

云数据库 MariaDB 操作指南



腾讯云

【 版权声明 】

©2013–2025 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分內容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【 商标声明 】

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【 服务声明 】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。

您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【 联系我们 】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或95716。

文档目录

操作指南

使用前须知

注意事项

TXSQL MariaDB 内核与 MySQL 5.6兼容性说明

实例管理

重命名数据库实例

为实例指定项目

变更实例规格

升级数据库引擎版本

调整部署节点

重启实例

隔离恢复销毁实例

设置数据复制方式

TPC-C 体验版（物理复制版本）

灾备/只读实例

账号管理

创建账号

修改账号权限

配置读写分离

修改连接数限制

克隆账号

重置账号密码

删除账号

修改网络

备份与回档

备份方式

下载备份文件

解压备份和日志文件

通过备份文件恢复实例

回档/克隆数据库

备份归档

数据迁移

使用 mysqldump 导入数据

使用 DTS 工具导入数据到 MariaDB 实例

数据同步

安全管理

信息安全说明

访问管理

概述

策略结构

支持的资源级权限

控制台示例

当前控制台功能已接入 CAM 操作说明

配置云数据库安全组

SSL 连接加密

TDE 数据加密

监控与告警

监控功能

告警功能

KILL 线程

参数模板与设置

数据库审计

 开通数据库审计

 查看审计日志

 修改日志保存时长

操作指南

使用前须知

注意事项

最近更新时间：2024-12-16 17:42:03

● 存储引擎

TDSQL MySQL 版集中式数据库当前支持的存储引擎如下，建议采用 InnoDB 存储引擎，其他存储引擎可能导致性能变低，您可以通过 `SHOW ENGINES` 命令查看当前数据库支持的存储引擎：

存储引擎	云数据库支持引擎	未默认开启原因
InnoDB	支持（默认）	-
MyISAM	支持（需 提交工单 开启）	可能引发数据不一致，并影响同步效率
Memory	支持	-
Merge	支持	-
Archive	支持（需提交工单开启）	可能引发数据不一致，并影响同步效率
Federated	不支持	可能引发安全风险
CSV	支持（需提交工单开启）	可能引发数据不一致，并影响同步效率
BLACKHOLE	支持	-
MRG_MyISAM	支持	-
PERFORMANCE_SCHEMA	支持（需提交工单开启）	可能引发数据不一致，并影响同步效率

● 云数据库实例升级注意事项

云数据库在进行实例升级的过程中会出现1 - 30秒左右的连接闪断（升级切换时间），需要您提前做好准备，并设置好程序跟数据库的自动重连，避免因升级的闪断导致您的服务不可用。

● 文档中的主从和主备等意

云数据库文案中描述的从机和备机，均指高可用方案下的数据库热备服务器，英文表述为 Slave，即主机故障可以实时切换到从机继续提供服务。

● 需使用强同步功能时，建议购买一主二从的配置

使用“强同步”复制时，如果主库与从库自建网络中断或从库出现问题，主库也会被锁住（hang），而此时如果只有一个主库或一个从库，那么是无法做高可用方案的。因为此时仅剩单一节点服务，如果故障则直接导致部分数据完全丢失或错乱。

● 长时间开启外网、并启用弱密码可能存在安全风险

长时间开启数据库外网 IP，容易被恶意用户探测扫描，若此时又启用弱密码，例如12345678、1234abcd 等，会有极大的安全风险。

● 回档须知

- 目前 TDSQL MySQL 版集中式数据库支持数据回档，但建议您在回档前，务必备份一次现网关键数据。
- 回档是直接回档到新购按量计费实例当中。
- 删除原实例不会影响回档实例。

● 云数据库锁定策略须知

云数据库具有锁定机制，如果您的实例存储空间且超过阈值（通常为103% - 130%，暂定为110%为只读阈值），系统就会将您的实例锁定，实例被锁定以后会变成只读，建议定期检查存储空间使用情况，同时也可以控制台中磁盘空间占用短信提醒。若因为财务原因无法及时升级实例容量，可提交工单请工作人员临时开放限制1 - 3个工作日。

● 故障切换

TDSQL MySQL 版集中式数据库采用一主一从、一主二从等的高可用模式。当主库出现故障时，TDSQL 会在1秒钟之内（平均200ms）切换到从库。但切换过程中可能有（小于等于）30秒钟时间不可访问（一部分时间用于故障检测，一部分时间用于数据同步），需要您设置好程序跟数据库的自动重连，避免因切换导致您的服务不可用。切换过程对业务透明（IP 端口不改变，不需要业务干预），业务仅需确保有业务有自动重连机制即可。

- **购买云数据库后，您还需要做什么？**

购买云数据库实例后，您不需要做数据库的基础运维（例如高可用、备份、安全补丁等），但是您还需要重点关注以下事情：

- 您云数据库实例的 CPU、IOPS、空间、连接数是否足够，如果不够需要优化或者升级。
- 您云数据库实例是否存在性能问题，是否有大量的慢 SQL，SQL 语句是否需要优化，是否有多余的索引或者缺失的索引等。
- **不能更改 mysql、information_schema、performance_schema 和 sysdb 库中的任何数据**
- **无法直接通过 SQL 语句进行账号和赋权相关操作，只能通过管理控制台进行；支持常见的19种权限，少量不常见权限不支持，具体支持的权限列表如下：**
 - 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
- **TDSQL MySQL 版集中式数据库不提供 root 账号**
- **外网地址建议仅用于日常维护，不宜用于对接业务服务器**

TXSQL MariaDB 内核与 MySQL 5.6兼容性说明

最近更新时间：2025-06-19 11:07:52

TXSQL MariaDB 内核和开源 MariaDB 的兼容性

TXSQL MariaDB完全兼容开源MariaDB。

MariaDB 与 MySQL 5.6兼容性

MariaDB 内核与 MySQL 5.6高度兼容，已用于 MySQL 数据库的代码、应用程序、驱动程序和工具，无需更改（或少量调整），即可与 MariaDB 内核配合使用。

- 数据文件和表定义文件二进制兼容。
- 所有的客户端 API 和协议都兼容。
- 所有的文件名、二进制文件、路径、端口号等都是相同。
- 所有的连接器，包括 PHP、Perl、Python、Java、.NET、Ruby、MySQL 的连接器在 MariaDB 上都可以正常使用，无需进行任何改动。
- 可使用 MySQL 客户端连接到 MariaDB。

MariaDB 内核和 MySQL 5.6的不兼容性

1. GTID 不兼容

MariaDB 的 GTID 和 MySQL 5.6 的 GTID 不兼容，即 MySQL 不能作为 MariaDB 的从库。

2. Binlog 默认配置不同

MariaDB 的 Binlog 默认采用 row 格式，而原生 MySQL 5.6 和原生 MariaDB 10.2.3 之前的版本，都默认采用 statement 格式。

3. CREATE TABLE ... SELECT 命令在基于行模式复制和基于命令模式复制

为使 CREATE TABLE ... SELECT 命令在基于行模式复制和基于命令模式复制的情况下都能正常工作，MariaDB 中的 CREATE TABLE ... SELECT 命令在从库上将会被转化为 CREATE OR REPLACE 命令执行，好处是即使从库中途宕机恢复后仍然能够正常工作。

3.1 默认值推导

Create table ... Select from 语句建表时，varchar(N) 类型的字段的缺省值的区别：

- MariaDB 10.1 没有默认值。
- MySQL 5.7 的默认值是 NULL。
- MySQL 5.5、5.6 的默认值是空串 ''。

decimal 列的默认值：MySQL 5.5、5.6 把推导为0.00，MariaDB 10.1 推导为 NULL。

示例：

```
----- MySQL 5.5 -----
create table t1
select least(_latin1'a',_latin2'b',_latin5'c' collate latin5_turkish_ci) as f1;
show create table t1;
Table      Create Table
t1  CREATE TABLE `t1` (
  `f1` varchar(1) CHARACTER SET latin5 NOT NULL DEFAULT ''
) ENGINE=MyISAM DEFAULT CHARSET=latin1
----- MySQL 5.7 -----
create table t1
select least(_latin1'a',_latin2'b',_latin5'c' collate latin5_turkish_ci) as f1;
show create table t1;
Table      Create Table
t1  CREATE TABLE `t1` (
  `f1` varchar(1) CHARACTER SET latin5 DEFAULT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1
----- MariaDB 10.1* -----
```

```

create table t1
select least(_latin1'a',_latin2'b',_latin5'c' collate latin5_turkish_ci) as f1;
show create table t1;
Table      Create Table
t1  CREATE TABLE `t1` (
  `f1` varchar(1) CHARACTER SET latin5 NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1

```

3.2 处理子查询中 select 语句的区别

在这条语句中 `SELECT a AS x, ROW(11, 12) = (SELECT MAX(x), 12), ROW(11, 12) IN (SELECT MAX(x), 12) FROM t1;`

- MySQL 5.5、5.6 处理子查询 `SELECT MAX(x), 12` 时，如果该子查询位于 `in` 后面则相当于 `SELECT MAX(x), 12 from t1`，如果子查询位于 `=` 后面，则相当于 `SELECT x, 12`，其中 `x` 就是当前行中 `a` 的别名。
- MySQL 5.7 和 MariaDB 10.1.* 中，子查询 `SELECT MAX(x), 12` 都等于 `SELECT x, 12`，其中 `x` 就是当前行中 `a` 的别名。

示例：

```

----- MySQL 5.5/5.6 -----
CREATE TABLE t1 (a INT);
INSERT INTO t1 VALUES (1), (2), (11);
SELECT a AS x, ROW(11, 12) = (SELECT MAX(x), 12), ROW(11, 12) IN (SELECT MAX(x), 12) FROM t1;
x  ROW(11, 12) = (SELECT MAX(x), 12)  ROW(11, 12) IN (SELECT MAX(x), 12)
1  0  1
2  0  1
11 1  1

----- MariaDB 10.1.* or MySQL 5.7-----
CREATE TABLE t1 (a INT);
INSERT INTO t1 VALUES (1), (2), (11);
SELECT a AS x, ROW(11, 12) = (SELECT MAX(x), 12), ROW(11, 12) IN (SELECT MAX(x), 12) FROM t1;
x  ROW(11, 12) = (SELECT MAX(x), 12)  ROW(11, 12) IN (SELECT MAX(x), 12)
1  0  0
2  0  0
11 1  1

```

3.3 对 NULL 在 ALL 和 SOME 中的处理

MySQL 5.5 中对于 `10 >= ALL (NULL, 1, 10)` 或者 `1 <= ALL (NULL, 1, 10)` 的判断中，因为 NULL 不可比，所以直接跳过与 NULL，即当作该 NULL 不存在。

MySQL 5.7 和 MariaDB 中，由于 NULL 属于未知值，在上述的对比中结果也应该是未知的，所以返回 NULL。

3.4 alter table inplace 操作

如果 alter table 仅交换列的顺序，MariaDB 允许使用 inplace 算法，但 MySQL 不允许。

MariaDB 执行 inplace alter table 后，show create table t1 后发现运行结果与 MySQL 用 ALGORITHM=COPY 时运行结果相同。

4. MySQL 和 MariaDB 的未定义行为

未定义行为 (undefined behavior)：指 MySQL、MariaDB 可以按照任意方式实现这种功能和行为，且版本之间可能发生变化而不需要通知用户或者明确指出。MySQL、MariaDB 对这些行为的实现可能产生相同的结果或者不同的结果。

对于现在和未来版本的这类不同或者相同之处，MariaDB 不会做任何结果保证，也不保证提供内核优化保证完全一致。

4.1 字符类型列与大小写无关的排序

字符类型列，排序 (order by 子句) 一般按照大小写无关的方式排序，这意味着对于除大小写有区别之外，内容完全相同的字段，排序后的顺序是未定义的。您可以使用 BINARY 关键字来强制大小写相关的排序：ORDER BY BINARY 列名。

示例：

MySQL 和 MariaDB 对如下示例的排序可能是完全随机的

```
mysql> SELECT email FROM t2 LEFT JOIN t1 ON kid = t2.id WHERE t1.id IS NULL order by email;

+-----+
| email |
+-----+
| email |
| eMail |
| EMail |
+-----+
3 rows in set (0.00 sec)
```

4.2 Auto_increment 字段溢出后的处理方式不同

INNODB 特定的未定义行为:

- 在所有自增列锁定模式下 (0, 1, 2), 如果给自增列字段指定负数, 自增机制的行为是未定义的。
- 在所有自增列锁定模式下 (0, 1, 2), 如果自增列字段值大于该自增列整数类型可以存储的最大整数值, 那么自增机制的行为是未定义的。

说明:

请不要向自增列随意插入 (错误) 数字。

4.3 FOUND_ROWS 统计方式不同

FOUND_ROWS() 返回值只有在查询语句中使用了 UNION ALL 时才精确。

如果只使用 UNION 不使用 ALL, 那么 MariaDB 会去重统计, MySQL 不会去重统计。如果使用了 UNION 的查询语句不使用 LIMIT 子句那么 SQL_CALC_FOUND_ROWS 关键字会被忽略, FOUND_ROWS() 返回的就是执行 UNION 时创建的临时表中的行数。

4.4 LOCK TABLES 语句上锁顺序不同

LOCK TABLES 语句按照如下方法上锁: 第一步是把所有要锁住的表按内部定义的方式排序; 但从用户角度来看, MySQL、MariaDB 这个顺序是未定义的。例如, 写一个 LOCK TABLES t1, t2, t3, MySQL、Mariadb 不会按照 t1, t2, t3 的顺序上锁。

因为对 MySQL、MariaDB 来说, 这个是未定义行为, MySQL 和 MariaDB 可能采用不同的方式去排序 t1, t2, t3, 然后再按照那个排好的顺序依次给他们上锁。

因此, 用户写的存储过程或者查询代码不应该依赖任何上锁顺序来保持正确性, 否则可能会发生死锁。

4.5 执行 RESET MASTER 语句的时机

RESET MASTER 不可以在有任何复制备机运行期间被执行, 此时执行 RESET MASTER 时候主机和备机的行为对 MariaDB 和 MySQL 是未定义的 (也是不支持的); 执行 RESET MASTER 期间可能发生各种错误 (不具有可预见性, 甚至不会发生错误), 且这些错误 MariaDB 和 MySQL 的官方开发团队不会认为是 bug, 也不会对发生的错误负责。

4.6 日期和时间类型转换为 year 类型

MySQL 5.5 中 year 与 date 类型的变量在比较时, 会将 date 类型转为 year 类型进行对比, 即 “2011-01-01” 被转为 2011。

MySQL 5.7 和 MariaDB 中 date 类型的变量仍然为 date, 所以在和 year 对比时不相等。

类似地, MariaDB 不能够把时间类型转换为 year 类型, 而 MySQL 5.6 会使用当前 session 的 timestamp 值的 year 部分作为每一个 TIME 类型值的 year, 因此当需要把一个 time 类型的值转换为 year 类型时, 就使用这个 session 的 timestamp 的 year。

4.7 未知字符的处理方式

- 不同版本的 MySQL 和 MariaDB 在做字符编码转换时是有区别的: 例如, unhex 如果不识别一个编码字节串, 那么在 MySQL 5.5、5.6、5.7 中它返回空串 ' ', 但是 MariaDB 10.1 会返回问号字符(?)。
- 语句 `UPDATE t1 SET a=unhex(code) ORDER BY code` 对表 t1 中的 a 字段进行赋值, 但由于 unhex 只能识别和转换特定范围内的字节串, 因此部分赋值是失败的。
- MySQL 5.5 默认的存储引擎是 MyISAM, 不支持事务。当对 t1 某行中的 a 赋值失败后就会退出该语句, 而已经赋值的仍然存储在 t1 中。
- MySQL 5.7 默认的存储引擎是 InnoDB, 因此当对 t1 某行中的 a 赋值失败后这个事务就会进行回滚, 因此已经赋值都会被回滚。
- MariaDB 默认的存储引擎为 InnoDB, 且当 unhex 无法找到一个字节串对应的字符时, 就会返回问号字符, 0x3F, 即字符 '?', 因此不管存储引擎是 InnoDB 和 MyISAM, 都会是操作成功。
- 当使用 insert into 语句插入 16 进制的字节串时, 如果无法找到对应的 utf8mb4 编码的字符时,

- MySQL 5.5、5.6 使用 heap 存储引擎时，忽略此未知字符。
- MariaDB 10.1 和 MySQL 5.7 用 0x3F（即问号字符）代替。
- 对于非法编码的字符串字段，MySQL 使用 InnoDB 存储引擎时直接返回错误，而 MariaDB 则将其替换为 3F 再将其插入。

4.8 时间类型精度

```
SELECT CAST(CAST('10:11:12.098700' AS TIME) AS DECIMAL(20,6));
CAST(CAST('10:11:12.098700' AS TIME) AS DECIMAL(20,6))
...
```

出现类似语句时，MySQL 5.5、5.6 与 MariaDB 10.1、MySQL 5.7 之间采用不同的处理方式：

- 在 MySQL 5.5、5.6 中返回 101112.098700，仍然能够保持精度。
- 在 MySQL 5.7 和 MariaDB 10.1 中返回 101112.000000，这是因为该语句没有指定 TIME 的精度，而 TIME 的默认精度为 0，因此“CAST('10:11:12.098700' AS TIME)”会丢失小数点后面的数值。

为了保证时间的精度不变，可以使用如下语句。

```
SELECT CAST(CAST('10:11:12.098700' AS TIME(6)) AS DECIMAL(20,6));
+-----+
| CAST(CAST('10:11:12.098700' AS TIME(6)) AS DECIMAL(20,6)) |
+-----+
| 101112.098700 |
+-----+
```

说明：

对于 TIME 默认的精度不统一，如果对时间精度有要求，为了升级或者迁移都应该明确制定 time 的具体精度。

```
CREATE TABLE t1(f1 TIME);
INSERT INTO t1 VALUES ('23:38:57');
SELECT TIMESTAMP(f1,'1') FROM t1;
```

MySQL 5.5、5.6 返回 NULL，MariaDB 10.1 和 MySQL 5.7 返回 2016-08-03 23:38:58。

- TIMESTAMP() 的第一个参数为 time 类型时，MySQL 5.5 无法自动转换为 timestamp 类型，因而返回 NULL。
- MySQL 5.7 和 MariaDB 则将 time 类型自动转为 timestamp 类型，即将当前的日期+输入的 time 变量。

