

云数据库 MariaDB

用户指南

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

用户指南

同城双活解决方案

功能限制

如何实现读写分离

自建只读实例

慢查询分析

如何创建数据库帐号

如何对数据库进行回档

修改数据复制方式

如何解压备份和日志文件

如何利用热点更新技术应对秒杀场景

如何利用备份文件恢复实例

数据库审计

数据库审计已支持语法

用户指南

同城双活解决方案

最近更新时间：2018-08-08 17:16:36

腾讯云 MariaDB 目前已经支持同城双中心双活能力，主要特性概况如下：

- 同城双中心部署
- 双中心可写：如果您的服务器部署在两中心不同子网中，可分别从两中心各自的服务器连接数据库写入数据。
- 故障自动转移/恢复
- 双中心唯一访问 IP

然而，仅是数据库同城双中心双活并不能实现业务系统级的容灾；事实上，单一系统/模块切换到同城灾备中心容易，但企业级系统业务内部复杂的关联、配置都是双中心面临的难题。

因此，**构建双活业务系统，需要业务在系统的设计、使用、管理、系统升级过程中时刻都以双中心为基础，双中心实时使用，配置互通为基础**，这样才能做到故障后，业务不修改或较少修改，即可快速恢复运行。这也是，腾讯云 MariaDB 同城双中心双活设计的目标，让两个中心的业务系统都能通过本地网络，完全正常读写数据库系统，且能够保证数据库强一致性。

设计标准

腾讯云 MariaDB 双活的设计标准参考《GB/T 20988-2007 信息安全技术 信息系统灾难恢复规范》，由于是数据库单一模块

- RTO≤60秒
- RPO≤5秒
- 故障切换时间≤5秒
- 故障检测时间≤30秒

这意味着，含故障检测时间从故障发生到完全切换完成约需 40 秒。

- 风险提示：真实环境测试下，需确保业务系统具有自动重连数据库机制；然而业务系统往往存在多个模块，每个模块可能与多个数据源相关，因此越是复杂系统恢复时间越长。

支持情况

已支持情况

实例版本：

- 标准版：一主一从（双节点）/一主二从（三节点）；
- 金融定制版：一主一从（双节点）/一主二从（三节点）；

网络要求：仅VPC网络；

已支持地域：

- 北京（北京一区、北京三区）
- 上海金融（金融一区、金融二区）

- 深圳金融（金融一区、金融二区）

价格说明

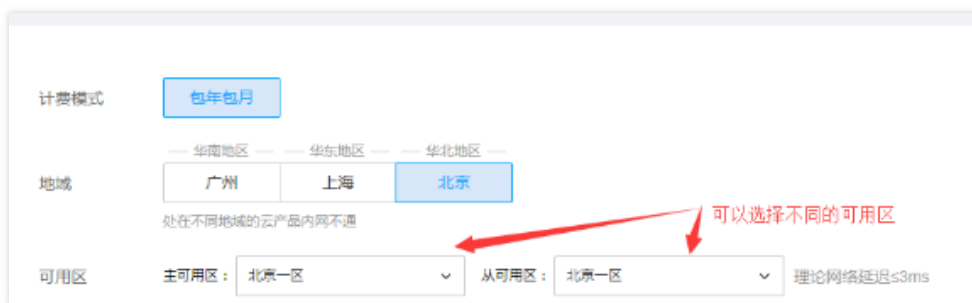
双可用区售价与单可用区相同；详见 [价格说明](#)

- 风险提示：在 2019.06.30 前免收双中心数据同步专线费用；如后续运营政策变更，价格将不高于当时腾讯云已公开的同城专线费用；如您不希望继续使用同城双中心方案，可免费迁移至同城单中心。

购买与使用

请至 [云数据库 MariaDB 购买页](#) 单击购买。

- 主、从可用区相同时，即单可用区部署；
- 主、从可用区不同时，即同城双中心部署；



注意：

- 主可用区即您主服务器所在区域，数据库应尽量分配在于主服务器相同的 VPC 子网内，减少访问延迟。从可用区，数据库从节点所在可用区；如果是一主二从 3 节点，主可用区将部署 2 个节点；如果是一主一从 2 节点，主可用区将部署 1 个节点。
- 金融云围拢方案如需支持同城双中心策略，可能需要先建设同城双中心围拢方案，详情请咨询您对应商务、架构师。

实例初始化

请参考 [实例初始化](#) 对实例进行初始化。

查看实例可用区详情

您可以在实例详情页查看。

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						
实例名: tdsql10266 更改						实例ID: tdsql-c8afcyok 复制
状态: 运行中						实例类型: 主实例
实例版本: 标准版(一主一从)						地域: 华东地区(上海)
内网地址: 更改						端口: 3306
所属网络: test_bernieli						外网地址: 打开
所属项目: 默认项目 转至其他项目						字符集: UTF8 更改
配置信息						
数据库版本: 基于MariaDB 10.1.9设计(兼容MySQL 5.6)						配置: 2GB内存, 10GB容量 升级
备份与日志空间: 5GB (免费赠送实例容量的50%)						已使用/总容量: 0MB/10GB
节点数量: 2						创建时间: 2018-04-02 17:53:21
到期时间: 2018-05-02 17:53:21 续费						
可用性信息						
数据复制方式: 异步 修改数据复制方式						部署方式: 多可用区
主库可用区: 上海二区 主从切换						从库可用区: 上海一区

主从切换

如果您要将主节点从某一可用区切换到另一可用区，您可以直接单击主从切换即可。主从切换是高危操作，需求验证登录帐号的 IP 地址；切换过程可能导致数据库连接闪断（≤1s），请保证业务有数据库重连机制；频繁切换将可能业务系统异常甚至数据异常。



内网地址: [192.168.10.25 更改](#) 端口: **3306**

所属网络: **test_bernieli**

所属项目: [默认项目 转至其他项目](#)

配置信息

数据库版本: **基于MariaDB 10.1.9设计(兼容MySQL 5.6)**


备份与日志空间: **5GB (免费赠送实例容量的50%)**

节点数量: **2**

到期时间: **2018-05-02 17:53:21 续费**

可用性信息

数据复制方式: **异步 修改数据复制方式**

主库可用区: **上海二区 主从切换** 

从库可用区: **上海一区**

主从切换 ×

手机号码:* [获取验证码](#)

接收短信手机号码 刷新 管理手机号

验证码:*

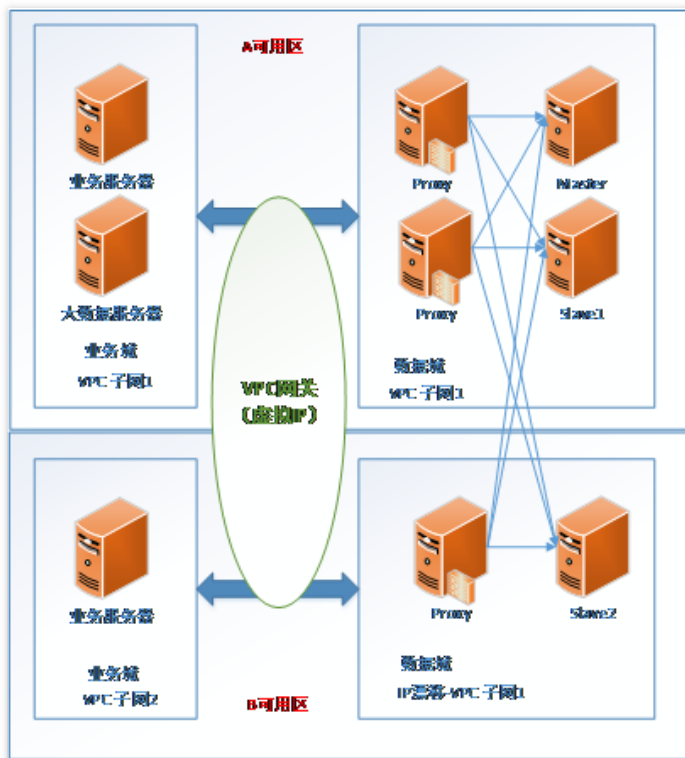
请输入6位数字验证码, 5分钟内有效

计划切换节点:*

技术原理简介

基于 MariaDB 高可用的主从架构和 VPC 可用区虚拟 IP 漂移特性的有机整合，实现了双中心同时读写；架构特点如下：

- MariaDB 每一个 DB 节点前端，混合部署 Proxy 模块；Proxy 模块负责将数据请求路由到对应的 DB 节点；
- 在 Proxy 模块前部署跨地域 VPC 网关，并支持虚拟 IP 漂移功能；



