TDSQL-C MySQL 版

操作指南



腾讯云

版权所有：腾讯云计算（北京）有限责任公司

【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内容。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】

腾讯云

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
- 操作总览
- 控制台切换集群页面视图
- 数据库连接
  - 配置服务器
    - 快速配置 Linux 云服务器
    - 快速配置 Windows 云服务器
  - 连接集群
    - 概览
    - 使用 DMC 数据库管理控制台连接集群
    - 使用内网或外网地址通过 Linux 云服务器连接集群
    - 使用内网或外网地址通过 Windows 云服务器连接集群
    - 云联网连接
  - 网络地址配置
    - 开启或关闭外网地址
    - 通过 CLB 开启外网服务
    - 修改内网地址
    - 更换网络
- 实例管理
  - 实例管理简介
  - 修改实例名
  - 修改字符集
  - 修改实例维护时间
  - 重启实例
  - 删除实例
  - 恢复实例
  - 任务列表
- 配置变更
  - 调整计算配置
  - 调整存储空间
  - 升级内核小版本
- 实例形态管理
  - 实例形态简介
  - 预置资源集群启用或关闭 Serverless
  - 预置资源实例转 Serverless
- 集群管理
  - 集群管理简介
  - 修改集群名
  - 修改集群项目
  - 删除集群
  - 恢复集群
- 只读实例管理
  - 只读实例简介
  - 创建只读实例
- 数据库代理
  - 数据库代理概述
  - 注意事项
  - 数据库代理内核特性
    - 内核小版本更新说明
    - 升级数据库代理内核小版本
  - 管理数据库代理

版权所有：腾讯云计算（北京）有限责任公司

开通数据库代理
修改或删除连接地址
设置单个或多个连接地址
查看和修改访问策略
调整数据库代理配置
切换数据库代理网络
查看数据库代理监控
重新负载均衡
关闭数据库代理
事务拆分功能
防闪断功能
接入模式
实例故障转发
自动读写分离功能
自动读写分离介绍
设置数据库代理读写属性
一致性级别
连接池功能
连接池概述
设置会话级连接池
其他功能
Hint 语法使用
账号管理
账号简介
控制台账号
注册腾讯云账号
实名认证
数据库账号
创建账号
自定义密码强度
简介
授权子用户使用功能
开启和关闭自定义密码强度
修改参数设置和参数同步
重置密码
修改账号权限
修改授权访问的主机地址
修改账号连接数
修改账号备注
克隆账号
删除账号
数据库管理
创建数据库
修改数据库账号权限
删除数据库
数据库管理工具（DMC）
DMC 简介
DMC 管理
参数配置
参数概述
设置实例参数
使用参数模板
参数公式

参数配置建议
多可用区部署
多可用区部署概述
设置多可用区部署
切换可用区
变更可用区部署
可用区迁移
全球数据库
概述
技术架构
使用功能
创建全球数据库
管理从集群
配置变更
连接全球数据库
删除全球数据库
全球数据库实践教程
备份与恢复
备份与回档概述
备份与回档费用说明
执行备份
自动备份
手动备份
跨地域备份
管理备份
查看备份列表
设置备份保留时间
设置备份下载规则
下载和解压备份
删除备份
回档
使用逻辑备份恢复数据
通过 Linux 恢复数据
通过控制台恢复数据
库表级回档至原集群
整集群回档至新集群
使用快照备份恢复数据
库表级回档至原集群
集群回档优化方案
整集群回档至新集群
操作日志
慢日志概述
查询和下载慢日志明细
错误日志
日志服务 CLS
迁移数据
一键迁移（从云数据库 MySQL 迁移至 TDSQL-C MySQL 版）
简介
执行一键迁移
常见问题
使用 DTS 服务迁移
通过命令行工具迁移
并行查询

版权所有：腾讯云计算（北京）有限责任公司

- 简介
  - 背景
  - 实现原理
  - 支持的语句场景和受限场景
- 设置并行查询
  - 开启或关闭并行查询
  - hint 语句控制
  - 查看并行查询
  - 并行查询指标
- 性能测试报告

列存索引 CSI
- 简介
  - 功能简介
  - 支持的语句和受限场景
- 列存索引使用方式
  - 基本设置
  - 参数配置与监控项
- 性能测试报告

分析引擎
- 简介
- 功能限制与兼容性说明
  - 兼容性与使用限制
  - 数据加载限制
  - 函数与运算符说明
    - 位运算支持说明
    - 四则运算支持说明
    - 比较运算支持说明
    - 逻辑运算支持说明
    - 字符串函数支持说明
    - 数学运算函数支持说明
    - 日期函数支持说明
    - 聚合函数支持说明
    - JSON 函数
    - 类型转换函数支持说明
  - 数据类型
- 只读分析引擎管理
  - 开启只读分析引擎
  - 访问只读分析引擎
  - 基本操作
  - 调整配置
  - 多节点高可用
  - 监控告警
  - 管理参数
- 数据加载
  - 加载数据到只读分析引擎
  - 查看数据加载状态
  - 控制数据加载速度
  - 取消部分对象加载到只读分析引擎
- 执行查询
  - 收集统计信息
  - SELECT 语句说明
  - CTE 语法使用说明
  - 窗口函数使用说明

版权所有：腾讯云计算（北京）有限责任公司

执行 SQL 并查看执行计划
特色功能
加速 ETL 回写
分页保序功能
性能优化
通过 HINT 优化 SQL 执行
Runtime Filter 使用手册
延迟物化
Merge Join
列存二级索引
系统变量
系统表
SLOW_LOG
性能白皮书
TPC-H 测试方法
TPC-H 性能结果
数据库安全和加密
安全组
简介
管理云数据库安全组
创建和管理云数据库安全组
修改或添加安全组规则
为云数据库匹配安全组
访问与授权
访问管理概述
授权策略语法
可授权的资源类型
加密
设置 SSL 加密
透明数据加密
监控与告警
简介
监控
支持的监控指标
查看监控
控制台查看监控数据
创建 Dashboard 查看监控数据
云 API 拉取监控数据
告警
监控指标告警
设置告警流程简介
设置告警策略
设置告警通知
查看告警历史
事件告警
事件告警简介
支持的事件类型
设置事件告警
SQL 基本操作
使用 SCF 连接 TDSQL-C MySQL 版
标签
标签简介
创建标签和绑定资源

版权所有：腾讯云计算（北京）有限责任公司

编辑和删除标签

操作指南
操作总览

最近更新时间：2025-06-20 20:43:41

本文介绍 TDSQL-C MySQL 版控制台常用的集群列表页和集群管理页内的相关功能，指导您轻松管理 TDSQL-C MySQL 版。

集群列表页

在 TDSQL-C MySQL 版控制台 的集群列表页可查看集群相关信息，以及管理集群。



功能	介绍
新建	在集群列表上方，选择**新建**，可跳转至购买页创建集群，操作请参见 创建集群 。
一键诊断	在集群列表上方，选择一键诊断，可跳转至 数据库智能管家 DBbrain 控制台，可对账号下集群的健康状态进行查看，可对实例进行诊断优化，详细了解请参见 诊断优化 。
续费	勾选目标集群后，在集群列表上方，单击**续费**，可为计算计费模式为包年包月的集群进行手动续费，操作请参见 续费说明 。
设置自动续费	勾选目标集群后，在集群列表上方，选择**更多操作 > 设置自动续费**，可将非自动续费且计算计费模式为包年包月的集群设置为每月到期后自动续费，支持批量设置（勾选多个集群），操作请参见 续费说明 。
编辑标签	勾选目标集群后，在集群列表上方，选择**更多 > 编辑标签**，可对集群资源的标签进行编辑管理，操作请参见 编辑和删除标签 。

集群管理页

登录 TDSQL-C MySQL 版控制台，在集群列表，单击集群 ID，可以进入集群管理页面。您可以切换查看集群详情、监控告警、账号管理、数据库管理、数据库代理、参数设置、安全组、备份管理、操作日志、资源包。



功能页	功能项	介绍
集群详情	修改集群名称	单击集群名称后的编辑图标，可以自定义集群名。

	详情/升级	单击详情/升级，可查询当前集群下所有实例的数据库内核版本，以及最新数据库内核版本的发布时间，当数据库内核版本不为最新版时，您可以选择手动升级。操作请参见 升级内核小版本 。
	调整	在所属项目后，单击调整，可以分配集群的所属项目。
	设置	如果集群计算计费模式为包年包月，可以单击续费设置后的设置，进行续费。
	存储计费模式	单击计费模式后的编辑图标，可以为存储计费模式为包年包月的集群调整存储空间大小，也可以将存储计费模式为按量计费的转为包年包月。
	部署方式	单击部署方式后的编辑图标，可以将集群从单可用区切换为多可用区，也可以从多可用区切换为单可用区，操作请参见 多可用区部署 。
	主备切换	如果集群是多可用区部署，单击主可用区和备可用区之间的切换图标，可以进行主备切换（整集群级别），操作请参见 切换可用区 。
	开启数据库代理	未开启数据库代理的集群，单击开启数据库代理，可从集群详情页快捷进入开通页面。
	重启	单击重启，可对实例进行重启操作。 • 重启期间，实例将无法正常访问，已有的连接会断掉，请您做好准备，以免造成影响。 • 重启期间，如果业务写入量很大，脏页过多，会导致重启失败。重启失败后，实例回到重启之前状态，实例仍可访问。 • 请确保在业务低峰期重启，保证重启成功率，降低对业务的影响。
	调整配置	单击调整配置的图标或者调整配置文字，可以对集群的计算配置进行调整，操作请参见 调整计算配置 。
	一键诊断	在读写实例或者只读实例版块，单击一键诊断图标，可跳转至 数据库智能管家 DBbrain 控制台 ，可对当前实例的健康状态进行查看，可对当前实例进行诊断优化，详细了解请参见 诊断优化 。
	退还/退款	在读写实例或者只读实例版块，单击退还退款图标，可销毁对应实例，需注意，实例形态为预置资源时，若集群下有读写实例和只读实例，则需先退还退款完只读实例，才可以退还退款读写实例；实例形态为 Serverless 时，若为单节点版，支持在读写实例版块，单击退还退款图标进行销毁，若为集群版，则不支持通过退还退款图标进行销毁，此时需要通过调整配置来调整只读实例的数量。
	修改实例名	单击实例名称后的编辑图标，可以自定义实例名。
	更换网络	单击所属网络后的更换网络，可以修改读写实例或者只读实例的网络。
	修改内网地址和端口	单击读写实例或只读实例后的编辑图标，可以修改读写实例或只读实例的内网地址和端口。
	开启外网	单击读写实例或只读实例外网后的开启，可以开启读写实例或只读实例的外网，操作请参见 开启或关闭外网地址 。
	添加只读实例	实例形态为预置资源时，单击添加只读实例，可为集群新增只读实例；实例形态为 Serverless 时，仅能通过调整配置操作来增加只读实例。
	扩容	实例形态为预置资源，且集群存储计费模式为包年包月时，可以下方存储规格后单击扩容，进行存储空间调整。
监控告警		您可以在监控告警页面，选择查询集群下读写实例或者只读实例的资源、引擎、部署相关的监控情况，可以添加时间对比、筛选时间粒度，详细请参见 监控与告警 。
账号管理	创建账号	除了系统默认的 root 账号，您也可以创建新的账号。
	重置密码	忘记密码或者更新密码，可以通过重置密码操作，root 账号和非 root 账号均支持。
	重置权限	可以对 root 账号重置权限，恢复默认系统分配权限。
	修改权限	可以对非 root 账号进行权限授予和修改。
	修改主机	可以对非 root 账号进行授权主机修改。
	修改连接数	可以修改非 root 账号限制的连接数。
	删除账号	可以删除非 root 账号。
	修改备注	可以为非 root 账号修改备注。

	克隆	可以选择一个非 root 账号进行克隆创建。
	自定义密码强度	支持自定义密码强度功能，开启后，将对创建账号、重置密码和账户克隆功能的密码设定进行限制。详细请参见 自定义密码强度 。
数据库管理	创建数据库	单击创建数据库，可为集群新建数据库。
	数据导入	单击数据导入，可进入数据库管理平台 DMC，进行数据导入到数据库操作。
	批量删除	单击**更多操作** > **批量删除**，可删除选中的多个非系统创建的数据库。
	删除	单击**删除**，可对应删除选中的非系统创建的数据库。
	修改	单击修改，可添加或修改具体账号对选中数据库的相关权限或备注。
数据库代理	立即开启	未开启数据库代理的集群，单击立即开启，会进入开启数据库代理的配置页面。
	概览	在数据库代理概览页，您可以查看数据库代理基本信息、调整数据库代理节点、关闭数据库代理、新增访问地址、调整数据库代理访问地址的配置、删除数据库代理访问地址、重新负载均衡等操作。
	访问策略	在数据库代理访问策略页，您可以查询当前数据库代理访问地址的部署情况、调整数据库代理访问地址配置、添加节点、添加只读实例等操作。
	性能监控	在数据库代理性能监控页，您可以查询每个数据库代理节点的监控情况，也可以单击设置告警，跳转至腾讯云可观测平台进行告警设置。
参数管理	修改参数	支持批量或单个修改参数的运行值。
	导入参数	支持从本地导入已有的参数配置模板。
	导出参数	支持将当前的参数设置导出为模板到本地。
	从模板导入	支持从控制台已有的参数模板中导入参数设置。
	另存为模板	支持将当前的参数设置另存为模板到控制台。
	最近修改记录	单击最近修改记录，可查询集群下对应实例的参数最近修改的记录。
安全组		在安全组页，您可以查看或配置集群下读写实例、只读组、数据库代理访问地址的安全组，配置多个安全组时，支持调整安全组优先级顺序或删除多余安全组。
备份管理	回档	支持通过快照备份，根据备份文件或者时间进行回档，操作请参见 使用快照备份恢复数据 。
	手动备份	可选择快照备份对整个集群进行备份，也可以选择逻辑备份对整个集群或者特定的库/表进行备份，当前系统每小时仅允许发起一次手动备份。详细请参见 手动备份 。
	自动备份设置	自动备份中快照备份为默认开启，逻辑备份需要手动开启，支持修改备份保留时间、设置逻辑备份开始时间段、开启跨地域逻辑备份等。详细请参见 自动备份 。
	日志备份设置	日志备份设置中，binlog 备份保留时间默认开启，可以修改保留时间，可以开启 binlog 日志的跨地域备份。
操作日志	慢日志	支持查询和下载慢日志。
	错误日志	支持查询和下载错误日志。
	日志服务 CLS	采集来源为 TDSQL-C MySQL 版实例的慢日志数据，并投递至日志服务（Cloud Log Service，CLS）进行分析。详细请参见 日志服务 CLS 。
资源包		实例形态为 Serverless 的集群支持资源包管理，可在资源包页面购买资源包、绑定资源包、调整资源包优先级顺序等操作。

控制台切换集群页面视图

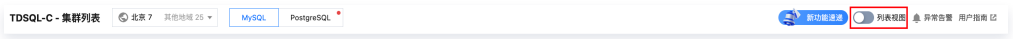
最近更新时间：2025-01-14 17:27:22

操作场景

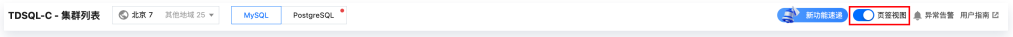
TDSQL-C MySQL 版控制台集群列表页面支持页签及列表视图，您可以参考本文进行页面视图切换。
在集群数小于等于10个时，相比列表视图，页签视图将为您呈现更加清晰的集群信息与实例详情。

