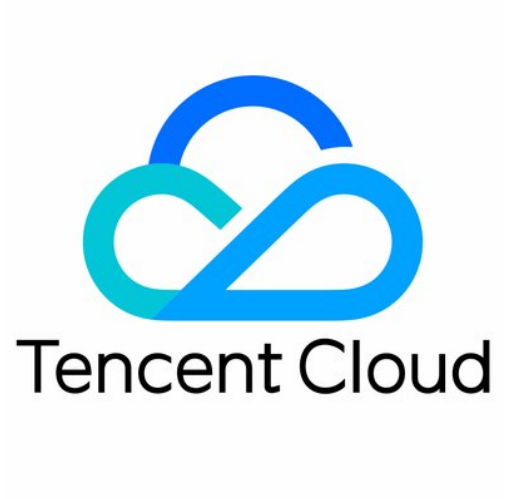
TencentDB for MariaDB

Development Guide

Product Documentation



Contents

Development Guide
Functional Limitations

Last updated：2024-01-11 15:21:58

1. You cannot change any data in the `mysql`, `information_schema`, `performance_schema`, or `sysdb` database.
2. SQL statements cannot be directly used to set accounts or grant permissions, which can only be done in the console.
The following 19 common permissions are supported, and only few rarely used permissions are not supported:
SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, REFERENCES, INDEX, ALTER
CREATE TEMPORARY TABLES, LOCK TABLES, EXECUTE, CREATE VIEW, SHOW VIEW
CREATE ROUTINE, ALTER ROUTINE, EVENT, TRIGGER, SHOW DATABASES
3. No root account is provided.
4. You are recommended to use the InnoDB storage engine.

Performance Test

Last updated：2024-01-11 15:21:58

Overview

Performance test is a comprehensive analysis service for database instance performance and health. It can analyze SQL statement performance, CPU utilization, IOPS utilization, memory utilization, disk utilization, connections, locks, hotspot tables, and transactions, helping you identify and address existing and potential health issues in your database through smart diagnosis and optimization.

Note:

For certain test items, the performance test report provides a series of optimization suggestions. Please carefully test the suggested measures before applying them so as to prevent the instance performance problems from getting worse.

Features

Health rating: you can view the current database performance score out of 100 points. If your database scores below 60 for a long time, please optimize your business or database configuration.

Report generation, viewing, and saving: you can create a report or view the last report as desired. The report can be saved as a webpage for download.

Performance test mainly includes the following features:

Resource analysis

Analyzes the usage of database instance resources (CPU, disk, and connections) in a certain period of time and displays an overall score.

Note:

As most instances have an "overuse when idle" policy enabled by default, you may observe that the maximum CPU utilization exceeds 100%. If this is persistent and the average is higher than the recommended value, you are recommended to scale your instance up.

System status

Sorts out key instance metrics, lists their status and time of occurrence, and suggests corresponding modifications.

Tablespace distribution

Lists the current top 10 tables in reverse order in terms of data space to help you identify oversized tables.

Redundant index detection

Lists the current possible redundant indexes (whose selectivity is below 1%) and suggests optimizations.

Note:

Because a query statement must first query the indices before querying tables through indices, if there are too many identical data entries in the index column, the performance to reduce the amount of data to be filtered may be compromised and is not as fast as full table scan.

Deadlock diagnosis

The deadlock diagnosis gets the information of the last deadlock in the database through `show engine innodb status` and displays it if the deadlock occurs within the selected diagnostic time period.

Note:

If deadlock occurs frequently, it means that two or more concurrent transactions are waiting for one another to give up locks. A fundamental solution is to modify the SQL running logic order and optimize the locking mechanism to reduce the probability of deadlock. A temporary solution is to kill the head blocker.

Lock wait diagnosis

Reports lock waits lasting over 60 seconds in the current time period.

Note:

Lock waits are normal, but sometimes your business may display lock wait timeout errors such as `Lock wait timeout exceeded;try restarting transaction`. MySQL's InnoDB lock information is saved in tables `innodb_trx`, `innodb_lock_waits`, and `innodb_locks` in the system database `information_schema`. Lock wait diagnosis analyzes the lock dependencies in the three tables in the instance, finds the head blocker (session or transaction) that holds a lock for longer than the specified time and blocks other sessions or transactions, and then kill it.

Note:

Currently, lock waits are supported only by InnoDB.

Long running session diagnosis

Lists the sessions whose `Command` column is not `Sleep` but execution time exceeds 10 seconds by diagnosing the `information_schema.processlist` table in the instance.

Note:

The best solution to long running sessions is to optimize SQL and proactively place session invalidation configuration in your business code. Of course, you can also make expired sessions automatically invalid by adjusting the `interactive_timeout` and `wait_timeout` parameters.

Slow query analysis

Lists the current top 20 slow query statements based on the number of executions in reverse order.

Note:

The slow query threshold can be adjusted by the `long_query_time` parameter. Slow queries may occur for many reasons. Generally, if your instance consumes reasonable amounts of resources but a lot of slow queries occur, you are recommended to check whether your business SQL and indexes are appropriate. If your instance has high performance overhead and a lot of slow queries occur, you are recommended to check whether your instance configuration is appropriate and optimize your business SQL and indexes. You can query more details of slow queries using the slow query analysis feature.

Database status check

Checks the health of databases in the current instance.

Others

Lists other values that require DBA's attention.

Intra-city Active-Active Solution

Last updated：2024-01-11 15:21:58

Currently, TencentDB for MariaDB supports the intra-city 2-DC active/active scheme, which has the following main features:
Intra-city 2-DC deployment
2-DC writability: If your servers are deployed in different subnets of two DCs, you can connect to the database and write data to it from any server in either DC.
Automatic failover and recovery
Unique access IP of both DCs
However, the intra-city 2-DC active-active scheme alone cannot implement disaster recovery at the business system level. Actually, it is easy to switch a single system/module to an intra-city disaster recovery DC, but the complicated correlation among and configurations in enterprise-level system businesses are challenges for the 2-DC scheme. Therefore, to build an active-active business system, the business must allow both DCs to run in real time and interwork with each other throughout all phases of design, use, management, and system upgrade, so that the business can be quickly resumed with little to no modification in case of failures. This is also the goal of designing TencentDB for MariaDB's intra-city 2-DC active-active scheme. It allows business systems in both DCs to fully read from and write to the database system over the local network while ensuring strong database consistency.