如上图，以写入数据为例，假设业务服务器部署在 A 可用区，PC 网关转发数据请求到 A 可用区的 Proxy 网关，再由 Proxy 做透明转发到 Master 节点。而假设业务服务器部署在 B 可用区，VPC 网关转发数据请求到 B 可用区的 Proxy 网关，再由 Proxy 做透明转发（通过腾讯云 BGP 专网）到 Master 节点；

无论是读请求还是写请求，整个过程对业务透明。如果是数据库异常，数据库集群则按如下原则处理：

1. 若 Master 与 Proxy 故障，集群自动选举最优 slave 提升为新 Master，系统通知 VPC 修改虚 IP 与物理 IP 关联关系，业务仅感知部分写连接中断。
2. 若 Master 故障，但 Proxy 正常，集群自动选举最优 slave 提升为新 Master，Proxy 将阻塞请求，直到主从切换完成；此时，业务仅感知到部分请求超时。
3. 若 Slave 故障（无论 Proxy 的是否故障），读写分离情况下，根据预先配置的只读账号**只读策略**（有 3 种）执行。
4. 若 A 可用区完全故障，VPC、数据库在 B 可用区仍然存活，此时，slave2 节点自动提升为 Master 后，**根据强同步策略调整该节点读写策略**，VPC 网络 IP 漂移到 B 可用区；此时集群将重试恢复 A 可用区节点，如果超过 30 分钟无法恢复，将自动在 B 节点重建至少 1 个 Slave 节点。由于有 IP 漂移策略，业务不需要修改数据库配置。
5. 假设 B 机房完全故障，MariaDB 集群相当于故障了 Slave 节点，处理方法与第 3 相同。

FAQ

1. 相对于单中心，同城双中心是否会导致性能下降？

在基于强同步复制方案下，由于跨中心延迟会略大于同机房内设备；因此 SQL 响应会有理论下降 5% 左右。

2. 是否存在 Master 节点从我主可用区切换到从可用区？

会的，如果不影响您的业务使用，可不用管它。如果您担心影响，可在业务低谷期，通过控制台主从切换功能，切换回去。

3.如何知道数据库集群做了主从切换？

请到控制台云监控告警策略>云数据库MariaDB>配置主从切换告警

4.如果一部分的读写请求读写到从可用区，因为网络延迟原因导致性能下降，但又不想放弃同城双中心这个特性怎么办？

您可以提交工单，说明实例ID、和您服务器在可用区部署方案，以及读写请求比例；可由腾讯云DBA帮您调整双可用区负载机制，让从可用区的承担的读写请求尽量少些。

5.如果期望从单中心更换为同城双中心架构，应该如何操作？

首先确认您所在区域是否支持同城双中心方案，目前已开发北京、上海金融、深圳金融三个地域；其次，您需要提交工单，说明需要调整的账号信息、实例ID、计划哪两个可用区、以及建议运维操作时间；最后，腾讯云工作人员会进行审核（已支持双可用区的）即可按需操作，否则将驳回该需求。

功能限制

最近更新时间：2018-05-25 15:59:21

功能限制

1.不能更改mysql、information_schema、performance_schema和 sysdb 库中的任何数据

2.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

3.不提供 Root 帐号

4.建议使用 innodb 存储引擎。

如何实现读写分离

最近更新时间：2018-04-10 09:28:41

1. 背景说明

云数据库 MariaDB (TDSQL)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 (TProxy) 自动分配到低负载从机上。

2. 全局自动读写分离

云数据库 MariaDB(TDSQL)支持全局自动读写分离能力，即业务无需做任何改动，即将所有 SELECT 请求发送到从机；该功能为灰度发布功能，需提交工单，说明“账号信息、实例所在地域、实例ID、需开启自动读写分离，以及是否知悉全局读写分离风险”

全局读写分离风险：从机天然存在数据延迟，全局自动读写分离即系统默认将所有 SELECT 请求发送到从机，并无任何策略（请参考只读账号的读写分离），因此可能导致业务读取到不可控的历史数据，造成数据失真，该功能仅建议用于对数据一致性无要求的业务系统。

注意您一旦提交工单开启该功能，即您已经确认自身业务系统能够承担该类风险。

3、基于只读帐号的读写分离

只读帐号是一类仅有读权限的账户，其默认从数据库集群中的从机（或只读实例）中读取数据。TDSQL 可以在【帐号管理】功能中，设置只读帐号和读策略，如下图：

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，设置只读账号的参数

主机:
IP形式，IP段以%结尾；支持填入%，127.0.0.1；为空默认等于%

设置密码:*
密码需要8-32个字符，不能包含[";]

确认密码:*

备注:
请输入备注说明，最多256个字符

在只读帐号设置选项中，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,如下图

只读帐号设置 ×

只读帐号非全局设置，调整不会影响其他只读帐号

帐号名: vitos

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”则备机延迟超时时从主机读取
选择“直接报错”则备机延迟报错
选择“只从备机读取”则忽略延迟参数，一直从备机读取（一般用于拉取binlog同步）

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.

配置建议：

假设您设计的是某交易系统

- 核心交易模块：设置常规帐号，可读写。
- 余额查询模块：设置只读帐号，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：设置只读帐号，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。