操作步骤

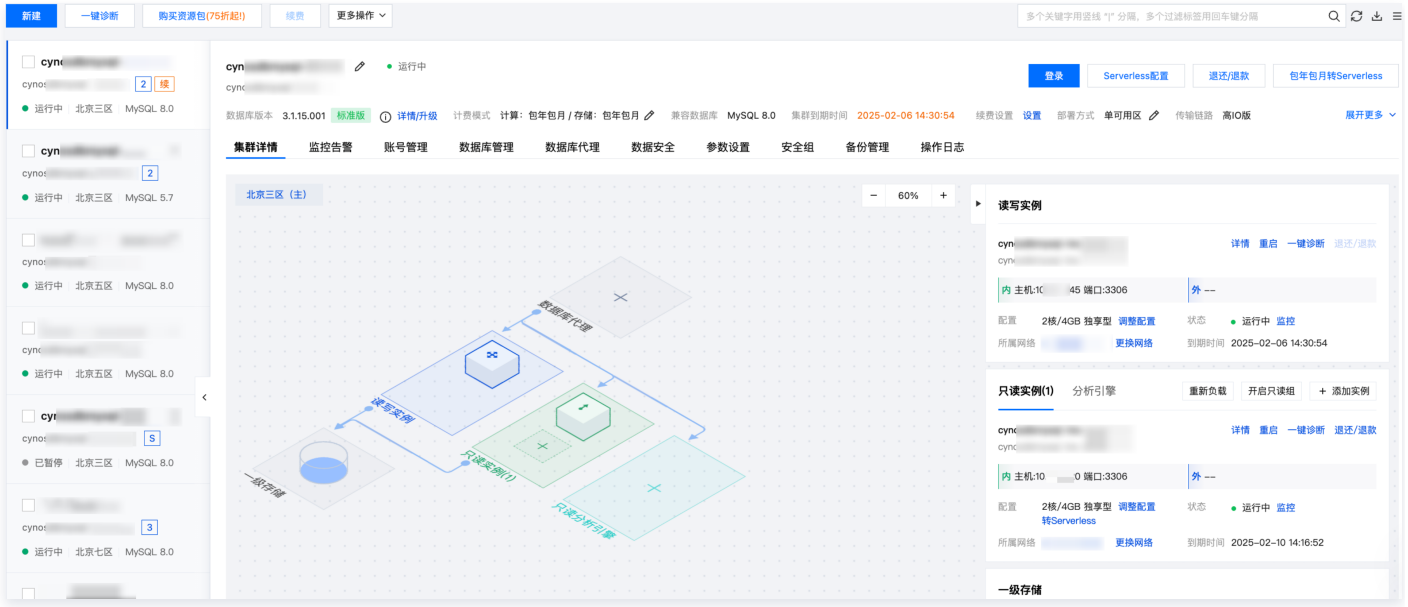
1. 登录 控制台，在集群列表右上方开启列表视图前的按钮。
   按钮开启前：



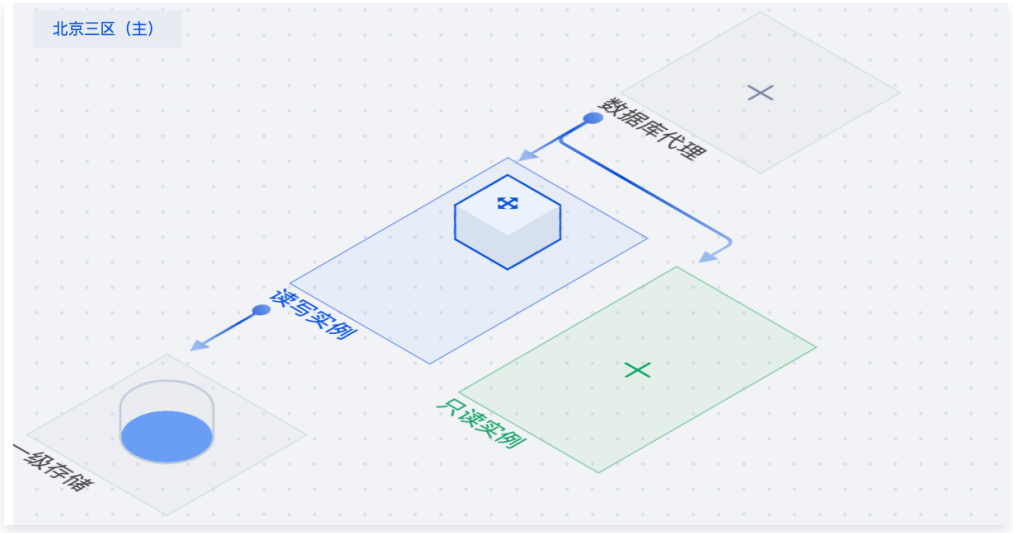
   按钮开启后：



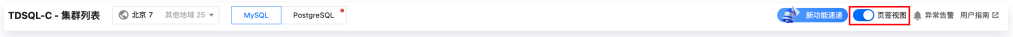
2. 成功切换后，页签视图如下：



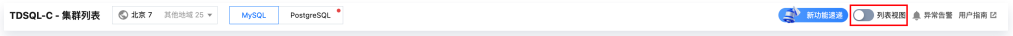
3. 您可以在左侧**集群列表**单击需要查询的集群，进入对应集群管理页。
4. 在**集群管理**页，您可对集群进行数据库版本升级、登录集群、销毁/退款、调整所属项目、更换网络、修改部署方式、编辑标签、续费设置、计费模式设置等操作。
5. 在**集群详情**页面，您可了解 TDSQL-C MySQL 版的集群架构和存储详情，并可对实例进行相关操作，详细操作请参见 实例管理简介 。



6. 如需切换列表视图，可在控制台右上方将页签视图前的按钮关闭。
   按钮关闭前：

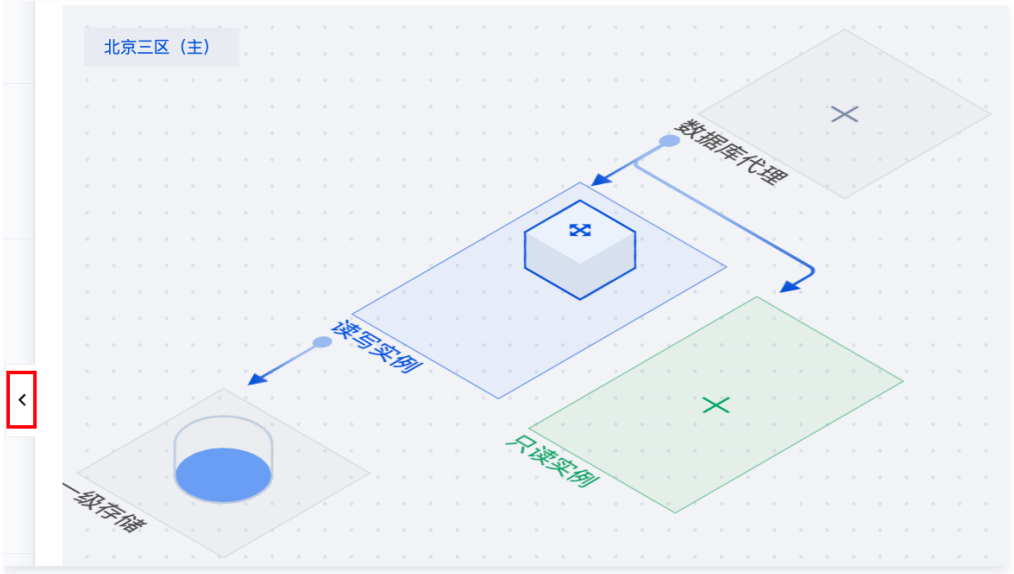


   按钮关闭后：



隐藏集群列表

在页签视图下，支持隐藏**集群列表**，便于更清晰的预览对应集群管理页，在任一集群管理页，单击下图中红框内的图标即可。再次单击对应图标，集群列表会重新展开。



数据库连接
配置服务器
快速配置 Linux 云服务器

最近更新时间：2024-05-13 14:58:55

云服务器（Cloud Virtual Machine，CVM）是腾讯云提供的可扩展的计算服务。使用云服务器避免了使用传统服务器时需要预估资源用量及前期投入，帮助您在短时间内快速启动任意数量的云服务器并即时部署应用程序。
本文为您介绍如何从零开始，以最简单的方式搭建一个 Linux 云服务器。您可按照以下文档，购买和配置您的第一台云服务器。若想了解搭建 Windows 云服务器的入门教程，可以参考 快速配置 Windows 云服务器 。

前提条件

已注册腾讯云账号，详细方法请参见 注册腾讯云账号 。

配置 Linux 云服务器

> **注意：**
> 为确保成功连接到 TDSQL-C MySQL 版，建议购买的云服务器 CVM 与 TDSQL-C MySQL 版满足以下几点：
> - CVM 和 TDSQL-C MySQL 版属于同一个腾讯云主账号。
> - CVM 和 TDSQL-C MySQL 版位于同一地域。
> - CVM 和 TDSQL-C MySQL 版的网络类型都是 VPC 且处于同一个 VPC 内。

步骤一：购买 Linux 云服务器

1. 进入 快速购买页面 。
2. 在购买页 > **快速配置**分页，完成云服务器配置后，单击**立即购买**。



配置说明如下：

配置项	说明
地域	选择与您的 TDSQL-C MySQL 版集群所在同一个地域。

实例	选择您需要的云服务器机型配置。这里我们选择 “普及配置（2核4GB）”。
操作系统	选择您需要的云服务器操作系统。这里我们选择 “CentOS 7.9 64位”。
公网 IP	勾选后会为您分配公网 IP，默认公网带宽为 “1Mbps”，您可以根据需求调整。
登录方式	自动生成的密码将在服务器创建完成后通过 站内信 发送。
默认配置	可展开查看可用区、安全组等6项默认配置。
自动续费	勾选后，若账户余额足够，则将在云服务器到期时按月自动续费。
协议	查阅并了解相关协议后勾选。
时长	购买时长，默认为 “1个月”。
数量	购买数量，默认为 “1台”。

当您付费完成后，即完成了云服务器的购买。云服务器可以作为个人虚拟机或者您建站的服务器。接下来，您可以登录您购买的这台服务器。

步骤二：登录云服务器

> **注意：**
> 通过快速配置购买的云服务器，系统将为您自动分配云服务器登录密码并发送到您的站内信中。此密码为登录云服务器的凭据。您可至 站内信 查收您的密码。

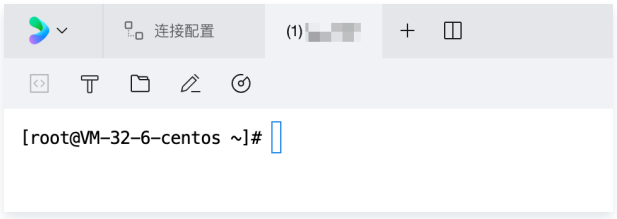
1. 登录 云服务器控制台，在上方选择地域。
2. 在实例列表中找到您刚购买的云服务器，在右侧操作栏中单击**登录**。



3. 在“标准登录 | Linux 实例”窗口中，输入**云服务器**的用户名（默认为 root）和密码，并单击**登录**即可正常登录。如下图所示：



4. 登录成功后，界面如下图所示：



快速配置 Windows 云服务器

最近更新时间：2024-05-13 14:58:55

云服务器（Cloud Virtual Machine，CVM）是腾讯云提供的可扩展的计算服务。使用云服务器避免了使用传统服务器时需要预估资源用量及前期投入，帮助您在短时间内快速启动任意数量的云服务器并即时部署应用程序。
本文为您介绍如何从零开始，以最简单的方式搭建一个 Windows 云服务器。您可按照以下文档，购买和配置您的第一台云服务器。若想了解搭建 Linux 云服务器的入门教程，可以参考 快速配置 Linux 云服务器 。

前提条件

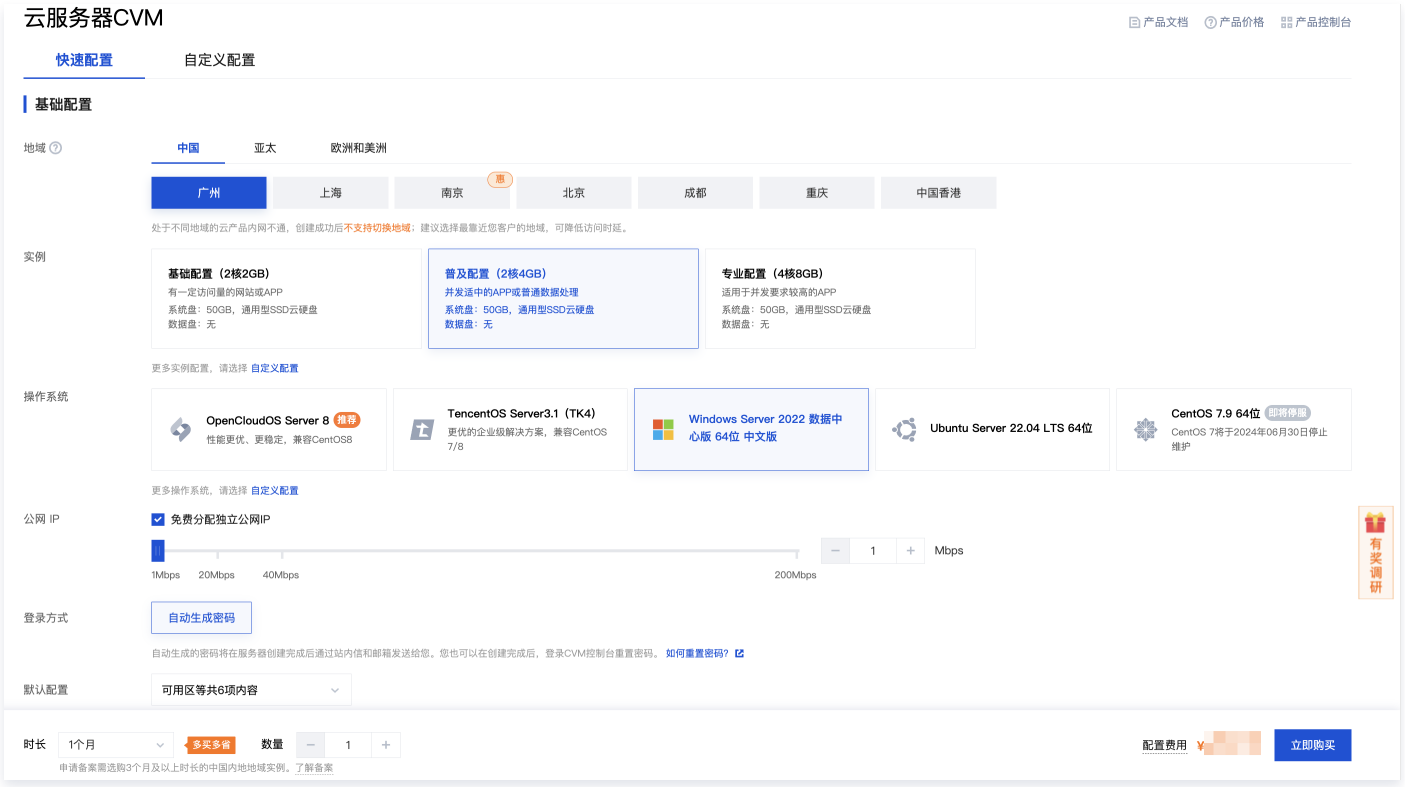
已注册腾讯云账号，详细方法请参见 注册腾讯云账号 。

配置 Windows 云服务器

> **注意：**
> 为确保成功连接到 TDSQL-C MySQL 版，建议购买的云服务器 CVM 与 TDSQL-C MySQL 版满足以下几点：
> - CVM 和 TDSQL-C MySQL 版属于同一个腾讯云主账号。
> - CVM 和 TDSQL-C MySQL 版位于同一地域。
> - CVM 和 TDSQL-C MySQL 版的网络类型都是 VPC 且处于同一个 VPC 内。

步骤一：购买 Windows 云服务器

1. 进入 快速购买页面 。
2. 在购买页 > **快速配置**分页，完成云服务器配置后，单击**立即购买**。



配置说明如下：

配置项	说明
地域	选择与您的 TDSQL-C MySQL 版集群所在同一个地域。
实例	选择您需要的云服务器机型配置。这里我们选择 “普及配置（2核4GB）”。
操作系统	选择您需要的云服务器操作系统。这里我们选择 “Windows Server 2022 数据中心版 64位中文版”。

公网 IP	勾选后会为您分配公网 IP，默认公网带宽为 “1Mbps”，您可以根据需求调整。
登录方式	自动生成的密码将在服务器创建完成后通过 站内信 发送。
默认配置	可展开查看可用区、安全组等6项默认配置。
自动续费	勾选后，若账户余额足够，则将在云服务器到期时按月自动续费。
协议	查阅并了解相关协议后勾选。
时长	购买时长，默认为 “1个月”。
数量	购买数量，默认为 “1台”。

当您付费完成后，即完成了云服务器的购买。云服务器可以作为个人虚拟机或者您建站的服务器。接下来，您可以登录您购买的这台服务器。

步骤二：登录云服务器

> **注意：**
> 通过快速配置购买的云服务器，系统将为您自动分配云服务器登录密码并发送到您的站内信中。此密码为登录云服务器的凭据。您可至 站内信 查收您的密码。

