TencentDB for MariaDB Development Guide

Contents

Development Guide
Functional Limitations

Last updated: 2023-08-31 09:37:33

1. You cannot change any data in the `mysql`, `information_schema`, `performance_schema`, or `sysdb` database.
2. Account and authorization related operations cannot be performed directly through SQL statements, they can only be managed via the control console. We support 19 common permissions, with a few uncommon ones not supported. The specific list of supported permissions is as follows: 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. The root account is not provided.
4. It is recommended to use the InnoDB storage engine.

Performance Test

Last updated: 2023-08-31 09:37:47

Feature 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.

Feature Overview

Health Rating: You can view the current comprehensive database performance score, with a maximum of 100 points. If the score remains below 60 for an extended period, consider optimizing your business or database configuration.
Report Generation, Viewing, and Saving: You can create reports at will and view the most recent one generated. Reports can be saved as webpages and downloaded for local review.
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 the idle overuse policy enabled by default, you may observe that the maximum CPU utilization can exceed 100%. If your CPU usage consistently goes beyond 100% and the average value is higher than the recommended level, it is advisable to expand your resources as soon as possible.

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

Deadlock diagnosis retrieves the last deadlock information from the database by diagnosing `show engine innodb status`. If the deadlock occurs within the user-selected diagnostic time period, it will be displayed.

> **Note**
> If deadlocks occur too frequently, it indicates that the SQL statements within transactions are prone to creating cycles when executed concurrently. The fundamental solution is to modify the SQL execution logic order, optimize the locking mechanism, and reduce the probability of deadlocks. A temporary solution is to kill the leading blocking session.

Lock wait diagnosis

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

> **Note**
> Lock waits are normal occurrences, but sometimes businesses may encounter errors like `Lock wait timeout exceeded; try restarting transaction`. In MySQL, InnoDB lock information is stored in the `information_schema` system library, under the three tables `innodb_trx, innodb_lock_waits, and innodb_locks`. Lock wait diagnosis involves analyzing the lock dependency relationships among these three tables to identify the leading transaction and session information of sessions that hold locks for longer than a

certain threshold and block other sessions, as well as the session information of the blocked transactions. The leading session is then terminated using the KILL command.

Note
Currently, lock waits are supported only by InnoDB.

Long running session diagnosis

Diagnose sessions with a Command other than Sleep and an execution time (Time) exceeding 10 seconds in the `information_schema.processlist` of the instance.

Note
The best way to resolve long sessions is to optimize SQL and proactively implement session expiration configurations in your business code. Alternatively, you can adjust the `interactive_timeout` and `wait_timeout` parameters to actively expire outdated sessions.

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 through the `long_query_time` configuration. There are various reasons for slow queries. Generally, if your instance has reasonable resource consumption and numerous slow queries, it is recommended to focus on the rationality of business SQL and indexes. If your instance has high performance consumption and numerous slow queries, it is suggested to pay attention to the appropriateness of instance configuration and optimize business SQL, indexes, etc. More detailed slow query data can be found under the slow query analysis feature.

DB status check

Checks the health status of the DB layer in the current database.

Others

Lists other values that require DBA's attention.

Intra-city Active-Active Solution

Last updated: 2023-08-31 09:38:05

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.
To build a dual-active business system, it is essential to base the design, usage, management, and system upgrade processes on the 2-DC architecture, with real-time usage and configuration intercommunication. This ensures that the business can quickly resume operation with little or no modification after a failure. The goal of TencentDB for MariaDB's intra-city 2-DC active/active design is to enable both business systems in the two DCs to read and write to the database system through their local networks while maintaining strong data consistency.

Design standards

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.

Support Status

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

The pricing for dual availability zones is the same as that for a single availability zone. For more information, see Pricing Details.

Purchase and use

Please visit the TencentDB for MariaDB purchase page and click "Buy".

- 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 master AZ is the region where your primary server is located. Ideally, the database should be allocated in the same VPC subnet as the primary server to minimize access latency. The replica AZ is where the database replica nodes are located. If there is a 3-node configuration with one master and two replicas, the master AZ will have two nodes deployed. If there is a 2-node configuration with one master and one replica, the master AZ will have one node deployed.
> - 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 availability zones

You can visit the MariaDB console and click on the instance ID or **Manage** in the **Operation** column to access the instance details page.

