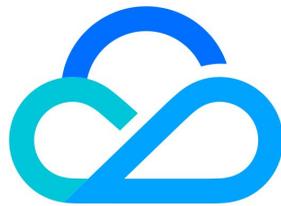


# 云数据库 MariaDB

## 快速入门

## 产品文档



腾讯云

**【版权声明】**

©2013-2018 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

**【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。

**【服务声明】**

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。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模式的承诺或保证。

## 文档目录

### 快速入门

#### 使用前须知

##### 注意事项

##### MariaDB(TDSQL)与MySQL5.6兼容性说明

#### 创建实例并连接到数据库

##### 初始化MariaDB数据库

##### 访问MariaDB数据库

##### 创建帐号

#### 迁移数据

##### 使用mysqldump导入数据

##### 数据导入至MariaDB(TDSQL)实例

# 快速入门

## 使用前须知

### 注意事项

最近更新时间：2018-05-25 17:00:26

#### 1、云数据库实例升级需要注意的事项

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。

当前我们正在对部分用户灰度开放“快速升级”能力，可能不受上述限制。

#### 2、为便于理解，云数据库文案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；

#### 3、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。

#### 4、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，如12345678、1234abcd等，会有极大的安全风险。

#### 5、MariaDB(TDSQL)回档须知。

- 目前MariaDB(TDSQL)支持数据回档，但我们建议您在回档前，务必备份一次现网关键数据。
- 回档是直接回档到临时实例中，而非现网主实例，以免回档完成以后影响现网实例。
- 临时实例可以切换为主实例，为避免混淆，切换后原主实例的所有备份数据将不可见，如需要这部分数据，可提交工单由腾讯云工作人员协助。
- 每个主实例有且仅能生成一个临时实例。
- 临时实例仅保存48小时，超过时间将自动销毁。
- 临时实例且不会主动生成全量备份，若有需要您可以利用第三方手工备份。

#### 6、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%，TDSQL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，同时也可以到TDSQL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。

#### 7、MariaDB(TDSQL)故障切换

MariaDB(TDSQL)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选举），需要您设置好程序跟TDSQL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。

#### 8、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（比如高可用、备份、安全补丁等），但是您还需要重点关注以下事情：

- 1)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。
- 2)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。

#### **9、不能更改mysql、information\_schema、performance\_schema和sysdb库中的任何数据**

**10、无法直接通过SQL语句进行帐号和赋权操作，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

#### **11、TDSQL不提供 Root 帐号**

**12、请采用InnoDB存储引擎，其他存储引擎可能导致性能变低**

**13、外网地址建议仅用于日常维护，不宜用于对接业务服务器**

# MariaDB(TDSQL)与MySQL5.6兼容性说明

最近更新时间：2018-06-06 10:32:58

## 前言

TDSQL是以MariaDB为基础开发的，我们选择MariaDB的原因不仅是因为考虑到未来MySQL的开源问题，还有就是经过多年的实际验证，对MariaDB的性能、功能和可靠性都充满信心。

期望读者们在阅读本文时，不要将“不兼容”看做是“洪水猛兽”；因为TDSQL（MariaDB）都将超越MySQL作为目标之一。“超越”意味着更强大、更丰富性能和功能，也意味着使用数据库将更标准和更灵活。因此，两个引擎的不兼容性将会一直存在。

本文所描述的内容，虽然经过大量实际项目验证和案例测试，但由于笔者能力有限，如果存在遗漏和错误，欢迎向 [vitosu@tencent.com](mailto:vitosu@tencent.com) 直接反馈。

## MariaDB(TDSQL)和开源MariaDB的兼容性

完全兼容MariaDB。

## MariaDB(TDSQL)与MySQL5.6兼容性

MariaDB(TDSQL)与MySQL5.6高度兼容，这意味着已用于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TDSQL配合使用。

- 数据文件和表定义文件是二进制兼容的；
- 所有的客户端API和协议都是兼容的；
- 所有的文件名、二进制文件、路径、端口号等都是相同的；
- 所有的连接器，包括PHP、Perl、Python、Java、.NET、Ruby、MySQL的连接器在MariaDB上都可以正常使用的，不需要进行任何改动。
- 可以使用MySQL客户端链接到TDSQL上。

## TDSQL和MySQL5.6的不兼容性

MariaDB(TDSQL)与MySQL5.6高度兼容，却有极少点不兼容；值得说明的时，MySQL各个版本之间，如5.5与5.6、5.6与5.7也并非完全兼容。

### 1.GTID不兼容

- MariaDB(TDSQL)的GTID和MySQL 5.6的GTID不能兼容，也就是说MySQL不能作为TDSQL的从库；

## 2. Binlog默认配置不同

- MariaDB(TDSQL)的Binlog默认采用row格式，而MySQL、MariaDB默认采用statement格式。

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

- 为了使CREAT 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，TDSQL中的CREAT TABLE ... SELECT命令在从库上将会被转化为CREAT OR RPLACE命令执行。这样的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：

- Mariadb10.1 没有默认值，
- mysql5.7的默认值是NULL，
- mysql5.5/5.6的默认值是空串 ""。

decimal列的默认值：mysql5.5/5.6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
----- mariadb10.1*-----
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;`

- Mysql5.5/5.6 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
- Mysql5.7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/SOME中的处理

Mysql5.5中对于  $10 \geq \text{ALL}(\text{NULL}, 1, 10)$  或者  $1 \leq \text{ALL}(\text{NULL}, 1, 10)$  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。

Mysql5.7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仅仅交换列的顺序，Mariadb允许使用inplace算法，但是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/MariaDB(TDSQL)的未定义行为

**未定义行为(undefined behavior)**的意思是，MySQL/Mariadb可以按照任意方式实现这种功能和行为，而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和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，TDSQL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。未定义行为官方说明：<https://mariadb.com/kb/en/mariadb/mariadb-vs-mysql-compatibility/>

### 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 ) 一般是按照大小写无关的方式排序的：这意味着对于除了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对下列面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

1. 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；
2. 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()的返回值只有在查询语句中使用了UNION ALL时候才是精确的。

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的查询语句不实用LIMIT子句那么 SQL\_CALC\_FOUND\_ROWS 关键字就被忽略了，FOUND\_ROWS()返回的就是执行UNION时候创建的临时表中的行数。

## 4.4 Lock tables 语句的上锁顺序不同

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照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，您写一个 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，然后再按照那个拍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），并且这些错误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

MySQL5.5 中YEAR与date类型的变量在比较时会把date类型转为year类型进行对比，即“2011-01-01”被转为2011。

MySQL5.7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mysql5.6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类型时候，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mysql5.5/5.6/5.7中它返回空串，但是mariadb10.1. 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:
  - mysql5.5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
  - mysql5.7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myisan，他都会是的操作成功
- 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，
  - mysql5.5/5.6使用heap存储引擎时，忽略这个未知字符
  - mariadb10.1和mysql5.7 把它用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MariaDB10.1/MySQL5.7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' (6)) AS DECIMAL(20,6));
+-----+
| CAST(CAST('10:11:12.098700' AS TIME(6)) AS DECIMAL(20,6)) |
+-----+
| 101112.098700 |
+-----+
```

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;
```