5. 附录：MariaDB 参数和 MySQL 参数

5.1 相同变量名不同参数

变量名相同意味着主要功能也相同。

参数名	MariaDB 10.1	MySQL 5.6
old_passwords	OFF	0
tmpdir	/tmp/5cXm2hHsWi/mysql.1	/data/home/tdengine/dongzhi/src/mysql-server-5.6/build_dongzhi/mysql-test/var/tmp/mysql.1
version	10.1.9-MariaDB-log	5.6.31-log
slow_query_log_file	/data/home/tdengine/dongzhi/src/tdsql-mariadb-10.1.9-release1/build_dongzhi/mysql-test/var/mysql.1/mysql-slow.log	/data/home/tdengine/dongzhi/src/mysql-server-5.6/build_dongzhi/mysql-test/var/mysql.1/mysql-slow.log
table_definition_cache	400	1400

datadir	/data/home/tdengine/dongzhi/src/tdsql-mariadb-10.1.9-release1/build_dongzhi/mysql-test/var/mysql.1/data/	/data/home/tdengine/dongzhi/src/mysql-server-5.6/build_dongzhi/mysql-test/var/mysql.1/data/
pid_file	/data/home/tdengine/dongzhi/src/tdsql-mariadb-10.1.9-release1/build_dongzhi/mysql-test/var/run/mysql.1.pid	/data/home/tdengine/dongzhi/src/mysql-server-5.6/build_dongzhi/mysql-test/var/run/mysql.1.pid
max_seeks_for_key	4294967295	18446744073709500000
slave_load_tmpdir	/tmp/5cXm2hHsWi/mysql.1	/data/home/tdengine/dongzhi/src/mysql-server-5.6/build_dongzhi/mysql-test/var/tmp/mysql.1
secure_file_priv	/data/home/tdengine/dongzhi/src/tdsql-mariadb-10.1.9-release1/build_dongzhi/mysql-test/var/	/data/home/tdengine/dongzhi/src/mysql-server-5.6/build_dongzhi/mysql-test/var/
sql_mode	NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION	NO_ENGINE_SUBSTITUTION
ssl_cert	/data/home/tdengine/dongzhi/src/tdsql-mariadb-10.1.9-release1/mysql-test/std_data/server-cert.pem	/data/home/tdengine/dongzhi/src/mysql-server-5.6/mysql-test/std_data/server-cert.pem
ssl_ca	/data/home/tdengine/dongzhi/src/tdsql-mariadb-10.1.9-release1/mysql-test/std_data/cacert.pem	/data/home/tdengine/dongzhi/src/mysql-server-5.6/mysql-test/std_data/cacert.pem
open_files_limit	1024	4161
binlog_checksum	NONE	CRC32
basedir	/data/home/tdengine/dongzhi/src/tdsql-mariadb-10.1.9-release1	/data/home/tdengine/dongzhi/src/mysql-server-5.6
query_alloc_block_size	16384	8192
innodb_max_dirty_pages_pct	75.000000	75
ssl_key	/data/home/tdengine/dongzhi/src/tdsql-mariadb-10.1.9-release1/mysql-test/std_data/server-key.pem	/data/home/tdengine/dongzhi/src/mysql-server-5.6/mysql-test/std_data/server-key.pem
myisam_sort_buffer_size	134216704	8388608
skip_name_resolve	ON	OFF
pseudo_thread_id	3	2
character_sets_dir	/data/home/tdengine/dongzhi/src/tdsql-mariadb-10.1.9-release1/sql/share/charsets/	/data/home/tdengine/dongzhi/src/mysql-server-5.6/sql/share/charsets/
innodb_adaptive_flushing_lwm	10	10
myisam_recover_options	DEFAULT	OFF
performance_schema_max_statement_classes	179	168
innodb_version	5.6.26-74.0	5.6.31
max_write_lock_count	4294967295	18446744073709500000

thread_cache_size	0	9
innodb_checksum_algorithm	INNODB	innodb
optimizer_switch	index_merge=on,index_merge_union=on,index_merge_sort_union=on,index_merge_intersection=on,index_merge_sort_intersection=off,engine_condition_pushdown=off,index_condition_pushdown=on,derived_merge=on,derived_with_keys=on,firstmatch=on,loosescan=on,materialization=on,in_to_exists=on,semijoin=on,partial_match_rowid_merge=on,partial_match_tablescan=on,subquery_cache=on,mrr=off,mrr_cost_based=off,mrr_sort_keys=off,outer_join_with_cache=on,semijoin_with_cache=on,join_cache_incremental=on,join_cache_hashed=on,join_cache_bka=on,optimize_join_buffer_size=off,table_elimination=on,extended_keys=on,exists_to_in=on	index_merge=on,index_merge_union=on,index_merge_sort_union=on,index_merge_intersection=on,engine_condition_pushdown=on,index_condition_pushdown=on,mrr=on,mrr_cost_based=on,block_nested_loop=on,batched_key_access=off,materialization=on,semijoin=on,loosescan=on,firstmatch=on,subquery_materialization_cost_based=on,use_index_extensions=on
timestamp	1471938276	1471937901
general_log_file	/data/home/tdengine/dongzhi/src/tdsql-mariadb-10.1.9-release1/build_dongzhi/mysql-test/var/mysql.1/mysql.1.log	/data/home/tdengine/dongzhi/src/mysql-server-5.6/build_dongzhi/mysql-test/var/mysql.1/mysql.1.log
myisam_stats_method	NULLS_UNEQUAL	nulls_unequal
innodb_log_compressed_pages	OFF	ON
query_prealloc_size	24576	0
rand_seed2	297895171	0
rand_seed1	605568929	0
socket	/tmp/5cXm2hHsWi/mysql.1.sock	/data/home/tdengine/dongzhi/src/mysql-server-5.6/build_dongzhi/mysql-test/var/tmp/mysql.1.sock
innodb_max_dirty_pages_pct_lwm	0.001	0
lc_messages_dir	/data/home/tdengine/dongzhi/src/tdsql-mariadb-10.1.9-release1/build_dongzhi/sql/share/	/data/home/tdengine/dongzhi/src/mysql-server-5.6/build_dongzhi/sql/share/
max_relay_log_size	1073741824	0
plugin_dir	/data/home/tdengine/dongzhi/src/tdsql-mariadb-10.1.9-release1/lib/plugin/	/data/home/tdengine/dongzhi/src/mysql-server-5.6/lib/plugin/
thread_stack	294912	262144

5.2 仅存在于 MariaDB 中的变量

- aria_block_size 8192
- aria_checkpoint_interval 30
- aria_checkpoint_log_activity 1048576
- aria_encrypt_tables OFF
- aria_force_start_after_recovery_failures 0

- aria_group_commit none
- aria_group_commit_interval 0
- aria_log_file_size 1073741824
- aria_log_purge_type immediate
- aria_max_sort_file_size 9223372036853727232
- aria_page_checksum ON
- aria_pagecache_age_threshold 300
- aria_pagecache_buffer_size 134217728
- aria_pagecache_division_limit 100
- aria_pagecache_file_hash_size 512
- aria_recover NORMAL
- aria_repair_threads 1
- aria_sort_buffer_size 268434432
- aria_stats_method nulls_unequal
- aria_sync_log_dir NEWFILE
- aria_used_for_temp_tables ON
- autoremoverelaylog ON
- binlog_annotate_row_events OFF
- binlog_commit_wait_count 0
- binlog_commit_wait_usec 100000
- binlog_optimize_thread_scheduling ON
- deadlock_search_depth_long 15
- deadlock_search_depth_short 4
- deadlock_timeout_long 50000000
- deadlock_timeout_short 10000
- debug_no_thread_alarm OFF
- default_master_connection
- default_regex_flags
- encrypt_binlog OFF
- encrypt_tmp_disk_tables OFF
- encrypt_tmp_files OFF
- enforce_storage_engine
- expensive_subquery_limit 100
- extra_max_connections 20
- extra_port 0
- flush_relay_logs_for_strong_consistency ON
- gtid_binlog_pos
- gtid_binlog_state
- gtid_current_pos
- gtid_domain_id 0
- gtid_ignore_duplicates OFF
- gtid_seq_no 0
- gtid_slave_pos
- gtid_strict_mode OFF
- histogram_size 0
- histogram_type SINGLE_PREC_HB
- in_transaction 0

- innodb_adaptive_hash_index_partitions 1
- innodb_background_scrub_data_check_interval 3600
- innodb_background_scrub_data_compressed OFF
- innodb_background_scrub_data_interval 604800
- innodb_background_scrub_data_uncompressed OFF
- innodb_buf_dump_status_frequency 0
- innodb_buffer_pool_populate OFF
- innodb_cleaner_lsn_age_factor HIGH_CHECKPOINT
- innodb_compression_algorithm none
- innodb_corrupt_table_action assert
- innodb_default_encryption_key_id 1
- innodb_defragment OFF
- innodb_defragment_fill_factor 0.900000
- innodb_defragment_fill_factor_n_recs 20
- innodb_defragment_frequency 40
- innodb_defragment_n_pages 7
- innodb_defragment_stats_accuracy 0
- innodb_disallow_writes OFF
- innodb_empty_free_list_algorithm BACKOFF
- innodb_encrypt_log OFF
- innodb_encrypt_tables OFF
- innodb_encryption_rotate_key_age 1
- innodb_encryption_rotation_iops 100
- innodb_encryption_threads 0
- innodb_fake_changes OFF
- innodb_fatal_semaphore_wait_threshold 600
- innodb_force_primary_key OFF
- innodb_foreground_preflush EXPONENTIAL_BACKOFF
- innodb_idle_flush_pct 100
- innodb_immediate_scrub_data_uncompressed OFF
- innodb_instrument_semaphores OFF
- innodb_kill_idle_transaction 0
- innodb_locking_fake_changes ON
- innodb_log_arch_dir ./
- innodb_log_arch_expire_sec 0
- innodb_log_archive OFF
- innodb_log_block_size 512
- innodb_log_checksum_algorithm INNODB
- innodb_max_bitmap_file_size 104857600
- innodb_max_changed_pages 1000000
- innodb_mtf_flush_threads 8
- innodb_prefix_index_cluster_optimization OFF
- innodb_sched_priority_cleaner 19
- innodb_scrub_log OFF
- innodb_scrub_log_speed 256
- innodb_show_locks_held 10
- innodb_show_verbose_locks 0

- innodb_simulate_comp_failures 0
- innodb_stats_modified_counter 0
- innodb_stats_traditional ON
- innodb_track_changed_pages OFF
- innodb_use_atomic_writes OFF
- innodb_use_fallocate OFF
- innodb_use_global_flush_log_at_trx_commit ON
- innodb_use_mtf flush OFF
- innodb_use_stacktrace OFF
- innodb_use_trim OFF
- join_buffer_space_limit 2097152
- join_cache_level 2
- key_cache_file_hash_size 512
- key_cache_segments 0
- last_gtid
- log_slow_filter
admin,filesort,filesort_on_disk,full_join,full_scan,query_cache,query_cache_miss,tmp_table,tmp_table_on_disk
- log_slow_rate_limit 1
- log_slow_verbosity
- log_tc_size 24576
- loglevel 3
- max_long_data_size 4194304
- max_statement_time 0.000000
- mrr_buffer_size 262144
- myisam_block_size 1024
- mysql56_temporal_format ON
- old_mode
- optimizer_selectivity_sampling_limit 100
- optimizer_use_condition_selectivity 1
- plugin_maturity unknown
- progress_report_time 5
- query_cache_strip_comments OFF
- relay_log_sync_threshold 134217728
- relay_log_sync_timeout 200
- relay_log_sync_txn_count 5
- replicate_annotate_row_events OFF
- replicate_do_db
- replicate_do_table
- replicate_events_marked_for_skip REPLICATE
- replicate_ignore_db
- replicate_ignore_table
- replicate_wild_do_table
- replicate_wild_ignore_table
- rowid_merge_buff_size 8388608
- rpl_semi_sync_master_enabled OFF
- rpl_semi_sync_master_timeout 10000
- rpl_semi_sync_master_trace_level 32

- rpl_semi_sync_master_wait_no_slave ON
- rpl_semi_sync_master_wait_point AFTER_COMMIT
- rpl_semi_sync_slave_enabled OFF
- rpl_semi_sync_slave_trace_level 32
- skip_parallel_replication OFF
- skip_replication OFF
- slave_current_parallel_transactions 0
- slave_ddl_exec_mode IDEMPOTENT
- slave_domain_parallel_threads 0
- slave_max_parallel_transactions 0
- slave_parallel_max_queued 131072
- slave_parallel_mode conservative
- slave_parallel_threads 0
- slave_run_triggers_for_rbr NO
- sqlasyn OFF
- sqlasyntimeout 10
- sqlasynwarntimeout 3
- strict_password_validation ON
- thread_pool_high_prio_mode transactions
- thread_pool_high_prio_tickets 4294967295
- thread_pool_idle_timeout 60
- thread_pool_max_threads 1000
- thread_pool_oversubscribe 3
- thread_pool_oversubscribe_parall 1
- thread_pool_size 8
- thread_pool_stall_limit 500
- use_stat_tables NEVER
- userstat OFF
- version_malloc_library system
- version_ssl_library OpenSSL 1.0.2d 9 Jul 2015
- wsrep_auto_increment_control ON
- wsrep_causal_reads OFF
- wsrep_certify_nonpk ON
- wsrep_cluster_address
- wsrep_cluster_name my_wsrep_cluster
- wsrep_convert_lock_to_trx OFF
- wsrep_data_home_dir /data/home/tdengine/dongzhi/src/tdsql-mariadb-10.1.9-release1/build_dongzhi/mysql-test/var/mysqld.1/data/
- wsrep_debug_option
- wsrep_debug OFF
- wsrep_desync OFF
- wsrep_dirty_reads OFF
- wsrep_drupal_282555_workaround OFF
- wsrep_forced_binlog_format NONE
- wsrep_gtid_domain_id 0
- wsrep_gtid_mode OFF
- wsrep_load_data_splitting ON

- wsrep_log_conflicts OFF
- wsrep_max_ws_rows 131072
- wsrep_max_ws_size 1073741824
- wsrep_mysql_replication_bundle 0
- wsrep_node_address
- wsrep_node_incoming_address AUTO
- wsrep_node_name
- wsrep_notify_cmd
- wsrep_on OFF
- wsrep_osu_method TOI
- wsrep_patch_version wsrep_25.11
- wsrep_provider none
- wsrep_provider_options
- wsrep_recover OFF
- wsrep_replicate_myisam OFF
- wsrep_restart_slave OFF
- wsrep_retry_autocommit 1
- wsrep_slave_fk_checks ON
- wsrep_slave_threads 1
- wsrep_slave_uk_checks OFF
- wsrep_sst_auth
- wsrep_sst_donor
- wsrep_sst_donor_rejects_queries OFF
- wsrep_sst_method rsync
- wsrep_sst_receive_address AUTO
- wsrep_start_position 00000000-0000-0000-0000-000000000000:-1
- wsrep_sync_wait 0

5.3 仅存在于 MySQL 5.6 中的变量

- avoid_temporal_upgrade OFF
- bind_address *
- binlog_error_action IGNORE_ERROR
- binlog_gtid_simple_recovery OFF
- binlog_max_flush_queue_time 0
- binlog_order_commits ON
- binlog_rows_query_log_events OFF
- binlogging_impossible_mode IGNORE_ERROR
- block_encryption_mode aes-128-ecb
- core_file ON
- disconnect_on_expired_password ON
- end_markers_in_json OFF
- enforce_gtid_consistency OFF
- eq_range_index_dive_limit 1
- gtid_executed
- gtid_mode OFF
- gtid_next AUTOMATIC
- gtid_owned

- gtid_purged
- innodb_tmpdir
- log_bin_use_v1_row_events OFF
- log_slow_admin_statements OFF
- log_slow_slave_statements OFF
- log_throttle_queries_not_using_indexes 0
- master_info_repository FILE
- new OFF
- optimizer_trace enabled=off,one_line=off
- optimizer_trace_features greedy_search=on,range_optimizer=on,dynamic_range=on,repeated_subselect=on
- optimizer_trace_limit 1
- optimizer_trace_max_mem_size 16384
- optimizer_trace_offset -1
- relay_log_info_repository FILE
- rpl_stop_slave_timeout 31536000
- server_id_bits 32
- server_uuid 9078a55d-6904-11e6-bfa9-ecf4bbcdc829
- sha256_password_private_key_path private_key.pem
- sha256_password_public_key_path public_key.pem
- show_old_temporals OFF
- simplified_binlog_gtid_recovery OFF
- slave_allow_batching OFF
- slave_checkpoint_group 512
- slave_checkpoint_period 300
- slave_parallel_workers 0
- slave_pending_jobs_size_max 16777216
- slave_rows_search_algorithms TABLE_SCAN,INDEX_SCAN
- table_open_cache_instances 1
- transaction_allow_batching OFF

实例管理

重命名数据库实例

最近更新時間：2024-12-16 17:42:03

操作場景

本文為您介紹通過控制台重命名 TDSQL MySQL 版集中式數據庫實例的操作。

❗ 說明：

- 重命名實例名稱，不會改變數據庫的內網 IP，也不會影響數據庫連接。
- 重命名實例名稱後，數據庫實例所屬項目和所屬網絡保持不變。
- 實例狀態處於其他任務流程中時（例如，升級中、初始化中等），無法重命名實例。

操作步驟

1. 登錄 [TDSQL 控制台](#)，在實例列表，單擊實例名的  圖標。或單擊實例名進入實例詳情頁，單擊實例名後的  圖標。
2. 在彈出的對話框，修改實例名稱，單擊**確定**。

❗ 說明：

重命名實例名稱時，不能使用現有的數據庫實例名稱。

修改實例名稱

實例名稱 * 

僅支持長度小於等於60的中文/英文/數字/"_" "."

確定

取消

为实例指定项目

最近更新时间：2024-12-30 14:54:32

TDSQL MySQL 版集中式数据库支持将实例分配至不同的项目进行管理。

项目是腾讯云为团队定义的一种资源划分方式，简单来说，您可以根据您的组织架构，为不同的团队分配不同的资源，而这样的划分方式在腾讯云中称为项目。

- 只读实例、灾备实例属于主实例的关联实例，需要与主实例处于同一项目中。
- 数据库实例在项目间进行分配和移动，不会影响实例对外提供的服务。
- 用户须在新购实例时为实例指定所属的项目，缺省为默认项目。
- 已指定项目的实例可通过单击 **控制台** 右上角的**转至其他项目**功能，或者在实例列表中单击**实例 ID** 进入实例详情页，单击**基本信息 > 所属项目**后的

编辑按钮，重新分配到其他项目。



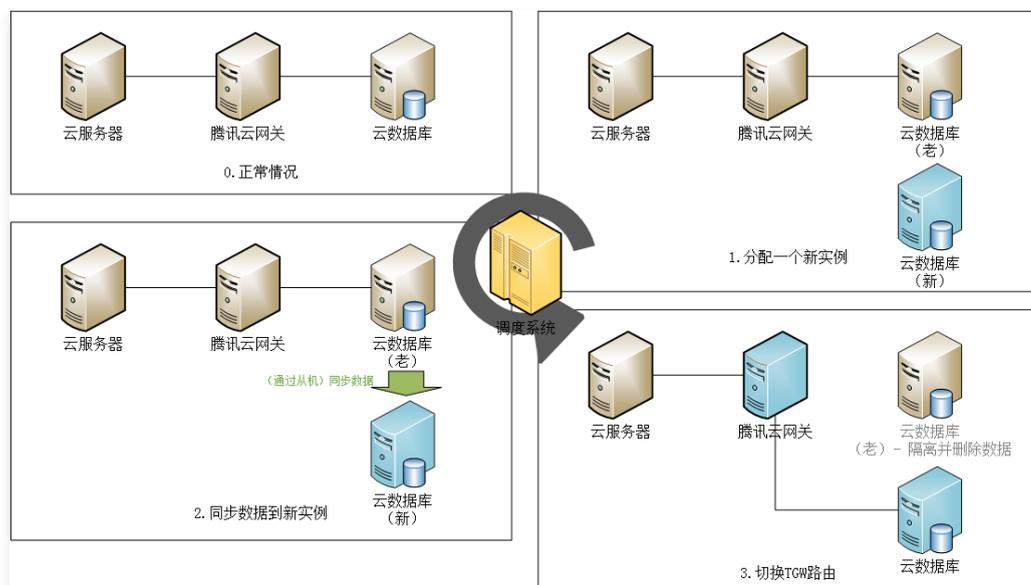
变更实例规格

最近更新时间：2024-12-16 17:42:03

配置变更原理

在控制台单击调整配置后，云运维管理系统通常按如下步骤执行：

1. 按需要变更的配置分配一个新的实例（简称“新实例”）。
2. 将需要调整配置的实例（简称“老实例”）的数据、配置等同步到新实例。
3. 同步完成后，在腾讯云网关做路由切换，切换到新实例继续使用。



变更实例规格

说明：

- 实例变更过程中，原实例的正常使用不受影响，如数据的导入、导出功能。
- 变配前后实例的名称、访问 IP、访问端口均不发生变化。
- 变配完成时会产生秒级的数据库连接断开，建议程序有自动重连功能。
- 在变配过程中，请尽量避免修改数据库的全局参数、实例名称、用户密码等操作。

1. 登录 [TDSQL 控制台](#)，在实例列表，选择操作列的更多 > 调整配置。
2. 在实例调整配置页面，选择所需规格、磁盘容量和指定时间切换，单击调整配置。

TDSQL MySQL 版集中式数据库提供变配定时切换能力，支持您在指定时间（通常是业务低谷）将数据库切换到新配置。

注意：

定时切换：支持您在指定时间开始（通常是业务低谷）将数据库切换到新配置，可配置的开始切换时间在72小时内。

- 切换时间通常有15分钟左右误差，因为切换过程中可能存在大事务大量写入请求，进而影响数据同步进度，此时系统会优先保障新老实例同步。
- 为保证切换的成功，可勾选失败后重试选项，将在切换失败的两小时后重试一次。

实例名称

到期时间

网络

实例版本 标准版 (一主一从)

当前规格 高IO版 - 1核, 2GB内存, 10GB存储空间, 2100次/秒

调整规格

磁盘容量 GB (步长为10GB)

适用 日独立用户数上百人的小型应用

备份空间 免费赠送购买实例容量的100% [详细说明](#)

调整配置耗时 预计3小时0.100分钟 实例负载高或大量数据写入时我们将延长调整配置时间, 优先保障实例稳定运行

指定时间切换

配置费用 元 (实例 [计费详情](#))

请您适当调整磁盘容量和内存容量

变配费用

用户自助升级数据库实例时，系统将计算实例规格间的差价，并从用户的账户中扣除差价，若账户余额不足则需要先进行充值。升级后将按照新的实例规格进行计费。

升级数据库引擎版本

最近更新时间：2024-12-16 17:42:03

升级数据库引擎大版本

升级数据库引擎大版本指从 TSQL 5.7升级到8.0等。

目前暂不支持大版本升级。如有需要，建议购买一个新版本实例，测试完成后，将老实例数据迁移至新实例，并切换业务系统。

升级数据库集群模块版本

例如需升级集群中的 Agent、Proxy 等功能模块到最新版本，不涉及业务系统兼容问题。如有需要，请 [提交工单](#) 并描述清楚“实例所在地域、实例 ID、升级需求、指定切换时间”后，由腾讯云工作人员处理。此处升级原理与升级实例规格相同，请参见 [升级原理](#)。

调整部署节点

最近更新时间：2024-12-16 17:42:03

本文将为您介绍如何通过控制台调整 TDSQL MySQL 版集中式数据库部署节点。您可以通过增加从节点获得多地备机支持，减缓执行压力并提高读取速度，也可以通过删除不必要的从节点，节约低耗时冗余性能开支。