Master-slave switch

To switch the master node from one availability zone (AZ) to another, simply click **Master/Replica Switch**. This is a high-risk operation that requires IP address verification of the login account. The switch may cause a brief database disconnection (≤1s), so ensure your business has a database reconnection mechanism. Frequent switching may lead to system anomalies or even data inconsistencies.

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.
- Deploy a cross-regional VPC gateway in front of the Proxy module, supporting virtual IP migration functionality.
  As illustrated above, taking data writing as an example, suppose the business server is deployed in Availability Zone A. The VPC gateway forwards data requests to the Proxy gateway in Availability Zone A, which then transparently forwards them to the Master node. If the business server is deployed in Availability Zone B, the VPC gateway forwards data requests to the Proxy gateway in Availability Zone B, which then transparently forwards them (via Tencent Cloud BGP private network) to the Master node.
  The entire process is transparent to the business, whether it is a read or write request. In case of a database exception, the database cluster handles it according to the following principles:
- 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.
- 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.
- In case of a Slave failure (regardless of whether the Proxy is faulty or not), during read/write separation, the pre-configured read-only account **read-only policy** (with three types) will be executed.

- If AZ A experiences a complete failure, the VPC and database in AZ B will still be operational. At this point, the slave2 node will be automatically promoted to Master, and **the read/write policy of this node will be adjusted according to the strong synchronization strategy**. The VPC network IP will migrate to AZ B. The cluster will then attempt to restore the node in AZ A. If the node cannot be restored within 30 minutes, at least one Slave node will be automatically rebuilt in AZ B. Due to the IP migration policy, there is no need to modify the database configuration for the business.
- 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, if it doesn't affect your business usage, you can ignore it. If you're concerned about the impact, you can switch back during off-peak hours using the primary/replica switch feature in the console.

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

Please go to the Tencent Cloud Observability Platform Console > Alarm Policy > TencentDB for MariaDB > Configure Primary-Replica Switch Alarm.

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 specifying the instance ID, your server's AZ deployment scheme, and the read/write request ratio. Tencent Cloud DBAs can help you adjust the dual-AZ load balancing mechanism to minimize the read/write requests handled by the secondary 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: 2023-08-31 09:38:26

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 the binlog (row format) of instances and encapsulates binlog events into JSON-formatted messages uploaded to the Kafka cluster. Message types include DML events, GTID events, XID events, and QUERY events. DML events consist of insert, update, and delete events, representing changes to data rows; GTID events signify the start of a transaction; XID events indicate the transaction's commitment; QUERY events represent DDL statements.
The DML message format is as follows:

```
{
  "logtype": "mysqlbinlog",         // Log type, unique value: mysqlbinlog
  "eventtype": 23,                  // Event type code, corresponding to the binlog event type

  "eventtypestr": "insert",         /*Event type string, including insert, update, delete, gtid
                                The INSERT, UPDATE, and DELETE events correspond to DML sta
                                xid event indicates the end of a transaction; query event indicat
  "db": "testdb",                   // Database name
  "table": "testtable",             // Table name
  "localip": "000.00.000.000",      // IP of the machine where the instance is located
  "localport": 0000,                // Instance port
  "begintime": 1511350073,          // Transaction start time, the start time of the transac
  "gtid": "0-2670193178-726233561",  // GTID, the gtid of the transaction where the curr
  "event_index":"4",                // Represents the sequence number of this event within th
```

```
    "where":[                    //where field, indicating the values of each field before the ro
    ],
    "field": [                   // The field attribute represents the values of each field after th
      "1",
      "'name1'"
    ]
}
```

GTID message format is as follows:

```
{                                // A GTID event represents the start of a transaction
    "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"
}
```

XID message format is as follows:

```
{                                // An XID event represents that a transaction has been committe
    "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"
}
```