Mysql5.5/5.6返回NULL, Mariadb10.1和Mysql5.7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
- 而mysql5.7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                    | TDSQL ( MariaDB 10.1 )  | MySQL 5.6  |
|------------------------|---|--|
| old_passwords          | OFF   | 0  |
| tmpdir                 | /tmp/5cXm2hHsWi/mysqlid.1   | /data/home/tdengine/dongzhi/src/mysql-server-5.6/build_dongzhi/mysql-test/var/tmp/mysqlid.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/mysqlid.1/mysqlid-slow.log | /data/home/tdengine/dongzhi/src/mysql-server-5.6/build_dongzhi/mysql-test/var/mysqlid.1/mysqlid-slow.log |
| table_definition_cache | 400   | 1400   |
| datadir                | /data/home/tdengine/dongzhi/src/tdsql-mariadb-10.1.9-release1/build_dongzhi/mysql-test/var/mysqlid.1/data/            | /data/home/tdengine/dongzhi/src/mysql-server-5.6/build_dongzhi/mysql-test/var/mysqlid.1/data/            |
| pid_file               | /data/home/tdengine/dongzhi/src/tdsql-mariadb-10.1.9-release1/build_dongzhi/mysql-test/var/run/mysqlid.1.pid          | /data/home/tdengine/dongzhi/src/mysql-server-5.6/build_dongzhi/mysql-test/var/run/mysqlid.1.pid          |
| max_seeks_for_key      | 4294967295  | 18446744073709500000   |
| slave_load_tmpdir      | /tmp/5cXm2hHsWi/mysqlid.1   | /data/home/tdengine/dongzhi/src/mysql-server-5.6/build_dongzhi/mysql-test/var/tmp/mysqlid.1              |
| secure_file_priv       | /data/home/tdengine/dongzhi/src/t   | /data/home/tdengine/dongzhi/src/   |

|   |   |  |
|---|---|--|
|   | dsql-mariadb-10.1.9-release1/build_dongzhi/mysql-test/var/  | 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_statements_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,<br>index_merge_union=on,  | index_merge=on,<br>index_merge_union=on,   |

|                                |  |  |
|--------------------------------|--|--|
|                                | index_merge_sort_union=on,<br>index_merge_intersection=on,<br>index_merge_sort_intersection=off,<br>engine_condition_pushdown=off,<br>index_condition_pushdown=on,<br>derived_merge=on,<br>derived_with_keys=on,<br>firstmatch=on,<br>loosescan=on,<br>materialization=on,<br>in_to_exists=on,<br>semijoin=on,<br>partial_match_rowid_merge=on,<br>partial_match_table_scan=on,<br>subquery_cache=on,<br>mrr=off,<br>mrr_cost_based=off,<br>mrr_sort_keys=off,<br>outer_join_with_cache=on,<br>semijoin_with_cache=on,<br>join_cache_incremental=on,<br>join_cache_hashed=on,<br>join_cache_bka=on,<br>optimize_join_buffer_size=off,<br>table_elimination=on,<br>extended_keys=on,<br>exists_to_in=on | index_merge_sort_union=on,<br>index_merge_intersection=on,<br>engine_condition_pushdown=on,<br>index_condition_pushdown=on,<br>mrr=on,<br>mrr_cost_based=on,<br>block_nested_loop=on,<br>batched_key_access=off,<br>materialization=on,<br>semijoin=on,<br>loosescan=on,<br>firstmatch=on,<br>subquery_materialization_cost_based=on,<br>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/mysqld.1/mysqld.log   | /data/home/tdengine/dongzhi/src/mysql-server-5.6/build_dongzhi/mysql-test/var/mysqld.1/mysqld.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/mysqld.1.sock  | /data/home/tdengine/dongzhi/src/mysql-server-5.6/build_dongzhi/mysql-test/var/tmp/mysqld.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TDSQL ( 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
- autoremove\_relaylog 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\_mtfush\_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\_mtfush 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 仅存在与mysql5.6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

# 创建实例并连接到数据库 初始化MariaDB数据库

最近更新时间：2017-12-22 09:43:15

在此步骤中，我们将对已经购买的MariaDB数据库执行初始化操作。

1. 在[腾讯云控制台](#)的左上角，单击【云产品】下的【关系型数据库】，进入数据库产品页面。



2. 在关系型数据库页面中，单击【TDSQL(MariaDB)】下的【实例列表】，找到目标地域（此例中以广州为例）中要操作的状态为“未初始化”的MariaDB数据库实例。



3. 单击【初始化】对要操作的MariaDB 数据库实例执行初始化。



4. 配置初始化参数，然后单击【确定】开始初始化。

- 支持的字符集：选择MySQL数据库支持的字符集。

- **表名大小写敏感**：表名是否大小写敏感，默认是。
- **开启强同步**：开启强同步可以保证在主机故障时备机数据的一致性。默认为不开启，即同步方式为异步同步。
- **innodb\_page\_size**：该数值为Innodb索引数据页长度，MariaDB默认值为16K。修改该值将影响索引创建，该值越小，性能越好。

### 实例初始化 ×

**支持字符集**      utf8    latin1    gbk    utf8mb4  
若字符集设置不当会导致数据库导入发生错误

**表名大小写敏感**      ×

**开启强同步**      ×  
关闭强同步将默认设置同步方式为异步同步，主机故障时，备机数据一致性可能无法保证。

**innodb\_page\_size**      4KB    8KB    16KB  
设置INNODB引擎数据页大小，初始化后将不能修改；MySQL默认值为16KB

确定
取消

5. 目标MariaDB实例的状态变为“**运行中**”，说明已初始化成功。

| TDSQL 默认项目 ▾ 华南地区（广州） ▾             |                              |        |        |           |                 |              |                |             |   |
|-------------------------------------|------------------------------|--------|--------|-----------|-----------------|--------------|----------------|-------------|---|
| + 新建                                |                              | 更多操作 ▾ |        | 请输入实例名称   |                 |              |                |             |   |
| <input type="checkbox"/>            | ID/实例名                       | 运行状态   | 实例类型 ▾ | 实例版本      | 规格              | 所属网络         | 内网地址           | 到期时间        | 操作                                      |
| <input checked="" type="checkbox"/> | tdsql-6gy3mopk<br>tdsql11045 | 运行中    | 主实例    | 标准版(一主一从) | 2G内存<br>10G存储空间 | monotest-... | 10.30.0.7:3306 | 2017-06-... | <a href="#">管理</a>   <a href="#">升级</a> |

# 访问MariaDB数据库

最近更新时间：2018-10-12 11:53:36

连接到MariaDB数据库的方式有两种：

- 内网访问：使用腾讯云中一台与MariaDB数据库实例网络相通的CVM来访问MariaDB数据库实例的内网地址。（注意：此台CVM需要与数据库处于某个VPC下的同一个子网中，关于VPC的更多信息请查看[VPC概述](#)。）
- 外网访问：在外网的Windows或者Linux主机中，安装数据库客户端来访问腾讯云中的MariaDB数据库实例的外网地址。

**安全提示：**外网访问需要开启数据库实例的外网地址，从而使您的数据库服务暴露在公网上，此操作可能导致数据库被入侵或攻击。建议您使用内网访问的方式来登录数据库。

无论从内网还是外网访问，都需要先创建帐号。

## 创建帐号

1. 在[腾讯云控制台](#)中，进入菜单【云产品】-【关系型数据库】-【TDSQL(MariaDB)】-【实例列表】，单击运行中的MariaDB数据库实例的ID名，进入详情页。



| ID/实例名   | 运行状态 | 实例类型 | 实例版本      | 规格              | 所属网络        | 内网地址               | 到期时间        | 操作                                      |
|--|------|------|-----------|-----------------|-------------|--------------------|-------------|---|
| <span style="border: 1px solid red; padding: 2px;">tdsql-6gy3mopk</span><br>tdsql11045 | 运行中  | 主实例  | 标准版(一主一从) | 2G内存<br>10G存储空间 | monotest... | 10.30.0.7:3306     | 2017-06-... | <a href="#">管理</a>   <a href="#">升级</a> |
| tdsql-a7q2xonk<br>tdsql11040   | 运行中  | 主实例  | 标准版(一主一从) | 2G内存<br>10G存储空间 | 基础网络        | 10.66.241.222:3306 | 2017-06-... | <a href="#">管理</a>   <a href="#">升级</a> |