说明：

- 实例调整过程中，原实例的正常使用不受影响。
- 调整前后实例的名称、访问 IP、访问端口均不发生变化，但 SQL 透传 ID (Setid) 会发生变更，请提前处理。
- 调整完成时会产生秒级的数据库连接断开，建议程序有自动重连功能。
- 在调整过程中，请尽量避免修改数据库的全局参数、实例名称、用户密码等操作。

调整节点部署地域

- 登录 [TDSQL 控制台](#)，在实例列表中，单击实例 ID，进入实例详情页。
- 在实例详情页的可用性信息 > 部署方式处，单击调整部署方式。

可用性信息 [主从切换](#)

数据复制方式 强同步(可退化)

当前复制方式 同步

部署方式 单可用区 [调整部署方式](#)

主库可用区 广州二区

从库可用区 广州二区, 广州二区

- 在调整部署方式的页面中，选择目标部署方式，下拉选择主节点和从节点的地域。

说明：

目标部署方式：可选择单可用区和多可用区，单可用区中从节点和主节点地区保持一致，多可用区中从节点可选择任意地区。

实例名称

运行状态 运行中

网络

内网地址

原部署方式 单可用区 (主库可用区: 广州二区, 从库可用区: 广州二区, 广州二区)

原实例版本 标准版 (一主二从)

目标部署方式

主节点

从节点

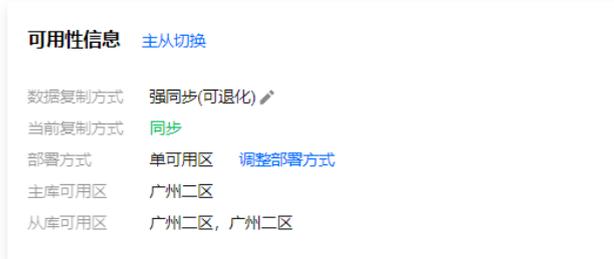
调整部署方式耗时 预计3小时20分钟 实例负载高或大量数据写入时我们将延长调整时间，优先保障实例稳定运行

指定时间切换

配置费用 (实例 [计费详情](#))

增减从节点

1. 登录 [TDSQL 控制台](#)，在实例列表中，单击实例 ID，进入实例详情页。
2. 在实例详情页的可用性信息 > 部署方式处，单击调整部署方式。



3. 在调整部署方式的页面中，单击添加从节点，可增加一个新的备机支持，最多增加五个从节点。

说明：

- **删除从节点：**单击后可删除已有从节点，单从节点不可删除。
- **定时切换：**支持您在指定时间开始（通常是业务低谷）将数据库切换到新配置，可配置的开始切换时间在72小时内。
- 切换时间通常有15分钟左右误差，因为切换过程中可能存在大事务大量写入请求，进而影响数据同步进度，此时系统会优先保障新老实例同步。
- 为保证切换的成功，可勾选失败后重试选项，将在切换失败的两小时后重试一次。



重启实例

最近更新时间：2024-12-16 17:42:03

本文为您介绍通过控制台重启实例的操作。

操作场景

重启实例是云数据库一种常用的维护方式，重启实例相当于本地数据库的重启操作。

注意事项

- **重启准备：**重启期间实例无法正常提供服务，因此在重启之前，请确保 TDSQL MySQL 版集中式数据库已暂停业务请求。重启期间，如果业务写入量大，会产生很多脏页，为减少业务不可用时长，可能会导致重启失败。
- **重启操作方式：**建议使用腾讯云提供的重启操作进行实例重启，而非在实例中运行重启命令。
- **重启时间：**重启操作一般只需要几分钟时间。
- **实例物理特性：**重启实例不改变实例的物理特性，实例的内网 IP 不会改变。

操作步骤

1. 登录 [TDSQL 控制台](#)，在实例列表勾选单个或多个需要重启的实例，在上方单击**重启**。



2. 在弹出的对话框，确认无误后，单击**确定**，即可重启单个实例或批量重启实例。

隔离恢复销毁实例

最近更新时间：2024-12-30 14:54:32

隔离实例

当您不再使用实例时，可通过隔离实例即让实例无法使用（但并非销毁或删除）。实例隔离后将不可被访问，并且实例将放置于回收站中，您可以在回收站中恢复实例，也可在回收站中将此实例彻底销毁。隔离后资源空间不会被释放且保留最基本的数据副本。隔离到期后，实例彻底销毁。

- 按量计费实例：可在 [控制台](#) 选择**销毁/退货**功能进行手动退还。实例退还后，状态变为**已隔离**，保留时效为**3天**，期间实例无法访问。如您想恢复该实例，可在回收站中进行恢复。
- 包年包月实例：可在 [控制台](#) 选择**销毁/退货**功能进行手动退还。实例退还后，状态变为**已隔离**，保留时效为**7天**，期间实例无法访问。如您想恢复该实例，可在回收站进行恢复。
- 独享集群实例：可在 [控制台](#) 选择**销毁/退货**功能进行手动退还。实例退还后，状态变为**已隔离**，保留时效为**3天**，期间实例无法访问。如您想恢复该实例，可在回收站中进行恢复。

退还后，实例的状态一旦变为**已隔离**时，就不再产生与该实例相关的费用。

注意：

- 隔离后，实例 IP 被释放，再次恢复可能无法获得原有 IP。
- 隔离后，实例无法进行升级、修改参数、创建修改账号、回档、修改实例名等修改操作。

操作步骤

1. 登录 [TDSQL 控制台](#)，在实例列表选择实例，在上方选择**更多操作 > 销毁/退货**。
2. 在弹出的对话框，勾选同意，单击**确定**。

销毁实例

您已选共 1 个实例。 [查看详情](#)

实例彻底销毁后**数据将无法找回**，请提前备份实例数据。

实例彻底销毁后IP资源同时释放，如果该实例有相关的灾备实例：

- 灾备实例将会断开同步连接，自动升级为主实例

实例彻底销毁后，退款处理：

- 5天无理由自助退还的金额将退还至原支付账户
- 普通自助退还的金额将按购买支付使用的现金和赠送金支付比例退还至您的腾讯云帐号
- 推广奖励渠道订单退款将收取订单实际现金支付金额的25%作为退款手续费
- 推广奖励渠道订单暂不支持自助退款，请提交工单发起退款申请

已阅读并同意 [销毁规则](#)

确定 **取消**

进入回收站，可发现实例状态变为**已隔离**。

恢复实例

恢复实例是在实例被隔离后恢复实例至正常运行的操作。恢复可能需要几分钟时间，另外，恢复实例可能会重新分配 IP，而非隔离前的 IP。

操作步骤

1. 登录 [TDSQL 控制台](#)，在实例列表选择实例，单击**恢复/开机**。
2. 在弹出的对话框，单击**确定**。

销毁实例

当您不需要某个实例时，可以对实例进行退还，实例退还后，状态变为“已隔离”。隔离中的实例到期后会彻底销毁。同时也可以进入回收站中执行**立即下线**将此实例彻底删除。

注意事项

- 实例彻底销毁后数据将无法找回，请提前备份实例数据。
- 实例彻底销毁后 IP 资源将同时释放，如果该实例有相关的灾备实例，灾备实例将会断开同步连接，自动升级为主实例。
- 实例彻底销毁后，退款处理：
 - 5天无理由自助退还的金额将退还至腾讯云账户。
 - 普通自助退还的金额将按购买支付使用的现金和赠送金支付比例退还至您的腾讯云账户。
 - 推广奖励渠道订单退款将收取订单实际现金支付金额的25%作为退款手续费。推广奖励渠道订单暂不支持自助退款，请 [提交工单](#) 发起退款申请。

设置数据复制方式

最近更新时间：2024-12-16 17:42:03

TDSQL MySQL 版集中式数据库提供了三种数据复制方式（强同步、强同步（可退化）、异步），用户可登录 [TDSQL 控制台](#)，进入实例管理页的实例详情进行查看。

数据复制

当前存在三种数据复制方式：

- 强同步（可退化）**：默认选择的复制方式，正常运行下，主备将采用 **强同步 MAR** 方式进行复制。当无备节点存活或备机延迟大于25s时，复制方式将退化为异步。

说明：

强同步（可退化）：

- 由于机器故障、网络、备机高负载等原因，若所有备节点 IODelay 延迟大于25s，此时强同步会退化为异步。触发退化后，若有一个备节点 IODelay 恢复，并且满足退化结束条件，异步变回强同步。
- 在发生 HA 切换时，如果切换前先出现了 IODelay，就会先退化为异步，再发生 HA 切换，此时 HA 切换可能会导致最后几个事务数据丢失。
- 一主一备实例，数据复制方式只能选择异步或强同步（可退化）。
- 退化结束条件：当 IODelay 延迟恢复到小于15s时，退化结束。一天之内，如果实例发生过两次退化，当第三次退化时，8小时内实例不再恢复为强同步复制，将以异步方式运行。8小时后，再判断是否结束退化，恢复为强同步。

- 强同步**：仅支持有两个及以上备节点的实例选择，主备始终采用强同步方式。因为存在多个备节点，单个备节点延迟或故障时，不会出现退化。

操作步骤

- 登录 [TDSQL 控制台](#)，在实例列表中，单击实例 ID，进入实例详情页。
- 在实例详情页的 **可用性信息 > 数据复制方式**处，可以看到实例当前的复制方式，单击编辑图标。



- 在弹出的页面中，选择新的复制方式（一主一备的实例只可选择强同步（可退化）和异步），并单击**确定**。



TPC-C 体验版（物理复制版本）

最近更新时间：2024-12-16 17:42:03

本文将介绍 TDSQL MySQL 版集中式数据库的 TPC-C 体验版（物理复制版本）的特性、性能及使用方式。该版本是 TDSQL 通过 TPC-C 基准测试的版本，在高 IO 并发性能上有较大的提升。

说明：

- 由 TPC 发起的 TPC-C 是全球数据库认可的一套性能评价标准，它模拟超大型高并发的极值场景，对数据库系统的软硬件协同能力要求极高。
- 2023年3月，腾讯云数据库 TDSQL 实现每分钟交易量8.14亿次，同时在超高压下稳定运行8小时，波动率为0.2%，成功打破纪录。
- TPC 官网链接：[官网链接](#)。

物理复制版本特性

- 支持物理复制。物理复制是将主库 WAL 日志流直接发给备库，备库根据 WAL 日志进行重做的一种复制方式，这种复制和 MySQL 原生复制有明显区别，复制期间不会产生 binlog 文件。

复制类型	特性
物理复制	<ul style="list-style-type: none">以 WAL 日志流传输，不涉及 binlog 和 gtid，减少了数据写盘量，实例整体的吞吐量和响应时间都得到了提升。拥有更好的并发性，redolog 的同步可以支持同时在主备库上执行事务，而无需等待主库上执行完成再执行备库。
原生复制	生成 binlog 进行同步复制，binlog 是较为统一的日志格式，有成熟的工具来进行解析，通过 binlog 可以提供回档、迁移、同步等能力，并使主备之间的复制方式更加多样化。

- 不支持创建灾备关系。
- 不支持 binlog 查看和备份克隆。
- 不支持全局一致性读。
- 不支持一级 list/range 和二级分区表。
- 不支持 SET 级全局索引。
- 不支持 TDE 透明加密。
- 不支持作为源数据库或目标数据库创建 DTS 数据迁移和数据同步任务。
- 不支持 DBS 数据库备份。

物理复制版本性能

经过测试，在高 IO 场景下，物理复制版本的性能有显著提升。

系统架构：x86

测试工具	sysbench 1.1.0			
实例类型	集中式实例，100G内存，1主1备，强同步，64张表，100w行/张，16G数据（全缓存）			
测试版本	TPCC 体验版		8.0.24版本	
测试开发	500个		500个	
导数耗时	1min58.9s		2min	
结果指标	TPS	QPS	TPS	QPS
点查	438420	438420	425332	425332
混合读写	20212	323399	15945	255129
索引更新	144874	144874	98494	98494
非索引更新	156298	156298	110851	110851

测试工具	TPCC	
实例类型	集中式实例，100G内存，1主1备，强同步	
测试版本	TPCC 体验版	8.0.24版本
测试并发	1000个	1000个
导数耗时	1min57.1s	2min
仓数	100仓（9G数据）	100仓（9G数据）
tpmC（NewOrders）	328433	265743
tpmTOTAL	729932	590454
Transaction Count	21898587	17714473

使用物理复制版本

1. 登录 [TDSQL 控制台](#)，单击**新建**创建新实例。
2. 在**数据库版本**选择 TPCC 体验。

3. 购买实例并体验使用。

说明：
当前 TPCC 体验版由白名单控制开放，如需体验请 [提交工单](#) 开通。

灾备/只读实例

最近更新时间：2024-12-30 14:54:32

本文将为您介绍如何通过控制台创建、管理灾备/只读实例。

操作场景

针对业务连续服务和数据可靠性有强需求或是监管需要的场景，TDSQL MySQL 版集中式数据库提供跨可用区、跨地域灾备/只读实例，帮助用户以较低的成本提升业务连续服务的能力，同时提升数据的可靠性。

说明：

灾备/只读实例费用与主实例相同，请参见 [产品定价](#)。

适用场景

- 异地灾备：灾备实例可用于对业务和数据进行多地备份，来保证数据的安全性。当一个可用区/地域发生故障，可以迅速切换到跨可用区、跨地域的灾备实例，尽可能减少故障对业务的影响。
- 就近接入：业务在一个可用区/地域作为主实例写入，另外的可用区/地域作为只读实例，为用户提供就近接入、异地读能力，改善访问速度。
- 多地域部署：TDSQL 数据库提供多地域部署能力，当一个可用区/地域遇到网络波动或者不可用的情况，能根据业务情况手动切换到另一个可用区/地域上。

功能特点

- 灾备/只读实例目前是同一种实例类型，区别于实例内同地域主从节点，如需考虑增减实例节点，可参考 [调整部署节点](#)。
- 提供独立的数据库连接地址，灾备/只读实例可提供读访问能力，用于就近接入、数据分析等场景，设备冗余成本低。
- 一个主实例可以创建多个灾备/只读实例，部署在不同的地域、可用区。
- 灾备/只读实例支持一主一从、一主两从高可用架构，避免了数据库的单一风险。
- 如果主实例发生故障，可在数秒内激活灾备/只读实例，恢复完整读写功能。
- 灾备/只读实例通过内网专线同步，具有较低的同步时延和更高的稳定性，同步链路质量远优于公网网络。
- 目前推广期专线流量费用免费，商业化收费时间将另行通知。

功能限制

- 灾备/只读实例暂不支持：参数设置、账号管理功能。
- 灾备/只读实例数据库版本默认保持与主实例相同，实例规格、硬盘大小需要大于等于主实例。

操作步骤

创建灾备/只读实例

1. 登录 [TDSQL 控制台](#)，在实例列表，单击实例 ID，进入实例管理页面。

2. 在实例详情页的实例架构图中，单击**添加灾备/只读实例**，进入实例购买页。



3. 在购买页中，设置灾备/只读实例的计费模式、地域等基本信息，单击**立即购买**。

说明:

- 创建时长受数据量的影响，期间主实例的控制台操作会被锁定，请妥善安排。
- 暂只支持整个实例数据同步，请确保磁盘空间充足。
- 请确保主实例状态为运行中并且没有任何任务执行，否则同步任务有可能失败。

4. 支付完成后，返回实例列表，初始化实例后，即可进行后续操作。

管理灾备/只读实例

查看灾备/只读实例

灾备/只读实例可在其所在地域查看，可在实例列表筛选出该地域全部灾备/只读实例。

实例 ID / 名称	运行状态	所属地域	实例类型	所属项目
[实例ID]	运行中	华北地区 (北京)	<input type="checkbox"/> 全部 <input type="checkbox"/> 主实例 <input type="checkbox"/> 灾备 / 只读实例 <input type="checkbox"/> 独享实例	默认项目

查看从属关系

在实例详情页的实例架构图中，可查看从属关系。



灾备/只读实例功能

灾备/只读实例提供实例详情、监控告警、参数配置、数据安全性、备份与恢复、性能优化功能。

实例详情 监控告警 参数配置 数据安全性 备份与恢复 性能优化

基本信息

实例名称

实例 ID

运行状态 运行中

实例类型 灾备 / 只读实例

实例架构图

华南地区 (广州)

灾备/只读实例升级为主实例

当您需要切换灾备/只读实例为主实例时，可在控制台主动切换灾备/只读实例为主实例。

1. 登录 **TDSQL 控制台**，在实例列表，选择所需灾备/只读实例，单击实例 ID，进入实例管理页面。
2. 在实例管理页面，单击右上角的**与主实例切换**，即可将灾备/只读实例升级为主实例。切换后将断开与原主实例的同步连接，恢复实例数据库数据写入能力和完整的 TDSQL MySQL 版集中式数据库功能。

注意：
同步连接断开后不可重连，请谨慎操作。

← 登录 销毁/退货 一键诊断 new 解除灾备同步 与主实例切换 重启 异常告警 1 刷新

实例详情 监控告警 参数配置 数据安全性 备份与恢复 性能优化

基本信息

实例名称

实例 ID

运行状态 运行中

实例类型 灾备 / 只读实例

实例架构图

华南地区 (广州) 灾备 / 只读文档 [↗](#)

账号管理

创建账号

最近更新时间：2024-12-30 14:54:32

操作场景

本文为您介绍通过控制台创建 TDSQL MySQL 版集中式数据库账号，用以管理连接数据库实例。

注意：

只有主实例可以创建账号，灾备 / 只读实例不可创建账号。

操作步骤

1. 登录 [TDSQL 控制台](#)，在实例列表中，单击实例 ID 或操作列的管理，进入实例管理页面。
2. 在实例管理页面，选择账号管理页，单击创建账号。



3. 在弹出的对话框，输入账号名、主机、密码等，确认无误后，单击确认，下一步。

- 账号名：需要1个 - 32个字符，由字母、数字或特殊字符组成；以字母开头。
- 主机：IP 形式，支持填入%。
- 密码：需要8个 - 32个字符，不能以 '/' 开头，且必须包含小写字母、大写字母、数字和符号 `()~!@#%&^*+={}|[]:;<>,./?`。
- 连接数限制：若填写0或者不填写则不做额外限制（但会受到最大连接数限制）。

创建账号

账号名 *

账号名需要1-32个字符，由字母、数字或特殊字符组成；以字母开头；特殊字符为_

创建为只读账号 是 否

如果选是，您可以在点击确定后，设置只读账号的参数

主机 *

IP形式，IP段以%结尾；支持填入%，127.0.0.1

设置密码 *

密码需要 8-32 个字符，不能以 '/' 开头，并且至少包含字母、数字和特殊字符 `()~!@#%&^*+={}|[]:;<>,./?` 中的两项

确认密码 *

密码需要 8-32 个字符，不能以 '/' 开头，并且至少包含字母、数字和特殊字符 `()~!@#%&^*+={}|[]:;<>,./?` 中的两项

连接数限制

- 0 +

请输入该账号连接数限制，若填写 0 或者不填写则不做额外限制（均会受到最大连接数限制）

备注

请输入备注说明，最多256个字符

[确认，下一步](#) [取消](#)

4. 当选择创建只读账号后，需要对只读账号 [配置读写分离](#)，确认无误后，单击确定。

- 选择主机则全部备机延迟时，从主机读取。

- 选择**直接报错**则全部备机延迟时，直接报错。
- 选择**只从备机读取**则忽略备机延迟，一直从备机读取（一般用于拉取 binlog 同步）。
- 若您的实例架构是一主一从，请谨慎选择**只从备机读取**，避免大事务等高负载任务，影响备机备份任务及可用性。

只读账号设置 ×

账号名

主机

只读请求分配策略 * 主机 直接报错 只从备机读取

选择“主机”则全部备机延迟时，从主机读取
 选择“直接报错”则全部备机延迟时，直接报错
 选择“只从备机读取”则忽略备机延迟参数，一直从备机读取（一般用于拉取 binlog 同步）
 若您的实例架构是一主一备，请谨慎选择“只从备机读取”，避免大事务等高负载任务，影响备机备份任务及可用性

只读固定备机 *

只读固定备机打开时，若主备延迟超过延迟参数则自动断开，不会切换到其他备机。
 只读固定备机关闭时，若主备延迟超过延迟参数则自动选择其他可用备机

只读备机延迟参数 * 秒

如果备机延迟超过本参数设置值，系统将认为备机发生故障，建议该参数值大于10

5. 进入修改权限对话框，根据需求分配权限后，单击**确定修改**即可完成权限分配。若需稍后设置权限，单击**取消修改**即可。

账号名称:

请选择数据库权限 (支持选择多个对象)

全局特权

- ALTER
- ALTER ROUTINE
- CREATE
- CREATE ROUTINE
- CREATE TEMPORARY TABLES
- CREATE VIEW
- DELETE
- DROP
- EVENT
- EXECUTE
- INDEX
- INSERT
- LOCK TABLES
- 全选