Design Standard

The active-active feature of TencentDB for MariaDB is designed based on "GB/T 20988-2007 Information Security Technology - Disaster Recovery Specifications for Information System". For a single database module:
RTO ≤ 60 seconds
RPO ≤ 5 seconds
Failover time ≤ 5 seconds
Failure detection time ≤ 30 seconds
This means that it takes about 40 seconds to complete failover after a failure occurs (including failure detection time).
Risk warning: When performing tests in a production environment, make sure that the business system has an automatic database reconnection mechanism. The business system usually has multiple modules, and each module may be associated with multiple data sources; therefore, the more complex the system, the longer the recovery time.

Applicable Regions of Condition Keys

Supported items

Instance Version:
Standard Edition: One primary and one replica (two nodes) or one primary and two replicas (three nodes);
Finance Edition: One primary and one replica (two nodes) or one primary and two replicas (three nodes);
Network requirement: VPC only
Supported regions:
Beijing (Beijing Zone 1, Beijing Zone 3)
Shanghai Finance (Finance Zone 1, Finance Zone 2)
Shenzhen Finance (Finance Zone 1, Finance Zone 2)

Pricing

Dual-AZ and single-AZ schemes are offered at the same price. For more information, see Billing Overview.

Purchase and use

Go to the purchase page of TencentDB for MariaDB and click **Buy Now**.
If the primary and replica AZs are the same, the single-AZ deployment scheme is used.
If the primary and replica AZs are different, the intra-city 2-DC deployment scheme is used.
Note:
The primary AZ is the zone where your primary server is located. The database should be deployed in the same VPC subnet as the primary server to reduce access delay. A replica AZ is the zone where a replica node of the database is located. For the one-primary-two-replica architecture (three nodes), two nodes will be deployed in the primary AZ. For the one-primary-one-replica architecture (two nodes), one node will be deployed in the primary AZ.
If intra-city 2-DC policy is required for the finance cloud cage solution, an intra-city 2-DC cage solution needs to be built first. For more information, contact your sales rep and architect.

Viewing details of instance AZs

Log in to the TencentDB for MariaDB console and click an instance ID or **Manage** in the **Operation** column to enter the instance details page and view the information.

Primary/replica switch

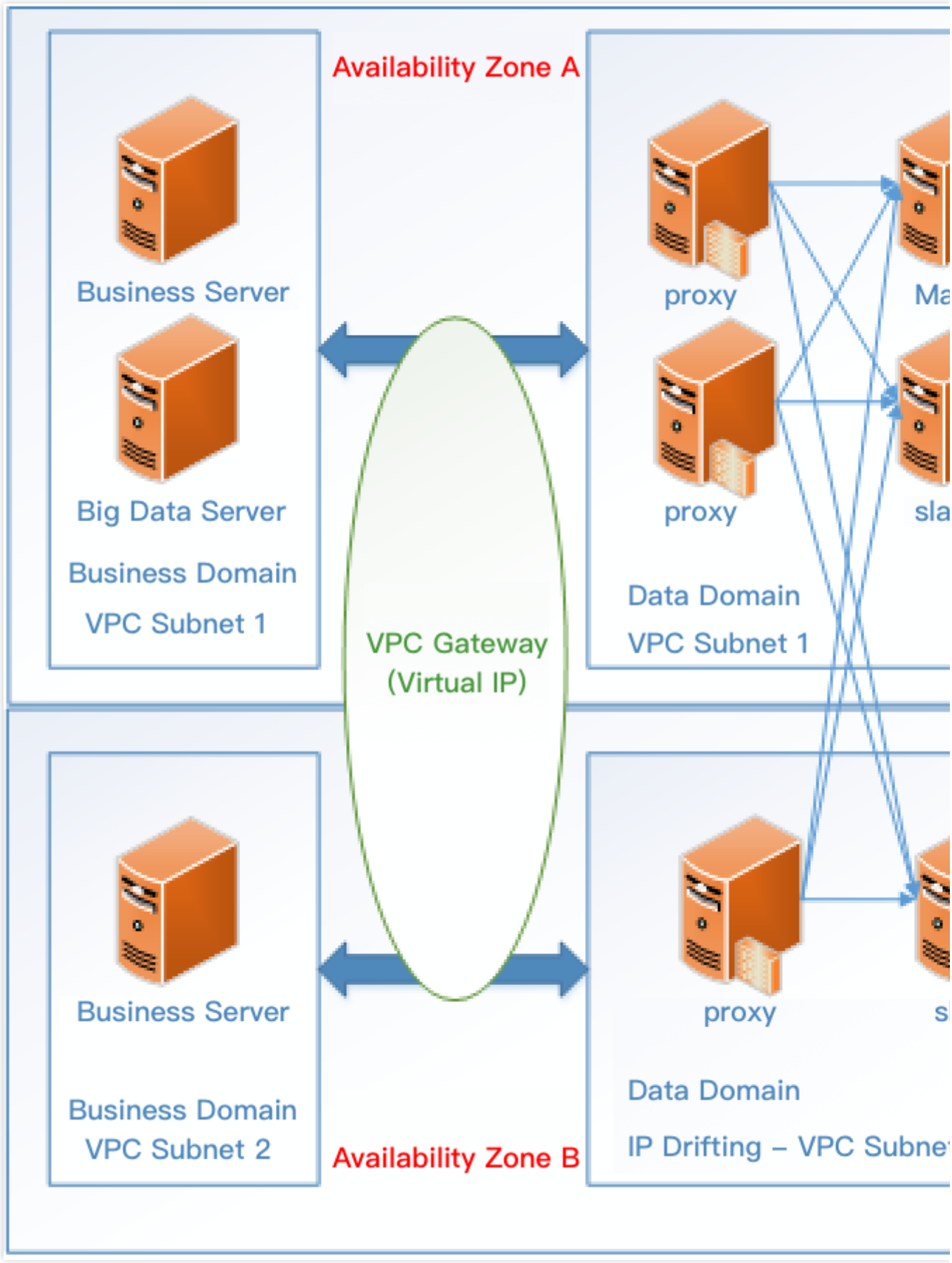
To switch the primary node from one AZ to another, click **Primary-Replica Switch** on the instance details page in the console. As this operation is highly risky, the IP address of the login account must be verified. The switch process may cause a momentary disconnection from the database (≤ 1 second). Make sure that your business has a database reconnection mechanism. Frequent switching may result in business system exceptions or even data exceptions.