2. 在[帐号管理](#)页单击【创建帐号】，并设置帐号的相关权限。此处以帐号test123为例，开通全部权限。创建帐号，设置密码，并单击【确定】。

创建帐号
×

帐号名:\*   
请输入用户名

创建为只读帐号:\*  是  否  
如果选是，您可以在点击确定后，设置只读帐号的参数

主机:\*   
只有%代表所有IP均可访问，可能存在安全问题，请谨慎选择

设置密码:\*   
密码必须采用数字、英文和字符[除单双引号、分号和空格]至少三种类型，长度为8~32位

确认密码:\*

备注:   
请输入备注说明，最多256个字符

设置帐号test123拥有的权限，单击【保存配置】。

设置权限
×

帐号名称 test123@%

设置数据库权限 [刷新](#)
[重置](#)

|   |  |  |
|---|--|--|
| <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 5px;">全部权限</div> <div style="border: 1px solid #ccc; padding: 2px;">+ 对象级别权限</div> | <input checked="" type="checkbox"/> SELECT<br><input checked="" type="checkbox"/> UPDATE<br><input checked="" type="checkbox"/> CREATE<br><input checked="" type="checkbox"/> REFERENCES<br><input checked="" type="checkbox"/> ALTER<br><input checked="" type="checkbox"/> LOCK TABLES<br><input checked="" type="checkbox"/> CREATE VIEW<br><input checked="" type="checkbox"/> CREATE ROUTINE<br><input checked="" type="checkbox"/> EVENT<br><input checked="" type="checkbox"/> 全部 | <input checked="" type="checkbox"/> INSERT<br><input checked="" type="checkbox"/> DELETE<br><input checked="" type="checkbox"/> DROP<br><input checked="" type="checkbox"/> INDEX<br><input checked="" type="checkbox"/> CREATE TEMPORARY TABLES<br><input checked="" type="checkbox"/> EXECUTE<br><input checked="" type="checkbox"/> SHOW VIEW<br><input checked="" type="checkbox"/> ALTER ROUTINE<br><input checked="" type="checkbox"/> TRIGGER |
|---|--|--|

保存设置
之后设置

查看帐号权限的配置是否正确，然后单击**关闭**完成配置。



如需了解更多关于帐号创建的信息，请参考[帐号创建](#)。

## 访问数据库

连接到MariaDB数据库的方式有两种：

- 内网访问：使用腾讯云中一台与MariaDB数据库实例网络相通的CVM来访问MariaDB数据库实例的内网地址。（注意：此台CVM需要与数据库处于某个VPC下的同一个子网中，关于VPC的更多信息请查看[VPC概述](#)。）
- 外网访问：在外网的Windows或者Linux主机中，安装数据库客户端来访问腾讯云中的MariaDB数据库实例的外网地址。

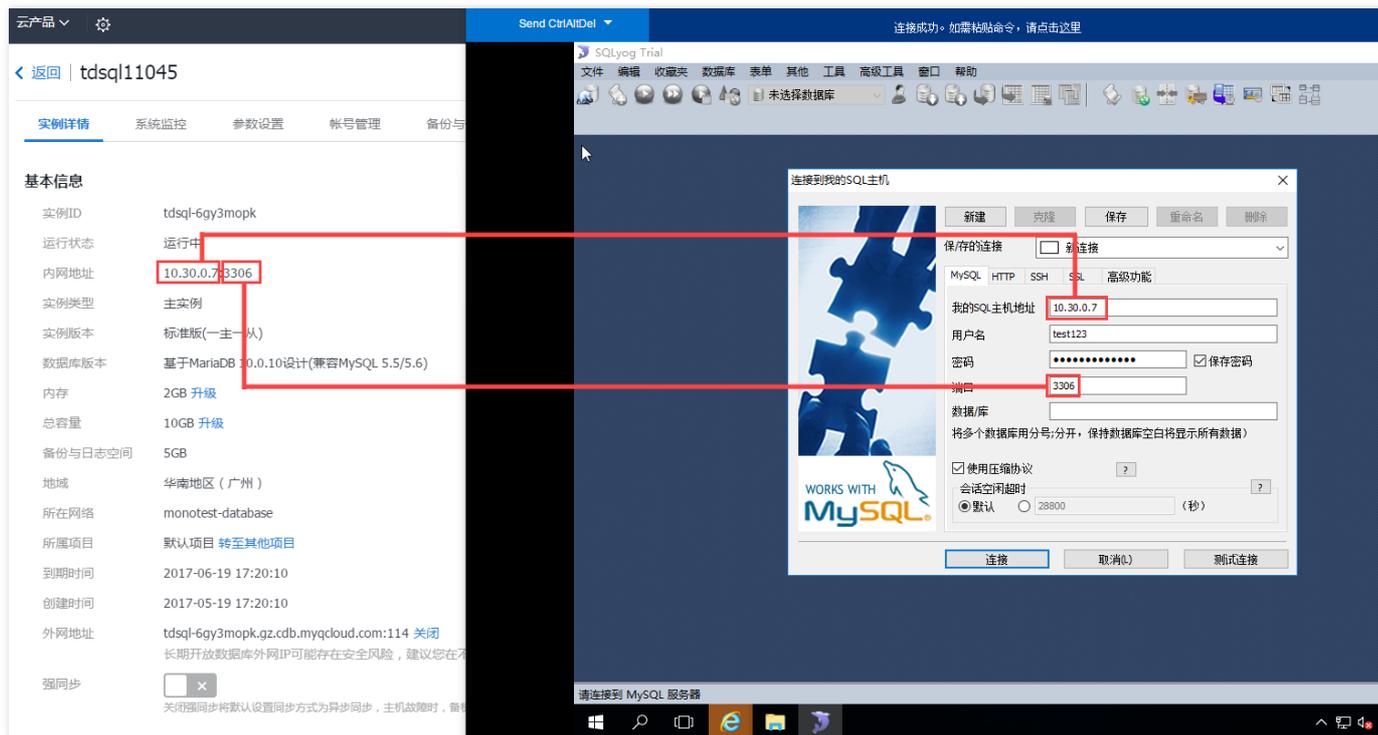
### 内网访问

1. 登录到与此数据库实例属于同一个可用区的网络可达的CVM主机，关于登录CVM主机请查看[Windows CVM入门](#)或[Linux CVM入门](#)。网络可达是指此CVM主机与MariaDB数据库实例都处于基础网络之中，或者处于同一个VPC中。
2. 请根据CVM的操作系统选择推荐的连接方式。