已选中修改对象

- ALTER
- ALTER ROUTINE
- CREATE
- CREATE ROUTINE
- CREATE TEMPORARY TABLES
- CREATE VIEW
- DELETE
- DROP
- EVENT
- EXECUTE
- INDEX
- INSERT
- LOCK TABLES

刷新 重置

相关 API

API 名称	描述
CreateAccount	创建账号

修改账号权限

最近更新时间：2024-12-16 17:42:03

操作场景

通过控制台可以对TDSQL MySQL版集中式数据库账号授予操作权限，可授予全局特权和对象级特权。

操作步骤

1. 登录 [TDSQL控制台](#)，在实例列表中，单击实例 ID 或操作列的管理，进入实例管理页面。
2. 在实例管理页面，选择账号管理页，找到需要修改权限的账号，单击修改权限。

账号名	主机	账号类型	只读分配策略	创建时间	更新时间	备注	操作
[模糊]	[模糊]	常规账号	无	2024-12-16 14:43:14	2024-12-16 14:43:14	[模糊]	修改权限 克隆账号 更多 ▼

3. 在弹出的对话框，选中或者取消需要授予的权限，单击确定。

- 全局特权：拥有实例下所有数据库的所有权限。
- 对象级特权：拥有实例下特定数据库的权限。

账号名称: [模糊]

请选择数据库权限 (支持选择多个对象)

全局特权

- ALTER
- ALTER ROUTINE
- CREATE
- CREATE ROUTINE
- CREATE TEMPORARY TABLES
- CREATE VIEW
- DELETE
- DROP
- EVENT
- EXECUTE
- INDEX
- INSERT
- LOCK TABLES
- 全选

已选中修改对象

- ALTER
- ALTER ROUTINE
- CREATE
- CREATE ROUTINE
- CREATE TEMPORARY TABLES
- CREATE VIEW
- DELETE
- DROP
- EVENT
- EXECUTE
- INDEX
- INSERT
- LOCK TABLES

↔

刷新 重置

确定修改
取消修改

相关 API

API 名称	描述
DescribeAccountPrivileges	查询账号权限
GrantAccountPrivileges	设置账号权限

配置读写分离

最近更新时间：2024-12-16 17:42:03

读写分离概述

TDSQL MySQL 版集中式数据库默认支持读写分离能力，架构中的每个从机都能支持只读能力，如果配置有多个从机，将由网关集群（TPProxy）自动分配到低负载从机上。

基于只读账号的读写分离

只读账号是一类仅有读权限的账号，默认从数据库集群中的从机（或只读实例）中读取数据。用户可以在 [控制台](#) 实例管理页的账号管理页中，设置只读账号和读策略：

创建账号

账号名 *

账号名需要1-32个字符，由字母、数字或特殊字符组成；以字母开头；特殊字符为_

创建为只读账号 是 否

如果选是，您可以在点击确定后，设置只读账号的参数

主机 *

IP形式，IP段以%结尾；支持填入%，127.0.0.1

设置密码 *

密码需要 8~32 个字符，不能以 '!' 开头，并且至少包含字母、数字和特殊字符
(~!@#%&^&*-+=_[]{};<>.,?/ 中的两项)

确认密码 *

密码需要 8~32 个字符，不能以 '!' 开头，并且至少包含字母、数字和特殊字符
(~!@#%&^&*-+=_[]{};<>.,?/ 中的两项)

连接数限制

请输入该账号连接数限制，若填写 0 或者不填写则不做额外限制（均会受到最大连接数限制）

备注

请输入备注说明，最多256个字符

[确认，下一步](#) [取消](#)

在只读账号设置选项中，您可以设置**只读请求分配策略**，定义在备机故障（或延迟较大）时的读策略。**只读备机延迟参数**定义数据同步延迟时间，并与**只读请求分配策略**配合使用：

- 选择**主机**则全部备机延迟时，从主机读取。
- 选择**直接报错**则全部备机延迟时，直接报错。
- 选择**只从备机读取**则忽略备机延迟，一直从备机读取（一般用于拉取 binlog 同步）。
- **只读备机延迟参数**定义数据同步延迟时间。

分配策略	只读固定备机	不只读固定备机
主机	延迟后，若大于延迟参数，则切换到主机读取	延迟后，若大于延迟参数，则先读其他备机，其他备机延迟后切换到主机读取
直接报错	延迟后，若大于延迟参数，则直接报错	延迟后，若大于延迟参数，则先读其他备机，其他备机延迟后直接报错

只从备机读取	延迟后，只读固定备机	-
--------	------------	---

只读账号设置

账号名

主机

只读请求分配策略 * 主机 直接报错 只从备机读取

选择“主机”则全部备机延迟时，从主机读取
 选择“直接报错”则全部备机延迟时，直接报错
 选择“只从备机读取”则忽略备机延迟参数，一直从备机读取（一般用于拉取 binlog 同步）
 若您的实例架构是一主一备，请谨慎选择“只从备机读取”，避免大事务等高负载任务，影响备机备份任务及可用性

只读固定备机 *



只读固定备机打开时，若主备延迟超过延迟参数则自动断开，不会切换到其他备机
 只读固定备机关闭时，若主备延迟超过延迟参数则自动选择其他可用备机

只读备机延迟参数 *

- 10 + 秒

如果备机延迟超过本参数设置值，系统将认为备机发生故障，建议该参数值大于10

确定

取消

! 说明:

- 若需要修改只读账号设置，可以在账号列表，在操作列选择更多> 修改只读分配策略。
- 若您的实例架构为一主一从，只读分离功能仅可用作低负载只读任务，请避免大事务等较高负载任务，影响从机备份任务及可用性。

配置建议（例如您设计的是某交易系统）：

- 核心交易模块：设置常规账号，可读写。
 - 余额查询模块：设置只读账号，默认读备机；请求分配策略：备机故障读主机，并将延迟参数设置在十秒内，以保证主从性能和用户查询的数据一致性。
 - 批量查询模块：设置只读账号，默认读备机；请求分配策略：备机故障报错，延迟参数可设置在三十秒以上，以保证不影响主库性能。
- 另外，由于强同步机制是将数据写入从机事务日志后，即返回应答，这时从机库表数据可能并未更新，因此也会有延迟。

基于注释的读写分离

通过如下视频，您可以了解基于注释的读写分离的使用：

[观看视频](#)

在每条需要从机“读”的 SQL 前，增加 `/*slave*/` 字段，且 `mysql` 后面要增加 `-c` 参数来解析注释 `mysql -c -e "/*slave*/sql"`，即可自动将读请求分配到从机，示例如下：

```
//主机读//
select * from emp order by sal, deptno desc;
//从机读//
/*slave*/ select * from emp order by sal, deptno desc;
```

! 注意:

- 该功能仅支持从机读（select），不支持其他操作，非 select 语句将失败。
- mysql 客户端需要增加 `-c` 参数来解析注释。
- `/*slave*/` 必须为小写，语句前后无空格。
- 从机出现异常而影响到 MAR（强同步）机制时，从机读操作将自动切换回主机。

只读实例（异地只读实例）

如果上述读写分离方案无法满足您的需求，TDSQL MySQL 版集中式数据库提供 [只读实例](#) 供您使用。只读实例是独立的数据库实例，不参与原主实例高可用切换，仅用于读性能扩展。

修改连接数限制

最近更新时间：2024-12-30 14:54:32

操作场景

通过控制台可以修改 TDSQL MySQL 版集中式数据库账号连接数限制，配置每个账号的连接上限。

操作步骤

1. 登录 [TDSQL 控制台](#)，在实例列表中，单击实例 ID 或操作列的管理，进入实例管理页面。
2. 在实例管理页面，选择账号管理页，找到需要修改连接数的账号，单击更多 > 修改配置。



3. 在弹出的设置配置对话框，修改连接数限制，单击确定。
连接数限制：若填写0则不做额外限制（但最大连接数上限为10240）。



克隆账号

最近更新时间：2024-12-30 14:54:32

操作场景

您可以通过控制台对 TDSQL MySQL 版集中式数据库账号进行克隆，保留原账号密码，并在不同主机上提供不同权限。

注意：

只有主实例可以管理账号，灾备 / 只读实例不具备账号管理功能。

操作步骤

1. 登录 [TDSQL 控制台](#)，在实例列表中，单击实例 ID 或操作列的管理，进入实例管理页面。
2. 在实例管理页面，选择账号管理页，找到需要克隆的账号，单击克隆账号。

账号名	主机	账号类型	只读分配策略	创建时间	更新时间	备注	操作
		常规账号	无	2024-12-16 14:43:14	2024-12-16 14:43:14		修改权限 克隆账号 更多 ▾

3. 在弹出的对话框，填写主机 IP，账号密码均复制原账号形式，单击确认，下一步。

克隆账号

账号名 *

账号名需要1-32个字符，由字母、数字或特殊字符组成；以字母开头；特殊字符为_

创建为只读账号 是 否

如果选是，您可以在点击确定后，设置只读账号的参数

主机 *

IP形式，IP段以%结尾；支持填入%，127.0.0.1

设置密码 *

密码需要 8~32 个字符，不能以 ' ' 开头，并且至少包含字母、数字和特殊字符
(~!@#%&^&*.+=_|{}|<>.,~/ 中的两项)

确认密码 *

密码需要 8~32 个字符，不能以 ' ' 开头，并且至少包含字母、数字和特殊字符
(~!@#%&^&*.+=_|{}|<>.,~/ 中的两项)

连接数限制

请输入该账号连接数限制，若填写 0 或者不填写则不做额外限制（均会受到最大连接数限制）

备注

请输入备注说明，最多256个字符

[确认，下一步](#) [取消](#)

4. 返回账号管理页，即可看到克隆的账号。

相关 API

API 名称	描述
CloneAccount	克隆实例账号

重置账号密码

最近更新时间：2024-12-30 14:54:32

操作场景

TDSQL MySQL 版集中式数据库支持重置实例密码，如您在使用数据库过程中，忘记了数据库账号密码或需修改密码，可通过控制台重新设置密码。

说明：

- 只有主实例可以管理账号，灾备 / 只读实例不具备账号管理功能。
- 为了数据安全，建议您定期更换密码，最长间隔不超过3个月。

操作步骤

- 登录 [TDSQL 控制台](#)，在实例列表中，单击实例 ID 或操作列的管理，进入实例管理页面。
- 在实例管理页面，选择账号管理页，找到需要重置密码的账号，选择更多 > 重置密码。



- 在重置密码对话框，输入新密码和确认密码，单击确定。

说明：

为避免创建、修改、删除账号信息给您的业务带来风险，建议配置 [访问管理](#)，谨慎重置密码。

重置密码

为避免创建、修改、删除账号信息给您的业务带来风险，建议配置访问管理谨慎控制本功能菜单权限（API关键词：*Account*）

实例名

账号名

主机

设置密码 *

密码需要 8~32 个字符，不能以 '/' 开头，并且至少包含字母、数字和特殊字符 (-!@#%&^*+._[]{}<>.,?/ 中的两项)

确认密码 *

密码需要 8~32 个字符，不能以 '/' 开头，并且至少包含字母、数字和特殊字符 (-!@#%&^*+._[]{}<>.,?/ 中的两项)

相关 API

API 名称	描述
--------	----

ResetAccountPassword

重置账号密码

删除账号

最近更新时间：2024-12-30 14:54:32

操作场景

本文为您介绍如何通过控制台删除 TDSQL MySQL 版集中式数据库账号。

说明：

- 只有主实例可以管理账号，灾备 / 只读实例不具备账号管理功能。
- 数据库账号删除后不可恢复，请保证账号不再使用后谨慎操作。

操作步骤

- 登录 [TDSQL 控制台](#)，在实例列表中，单击实例 ID 或操作列的管理，进入实例管理页面。
- 在实例管理页面，选择账号管理页，找到需要删除的账号，选择更多 > 删除账号。

账号名	主机	账号类型	只读分配策略	创建时间	更新时间	备注	操作
		常规账号	无	2024-12-16 14:43:14	2024-12-16 14:43:14		修改权限 克隆账号 更多 ▼

[修改配置](#)
[重置密码](#)
[删除账号](#)

- 在弹出的对话框，确认无误后，单击确定。



相关 API

API 名称	描述
DeleteAccount	删除账号

修改网络

最近更新时间：2024-12-16 17:42:03

当业务需要修改数据库实例访问地址或者更换私有网络时，可使用通过网络修改功能来调整网络。

⚠ 注意：

修改实例网络属于高危操作，请务必在业务低谷期谨慎操作。修改后，原地址将持续生效24小时（除非被另外业务占用），此时新 IP 地址与旧 IP 地址将同时生效，请尽快切换业务配置。

修改内网地址

在 VPC 网络下，TDSQL MySQL 版集中式数据库支持修改内网地址。

1. 登录 [TDSQL 控制台](#)，在实例列表，单击实例 ID，进入实例详情页。
2. 在实例详情页的“内网地址”处，单击  图标更改，前提条件是当前子网仍有可用 IP。



3. 在弹出的对话框，修改内网地址后，单击确定。

VPC 网络内切换子网

TDSQL MySQL 版集中式数据库支持实例在 VPC 网络中切换子网。

1. 登录 [TDSQL 控制台](#)，在实例列表，单击实例 ID，进入实例详情页。
2. 在实例详情页的所属网络处，单击更换网络。
3. 在弹出的对话框，选择子网后，选择自动分配 IP 或者指定 IP，单击确定。

ⓘ 说明：

- 更换子网，旧的 IP 地址将为您保留24小时，请在24小时内进行业务 IP 地址进行切换。
- 您也可以自行设置地址回收时间，可设置范围0 - 168小时。若回收地址设为0小时，更换网络后会立即回收旧 IP 地址，可能会对业务产生影响，请谨慎选择。
- 由于产品支持同城双活架构，建议您优先选择与业务服务器相同，或与主节点所在地域相同的 VPC 子网。

切换 VPC

1. 登录 [TDSQL 控制台](#)，在实例列表，单击实例 ID，进入实例详情页。
2. 在实例详情页的所属网络处，单击更换网络。
3. 在弹出的对话框，选择 VPC 后，选择自动分配 IP 或者指定 IP，单击确定。

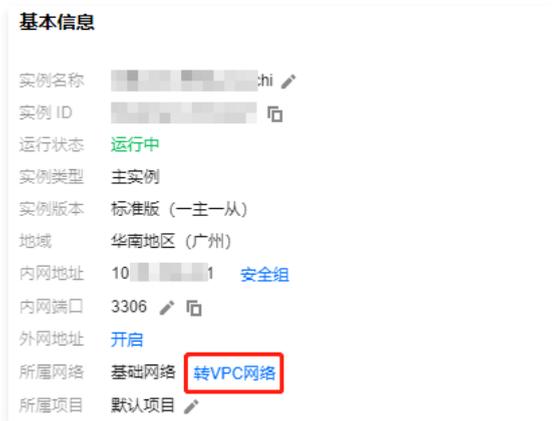
ⓘ 说明：

- 更换 VPC，旧的 IP 地址将为您保留24小时，请在24小时内进行业务 IP 地址进行切换。
- 您也可以自行设置地址回收时间，可设置范围0 - 168小时。若回收地址设为0小时，更换网络后会立即回收旧 IP 地址，可能会对业务产生影响，请谨慎选择。
- 由于产品支持同城双活架构，建议您优先选择与业务服务器相同，或与主节点所在地域相同的 VPC 子网。

基础网络转 VPC 网络

TDSQL MySQL 版集中式数据库支持实例从基础网络转入 VPC 网络。

1. 登录 [TDSQL 控制台](#)，在实例列表，单击实例 ID，进入实例详情页。
2. 在实例详情页的所属网络处，单击转 VPC 网络。



3. 在弹出的对话框，选择 VPC 后，选择自动分配 IP 或者指定 IP，单击确定。

注意：

- 基础网络切换至 VPC 网络后，无法再切换回来。
- 切换后，VPC 网络访问立即生效，原有基础网络的访问将保留24小时，请在24小时内将该实例相关联的实例迁至 VPC 网络，以保证相关实例的访问。
- 由于产品支持同城双活架构，建议您优先选择与业务服务器相同，或与主节点所在地域相同的 VPC 子网。

查看待回收保留地址

1. 登录 [TDSQL 控制台](#)，在实例列表中，单击实例 ID 或操作列的管理，进入实例管理页面。
2. 在实例管理页面，选择实例详情>基本信息，单击保留地址。

说明：

- 只有在修改网络地址/端口后，保留地址才予以展示。
- 保留地址到期自动释放，不再予以展示。



3. 在弹出的对话框中，查看当前处于保留地址的 IP、端口、所属网络、到期时间，单击立即释放可释放对应保留地址。

保留地址 ×

IP 地址	端口	私有网络	子网	到期时间	操作
██████	██████	██████	██████████	██████████	立即释放

共 1 条

备份与回档

备份方式

最近更新時間：2024-12-30 14:54:32

TDSQL MySQL 版集中式数据库支持全量备份和增量备份。备份采用 LZ4 方式压缩，使用方式请参见 [解压备份和日志文件](#)。

备份类型

全量备份

全量备份可设置备份保存时长和备份执行时间，备份保存时长默认为7天。

增量备份

增量备份以 binlog 方式提供，binlog 实时生成（将占用一定数据盘空间），并定期上传至云数据库备份系统。

自定义备份时间

1. 登录 [TDSQL 控制台](#)，单击实例 ID 或操作列的管理，进入实例管理页面。
2. 在实例管理页面，单击备份与恢复。
3. 在备份与恢复 > 备份和日志设置页，可设置存储时间和备份执行时间。
 - 存储时长：数据和日志备份文件保留的天数，默认为7天，可设置1天 - 3650天。
 - 备份执行时间：每日开始备份的时间，默认为全天，可设置任意时间段，以小时为单位。

说明：

日志备份默认开启，不可关闭。日志包括错误日志，慢日志，事务日志（binlog）等。

实例详情 监控告警 参数配置 数据安全 备份与恢复 性能优化

冷备列表 克隆实例 Binlog列表 备份和日志设置

常规备份 [设置常规备份](#) [刷新](#)

存储时长 7天

备份执行时间 00:00-23:59

备份周期 星期一、星期二、星期三、星期四、星期五、星期六、星期日

说明 赠送实例容量 100% 的免费存储空间

备份归档 [设置备份归档](#) [刷新](#)

1. 备份归档仅对全量备份文件生效，Binlog 文件不会归档存储；

2. 从归档存储中读取备份数据并恢复可能需要较长时间（最长可达 5-12 小时），不适用于需要快速恢复备份的场景。

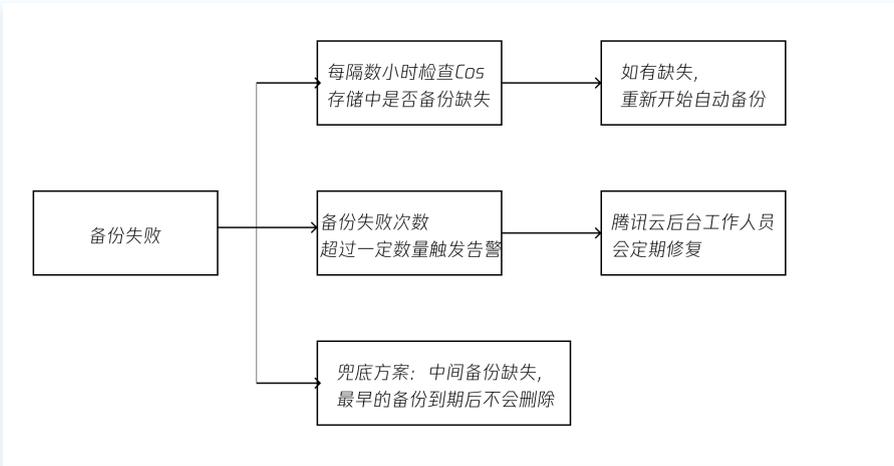
状态 未开启备份归档

说明：

若备份失败，存在以下三种处理逻辑：

- 自动修复：数据库每隔数小时检查 COS 存储中是否有备份缺失，如有备份缺失，则重新启动一次自动备份。

- 告警检查：备份失败次数超过一定次数后，将在腾讯云后台生成告警，后台工作人员将针对失败原因进行处理。
- 兜底方案：未避免备份失败较多次导致无备份可用，数据库会保证至少保留一个备份，即中间备份缺失后，最早的备份到期后也不会删除。



下载备份文件

最近更新时间：2024-12-16 17:42:03

用户可通过控制台下载 TDSQL MySQL 版集中式数据库的冷备数据、binlog。

下载备份文件和 binlog

1. 登录 [TDSQL 控制台](#)，单击实例 ID 或操作列的管理，进入实例管理页面。
2. 选择备份与恢复 > 冷备列表 或 Binlog 列表。
3. 选择需要下载的备份，在操作列单击下载。
4. 在弹出的对话框，提供了在 VPC 内网中下载此备份的地址，单击获取下载链接。
5. 登录到云数据库所在 VPC 下的 CVM（Linux 系统）中，运用 wget 命令进行下载。

