

Database Audit Operation Guide Product Documentation



Copyright Notice

©2013-2024 Tencent Cloud. All rights reserved.

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

Trademark Notice



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

Service Statement

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

Contents

Operation Guide

Viewing Audit Log

Modifying Log Retention Period

Authorizing Sub-account to Use Database Audit

Operation Guide

Viewing Audit Log

Last updated : 2024-09-06 12:13:36

Viewing Log

1. Log in to the [TencentDB for MySQL](#), [TDSQL-C for MySQL](#), or [TencentDB for MongoDB](#) console, select **Database Audit** on the left sidebar, select a region at the top, and click the **Audit Log** tab.
2. In the audit instance section on the **Audit Log** tab, select a database instance with audit enabled to view its SQL audit logs. Or, on the **Audit Instance** tab, click an instance ID to enter the **Audit Log** tab and view audit logs.

Note:

The audit log display time is down to milliseconds, facilitating more precise sorting and problem analysis of SQL commands.

Tool list

The screenshot shows a search interface for audit logs. It includes a dropdown menu for 'Audit Instance' with the text 'Select an instance', a date range selector with 'Invalid date ~ Invalid date', and a search input field for 'SQL Details' with the text 'Includ', 'and', 'Segme', and a placeholder 'Enter the SQL command details and separate them by line break'.

In the **time box**, select a time period to view audit results in the selected time period.

Note:

You can select any time period with data for search. Up to the first 60,000 eligible records can be displayed.

You can search by key tag to view related audit results. Common key tags include SQL command, client IP, database name, database account, SQL type, policy name, execution time, affected rows, and returned rows.

When entering multiple key tags for search, you can separate them by pressing "Enter".

You can filter IP addresses using the wildcard ". For example, if you enter "client IP: 9.223.23.2", IP addresses that start with "9.223.23.2" will be searched.

Combo search is supported. Selecting the key tag "SQL Command" allows you to separate filters using commas or spaces, and the logic relationship between them is AND. You may also use vertical bars ("|") to separate filters, and the logic relationship between them is OR.

Note:

When filtering by SQL command, the symbol `*` does not represent a fuzzy match. All SQL command searches are fuzzy searches.

Log list

In the **SQL Type** drop-down list, you can select multiple SQL types for filtering.

The **Returned Rows** field represents the specific number of rows returned by executing the SQL command, which is mainly used to determine the impact of `SELECT` commands.

SQL Audit Fields

TencentDB for MySQL and TDSQL-C for MySQL

The following fields are supported in TencentDB for MySQL and TDSQL-C for MySQL audit logs. You can click the following icon on the **Audit Log** tab in the [TencentDB for MySQL](#) or [TDSQL-C for MySQL](#) console to get and view the complete SQL audit logs.

No.	Field Name	Description	Remarks
1	host	Client IP	-
2	dbname	Database name	-

3	user	Username	-
4	sql	SQL statement	-
5	sqlType	SQL statement type	-
6	errCode	Error code	0 indicates success
7	affectRows	Number of affected rows	-
8	checkRows	Number of scanned rows	-
9	sentRows	Number of returned rows	-
10	threadId	Thread ID	-
11	ruleNum	Audit rule ID	-
12	policyName	Audit policy name	-
13	instanceName	Instance name	-
14	timestamp	Start time (s)	-
15	nsTime	Start Time (ns), which forms the start time accurate down to the nanosecond together with <code>timestamp</code>	Example: timestamp.nsTime 1577953020.887984015
16	execTime	Execution time (ms)	-
17	cpuTime	CPU time (μ s)	-
18	lockWaitTime	Lock wait time (μ s)	-
19	ioWaitTime	IO wait time (μ s)	-
20	trxLivingTime	Transaction execution time (μ s)	-

Modifying Log Retention Period

Last updated : 2024-09-06 12:14:49

This document describes how to modify the log retention period after the database audit service is activated.

Directions

1. Log in to the [TencentDB for MySQL](#), [TDSQL-C for MySQL](#), or [TencentDB for MongoDB](#) console, select **Database Audit** on the left sidebar, select a region at the top, and click the **Audit Log** tab.
2. In the top-right corner of the **Audit Log** tab, click **Configure**.
3. In the pop-up window, modify the log retention period and click **Submit**.

