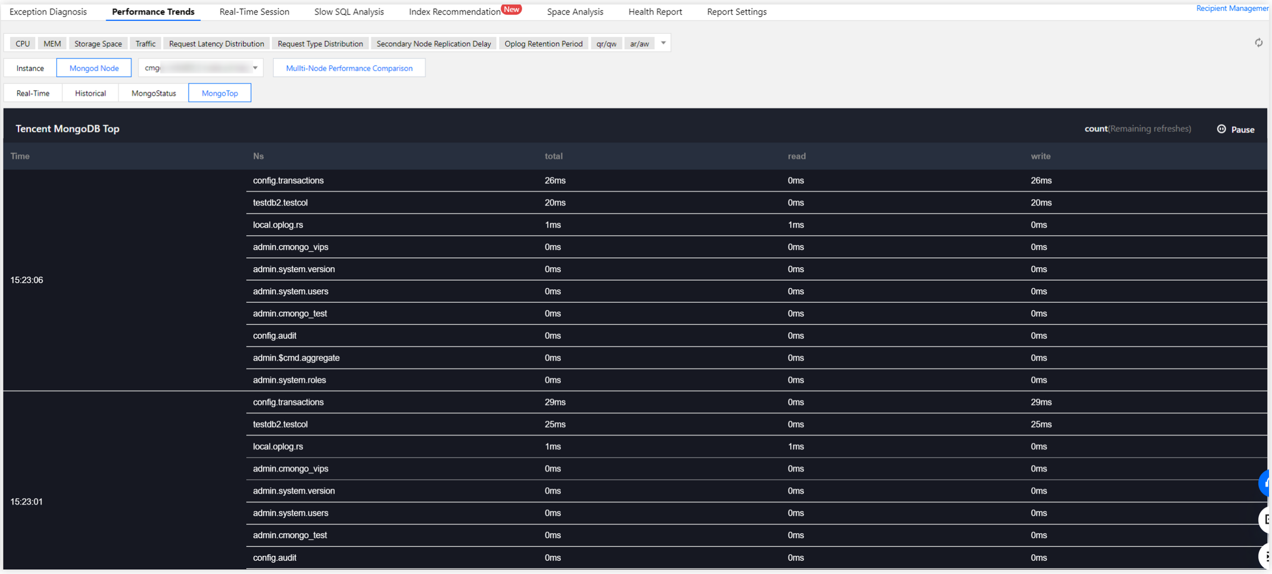
Last updated：2022-08-13 16:25:49

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.

Real-Time Session

Last updated：2022-08-13 16:25:50

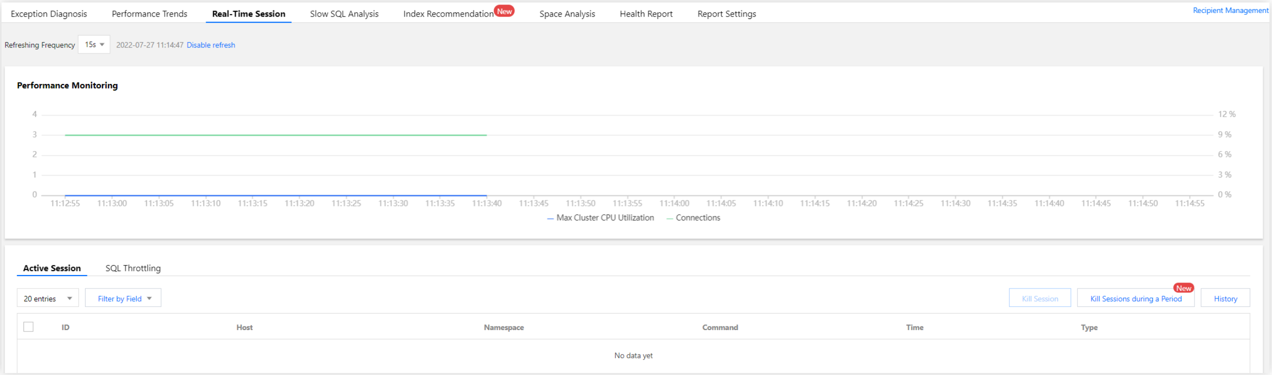
Feature description

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.



Refreshing Frequency 15s 2022-07-27 11:14:47

20 entries

□	ID	Host	Namespace	Command	Time	Type
			No data yet			

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**.