说明：

- 外网下载：请在左侧导航数据库备份页的下载设置中开启外网下载，下载链接可直接复制到浏览器进行下载。
- 内网下载：请在 VPC 网络中进行访问，使用 wget 命令下载：`wget -O <自定义名称.log> '<备份文件下载地址>'`。
- 地址有效期为15分钟，过期后请重新刷新页面获取。

下载

温馨提示

1. 为保证数据安全，地址有效期为 15 分钟，过期后请重新获取
2. 外网下载：请前往 数据库备份>下载设置 中开启外网下载，下载链接可直接复制到浏览器进行下载
3. 内网下载：请在VPC网络中进行访问，使用wget命令下载：`wget -O <自定义名称.log> '<备份文件下载地址>'`
4. 查看下载帮助文档 [🔗](#)

获取下载链接

关闭

下载慢查询日志

1. 登录 [TDSQL 控制台](#)，单击实例 ID 或操作列的管理，进入实例管理页面。
2. 选择性能优化 > 慢查询日志页。
3. 选择需要下载的备份，在操作列单击下载。

说明：

若备份大小为0KB，无慢查询记录，则无法下载。

4. 在弹出的对话框，提供了在 VPC 内网中下载此日志的地址，单击获取下载链接。
5. 登录到云数据库所在 VPC 下的 CVM（Linux 系统）中，运用 wget 命令进行下载。

说明：

- 外网下载：请在左侧导航数据库备份页的下载设置中开启外网下载，下载链接可直接复制到浏览器进行下载。
- 内网下载：请在 VPC 网络中进行访问，使用 wget 命令下载：`wget -O <自定义名称.log> '<慢日志下载地址>'`。
- 地址有效期为15分钟，过期后请重新刷新页面获取。

解压备份和日志文件

最近更新时间：2024-12-16 17:42:03

出于压缩性能和压缩比的综合考虑，TDSQL MySQL 版集中式数据库的备份文件和日志文件（binlog 文件）采用 LZ4（Extremely Fast Compression algorithm）工具进行压缩，您可以选用 LZ4 工具进行解压。本文为您介绍解压工具的操作指引。

Windows

安装工具

下载安装 LZ4 工具，按指引完成安装。

解压文件

右键单击需要解压的 lz4 文件，选择 **Decode with LZ4** 项即可完成解压。

Linux

安装工具

腾讯云服务器的 yum 库中有 LZ4 组件，[登录云服务器](#) 执行如下命令即可安装。

```
$ yum install lz4
```

直接执行 **lz4** 返回类似如下图所示，表示安装正确。

```
[root@UM_240_177_centos ~]# lz4
Incorrect parameters
Usage :
      lz4 [arg] [input] [output]

input   : a filename
          with no FILE, or when FILE is - or stdin, read standard input
Arguments :
-1      : Fast compression (default)
-9      : High compression
-d      : decompression (default for .lz4 extension)
-z      : force compression
-f      : overwrite output without prompting
-h/-H   : display help/long help and exit
```

解压文件

执行如下命令即可完成解压。

```
$ lz4 -d xxx.lz4
```

通过备份文件恢复实例

最近更新时间：2024-12-16 17:42:03

TDSQL MySQL 版集中式数据库可通过回档功能来查看历史数据，如果您需要在本地恢复您的数据库实例，可按照本文的步骤进行操作恢复历史数据。

前提条件

准备服务器

如您需在本地恢复数据库实例，请确保服务器基本配置如下：

- CPU：2核或以上。
- 内存：4G或以上。
- 磁盘空间：必须超过数据库已用空间并留足系统所需的临时空间。
- 操作系统：centos。

准备数据库

❗ 说明：

请注意，本文档提供的本地安装的指导仅为示意，实际安装方法可能会因您的服务器和网络配置而有所不同。以下是一些可能影响安装过程的因素：

1. 操作系统版本：

- 不同的操作系统和版本可能需要不同的安装步骤和命令。例如，CentOS 和 Ubuntu 的包管理器 and 命令有所不同。

2. 网络配置：

- 如果您的服务器在防火墙后面或有特定的网络限制，可能需要额外的配置来允许下载和安装 MariaDB。

3. 权限问题：

- 安装 MariaDB 可能需要管理员权限。如果您没有适当的权限，可能需要联系系统管理员或使用 sudo 命令。

4. 软件源和镜像：

- 有些用户可能需要使用特定的镜像源或软件仓库，这可能会影响安装命令和配置文件的内容。

5. 依赖关系：

- 安装 MariaDB 可能需要其他软件包或依赖项。如果这些依赖项未安装或版本不兼容，可能会导致安装失败。

⚠ 注意：

请保证本地安装的数据库版本和云上数据库实例的版本保持一致。

下文以安装 MariaDB 10.0.10 为例：

1. 添加 yum 源。

```
vi /etc/yum.repos.d/mariadb-10.0.10.repo :
# MariaDB 10.0 CentOS repository list - created 2016-05-30 02:16 UTC
# http://downloads.mariadb.org/mariadb/repositories/
[mariadb]
name = MariaDB
# baseurl = http://yum.mariadb.org/10.0/centos7-amd64
baseurl = http://archive.mariadb.org/mariadb-10.0.10/yum/centos6-amd64/
gpgkey=https://yum.mariadb.org/RPM-GPG-KEY-MariaDB
gpgcheck=0
```

2. 检查配置 yum 源对应的 MariaDB 版本是否为10.0.10。

```
yum makecache
yum info MariaDB-server
```

- 安装 MariaDB-server。

```
yum install MariaDB-server
```

说明:

提示与旧版本冲突时，需要先移除之前的安装包，例如 `yum remove mariadb-libs`。

安装辅助工具

1. 安装 MariaDB 客户端。

```
yum install MariaDB-client
```

- 安装 LZ4 解压软件，请参见 [解压备份文件和日志文件](#)。LZ4 默认安装在 `mysqlagent/bin` 目录下，也可以将其放置在 `/usr/bin` 目录下，当环境变量引用。

```
yum install -y lz4
percona-xtrabackup
yum install https://mirrors.tuna.tsinghua.edu.cn/percona/centos/7/RPMS/noarch/percona-release-0.1-3.noarch.rpm
yum install percona-xtrabackup
```

下载备份

在 [TDSQL 控制台](#)，单击实例 ID 进入实例管理页，在备份与恢复页获取备份下载地址。

下载命令示例：

```
wget --content-disposition 'http://1x.2xx.0.27:8083/2/noshard1/set_1464144850_587/1464552298xxxxxxxx'
```

通过备份文件恢复数据库（未加密）

1. 进入备份文件下载目录，通过 lz4 解压冷备文件

```
lz4 -d set_1464144850_587.1464552298.xtrabackup.lz4
```

2. 使用 xbstream 工具解压到临时目录 xtrabackuptmp

```
mkdir xtrabackuptmp/
mv set_1464144850_587.1464552298.xtrabackup xtrabackuptmp/
cd xtrabackuptmp/
xbstream -x < set_1464144850_587.1464552298.xtrabackup
```

解压完后，目录和文件内容如下：

```
[root@VM_0_2_centos xtrabackuptmp]# ll
total 6347108
-rw-r----- 1 root root      358 May 30 16:28 backup-my.cnf
-rw-r--r--  1 root root 1073741824 May 30 16:31 ib_logfile0
-rw-r--r--  1 root root 1073741824 May 30 16:31 ib_logfile1
-rw-r--r--  1 root root 1073741824 May 30 16:31 ib_logfile2
-rw-r--r--  1 root root 1073741824 May 30 16:31 ib_logfile3
-rw-r----- 1 root root 2147483648 May 30 16:31 ibdata1
drwx----- 2 root root      4096 May 30 16:28 mysql
drwx----- 2 root root      4096 May 30 16:28 performance_schema
-rw-r--r--  1 root root 2149044297 May 30 16:27 set_1464144850_587.1464552298.xtrabackup
drwx----- 2 root root      4096 May 30 16:28 sysdb
drwx----- 2 root root      4096 May 30 16:28 test
-rw-r----- 1 root root        25 May 30 16:28 xtrabackup_binlog_info
-rw-r--r--  1 root root         58 May 30 16:30 xtrabackup_binlog_pos_innodb
-rw-r----- 1 root root        117 May 30 16:30 xtrabackup_checkpoints
-rw-r----- 1 root root         858 May 30 16:28 xtrabackup_info
-rw-r----- 1 root root    2097152 May 30 16:30 xtrabackup_logfile
[root@VM_0_2_centos xtrabackuptmp]#
```

3. 使用 innobackupex 应用日志

```
mkdir /root/dblogs_tmp
innobackupex --apply-log --use-memory=1G --tmpdir='/root/dblogs_tmp/' /root/xtrabackuptmp/
```

操作成功后，会显示 `completed OK!`，如下所示：

```
InnoDB: Initializing buffer pool, size = 1.0G
InnoDB: Completed initialization of buffer pool
InnoDB: Setting log file ./ib_logfile101 size to 1024 MB
InnoDB: Progress in MB: 100 200 300 400 500 600 700 800 900 1000
InnoDB: Setting log file ./ib_logfile1 size to 1024 MB
InnoDB: Progress in MB: 100 200 300 400 500 600 700 800 900 1000
InnoDB: Setting log file ./ib_logfile2 size to 1024 MB
InnoDB: Progress in MB: 100 200 300 400 500 600 700 800 900 1000
InnoDB: Setting log file ./ib_logfile3 size to 1024 MB
InnoDB: Progress in MB: 100 200 300 400 500 600 700 800 900 1000
InnoDB: Renaming log file ./ib_logfile101 to ./ib_logfile0
InnoDB: New log files created, LSN=210681783
InnoDB: Highest supported file format is Barracuda.
InnoDB: 128 rollback segment(s) are active.
InnoDB: Waiting for purge to start
InnoDB: 5.6.24 started; log sequence number 210681868
xtrabackup: starting shutdown with innodb_fast_shutdown = 1
InnoDB: FTS optimize thread exiting.
InnoDB: Starting shutdown...
InnoDB: Shutdown completed; log sequence number 210683014
160530 16:31:18 completed OK!
[root@VM_0_2_centos ~]# ls
```

4. 停止数据库，清空数据文件

```
service mysql stop
```

清空数据文件（数据目录、表空间目录、日志目录）：

```
mkdir /var/lib/mysql-backup
mv /var/lib/mysql/* /var/lib/mysql-backup
```

5. 修改数据库参数文件

修改数据库参数文件（`/etc/my.cnf.d/server.cnf`），具体参数数值请参考解压文件中 `backup-my.cnf` 的参数。不能直接用 `backup-my.cnf` 替换参数文件。

```
[mysqld]
skip-name-resolve
datadir=/var/lib/mysql
innodb_checksum_algorithm=innodb
innodb_log_checksum_algorithm=innodb
```

```
innodb_data_file_path=ibdata1:2G:autoextend
innodb_log_files_in_group=4
innodb_log_file_size=1073741824
innodb_page_size=4096
innodb_log_block_size=512
innodb_undo_tablespaces=0
```

6. 使用 innobackupex 加载镜像

```
innobackupex --defaults-file=/etc/my.cnf --move-back /root/xtrabackuptmp/
```

成功后，显示 Complete OK!，如下所示：

```
160530 15:17:36 [01] Moving ./performance_schema/mutex_instances.frm to /var/lib/mysql/performance_schema/mutex_instances.frm
160530 15:17:36 [01] ..done
160530 15:17:36 [01] Moving ./performance_schema/events_statements_history_long.frm to /var/lib/mysql/performance_schema/events_statements_history_long.frm
160530 15:17:36 [01] ..done
160530 15:17:36 [01] Moving ./performance_schema/events_waits_summary_by_thread_by_event_name.frm to /var/lib/mysql/performance_schema/events_waits_summary_by_thread_by_event_name.frm
160530 15:17:36 [01] ..done
160530 15:17:36 [01] Moving ./performance_schema/events_stages_summary_by_user_by_event_name.frm to /var/lib/mysql/performance_schema/events_stages_summary_by_user_by_event_name.frm
160530 15:17:36 [01] ..done
160530 15:17:36 [01] Moving ./performance_schema/events_statements_summary_global_by_event_name.frm to /var/lib/mysql/performance_schema/events_statements_summary_global_by_event_name.frm
160530 15:17:36 [01] ..done
160530 15:17:36 [01] Moving ./performance_schema/users.frm to /var/lib/mysql/performance_schema/users.frm
160530 15:17:36 [01] ..done
160530 15:17:36 [01] Moving ./performance_schema/events_statements_history.frm to /var/lib/mysql/performance_schema/events_statements_history.frm
160530 15:17:36 [01] ..done
160530 15:17:36 [01] Moving ./performance_schema/events_waits_summary_by_host_by_event_name.frm to /var/lib/mysql/performance_schema/events_waits_summary_by_host_by_event_name.frm
160530 15:17:36 [01] ..done
160530 15:17:36 [01] Moving ./performance_schema/socket_summary_by_instance.frm to /var/lib/mysql/performance_schema/socket_summary_by_instance.frm
160530 15:17:36 [01] ..done
160530 15:17:36 [01] Moving ./performance_schema/events_statements_summary_by_digest.frm to /var/lib/mysql/performance_schema/events_statements_summary_by_digest.frm
160530 15:17:36 [01] ..done
160530 15:17:36 [01] Moving ./performance_schema/events_waits_summary_by_user_by_event_name.frm to /var/lib/mysql/performance_schema/events_waits_summary_by_user_by_event_name.frm
160530 15:17:36 [01] ..done
160530 15:17:36 [01] Moving ./performance_schema/events_waits_summary_by_account_by_event_name.frm to /var/lib/mysql/performance_schema/events_waits_summary_by_account_by_event_name.frm
160530 15:17:36 [01] ..done
160530 15:17:36 [01] Moving ./performance_schema/events_stages_summary_by_host_by_event_name.frm to /var/lib/mysql/performance_schema/events_stages_summary_by_host_by_event_name.frm
160530 15:17:36 [01] ..done
160530 15:17:36 [01] Moving ./performance_schema/file_summary_by_event_name.frm to /var/lib/mysql/performance_schema/file_summary_by_event_name.frm
160530 15:17:36 [01] ..done
160530 15:17:36 [01] completed OK!
```

7. 启动数据库

```
chmod 777 -R /var/lib/mysql
service start mysql
```

如果遇到启动失败，则需要检查错误信息，修复后再重新启动。

8. 连接数据库查看数据

数据库启动后，您可能需要通过原来的账号密码来连接数据库查看数据。

通过备份文件恢复数据库（已加密）

数据透明加密（TDE）当前仅支持 MySQL 8.0.24 版本和香港地区 Percona 5.7 版本，后续将陆续开放。您可以通过在 [TDSQL 控制台](#) 的实例管理页的 [数据安全](#) > [数据加密](#) 进行访问。

开启数据加密后，暂时不支持用备份文件在本地恢复数据库实例，推荐采用 [回档数据库](#) 进行恢复。

❗ 说明：

如需使用数据加密功能，请 [提交工单](#) 申请。

回档/克隆数据库

最近更新时间：2025-04-23 10:39:52

回档/克隆说明

TDSQL MySQL 版集中式数据库可以根据备份和日志保持情况，回档/克隆到1~3650天内的任意时刻。通过数据库回档/克隆能力，可以最大程度地减少系统损失。

TDSQL MySQL 版集中式数据库回档/克隆功能不会影响现网生产实例，可直接回档/克隆到腾讯云创建的一个新的按量计费实例。回档/克隆的新实例是一个按量计费实例，用户可以根据需要自行选择配置。

限制条件

- 回档/克隆、创建临时实例过程中，现网生产实例的部分管理功能将不可用，操作完成即可恢复数据库管理操作。
- 回档/克隆操作有可能会对二进制日志（binlog）进行强行分片，即未达到100MB也会被备份为一个独立文件。
- 回档/克隆后的新购实例会具备现网生产实例的参数信息（如账户、数据库参数等），请注意账号管理。

回档/克隆实例

- 登录 [TDSQL 控制台](#)，单击实例 ID，进入实例管理页。
- 在实例管理页，选择备份与恢复 > 克隆实例，单击新建克隆实例。



- 在弹出的对话框中，设置回档时间后，单击确定。



- 在克隆实例页面，根据需要变更配置，单击立即购买后等待实例克隆完成。

⚠ 注意：

若用户选择的克隆实例规格小于源实例规格，可能导致回放 binlog 的过程较慢。因此建议您在创建克隆实例时选择与源实例相同的规格。

备份归档

最近更新时间：2025-05-26 10:30:52

备份归档开启场景

用户可以启用备份归档功能，将需要长期保存的冷备沉降至归档存储，以降低存储成本。

说明：

全量备份文件生成后的91天后才可沉降至归档存储。

归档地域限制

此次备份分级存储的支持情况按不同地域分类如下：

支持情况	地域
国内支持区域	北京、重庆、成都、上海、南京、广州
不支持归档存储的区域	雅加达、北京金融、上海金融
境外支持区域	中国香港、新加坡、首尔、曼谷、东京、硅谷、弗吉尼亚
国内支持的金融区	深圳金融

归档开启操作

1. 登录 [TDSQL 控制台](#)，单击实例 ID，进入实例管理页。
2. 在实例管理页，选择备份与恢复 > 备份和日志设置，设置备份归档。

备份归档

设置备份归档 刷新

1. 备份归档仅对全量备份文件生效，Binlog 文件不会归档存储；

2. 从归档存储中读取备份数据并恢复可能需要较长时间（最长可达 5-12 小时），不适用于需要快速恢复备份的场景。

状态 开启备份归档

备份归档设置 全量备份文件生成后的 天后沉降至归档存储。

计费说明

备份归档计费详情请参见 [备份空间计费](#)。

数据迁移

使用 mysqldump 导入数据

最近更新时间：2024-08-07 15:19:21

使用 mysqldump 工具导入数据简单易上手，但停机时间较长，适用于数据量不大，或允许停机时间较长的场景。

1. 使用 mysqldump 数据导出工具

1.1 仅导出数据：使用 mysqldump 数据导出工具，将本地数据库数据导出为数据文件。

说明：

- 导出期间请勿进行数据更新。本步骤仅导出数据，不包括存储过程、触发器和函数。
- 导出账号需拥有 `select on *.*` 的权限。

```
mysqldump -h localIp -u userName -p --opt --default-character-set=utf8 --hex-blob dbName --skip-triggers > /tmp/dbName.sql
```

参数说明：

- `localIp`：本地数据库服务器 IP 地址。
- `userName`：本地数据库的迁移账号。
- `dbName`：需要迁移的数据库名。
- `/tmp/dbName.sql`：备份生成的文件名。

1.2 使用 mysqldump 同时导出数据、存储过程、触发器和函数。

说明：

若数据库中没有使用存储过程、触发器和函数，请跳过此步骤。导出存储过程、触发器和函数时，需要将 `definer` 去掉，以兼容云数据库。

```
mysqldump -h localIp -u userName -p --opt --default-character-set=utf8 --hex-blob dbName -R | sed -e 's/DEFINER[ ]*=[ ]*[^(]*\*/\*/' > /tmp/triggerProcedure.sql
```

参数说明：

- `localIp`：本地数据库服务器 IP 地址。
- `userName`：本地数据库的迁移账号。
- `dbName`：需要迁移的数据库名。
- `/tmp/triggerProcedure.sql`：备份生成的文件名。

2. 将数据文件和存储过程文件上传至云服务器 CVM。请确保 CVM 和云数据库能正常连通，且 CVM 存储空间足够。

3. 登录 CVM，将数据文件和存储过程文件导入至目标云数据库。请确保您拥有相应权限的数据库账号，否则需至控制台生成账号。

```
mysql -h xxx.xxx.xxx.xxx:xxxx -u userName -p dbName < /tmp/dbName.sql  
mysql -h xxx.xxx.xxx.xxx:xxxx -u userName -p dbName < /tmp/triggerProcedure.sql
```

参数说明：

- `xxx.xxx.xxx.xxx:xxxx`：实例连接地址，本文以内网地址为例。
- `userName`：云数据库的迁移账号。
- `dbName`：需要导入的数据库名。
- `/tmp/dbName.sql`：需要导入的数据文件名。
- `/tmp/triggerProcedure.sql`：需要导入的存储过程文件名。

使用 DTS 工具导入数据到 MariaDB 实例

最近更新时间：2024-12-16 17:42:03

数据库迁移工具已升级并更换入口至 [数据库传输服务 DTS](#)，详细迁移介绍请参见 [迁移至 TDSQL\(MariaDB\)](#)。

数据同步

最近更新时间：2024-12-16 17:42:03

云数据库可使用 [数据传输服务 DTS](#) 进行数据同步，详细迁移介绍请参见 [TDSQL \(MariaDB\) 数据同步](#)。

安全管理

信息安全说明

最近更新时间：2021-03-01 15:23:18

特对本文做如下声明

1. 本文档意在向客户介绍腾讯云对于云数据库产品、服务的安全概况，部分产品、服务的内容可能有所调整。如您对此有强制要求，建议您与腾讯云以书面商业合同（SLA）进行约定。否则，腾讯云对本文档内容不做任何明示或模式的承诺或保证。
2. 安全特性范围较广，本文仅涉及“部分”技术安全要点。
3. 本文档不作为国家或行业信息安全相关标准、要求参考文档。
4. 本文经过可读性加工，若存在描述不准确的地方请参考第1点。
5. 文档解释权归腾讯云所有。