Authorizing Sub-account to Use Database Audit

Last updated : 2024-09-06 12:18:10

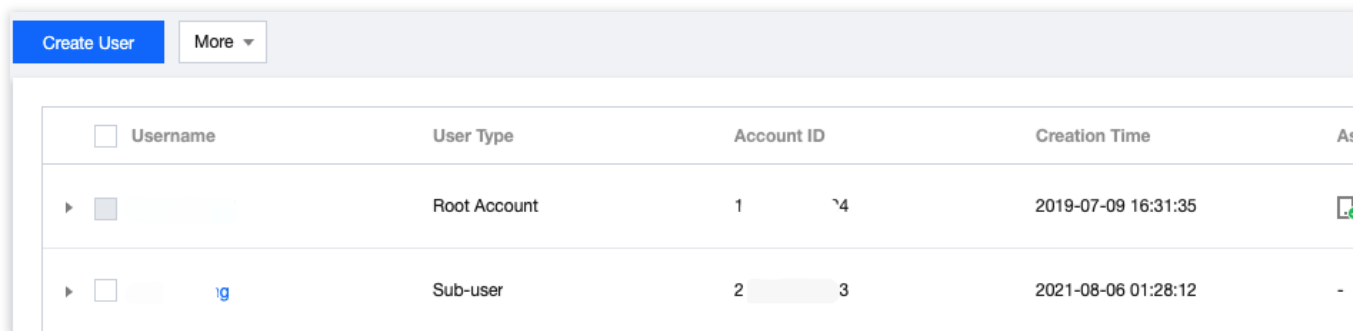
By default, sub-accounts have no permission to use TencentDB for MySQL Database Audit. Therefore, you need to create policies to allow sub-accounts to use it.

If you don't need to manage sub-accounts' access to resources related to TencentDB for MySQL Database Audit, you can ignore this document.

[Cloud Access Management \(CAM\)](#) is a web-based Tencent Cloud service that helps you securely manage and control access to your Tencent Cloud resources. By using CAM, you can create, manage, and terminate users and user groups. You can manage identities and policies to allow specific users to access your Tencent Cloud resources. 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, please see [Syntax Logic](#).

Authorizing Sub-account

1. Log in to the [CAM console](#) as a root account, select the target sub-user in the user list, and click **Authorize**.



<input type="checkbox"/>	Username	User Type	Account ID	Creation Time	As
<input type="checkbox"/>		Root Account	1 4	2019-07-09 16:31:35	
<input checked="" type="checkbox"/>	g	Sub-user	2 3	2021-08-06 01:28:12	-

2. In the pop-up window, select the **QcloudCDBFullAccess** or **QcloudCDBInnerReadOnlyAccess** preset policy and click **OK** to complete the authorization.

Note:

MySQL Database Audit is a module in TencentDB for MySQL, so the above two preset policies of TencentDB for MySQL already cover the permission policies required by it. If the sub-user only needs the permission to use this module, please see [Custom MySQL Database Audit Policy](#).

Associate Policy

Select Policies (619 Total) 0 selected

Support search by policy name/description/remarks Q

Policy Name	Policy type ▾
<input checked="" type="checkbox"/> AdministratorAccess This policy allows you to manage all user...	Preset Policy
<input type="checkbox"/> ReadOnlyAccess This policy authorizes you with the read-...	Preset Policy
<input type="checkbox"/> QCloudResourceFullAccess This policy allows you to manage all clou...	Preset Policy
<input type="checkbox"/> QCloudFinanceFullAccess This policy allows you to manage all fina...	Preset Policy
<input type="checkbox"/> QcloudAdvisorFullAccess This policy allows you to manage all adv...	Preset Policy

Support for holding shift key down for multiple selection

↔

Policy Name

Confirm Cancel

Policy Syntax

The CAM policy for MySQL Database Audit is described as follows:

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

version is required. Currently, only the value "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`, and `resource`. One policy has only one `statement`.

effect is required. It describes the result of a statement. The result can be "allow" or an "explicit deny".

action is required. It describes the allowed or denied action (operation). An operation can be an API (prefixed with "name") or a feature set (a set of specific APIs prefixed with "permid").

resource is required. It describes the details of authorization.

API Operation

In a CAM policy statement, you can specify any API operation from any service that supports CAM. APIs prefixed with `name/cdb:` should be used for Database Audit. To specify multiple operations in a single statement, separate them with commas as shown below:

```
"action":["name/cdb:action1","name/cdb:action2"]
```

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

```
"action":["name/cdb:Describe*"]
```

Resource Path

Resource paths are generally in the following format:

```
qcs::service_type::account:resource
```

`service_type`: describes the product abbreviation, such as `cdb` here.

`account`: describes the root account of the resource owner, such as `uin/326xxx46`.

`resource`: describes the detailed resource information of the specific service. Each TencentDB for MySQL instance (`instanceId`) is a resource.

Example:

```
"resource":["qcs::cdb::uin/326xxx46:instanceId/cdb-kf291vh3"]
```

Here, `cdb-kf291vh3` is the ID of the TencentDB for MySQL instance resource, i.e., the `resource` in the CAM policy statement.