1. 登录 云服务器控制台 。
2. 在实例的管理页面，根据实际使用的视图模式进行操作：

列表视图

找到需要登录的 Windows 云服务器，单击右侧的**登录**。如下图所示：



页签视图

选择需要登录的 Windows 云服务器页签，单击**登录**。如下图所示：

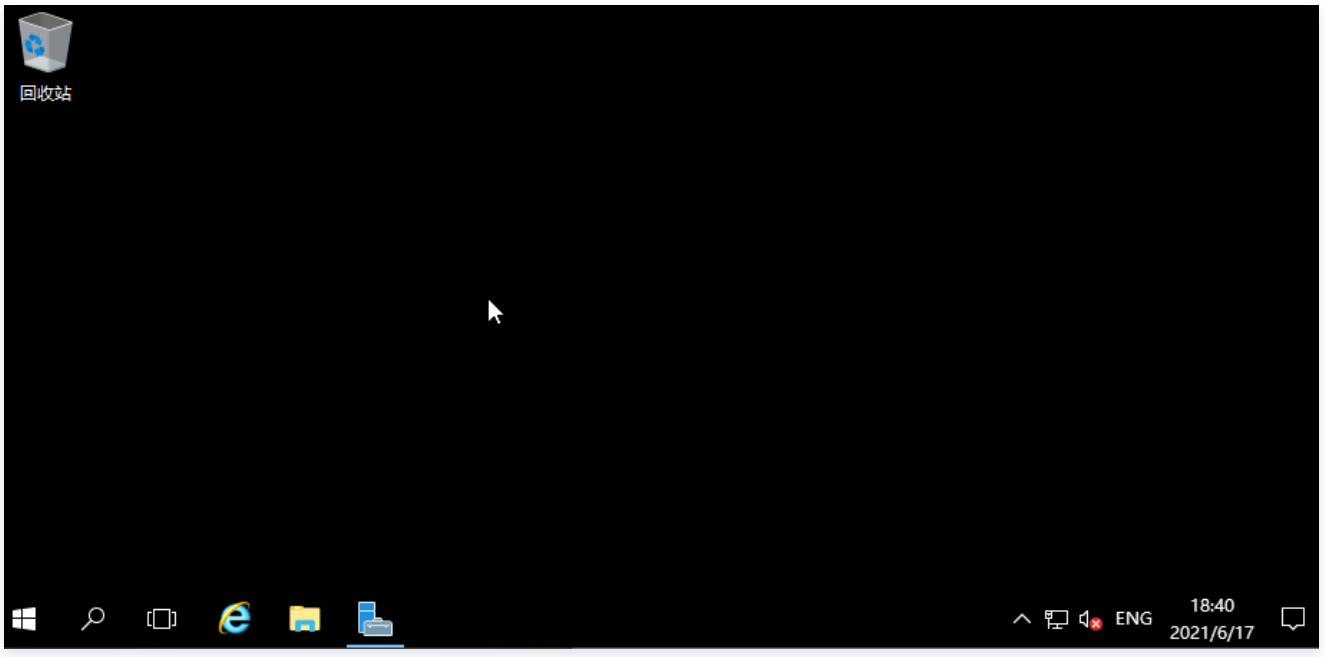


3. 在打开的“标准登录 | Windows 实例”窗口中，根据实际情况填写登录信息。如下图所示：



- **端口**：默认为3389，请按需填写。
- **用户名**：Windows 实例用户名默认为 `Administrator`，请按需填写。
- **密码**：填写已从 站内信 中获取的登录密码。

4. **单击登录**，即可登录 Windows 实例。
   本文以登录操作系统为 Windows Server 2016 数据中心版64位中文版的云服务器为例，登录成功则出现如下图所示界面：



连接集群
概览

最近更新时间：2024-06-13 10:19:31

本章节为您介绍连接 TDSQL-C MySQL 版的方式、相关应用场景以及连接相关故障处理。

> **说明：**
> 外网连接易受网络影响，如您对网络有较高要求，推荐使用内网连接或者云联网连接。

连接方式

连接方式	应用场景	是否计费	常见问题
使用 DMC 数据库管理控制台连接集群	无 CVM，直接通过 TDSQL-C MySQL 版控制台登录到 DMC 数据库管理控制台进行数据库管理。具体场景说明请参见 场景一 。	暂不收费	点击跳转
使用内网或外网地址通过 Linux 云服务器连接集群	CVM 为 Linux 云服务器，可使用内网或外网地址连接。具体场景说明请参见 场景二 。	暂不收费	点击跳转
使用内网或外网地址通过 Windows 云服务器连接集群	CVM 为 Windows 云服务器，可使用内网或外网地址连接。具体场景说明请参见 场景三 。	暂不收费	点击跳转
云联网连接	CVM 和 TDSQL-C MySQL 版 处于不同 VPC，或使用轻量应用服务器连接 TDSQL-C MySQL 版。具体场景说明请参见 场景四 。	云联网计费说明	点击跳转

应用场景说明

场景一：控制台连接

无 CVM，通过腾讯云账号登录 TDSQL-C MySQL 版控制台，找到目标集群，在集群管理页单击**登录**，跳转至 DMC 数据库管理控制台 登录后进行数据库管理的场景，DMC 可提供库表级操作、实时监控、实例会话管理、SQL 窗口、数据管理等服务。

场景二：使用内网或外网地址通过 Linux 云服务器连接集群

- **内网地址连接**
  CVM 为 Linux 云服务器，和 TDSQL-C MySQL 版属于同一个腾讯云主账号，位于同一个地域，网络类型都是 VPC 且处于同一个 VPC 内。
  举例：
  例如账号1下，广州 VPC1 中的子网 A 下的 CVM，和账号1下，广州 VPC1 中的子网 B 下的 TDSQL-C MySQL 版。
- **外网地址连接**
  CVM 为 Linux 云服务器，无法使用内网连接的场景，可以通过外网连接，需要 TDSQL-C MySQL 版已 开启外网 。
  举例：
  - 例如账号1下，广州 VPC1 中的子网 A 下的 CVM 和广州地域下开启了外网的 TDSQL-C MySQL 版。
  - 例如账号1下，广州 VPC1 中的子网 A 下的 CVM 和北京地域下开启了外网的 TDSQL-C MySQL 版。
  - 例如账号1下，广州 VPC1 中的子网 A 下的 CVM 和账号2下，北京地域下开启了外网的 TDSQL-C MySQL 版。

场景三：使用内网或外网地址通过 Windows 云服务器连接集群

- **内网地址连接**
  CVM 为 Windows 云服务器，和 TDSQL-C MySQL 版属于同一个腾讯云主账号，位于同一个地域，网络类型都是 VPC 且处于同一个 VPC 内。
  举例：
  例如账号1下，广州 VPC1 中的子网 A 下的 CVM，和账号1下，广州 VPC1 中的子网 B 下的 TDSQL-C MySQL 版。
- **外网地址连接**
  CVM 为 Windows 云服务器，无法使用内网连接的场景，可以通过外网连接，需要 TDSQL-C MySQL 版已 开启外网 。
  举例：
  - 例如账号1下，广州 VPC1 中的子网 A 下的 CVM 和广州地域下开启了外网的 TDSQL-C MySQL 版。
  - 例如账号1下，广州 VPC1 中的子网 A 下的 CVM 和北京地域下开启了外网的 TDSQL-C MySQL 版。

版权所有：腾讯云计算（北京）有限责任公司

- 例如账号1下，广州 VPC1 中的子网 A 下的 CVM 和账号2下，北京地域下开启了外网的 TDSQL-C MySQL 版。

场景四：云联网连接

1. CVM 和 TDSQL-C MySQL 版处于不同 VPC（包括同账号/不同账号，同地域/不同地域）。
2. 同一腾讯云账号下，购买的服务器为 轻量应用服务器，连接 TDSQL-C MySQL 版，需通过云联网连接。

举例：

- 例如账号1下，广州 VPC1 中的子网 A 下的 CVM 和广州 VPC2 中的子网 B 下的 TDSQL-C MySQL 版。
- 例如账号1下，广州 VPC1 中的子网 A 下的 CVM 和北京 VPC2 中的子网 B 下的 TDSQL-C MySQL 版。
- 例如账号1下，广州 VPC1 中的子网 A 下的 CVM 和账号2下，北京 VPC2 中的子网 B 下的 TDSQL-C MySQL 版。
- 例如账号1下，广州 VPC1 中的子网 A 下的轻量应用服务器和广州 VPC2 中的子网 B 下的 TDSQL-C MySQL 版。

连接相关故障处理

请参见 故障处理 > 连接相关 。

使用 DMC 数据库管理控制台连接集群

最近更新时间：2024-12-20 17:15:53

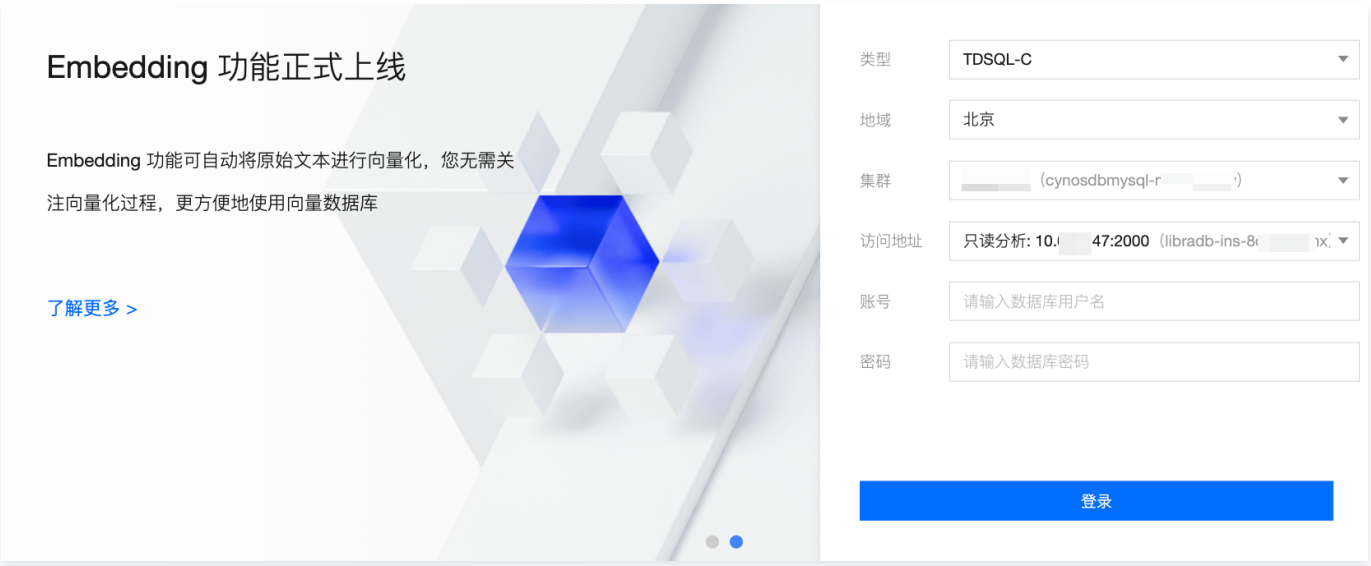
DMC（Database Management Center）是腾讯云的一款数据库管理工具，提供库表级操作、实时监控、实例会话管理、SQL 窗口、数据管理为一体的数据库管理服务。本文为您介绍如何使用 DMC 数据库管理控制台连接 TDSQL-C MySQL 版。

前提条件

- 准备好 TDSQL-C MySQL 版集群，请参见 创建集群 。
- 准备好数据库账号，请参见 创建数据库账号 。

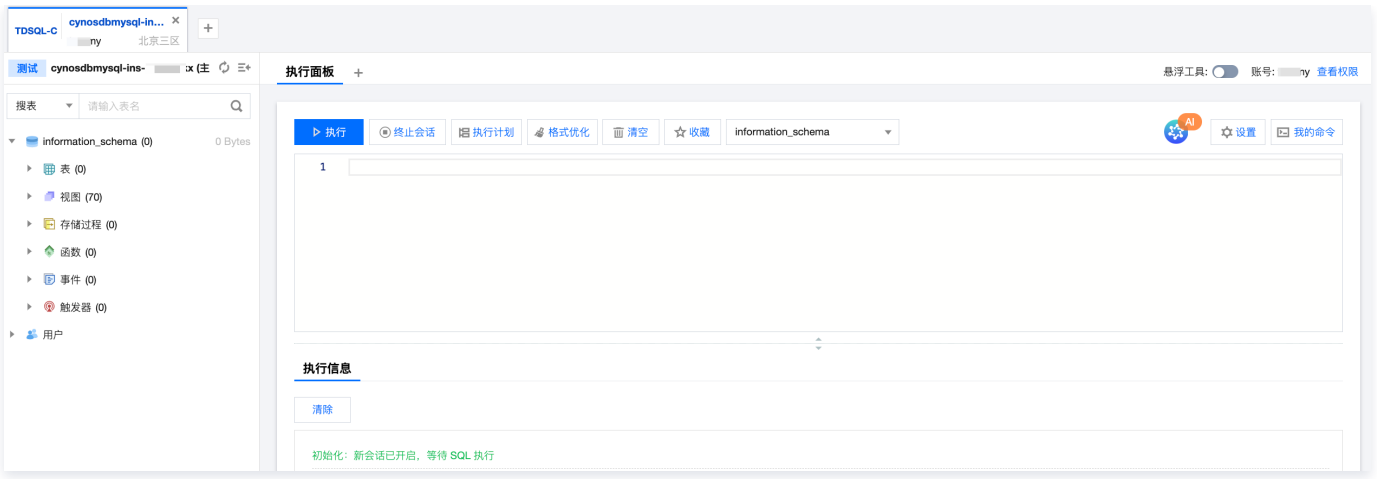
操作步骤

1. 您可在 登录实例 里选择一种方式登录 TDSQL-C MySQL 版。



> **说明：**
> 登录时的选项设置说明如下：
> - **类型**：选择 TDSQL-C。
> - **地域**：选择需要连接的集群所在的地域。
> - **集群**：选择需要连接登录的集群。单击下拉键后可搜索集群 ID 进行快捷筛选。
> - **访问地址**：选择需要连接登录的集群下的具体访问地址，支持选择集群下的**主实例**、**只读实例**、**只读分析引擎**、**数据库代理**。
> - **账号**：输入此集群对应的账号名。
> - **密码**：输入账号对应的密码。

2. 成功登录后的界面如下图所示。



其他连接集群的方法

- 使用内网或外网地址通过 Linux 云服务器连接集群
- 使用内网或外网地址通过 Windows 云服务器连接集群

常见问题

通过 DMC 数据库管理控制台连接是否收费?

免费使用，暂不收费。

登录失败，错误码：ER_ACCESS_DENIED_ERROR[-10001] [2c67ce67-82fc-e5df-7831-18feb87c7367]，如何处理?

原因1：账号或密码错误
解决方法：请 重置密码 后登录，并确保账号名输入正确。
原因2：使用账号的主机地址被限制
解决方法：请 修改授权访问的主机地址 后登录，使通过此账号登录的 IP 地址具备登录权限。
详细说明请参见 DMC 数据库管理控制台连接失败 。

使用内网或外网地址通过 Linux 云服务器连接集群

最近更新时间：2024-10-18 16:22:11

TDSQL-C MySQL 版支持实例级独立 IP 地址，即您可以通过集群下读写实例或任一只读实例的 IP 地址连接访问对应实例。本文为您介绍，如何使用内网或外网地址通过 Linux 云服务器连接 TDSQL-C MySQL 版。

前提条件

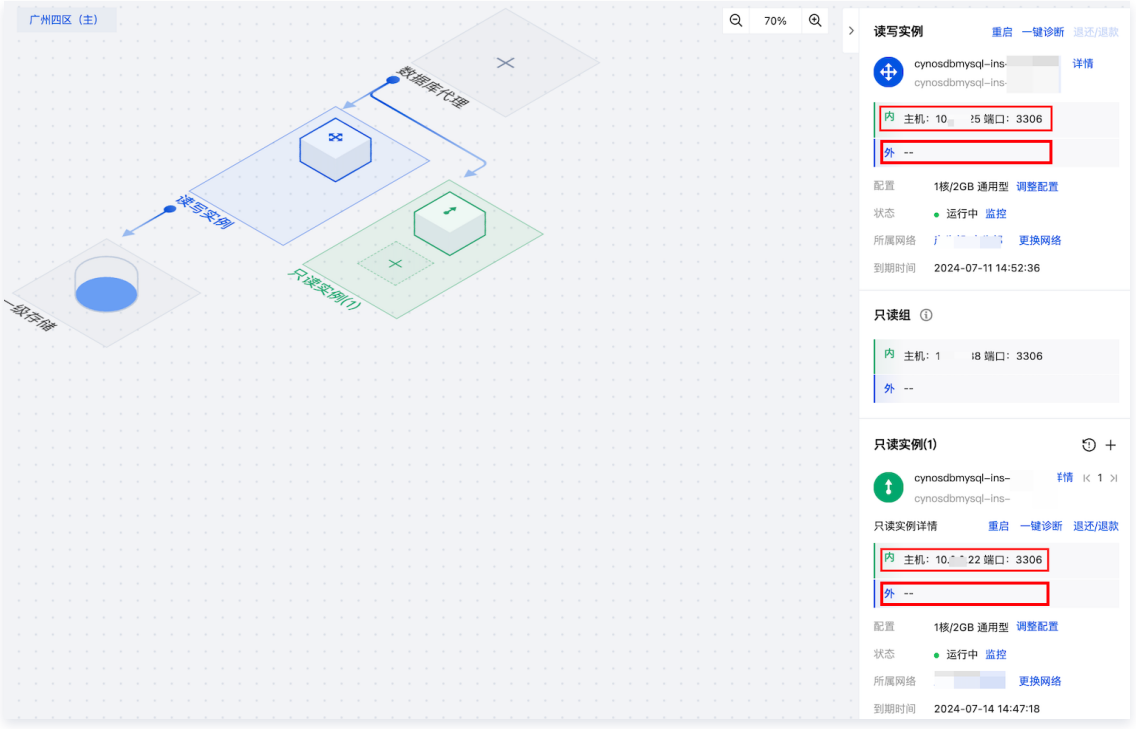
- 准备好 TDSQL-C MySQL 版集群，请参见 创建集群 。
- 准备好数据库账号，请参见 创建数据库账号 。