4、基于注释的读写分离

在每条需要从机“读”的 SQL 前，增加 `/slave/` 字段，并且 `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 ;
```

注意事项：

- 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。

4、支持只读实例（异地只读实例）

如果上述读写分离方案仍无法满足您的需求，云数据库 MariaDB（TDSQL）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。现在对部分客户开启该能力，如有需要，请联系您的专职商务、架构师、售后同事。

自建只读实例

最近更新时间：2018-05-25 17:01:17

2017年4月1日以后申请的MariaDB(TDSQL)实例默认支持该能力，2017年4月1日以前的实例请提交工单安排升级，升级过程可能会有一分钟左右闪断，请注意协调到业务低谷期。

1. 概述

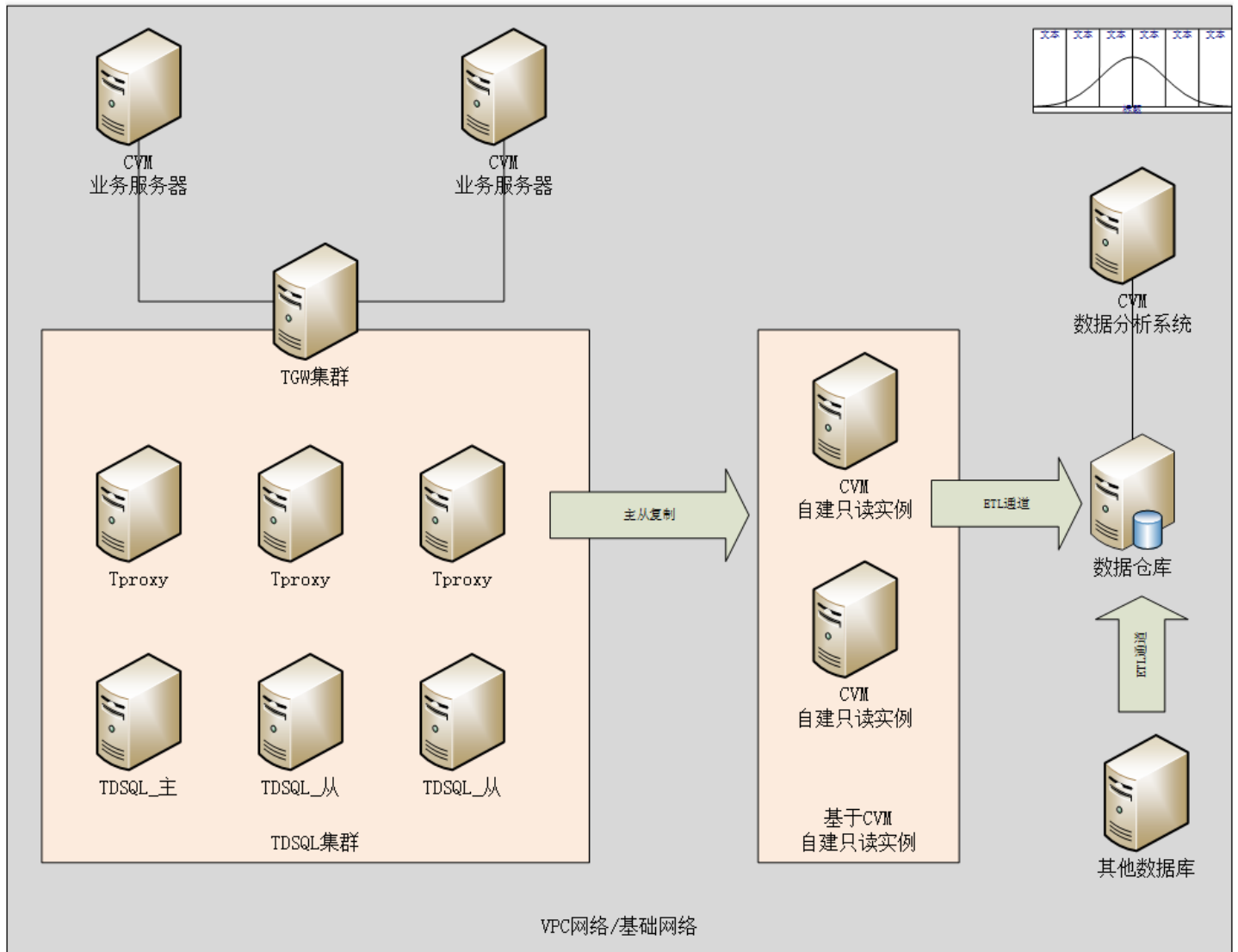
1.1 概述

如果您仅需要实现[读写分离](#)，目前MariaDB(TDSQL)所有从机都支持只读，详情可查看文档[读写分离技术](#)。

然而，您可能仍然需要额外只读实例，以实现更灵活的控制，并实现构建一些复杂业务系统。因此，MariaDB(TDSQL)支持**利用CVM自建只读实例**的方案。

1.2 自建只读实例的架构

如下图所示，自建只读实例是在MariaDB(TDSQL)集群外，利用主从同步功能和TPProxy的binlog同步功能，将数据“同步”到一个或多个自建只读实例。这样，您可以在自主控制的CVM服务器上，安装任何软件或做任何数据操作。



为保证性能和MariaDB(TDSQL)集群的稳定性，我们在此方案下，仅提供异步同步方案，且在集群压力过大时提高数据同步延迟以减少对MariaDB(TDSQL)主集群的性能消耗；这意味着，您自建的只读实例的数据延迟可能在几秒到几分钟不等。对延迟有苛刻要求的情况下，建议您使用MariaDB(TDSQL)的读写分离技术。

2. 建设方案

2.1 明确MariaDB(TDSQL)内核版本

通常您可以在实例列表或实例详情中，查看TDSQL的内核版本；目前已经支持MariaDB 10.0.10，MariaDB 10.1.9；（未来可支持更多版本，但基本方案不变）

2.2 创建适当规格的CVM

根据您现有实例内存、磁盘规格信息，创建CVM。

- **CVM配置**：我们建议您的CVM配置比MariaDB(TDSQL)实例的大约20%，且数据盘至少大50GB，内存至少大2GiB。如果您还要在CVM上安装其他软件，再根据软件需要适当加大CVM配置
- **操作系统**：推荐centos 6/centos 7

- **数据盘格式化**：要求ext3文件系统，且挂载在/data目录下。
- **安装MariaDB**：必须安装与MariaDB(TDSQL)相同的版本。

2.3 安装MariaDB举例

以centos为例，安装(mariadb 10.0.10)版本的步骤如下：

- 1.配置添加yum源文件(vi /etc/yum.repos.d/mariadb.repo):

10.0.10版本的官网yum源为：

```
...
# 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
...
```

10.1.9版本的官网yum源为：

```
...
# MariaDB 10.1 CentOS repository list - created 2016-05-30 02:16 UTC
# http://downloads.mariadb.org/mariadb/repositories/
[mariadb]
name = MariaDB
#baseurl = http://yum.mariadb.org/10.1/centos7-amd64
baseurl = http://archive.mariadb.org/mariadb-10.1.9/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版本是否正确

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

- 3.安装MariaDB-server和Client

```
yum install MariaDB-server MariaDB-client MariaDB-devel -y
```

注：如果有提示和以前安装的版本冲突，您可能需要先移除以前安装的包，例如: yum remove mariadb-libs.

2.4 安装mydumper

下载地址：<https://launchpad.net/mydumper>

编译安装：`yum install make cmake pcre-devel glib2-devel zlib-devel gcc-c++`

解压 `mydumper-0.9.1.tar.gz` 进入目录并安装：

`cmake .`

`make && make install`

安装完后运行 `mydumper -V` 可以看到版本信息

2.5 配置CVM上的mariadb并启动

删除文件 `/etc/my.cnf.d/enable_encryption.preset`

`rm -f /etc/my.cnf.d/enable_encryption.preset`

创建目录：

`mkdir -p /data/dbdata/{data,tmpdir}`

`mkdir -p /data/dblogs/relay`

修改配置文件 `/etc/my.cnf.d/server.cnf`：

```
[mysqld]
character-set-server = utf8
collation-server = utf8_general_ci
innodb_page_size=16384
lower_case_table_names=1
server-id=10
innodb_buffer_pool_size=2097152000
skip-name-resolve
datadir = /data/dbdata/data/
relay-log = /data/dblogs/relay/relay.log
log-error = /data/dblogs/mysql.err
tmpdir = /data/dbdata/tmpdir
```

注意事项：

在配置这些参数时，可以在原实例上使用 `show global variables like xxx` 来查看原实例的参数做参考。其中：

`character-set-server`、`lower_case_table_names`、`innodb_page_size` 等参数要和原TDSQL实例一样

`innodb_buffer_pool_size` 的值可以参考tdsql的参数。

`server-id`要和原TDSQL实例不同

运行 `mysql_install_db` 安装db

修改文件所有者：

`chown -R mysql:mysql /data/dbdata/ /data/dblogs/`

启动mariadb：

`service mysql start`

修改root密码并测试登录：

`mysqladmin -u root password 'root';`

`mysql -uroot -proot -hlocalhost -P3306`

2.6 创建用于同步的数据库账号

如果准备从主机上同步，创建普通账号，如果从备机上同步，则创建只读账号（建议从备机上进行同步，避免影响主机）。

设置数据库账号的权限至少包含全局 SELECT, REPLICATION SLAVE, REPLICATION CLIENT, RELOAD 的权限（如果在控制台进行设置权限失败，或者没有所需的权限，请建工单联系客服为您的同步账号添加所需的权限）。

2.7 使用mydumper导出TDSQL数据

注意：第一次导出过程中会有锁表操作，请评估业务是否可以接受，建议在业务低峰期操作。

使用mydumper导出tdsql数据(-B后面跟着的是要进行同步的数据库)：

```
mydumper -t 4 --host 10.66.183.239 --port 3306 --user=user --password=password001 --outputdir=./export_tdsq| -B syncdb
```

使用myloader导入数据(-B后面跟着的是要进行同步的数据库)：

```
myloader --host localhost --port 3306 --user=root --password=root -d ./export_tdsq| -o -B syncdb
```

2.8 建立主从同步关系

查看7中导出的metadata文件信息：

```
cat export_tdsq|/metadata
```

里面应该要有binlog位置和gtid的信息，否则无法进行建主从同步关系：

```
...  
SHOW MASTER STATUS:  
Log: binlog.000003  
Pos: 64233211  
GTID:0-2694363359-883267
```