QUERY message format is as follows:

```
{
```

```
    "logtype":"mysqlbinlog",
    "eventtype":2,
    "eventtypestr": "query", // QUERY event corresponds to DDL statements
    "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

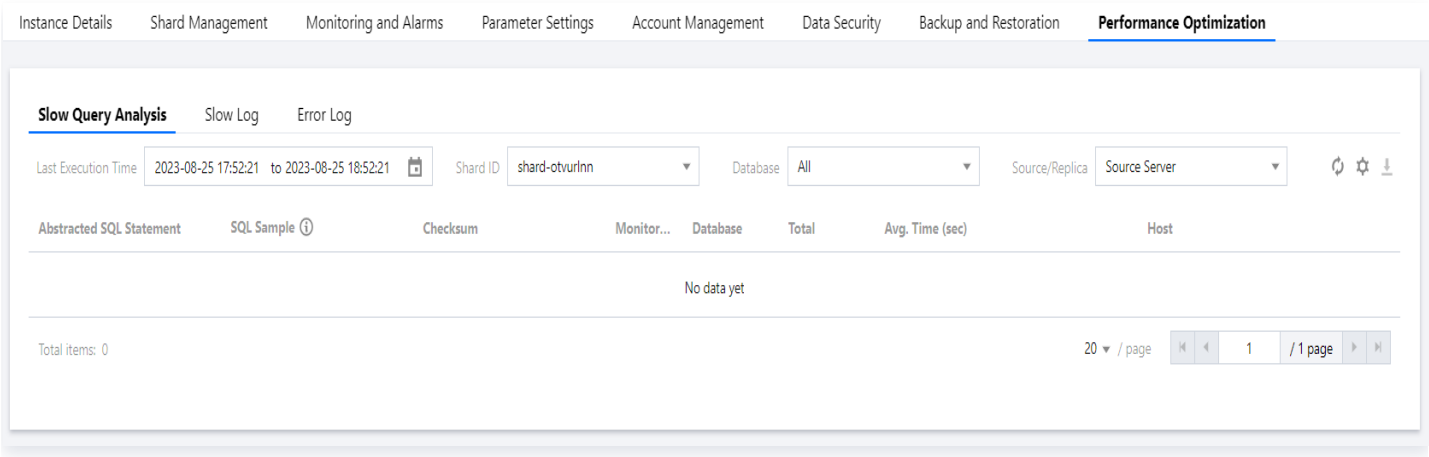
Customers can obtain real-time data by accessing message events stored in the Kafka cluster. They can use Tencent Cloud's data subscription API to fetch messages in real-time and process them accordingly.

Slow query analysis

Last updated: 2023-08-31 09:38:44

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 capabilities under the **Performance Optimization** module 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.

Reserved Keywords

Last updated: 2023-08-31 09:40:16

This article enumerates the reserved keywords currently involved in TencentDB for MariaDB. When used in SQL, they need to be parsed with `''`.
Primary Reserved Keyword Change Log

Keyword	Change content	Version
CYCLE	Allow CYCLE keyword as table and field name	22.3.0
SEQUENCE	Allow SEQUENCE keyword as table and field name	22.3.0
TDSQL_SUBDISTRIBUTED	Proxy Multi-level Partition Table	22.3.0
WITHOUT	Proxy Recycle Bin Syntax	22.3.0
RECYCLE_BIN	Proxy Recycle Bin Syntax	22.3.0
RECYCLE_ID	Proxy Recycle Bin Syntax	22.3.0
CLEAR	Proxy Recycle Bin Syntax	22.3.0
TEMPLATE	Set-level Global Index	22.2.0
TDSQL_AUTOINCVAL	Manually obtain auto-increment ID using command	22.1.4
MANUAL_SWITCH	Rebalance Manual Route Switching	22.1.0
SWITCH	Rebalance Manual Route Switching	22.1.0
SWITCH_DATETIME	Rebalance Manual Route Switching	22.1.0
DDL	Query DDL task-related information	20.8
OFFLINE	Offline Rebalance	19.1
REBALANCE_TASK	Query Rebalance Task Information	19.1
TDSQL_RESETVAL	Reset sequence number	17.2
TDSQL_DISTRIBUTED	Range/List Partitioned Table	16.3
SKIP	Options for SELECT FOR UPDATE	15.1

LOCKED	Options for SELECT FOR UPDATE	15.1
NOWAIT	Options for SELECT FOR UPDATE	15.1
OF	Options for SELECT FOR UPDATE	15.1
CYCLE	Proxy compatible with DB sequence syntax	15.1
INCREMENT	Proxy compatible with DB sequence syntax	15.1
LASTVAL	Proxy compatible with DB sequence syntax	15.1
MAXVALUE	Proxy compatible with DB sequence syntax	15.1
NEXTVAL	Proxy compatible with DB sequence syntax	15.1
NOCACHE	Proxy compatible with DB sequence syntax	15.1
NOCYCLE	Proxy compatible with DB sequence syntax	15.1
NOMAXVALUE	Proxy compatible with DB sequence syntax	15.1
NOMINVALUE	Proxy compatible with DB sequence syntax	15.1
PREVIOUS	Proxy compatible with DB sequence syntax	15.1
RESTART	Proxy compatible with DB sequence syntax	15.1
REUSE	Proxy compatible with DB sequence syntax	15.1
SEQUENCE	Proxy compatible with DB sequence syntax	15.1
SETVAL	Proxy compatible with DB sequence syntax	15.1

TDSQL_CYCLE	Proxy supports sequence syntax with tdsql_ prefix	13.21
TDSQL_INCREMENT	Proxy supports sequence syntax with tdsql_ prefix	13.21
TDSQL_LASTVAL	Proxy supports sequence syntax with tdsql_ prefix	13.21
TDSQL_MINVALUE	Proxy supports sequence syntax with tdsql_ prefix	13.21
TDSQL_NEXTVAL	Proxy supports sequence syntax with tdsql_ prefix	13.21
TDSQL_NOCACHE	Proxy supports sequence syntax with tdsql_ prefix	13.21
TDSQL_NOCYCLE	Proxy supports sequence syntax with tdsql_ prefix	13.21
TDSQL_NOMAXVALUE	Proxy supports sequence syntax with tdsql_ prefix	13.21
TDSQL_NOMINVALUE	Proxy supports sequence syntax with tdsql_ prefix	13.21
TDSQL_PREVIOUS	Proxy supports sequence syntax with tdsql_ prefix	13.21
TDSQL_RESTART	Proxy supports sequence syntax with tdsql_ prefix	13.21
TDSQL_REUSE	Proxy supports sequence syntax with tdsql_ prefix	13.21
TDSQL_SEQUENCE	Proxy supports sequence syntax with tdsql_ prefix	13.21
TDSQL_SETVAL	Proxy supports sequence syntax with tdsql_ prefix	13.21

Earlier Reserved Keyword List

&&, <, <=, <>, !=, =, >, >=, <<, >>, <=>, ACCESSIBLE, ACTION, ADD, ADMIN, AFTER, AUTOEXTEND_SIZE, AUTO, AVG, AVG_ROW_LENGTH, BACKUP, BEFORE, BEGIN, BETWEEN CATALOG_NAME, CHAIN, CHANGE, CHANGED, CHAR, CHARACTER, CHARSET, CHECK, CHE