步骤一：查询需要连接的实例内/外网 IP 地址

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在集群列表页，根据实际使用的视图模式进行操作：

页签视图

1. 在左侧集群列表，单击**目标集群**，进入集群管理页。
2. 在集群详情下，找到需要查询内/外网 IP 地址的实例，在其所属网络下方可查看该实例的内/外网 IP 地址。



列表视图

1. 在集群列表中，找到目标集群，单击**集群 ID** 或操作列的**管理**，进入集群管理页面。
2. 在集群管理页面，选择实例列表页，选择读写实例或只读实例，在**字段：内/外网地址**下查看对应实例的内/外网 IP 地址。

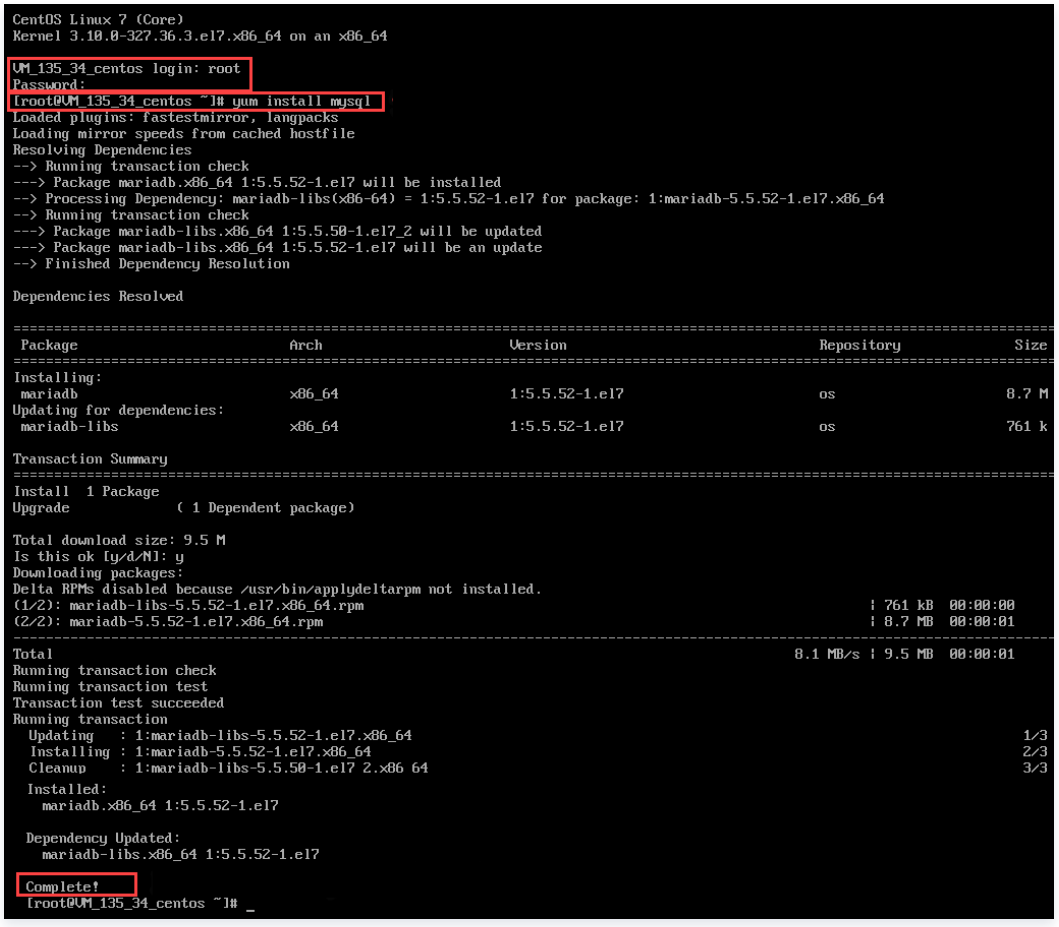


步骤二：连接集群下的目标实例

1. 登录到 Linux 云服务器，请参见 快速配置 Linux 云服务器 。
2. 以 CentOS 7.2 64 位系统的云服务器为例，执行如下命令安装 MySQL 客户端。

```
yum install mysql
```

提示 `Complete!` 说明 MySQL 客户端安装完成。



3. 根据不同连接方式，选择相应的操作：

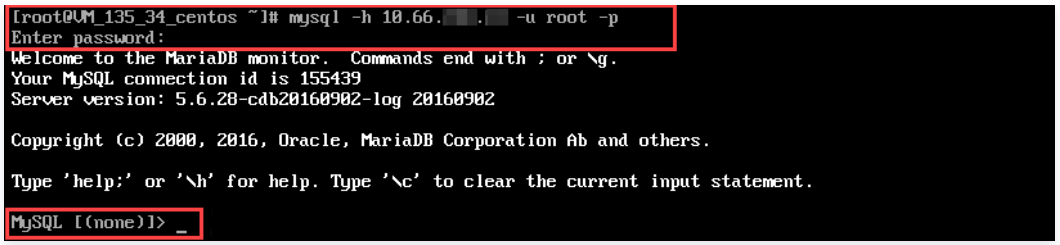
- **内网连接时：**

  3.1 执行如下命令，登录到 TDSQL-C MySQL 版集群。

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

  - hostname：替换为 TDSQL-C MySQL 版集群下目标实例的内网地址，内网地址查询方法参见 步骤一 。
  - port：替换为内网端口号。
  - username：替换为在 前提条件 中创建的账号名，此处以默认的账号名 root 为例。
    示例：内网地址为10.0.168.14:5308，账号名为 root，连接命令输入为 `mysql -h 10.0.168.14 -P 5308 -u root -p` 。

3.2 在提示 `Enter password:` 后输入以上命令中账号对应的密码，如忘记密码可参见 重置密码 进行修改。
本例中提示 `MySQL [(none)]>` 说明成功登录到 TDSQL-C MySQL 版。



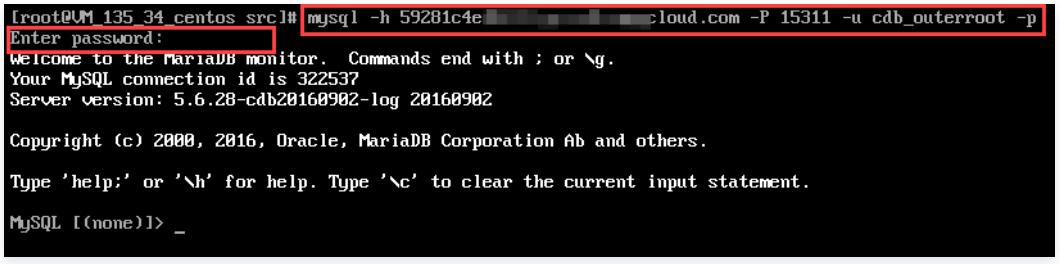
- **外网连接时：**

3.3 执行如下命令，登录到 TDSQL-C MySQL 版集群。

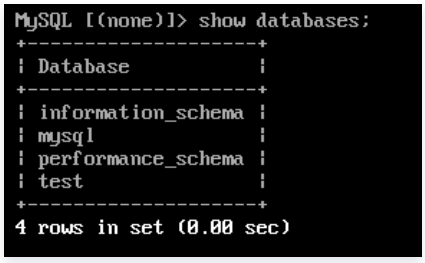
```
mysql -h hostname -P port -u username -p
```

  - hostname：替换为 TDSQL-C MySQL 版集群下目标实例的外网地址，外网地址查询方法参见 步骤一 。若外网地址未开启，请参见 开启或关闭外网地址 。
  - port：替换为外网端口号。
  - username：替换为外网连接的账号名，用于外网连接，建议您在控制台单独创建账号便于连接控制管理。

3.4 在提示 `Enter password:` 后输入外网连接账号名对应的密码，如忘记密码可参见 重置密码 进行修改。
本例中 hostname 为 59281c4exxx.myqcloud.com，外网端口号为15311。



4. 在 `MySQL [(none)]>` 提示符下可以发送 SQL 语句到要执行的 TDSQL-C MySQL 版服务器，具体命令行请参见 mysql Client Commands 。
下图中以 `show databases;` 为例：



常见问题

内网和外网连接是否收费?

内网和外网连接免费使用，暂不收费。

如果忘记账号名以及密码，怎么重置后登录集群?

您可以在 TDSQL-C MySQL 版控制台，集群管理页面的**账号管理**下查看您创建的用于连接集群的账号名，当您忘记密码时，可通过重置密码操作修改密码后再登录集群。



通过内网或外网连接 TDSQL-C MySQL 版是否需要为集群配置安全组?

需要配置安全组，详细步骤请参见 配置安全组 。注意，若开启外网，通过外网连接时，配置的安全组规则需放通内网的访问端口。

如果云服务器的安全组规则匹配不当，导致无法登录 Linux 实例怎么办?

如果云服务器的安全组规则匹配不当，如未放通相应端口导致无法登录 Linux 实例，您可以通过 安全组（端口）验通工具 检查实例当前配置的安全组连通性。



通过一键检测可得知登录失败的可能原因，注意，需放通的端口号，以您实际使用的云服务器操作系统为准，如 Linux 云服务器，通常情况下端口为22或者36000，windows 云服务器，通常情况下端口为3389。



然后在云服务器实例详情页，选择**安全组 > 编辑规则**，然后放通对应端口，详细操作请参见 添加安全组规则 。

无法连接集群还与哪些设置有关?

当无法连接集群时，您可以检查云服务器 CVM 和 TDSQL-C MySQL 是否在同一腾讯云账号下，是否处于同一地域，是否为同一 VPC 网络。需确保以上三点设置相同，才可以正常连接集群。更多连接相关故障问题处理请参见 故障处理 > 连接相关 。

使用内网或外网地址通过 Windows 云服务器连接集群

最近更新时间：2024-07-02 17:05:31

TDSQL-C MySQL 版支持实例级独立 IP 地址，即您可以通过集群下读写实例或任一只读实例的 IP 地址连接访问对应实例。本文为您介绍，如何使用内网或外网地址通过 Windows 云服务器连接 TDSQL-C MySQL 版。

前提条件

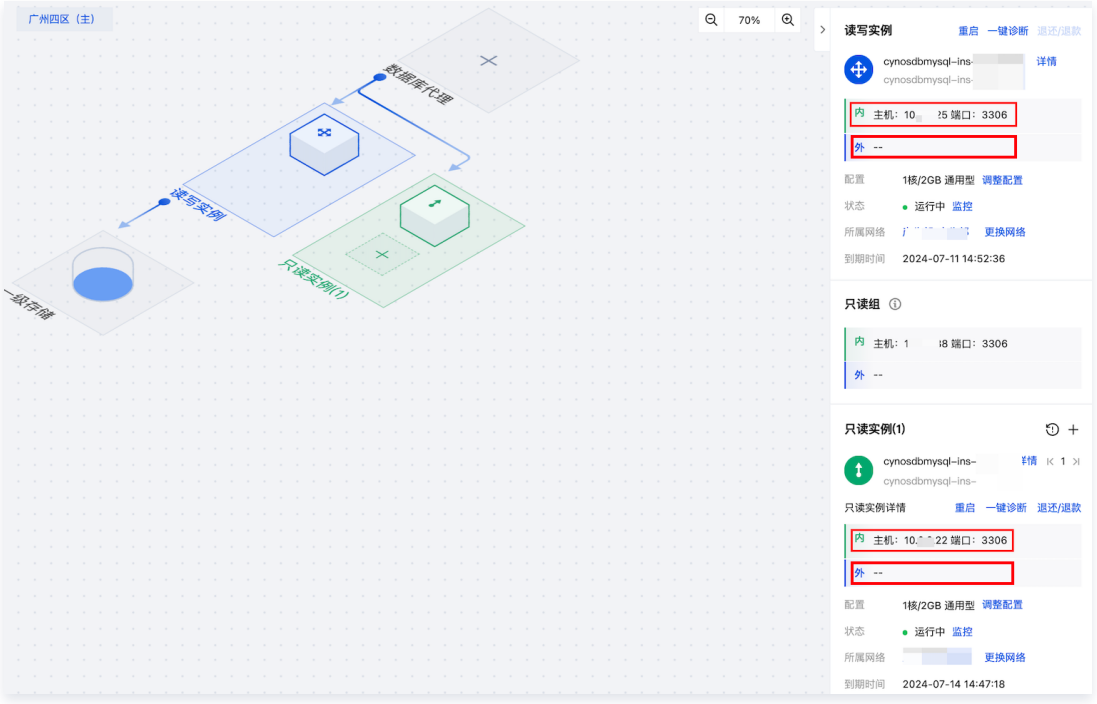
- 准备好 TDSQL-C MySQL 版集群，请参见 创建集群 。
- 准备好数据库账号，请参见 创建数据库账号 。

步骤一：查询需要连接的实例内/外网 IP 地址

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在集群列表页，根据实际使用的视图模式进行操作：

页签视图

1. 在左侧集群列表，单击目标集群，进入集群管理页。
2. 在集群详情下，找到需要查询内/外网 IP 地址的实例，在其所属网络下方可查看该实例的内/外网 IP 地址。



列表视图

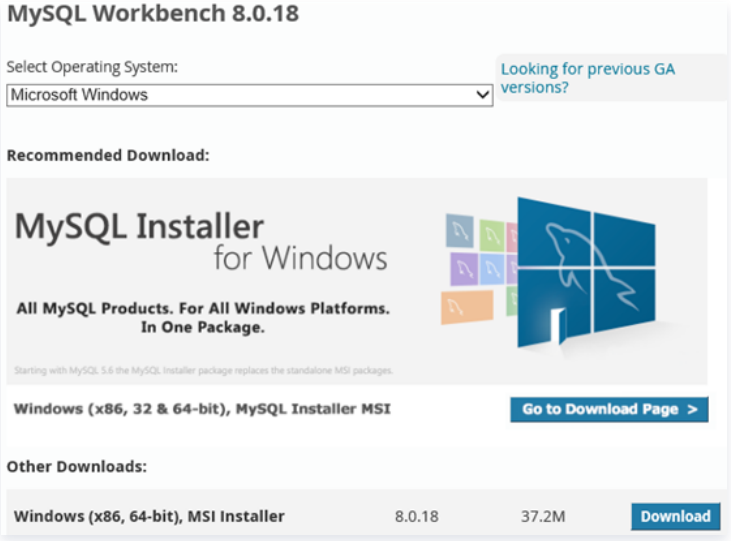
1. 在集群列表中，找到目标集群，单击**集群 ID** 或操作列的**管理**，进入集群管理页面。
2. 在集群管理页面，选择实例列表页，选择读写实例或只读实例，在**字段：内/外网地址**下查看对应实例的内/外网 IP 地址。



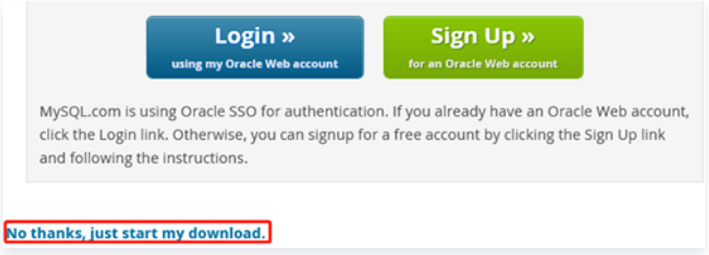
步骤二：连接集群下的目标实例

1. 登录到 Windows 云服务器，请参见 快速配置 Windows 云服务器 。
2. 下载一个标准的 SQL 客户端。

> **说明：**
> 推荐您下载 MySQL Workbench，并根据您的系统来下载适配版本的安装程序，安装程序请参见 下载地址 。

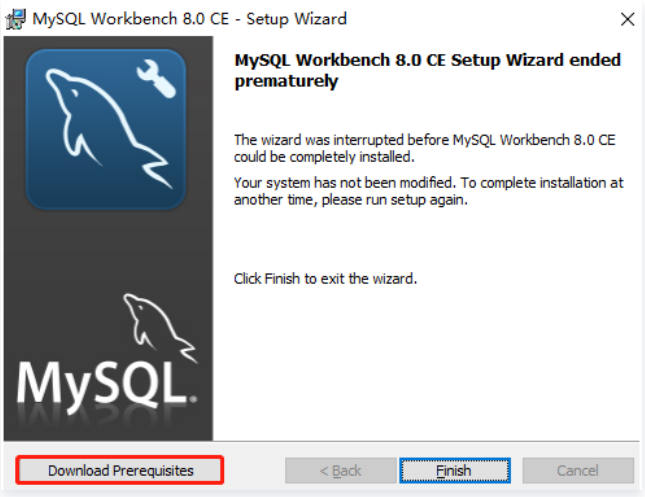


3. 界面将提示 **Login**、**Sign Up** 和 **No, thanks, just start my download.**，选择 **No thanks, just start my download.** 来快速下载。