登录CVM的db配置主从：

配置要进行同步的db：

```
set global replicate_do_db='syncdb';
```

设置gtid_slave_pos点：

```
SET GLOBAL gtid_slave_pos = "0-2694363359-883267";
```

配置主从关系：

```
CHANGE MASTER TO
```

```
MASTER_HOST='10.66.183.239',MASTER_PORT=3306,MASTER_USER='user',MASTER_PASSWORD='password111',master_use_gtid=slave_pos;
```

启动备机：

```
START SLAVE;
```

查看slave状态：

```
show slave status\G
```

```
...
```

其中Slave_IO_Running和Slave_SQL_Running为running状态，则配置成功。

5. 使用注意事项及功能限制

-
- 此处基于CVM部署的只读实例，并不会被纳入MariaDB(TDSQL)集群进行管理，这意味着如果MariaDB(TDSQL)集群中节点故障，该方案部署的只读实例（永远）不会被切换为主机，所以他不会提升系统可用性。
 - 您可以挂在多个自建只读实例，但从业务性能建议您不超过4个。
 - 自建只读实例功能当前免费。

慢查询分析

最近更新时间：2018-06-28 17:43:11

1. 功能说明

将超过指定时间的SQL语句查询称为“慢查询”，对应语句称为“慢查询语句”，而数据库管理员（DBA）对慢查询语句进行分析并找到慢查询出现原因的过程叫做慢查询分析。

腾讯云云数据库（TencentDB for MariaDB(TDSQL)）在【实例管理】-【性能优化】模块下，提供慢查询分析能力。

2. 主要参数说明

2.1. 主要默认设置

- 慢查询功能：默认开启
- 慢查询时间（long_query_time）：默认配置为1s；即慢查询语句查询时间超过1s的才被记录。
- 分析数据输出延迟：1~5分钟。
- 日志记录时长：30天，根据备份和日志设置周期决定。

2.2. 分析列表字段说明

- 校验值（checksum）：标示慢查询语句的一串序列数字，默认64bit；
- 抽象后的慢查询语句（fingerprint）：隐去用户数据以后的慢查询语句；
- 数据库：出现慢查询语句的数据库；
- 账号：出现慢查询语句的账号；
- 最后执行时间（last_seen）：时间范围内，最后一次出现慢查询语句的时间；
- 首次执行时间（first_seen）：时间范围内，第一次出现慢查询语句的时间；
- 总次数（ts_cnt）：时间范围内，慢查询语句出现的次数；
- 总次数占比：时间范围内，慢查询语句占有所有慢查询语句次数的占比；
- 总时间（query_time_sum）：时间范围内，慢查询语句查询总耗时；
- 总时间占比：时间范围内，慢查询语句查询总耗时的占比；
- 平均时间（query_time_avg）：慢查询语句总时间除以总次数的平均时间；
- 最小时间（query_time_min）：慢查询语句出现的最小时间；
- 最大时间（query_time_max）：慢查询语句出现的最大时间；
- 总锁时间（lock_time_sum）：慢查询语句出现锁的总耗时；
- 总锁时间占比：时间范围内，慢查询语句占有所有慢查询语句锁时间的占比；
- 平均锁时间（lock_time_avg）：慢查询语句总锁时间除以总锁次数的平均时间；
- 最小锁时间（lock_time_min）：慢查询语句锁出现的最小时间；
- 最大锁时间（lock_time_max）：慢查询语句锁出现的最大时间；
- 发送行数（Rows_sent_sum）：该条慢查询语句发送的数据行数总和；
- 扫描行数（Rows_examined_sum）：该条慢查询语句扫描的数据行数总和。