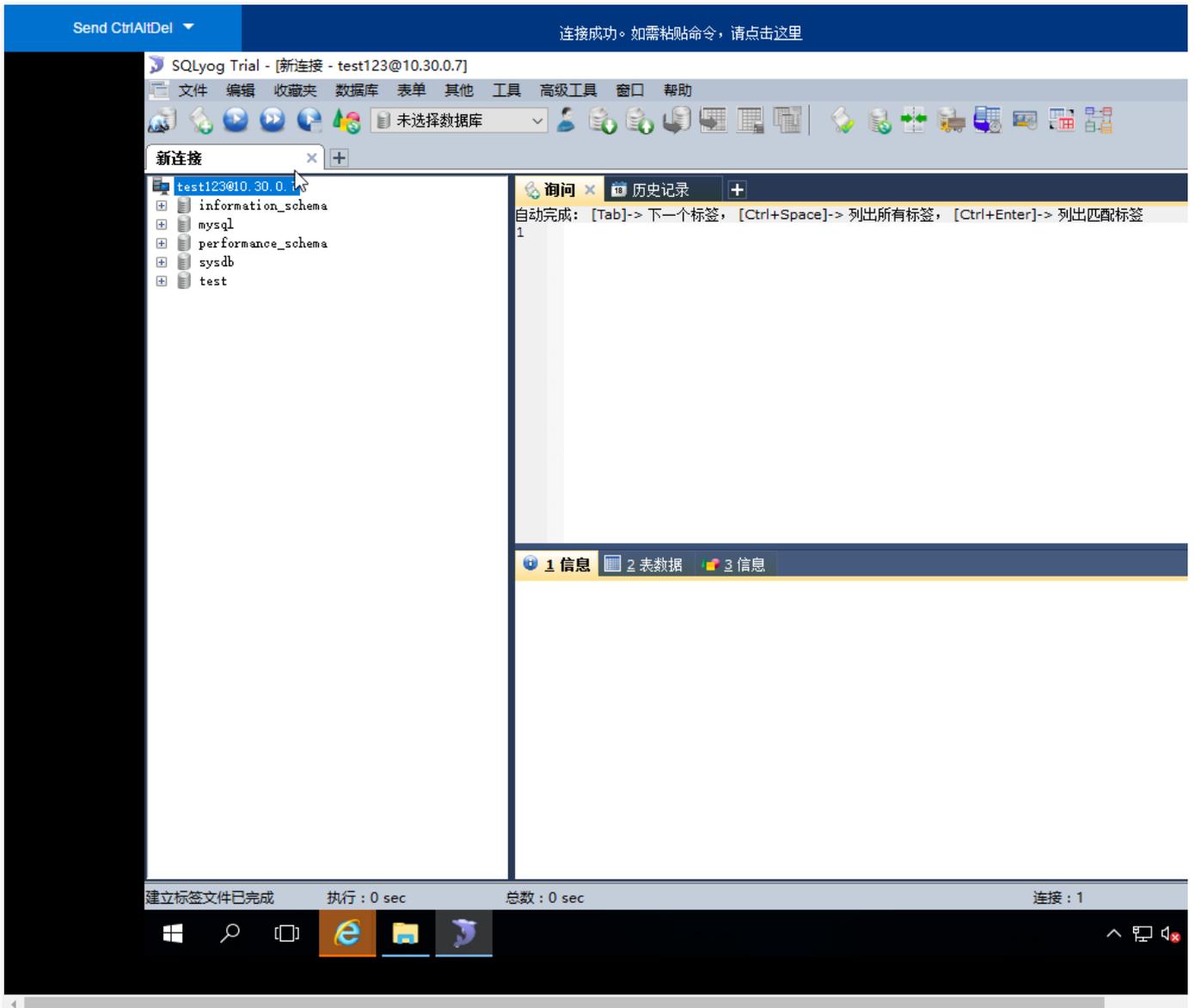
#### -从Windows系统登录

- 1). 下载并安装Mariadb的客户端。此步骤中我们推荐您下载sqlyog，官网地址如下：<https://www.webyog.com/>。
- 2). 打开sqlyog，输入输入MariaDB数据库实例的内网IP和端口号，数据库帐号以及密码。
  - 我的SQL主机地址：此例中输入10.30.0.7。
  - 用户名：用前文中创建的用户名test123。
  - 密码：用户test123对应的密码。

- 端口：此例中输入3306。



3). 登录成功的界面如图所示，在此页面上您可以看到MariaDB数据库的各种模式和对象，您可以开始创建表，进行数据插入和查询等操作。



**-从Linux系统登录**

1). 以CentOS 7.2 64位系统的CVM为例，利用CentOS自带的包管理软件Yum去腾讯云的镜像源下载安装MySQL客户端。

相关命令为：

```
yum install mysql
```

图示如下：

```

CentOS Linux 7 (Core)
Kernel 3.10.0-327.36.3.el7.x86_64 on an x86_64

UM_135_34_centos login: root 登录到CVM
Password:
[root@UM_135_34_centos ~]# yum install mysql 安装MySQL客户端
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
Resolving Dependencies
--> Running transaction check
--> Package mariadb.x86_64 1:5.5.52-1.el7 will be installed
--> Processing Dependency: mariadb-libs(x86-64) = 1:5.5.52-1.el7 for package: 1:mariadb-5.5.52-1.el7.x86_64
--> Running transaction check
--> Package mariadb-libs.x86_64 1:5.5.50-1.el7_2 will be updated
--> Package mariadb-libs.x86_64 1:5.5.52-1.el7 will be an update
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package                Arch          Version           Repository        Size
=====
Installing:
mariadb                x86_64        1:5.5.52-1.el7   os                8.7 M
Updating for dependencies:
mariadb-libs           x86_64        1:5.5.52-1.el7   os                761 k
=====

Transaction Summary
=====
Install 1 Package
Upgrade ( 1 Dependent package)

Total download size: 9.5 M
Is this ok [y/d/N]: y
Downloading packages:
Delta RPMs disabled because /usr/bin/applydeltarpm not installed.
(1/2): mariadb-libs-5.5.52-1.el7.x86_64.rpm | 761 kB 00:00:00
(2/2): mariadb-5.5.52-1.el7.x86_64.rpm | 8.7 MB 00:00:01
-----
Total | 8.1 MB/s | 9.5 MB 00:00:01
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Updating   : 1:mariadb-libs-5.5.52-1.el7.x86_64 | 1/3
  Installing : 1:mariadb-5.5.52-1.el7.x86_64 | 2/3
  Cleanup    : 1:mariadb-libs-5.5.50-1.el7_2.x86_64 | 3/3
Installed:
  mariadb.x86_64 1:5.5.52-1.el7

Dependency Updated:
  mariadb-libs.x86_64 1:5.5.52-1.el7

Complete! 此提示说明MySQL客户端安装完成
[root@UM_135_34_centos ~]#

```

2). 使用MySQL命令行工具登录到MariaDB数据库。

相关命令为：

```
mysql -h hostname -u username -p
```

请将hostname替换为目标MariaDB数据库实例的内网IP地址，将username替换为之前创建的用户test123，并在提示**Enter password**：后输入账户test123对应的密码。

本例中hostname为10.30.0.7。

```

[root@UM_0_9_centos ~]# mysql -h 10.30.0.7 -u test123 -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 989772270
Server version: 10.0.10-proxy Source distribution

Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
MySQL [(none)]>
    
```

登录MariaDB数据库实例

此提示说明已成功登录到MariaDB

3). 在MySQL>提示符下可以发送SQL语句到要执行的MariaDB服务器，具体命令行请参考此网站：

<https://dev.mysql.com/doc/refman/5.7/en/mysql-commands.html>

下图中以 show databases ; 为例。

```

MySQL [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| test |
+-----+
4 rows in set (0.00 sec)
    
```

## 外网访问

**安全提示：**外网访问需要开启数据库实例的外网地址，从而使您的数据库服务暴露在公网上，此操作可能导致数据库入侵或攻击。

1. 获取数据库的外网地址。

1). 单击运行中的MariaDB数据库实例的ID名，进入详情页。

| ID/实例名   | 运行状态 | 实例类型 | 实例版本      | 规格              | 所属网络        | 内网地址               | 到期时间        | 操作      |
|--|------|------|-----------|-----------------|-------------|--------------------|-------------|---------|
| <span style="border: 1px solid red; padding: 2px;">tdsql-6gy3mopk</span><br>tdsql11045 | 运行中  | 主实例  | 标准版(一主一从) | 2G内存<br>10G存储空间 | monotest... | 10.30.0.7:3306     | 2017-06-... | 管理   升级 |
| tdsql-a7q2xonk<br>tdsql11040   | 运行中  | 主实例  | 标准版(一主一从) | 2G内存<br>10G存储空间 | 基础网络        | 10.66.241.222:3306 | 2017-06-... | 管理   升级 |