1、概述

腾讯云云数据库通过下列认证并符合下列认证的安全要求：

- ISO22301认证
- ISO27001认证
- ISO20000认证
- ISO9001认证
- 可信云服务认证
- 信息安全等级保护（三级）
- STAR认证

云数据库部分功能设计也参考：

- GBT 20273-2006 信息安全技术 数据库管理系统安全技术要求（二级以上）
- JRT 0072-2012 金融行业信息系统信息安全等级保护测评指南（四级）

2、腾讯云云数据库服务安全保障（运维安全说明）

2.1 综述

云数据库的管理安全与技术安全要求符合国家信息安全等级保护（三级），部分要求达到金融行业信息安全（四级）标准。

2.2 内部人员、系统身份鉴别

为提高数据库主机系统安全性，保障各种运维的安全性，腾讯云进行一系列的加固措施，包括但不限于：

- 对登录操作系统和数据库系统的用户进行身份标识和鉴别，且保证用户名的唯一性。
- 根据基本要求配置用户名/口令；口令必须具备采用3种以上字符、长度不少于8位并定期更换。
- 启用登录失败处理功能，登录失败后采取结束会话、限制非法登录次数和自动退出等措施。
- 远程管理时通过腾讯企业IT监控下的访问方式，提供内部风控审计，敏感操作均进行加密。
- 对数据库主机管理员登录运维系统进行双因素认证方式，即采用动态令牌+密码进行身份鉴别。

2.3 内部人员、系统访问控制

腾讯云云数据库管理系统和管理人员提供自主访问控制方案，包括但不限于：

- 内部运维人员和系统全部基于腾讯云安全策略进行控制（满足审核要求）。
- 主体的粒度为用户级，客体的粒度精确到数据库表级。
- 提供严格的代码管理和访问控制。
- 高危系统仅腾讯内网（开发网）可访问；与互联网严格物理隔离。

2.4 内部安全审计

提供全面的安全审计和风控机制：审计功能包括但不限于数据库操作审计、管理系统操作审计、文件操作审计、外挂设备操作审计、非法外联审计、IP地址更改审计、服务与进程审计等。

审计范围覆盖到服务器上的每个操作系统用户和数据库用户；例如包括：腾讯云管理员行为、系统资源的异常使用和重要系统命令的使用等系统内重要的安全相关事件；审计记录包括事件的日期、时间、类型、主体标识、客体标识和结果等；审计记录保存1年以上，且存储在安全等级更高的位置，避免受到未预期的删除、修改或覆盖等。

- 数据库安全审计：对数据库主机和数据库所有操作，均经过数据库安全审计系统。
- 管理系统操作审计：腾讯云对内外管理系统的操作均记录详细操作日志，以便于风险追溯。
- 定期风险评估：腾讯云安全团队定期对数据库运维管理进行安全评估。

2.5 内部入侵防范

针对数据库主机的入侵防范，腾讯云已经从多个角度进行处理：

- 入侵检测系统可以起到防范针对数据库主机的入侵行为。
- 部署漏洞扫描并定期进行系统安全性检测。
- 部署终端安全管理系统，开启补丁分发功能模块及时进行系统补丁升级。
- 操作系统的安装遵循最小安装的原则，仅安装需要的组件和应用程序，关闭多余服务等。
- 另外根据系统类型进行其它安全配置的加固处理。

2.6 备份与恢复

腾讯云数据库默认提供数据的备份与恢复功能。

2.7 客体安全重用

若客户退还设备或故障更换设备，腾讯云将及时清除剩余信息，从而保证用户的鉴别信息、文件、目录、数据库记录等敏感信息所在的存储空间（内存、硬盘）被及时释放或再分配给其他用户前得到完全清除。

2.8 抗抵赖

腾讯云内部运维人员登录系统，均通过双因素认证和抗抵赖方案，相关人员均签订了保密协议。

访问管理

概述

最近更新时间：2023-02-06 17:55:29

如果您在腾讯云中使用到了云数据库、云服务器、私有网络等服务，这些服务由不同的人管理，但都共享您的云账号密钥，将存在以下问题：

- 您的密钥由多人共享，泄密风险高。
- 您无法限制他人的访问权限，易产生误操作造成安全风险。

这个时候，访问管理 CAM 应运而生。CAM 的详细介绍，请参见 [CAM 概述](#)。

接入 CAM 后，可通过子账号实现不同的人管理不同的服务，以避免以上的问题。默认情况下，子账号没有使用云数据库实例以及云数据库相关资源的权限。

因此，我们就需要创建策略来允许子账号使用他们所需要的资源或者权限。

策略是定义和描述一条或多条权限的语法规则，策略通过授权一个用户或者一组用户来允许或拒绝使用指定资源。有关 CAM 策略的更多基本信息，请参见 [策略语法](#)。有关 CAM 策略的更多使用信息，请参见 [策略](#)。

如果您不需要对子账号进行云数据库相关资源的访问管理，您可以跳过此章节。跳过这些部分并不影响您对文档中其余部分的理解和使用。

入门

CAM 策略必须授权使用一个或多个云数据库操作或者必须拒绝使用一个或多个云数据库操作。同时还必须指定可以用于操作的资源（可以是全部资源，某些操作也可以是部分资源），策略还可以包含操作资源所设置的条件。

⚠ 注意

- 建议用户使用 CAM 策略来管理云数据库资源和授权云数据库操作，对于存量分项目权限的用户体验不变，但不建议再继续使用分项目权限来管理资源与授权操作。
- 云数据库暂时不支持相关生效条件设置。

策略结构

最近更新时间：2024-05-16 14:57:01

策略语法

CAM 策略配置示例：

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

- **版本 version**：必填项，目前允许值为"2.0"（该值实际代表 CAM 接受的云 API 版本）。
- **语句 statement**：用来描述一条或多条权限的详细信息。该元素包括 effect、action、resource、condition 等多个其他元素的权限或权限集合。一条策略有且仅有一个 statement 元素。
- **操作 action**：用来描述允许或拒绝的操作。操作 action 实际填入的是以 "mariadb:" 前缀描述，[云数据库 MariaDB API](#) 为后缀的一串字符串。该元素是必填项。
 - **资源 resource**：描述授权的具体数据。资源是用六段式描述。每款产品的资源定义详情会有所区别。有关如何指定资源的信息，请参阅您编写的资源声明所对应的产品文档。该元素是必填项。
 - **生效条件 condition**：描述策略生效的约束条件。条件包括操作符、操作键和操作值组成。条件值可包括时间、IP 地址等信息。有些服务允许您在条件中指定其他值。该元素是非必填项。
 - **影响 effect**：描述声明产生的结果是“允许”还是“显式拒绝”。包括 allow（允许）和 deny（显式拒绝）两种情况。该元素是必填项。

⚠ 注意

云数据库 MariaDB 在访问管理的接口关键词为 mariadb。

云数据库的操作

在云数据库策略语句中，您可以从支持云数据库的任何服务中指定任意的 API 操作。对于云数据库，请使用以 mariadb: 为前缀的 API。例如 mariadb:CreateDBInstance（创建实例-包年包月）或者 mariadb:CloseDBExtranetAccess（关闭外网访问）。

- 如果您要在单个语句中指定多个操作的时候，请使用英文逗号将它们隔开，如下所示：

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

- 您也可以使用通配符指定多项操作。例如，您可以指定名字以单词 "Describe" 开头的所有操作，如下所示：

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

- 如果您要指定云数据库中所有操作，请使用 * 通配符，如下所示：

```
"action": ["mariadb:*"]
```

云数据库的资源

每个 CAM 策略语句都有适用于自己的资源。

资源的一般形式如下：

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

- **project_id**: 描述项目信息，仅为了兼容 CAM 早期逻辑，无需填写。
- **service_type**: 产品简称，如 mariadb。
- **region**: 地域信息，如 ap-guangzhou。详情请参考 [地域相关信息](#)。
- **account**: 资源拥有者的主账号信息，如 uin/65xxx763。
- **resource**: 各产品的具体资源详情，如 instance/instance_id1 或者 instance/*。

例如，

- 您可以使用特定实例（tdsql-k05xdcta）在语句中指定它，如下所示：

```
"resource": [ "qcs::mariadb:ap-guangzhou:uin/65xxx763:instance/tdsql-k05xdcta" ]
```

- 您还可以使用 * 通配符指定属于特定账户的所有实例，如下所示：

```
"resource": [ "qcs::mariadb:ap-guangzhou:uin/65xxx763:instance/*" ]
```

- 您要指定所有资源，或者如果特定 API 操作不支持资源级权限，请在 Resource 元素中使用 * 通配符，如下所示：

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

- 如果您想要在一条指令中同时指定多个资源，请使用英文逗号将它们隔开，如下所示为指定两个资源的例子：

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

下表描述了云数据库能够使用的资源和对应的资源描述方法。

在下表中，\$ 为前缀的单词均为代称。

- 其中，region 指代的是地域。
- 其中，account 指代的是账户 ID。

资源	授权策略中的资源描述方法
实例	qcs::mariadb:\$region:\$account:instance/\$instanceId

支持的资源级权限

最近更新时间：2024-12-16 17:42:03

注意：

TDSQL MySQL 版集中式数据库在访问管理的接口关键词为 mariadb。

资源级权限指的是能够指定允许用户对哪些资源具有执行操作的能力。云数据库部分支持资源级权限，这意味着对于某些云数据库操作，您可以控制何时允许用户执行操作（基于必须满足的条件）或是允许用户使用的特定资源。下表将向您介绍云数据库可授权的资源类型。

CAM 中可授权的资源类型：

资源类型	授权策略中的资源描述方法
云数据库实例相关	<pre>qcs::mariadb:\$region:\$account:instance/* qcs::mariadb:\$region:\$account:instance/\$instanceId</pre>

下表将介绍当前支持资源级权限的云数据库 API 操作，以及每个操作支持的资源和条件密钥。指定资源路径的时候，您可以在路径中使用 * 通配符。

说明：

表中未列出的云数据库 API 操作，即表示该云数据库 API 操作不支持资源级权限。针对不支持资源级权限的云数据库 API 操作，您仍可以向用户授予使用该操作的权限，但策略语句的资源元素必须指定为 *。

下列操作可支持资源级权限

操作名	API 名	配置后控制台是否生效
恢复独享实例	ActiveDedicatedDBInstance	YES
绑定安全组	AssociateSecurityGroups	YES
检查 IP 状态	CheckIpStatus	YES
克隆账号	CloneAccount	YES
关闭外网地址	CloseDBExtranetAccess	YES
复制权限	CopyAccountPrivileges	YES
创建账号	CreateAccount	YES
创建参数模板	CreateConfigTemplate	YES
创建实例	CreateDBInstance	YES
回档实例	CreateTplInstances	YES
删除账号	DeleteAccount	YES
删除参数模板	DeleteConfigTemplate	YES
删除临时实例	DeleteTplInstance	YES
获取权限列表	DescribeAccountPrivileges	YES
查询账号列表	DescribeAccounts	YES
查询审计日志	DescribeAuditLogs	YES
查询审计规则详情	DescribeAuditRuleDetail	YES
查询审计规则列表	DescribeAuditRules	YES

查询审计策略	DescribeAuditStrategies	YES
获取自定义备份时间	DescribeBackupTime	YES
批量实例续费询价	DescribeBatchRenewalPrice	YES
查询 binlog 时间	DescribeBinlogTime	YES
查询参数配置历史	DescribeConfigHistories	YES
查询参数模板	DescribeConfigTemplate	YES
查询参数模板列表	DescribeConfigTemplates	YES
查询实例对象	DescribeDatabaseObjects	YES
查询实例 DB 名	DescribeDatabases	YES
查询实例表的列信息	DescribeDatabaseTable	YES
查询监控信息详情	DescribeDBDetailMetrics	YES
查询实例密钥信息	DescribeDBEncryptAttributes	YES
查询实例详情	DescribeDBInstanceDetail	YES
查询实例 HA 信息	DescribeDBInstanceHAInfo	YES
查询实例规格	DescribeDBInstanceSpecs	YES
查询实例列表	DescribeDBInstances	YES
获取日志列表	DescribeDBLogFiles	YES
查询监控信息	DescribeDBMetrics	YES
查看数据库参数	DescribeDBParameters	YES
查看实例性能数据	DescribeDBPerformance	YES
查看实例性能数据详情	DescribeDBPerformanceDetails	YES
查看实例资源使用情况	DescribeDBResourceUsage	YES
查看实例资源使用详情	DescribeDBResourceUsageDetails	YES
查询实例安全组信息	DescribeDBSecurityGroups	YES
获取慢查询记录详情	DescribeDBSlowLogAnalysis	YES
查询慢查询日志列表	DescribeDBSlowLogs	YES
查询实例同步模式	DescribeDBSyncMode	YES
查询临时实例	DescribeDBTmpInstances	YES
查询默认参数模板	DescribeDefaultConfigTemplate	YES
查询独享集群规格	DescribeFenceDBInstanceSpecs	YES
查询流程状态	DescribeFlow	YES
查询实例 Proxy 配置	DescribeInstanceProxyConfig	YES
查询实例 SSL 状态	DescribeInstanceSSLAttributes	YES
查询 DBA 最新检查结果	DescribeLatestCloudDBAReport	YES

查看备份日志设置	DescribeLogFileRetentionPeriod	YES
查询订单信息	DescribeOrders	YES
查询价格	DescribePrice	YES
查询项目	DescribeProjects	YES
查询项目安全组信息	DescribeProjectSecurityGroups	YES
查询实例续费价格	DescribeRenewalPrice	YES
查询售卖可用区	DescribeSaleInfo	YES
获取 SQL 日志	DescribeSqlLogs	YES
查询同步任务列表	DescribeSyncTasks	YES
查询实例升级价格	DescribeUpgradePrice	YES
查询用户任务信息	DescribeUserTasks	YES
安全组批量解绑云资源	DisassociateSecurityGroups	YES
设置权限	GrantAccountPrivileges	YES
初始化实例	InitDBInstances	YES
隔离专享实例	IsolateDedicatedDBInstance	YES
修改账号备注	ModifyAccountDescription	YES
批量设置自动续费	ModifyAutoRenewFlag	YES
设置自定义备份时间	ModifyBackupTime	YES
修改参数模板	ModifyConfigTemplate	YES
修改实例加密信息	ModifyDBEncryptAttributes	YES
修改实例名称	ModifyDBInstanceName	YES
修改云数据库安全组	ModifyDBInstanceSecurityGroups	YES
修改实例所属项目	ModifyDBInstancesProject	YES
修改数据库参数	ModifyDBParameters	YES
修改实例同步模式	ModifyDBSyncMode	YES
修改实例网络	ModifyInstanceNetwork	YES
修改备注信息	ModifyInstanceRemark	YES
修改 SSL 信息	ModifyInstanceSSLAttributes	YES
修改实例 VIP	ModifyInstanceVip	YES
修改备份日志设置	ModifyLogFileRetentionPeriod	YES
开放外网地址	OpenDBExtranetAccess	YES
续费实例	RenewDBInstance	YES
重置账号密码	ResetAccountPassword	YES
启动智能 DBA	StartSmartDBA	YES

切换实例 HA	SwitchDBInstanceHA	YES
临时实例替换原实例	SwitchRollbackInstance	YES
销毁独享实例	TerminateDedicatedDBInstance	YES
实例扩容	UpgradeDBInstance	YES
升级独享实例	UpgradeDedicatedDBInstance	YES

控制台示例

最近更新时间：2024-12-16 17:42:03

云数据库访问管理策略示例

您可以通过使用 CAM 策略让用户拥有在云数据库控制台中查看和使用特定资源的权限。该部分的示例能够使用户使用控制台的特定部分的策略。目前仅支持按策略语法创建方式配置 TDSQL MySQL 版集中式数据库支持访问管理 CAM 功能。

注意：

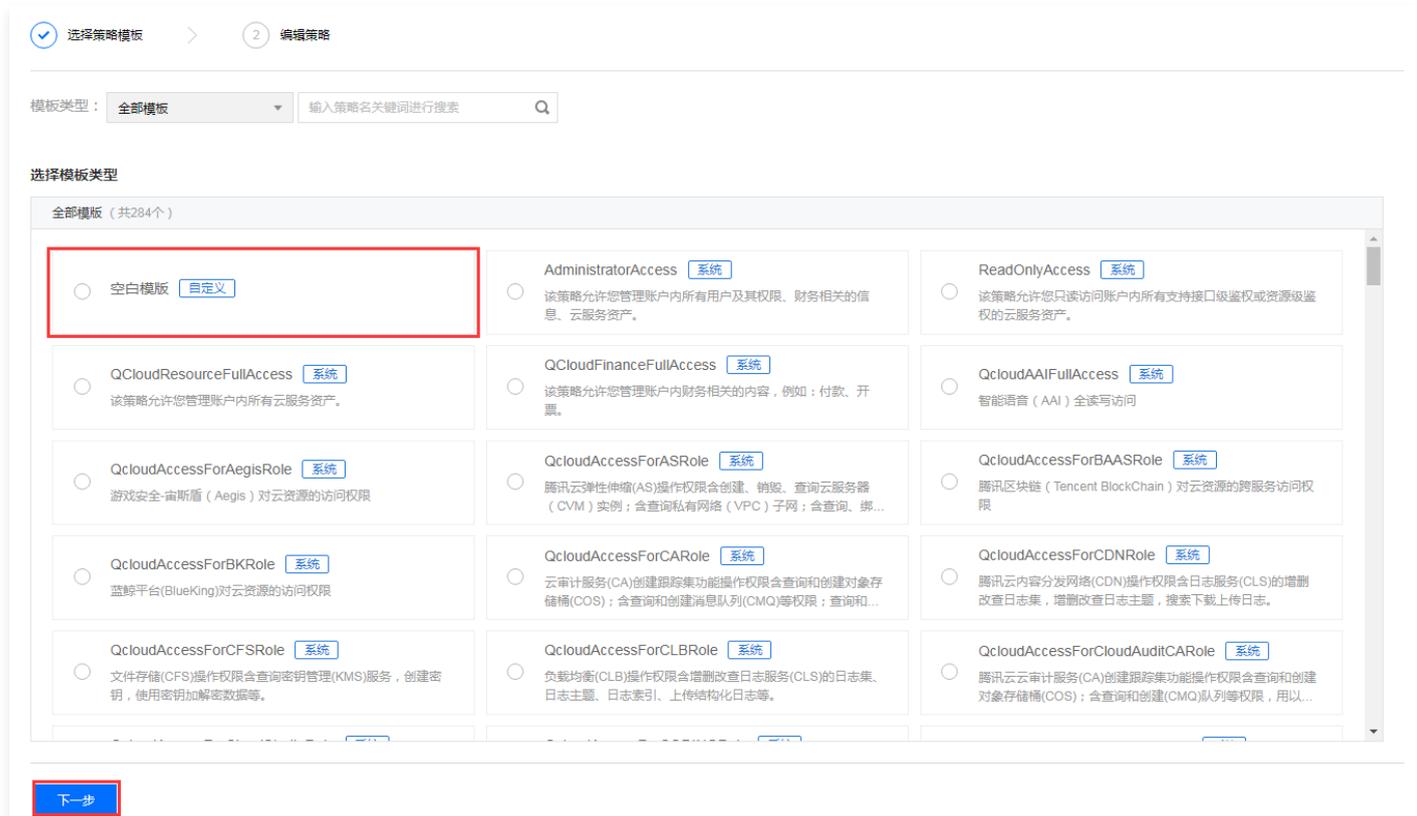
TDSQL MySQL 版集中式数据库在访问管理的接口关键词为 mariadb。

第一步、创建自定义策略语法

1. 进入 [策略语法](#) 配置页面，单击新建自定义策略。
2. 在弹出的对话框，选择按策略语法创建。



3. 选择空白模板并单击下一步。



4. 填入对应的策略语法。



第二步、关联子账号/协作者并验证

创建策略完成后，选择关联用户/组。关联完成后，更换浏览器（或主机），通过使用子账号/协作者验证是否正常。如果策略语法写作无误，您可以观察到：

- 您能正常访问预期目标产品和资源，并拥有预期的全部功能。

- 访问其他未授权产品或资源时提示“您没有权限执行此操作”。

⚠ 注意：

- 为避免多个策略语法影响，建议一次只让子账号关联一个策略。
- 修改某账号访问控制权限后，预估会有1分钟以内的延迟。

附录：常用的策略语法

放通云数据库的全部实例全部功能策略

如果您想让用户拥有创建和管理云数据库实例的权限，您可以对该用户使用名称为：QcloudMariaDBFullAccess 的策略。
策略语法如下：

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

云数据库全部实例仅查询功能策略

如果您只想让用户拥有查询云数据库实例的权限，但是不具有创建、删除和修改的权限，您可以对该用户使用名称为 QcloudMariaDBInnerReadOnlyAccess 的策略。
策略语法如下：

```
{
  "version": "2.0",
  "statement": [
    {
      "action": [
        "mariadb:Describe*"
      ],
      "resource": "*",
      "effect": "allow"
    }
  ]
}
```