如何创建数据库帐号

最近更新时间：2018-06-28 17:43:39

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. 进入帐号管理页面，选择帐号管理功能

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



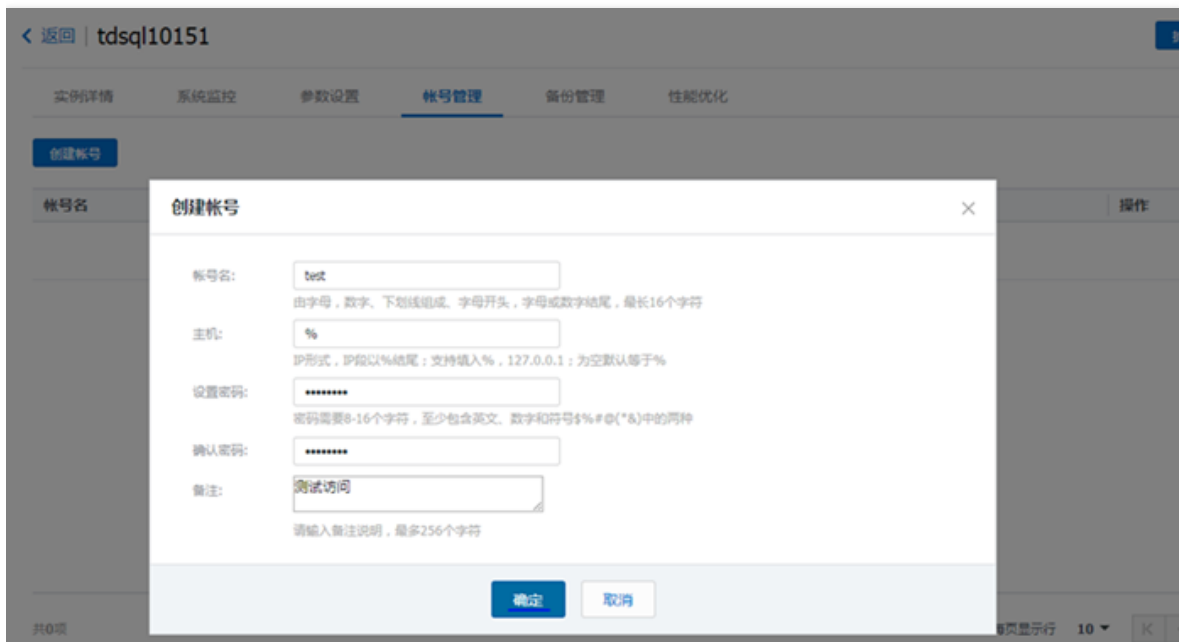
您可以在管理页面看到“帐号管理”，如下图所示：



2. 创建账号

这里的帐号由帐号名、主机、密码三部分组成；

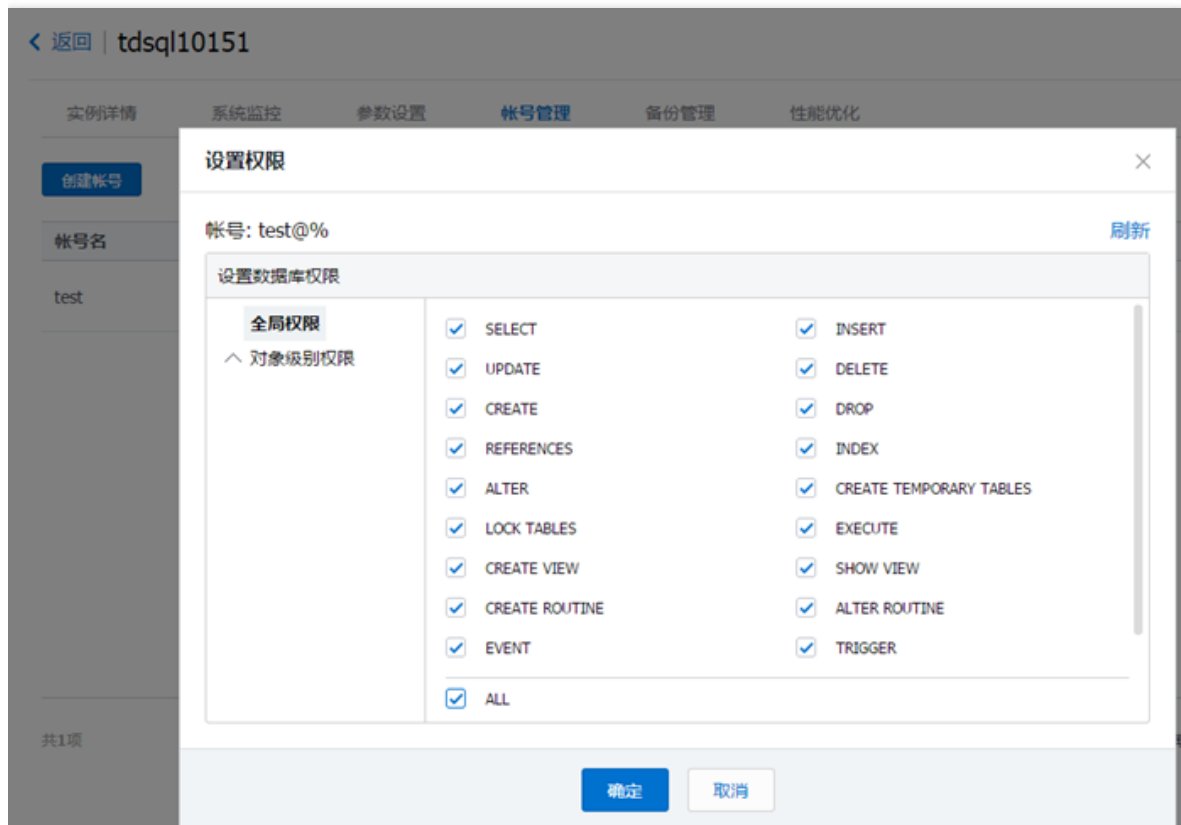
- **帐号名**：由字母、数字、下划线组成，字母开通，字母或数字结尾，最长16个字符组成；
- **主机**：也可以理解为HOST，支持IP、IP段、%三种形式；%代表结尾符，例如我们要支持10.10.10.1~10.10.10.254的所有主机IP，可以输入10.10.10.%；不输入代表
- **是否只读帐号**：选中表示该帐号只能使用读请求（select）；
- **密码**：密码为8~16位的任意字符；腾讯云建议您的密码至少包括英文、数字和符号等，并定期修改密码。



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)L的权限为对象级权限，权限包括常见的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。

如何对数据库进行回档

最近更新时间：2017-06-05 15:57:33

1. 数据库回档说明

通过数据库回档能力，可以最大程度地减少系统损失；CDB for MariaDB(TDSQL)可以根据备份和日志保持情况，回档到30天内的任意时刻。

CDB for MariaDB(TDSQL)的回档功能不会影响现网生产实例，CDB for MariaDB(TDSQL)的回档可以直接回档到腾讯云创建的一个临时实例。临时实例可以支持读写，这样可以更灵活的对回档实例进行修改（如清理脏数据）；临时实例数据修正完毕后，可以通过“实例切换”将临时实例切换到现网生产实例。

回档过程中有以下限制条件：

- 在进行回档并创建临时实例的过程中，实例切换过程中，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；
- 回档操作有可能会对二进制日志（BINLOG）进行强行分片，即未达到100M 也会被备份为一个独立文件；
- 回档后的临时实例具备现网生产实例的参数信息（比如账户、数据库参数等），但不能修改。

具体回档方法请见下：

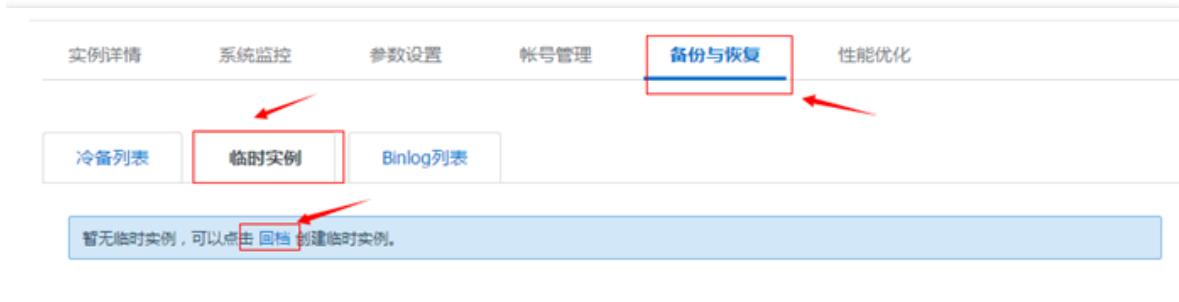


2. 临时实例

临时实例是CDB for MariaDB(TDSQL)基于备份生产的一个仅用于临时调整的实例，有效期仅有48小时。临时实例具备以下特性：

- 具备读写特性；
- 生命周期仅有48小时，48小时后删除；
- 一个实例同时只具备一个临时实例；
- 临时实例可以随意删除。

临时实例查看的位置：



3. 实例切换

实例切换是将临时实例切换为现网生产环境。当用户检查临时实例数据回档OK，可以对临时实例进行切换，此时会更换源实例和临时实例的内网链接。

- 切换过程中，现网生产实例可能闪断1秒；
- 切换完成后，原生产实例会被删除；
- 切换完成后，临时实例被替换为现网生产实例，而原临时实例也会被删除。

切换操作见下图：



修改数据复制方式

最近更新时间：2018-04-10 16:50:41

数据复制方式

数据复制方式（又名数据同步方式），是指在数据库高可用方案下，主从数据节点数据复制的机制。目前云数据库 MariaDB 已支持：

- **异步复制**：应用发起更新（含增加、删除、修改操作）请求，Master 完成相应操作后立即响应应用，Master 向 Slave 异步复制数据。因此异步复制方式下，Slave 不可用不影响主库上的操作，而 Master 不可用有概率会引起数据不一致。
- **强同步（不可退化）复制**：应用发起更新请求，Master 完成操作后向 Slave 复制数据，Slave 接收到数据后向 Master 返回成功信息，Master 接到 Slave 的反馈后再应答给应用。Master 向 Slave 复制数据是同步进行的，因此 Slave 不可用会影响 Master 上的操作，而 Master 不可用不会引起数据不一致。

注意：

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。——因为单一服务器服务，如果股指则直接导致部分数据完全丢失，不符合金融级数据安全要求。

- **强同步（可退化）复制**：业务系统中批处理、事务大量写入数据会导致从机严重延迟；加上强同步（不可退化）仅余单节点会被锁住；这些原本保证数据一致性机制可能导致业务系统异常。为解决这个问题，云数据库提供在强同步机制上可退化为异步的方案，**当从机延迟大于等于十五秒时，系统将自动把强同步退化异步；但从机延迟小于十五秒时，系统将自动减异步升级为强同步。**强同步（可退化）是一种高效的保证数据最终一致性的方案。

此处与 Google 开源的半同步机制不同点在于，强同步采用的是线程池，且不占用工作线程模式，且退化方案由于半同步。

修改数据复制方式

您可以在【实例初始化】阶段设定数据复制方式，也可以在【实例详情】>【数据复制方式】处修改数据复制方式。

修改过程不影响实例正常运行，修改后小于等于五秒即生效。

云数据库 MariaDB（TDSQL）对一主一从仅提供“强同步（可退化）、异步复制”两种方案；如需更改的数据一致性要求，请购买一主二从3节点版本。

如何解压备份和日志文件

最近更新时间：2018-05-25 16:56:23

出于压缩性能和压缩比的综合考虑，MariaDB(TDSQL)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。由于常见的解压工具不支持该格式，本文特别给出解压工具和操作指引。