How It Works

By integrating the highly available primary/replica architecture of TencentDB for MariaDB with virtual IP drifting of VPC AZ, simultaneous reads from and writes to two DCs can be implemented. This architecture has the following features:

Proxy modules are deployed in a hybrid manner on the frontend of each TencentDB for MariaDB database node, which are responsible for routing data requests to corresponding database nodes.

Cross-region VPC gateway is deployed on the frontend of the proxy module to support virtual IP drifting.



Taking data writes as an example: as shown above, if the business server is deployed in AZ A, the VPC gateway will forward the data request to the proxy gateway in AZ A which will then transparently forward it to the primary node. If the business server is deployed in AZ B, the VPC gateway will forward the data request to proxy gateway in AZ B which will then forward it to the primary node over Tencent Cloud's BGP network.
No matter whether it is a read or write request, the entire process is imperceptible to the business. In case of database exception, the database cluster will be processed as follows:
1. If both the primary and proxy fail, the cluster will automatically promote the optimal replica to the new primary. The system will notify the VPC to modify the association between the virtual and physical IPs. The business will only perceive that some write requests are disconnected.
2. If the primary fails but the proxy is normal, the cluster will automatically promote the optimal replica to the new primary. The proxy will block requests until primary/replica switch is completed. In this case, the business will only perceive that some requests time out.
3. If the replica fails (no matter whether the proxy fails), if read/write separation is enabled, an operation will be performed according to the read-only policy of the preconfigured read-only account (there are three types of policies).
4. If AZ A completely fails, while the VPC and database are still working in AZ B, the replica2 node will be automatically promoted to the primary node. **The read/write policy of the node will be adjusted according to the strong sync policy**, and the VPC IP will drift to AZ B. In this case, the cluster will try to recover the node in AZ A. If the node cannot be recovered within 30 minutes, at least one replica node will be automatically created on node B. As there is an IP drifting policy, no database configuration modification is required of the business.
5. If DC B completely fails, it is equivalent to the failure of a replica node in the TencentDB for MariaDB cluster, and the failure can be processed in the same way as described in item 3 above.

FAQs

Compared with intra-city 1-DC, will the intra-city 2-DC scheme cause a decrease in performance?

Based on the strong sync replication scheme, as the cross-DC delay is slightly larger than that between devices in the same DC, the speed of SQL response will drop by about 5% in theory.

Is it possible for a primary node to switch from the primary AZ to the replica AZ?

Yes. You can ignore this if it does not affect the use of your business. If you are concerned about the impact, you can switch back by using the primary/replica switch feature in the console during off-peak hours.

How do I know that primary/replica switch is performed in the database cluster?

Log in to the Cloud Monitor console, select **Alarm Policy** on the left sidebar, and click **Create**. On the displayed page, select TencentDB for MariaDB as the **Policy Type** and configure an alarm on primary/replica switch.

If part of the read or write requests are handled by the replica AZ, the network delay will cause a decrease in performance, but I need the intra-city 2-DC feature. What should I do?

You can submit a ticket indicating the instance ID, deployment scheme of your servers in the AZ, and the ratio between read and write requests. Tencent Cloud DBA can help you adjust the dual-AZ loading mechanism to minimize the number of read and write requests sustained by the replica AZ.

What should I do if I want to change from the 1-DC architecture to intra-city 2-DC architecture?

Check whether the intra-city 2-DC scheme is supported in your region. It is now available in Beijing, Shanghai Finance, and Shenzhen Finance regions. Then, submit a ticket indicating the information of the account to be adjusted, instance ID, two AZs to be used, and recommended Ops time. Tencent Cloud staff will conduct an audit. If your request is eligible, the operation can be performed; otherwise, it will be rejected.

Binlog Consumption Format

Last updated：2024-01-11 15:21:58

Data subscription helps you get incremental data from TencentDB for MariaDB and TDSQL, so that you can flexibly process real-time incremental data based on your actual business needs.