以上策略是通过让用户分别对云数据库中所有以单词“Describe”开头的操作进行 CAM 策略授权来达到目的。

ⓘ 说明：

目前并未涵盖全部功能 API 接口，您可能会观察到少量操作并未纳入 CAM 管理，这属于正常现象。

授权用户拥有特定地域云数据库的操作权限策略

如果您想要授权用户拥有特定地域的云数据库的操作权限，可将以下策略关联到该用户。以下策略允许用户拥有对广州地域的云数据库机器的操作权限。

```
{
  "version": "2.0",
```

```
"statement": [
  {
    "action": "mariadb:*",
    "resource": "qcs::mariadb:ap-guangzhou:*",
    "effect": "allow"
  }
]
```

授权用户拥有若干特定地域云数据库的操作权限策略

如果您想要授权用户拥有特定地域的云数据库的操作权限，可将以下策略关联到该用户。以下策略允许用户拥有对广州成都地域的云数据库机器的操作权限。

```
{
  "version": "2.0",
  "statement": [
    {
      "action": "mariadb:*",
      "resource": "qcs::mariadb:ap-guangzhou:*", "qcs::mariadb:ap-chengdu:*",
      "effect": "allow"
    }
  ]
}
```

授权用户拥有特定云数据库的操作权限策略

如果您想要授权用户拥有特定云数据库操作权限，可将以下策略关联到该用户。以下策略允许用户拥有对 ID 为 tdsq1-xxx，广州地域的云数据库实例的操作权限。

```
{
  "version": "2.0",
  "statement": [
    {
      "action": [
        "mariadb:*"
      ],
      "resource": "qcs::mariadb:ap-guangzhou::instance/tdsq1-xxx",
      "effect": "allow"
    }
  ]
}
```

授权用户拥有若干云数据库的操作权限策略

如果您想要授权用户拥有批量云数据库操作权限，可将以下策略关联到该用户。以下策略允许用户拥有对 ID 为 tdsq1-xxx、tdsq1-yyy 广州地域的云数据库实例的操作权限，和对 ID 为 tdsq1-zzz 北京地域的云数据库实例的操作权限。

```
{
  "version": "2.0",
  "statement": [
    {
      "action": "mariadb:*",
      "resource": ["qcs::mariadb:ap-guangzhou::instance/tdsq1-xxx", "qcs::mariadb:ap-guangzhou::instance/tdsq1-yyy", "qcs::mariadb:ap-beijing::instance/tdsq1-zzz"],
      "effect": "allow"
    }
  ]
}
```

```
}
```

授权用户拥有若干云数据库的若干操作权限策略

如果您想要授权用户拥有批量云数据库操作权限，可将以下策略关联到该用户。以下策略允许用户拥有对 ID 为 tdsq1-xxx、tdsq1-yyy 广州地域的云数据库实例的若干操作权限，和对 ID 为 tdsq1-zzz 北京地域的云数据库实例的若干操作权限。

```
{
  "version": "2.0",
  "statement": [
    {
      "action": "mariadb:Describe*", "mariadb:Create*",
      "resource": ["qcs::mariadb:ap-guangzhou::instance/tdsq1-xxx", "qcs::mariadb:ap-guangzhou::instance/tdsq1-yyy", "qcs::mariadb:ap-beijing::instance/tdsq1-zzz"],
      "effect": "allow"
    }
  ]
}
```

拒绝用户拥有云数据库的创建数据库账号权限

如果您想要拒绝某用户拥有创建数据库账号权限，可做如下配置，即配置 `"effect": "deny"`。

```
{
  "version": "2.0",
  "statement": [
    {
      "action": "mariadb:CreateAccount",
      "resource": "*",
      "effect": "deny"
    }
  ]
}
```

其他自定义策略

如果您觉得预设策略不能满足您所想要的要求，您也可以创建自定义策略。自定义的策略语法如下：

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

- Action 中换成您要进行允许或拒绝的操作。
- Resource 中换成您要授权的具体资源。
- Effect 中换成允许或者拒绝。

当前控制台功能已接入 CAM 操作说明

最近更新时间：2024-05-16 15:13:21

下列操作可支持资源级权限

操作名	API 名	配置后控制台是否生效
恢复独享实例	ActiveDedicatedDBInstance	YES
绑定安全组	AssociateSecurityGroups	YES
检查 IP 状态	CheckIpStatus	YES
克隆账号	CloneAccount	YES
关闭外网地址	CloseDBExtranetAccess	YES
复制权限	CopyAccountPrivileges	YES
创建账号	CreateAccount	YES
创建参数模板	CreateConfigTemplate	YES
创建实例	CreateDBInstance	YES
回档实例	CreateTmpInstances	YES
删除账号	DeleteAccount	YES
删除参数模板	DeleteConfigTemplate	YES
删除临时实例	DeleteTmpInstance	YES
获取权限列表	DescribeAccountPrivileges	YES
查询账号列表	DescribeAccounts	YES
查询审计日志	DescribeAuditLogs	YES
查询审计规则详情	DescribeAuditRuleDetail	YES
查询审计规则列表	DescribeAuditRules	YES
查询审计策略	DescribeAuditStrategies	YES
获取自定义备份时间	DescribeBackupTime	YES
批量实例续费询价	DescribeBatchRenewalPrice	YES
查询 binlog 时间	DescribeBinlogTime	YES
查询参数配置历史	DescribeConfigHistories	YES
查询参数模板	DescribeConfigTemplate	YES
查询参数模板列表	DescribeConfigTemplates	YES
查询实例对象	DescribeDatabaseObjects	YES
查询实例 DB 名	DescribeDatabases	YES
查询实例表的列信息	DescribeDatabaseTable	YES
查询监控信息详情	DescribeDBDetailMetrics	YES
查询实例密钥信息	DescribeDBEncryptAttributes	YES

查询实例详情	DescribeDBInstanceDetail	YES
查询实例 HA 信息	DescribeDBInstanceHAInfo	YES
查询实例规格	DescribeDBInstanceSpecs	YES
查询实例列表	DescribeDBInstances	YES
获取日志列表	DescribeDBLogFiles	YES
查询监控信息	DescribeDBMetrics	YES
查看数据库参数	DescribeDBParameters	YES
查看实例性能数据	DescribeDBPerformance	YES
查看实例性能数据详情	DescribeDBPerformanceDetails	YES
查看实例资源使用情况	DescribeDBResourceUsage	YES
查看实例资源使用详情	DescribeDBResourceUsageDetails	YES
查询实例安全组信息	DescribeDBSecurityGroups	YES
获取慢查询记录详情	DescribeDBSlowLogAnalysis	YES
查询慢查询日志列表	DescribeDBSlowLogs	YES
查询实例同步模式	DescribeDBSyncMode	YES
查询临时实例	DescribeDBTmpInstances	YES
查询默认参数模板	DescribeDefaultConfigTemplate	YES
查询独享集群规格	DescribeFenceDBInstanceSpecs	YES
查询流程状态	DescribeFlow	YES
查询实例 Proxy 配置	DescribeInstanceProxyConfig	YES
查询实例 SSL 状态	DescribeInstanceSSLAttributes	YES
查询 DBA 最新检查结果	DescribeLatestCloudDBAReport	YES
查看备份日志设置	DescribeLogFileRetentionPeriod	YES
查询订单信息	DescribeOrders	YES
查询价格	DescribePrice	YES
查询项目	DescribeProjects	YES
查询项目安全组信息	DescribeProjectSecurityGroups	YES
查询实例续费价格	DescribeRenewalPrice	YES
查询售卖可用区	DescribeSaleInfo	YES
获取 SQL 日志	DescribeSqlLogs	YES
查询同步任务列表	DescribeSyncTasks	YES
查询实例升级价格	DescribeUpgradePrice	YES
查询用户任务信息	DescribeUserTasks	YES
安全组批量解绑云资源	DisassociateSecurityGroups	YES

设置权限	GrantAccountPrivileges	YES
初始化实例	InitDBInstances	YES
隔离独享实例	IsolateDedicatedDBInstance	YES
修改账号备注	ModifyAccountDescription	YES
批量设置自动续费	ModifyAutoRenewFlag	YES
设置自定义备份时间	ModifyBackupTime	YES
修改参数模板	ModifyConfigTemplate	YES
修改实例加密信息	ModifyDBEncryptAttributes	YES
修改实例名称	ModifyDBInstanceName	YES
修改云数据库安全组	ModifyDBInstanceSecurityGroups	YES
修改实例所属项目	ModifyDBInstancesProject	YES
修改数据库参数	ModifyDBParameters	YES
修改实例同步模式	ModifyDBSyncMode	YES
修改实例网络	ModifyInstanceNetwork	YES
修改备注信息	ModifyInstanceRemark	YES
修改 SSL 信息	ModifyInstanceSSLAttributes	YES
修改实例 VIP	ModifyInstanceVip	YES
修改备份日志设置	ModifyLogFileRetentionPeriod	YES
开放外网地址	OpenDBExtranetAccess	YES
续费实例	RenewDBInstance	YES
重置账号密码	ResetAccountPassword	YES
启动智能 DBA	StartSmartDBA	YES
切换实例 HA	SwitchDBInstanceHA	YES
临时实例替换原实例	SwitchRollbackInstance	YES
销毁独享实例	TerminateDedicatedDBInstance	YES
实例扩容	UpgradeDBInstance	YES
升级独享实例	UpgradeDedicatedDBInstance	YES

配置云数据库安全组

最近更新時間：2024-12-16 17:42:03

安全組是一種有狀態的包含過濾功能的虛擬防火牆，用於設置單台或多台云數據庫的網絡訪問控制，是騰訊雲提供的重要的網絡安全隔離手段。

安全組是一個邏輯上的分組，您可以將同一地域內具有相同網絡安全隔離需求的私有網絡云數據庫實例加到同一個安全組內，暫不支持基礎網絡云數據庫。云數據庫與云服务器等共享安全組列表，安全組內基於規則匹配，云數據庫不支持的規則自動不生效。

❗ 說明：

云數據庫安全組目前支持私有網絡 VPC 內網訪問的網絡控制，對於廣州、上海、北京、成都、南京、香港、重慶、新加坡、美國硅谷、首爾、日本、法蘭克福、弗吉尼亞地域開啟外網訪問的實例，安全組支持外網環境。

管理云數據庫安全組

- 登錄 [TDSQL 控制台](#)，在實例列表，單擊實例 ID，進入實例管理頁面。
- 在實例管理頁面，選擇 [數據安全性](#) > [安全組](#) 頁，可管理云數據庫安全組。

⚠ 注意：

- 云數據庫共享云服务器的安全組規則，您可以根據實際情況在安全組管理頁面匹配或調整優先級。
- 云數據庫安全組管理頁面不支持創建、刪除安全組規則本身；有創建、刪除、調整安全組規則，請參考私有網絡 [管理安全組](#)。

安全組策略

安全組策略分為允許和拒絕流量。您可以通過安全組策略對實例的入流量進行安全過濾，實例可以是私有網絡云數據庫實例。

云數據庫安全組默認策略

當前購買云數據庫且為 VPC 網絡時，可以無需關聯任何安全組，此時默認策略為“放通全部 IP 和端口”。

安全組模板

安全組支持自定義創建和模板創建，通過配置安全組規則對出入云服务器的數據包進行控制。

放通全部端口：允許全部 IP 訪問云數據庫。有一定安全風險，請謹慎操作。

安全組規則

安全組規則可控制允許到達與安全組相關聯實例的入站流量，以及允許離開實例的出站流量（從上到下依次篩選規則）。默認情況下，新建安全組將 All Drop（拒絕）所有流量。您可以隨時修改安全組的規則，新規則保存後立即生效。

對於安全組的每條規則，有以下几項內容：

- 協議端口：因為云數據庫僅能通過指定端口訪問，所以安全組設置為非數據庫訪問端口不會對本實例生效，您可以通過查看實例連接端口來進行配置，如當前訪問端口為 3306，則端口可以設置為 ALL 或者 TCP:3306。
- 授權類型：地址段（CIDR/IP）訪問；
- 來源（入站規則）或目標（出站規則），請指定以下選項之一：
 - 用 CIDR 表示法，指定的單個 IP 地址。
 - 用 CIDR 表示法，指定的 IP 地址範圍（例如，203.0.113.0/24）。
- 策略：允許或拒絕。

安全組優先級

您在實例控制台中配置的安全組優先級，數字越小優先級越高。實例綁定多個安全組時，優先級將作為判斷該實例總的安全規則的評估依據。

另外，如果實例綁定的多個安全組的最後一條策略是 **ALL Traffic 拒絕**，那麼除了優先級最低的安全組，其它安全組的最後一條策略 **ALL Traffic 拒絕** 將失效。

安全組的限制

- 安全組適用於私有網絡 [網絡環境](#) 下的云數據庫實例。
- 安全組策略目前僅對內網 IP 生效，開啟數據庫外網訪問時請配合 CVM 轉發使用，保證業務獲得最佳安全保障。

- 每个用户在每个地域每个项目下最多可设置50个安全组。
- 一个安全组进站方向、出站方向的访问策略，各最多可设定100条。由于云数据库没有主动出站流量，因此出站规则对云数据库不生效。
- 一个云数据库可以加入多个安全组，一个安全组可同时关联多个云数据库，数量无限制。

注意：

安全组内实例个数虽无限制，但不宜过多。

功能描述	数量
安全组	50个/地域
访问策略	100条/进站方向，100条/出站方向
实例关联安全组个数	无限制
安全组内实例的个数	无限制

创建、管理和删除安全组规则

云数据库共享云服务器的安全组规则，您可以根据实际情况在云数据库安全组管理页面匹配或调整优先级。

创建、管理和删除安全组规则请至 [安全组管理页面](#)，并参考文档 [管理安全组](#) 操作。

SSL 连接加密

最近更新时间：2024-12-16 17:42:03

SSL 连接加密简介

SSL 连接加密背景

当使用非加密方式连接数据库时，在网络中传输的所有信息都是明文，因此存在被非法用户窃听、篡改、冒充的三大风险；而 SSL/TLS 协议是为解决这三大风险而设计的，理论上可达到：

1. 所有信息都是加密传播，第三方无法窃听。
2. 具有校验机制，一旦被篡改，通信双方会立刻发现。
3. 配备身份证书，防止身份被冒充。

SSL 连接加密概述

SSL 协议要求建立在可靠的传输层协议（TCP）之上。SSL 协议的优势在于它是与应用层协议独立无关的，高层的应用层协议（例如：HTTP、FTP、TELNET 等）能透明地建立于 SSL 协议之上。SSL 协议在应用层协议通信之前就已经完成加密算法、通信密钥的协商及服务器认证工作。在此之后应用层协议所传送的数据都会被加密，从而保证通信的私密性。

然而，加密和解密过程需要耗费系统大量的开销，严重降低机器的性能，相关测试数据表明使用 SSL/TLS 协议传输数据的工作效率只有使用不使用协议传输的十分之一。假如为了安全保密，将一个数据库所有的数据通讯应用都启用 SSL 技术来加密，并使用 TLS 协议进行传输，那么该业务系统的性能和效率将会大大降低，而且没有这个必要，因为一般来说并不是所有数据都要求那么高的安全保密级别。

SSL 加密并不保护数据本身，而是确保了来往于数据库和服务器之间的流量安全。从某个角度来看，由于企业内网天然的安全和隔离性，管理员通常可以放心内网传输安全，并在必须使用时再进行应用 SSL 连接加密。当然，通常建议管理员利用更合理的方法来应对企业内网的安全和隔离，而非仅仅依赖于 SSL 连接加密。

相关名词

- **SSL (Secure Sockets Layer)**：安全套接层，是一种安全协议，目的是为互联网通信提供安全及数据完整性保障，使用 X.509 认证。
- **TLS (Transport Layer Security)**：安全传输层，IETF 将 SSL 标准化后的产物。TLS 可以理解为 SSL 的升级版，TLS 目前有三个版本：TLS1.0、TLS1.1、TLS1.2，目前常用的为 TLS1.2，server 配置通常三个版本均支持。
- **X.509 标准**：SSL 证书格式遵循 X.509 标准，X.509 是由国际电信联盟（ITU-T）制定的数字证书标准。X.509 给出的鉴别框架是一种基于公开密钥体制的鉴别业务密钥管理，即一个用户有两把密钥，公钥和私钥，同时该标准也规范了公开密钥认证、证书吊销列表、授权证书、证书路径验证算法等。
- **Openssl**：一个开源的加密库，由 C 语言写成，SSL/TLS 协议基于该库进行的加解密。
- **认证机构 CA (Certificate Authority)**：在 HTTPS 中是一个很重要的角色，通常称之为认证中心，从广义上讲，认证中心还应该包括证书申请注册机构 RA (Registration Authority)，它是数字证书的申请注册、证书签发的管理机构。而在公司内网，通常也可以自己搭建认证服务器，发送证书，简称“自签发证书”。
- **公钥 (Public-key)**：公共证书，由 CA 中心颁发的合法文件，可以在互联网传播。公钥证书文件的扩展名包括 crt、cer、key、der、pem、pem。公钥中包含颁发给哪个域名、公司名、加密算法、组织机构、有效期等信息。
- **私钥 (private-key)**：即通常就叫所谓的私钥，私钥在生成 CSR 文件的时候同时生产，后缀通常为 .key，由使用者自己保管，不可在互联网传播，极其重要。

配置 SSL 连接加密

说明：

- SSL 连接加密仅为连接加密，不会对数据加密。
- 由于启用 SSL 加密依赖当前 SQL 引擎版本，如果 SQL 引擎不匹配，后台会先进行**静默升级**（静默升级优先确保现有连接和实例稳定影响，业务无感知升级过程，升级过程可能历时1-2小时或更长）。

内网链路相对较安全，通常无需对连接加密；由于 SSL 加密的固有缺陷，启用 SSL 加密会存在以下问题：

- 部分客户端（含应用程序）需采用 SSL 连接加密模式。
- 会显著增加 CPU 使用率，并根据通讯数据量大小线性增加。
- 明显增加网络连接响应时间。

您可以在 [TDSQL控制台](#) 实例管理页的[数据安全](#) > [连接加密](#)中关闭 SSL (Secure Sockets Layer) 加密。

SSL 连接加密支持情况

目前已经支持以下版本的安全协议：

- TLS1.0
- TLS1.1
- TLS1.2（默认）

说明：

“标准模式”又称为“无验证模式”，即客户端不需要向服务器端发送一个有效的 X.509 证书，即可启用 SSL 加密，这是因为数据库有账号口令认证，因此通常不需要证书认证。

连接启用连接加密的实例

MySQL/MariaDB 客户端

使用 SSL 连接加密的方式连接数据库：

```
mysql -P 3306 -h 10.xx.xx.168 -u test -p --ssl
```

登录成功后，可以使用 `\s` 命令查询当前登录状态：

```
MariaDB [(none)]> \s
-----
mysql Ver 15.1 Distrib 5.5.64-MariaDB, for Linux (x86_64) using readline 5.1

Connection id:          4387887
Current database:
Current user:           readonly@172.17.0.59
SSL:                   Cipher in use is AES256-GCM-SHA384
Current pager:         stdout
Using outfile:         ''
Using delimiter:       ;
Server:                MariaDB
Server version:        10.1.9-MariaDB-V2.0R500D002-20190826-1611 Source distribution
Protocol version:      10
Connection:            172.17.0.8 via TCP/IP
Server characterset:   utf8
Db characterset:       utf8
Client characterset:   utf8
Conn. characterset:    utf8
TCP port:              3306
Uptime:                23 days 9 hours 8 min 38 sec
```

其中 `SSL: Cipher in use is AES256-GCM-SHA384` 表示该用户是使用 SSL 连接至 MySQL 服务器。

Navicat 等图形化客户端

直接选择使用 SSL，无需选择使用验证即可连接。

JDBC 等连接程序示例

如果实例开启 SSL 连接加密，但 JDBC&ODBC 没有配置 SSL 加密，业务会报 WARN 错误。

JDBC 连接串添加 `useSSL` 参数：

```
connection = DriverManager.getConnection("jdbc:mysql://ip:port/jsp_db?
useSSL=true&verifyServerCertificate=false","root","123456");
```

也可以修改在 `Properties` 对象中设置 `useSSL` 的值：

```
properties.setProperty("useSSL", "true");
```

要显示指定使用 TLS1.2 协议，可以在启动 JVM 时传递下列选项：

```
-Djavax.net.debug=all -Djdk.tls.client.protocols="TLSv1.2" -Dhttps.protocols="TLSv1.2"
```

TDE 数据加密

最近更新时间：2024-12-16 17:42:03

操作场景

TDSQL MySQL 版集中式数据库提供透明数据加密（Transparent Data Encryption，TDE）功能，透明加密指数据的加解密操作对用户透明，支持对数据文件进行实时 I/O 加密和解密，在数据写入磁盘前进行加密，从磁盘读入内存时进行解密，可满足静态数据加密的合规性要求。