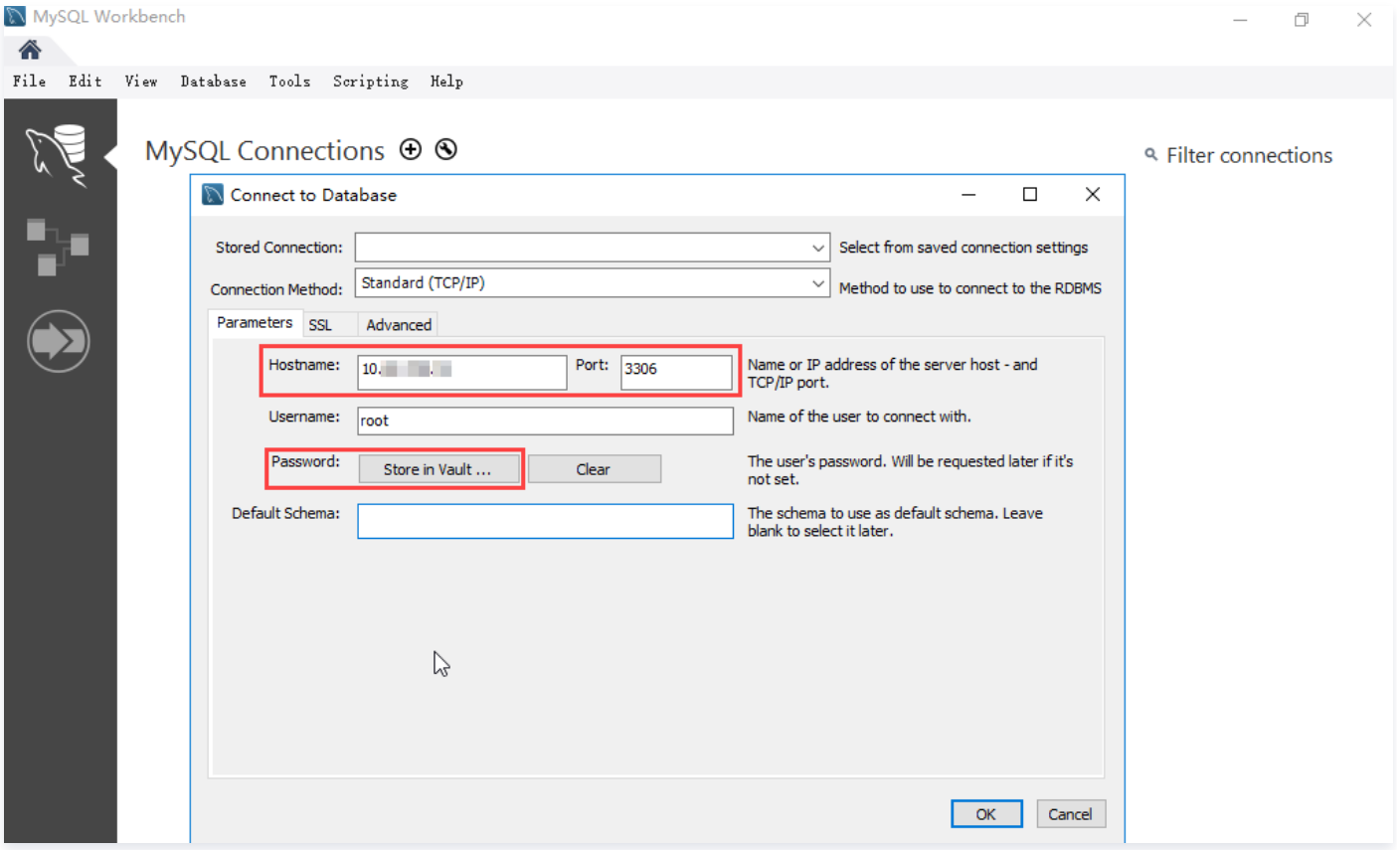


4. 在此台云服务器上安装 MySQL Workbench。

> **说明：**
> - 此电脑上需要安装 Microsoft .NET Framework 4.5 和 Visual C++ Redistributable for Visual Studio 2015。
> - 您可以单击 MySQL Workbench 安装向导中的 **Download Prerequisites**，跳转至对应页面下载并安装这两个软件，然后安装 MySQL Workbench。

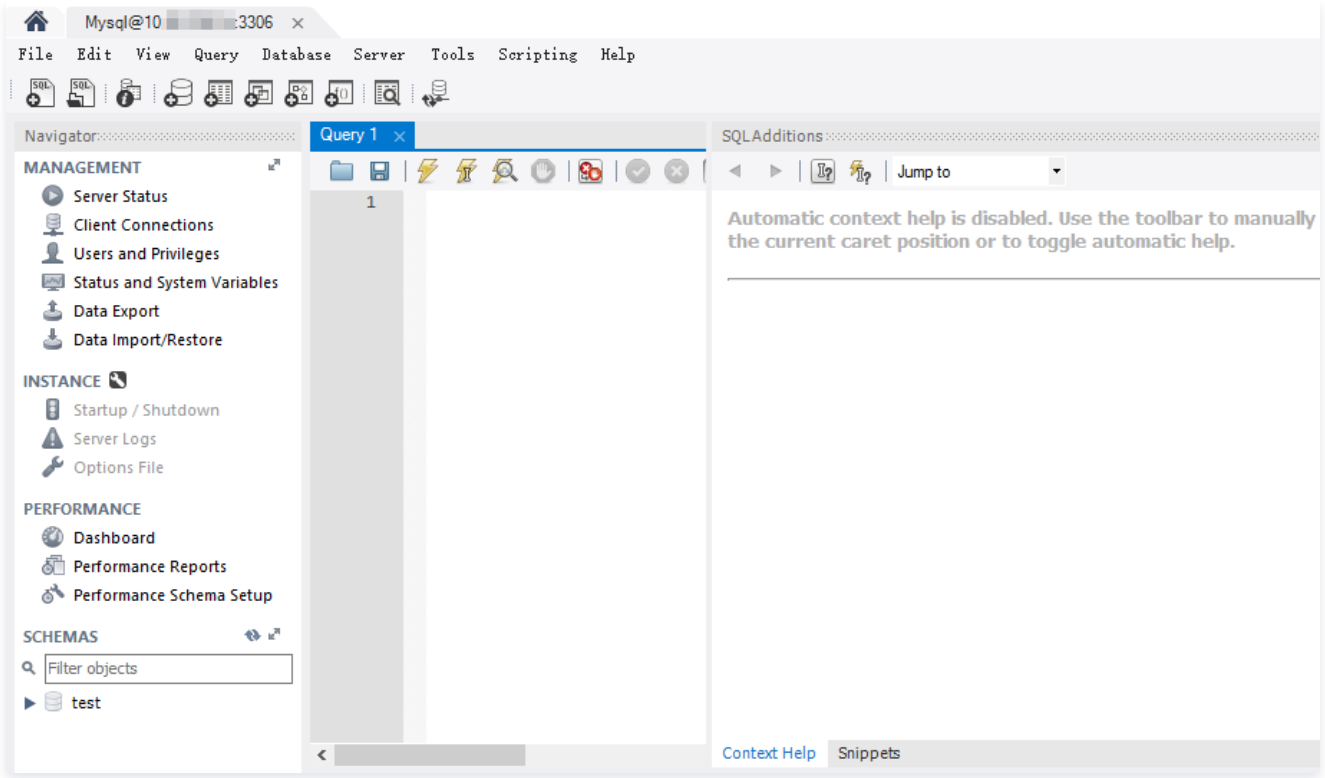


5. 打开 MySQL Workbench，选择 **Database** > **Connect to Database**，输入 TDSQL-C MySQL 数据库集群的内网（或外网）地址和用户名、密码，单击 **OK** 进行登录。



参数	说明
Hostname	输入 TDSQL-C MySQL 版集群下目标实例的内/外网地址，内/外网地址查询方法参见 步骤一 。若为外网地址，请确认是否已开启，如未开启，请参见 开启或关闭外网地址 。
Port	内网（或外网）对应端口。
Username	填写创建数据库账号时的账号名，此处以默认账号 root 为例。
Password	Username 对应的密码。如忘记密码可在控制台进行修改。

6. 登录成功的页面如图所示，在此页面上您可以看到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。



常见问题

内网和外网连接是否收费?

内网和外网连接免费使用，暂不收费。

如果忘记账号名以及密码，怎么重置后登录集群?

您可以在 TDSQL-C MySQL 版控制台，集群管理页面的**账号管理**下查看您创建的用于连接集群的账号名，当您忘记密码时，可通过**重置密码**操作修改密码后再登录集群。



通过内网或外网连接 TDSQL-C MySQL 版是否需要为集群配置安全组?

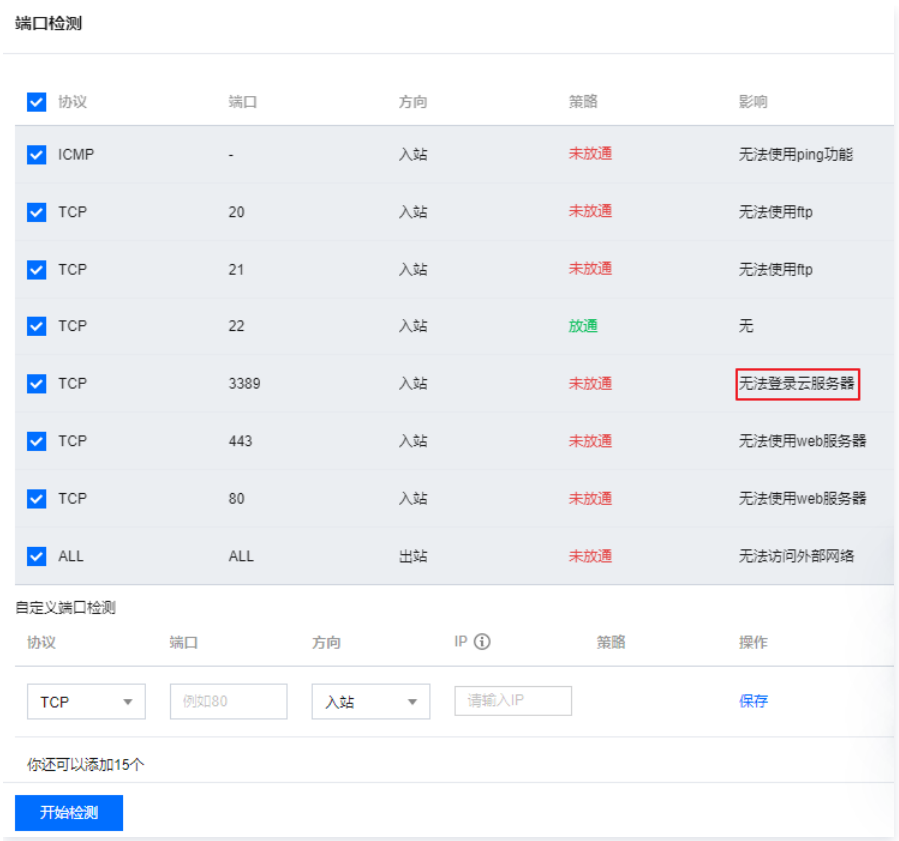
需要配置安全组，详细步骤请参见 配置安全组 。注意，若开启外网，通过外网连接时，配置的安全组规则需放通内网的访问端口。

如果云服务器实例的安全组规则匹配不当，导致无法登录 Linux 实例怎么办?

如果云服务器实例的安全组规则匹配不当，如未放通相应端口导致无法登录 Linux 实例，您可以通过 安全组（端口）验通工具 检查实例当前配置的安全组连通性。



通过一键检测可得知登录失败的可能原因。



然后在云服务器实例详情页，选择**安全组 > 编辑规则**，然后放通对应端口，详细操作请参见 添加安全组规则 。

无法连接集群还与哪些设置有关?

当无法连接集群时，您可以检查云服务器 CVM 和 TDSQL-C MySQL 是否在同一腾讯云账号下，是否处于同一地域，是否为同一 VPC 网络。需确保以上三点设置相同，才可以正常连接集群。更多连接相关故障问题处理请参见 故障处理 > 连接相关 。

云联网连接

最近更新时间：2024-07-02 17:05:31

本文为您介绍使用云联网连接数据库的方法。

背景

创建 TDSQL-C MySQL 版实例后，若云服务器和实例处于不同 VPC（包括同账号/不同账号，同地域/不同地域），可以通过云联网实现连接数据库；同一腾讯云账号下，若您购买的服务器为 轻量应用服务器，由于轻量应用服务器使用腾讯云自动分配的 私有网络 VPC 进行网络隔离，默认情况下内网不与处于私有网络 VPC 中的 TDSQL-C MySQL 版互通，因此，无法通过轻量应用服务器直接访问 TDSQL-C MySQL 版，需通过关联云联网实现连接数据库。

准备工作

- 准备好 TDSQL-C MySQL 版实例，请参见 创建集群（预置资源） 或者 创建 Serverless 版集群。
- 准备好云服务器，本文以轻量应用服务器为例，请参见 快速创建 Linux 轻量应用服务器。

注意事项

- 轻量应用服务器内网互联功能本身免费，您仅需关注云联网产品计费信息，详情请参见 云联网计费总览。
- 轻量应用服务器不支持通过关联云联网实现跨境内网互联，即使该云联网已购买跨境带宽。
- 同一账号下：
  - 同地域内的所有轻量应用服务器处于同一个 VPC 中，一个 VPC 只能同时关联一个云联网。
  - 不同地域内的轻量应用服务器处于不同的 VPC 中，不同 VPC 需要分别执行关联云联网操作。
  - 如果某地域中不存在轻量应用服务器实例，则用户无法在该地域执行关联云联网操作。

操作步骤

1. 登录 轻量应用服务器控制台，选择左侧导航栏中的内网互联。
2. 选择 TDSQL-C MySQL 版实例所在的地域，申请**关联云联网**。



3. 在弹出的**关联云联网**窗口中，选择**新建云联网**。



4. 在弹窗下完成云联网实例的创建，然后回到轻量应用服务器内网互联的关联云联网界面，可刷新出新建的云联网实例，然后单击**确定**。

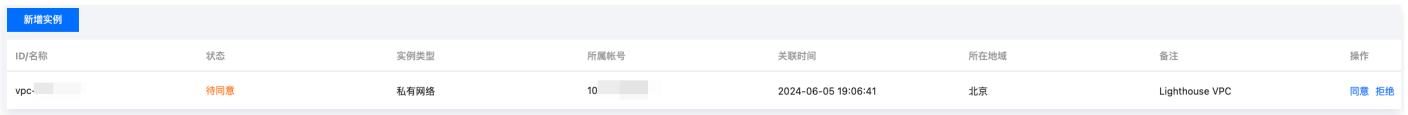


> **注意：**
> - 仅支持关联同账号下的云联网。
> - 提交关联申请后，请您在7天内登录 云联网控制台 同意申请。否则7天后申请过期，需进行重新关联申请。

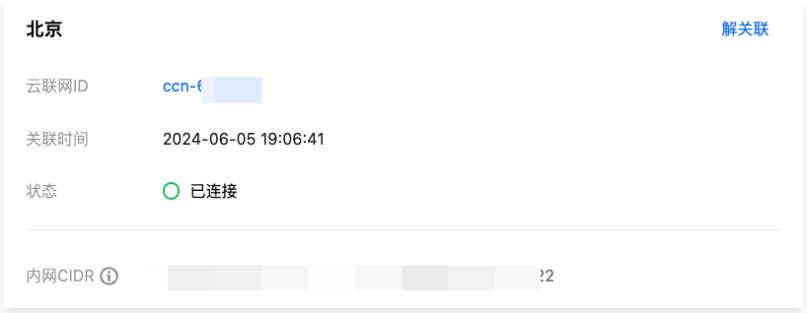
5. 登录 云联网控制台 ，单击目标云联网 ID 进入云联网详情页。



6. 在云联网详情页 > 关联实例下，找到对应的关联申请，单击操作列的**同意**，然后在弹窗下单击**确定**。
   轻量应用服务器的 VPC 实例默认会添加备注为“Lighthouse VPC”，请注意选择。



7. 单击新增实例，在弹窗下，添加目标 TDSQL-C MySQL 版的 VPC 网络地址。选择地域，VPC，单击**确定**。
8. 返回 轻量应用服务器控制台 的内网互联界面刷新，显示“已连接”。