Feature list

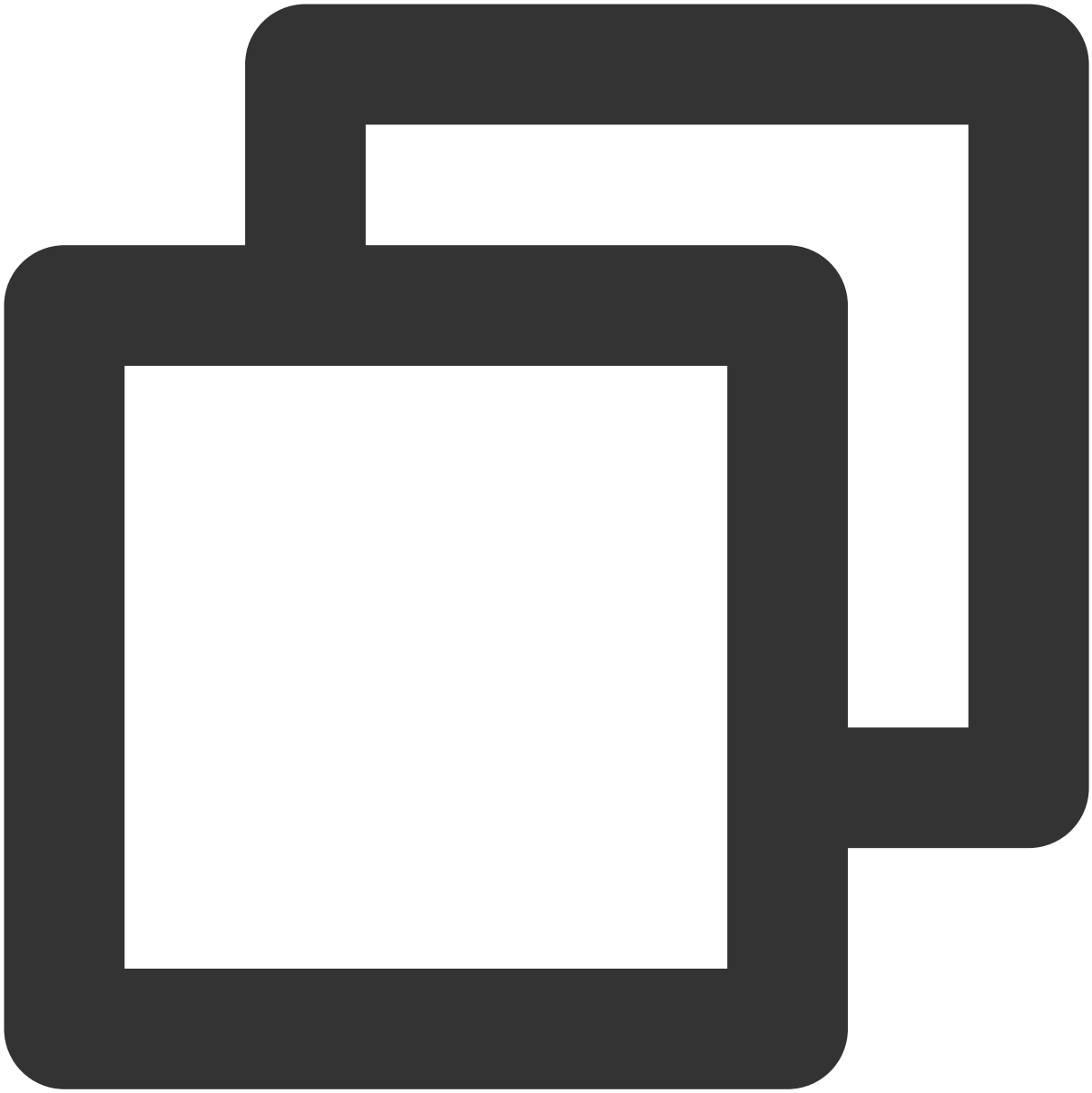
Data subscription is supported for TencentDB for MariaDB and TDSQL instances in public cloud.
Data subscription is supported for TencentDB for MariaDB and TDSQL instances in private cloud.

Data source type

TencentDB for MariaDB and TDSQL.

Message format

The data subscription feature parses instance binlogs (in row format), encapsulates binlog events into messages in JSON format, and uploads them to a Kafka cluster. Message types include `DML`, `GTID`, `XID`, and `QUERY` events, which represent modification to data rows, start of transactions, commit of transactions, and `DDL` statements, respectively. DML events include `insert`, `update`, and `delete` events.
The message format of a `DML` event is as follows:



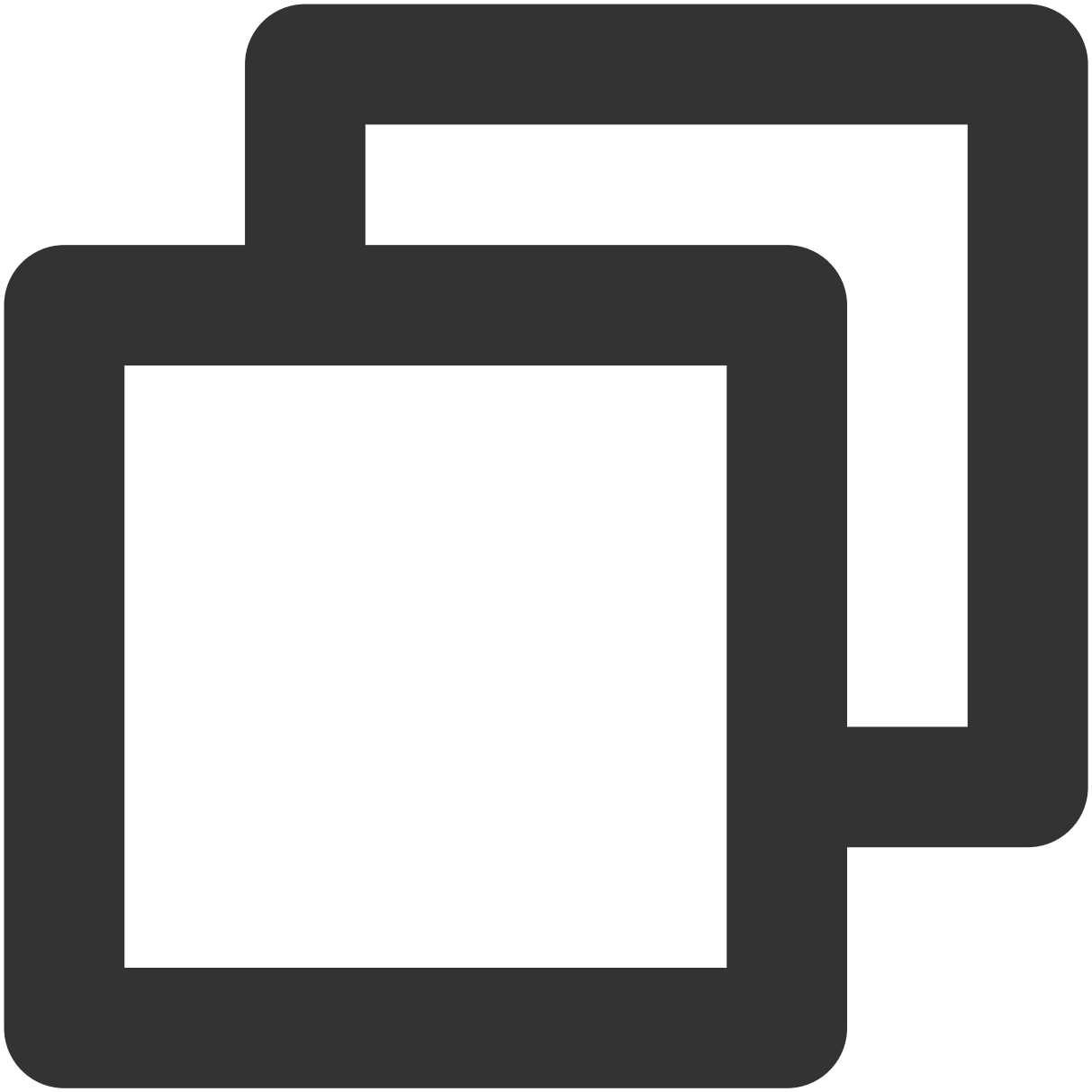
```
{
    "logtype":"mysqlbinlog",          // Log type. The value is unique and must be
    "eventtype":23,                   // Event type code of a binlog in MySQL

    "eventtypestr":"insert",          / *Event type string. The value can be `inser
                                        An `insert`, `update`, or `delete` event in
                                        An `xid` event indicates end of a transacti
    "db":"testdb",                    // Database name
    "table":"testtable",              // Table name
    "localip":"000.00.000.000",       // IP of the server where the instance is loc
    "localport":0000,                 // Instance port
```

```
    "begintime":1511350073,           // Start time of the transaction to which the
    "gtid":"0-2670193178-726233561",  // `GTID` of the transaction to which the cur
    "event_index":"4",                // Number of the event in the transaction
    "where":[                         // `where` field, which indicates the value o

    ],
    "field":[                         // `field` field, which indicates the value o
        "1",
        "'name1'"
    ]
}
```