```
COLUMNS, COLUMN_ADD, COLUMN_CHECK, COLUMN_CREATE, COLUMN_DELETE, COLUMN
CONSTRAINT_CATALOG, CONSTRAINT_NAME, CONSTRAINT_SCHEMA, CONTAINS, CONTEXT
CURRENT_TIMESTAMP, CURRENT_USER, CURSOR, CURSOR_NAME, CYCLE, DATA, DATABAS
DEFAULT, DEFINER, DELAYED, DELAY_KEY_WRITE, DELETE, DESC, DESCRIBE, DES_KEY_FIL
DUPLICATE, DYNAMIC, EACH, ELSE, ELSEIF, ENABLE, ENCLOSED, END, ENDS, ENGINE, ENG
EXPORT, EXPLAIN, EXTENDED, EXTENT_SIZE, FALSE, FAST, FAULTS, FETCH, FIELDS, FILE, F
GEOMETRY, GEOMETRYCOLLECTION, GET_FORMAT, GET, GLOBAL, GRANT, GRANTS, GROU
IF, IGNORE, IGNORE_SERVER_IDS, IMPORT, IN, INCREMENT, INDEX, INDEXES, INDEX_STAT
INTEGER, INTERVAL, INTO, IO, IO_THREAD, IPC, IS, ISOLATION, ISSUER, ITERATE, INVOKER
LEVEL, LIKE, LIMIT, LINEAR, LINES, LINESTRING, LIST, LOAD, LOCAL, LOCALTIME, LOCALTIM
MASTER_GTID_POS, MASTER_HOST, MASTER_LOG_FILE, MASTER_LOG_POS, MASTER_PASS
MASTER_SSL_CRL, MASTER_SSL_CRLPATH, MASTER_SSL_KEY, MASTER_SSL_VERIFY_SERVI
MAX_SIZE, MAX_UPDATES_PER_HOUR, MAX_USER_CONNECTIONS, MAXVALUE, MEDIUM, M
MINUTE_SECOND, MINVALUE, MIN_ROWS, MOD, MODE, MODIFIES, MODIFY, MONTH, MULT
NEXTVAL, NO, NOMAXVALUE, NOMINVALUE, NOCACHE, NOCYCLE, NO_WAIT, NODEGROUP
OPTIONS, OPTION, OPTIONALLY, OR, ORDER, OUT, OUTER, OUTFILE, OWNER, PACK_KEYS,
PLUGINS, POINT, POLYGON, PORT, PRECISION, PREPARE, PRESERVE, PREV, PREVIOUS, PRI
READ_ONLY, READ_WRITE, READS, REAL, REBUILD, RECOVER, REDO_BUFFER_SIZE, REDO
REORGANIZE, REPAIR, REPEATABLE, REPLACE, REPLICATION, REPEAT, REQUIRE, RESET, R
ROLE, ROLLBACK, ROLLUP, ROUTINE, ROW, ROW_COUNT, ROWS, ROW_FORMAT, RTREE, S
SERIALIZABLE, SESSION, SERVER, SET, SETVAL, SHARDKEY, SHARE, SHOW, SHUTDOWN,
SPECIFIC, SQL, SQLEXCEPTION, SQLSTATE, SQLWARNING, SQL_BIG_RESULT, SQL_BUFFER_
SQL_TSI_DAY, SQL_TSI_WEEK, SQL_TSI_MONTH, SQL_TSI_QUARTER, SQL_TSI_YEAR, SSL, S
STRING, SUBCLASS_ORIGIN, SUBJECT, SUBPARTITION, SUBPARTITIONS, SUPER, SUSPEND,
THAN, THEN, TIME, TIMESTAMP, TIMESTAMPADD, TIMESTAMPDIFF, TINYBLOB, TINYINT, TIN
UNDO_BUFFER_SIZE, UNDOFILE, UNDO, UNICODE, UNION, UNIQUE, UNKNOWN, UNLOCK,
UTC_TIMESTAMP, VALUE, VALUES, VARBINARY, VARCHAR, VARCHARACTER, VARIABLES, VA
YEAR, YEAR_MONTH, ZEROFILL, ||, BOOST
```

Database audit

Last updated: 2023-08-31 09:42:02

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

Overview

Background Description

Enterprises using databases may face the following security risks, which require a comprehensive post-event audit and tracing mechanism. Database audit capabilities are designed to address these risks.

Management Risks

- Business system security risks caused by faulty, non-compliant, and unauthorized operations by system administrators.
- The responsibility becomes blurred when multiple individuals share a single account.
- Faulty and malicious operations and tampering by third-party development and maintenance personnel.
- The root account has excessive permissions, making it impossible to audit and monitor.

Technical Risks

- Backdoors or vulnerabilities introduced by application system developers.
- Backdoors left by former employees.

Policy Risks

- Unable to meet the explicit requirements of the National Graded Protection (Level 3) (7.1.3.3).
- Inability to meet the requirements defined by industry-specific information security compliance documents, such as the "Implementation Guidelines for Information Security Grading Protection of Information Systems in the Financial Industry" issued by the People's Bank of China.

Term Definitions

Audit Policy: A strategy that defines which user behaviors to audit and how to respond. **Audit Policy = Audit Object + Audit Rule + Response Action** To configure an audit policy, you need to specify the audit content. If, after parsing, certain (user or system) behaviors match an audit

rule and occur during the policy's effective time, the audit engine will respond according to the defined response method, such as issuing an alert.
Audit Rules: In audit policies, a collection of behaviors that need to be audited is referred to as rules. Rules are composed of rule parameters, and each rule parameter defines a specific behavior matching characteristic.

Capabilities and restrictions

Tencent Cloud provides database audit capabilities, with audit logs saved for 7 days by default, to help enterprises manage potential risks associated with database access and improve data security levels.

Audit operation

Enabling Database Audit

Users of TencentDB for MariaDB can enable database audit for free. The activation portal can be found 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.