2). 在实例详情页单击外网地址后的【打开】，开启此数据库实例的外网地址。

返回 | tdsq11045
转至其他项目 回档

实例详情
系统监控
参数设置
帐号管理
备份与恢复
性能优化

### 基本信息

|         |   |
|---------|---|
| 实例ID    | tdsql-6gy3mopk  |
| 运行状态    | 运行中   |
| 内网地址    | 10.30.0.7:3306  |
| 实例类型    | 主实例   |
| 实例版本    | 标准版(一主一从)   |
| 数据库版本   | 基于MariaDB 10.0.10设计(兼容MySQL 5.5/5.6)  |
| 内存      | 2GB <a href="#">升级</a>  |
| 总容量     | 10GB <a href="#">升级</a>   |
| 备份与日志空间 | 5GB   |
| 地域      | 华南地区(广州)  |
| 所在网络    | monotest-database   |
| 所属项目    | 默认项目 <a href="#">转至其他项目</a>   |
| 到期时间    | 2017-06-19 17:20:10   |
| 创建时间    | 2017-05-19 17:20:10   |
| 外网地址    | <span style="border: 1px solid red; padding: 2px;">打开</span>                          |
| 强同步     | <input type="checkbox"/> <span style="border: 1px solid #ccc; padding: 2px;">×</span> |

关闭强同步将默认设置同步方式为异步同步，主机故障时，备机数据一致性可能无法保证。

提示
×

1. 外网地址启用后，您可以使用系统分配的域名和端口通过外网访问腾讯云TDSQL;
2. 生效时间大概需要5分钟;
3. TDSQL外网访问仅用于开发或辅助管理数据库，业务访问请使用内网访问。

确定
取消

3). 查看此数据库实例的外网地址。

下图中，此数据库的外网域名为：`tdsql-6gy3mopk.gz.cdb.myqcloud.com`，端口号为：114

< 返回 | tdsq11045 转至其他项目 回档

实例详情 系统监控 参数设置 帐号管理 备份与恢复 性能优化

### 基本信息

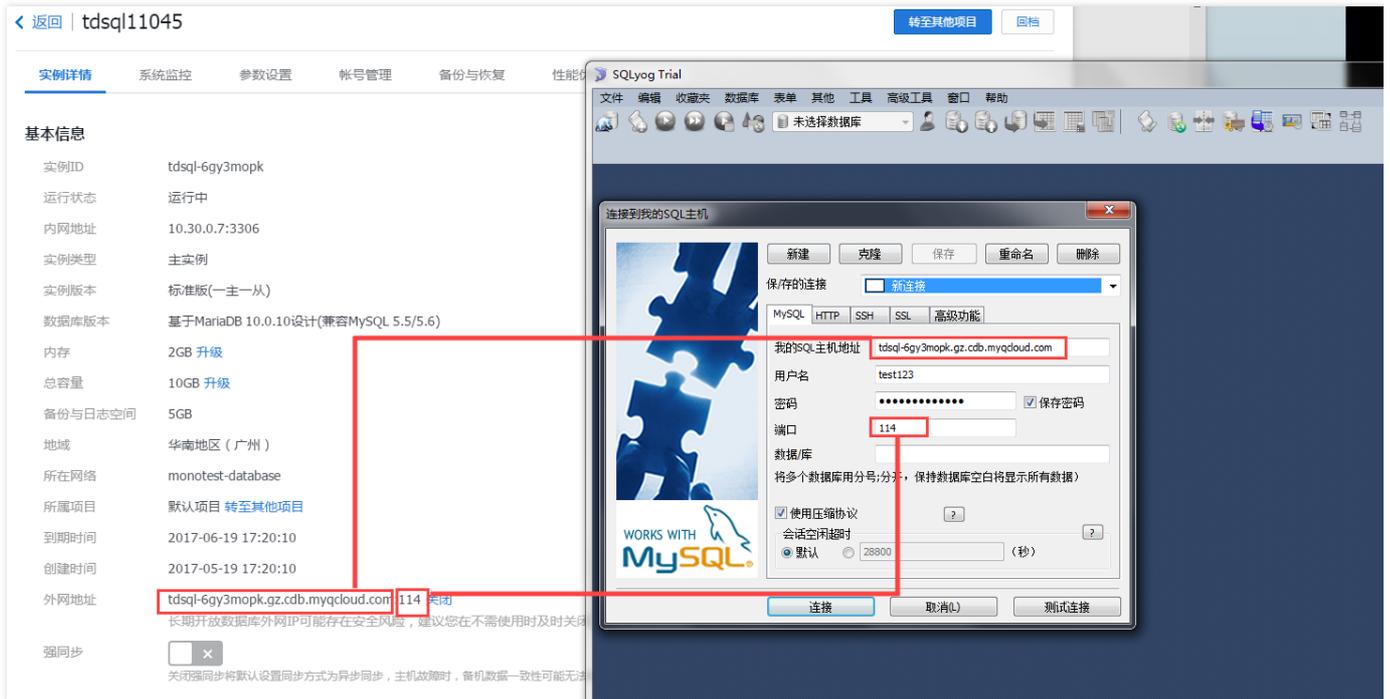
|         |  |
|---------|--|
| 实例ID    | tdsql-6gy3mopk   |
| 运行状态    | 运行中  |
| 内网地址    | 10.30.0.7:3306   |
| 实例类型    | 主实例  |
| 实例版本    | 标准版(一主一从)  |
| 数据库版本   | 基于MariaDB 10.0.10设计(兼容MySQL 5.5/5.6)   |
| 内存      | 2GB <a href="#">升级</a>   |
| 总容量     | 10GB <a href="#">升级</a>  |
| 备份与日志空间 | 5GB  |
| 地域      | 华南地区(广州)   |
| 所在网络    | monotest-database  |
| 所属项目    | 默认项目 <a href="#">转至其他项目</a>  |
| 到期时间    | 2017-06-19 17:20:10  |
| 创建时间    | 2017-05-19 17:20:10  |
| 外网地址    | <span style="border: 1px solid red; padding: 2px;">tdsql-6gy3mopk.gz.cdb.myqcloud.com:114</span> <a href="#">关闭</a><br><small>长期开放数据库外网IP可能存在安全风险，建议您在不需使用时及时关闭。</small> |
| 强同步     | <input type="checkbox"/> <span style="border: 1px solid gray; padding: 2px;">×</span><br><small>关闭强同步将默认设置同步方式为异步同步，主机故障时，备机数据一致性可能无法保证。</small>                         |