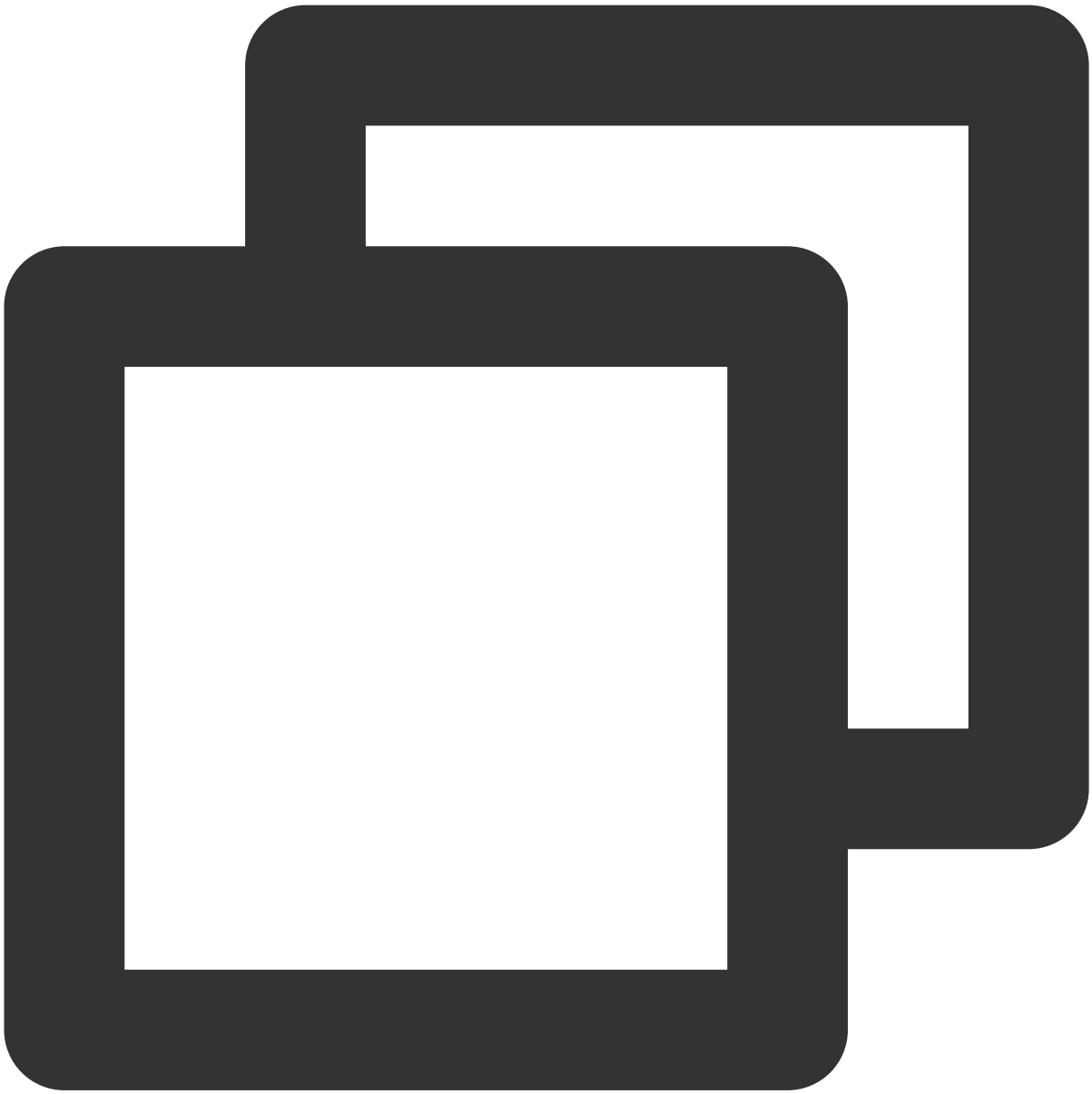
The message format of a `GTID` event is as follows:



```
{                                   // A `GTID` event represents the start of a tr
    "logtype":"mysqlbinlog",
    "eventtype":33,
    "eventtypestr":"gtid",
    "db":"sysdb",
    "table":"statustableforhb",
    "localip":"10.231.23.241",
    "localport":8810,
    "begintime":1511419963,
    "gtid":"35be190b-d019-11e7-ab7a-a0423f32c225:469",
    "event_index":"1"
```

```
}
```