- For MariaDB instances purchased before June 5, 2016, a restart upgrade is required to support this capability. As the restart upgrade may cause a 1 to 5-second service interruption, you can contact Tencent Cloud staff to schedule an upgrade time.
- Database audit logs will be displayed in plain text, so it is recommended that you enable Two-Factor Authentication.
  There may be a few minutes of initialization time when enabling the audit feature; please be patient.

Creating an audit rule

After enabling the audit feature, logs will be automatically forwarded to the audit cluster through the MariaDB gateway cluster. However, without established audit rules and policies, logs will not be persistently recorded and displayed. Therefore, you can store logs in the audit cluster by **creating audit rules** and **associating audit policies**.

1. Navigate to the Audit Rules page and click **Create New Rule**.
2. Enter the audit rule name and click **Next**.
3. Navigate to the parameter settings page and fill in the rule parameters (at least one of the listed rule parameters must be filled in, but not all are required).

- **Conditional relationship of parameters in the rule:** AND relationship; the relationship between various parameters in the rule is AND, meaning that all parameters must meet the conditions for the rule to be successfully matched.

- **Characteristic String:** Defines the specific content of a parameter, which is the particular feature of the operation object. To achieve precise matching, users only define the keywords of the parameters they are concerned about, so the audit system only needs to record the user-defined rules, improving audit search efficiency. Note: An empty characteristic string indicates that the parameter is not a concern, i.e., "match all."
- **Match Type:** The relationship between the parameter object and its characteristics.
  - **Includes:** Indicates that a successful match occurs when the feature string appears in the network field.
  - **Does not contain:** Indicates a successful match if the feature string does not appear in the network field.
  - **Equals:** Indicates that the network field matches the characteristic string, resulting in a successful match.
  - **Not equal to:** Indicates that the network field matches successfully if it is not equal to the feature string.
  - **Regular Expression:** Represents a characteristic string and supports standard regular expression syntax.

4. All newly created rules can be viewed in the rule list.
5. Once the audit rules are set, you can modify them at any time. For similar rules, you can create new ones by **cloning rules** to improve efficiency.

Creating an audit policy

An audit policy combines audit rules, audit objects, and response methods to form a complete audit plan. Users can create multiple audit policies for a single instance. When the audit engine parses these policies, it will **match them according to the user's configured priority order from front to back**.

1. Select the **Audit Policy** page and click **Create New Policy**.
2. Complete the policy requirements, select the instances to be audited based on your needs, and choose the corresponding rules (currently, configuring alarms is not supported).
3. Adjusting Priority: For multiple policies under the same instance, the priority can be adjusted; the smaller the priority number, the higher the priority level. After adjusting the priority, it is expected to take effect within 1 minute.
4. You can modify the audit policy in real-time using the modification feature. The changes are expected to take effect within 5 minutes, and the new policy will be applied for audit monitoring. Logs prior to the audit policy modification will not be altered.

View logs

SQL statements matching the audit policy will be displayed on the audit log page, where you can directly click to view or search. Please note:

- Audit logs are displayed in plain text. It is recommended that you enable Secondary Login Authentication to ensure log control.
- Logs will be recorded starting from the creation of the audit policy, and historical data will not be logged.
- Transactions, stored procedures, and other operations may be recorded as single statements. For more information, see Supported Syntax for Database Audit.
- Currently, the maximum supported size for a single SQL statement is 1 K. Any excess will be truncated.

Syntax Supported

Last updated: 2023-08-31 09:44:46

> **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 the vast majority of SQL statements. If you find any shortcomings, please contact us to provide feedback.

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