Windows

下载工具

单击[这里](#)下载解压工具。

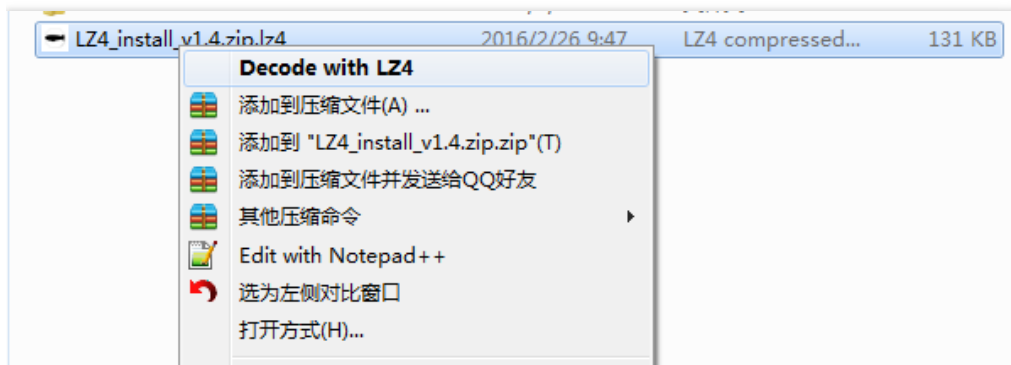
安装工具

双击zip文档，解压后得到LZ4installv1.4.exe，双击运行，然后按指引完成安装。

如果只是用来解压我们的文件，最后一步的复选框可以忽略。

解压文件

如下图，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

腾讯云的云主机的yum库中有LZ4组件，直接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
```

如何利用热点更新技术应对秒杀场景

最近更新时间：2018-06-28 17:51:31

1 引言

1.1 背景

在“秒杀”和“限时抢购”等这样的场景下，大量的用户在极短的时间内请求大量商品。而体现在 MySQL 数据库中，同一商品在数据库里肯定是一行存储，所以会有大量的线程来竞争 InnoDB 行锁，当并发度越高时等待的线程也会越多，TPS 会下降 RT 会上升，数据库的吞吐量会严重受到影响。本文档描述 Percona 解决“秒杀”和“限时抢购”所做的特殊优化——热点更新技术的。

1.2 使用简介

热点更新：采用如下示例的语句对某个数据对象频繁进行更新。

目前仅支持 Percona 5.7.17 版本（白名单限制开放）可在 [CDB for MariaDB 购买页中购买](#)

```
UPDATE COMMIT_ON_SUCCESS ROLLBACK_ON_FAIL QUEUE_ON_PK 88 TARGET_AFFECT_ROW 1 table_name SET k=k+1 WH
```

2 UPDATE 和 INSERT 语法变化

UPDATE 和 INSERT 的 SQL 语句可以增加新关键字，以表达热点更新的功能。红色为新增内容。

2.1 UPDATE 语法

```
UPDATE [LOW_PRIORITY]

[COMMIT_ON_SUCCESS] [ROLLBACK_ON_FAIL] [QUEUE_ON_PK expr1] [TARGET_AFFECT_ROW expr2]

[IGNORE] table_reference

SET col_name1={expr1|DEFAULT} [, col_name2={expr2|DEFAULT}] ...

[WHERE where_condition]

[ORDER BY ...]

[LIMIT row_count]
```

2.2 INSERT 语法

```
INSERT [LOW_PRIORITY | DELAYED | HIGH_PRIORITY]

      [COMMIT_ON_SUCCESS] [ROLLBACK_ON_FAIL] [QUEUE_ON_PK expr]

      [IGNORE]

[INTO] tbl_name

[PARTITION (partition_name,...)]

[(col_name,...)]

{VALUES | VALUE} ({expr | DEFAULT},...),(...),...

[ ON DUPLICATE KEY UPDATE

col_name=expr

[, col_name=expr] ... ]
```

2.3 说明

1. UPDATE 只支持单对象更新，即支持"single-table-syntax"，不支持"multiple-table-syntax"。
2. 只支持单机场景，XA 场景之后的迭代版本由 proxy 实现。
3. INSERT 的三种语法都支持，这里只列举一种。
4. 标准语法参考官方标准：

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

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

1. 对 QUEUE_ON_PK 指定的 expr 的值的对象，实施热点更新功能，通常 expr 的值是一个正整数。
2. 参数含义：
 - COMMIT_ON_SUCCESS：更新操作成功后，立即提交。适合单语句作为一个事务。
 - ROLLBACK_ON_FAIL：更新操作失败后，立即回滚。适合单语句作为一个事务。
 - QUEUE_ON_PK expr：指定热点更新对象，对被更新的对象封锁和解锁。被更新的对象总数不超过 hot_commodity_query_size，即，具有不同值的 expr 的个数不超过 hot_commodity_query_size。expr 取值自由，但建议与主键保持一致，也可以不一致。
 - TARGET_AFFECT_ROW expr：指定热点更新影响的数据行。expr 是一个正整数（[1, MAX], MAX 是 8 位正数的最大值）。通常 expr 为 1，表示只有一行受到影响。

2.4 建议

使用时，只在单语句事务中增加全部新增的参数使用，并且建议蓝色字体值匹配（可以不匹配）。

```
UPDATE COMMIT_ON_SUCCESS ROLLBACK_ON_FAIL QUEUE_ON_PK 88 TARGET_AFFECT_ROW 1 table_name SET k=k+1 WH
```

2.5 示例

```
CREATE DATABASE hc_db;
```

```
CREATE TABLE hc_tbl(a INT PRIMARY KEY, b INT, c INT);
```

```
CREATE TABLE hc_tbl_2(a INT PRIMARY KEY, b INT, c INT);
```

2.5.1 INSERT 示例

```
INSERT COMMIT_ON_SUCCESS ROLLBACK_ON_FAIL QUEUE_ON_PK 1 INTO hc_tbl VALUES(1, 1, 1);
```

```
INSERT COMMIT_ON_SUCCESS ROLLBACK_ON_FAIL QUEUE_ON_PK 1 INTO hc_tbl SET a= 2;
```

```
INSERT COMMIT_ON_SUCCESS ROLLBACK_ON_FAIL QUEUE_ON_PK 1 INTO hc_tbl_2 SELECT * FROM hc_tbl;
```

2.5.2 UPDATE 示例

```
UPDATE COMMIT_ON_SUCCESS ROLLBACK_ON_FAIL QUEUE_ON_PK 1 TARGET_AFFECT_ROW 1 hc_tbl SET b= b+1 WHERE a
```

QUEUE_ON_PK expr 中 expr 不一定和 WHERE clause 中的值一致

```
UPDATE COMMIT_ON_SUCCESS ROLLBACK_ON_FAIL QUEUE_ON_PK 2 TARGET_AFFECT_ROW 1 hc_tbl SET b= b+1 WHERE a
```

3 新增参数说明

参数名	功能	类型	默认值	其他
hot_commodity_enable	控制热点更新功能的开闭	布尔型	true 打开热点更新功能	运行中关闭此参数，新的事务不再使用热点更新。最好是系统启动前就设置好，而不是运行时改变
hot_commodity_trace	控制跟踪功能的开闭	布尔型	false 关闭跟踪功能	打开时，跟踪信息会输出到标准输出
hot_commodity_query_size	控制允许对多少个热点更新对象进行更新/插入操作	数值型	10000	起到限流的作用
hot_commodity_query_size_modify_enable	控制能否修改 hot_commodity_query_size	布尔型	false 不允许修改 hot_commodity_query_size	方便在单元测试中修改 hot_commodity_query_size

注意：如果 MySQL server 启动的时候，参数 `hot_commodity_enable` 是关闭的，则需要设置其为打开，重新启动 server，才能初始化全局的数据对象表。但如果 `hot_commodity_query_size` 值为 0，即使打开了 `hot_commodity_enable`，也不能使用热点更新。所以热点更新功能需要同时设置：

1. `hot_commodity_enable =ON`
2. `hot_commodity_query_size =10000` 为一个大于 0 的数值，建议控制在 10000、20000 左右，需要根据硬件环境和应用压力等实际情况测试确定其适合的值。
3. 启动 server

如何利用备份文件恢复实例

最近更新时间：2017-06-05 15:58:52

MariaDB(TDSQL)可以通过回档功能来查看历史数据，如果您需要在自己的机器上进行恢复，可以按照本文档的步骤进行操作恢复历史数据。

1、准备恢复所需环境

注意：请确保使用处于同一个基础网络或私有网络的云主机，以保障网络正常连通

基本配置建议：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；操作系统：centos最新版

基本环境需要：例如 **Mariadb 10.0.10**

安装过程举例：

- 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
```