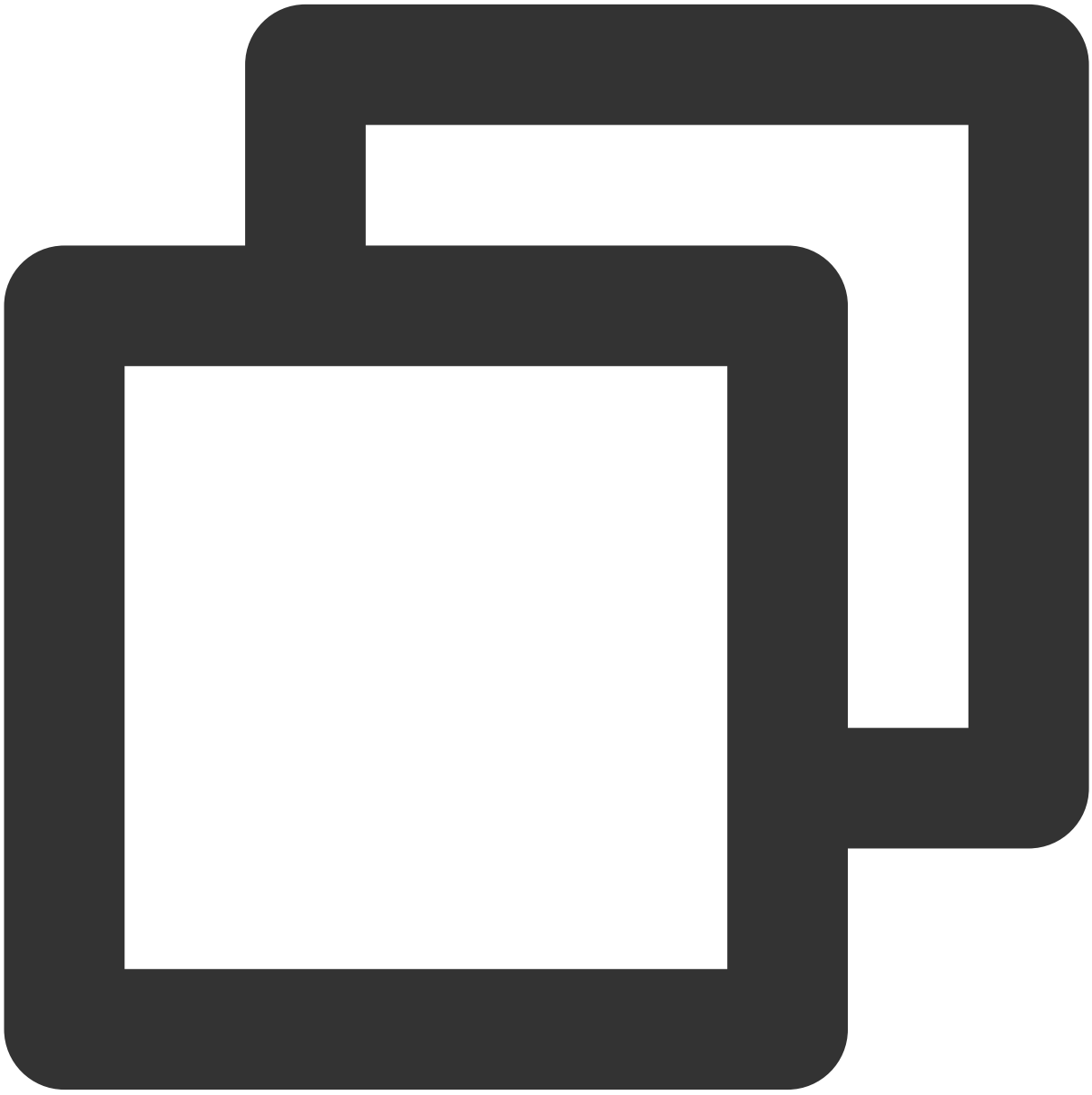
The message format of an `XID` event is as follows:



```
{                                   // An `XID` event represents that a transactio
    "logtype":"mysqlbinlog",
    "eventtype":16,
    "eventtypestr":"xid",
    "db":"testsummer",
    "table":"test_table1",
    "localip":"10.231.23.241",
```

```
    "localport":8810,
    "begintime":1511419963,
    "gtid":"35be190b-d019-11e7-ab7a-a0423f32c225:469",
    "event_index":"5",
    "xid":"11866"
}
```

The message format of a `QUERY` event is as follows:



```
{
    "logtype":"mysqlbinlog",
```

```
    "eventtype":2,
    "eventtypestr":"query",         // `DDL` statement of a `QUERY` event
    "db":"testsummer",
    "table":"statustableforhb",
    "localip":"10.231.23.241",
    "localport":8810,
    "begintime":1511419941,
    "gtid":"35be190b-d019-11e7-ab7a-a0423f32c225:452",
    "event_index":"2",
    "sql":"create table test_table1 (id int primary key,name varchar(20))"
}
```

Subscription method

You can use message events stored in a Kafka cluster to get the data in real time and process messages after getting them through the data subscription API provided by Tencent Cloud.

Slow Query Analysis

Last updated：2024-01-11 15:21:58

1. Feature Description

A SQL statement query that takes more time than the specified value is referred to as a "slow query", and the corresponding statement is called a "slow query statement". The process where a database administrator (DBA) analyzes slow query statements and finds out the reasons why slow queries occur is known as "slow query analysis". TencentDB for MariaDB provides slow query analysis on the **Performance Optimization** tab on the instance management page.

2. Main Parameters

2.1. Main default settings

Slow log feature: Enabled by default.
Slow query threshold (long_query_time): 1 second by default, that is, only query statements executed for more than 1 second will be logged.
Analyzed data output delay: 1–5 minutes.
Logging duration: 30 days, depending on the backup and log settings.