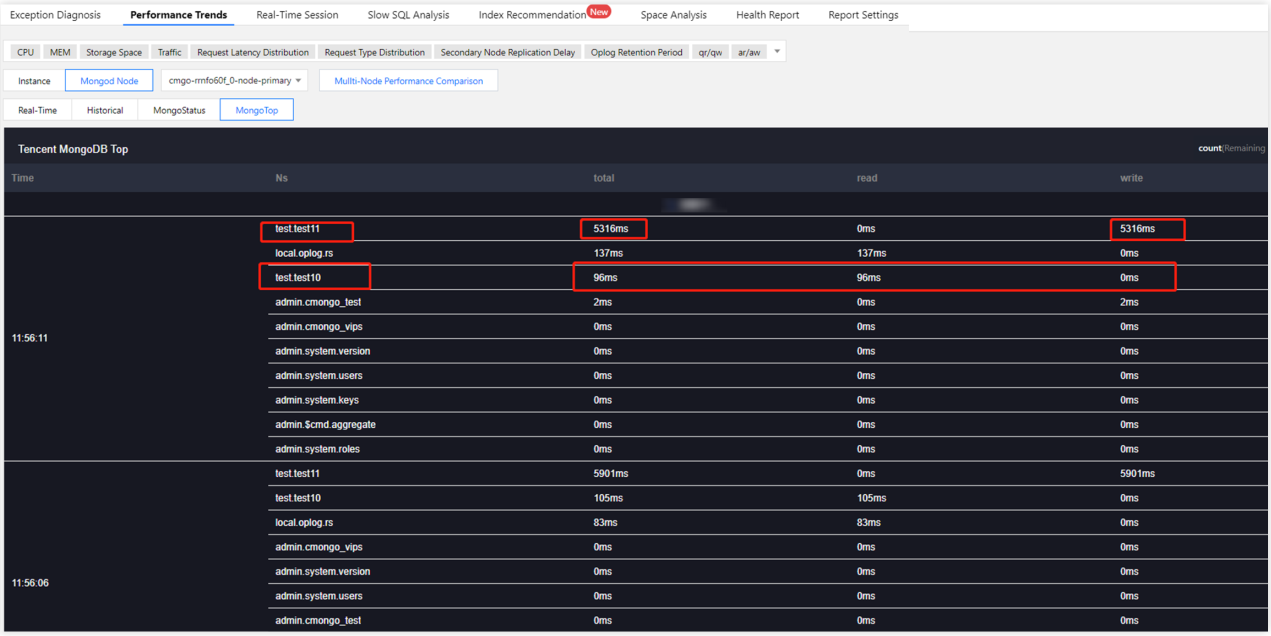
After a SQL throttling task is enabled, its status will change to **Terminated** once its remaining time decreases to zero.

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

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.

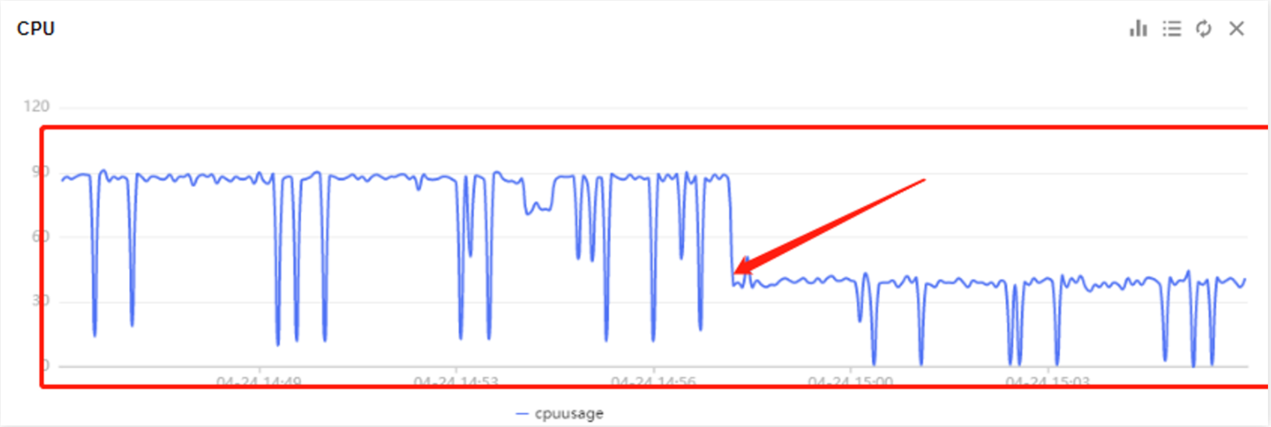


Tencent MongoDB Top

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).

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.

Best Practices

Last updated：2022-08-13 16:25:50

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.