- 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
```

注：如果有提示和以前安装的版本冲突，您可能需要先移除以前安装的包，例如: `yum remove mariadb-libs`

- 安装mariadb客户端

```
yum install MariaDB-client
```

- 安装lz4解压软件

```
yum install -y lz4
percona-xtrabackup
```

```
yum install http://www.percona.com/downloads/percona-release/redhat/0.1-3/percona-release-0.1-3.noarch.rpm
yum install percona-xtrabackup
```

2、下载备份

首先在“腾讯云管理中心>云数据库>管理中心>备份与恢复”获取备份下载地址

The screenshot shows the 'Backup and Recovery' (备份与恢复) section of the Tencent Cloud console. A table lists backup files with columns for 'Backup File' (备份文件), 'Backup Time' (备份时间), 'Backup Size' (备份大小), and 'Action' (操作). A modal dialog box titled 'Download Address' (下载地址) is open, displaying a VPC network address: `http://169.254.0.27:8083/2/noshard1/set_1464144850_58:` with a 'Copy' (复制) button. A warning message states: 'To ensure data security, we currently only provide internal network addresses. The address is valid for 15 minutes. After expiration, please refresh the page to obtain the address again. VPC network addresses must be accessed via the VPC network.' (为保证数据安全，目前暂只提供内网地址，地址有效期为15分钟，过期后请刷新页面重新获取，VPC网络地址请在VPC网络进行访问。)

下载命令示例

```
wget --content-disposition 'http://169.254.0.27:8083/2/noshard1/set_1464144850_58/1464552298xxxxxxx'
```

3、恢复备份文件

- 1) 进入备份文件下载目录，解压通过lz4解压冷备文件

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

- 2) 使用xstream工具解压到临时目录xtrabackuptmp中

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

解压完后，目录和文件情况大概如下：

```
[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 16:30 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) 应用日志

使用innobackupex应用日志

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

成功操作后，会显示completed OK!信息，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

停止数据库

```
service mysql stop
```

清空数据文件

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

- 5) 修改数据库参数文件 (/etc/my.cnf.d/server.cnf) , 具体参数数值参考步骤解压开的backup-my.cnf的参数:
注: 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!, 信息展示如下:

```
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 completed OK!
```

- 7) 启动数据库

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

注: 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。

数据库审计

最近更新时间：2018-06-28 18:08:19

1 概述

1.1 背景说明

企业使用数据库，可能面临如下安全风险，该类风险需要完整的事后审计和追溯机制，数据库审计能力就由此诞生。

管理风险

- 系统管理员存在的误操作、违规操作、越权操作，损害业务系统安全运行；
- 多人公用一个帐号，责任难以分清；
- 第三方开发维护人员的误操作，恶意操作和篡改；
- Root 帐号权限过大，无法审计监控；

技术风险

- 应用系统开发商后门或漏洞；
- 离职员工留下后门。

政策风险

- 无法达到国家等级保护（三级）明确要求（7.1.3.3）；
- 满足不了行业信息安全合规性文件要求——如人行《金融行业信息系统信息安全等级保护实施指引》；

1.2 术语定义

审计策略：定义对哪些用户行为进行审计以及如何响应的策略。**【审核策略】=【审核对象】+【审计规则】+【响应动作】** 即配置一条审计策略，需要指定审计内容，如果经过解析，某些（用户或系统）行为的特征正好符合某个审计规则，且恰好在策略生效时间，审计引擎就会按照此策略定义的响应方式进行响应，例如告警等。

审计规则：审计策略中，规定了需要审计的一系列行为的集合，称为规则。规则由规则参数组成，每个规则参数定义了一种具体的行为匹配特征。

1.3 产品能力于限制条件

腾讯云提供数据库审计能力，审计日志默认保存15天（后续版本可延长保存时间），帮助企业对可能存在的数据库访问进行风险控制，提高数据安全等级。

2 审计操作

2.1 开通数据库审计

目前，凡使用云数据库MariaDB（TDSQL）的用户，均可免费开通数据库审计；开通入口在“腾讯云管理中心>云数据库>MariaDB（TDSQL）>数据库审计”页面。

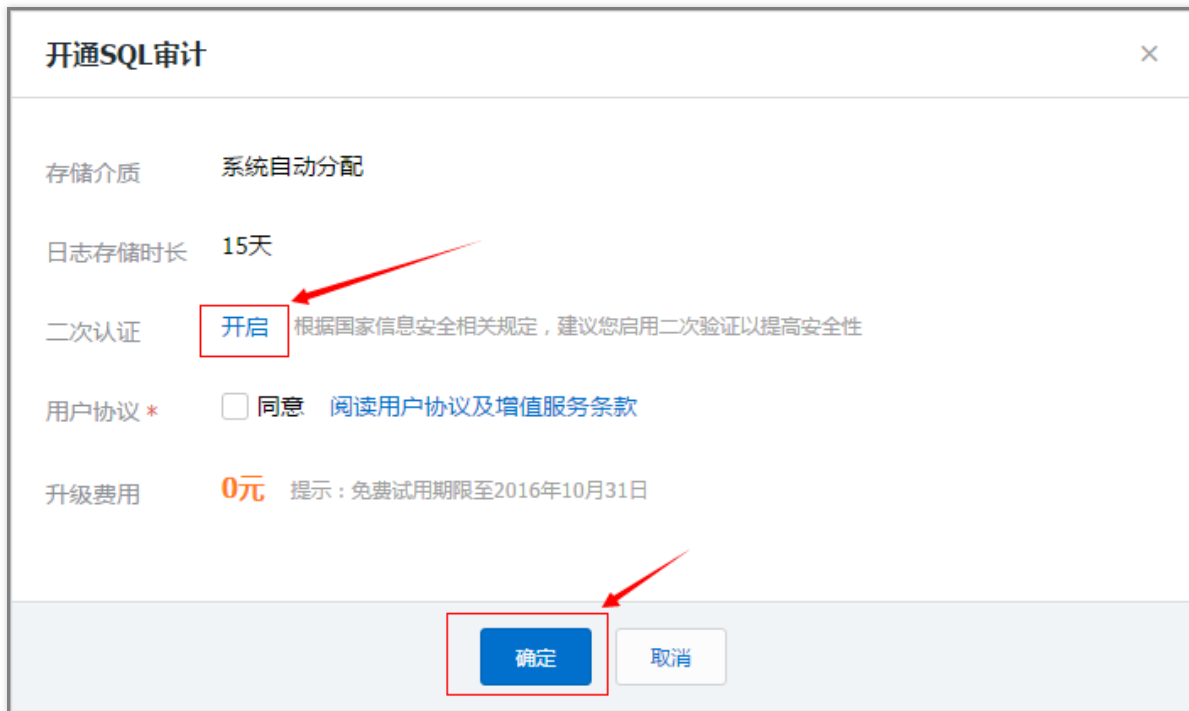
开通审计存在以下注意事项，请知悉：

- 您必须至少拥有1个MariaDB（TDSQL）实例，且未下线或隔离；否则系统会自动关闭您的审计功能。

- .2016年6月5日之前购买的MariaDB (TDSQL) 实例，需重启升级后方可支持该能力，由于重启升级可能会导致1~5秒的业务中断，您可以联系腾讯云工作人员预约升级时间。
- 数据库审计日志将以明文形式展示，因此建议您开启[二次登录认证](#)，当然，这不是必选项。