2.2. Fields in the analysis list

Checksum (checksum): A sequence of digits used to identify a slow query statement (64-bit by default);
Abstracted SQL Statement (fingerprint): A slow query statement with user data hidden.
Database: The database in which the slow query statement was executed.
Account: The account under which the slow query statement was executed;
Last Execution Time (last_seen): The time when the slow query statement was last executed within the specified time range.
First Execution Time (first_seen): The time when the slow query statement was first executed within the specified time range.
Total (ts_cnt): The number of executions of the slow query statement within the specified time range.
Execution Proportion (%): The ratio of total executions of the slow query statement to the total executions of all slow query statements within the specified time range.
Total Time (query_time_sum): The total time consumed by the slow query statement within the specified time range.
Total Time (%): The ratio in percentage of the total time consumed by the slow query statement to the total time consumed by all slow query statements within the specified time range.

Average Time (query_time_avg): The average time is calculated by dividing the total time consumed by the slow query statement by the total number of executions of the slow query statement.
Min Time (query_time_min): The minimum among all execution time of the slow query statement.
Max Time (query_time_max): The maximum among all execution time of the slow query statement.
Total Lock Time (lock_time_sum): The total lock time of the slow query statement.
Total Lock Time Ratio: The ratio in percentage of the total lock time of the slow query statement to the total lock time of all slow query statements.
Average Lock Time (lock_time_avg): The average time calculated by dividing the total lock time of the slow query statement by the total number of locks of the slow query statement.
Min Lock Time (lock_time_min): The minimum among all lock time of the slow query statement.
Max Lock Time (lock_time_max): The maximum among all lock time of the slow query statement.
Sent Rows (Rows_sent_sum): The total number of data rows sent by the slow query statement.
Scanned Rows (Rows_examined_sum): The total number of data rows scanned by the slow query statement.
Host Address (Host): The host from which this slow query comes.

Database Audit

Last updated：2024-01-11 15:21:58

Note:

The database audit feature is being upgraded, during which new instances won't support this feature. But it will be available again very soon.

Overview

Background

You may face the following security risks when using databases, which calls for a complete post-transaction auditing and tracking mechanism. This is where database auditing comes into play.

Administrative risks

Business system security risks caused by faulty, non-compliant, and unauthorized operations by system administrators.

Difficulty in defining accountability due to account sharing.

Faulty and malicious operations and tampering by third-party development and maintenance personnel.

Excessive permissions granted to the root account, which cannot be audited and monitored.

Technical risks

Backdoors or vulnerabilities of application system developers.

Backdoors left by former employees.

Political risks

Inability to satisfy the requirements defined by China's Cybersecurity Classified Protection Certification (Level 3, 7.1.3.3).

Inability to satisfy the requirements defined by industry-specific information security compliance documents, such as Testing and Evaluation Guide for Classified Protection of Information System of Financial Industry stipulated by PBOC.

Terminology

Audit policy: a policy that defines what behaviors are to be audited and how to audit them. **Audit policy = audit object + audit rule + responsive action**. In other words, when configuring an audit policy, you need to specify the things to be audited, and if the characteristics of certain (user or system) behaviors hit an audit rule after being analyzed during the validity period of the policy, then the audit engine will take responsive actions as defined in the policy, such as sending an alarm.

Audit rule: a rule is a collection of behaviors that need to be audited as defined in an audit policy. A rule consists of rule parameters, each of which defines a specific characteristic for behavior matching.

Capabilities and restrictions

TencentDB provides a database audit feature. Audit logs are retained for 7 days by default to help you perform risk management on database access and improve database security.

Audit Operations

Enabling database audit

You can enable database audit for TencentDB for MariaDB free of charge on the database audit page.

> Precautions on enabling audit:

You must have at least one TencentDB for MariaDB instance which is not deactivated or isolated; otherwise, the system will automatically disable the audit feature.
TencentDB for MariaDB instances purchased before June 5, 2016 need to be restarted and upgraded before they can support the audit feature. As this process may cause business interruption for 1–5 seconds, you can contact Tencent Cloud customer service to schedule an upgrade time.
Database audit logs are in plaintext. Therefore, you are recommended to enable MFA.
It takes several minutes to initialize the audit feature. Please wait patiently.

Creating an audit rule

After the audit feature is enabled, logs will be forwarded to the audit cluster through the TencentDB for MariaDB gateway cluster. As no audit rule or policy has been created, the logs will not be recorded and displayed persistently. Therefore, you can select **Create Audit Rule** > **Associate Audit Policy** to store logs in the audit cluster.
1. Go to the audit rule page and click **Create Rule**.
2. Enter the audit rule name and click **Next**.
3. Go to the parameter setting page and set the rule parameters (at least one of the listed parameters is required, but you do not need to set all of them).
Logic of Rule Parameters: the logic between each parameter in a rule is "AND", which means that the rule will only be considered a match when all parameters meet the conditions.
Characteristic String: it defines the parameter details, i.e., the specific characteristics of audit objects. To implement exact match, you only need to define keywords of the desired parameters, so that the audit system can record only custom rules to improve audit efficiency. Note: an empty string indicates that the parameter is ignored, i.e., "match all".
Match Type: relationship between parameter object and characteristic.
Included: it means the match will be successful if a characteristic string is displayed in the network field.
Excluded: it means the match will be successful if a characteristic string is not displayed in the network field.

Equal to: it means the match will be successful if the network field is equal to the characteristic string.
Not equal to: it means the match will be successful if the network field is not equal to the characteristic string.
Regex: it represents the characteristic string and supports standard regex syntax.
4. You can view all created rules in the rule list.
5. After completing audit rule settings, you can modify them at any time. You can create similar rules through **clone rule** to improve efficiency.

Creating an audit policy

An audit policy is an audit scheme composed of audit rules, audit objects, and responsive actions. You can set multiple audit policies for one instance. When the audit engine parses policies, it will **match them in the order of configured priority from top to bottom**.
1. Select **Audit Policy** and click **Create Policy**.
2. Enter the policy requirement, select the instance to be audited based on your needs, and select the corresponding rules (alarm configuration is not supported currently).
3. Adjust the priority: you can adjust the priority of multiple policies under the same instance; the smaller the number, the higher the priority. The adjustment will take effect in 1 minute.
4. You can modify audit policies in real time by using the modification feature. Modified policies will take effect in about 5 minutes and then be used for audit and monitoring, but logs recorded prior to the modification will not be modified.