## 2. 登录到数据库

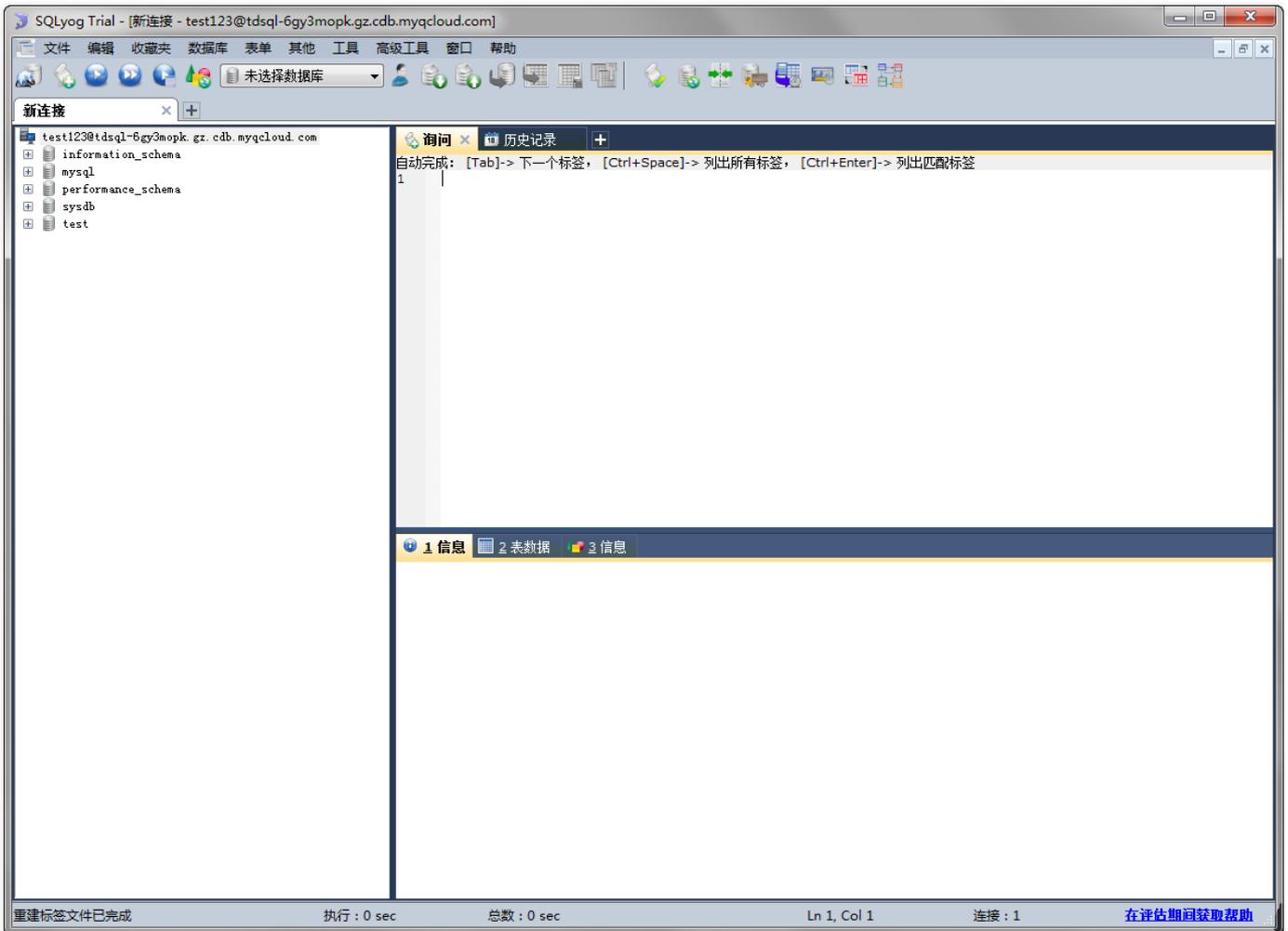
### -从Windows系统登录

- 1). 下载一个MariaDB客户端并安装。此步骤中我们推荐您下载sqlyog，官网地址如下：<https://www.webyog.com/>。
- 2). 打开sqlyog，输入输入MariaDB数据库实例的外网域名和端口号，数据库帐号以及密码。
  - 我的SQL主机地址：此例中输入tdsql-6gy3mopk.gz.cdb.myqcloud.com。
  - 用户名：用前文中创建的用户名test123。
  - 密码：用户test123对应的密码。

- 端口：此例中输入114。



3). 登录成功的界面如图所示，在此页面上您可以看到MariaDB数据库的各种模式和对象，您可以开始创建表，进行数据插入和查询等操作。



### -从Linux系统登录

1). 以CentOS 7.2 64位系统的CVM为例，去官网下载安装MySQL客户端。具体命令为：

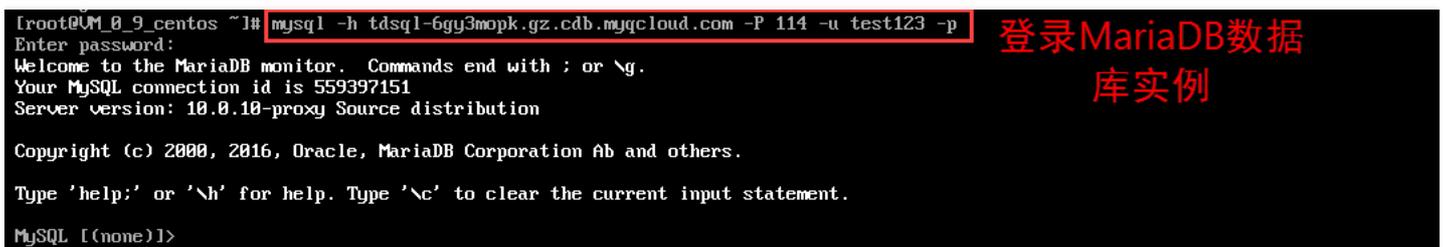
```
yum intall mysql
```

2). 使用MySQL命令行工具登录到MariaDB数据库。相关命令为：

```
mysql -h hostname -P port -u username -p
```

请将hostname替换为目标MariaDB数据库实例的外网IP地址，将username替换为之前创建的用户test123，并在提示**Enter password**：后输入账户test123对应的密码。

本例中hostname为tdsql-6gy3mopk.gz.cdb.myqcloud.com，port为114。



3). 在MySQL>提示符下可以发送SQL语句到要执行的MariaDB服务器，具体命令行请参考此网站：

<https://dev.mysql.com/doc/refman/5.7/en/mysql-commands.html>

下图中以 `show databases ;` 为例。

```
MySQL [(none)]> show databases;
+-----+
| Database           |
+-----+
| information_schema |
| mysql              |
| performance_schema |
| test               |
+-----+
4 rows in set (0.00 sec)
```

# 创建帐号

最近更新时间：2018-06-28 18:09:24

您可以通过“创建帐号”、“获取实例地址”两步轻松连接MariaDB(TDSQL)实例。

## 1. 进入账号管理页面，选择账号管理功能

在您正式开始使用MariaDB(TDSQL)服务之前，您需要进入MariaDB(TDSQL)实例的管理页面，创建用于登录MariaDB(TDSQL)的数据库帐号，并设置相应的访问权限。