开通审计界面



开通审计可能存在几分钟初始化时间，请耐心等待。

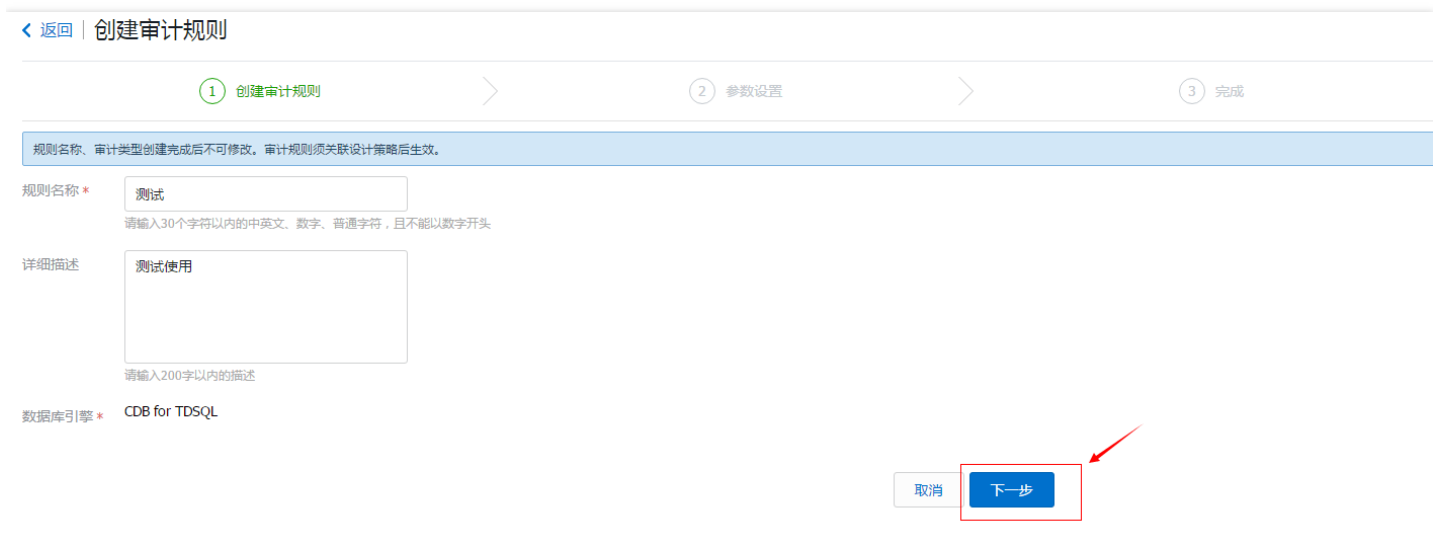
2.2 新建审计规则

审计功能开通后，日志会自动通过MariaDB（TDSQL）网关集群转发到审计集群，但由于没有建立审计规则和审计策略，日志不会持久化记录并展示。因此您可以通过【新建审计规则】>【关联审计策略】让日志存储在审计集群中。

1、进入审计界面，选择"审计规则"按钮，新建规则



2、填写审计规则名称，选择下一步



3、进入参数设置页面，填写规则参数（所列规则参数需至少填写一个，但不必全部填写）。

填写规则参数需注意以下能力：

- **规则中参数的条件关系**：与关系；规则中各参数之间是与的关系，即各参数都满足条件规则才会匹配成功。
- **特征串**：定义参数的具体内容，也就是操作对象的具体特征。为了达到精确匹配，用户只定义自己关心的参数的关键字，这样审计系统就只需记录用户定义的规则，提高审计检索效率。注意：特征串为空表示不关心该参数，即“匹配所有”。
- **匹配类型**：参数对象和特征的关系。
 - **包含**：表示网络字段中出现了特征串就匹配成功；
 - **不包含**：表示网络字段中没有出现特征串则匹配成功；
 - **等于**：表示网络字段等于特征串则匹配成功；
 - **不等于**：表示网络字段不等于特征串则匹配成功；

- 正则表达式：则表示特征串，支持标准的正则表达式语法。

4、所有新建规则，可以在规则列表中看到

规则名称	规则类型	描述	创建时间/最后修改时间	操作
测试	SQL审计	测试使用	2016-05-25 15:51:15 / 2016-05-25 15:51:15...	详情 修改 克隆规则 删除

5、设置完成审计规则后，您可以随时“修改”。类似的规则，您可以通过“克隆规则”的方式新建，以提高您的效率。

2.3 新建审计策略

审计策略是将“审计规则、审计对象、响应方式”组合起来形成完整的审计方案。用户可以对一个实例同时制定多条审计策略，审计引擎解析时，将按用户的策略配置顺序从前到后的优先级匹配。创建审计策略的方式如下：

- 1、进入策略生成窗：选择“审计策略”单击“新建策略按钮”

- 2、填写策略要求：根据需求选择需审计实例，并选择对应规则（目前暂不支持配置告警）。

新建策略
✕

新建或修改审计策略，预计5分钟后生效，原有审计策略的审计日志不会被清除或修改。

审计实例 tdsql10246 ▼

策略名称 测试策略
仅支持长度小于30的中文/英文/数字/"-"/"/"_

选择规则 测试 ▼
请选择规则

响应动作 放行且记录

警告策略 无

状态

确定
取消

3、调整优先级：对于相同实例下的多条策略，可以进行优先级调整；优先级数字越小，优先等级越高。优先级调整后，预计1分钟内生效。

审计策略将按优先级从小到大顺序执行，某条语句匹配到较高优先级的策略后，将不会执行后续优先级的策略。

审计实例： tdsql10246 ▼ + 新建策略

优先级	策略名	状态	响应方式	审计实例	规则	创建时间/修改时间	操作
1	测试策略	启动	记录/放行	tdsql10246	测试	2016-06-12 14:30:35 / 2016-06-12 14:30:...	^ v 修改 删除
2	测试二	启动	记录/放行	tdsql10246	测试	2016-06-12 14:38:04 / 2016-06-12 14:38:...	^ v 修改 删除

4、您可以通过修改功能，实时修改审计策略；修改完成后预计5分钟内生效，并按新策略进行批评，修改审计策略前的日志不会被修改。

2.4 查看日志

匹配到审计策略的SQL语句将展现在审计日志页面，您可以直接单击查看或搜索。注意事项：

- 因设计要求，审计日志明文方式展现，我们再次建议您开启[二次登录认证](#)，以确保日志可控。
- 日志将以审计策略创建时开始记录，历史数据不做记录。
- 事务、存储过程等可能会被记录为单条语句，详见[数据库审计已支持语法说明](#)
- 目前支持单条SQL语句最大1k，超过部分会被截断。

数据库审计已支持语法

最近更新时间：2017-11-21 17:19:23

通过测试，数据库审计目前已经支持绝大多数SQL语句，如有发现存在不足，请联系腾讯云工作人员或发送邮件到 vitosu@tencent.com 反馈。

1、已支持DCL,DDL,DML语句的解析

Insert, Replace, Select, Union, Update, Delete, CreateDatabase, CreateEvent, CreateFunction, CreateIndex, CreateLog, CreateTable, CreateServer, CreateProcedure, CreateTablespace, CreateTrigger, CreateView, CreateUDF, CreateUser, ShowCharset, ShowCollation, ShowColumns, ShowCreate, ShowCreateDatabase, ShowDatabases, ShowEngines, ShowError, ShowEvents, ShowFunction, ShowGrants, ShowLogEvents, ShowLogs, ShowProcedure, ShowOpenTables, ShowPlugins, ShowProcessList, ShowMasterStatus, ShowPrivileges, ShowProfiles, ShowSlaveHosts, ShowSlaveStatus, ShowTableStatus, ShowWarnings, ShowVariables, ShowStatus, ShowTriggers, Call, DropProcedure, DropDatabase, DropEvent, DropFunction, DropIndex, DropLogfile, DropServer, DropTables, DropTablespace, DropTrigger, DropUser, DropView, AlterDatabase, AlterEvent, AlterFunction, AlterLogfile, AlterProcedure, AlterServer, AlterTable, AlterTablespace, AlterUser, AlterView, Rollback, Commit, Begin, Set, SetTrans, SetPassword, Release, Grant, RenameTable, RenameUser, Revoke, Install, StopSlave, StartSlave, StartTrans, Use, DescribeTable, DescribeStmt, Flush, Load, LoadIndex, FlushTables, Reset, CacheIndex, TruncateTable, Lock, Unlock, SavePoint, Help, Do, SubQuery, ShowTables, Execute, Deallocate, Binlog, Kill, Partition, PrepareRepairXACheckCheckSumAnalyzeChangeOptimizePurgeHandlerSignalResignal

2、事务和存储过程可能被拆分为多条语句