9. 在 云联网控制台，单击目标云联网 ID 进入云联网详情页，查看路由是否有效，如果存在启用路由为关闭状态，请开启。



10. 在 轻量应用服务器控制台，找到目标轻量应用服务器，登录服务器。
11. 登录成功后，执行如下命令，下载安装 MySQL，提示“Complete”表示安装成功。

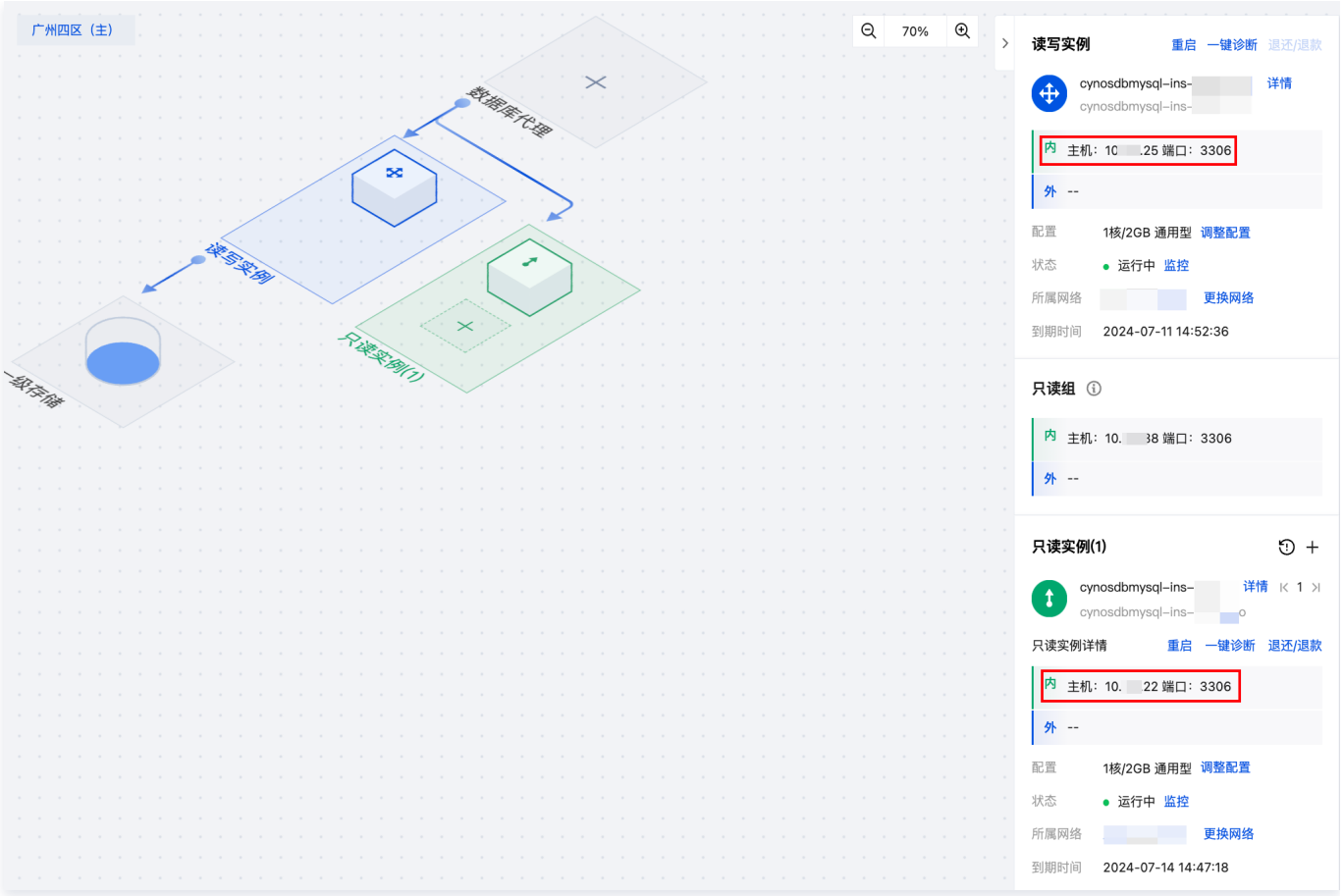
```
yum install mysql -y
```

```
[root@VM-20-12-opencloudos ~]# yum -y install mariadb-server
BaseOS 9 - x86_64                                                                    39 kB/s | 3.9 kB     00:00
BaseOS 9 - x86_64                                                                   1.1 MB/s | 824 kB     00:00
AppStream 9 - x86_64                                                                 34 kB/s | 3.8 kB     00:00
AppStream 9 - x86_64                                                                5.2 MB/s | 7.1 MB     00:01
extras 9 - x86_64                                                                    34 kB/s | 3.7 kB     00:00
Extra Packages for OpenCloudOS 9 - EPOL                                              29 kB/s | 3.0 kB     00:00
Extra Packages for OpenCloudOS 9 - EPOL                                             5.0 MB/s | 3.8 MB     00:00
Dependencies resolved.
==========================================================================================================================
 Package                    Architecture                  Version                    Repository                    Size
```

12. 执行如下命令，连接登录 TDSQL-C MySQL 版。

```
mysql -h <内网 IP 地址> -u <用户名，默认 root> -P <端口号，默认3306> -p
```

- <内网 IP 地址>：替换为目标数据库实例的内网地址，在 TDSQL-C MySQL 版控制台 的集群详情页可查看需要连接的实例的内网地址。



- <用户名，默认 root>：替换为目标数据库实例的账号名，默认的账号名是 root。
- <内网端口号>：替换为目标数据库实例的内网端口号，默认3306，如修改了端口号，可在 TDSQL-C MySQL 版控制台 的集群详情页查看新内网端口号。

13. 连接成功后，界面如下所示。

```
[root@VM-20-12-opencloudos ~]# mysql -h 10.0.0.39 -u test1 -P 3306 -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 5004
Server version: 8.0.22-cynos 20220401

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]>
```

常见问题

云联网连接是否收费?

使用云联网会产生费用，具体请参见 云联网计费说明 。

什么情况下使用云联网连接 TDSQL-C MySQL 版?

1. 云服务器和 TDSQL-C MySQL 版实例处于不同 VPC（包括同账号/不同账号，同地域/不同地域），可以通过 云联网 实现连接数据库。
2. 若您购买的服务器为 轻量应用服务器，则需要通过 云联网 实现连接数据库。

轻量应用服务器如何实现内网互联?

轻量应用服务器使用腾讯云自动分配的私有网络 VPC 进行网络隔离，默认情况下内网不与云服务器 CVM、云数据库等其他处于私有网络 VPC 中的腾讯云资源内网互通，需通过关联云联网实现。详情请参见 内网互联 。

> **说明：**
> 轻量应用服务器仅支持关联同账号下的云联网，跨账号轻量应用服务器无法互通。

云联网的服务等级是什么？

腾讯云云联网服务提供三种可用性等级：白金、金、银。

- 同地域的服务等级均为金，且不可修改。
- 跨地域互通时，提供白金、金、银三个服务等级，用户创建云联网时可选择服务等级。
  不同的服务等级费用不同，您可以根据业务需要进行选择，详细内容请参见 计费说明 。

云联网连接失败如何分析和处理？

请参见 云联网连接失败 。

网络地址配置
开启或关闭外网地址

最近更新时间：2025-02-28 16:26:12

除了通过内网地址连接集群，在开启外网后，您也可以在外网通过系统分配的域名和端口访问 TDSQL-C MySQL 版，生效时间大概需要5分钟，TDSQL-C MySQL 版外网访问仅用于开发或辅助管理数据库，业务方面的访问请使用内网访问。
本文为您介绍如何通过控制台开启或关闭集群下的读写外网地址和只读外网地址。

说明：
- 外网访问出现的故障，不会计入 TDSQL-C MySQL 版服务的整体可用性计算。
- TDSQL-C MySQL 版外网访问会降低实例的安全性，仅推荐用于管理、测试或辅助管理数据库，不提供可用性 SLA 保证，业务访问请使用内网访问。
- TDSQL-C MySQL 版外网访问开通后将受到 安全组 网络访问策略的控制，请提前配置好相应策略。
- 目前外网功能暂时免费提供，不保证外网带宽和流量稳定。
- 仅当您的集群下有只读实例时，在集群详情页连接信息处才会显示只读外网地址的开启按钮。
- 建议您使用外网域名而不是 IP 进行访问，因为调整数据库实例规格、重新开通外网、网络升级等都可能导致外网 IP 发生变化，使用外网域名访问能够保证对业务影响最小，不用修改应用程序。

操作场景

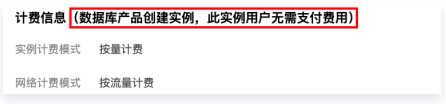
TDSQL-C MySQL 版集群下包含读写实例和只读实例，均支持内网和外网两种地址类型，默认提供内网地址供您内部访问实例，如果需要通过外网地址访问，您需要先开启外网地址，开启后也可根据需要关闭，外网地址为系统自动分配，暂不支持自定义修改。

外网升级新架构的说明

一、新架构发布时间

为了提升数据库外网链路的安全及可靠性，**自北京时间2024年5月起**，TDSQL-C MySQL 版发布了新的外网架构，其采用了 CLB 作为外网底层架构。

二、新老架构差异对比

差异项	老外网架构	新外网架构（CLB）
架构差异	老外网架构为单点部署架构，当出现单点故障时恢复较慢且不具备高可用能力。	新外网架构可以通过流量分发扩展应用系统对外的服务能力，通过消除单点故障提升应用系统的可用性。
是否涉及产品联动	不涉及	涉及，开启外网后，系统会在 负载均衡 CLB 控制台 上自动创建一个同地域的免费的简约型负载均衡实例，用于提供公网能力。 

三、注意事项

- **目前 TDSQL-C MySQL 版开启外网后为 CLB 架构，系统会在 负载均衡 CLB 控制台 上自动创建一个同地域的免费的简约型负载均衡实例**，用于提供公网能力，需要注意 CLB 架构的策略（如下表），如有更高性能要求，您也可以直接购买 CLB 来实现。

分类	并发连接数	新建连接数	包量	入带宽	出带宽
CLB	2000	200/s	不限制	20Mbps	20Mbps



> **说明：**
> - 因开启外网地址而自动创建的负载均衡实例，您可以免费使用。
> - 关闭外网地址后，在负载均衡控制台上对应的负载均衡实例会自动删除。

- TDSQL-C MySQL 版开启外网的功能免费，但开启流程会校验账户的费用情况，如果账户欠费（此处含义为账户余额小于0），将无法开启。如需开启外网能力，请确认您的账户不处于欠费状态。开启成功后，因开启外网而自动创建的负载均衡实例不会产生费用。
- 2024年5月中旬开始，CLB 健康探测源 IP 为100.64.0.0/10网段，外网开启后，若您的简约型负载均衡实例健康状态显示为异常，您可在对 TDSQL-C MySQL 版配置安全组时，**放通100.64.0.0/10网段**，以解决健康检查失败导致负载均衡实例健康状态显示为异常的问题，操作请参考 修改或添加安全组规则 。



- 您需要对上述简约型负载均衡实例配置监控告警，以通过开启外网地址后的指标情况（公网新建连接数、公网连接数、公网出带宽、公网入带宽等）来监测公网连接，操作步骤请参考 设置告警 ，策略类型选择如下图所示。



支持地域及实例形态

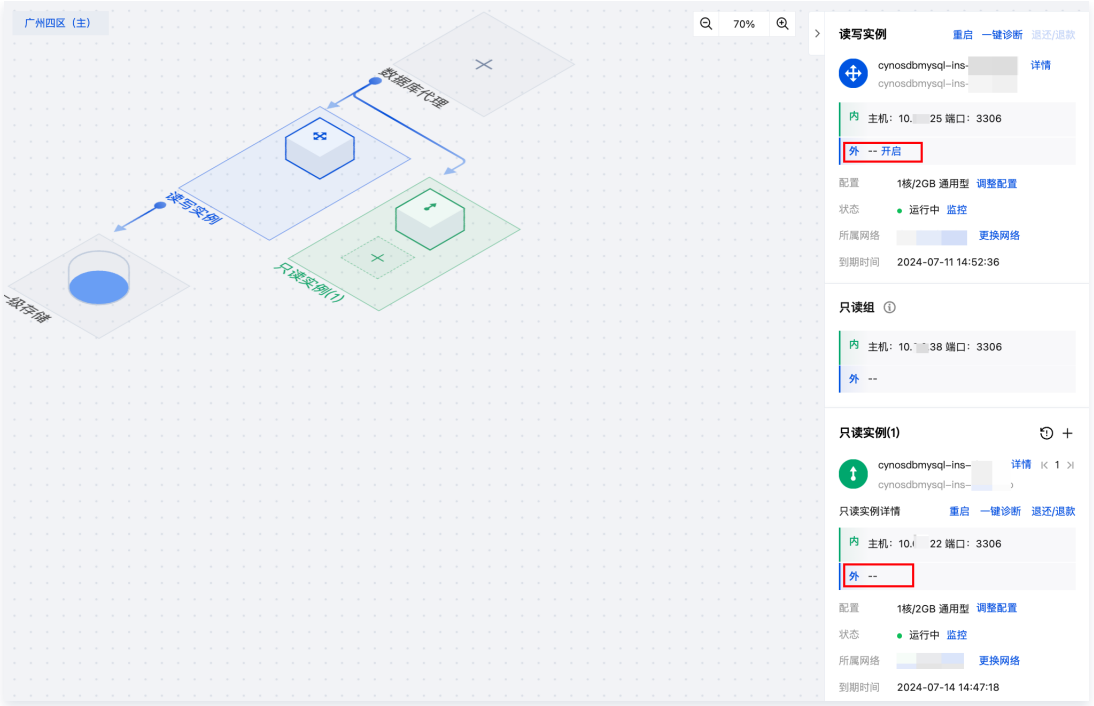
- 广州、上海、北京、成都、重庆、南京、中国香港、新加坡、首尔、东京、硅谷、法兰克福、弗吉尼亚地域的集群，支持开启外网地址。部分地域陆续开放中，已开放地域请以控制台实际支持为准。
- 实例形态为预置资源和 Serverless 的集群，均支持为集群内的读写实例或只读实例开启外网地址，开启后，您可通过外网地址访问对应实例。

开启集群读写/只读外网地址

在集群的管理页面，根据实际使用的视图模式进行操作：

页签视图

1. 登录 TDSQL-C MySQL 版控制台 ，在左侧集群列表，单击**目标集群**，进入集群管理页。
2. 在集群详情下，找到需要开启外网的实例，鼠标移动到读写/只读地址的外网处，会显示开启字样，单击**开启**。



3. 单击开启后，系统会根据实例绑定安全组的情况分如下几种操作情形。

情形一：当前安全组配置未放通端口

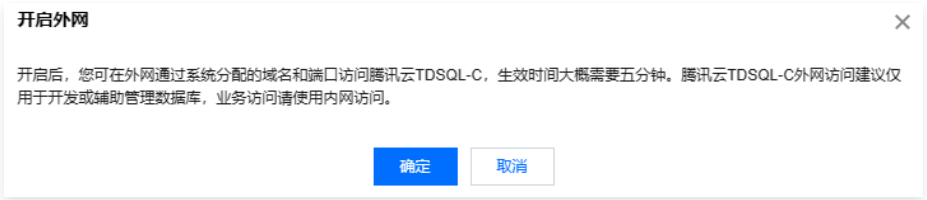
如需开启外网，您需要单击**授权并创建**（系统将申请您的授权自动为本集群绑定已放通内网端口的安全组（当前端口：3306）方便您进行外网连接，后续您可进入安全组界面进行后续设置或换绑新的安全组）。

> **注意：**
> - 开通外网需在安全组放通内网策略，方可进行访问，检测到您当前安全组配置未放通端口，为保证连接成功， 系统将自动为本集群放通端口策略并允许0.0.0.0/0和::/0访问，为了您的网络安全，建议后续手动修改为固定 IP 访问。
> - 启用后，您可以在外网通过系统分配的域名和端口访问 TDSQL-C MySQL 版，生效时间大概需要5分钟。
> - 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。