```
Insert,Replace,Select,Union,Update,Delete,CreateDatabase:,CreateEvent,CreateFuncti
CreateTable,CreateServer,CreateProcedure,CreateTablespace,CreateTrigger,CreateVie
ShowCharset,ShowCollation,ShowColumns,ShowCreate,ShowCreateDatabase,ShowDa
ShowEvents,ShowFunction,ShowGrants,ShowLogEvents,ShowLogs,ShowProcedure,Shc
ShowProcessList,ShowMasterStatus,ShowPrivileges,ShowProfiles,ShowSlaveHosts,Sho
ShowWarnings,ShowVariables,ShowStatus,ShowTriggers,Call,DropProcedure,DropData
DropIndex,DropLogfile,DropServer,DropTables,DropTablespace,DropTrigger,DropUser,D
AlterEvent,AlterFunction,AlterLogfile,AlterProcedure,AlterServer,AlterTable,AlterTables
AlterView,Rollback,Commit,Begin,Set,SetTrans,SetPassword,Release,Grant,RenameTab
Install,StopSlave,StartSlave,StartTrans,Use,DescribeTable,DescribeStmt,Flush,Load,Loa
Reset,CacheIndex,TruncateTable,Lock,Unlock,SavePoint,Help,Do,SubQuery,ShowTables
Kill,Partition,PrepareRepairXACheckCheckSumAnalyzeChangeOptimizePurgeHandlerSi
```

- Transaction and procedures may be divided into multiple statements.