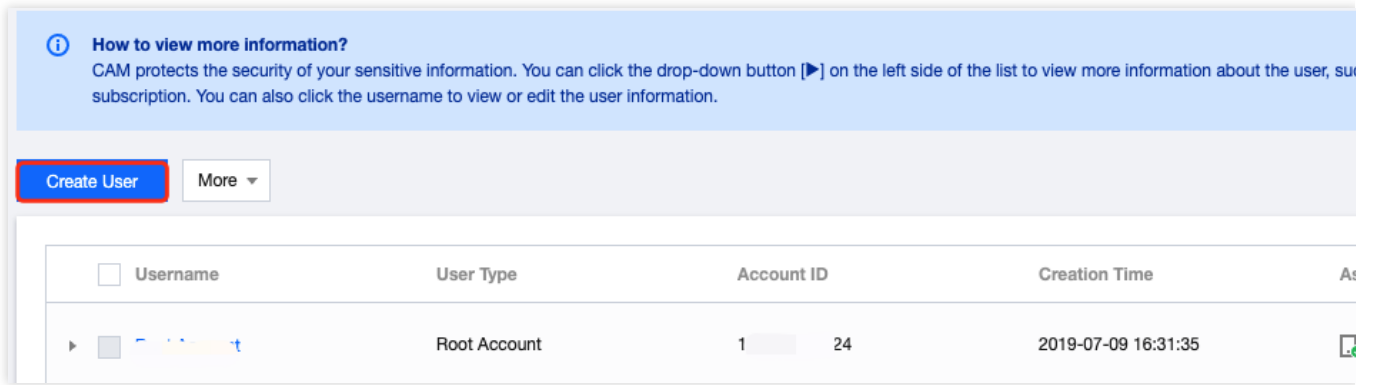
Example

The following example only shows the usage of CAM.

```
{
  "version": "2.0",
  "statement": [
    {
      "effect": "allow",
      "action": [
        "name/cdb: DescribeAuditRules"
      ],
      "resource": [
        "*"
      ]
    },
    {
      "effect": "allow",
      "action": [
        "name/cdb: CreateAuditPolicy"
      ],
      "resource": [
        "*"
      ]
    },
    {
      "effect": "allow",
      "action": [
        "name/cdb: DescribeAuditLogFiles"
      ],
      "resource": [
        "qcs::cdb::uin/326xxx46:instanceId/cdb-kf291vh3"
      ]
    }
  ]
}
```

Custom MySQL Database Audit Policy

1. Log in to the [CAM console](#) as the root account and click **Create Custom Policy** in the policy list.



How to view more information?
CAM protects the security of your sensitive information. You can click the drop-down button [▶] on the left side of the list to view more information about the user, subscription. You can also click the username to view or edit the user information.

Create User More ▾

<input type="checkbox"/> Username	User Type	Account ID	Creation Time	As
▶ <input type="checkbox"/> root	Root Account	1 24	2019-07-09 16:31:35	<input type="checkbox"/>

2. In the pop-up window, select **Create by Policy Generator**.

3. On the **Select Service and Action** page, select configuration items, click **Add Statement**, and click **Next**.

Service: select **TencentDB for MySQL**.

Action: select all APIs of MySQL Database Audit.

Resource: for more information, please see [Resource Description Method](#). You can enter * to indicate that the audit logs of all TencentDB for MySQL instances can be manipulated.

1 Edit Policy > 2 Associate Users/User Groups

Visual Policy Generator JSON

Cloud Database(165 actions)

Effect Allow Deny

Service Cloud Database (cdb)

Action [Collapse](#)

Select actions

All actions (cdb:*) [Fold](#)

Select Action

Filter Actions

<input checked="" type="checkbox"/>	Action Name	Description
<input checked="" type="checkbox"/>	AddInstanceInDeployGroup	add instance in deploy group
<input checked="" type="checkbox"/>	AddTimeWindow	add a maintenance time window f...
<input checked="" type="checkbox"/>	AssociateSecurityGroups	bind security groups to instances i...
<input checked="" type="checkbox"/>	BalanceRoGroupLoad	load balancing in ro group
<input checked="" type="checkbox"/>	CancelBatchOperation	stop batch import task
<input checked="" type="checkbox"/>	CancelDBInstanceTask	cancel task

Support for holding shift key down for multiple selection

Action Type

- Read (56 selected) [Show More](#)
- Write (89 selected) [Show More](#)
- List (20 selected) [Show More](#)

Selected (165)

Action Name	Description
AddInstanceInDeployGroup	add instance in deploy group
AddTimeWindow	add a maintenance time window
AssociateSecurityGroups	bind security groups to instance.
BalanceRoGroupLoad	load balancing in ro group
CancelBatchOperation	stop batch import task
CancelDBInstanceTask	cancel task

Resource All (*)

Condition Source IP ⓘ [Add other conditions.](#)

[+ Add Permissions](#)

[Next](#)

4. On the **Edit Policy** page, enter the **Policy Name** (such as `SQLAuditFullAccess`) as required and **Description** and click **Done**.

✓ Edit Policy > 2 Associate Users/User Groups

Basic Info

Policy Name *

Description

Associate Users/User Groups

Authorized Users [Select Users](#)

Authorized User Groups [Select User Groups](#)

5. Return to the policy list and you can view the custom policy just created.

<input type="checkbox"/>	Policy Name	Description	Last Modified
<input type="checkbox"/>	QcloudAuditFullAccess	Full read-write access to CloudAudit	2021-06-02 14:41:02
<input type="checkbox"/>	QcloudAuditReadOnlyAccess	Read-only access to CloudAudit	2021-06-28 20:12:28
<input type="checkbox"/>	QcloudAccessForCARole	CloudAudit permissions (including but not limited to): COS (COSDataWr...	2021-08-09 10:48:46
<input type="checkbox"/>	QcloudAccessForCloudAuditCARole	CloudAudit permissions (including but not limited to): COS (getBucket/p...	2021-08-09 10:48:46

10