情形二：当前安全组配置已放通端口

在弹窗下，阅读提示并单击**确定**即可开启外网。



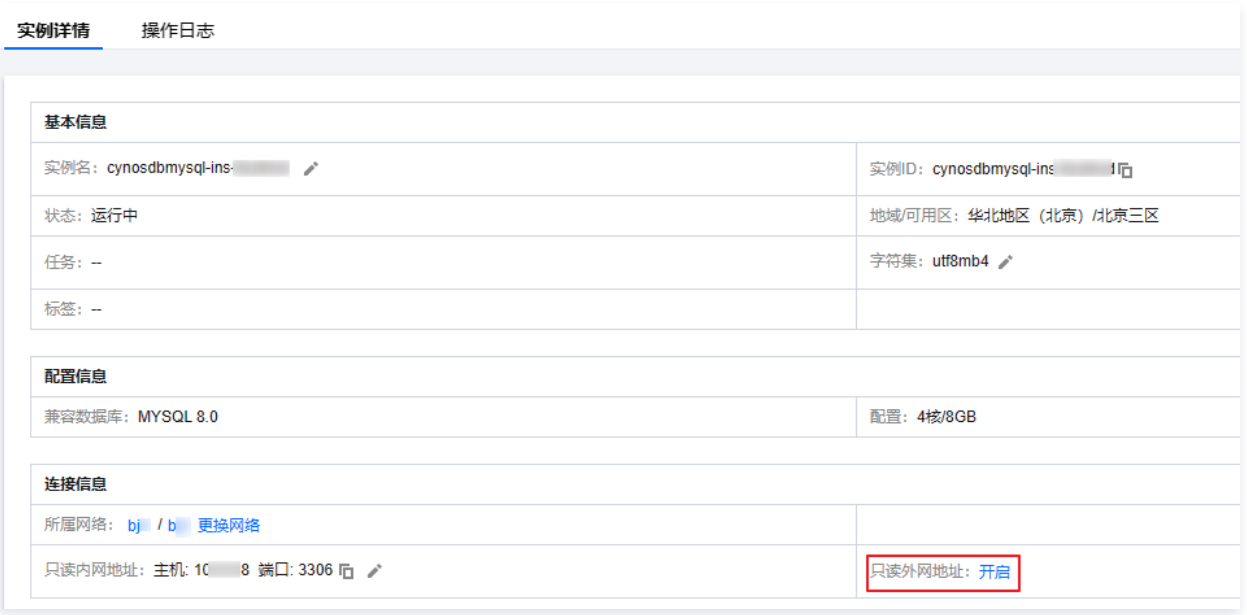
4. 成功开启后，您可在对应实例外网地址处查看外网的主机和端口，外网主机和端口不支持修改。

列表视图

1. 登录 TDSQL-C MySQL 版控制台，在集群列表中，找到需要的集群，单击**集群 ID**，进入集群管理页面。
2. 在**集群管理**页 > **实例列表**页，选择需要开启外网的实例（读写实例或只读实例），单击其**实例 ID** 或**操作**列的**管理**进入实例详情页。



3. 在**实例详情**页 > **连接信息** > **外网地址**后单击**开启**。



4. 单击开启后，系统会根据实例绑定安全组的情况分如下几种操作情形。

情形一：当前安全组配置未放通端口

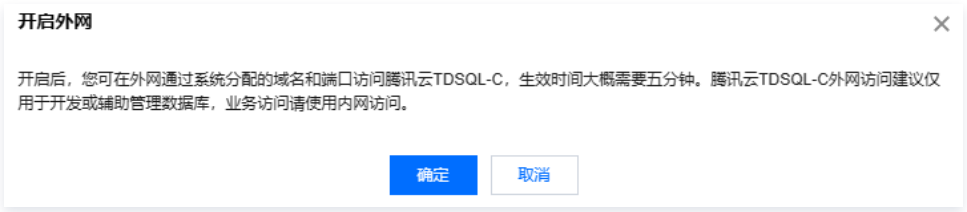
如需开启外网，您需要单击**授权并创建**（系统将申请您的授权自动为本集群绑定已放通内网端口的安全组（当前端口：3306）方便您进行外网连接，后续您可进入安全组界面进行后续设置或换绑新的安全组）。

> **注意：**
> - 开通外网需在安全组放通内网策略，方可进行访问，检测到您当前安全组配置未放通端口，为保证连接成功， 系统将自动为本集群放通端口策略并允许0.0.0.0/0和::/0访问，为了您的网络安全，建议后续手动修改为固定IP访问。
> - 启用后，您可以在外网通过系统分配的域名和端口访问 TDSQL-C MySQL 版，生效时间大概需要5分钟。
> - 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。



情形二：当前安全组配置已放通端口

在弹窗下，单击阅读提示并单击**确定**即可开启外网。



5. 成功开启后，您可连接信息下的外网地址处查看外网的主机和端口，外网主机和端口不支持修改。

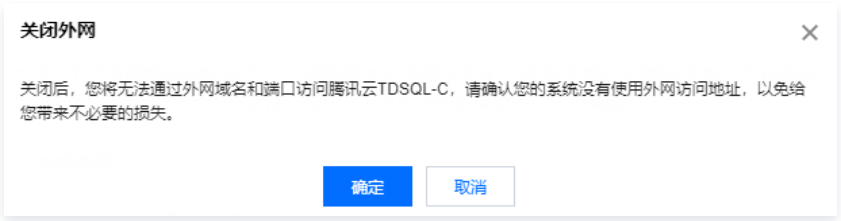
关闭集群读写/只读外网地址

> **注意：**
> 关闭外网地址后，您将无法通过外网域名和端口访问 TDSQL-C MySQL 集群，请确认您的系统没有使用外网访问地址，以免给您带来不必要的损失。

在集群的管理页面，根据实际使用的视图模式进行操作：

页签视图

1. 登录 TDSQL-C MySQL 版控制台，在左侧集群列表，单击**目标集群**，进入集群管理页。
2. 在集群详情下，找到需要关闭外网的实例，在外网地址后单击**关闭**。
3. 在弹出的对话框，确认关闭外网的注意事项后，单击**确定**。



列表视图

1. 登录 TDSQL-C MySQL 版控制台，在集群列表中，找到需要的集群，单击**集群 ID**，进入集群管理页面。
2. 在**集群管理**页 > **实例列表**页，选择需要关闭外网的实例（读写实例或只读实例），单击其**实例 ID** 或操作列的**管理**进入实例详情页。



3. 在**实例详情**页 > **连接信息** > **外网地址**后单击**关闭**。
4. 在弹出的对话框，确认关闭外网的注意事项后，单击**确定**。



通过 CLB 开启外网服务

最近更新时间：2025-04-27 17:58:22

TDSQL-C MySQL 版支持内网和外网两种地址类型，默认提供内网地址供您内部访问实例，如果需要使用外网访问，除了开启外网地址后，通过 Linux 或者 Windows 云服务器连接访问实例，也可通过负载均衡 CLB 开启外网服务进行访问，通过 CLB 开启外网服务必须配置安全组规则。
以下为您介绍通过 CLB 开启外网服务，并通过 MySQL workbench 连接到实例的方法。

前提条件

已申请使用后端服务功能。

1. 进入 负载均衡跨地域绑定2.0申请页 。
2. 根据需要填好资料，填写完后提交申请。
3. 提交完内测申请后，提单至 CLB ，申请使用后端服务功能。

步骤1：新购负载均衡

> **说明：**
> 如果在 TDSQL-C MySQL 版同地域已经有负载均衡实例，就可以不用购买。

进入 负载均衡购买页，选择完配置后单击**立即购买**。详细购买步骤可参考 购买方式 。

> **注意：**
> - 地域需选择 TDSQL-C MySQL 版所在的地域。
> - 仅公网 CLB 实例支持后端服务功能，因此，购买页面的网络类型选择为公网。
> - 通过 CLB 开启外网服务的场景，仅适用于负载均衡 CLB 实例与 TDSQL-C MySQL 版实例属于同一 VPC 网络，非同一 VPC 网络暂不支持。

步骤2：配置负载均衡

1. 启用后端服务功能（启用后 CLB 支持绑定其他内网 IP ）。
   1.1 登录 负载均衡控制台，选择地域，在实例管理列表，单击**实例 ID**，进入实例管理页面。
   1.2 在**基本信息**页的**后端服务**处，单击**点击配置**。



   1.3 在弹出的对话框，单击**提交**即可开启。



2. 配置外网监听端口。
   2.1 登录 负载均衡控制台，选择地域，在实例管理列表，单击**实例 ID**，进入实例管理页面。
   2.2 在实例管理页面，选择**监听器管理**页，在 **TCP/UDP/TCP SSL/QUIC 监听器**下方，单击**新建**。



2.3 在弹出的对话框，逐步完成设置，然后单击**提交**即可完成创建。



步骤3：绑定 TDSQL-C MySQL 版实例内网 IP

1. 创建好监听器后，在**监听器管理**页，单击创建好的监听器，然后单击右侧的**绑定**。



2. 在弹出的对话框，选择目标类型为 **IP 类型**，单击**添加内网 IP**，输入待连接 TDSQL-C MySQL 版实例的内网 IP 地址和端口，单击**确认**完成绑定。

> **注意：**
> 登录的账号必须是标准账号（带宽上移），如无法绑定，请 提交工单 协助处理。



步骤4：配置安全组

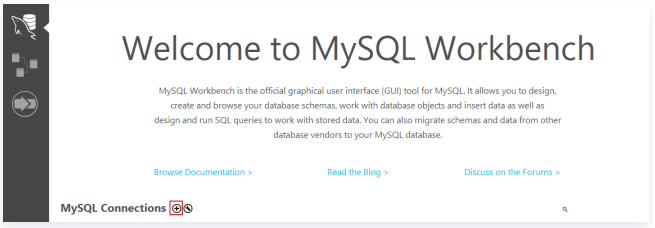
1. 登录 TDSQL-C MySQL 版控制台，在左侧集群列表，单击**目标集群**，进入集群管理页。
2. 在集群管理页面，选择**安全组**页，然后选择集群下的目标连接对象，例如：要连接集群下的读写实例，请选择读写实例，要连接集群下的只读实例，请选择只读实例。



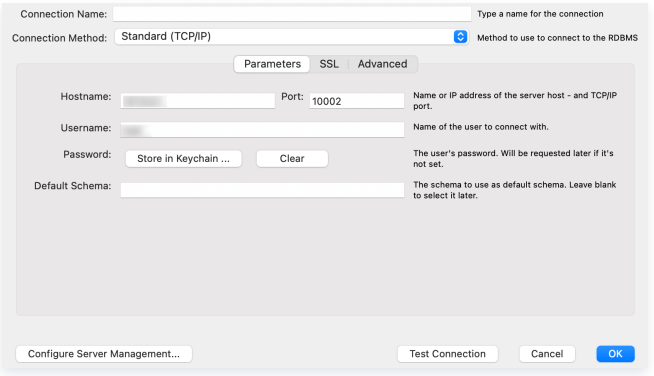
3. 单击**配置安全组**，配置安全组规则为放通全部端口，确认安全组允许外部 IP 访问，详细配置方法请参见 配置安全组 。

步骤5：通过 MySQL Workbench 客户端连接实例

1. 进入 MySQL Workbench 官方下载页面。
2. 根据您的系统来选择适配版本的安装程序，单击 **Download**。
3. 在跳转页面中选择 **No thanks, just start my download.** 以快速下载。
4. 下载完成后，打开下载的安装程序，安装 MySQL Workbench。
5. 安装完成后打开 MySQL Workbench，在 MySQL Connections 后单击加号添加待连接的实例信息。



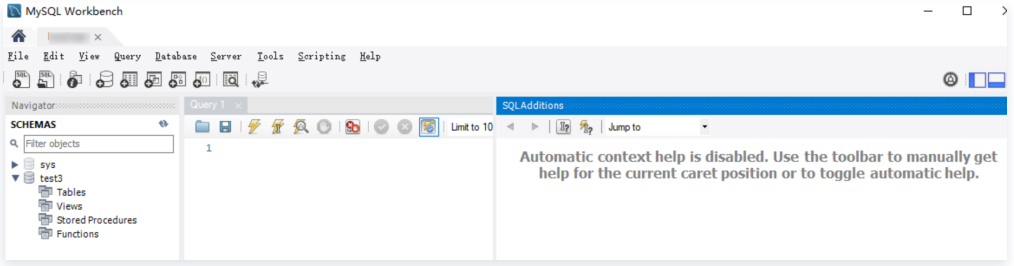
6. 在弹出的窗口下，完成如下配置后，单击 **OK**。



参数	说明
Connection name	为此连接命名。

Connection Method	连接方法，选择 Standard（TCP/IP）。
Hostname	输入负载均衡 CLB 实例的地址。在 CLB 实例详情页的基本信息下可查询 VIP 信息。
Port	输入负载均衡 CLB 实例的端口。在 **CLB 实例详情**页 > **监控器管理**下可查询 TCP 端口号。
Username	输入待连接 TDSQL-C MySQL 版实例的账号名。在**集群管理**页 > **账号管理**下创建的账号。
Store in Keychain	单击 Store in Keychain，在弹窗下输入所填写的 Username 对应的密码，然后单击 **OK**。

7. 返回 MySQL Workbench 首页，单击刚创建的待连接实例信息连接到 TDSQL-C MySQL 版实例。
8. 成功连接后的界面如下所示。



修改内网地址

最近更新时间：2025-01-14 17:27:22

本文为您介绍如何通过控制台修改 TDSQL-C MySQL 版的读写/只读内网地址。

操作指南

在集群的管理页面，根据实际使用的视图模式进行操作：

页签视图

1. 登录 TDSQL-C MySQL 版控制台，在左侧集群列表，单击**目标集群**，进入集群管理页。
2. 在集群详情下找到目标实例，鼠标移动到内网地址后，单击编辑图标。



3. 在弹出的对话框，修改内网地址端口后，单击**确定**。

> **注意：**
> 修改内网地址会影响正在访问的数据库业务。



- **内网地址**：您可在内网地址可选段范围自定义内网地址。
- **自定义端口**：您可自定义端口值，取值范围为1024 - 65535。
- **旧 IP 地址回收时间**：回收时间系统默认为0小时，表示会立即回收旧 IP 地址，随即不可通过旧 IP 地址访问实例。此时间支持自定义，范围是：0 - 168小时。

列表视图

1. 登录 TDSQL-C MySQL 版控制台，在集群列表中，找到需要的集群，单击**集群 ID**，进入集群管理页面。
2. 选择实例列表页，选择需修改内网地址的实例，单击其**实例 ID** 进入实例详情页。
3. 在**实例详情页** > **连接信息** > **内网地址**后单击编辑图标。

版权所有：腾讯云计算（北京）有限责任公司



4. 在弹出的对话框，修改内网地址端口后，单击**确定**。

> **注意：**
> 修改内网地址会影响正在访问的数据库业务。



- **内网地址**：您可在内网地址可选段范围自定义内网地址。
- **自定义端口**：您可自定义端口值，取值范围为1024 - 65535。
- **旧 IP 地址回收时间**：回收时间系统默认为0小时，表示会立即回收旧 IP 地址，随即不可通过旧 IP 地址访问实例。此时间支持自定义，范围是：0 - 168小时。

更换网络

最近更新时间：2024-07-02 17:05:31

您可根据业务需要对 TDSQL-C MySQL 版集群下读写实例或任一只读实例的网络进行更换，本文为您介绍通过控制台更换网络。

操作场景

TDSQL-C MySQL 版支持设置私有网络 VPC，为您带来优质的服务。在此基础之上，我们支持更改集群下读写实例或任一只读实例的所属网络，方便您进行网络管理。

注意事项

- 更换网络会导致集群下所有内网 IP 发生变化，系统将自动分配新的 IP 地址，请及时修改客户端程序。
- 旧的 IP 地址默认24小时后会失效，失效时间可在控制台修改。当回收时间设置为0，更换网络后旧 IP 地址会被立即回收。
- 只能选择集群所在地域下的 VPC 网络与子网。

子网说明

子网是私有网络的一个网络空间，为逻辑概念，同一私有网络下可以创建不同可用区的子网，同一私有网络下不同可用区的子网默认可以内网互通，即使您选择集群所在地域下其他可用区的子网 IP，实际业务连接访问为就近逻辑，不会增加网络时延。

操作步骤

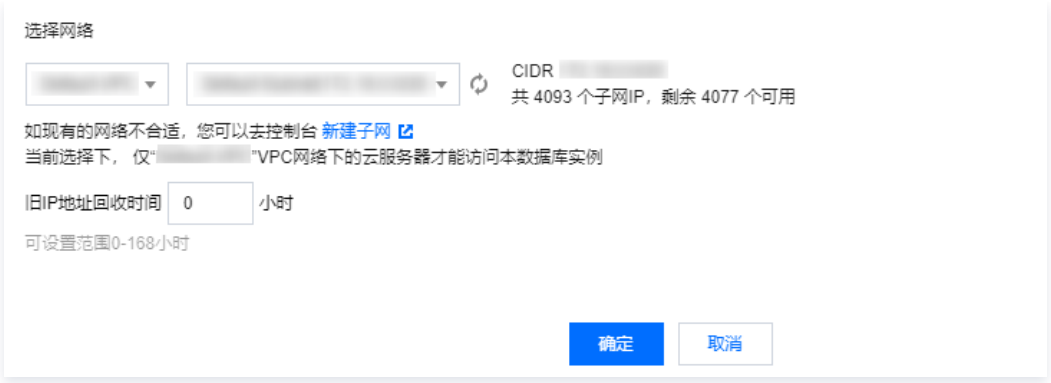
在集群的管理页面，根据实际使用的视图模式进行操作：

页签视图

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在左侧集群列表，单击目标集群，进入集群管理页。
3. 在集群详情下找到目标实例，在其所属网络后单击**更换网络**。