本文为您介绍如何通过控制台开启数据加密功能，以及如何加密或解密数据。

限制条件

- TDE 加密功能当前仅支持 MySQL 8.0.24 及以上版本和 Percona 5.7 版本。
- 需开通 [密钥管理服务 KMS](#)。如未开通，请根据 TDE 加密功能引导进行开通。
- 需获得 KMS 密钥权限。如未开通，请根据 TDE 加密功能引导进行开通。

注意事项

- 开启 KMS 后，可能产生密钥管理服务（KMS）费用，请参见 [KMS 购买指南](#)。
- TDE 加密功能开通后无法关闭。
- 若已开通灾备/只读实例，暂时不支持 TDE 加密功能。
- 开启 TDE 加密功能后，暂时不支持创建灾备/只读实例。
- 开启 TDE 加密功能后，暂时不支持用备份文件在本地恢复数据库实例，推荐采用 [回档数据库](#) 进行恢复。
- 开启 TDE 加密功能后，可提高静态数据的安全性，但同时会影响访问加密数据库的读写性能，请根据实际情况选择开启 TDE 加密功能。
- 开启 TDE 加密功能后，会增加 CPU 资源的消耗，大约会影响5%左右的性能。

操作步骤

1. 登录 [TDSQL 控制台](#)，单击实例 ID 或操作列的管理，进入实例管理页面。
2. 在实例管理页，选择[数据安全](#) > [数据加密](#)，单击“加密状态”的开关。



3. 在弹出的对话框，开通 KMS 服务和授予 KMS 密钥权限，选择密钥后，单击确定。

开通设置

开通注意事项

1. 数据加密功能可能产生密钥管理服务 (KMS) 费用, 详见《[KMS购买指南](#)》。
2. 功能开启后, 还需进行 DDL 操作才能进行表空间数据加密或解密
3. 如需关闭本功能, 需解密全部表空间, 且关闭KMS服务后, 提交工单处理
4. 功能开启后, 当前实例将暂时屏蔽创建灾备/只读实例的能力

KMS 服务 已开通

KMS 密钥授权 已授权

选择密钥 使用自动生成的密钥 我已阅读并同意 [《云数据库服务等级协议》](#) 关于数据加密的补充申明 我已阅读并同意 [《信息安全免责声明》](#)

确定

取消

4. 开启了数据加密之后, 需要对数据库表进行 DDL 操作才能进行数据加密或者解密, 具体做法如下:

- 数据表创建时加密:

```
CREATE TABLE t1 (c1 INT) ENCRYPTION='Y'
```

- 已创建数据表加密:

```
ALTER TABLE t1 ENCRYPTION='Y'
```

- 数据表解密:

```
ALTER TABLE t1 ENCRYPTION='N'
```

监控与告警

监控功能

最近更新时间：2024-12-16 17:42:03

性能监控

为方便用户查看和掌握实例的运行信息，TDSQL MySQL 版集中式数据库提供丰富的性能监控项，用户可登录 [TDSQL 控制台](#)，在实例管理页的**系统监控**页查看各项指标。

从机监控

TDSQL MySQL 版集中式数据库支持从机性能监控，在**系统监控**页根据提示选择即可切换到从机数据。

已提供监控指标

支持最新版本腾讯云可观测平台，支持 dashboard、告警指标更丰富、指标命名更规范，可参考 [腾讯云可观测平台指标](#) 获取节点和实例监控。

指标名	单位	注意事项
CPU 利用率 (CPU Utilization)	%	MariaDB 采用灵活的 CPU 限制，允许您的实例闲时使用设备额外 CPU 资源，此时 CPU 利用率会超过100%。
缓存命中率 (Buffer Cache Hit Ratio)	%	SELECT 或预处理查询直接从内存中获取数据的比例，建议大于90%，否则需增加内存规格。
可用缓存空间 (Freeable Memory) —— 计划删除	GB	实际采集为 Innodb_buffer 的可用空间，因数据库通常采用 LRU 调度方案，正常情况下该值将趋于零；处理大事务时，该值可能为负，即数据库内存使用超过实际分配值。
磁盘空间利用率 (Storage Space Utilization)	%	当前实例的数据占用空间、日志占用空间、临时占用空间、系统文件占用空间占购买磁盘空间的比例，建议小于80%，否则需增加磁盘空间。
可用磁盘空间 (Free Storage Space)	GB	当前实例空余磁盘大小绝对值，建议大于购买磁盘空间的20%，否则需增加磁盘空间。
已用日志磁盘空间 (Binary Log Disk Usage)	GB	数据磁盘上临时存储于数据库空间大小，通常该值随着写入数据比例的变化而变化。请注意，该值并非备份系统中日志存储空间大小。
总连接数 (DB Connections)	个	客户端到数据库服务器的连接总数。
活跃连接数 (Active Connections)	个	客户端到数据库服务器的活跃连接总数。
IO 利用率 (IO Utilization)	%	MariaDB 采用灵活的 IO 限制，允许实例闲时使用设备额外 IO 资源，此时 IO 利用率会超过 100%。
SQL 总数 (SQL Throughput)	次/秒	所有 DDL、DML、DCL 的总数量。
SQL 错误数 (SQL Error Throughput)	次/秒	所有 DDL、DML、DCL 中运行错误的总数量，该值如果过大，请尽快检查业务日志。
SQL 成功数 (SQL Success Throughput)	次/秒	所有 DDL、DML、DCL 中运行成功的总数量。
慢查询数 (Slow Query)	次/秒	SQL 语句执行时间超过 long_query_time 设置值的语句数据量，详细情况可至性能优化页面查看详情。
耗时 (1ms - 5ms) 请求数 (DML Latency 1ms - 5ms)	次/秒	SQL 语句执行时间的综合统计。
耗时 (5ms - 20ms) 请求数 (DML Latency 5ms - 20ms)	次/秒	SQL 语句执行时间的综合统计。

耗时 (20ms - 30ms) 请求数 (DML Latency 20ms - 30ms)	次/秒	SQL 语句执行时间的综合统计。
耗时 (大于30ms) 请求数 (DML Latency more than 30ms)	次/秒	SQL 语句执行时间的综合统计。
总请求数 (DML Throughput)	次/秒	DML 语句的总数量。
查询数 (SELECT)	次/秒	SELECT 语句的总数量, 如果该值占比较大, 可以启用读写分离方案。
更新数 (UPDATE)	次/秒	UPDATE 语句的总数量。
插入数 (INSERT)	次/秒	INSERT 语句的总数量。
覆盖数 (REPLACE)	次/秒	REPLACE 语句的总数量。
查询覆盖数 (REPLACE_SELECT)	次/秒	replace select 语句的总数量。
删除数 (DELETE)	次/秒	delete 语句的总数量。
主从切换 (Master Switch)	次	发生主机切换到从机的情况。
主从延迟 (Replica Lag)	秒	从机与主机数据延迟, 强同步的原理是将数据写入从机 binlog 便返回事务应答, 此时数据尚未完全写入磁盘, 因此仍然会有延迟。
innodb 磁盘读页次数 (Innodb Buffer Pool Read)	次/秒	用于分析 innodb 存储引擎当前性能的指标。
innodb 缓冲池读页次数 (Innodb Buffer Pool Read Requests)	次/秒	用于分析 innodb 存储引擎当前性能的指标。
innodb 缓冲池预读页次数 (Innodb Buffer Pool Read Ahead)	次/秒	用于分析 innodb 存储引擎当前性能的指标。
innodb 执行 DELETE 行数 (Innodb Rows Deleted)	次/秒	用于分析 innodb 存储引擎当前性能的指标。
innodb 执行 INSERT 行数 (Innodb Rows Insert)	次/秒	用于分析 innodb 存储引擎当前性能的指标。
innodb 执行 Read 行数 (Innodb Rows Read)	次/秒	用于分析 innodb 存储引擎当前性能的指标。
innodb 执行 UPDATE 行数 (Innodb Rows Updated)	次/秒	用于分析 innodb 存储引擎当前性能的指标。

告警功能

最近更新时间：2024-12-16 17:42:03

TDSQL MySQL 版集中式数据库支持关键性能指标的短信、邮件、站内信等告警，您可在 [腾讯云可观测平台控制台](#) 的告警策略页进行设置。

KILL 线程

最近更新时间：2022-04-24 15:01:16

在使用数据库运行某些语句时，会因数据量太大而导致死锁，没有反应。这个时候，就需要 KILL 掉某个正在消耗资源的 query 语句即可，KILL 命令的语法格式如下：

```
KILL [CONNECTION | QUERY] thread_id
```

每个与 mysql 的连接都在一个独立的线程里运行，您可以使用 SHOW PROCESSLIST 语句查看哪些线程正在运行，并使用 KILL thread_id 语句终止一个线程。

```
mysql> show processlist;
+-----+-----+-----+-----+-----+-----+-----+-----+
| Id      | User      | Host                | db      | Command | Time | State      | Info                |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 924107  | sutest   | 10.0.0.8:38314     | NULL   | Query   | 0    | starting  | show processlist   |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 924114  | sutest   | 10.0.0.8:38318     | test   | Query   | 264  | User sleep | select sleep(20000) |
+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> kill 924114;
Query OK, 0 rows affected (0.00 sec)
```

如果您的业务有较多线程，无法准确判断哪些事务未提交，可以采用类似 SQL 进行查询线程 ID（举例）：

```
SELECT
  it.trx_id AS trx_id,
  it.trx_state AS trx_state,
  it.trx_started AS trx_started,
  it.trx_mysql_thread_id AS trx_mysql_thread_id,
  CURRENT_TIMESTAMP - it.trx_started AS RUN_TIME,
  pl.user AS USER,
  pl.host AS HOST,
  pl.db AS db,
  pl.time AS trx_run_time,
  pl.INFO as INFO
FROM
  information_schema.INNODB_TRX it,
  information_schema.processlist pl
WHERE
  pl.id=it.trx_mysql_thread_id
ORDER BY RUN_TIME DESC LIMIT 10;
```

如果您的业务有较多线程，无法准确判断哪些事务处于锁等待，可以采用类似 SQL 进行查询线程 ID（举例）：

```
SELECT
  r.trx_id waiting_trx_id,
  r.trx_mysql_thread_id waiting_thread,
  TIMESTAMPDIFF( SECOND, r.trx_wait_started, CURRENT_TIMESTAMP ) wait_time,
```

```
r.trx_query waiting_query,  
l.lock_table waiting_table_lock,  
b.trx_id blocking_trx_id,  
b.trx_mysql_thread_id blocking_thread,  
SUBSTRING( p. HOST, 1, INSTR(p. HOST, ':') - 1 ) blocking_host,  
SUBSTRING(p. HOST, INSTR(p. HOST, ':') + 1) blocking_port,  
IF (p.COMMAND = 'Sleep', p.TIME, 0) idel_in_trx,  
b.trx_query blocking_query FROM information_schema.INNODB_LOCK_WAITS w INNER JOIN  
information_schema.INNODB_TRX b ON b.trx_id = w.blocking_trx_id INNER JOIN  
information_schema.INNODB_TRX r ON r.trx_id = w.requesting_trx_id INNER JOIN  
information_schema.INNODB_LOCKS l ON w.requested_lock_id = l.lock_id LEFT JOIN information_schema.  
PROCESSLIST p ON p.ID = b.trx_mysql_thread_id ORDER BY wait_time DESC;
```

风险提示：大事务 KILL 之后，事务需要回滚，数据量较大的情况下也需等待很久，此时可以到控制台单击主从切换，将从机切换为主，以快速恢复业务。但**请务必知悉：**使用异步同步、强同步（可退化）复制方案时，由于主从数据同步有延迟，可能丢失/错乱部分数据，请谨慎操作主从切换。

参数模板与设置

最近更新时间：2025-06-19 11:07:52

参数模板概述

您可以使用数据库参数模板中的参数来管理数据库引擎配置。数据库参数组像是引擎配置值的容器，这些值可应用于一个或多个数据库实例。当您已创建一个数据库参数模板，并想在新的数据库参数模板中包含该组中的大部分自定义参数和值时，复制参数模板是一个方便的解决方案。如您想使用自己的数据库参数模板，只需创建一个新的数据库参数模板，修改所需的参数并修改您的数据库实例，即可使用新的数据库参数模板。需要注意的是，已经应用该参数模板的所有数据库实例并不会获得该数据库参数模板的所有参数更新。如您需要将新的参数应用到批量数据库实例上，可通过批量参数设置时导入模板来重新应用。

管理参数模板

参数模板支持如下功能，您可登录 [TDSQL 控制台](#)，单击相应实例 ID，进入管理页面进行配置：

- 支持默认参数模板。
- 支持新建模板，以默认参数为基础，稍加修改就能生成自定义的参数优化方案。
- 支持从配置文件 my.conf 导入生成模板。
- 支持从参数配置保存为模板。
- 单个或多个实例进行参数设置时，支持从模板导入。



参数说明

- 实例创建时，均已默认参数初始化。
- 可以通过参数模板，参数配置等入口修改实例参数。
- 不同实例参数相互隔离，互不影响。
- 为避免误操作，参数设置仅提供常见参数，如需其他参数，请 [提交工单](#) 说明“实例 ID、需添加参数名”申请添加。

参数名	是否需要重启 (0为否、1为是)	参数默认值	当前运行值 (可设置)	可修改参数值
auto_increment_increment	0	1	1	[1-65535]
auto_increment_offset	0	1	1	[1-65535]
autocommit	0	ON	ON	[ON、OFF]
character_set_server	0	utf8	utf8	[utf8、latin1、gbk、utf8mb4]
connect_timeout	0	10	10	[1-3600]
default_week_format	0	0	0	[0-7]

delay_key_write	0	ON	ON	[ON、OFF、ALL]
delayed_insert_limit	0	100	100	[1-4294967295]
delayed_insert_timeout	0	300	300	[1-3600]
delayed_queue_size	0	1000	1000	[1-4294967295]
div_precision_increment	0	4	4	[0-30]
group_concat_max_len	0	1024	1024	[4-18446744073709547520]
innodb_concurrency_tickets	0	5000	5000	[100-10000]
innodb_large_prefix	0	OFF	OFF	[OFF、ON]
innodb_lock_wait_timeout	0	50	50	[1-1073741824]
innodb_max_dirty_pages_pct	0	10	70	[10-90]
innodb_old_blocks_pct	0	37	37	[5-95]
innodb_old_blocks_time	0	1000	1000	[0-1000]
innodb_purge_batch_size	0	300	300	[1-1024]
innodb_readahead_threshold	0	56	56	[0-64]
innodb_stats_method	0	nulls_equal	nulls_equal	[nulls_equal、nulls_unequal、nulls_ignored]
innodb_stats_on_metadata	0	OFF	OFF	[ON、OFF]
innodb_stats_sample_pages	0	8	8	[1-4294967296]
innodb_strict_mode	0	OFF	OFF	[ON、OFF]
innodb_table_locks	0	ON	ON	[ON、OFF]
innodb_thread_concurrency	0	0	0	[0-128]
innodb_thread_sleep_delay	0	10000	10000	[1-3600000]
interactive_timeout	0	28800	28800	[10-86400]

key_cache_age_threshold	0	300	300	[100-4294967295]
key_cache_block_size	0	1024	1024	[512-16384]
key_cache_division_limit	0	100	100	[1-100]
lock_wait_timeout	0	5	5	[1-31536000]
log_queries_not_using_indexes	0	OFF	OFF	[ON、OFF]
long_query_time	0	1.000000	1.000000	[0.5-10]
low_priority_updates	0	OFF	OFF	[OFF、ON]
max_allowed_packet	0	134217728	1073741824	[16384-1073741824]
max_connect_errors	0	2000	2000	[1-4096]
max_connections	0	4096	4096	[1-32768]
mysam_sort_buffer_size	0	4194304	4194304	[262144-16777216]
net_buffer_length	0	16384	16384	[4096、8192、16384、32768、65536、1048576]
net_read_timeout	0	30	30	[1-3153600]
net_retry_count	0	10	10	[1-4294967295]
net_write_timeout	0	60	60	[1-3153600]
query_alloc_block_size	0	8192	8192	[1024-16384]
query_cache_limit	0	1048576	1048576	[1-1048576]
query_cache_size	0	0	0	[0-104857600]
query_cache_type	0	OFF	OFF	[OFF、ON、DEMAND]
query_prealloc_size	0	8192	8192	[8192-1048576]
slow_launch_time	0	2	2	[1-1024]
sort_buffer_size	0	2097152	2097152	[32768-1073741824]

e				
sqlasyn	0	ON	ON	[ON、OFF]
sqlasyntimeout	0	10	30	[10-100]
table_definition_cache	0	400	400	[400-2048]
table_open_cache	0	1024	10240	[400-524288]
tmp_table_size	0	33554432	33554432	[262144-67108864]
tx_isolation	0	REPEATABLE-READ	REPEATABLE-READ	[REPEATABLE-READ、SERIALIZABLE、READ-COMMITTED、READ-UNCOMMITTED]
wait_timeout	0	28800	28800	[60-259200]

数据库审计

开通数据库审计

最近更新時間：2024-12-16 17:42:03

TDSQL MySQL 版集中式数据库具备数据库审计能力，记录对数据库的访问及 SQL 语句执行情况，帮助企业进行风险控制，提高数据安全等级。

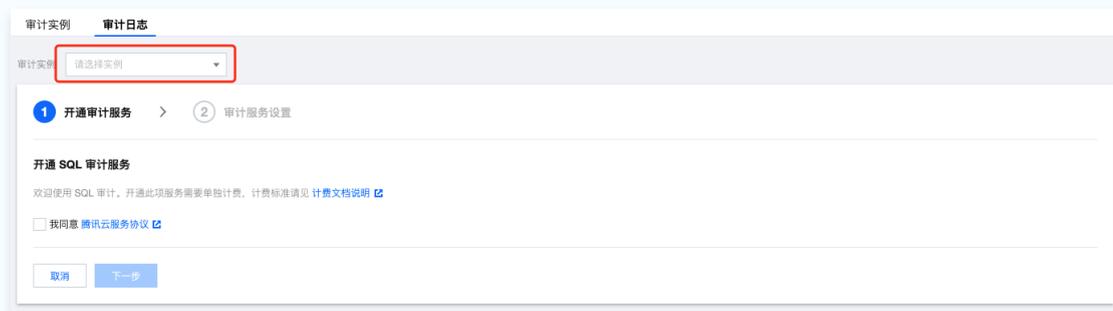
开通 SQL 审计服务

1. 登录 [TDSQL 控制台](#)，在左侧导航选择数据库审计页，在上方选择地域后，在审计实例页，单击未开启过滤未开启审计的实例。



说明：

或在[审计日志页](#)的审计实例处，直接搜索未开通的实例进行开通。



2. 在[审计实例页](#)，单击需要开通审计的实例 ID 进入开通页面，勾选同意协议，单击下一步。

3. 在 [SQL 审计服务设置页](#)，选择审计保存时长，单击开通。

说明：

- 审计日志保存时长支持选择7天、30天、3个月、6个月、1年、3年、5年。开通完后也可在控制台修改保存时长，请参见 [修改日志保存时长](#)。
- 为保证满足安全合规性对 SQL 日志保留时长的要求，建议用户选择180天及以上的保存时长。

查看审计日志

开通审计后，可在[审计日志页](#)查看对应的 SQL 审计日志，请参见 [查看审计日志](#)。

查看审计日志

最近更新时间：2024-12-16 17:42:03

查看日志

1. 登录 [TDSQL 控制台](#)，在左侧导航选择数据库审计页，在上方选择地域后，选择审计日志页。
2. 在审计日志页的审计实例处，选择已经开启审计的数据库实例，即可查看对应的 SQL 审计日志。或在审计实例页，单击已开启审计的实例 ID，跳转至审计日志页查看对应日志。

工具列表



- 在时间框，选择时间段，可查看所选时间段内相关审计结果。

说明：

搜索时间段支持选取存在数据的任意时间段进行搜索，最多展示符合条件的前60000条记录。

- 在搜索框，选择关键标签（SQL 命令、客户端 IP、数据库名、用户账号、执行时间、影响行数、返回行数等）进行搜索，可查看相关审计结果。
- 输入多组关键标签进行搜索时，可使用回车键分割关键标签。
- IP 地址支持使用 * 作为条件进行筛选。如搜索 客户端 IP: 10.0.0.0*，则匹配以 10.0.0.0 开头的 IP 地址。

日志列表

- 返回行数字段代表执行 SQL 返回的具体行数，主要用于判断 SELECT 类型 SQL 的影响。
- 影响行数字段代表执行 SQL 修改的具体行数，主要用于判断改写类 SQL 的影响。

审计日志下载

用户可以在控制台的审计日志页面单击如下图标，获取并查看完整的 SQL 审计日志。



修改日志保存时长

最近更新时间：2024-12-16 17:42:03

本文为您介绍如何在开通审计服务后，修改日志保存时长。

操作步骤

1. 登录 [TDSQL 控制台](#)，在左侧导航选择数据库审计页，在上方选择地域后，选择审计日志页。
2. 在审计日志页的右上角，单击服务设置。



3. 在弹出的对话框，修改日志保存时长，单击提交。