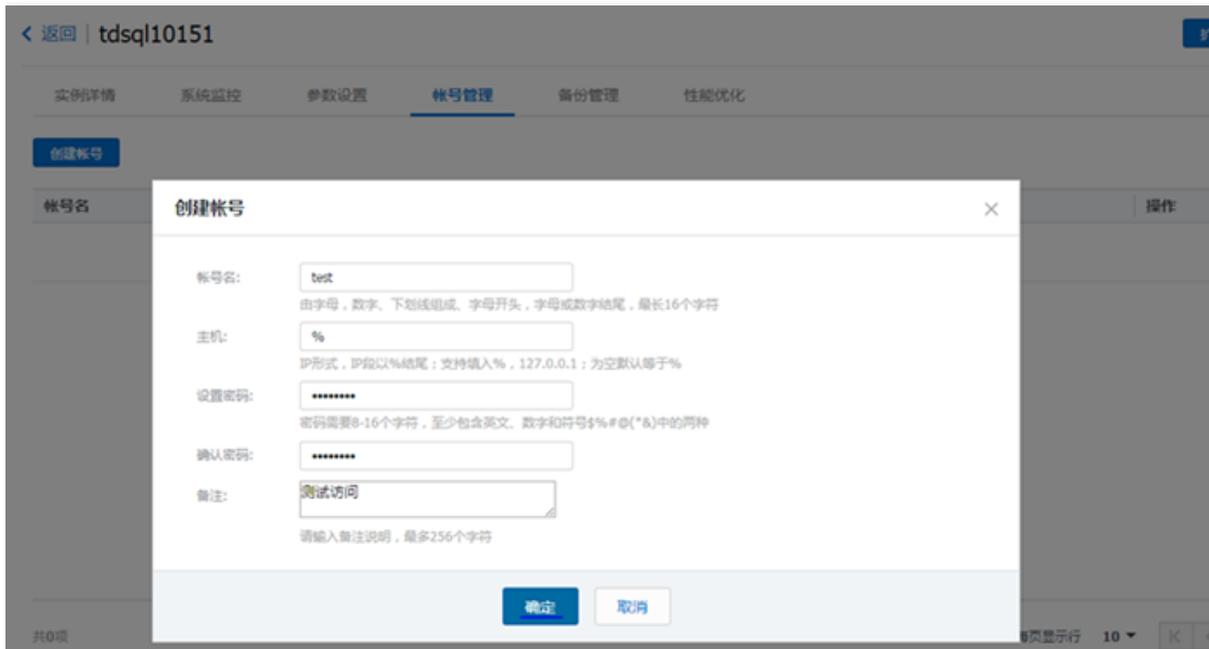
您可以在管理页面看到“帐号管理”，如下图所示：



## 2. 创建账号

这里的帐号由帐号名、主机、密码三部分组成；

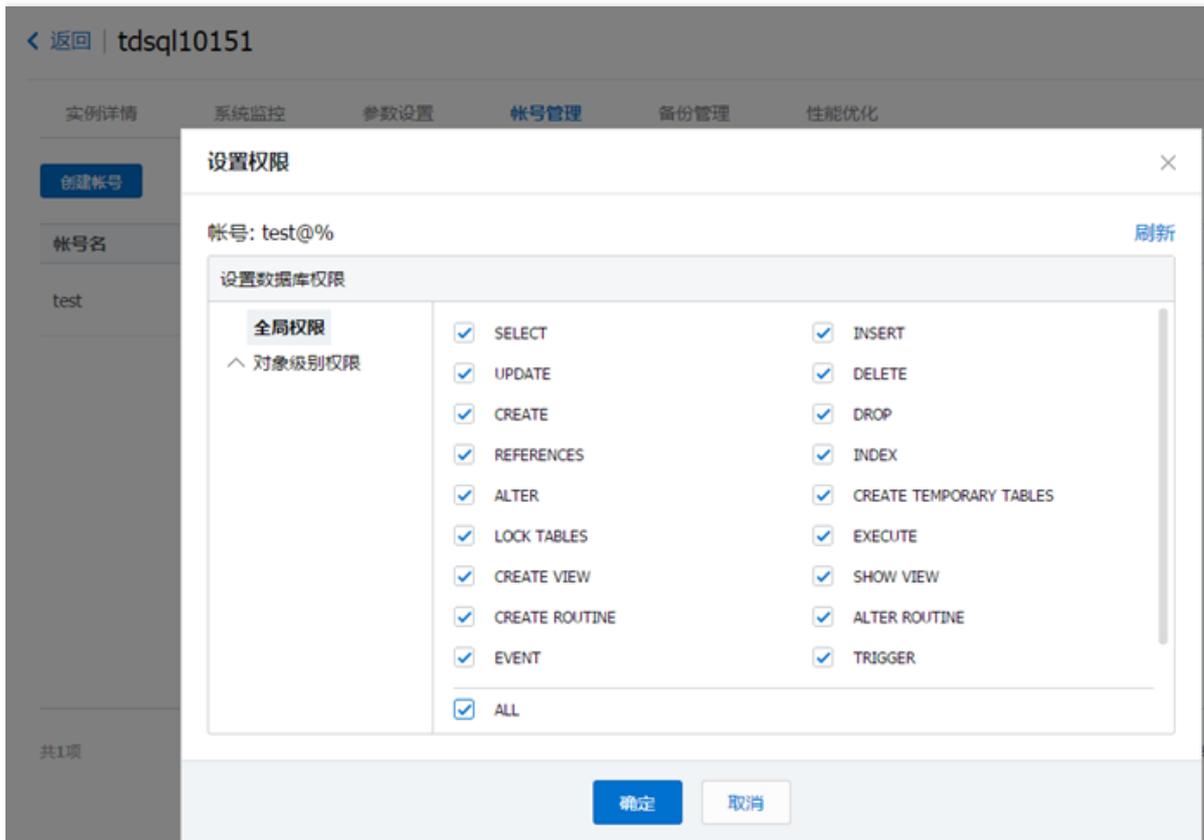
- **帐号名**：由字母、数字、下划线组成，字母开通，字母或数字结尾，最长16个字符组成；
- **主机**：也可以理解为HOST，支持IP、IP段、%三种形式；%代表结尾符，例如我们要支持10.10.10.1~10.10.10.254的所有主机IP，可以输入10.10.10.%；不输入代表
- **是否只读帐号**：选中表示该帐号只能使用读请求（select）；
- **密码**：密码为8~16位的任意字符；腾讯云建议您的密码至少包括英文、数字和符号等，并定期修改密码。



注意：同一个帐号名不同的主机IP需要独立设置权限，您可以选择账号管理中“克隆”功能，帮助您快速创建克隆相似的帐号和权限设置。

### 3.设置权限

MariaDB(TDSQL)的权限为对象级权限，权限包括数据库常见的19种权限；也可以为表、视图、函数、触发器等对象设置权限。



注意：未创建数据库时，无法设置对象级别权限。

#### 4.限制条件

因产品设计思路与信息安全要求限制，暂不支持通过命令行 `insert into mysql.user`、`grant`、`drop` 创建、修改帐号。

# 迁移数据

## 使用mysqldump导入数据

最近更新时间：2018-06-06 10:42:46

使用 mysqldump 工具的优点是简单易用、容易上手，缺点是停机时间较长，因此它适用于数据量不大，或者允许停机的时间较长的情况。

### 1. 使用 mysqldump 的数据导出工具，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。本步骤仅仅导出数据，不包括存储过程、触发器及函数。

```
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

### 2. 使用 mysqldump 导出存储过程、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，可跳过此步骤。在导出存储过程、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

### 3. 将数据文件和存储过程文件上传到云服务器 CVM 上。

请确保CVM和云数据库CDB能正常连通，且CVM存储空间足够。

### 4. 登录 CVM，将数据文件和存储过程文件导入到目标云数据库中。

请确保您拥有相应权限的数据库帐号，否则可能需要前往控制台生成帐号

```
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，本例以内网地址为例

userName：CDB 数据库的迁移账号

dbName：需要导入的数据库名

/tmp/dbName.sql：要导入的数据文件名

/tmp/triggerProcedure.sql：要导入的存储过程文件名

# 数据导入至MariaDB(TDSQL)实例

最近更新时间：2018-06-28 17:42:06

您可以通过腾讯云数据库迁移工具将业务数据迁移到MariaDB (TDSQL) 中，迁移工具可以把同地域下不同类型的源数据库的数据迁移到MariaDB (TDSQL) 中，目前支持的源库类型有：