Viewing logs

SQL statements that hit audit policies are displayed on the audit logs page. You can click them view or search. Pay attention to the following points:
Audit logs are in plaintext. Therefore, you are recommended to enable MFA to ensure log security.
Logs are recorded only after an audit policy is created. Historical data is not recorded.
Each transaction and stored procedure may be recorded as a single statement. For more information, please see Syntax Supported by Database Audit.
The allowed maximum length of one SQL statement is 1 KB. Excessive content will be truncated.

Syntax Supported

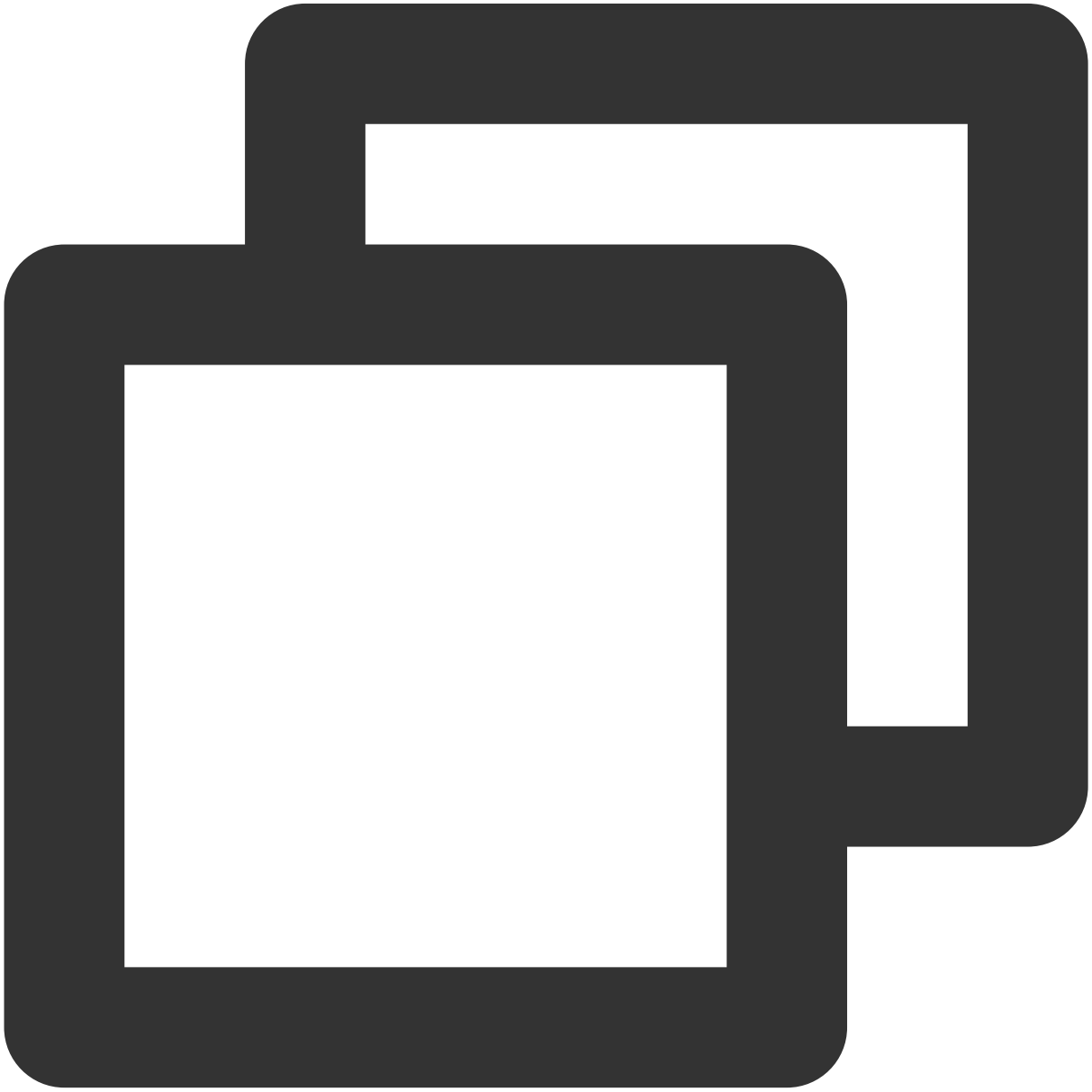
Last updated：2024-01-11 15:21:58

Note:

As the database audit feature is being refactored and upgraded, it is not available for newly purchased instances during this period.

Database audit currently supports most SQL statements. If you find any deficiency, contact us for feedback.

Parsing of DCL, DDL, and DML statements is supported.



```
Insert,Replace,Select,Union,Update,Delete,CreateDatabase:,CreateEvent,CreateFunctio
CreateTable,CreateServer,CreateProcedure,CreateTablespace,CreateTrigger,CreateView,
ShowCharset,ShowCollation,ShowColumns,ShowCreate,ShowCreateDatabase,ShowDatabases,S
ShowEvents,ShowFunction,ShowGrants,ShowLogEvents,ShowLogs,ShowProcedure,ShowOpenTab
ShowProcessList,ShowMasterStatus,ShowPrivileges,ShowProfiles,ShowSlaveHosts,ShowSla
ShowWarnings,ShowVariables,ShowStatus,ShowTriggers,Call,DropProcedure,DropDatabase,
DropIndex,DropLogfile,DropServer,DropTables,DropTablespace,DropTrigger,DropUser,Dro
AlterEvent,AlterFunction,AlterLogfile,AlterProcedure,AlterServer,AlterTable,AlterTa
AlterView,Rollback,Commit,Begin,Set,SetTrans,SetPassword,Release,Grant,RenameTable,
Install,StopSlave,StartSlave,StartTrans,Use,DescribeTable,DescribeStmt,Flush,Load,L
Reset,CacheIndex,TruncateTable,Lock,Unlock,SavePoint,Help,Do,SubQuery,ShowTables,Ex
Kill,Partition,PrepareRepairXACheckCheckSumAnalyzeChangeOptimizePurgeHandlerSignalR
```

Transaction and procedures may be divided into multiple statements.