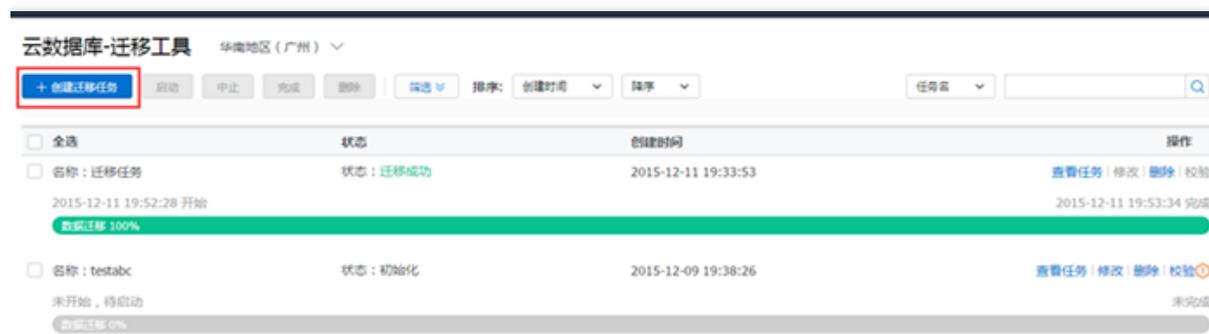
1. 源数据库来自在腾讯云基础网络中云服务自建的MySQL实例（即将支持VPC网络）；
2. 源数据库在腾讯云的MySQL实例；
3. 源数据库在另外一个MariaDB (TDSQL) 实例；

迁移步骤如下：

## 1.1. 进入迁移工具页面



## 1.2. 创建迁移任务



单击创建迁移任务，输入任务名称、源库和目标MariaDB (TDSQL) 的信息。

✔ 任务初始化

---

**任务设置**

任务名称 \*  ✔

定时执行  × 定时执行意味着该任务从指定时间开始排队

---

**源库信息**

源库类型 \*  ✔

CVM实例ID \*  ✔

IP \*  ✔

端口 \*  ✔

账号 \*  ✔

密码 \*  ✔

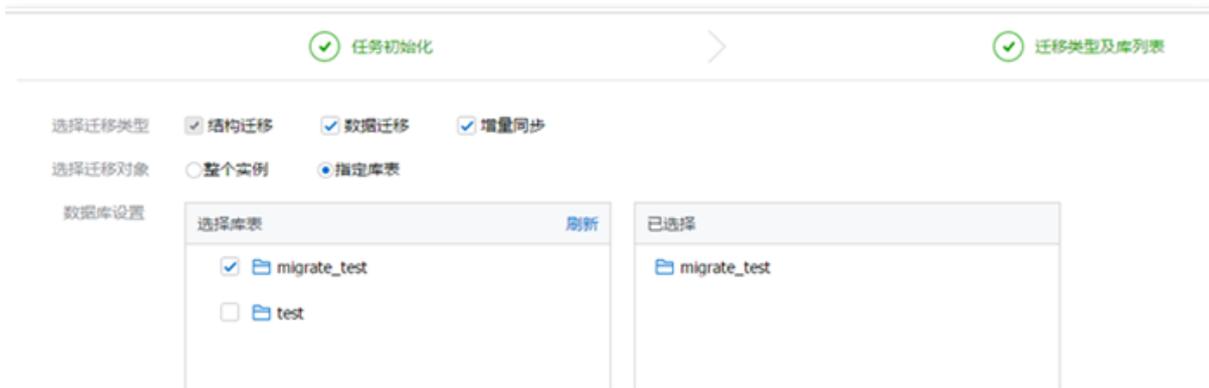
---

**目标库信息**

目标库类型 \*  ✔

实例ID \*  ✔

然后选择要迁移的数据库，创建并检查迁移任务信息。



**数据迁移**：将选中数据库中的数据导出，然后在MariaDB（TDSQL）中导入。

**增量同步**：在进行数据导出导入后，设置MariaDB（TDSQL）为源库的备库，进行主备增量同步。

### 1.3. 校验迁移任务信息

在创建完迁移任务后，您需要对迁移任务信息进行校验，只有所有校验项通过后才能启动迁移任务。任务校验存在3种状态：

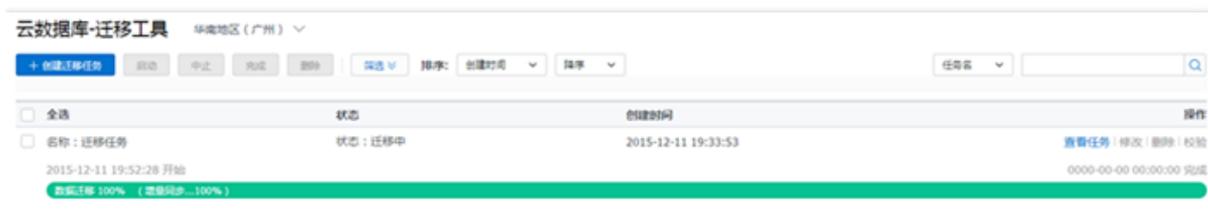
- 通过：表示校验完全通过
- 警告：表示校验不通过，迁移过程中或迁移后可能影响数据库正常运行但不影响迁移任务的执行。
- 失败：表示校验不通过，无法进行迁移。如果校验失败，请根据出错的校验项，检查并修改迁移任务信息，然后重试校验。失败原因请参考：“校验失败说明”



### 1.4. 启动迁移

在校验通过后，您可以启动迁移任务，如果您设定了迁移任务的定时时间，则迁移任务会在设定的时间开始排队并执行，如果没有设置定时任务，则迁移任务会立即执行。

迁移启动后，您可以在迁移任务下看到对应的迁移进度信息。



注：由于系统设计限制，一次性提交或排队多个迁移任务将按排队时间串行执行。

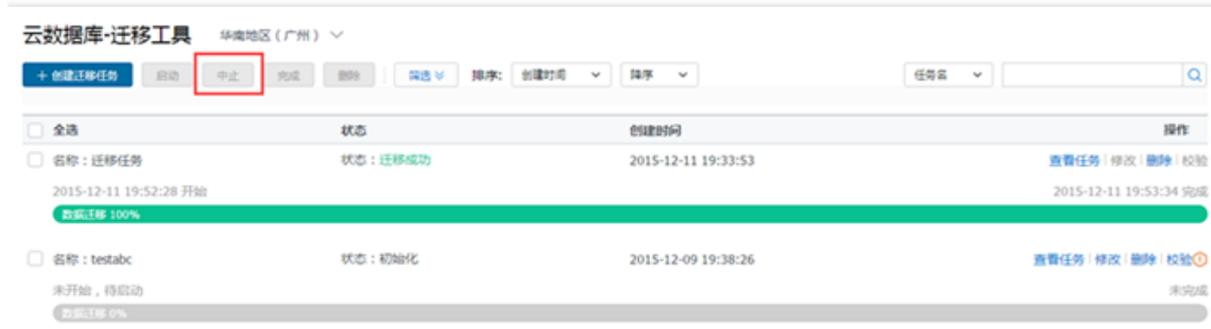
### 1.5. 增量同步

如果您在创建迁移任务时选择了增量同步选项，那么数据迁移完成后，我们会把目标MariaDB（TDSQL）库设置成源数据库的备库，通过主备同步来把迁移过程中源库的新增的数据同步到目标MariaDB（TDSQL）库中。此时，源库上的修改都会同步到目标MariaDB（TDSQL）中，您可以把业务切换到目标MariaDB（TDSQL）上，然后单击完成迁移。

单击完成迁移后，主备同步关系会断开。

## 1.6. 中止迁移

在迁移过程中，如果您需要停止迁移，可以单击中止按钮进行中止。



**注意：再次启动可能导致校验失败或任务失败，您可能需要手动清空目标库内的数据，才能再次启动迁移任务。**

## 2.通过第三方工具迁入

如果您的源库不在腾讯云上或者在VPC网络环境内，那么您可以通过第三方工具来先把源库的数据导出，然后上传到腾讯云服务器上，再把数据导入到MariaDB（TDSQL）。

常见的导入导出工具有：mysqldump、mydumper、MySQL GUI Tools等。