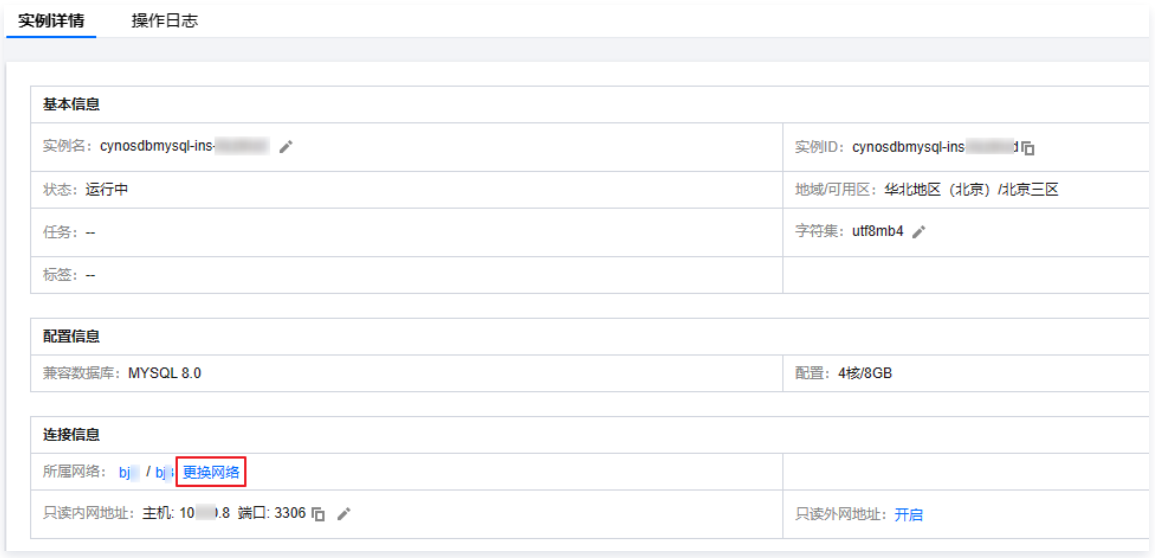
4. 在更换网络窗口，选择网络，设置旧 IP 地址回收时间，单击**确定**。



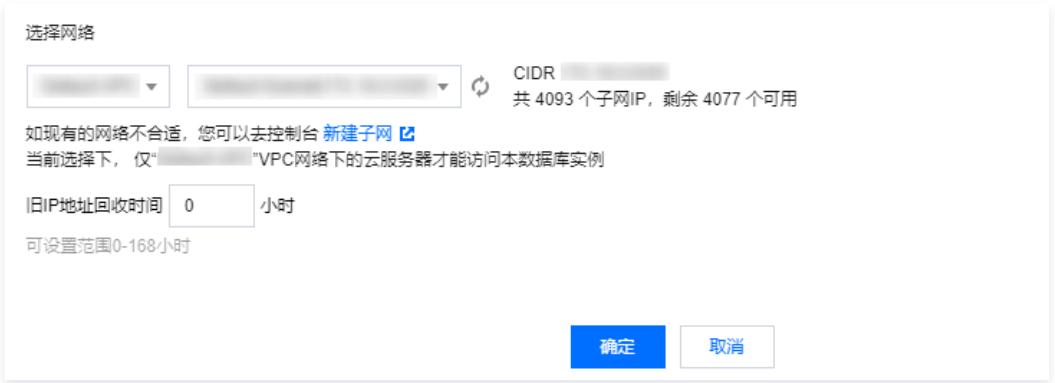
5. 更改网络完成后，您可在集群详情下对应实例的**所属网络**查看新网络，在**读写/只读地址**查看新内网地址。

列表视图

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在上方选择地域，在集群列表单击集群 ID 或**操作**列的**管理**，进入集群管理页面。
3. 单击实例列表，选择目标实例（读写或只读实例），单击其实例 ID 进入实例详情页。
4. 在**实例详情**页的**连接信息** > **所属网络**后，单击**更换网络**。



5. 在更换网络窗口，选择网络，设置旧 IP 地址回收时间，单击**确定**。



6. 更改网络完成后，您可在对应实例详情页的**所属网络**查看新网络，在**读写/只读内网地址**查看新内网地址。

实例管理

实例管理简介

最近更新时间：2025-01-15 10:06:42

本文为您介绍什么是实例，如何查看实例列表以及您可对实例进行的操作。

实例介绍

实例是腾讯云上的数据库资源，TDSQL-C MySQL 版最基本的管理单元为集群，一个集群中可以包含多个实例，其中有读写实例和只读实例。
集群下的实例类型和支持创建的数量等说明如下：

集群下包含的实例类型	可创建数量	支持 IP
读写实例	1	默认开启读写内网地址，可手动开启读写外网地址
只读实例	15	创建后默认开启只读内网地址，可手动开启只读外网地址

访问地址

TDSQL-C MySQL 版支持读写访问地址与只读访问地址，读写访问地址对应 TDSQL-C MySQL 版的读写实例，可以提供写入与读取服务。只读访问地址后端对应一个或多个只读实例，可实现只读实例后端的负载均衡。

实例管理概览

您可通过控制台对集群下的实例进行管理设置，具体可变更项和说明如下表所示：

实例管理项	说明	操作方法
实例名	a. 您可对集群下的读写/只读实例进行名称修改 b. 实例名仅支持长度小于60的中文、英文、数字、`-`、`_`、`.`	设置实例名称
字符集	a. 您可修改对应实例的字符集 b. 目前支持的字符集有：UTF8、GBK、LATIN1、UTF8MB4	修改字符集
调整配置	a. 您可根据业务需要变更实例配置 b. 可对实例的计算规格与存储进行调整	调整计算配置 调整存储空间
维护时间	a. 您可设置实例的维护时间和周期，便于为相关操作预设维护信息 b. 维护周期可连续或间断选择从周一到周日 c. 维护时间可设置开始时间和持续时间	设置实例维护时间
重启实例	a. 当您需要对实例进行例行维护，或者清理数据库的 buffer 时，可以重启实例 b. 为避免影响您的业务正常运行，请确保该实例目前无其他任务项	重启实例
删除实例	请确定此实例您不再使用，再执行删除	删除实例
恢复实例	a. 您可在误删或其他需要恢复的情况下恢复已隔离状态的实例 b. 回收站会显示该实例可恢复的期限	恢复实例

查看实例列表

在集群的管理页面，根据实际使用的视图模式进行操作：

页签视图

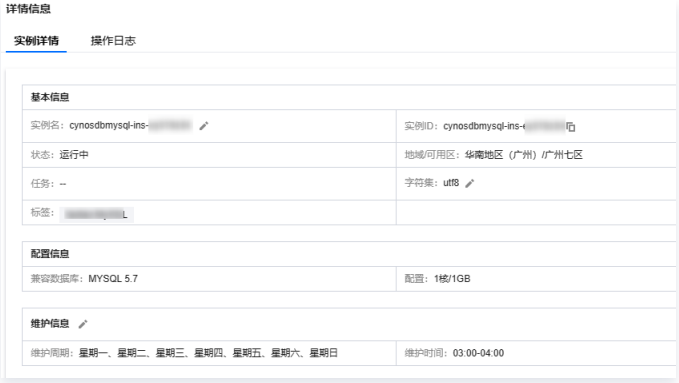
1. 登录 TDSQL-C MySQL 版控制台，在左侧集群列表，单击目标集群，进入集群管理页。
2. 在集群详情下，可以查看此集群下拥有的所有读写实例和只读实例。



3. 如果集群下拥有的只读实例较多，您可单击只读实例下的图标切换查看。



4. 您可查看到读写/只读实例的信息为：实例 ID、实例名称、实例运行状态、读写/只读内网或外网地址，可对实例进行的操作为：重启实例、实例名称修改、内网地址修改、外网地址开启或关闭、重启实例、调整配置、销毁实例等，单击**详情**，可进入对应实例详情页面。



5. 在实例详情页，您可查看实例的基本信息、配置信息、维护信息，也可以查看实例的操作日志。

列表视图

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在上方选择地域，在集群列表中，找到目标集群，单击集群 ID 或**操作**列的**管理**进入集群管理页面。
3. 在集群管理页面，选择实例列表页，即可查看到当前集群下的读写实例和只读实例。
   - 实例列表中默认展示**实例ID/实例名、监控、可用区、内外网地址、到期时间、实例类型、实例状态、实例配置、操作**等。
   - 若集群下实例较多，也可通过右侧搜索对话框中输入实例 ID 或者实例名的关键字，快速找到对应的实例。
   - 单击**实例 ID** 可进入实例详情页进行实例信息及操作日志查询。


集群详情
实例列表
监控告警
账号管理
数据库管理
数据库代理
数据安全
参数设置
安全组
备份管理
操作日志
读写实例
只读实例
只读分析引擎
请输入实例名、实例
实例 ID / 实例名
监控
可用区
内/外网地址
到期时间
实例类型
实例状态
实例配置
操作
cynosdbmysql-ins
cynosdbmysql-ins
北京三区
(内) 10 5:3306
(外) 未开启
2025-02-06 14:30:54
读写实例
运行中
独享型
2核/4GB
管理 重启 更多

修改实例名

最近更新时间：2025-01-15 10:06:42

实例名主要用于对 TDSQL-C MySQL 版集群下的读写或只读实例进行区分和管理，默认实例名和实例 ID 保持一致，可以通过控制台对实例名称进行修改，实例 ID 默认不支持修改。

操作步骤

在集群列表页面，根据实际使用的视图模式进行操作：

页签视图

在集群详情页修改实例名

1. 登录 TDSQL-C MySQL 版控制台，在左侧集群列表，单击目标集群，进入集群管理页。
2. 在集群详情下，找到目标实例，鼠标移动到实例名称位置，会显示编辑图标，单击实例名称后的编辑图标。

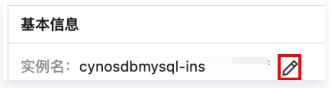


3. 在弹出的对话框，输入要修改的实例名，单击**确定**。

> **说明：**
> 实例名仅支持长度小于60的中文，英文，数字，-(中横线)，_(下划线)，.(英文标点)。

在实例详情页修改实例名

如果您当前的操作场景是在集群下的某一个实例的详情页面，也可以直接在此页面修改实例名称，在基本信息下，单击实例名后的修改按钮，改好后单击**确定**。



列表视图

在实例列表中修改实例名

1. 登录 控制台，在集群列表，找到需要的集群，单击集群 ID，进入集群管理页面。
2. 在集群管理页面，选择**实例列表**页，找到要修改的实例，单击实例名后的修改按钮。



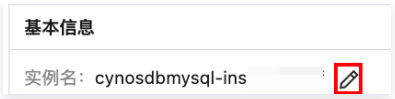
3. 在弹出的对话框，输入要修改的实例名，单击**确定**。

> **说明：**
> 实例名仅支持长度小于60的中文，英文，数字，-(中横线)，_(下划线)，.(英文标点)。

在实例详情中修改实例名

1. 登录 控制台，在集群列表，找到需要的集群，单击集群 ID，进入集群管理页面。
2. 在集群管理页面，选择**实例列表**页，找到要修改的实例，单击实例 ID，进入实例详情页。

3. 在实例详情页**基本信息**处，单击实例名后的修改按钮。



4. 在弹出的对话框，输入要修改的实例名，单击**确定**。

> **说明：**
> 实例名仅支持长度小于60的中文，英文，数字，-(中横线)，_(下划线)，.(英文标点)。

修改字符集

最近更新时间：2024-07-02 17:05:31

字符集包括字符编码集和字符编码，可以简单理解为一个映射关系，一个编码规则。将字符集对应的码点映射为一个个二进制序列，从而使得计算机可以存储和处理。
本文为您介绍如何通过控制台修改实例的字符集。

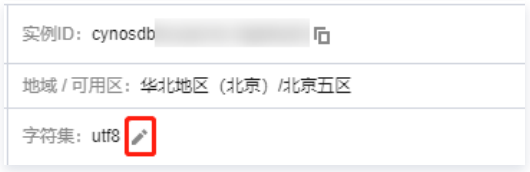
> **说明：**
> 修改实例字符集会引起实例重启，建议在实例维护时间内修改。

操作步骤

在集群列表页，根据实际使用的视图模式进行操作：

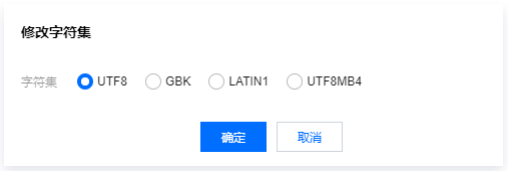
页签视图

1. 登录 TDSQL-C MySQL 版控制台，在左侧集群列表，单击目标集群，进入集群管理页。
2. 在集群详情下，找到目标实例，单击实例 ID 后的**详情**，进入实例详情页。
3. 在实例详情页的**字符集**处，单击修改图标。



4. 在弹出的对话框，设置字符集，单击**确定**。

> **说明：**
> 支持 UTF8、GBK、LATIN1、UTF8MB4 字符集，字符集说明请参见 MySQL 官方文档 。

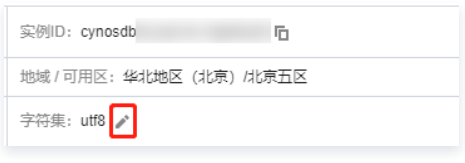


列表视图

1. 登录 TDSQL-C MySQL 版控制台，在集群列表，找到需要修改字符集的集群，单击**集群 ID**，进入集群管理页面。
2. 在集群管理页面，选择**实例列表**页，找到要修改的读写实例或只读实例，单击**实例 ID**，进入实例详情页。

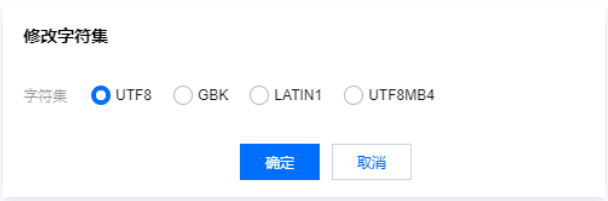


3. 在实例详情页的**字符集**处，单击修改图标。



4. 在弹出的对话框，设置字符集，单击**确定**。

> **说明：**
> 支持 UTF8、GBK、LATIN1、UTF8MB4 字符集，字符集说明请参见 MySQL 官方文档 。



修改实例维护时间

最近更新时间：2025-01-15 10:06:42

当实例在进行配置变更或者其他对实例会产生闪断或者影响的操作时，可指定实例的维护时间。本文为您介绍如何修改实例的维护时间。

操作步骤

在集群列表，根据实际使用的视图模式进行操作：

页签视图

1. 登录 TDSQL-C MySQL 版控制台，在左侧集群列表，单击目标集群，进入集群管理页。
2. 在集群详情下，找到目标实例，单击**实例 ID** 后的**详情**，进入实例详情页。
3. 在实例详情页维护信息后，单击修改图标。



4. 在弹出的对话框，修改维护周期和时间，单击**确定**。
   - **维护周期**代表每周哪一天可以为维护时间，每周必须包含一个维护时间。
   - **维护时间**代表每天的时间窗口，可指定每天的任意一个时刻，持续时间支持1、1.5、2、2.5这4种维护时长。



列表视图

1. 登录 TDSQL-C MySQL 版控制台，在集群列表中，找到需要的集群，单击**集群 ID**，进入集群管理页面。
2. 在集群管理页面，选择**实例列表**页，找到想要修改的实例，单击**实例 ID**，进入实例详情页。
3. 在实例详情页的**维护信息**处，单击修改图标。



4. 在弹出的对话框，修改维护周期和时间，单击**确定**。
   - **维护周期**代表每周哪一天可以为维护时间，每周必须包含一个维护时间。
   - **维护时间**代表每天的时间窗口，可指定每天的任意一个时刻，持续时间支持1、1.5、2、2.5这4种维护时长。



重启实例

最近更新时间：2025-01-15 10:06:42

当需要对实例进行例行维护，清理数据库的 buffer，或出现性能问题时，可以手动重启实例。

版权所有：腾讯云计算（北京）有限责任公司

操作步骤

在集群列表页，根据实际使用的视图模式进行操作：

页签视图

1. 登录 TDSQL-C MySQL 版控制台，在左侧集群列表，单击目标集群，进入集群管理页。
2. 在集群详情下，找到目标实例，单击**重启**。



3. 在弹出的对话框，确认待重启实例的信息，单击**确定**。
4. 待该实例状态从**重启中**恢复为**运行中**，即完成实例重启。

列表视图

1. 登录 TDSQL-C MySQL 版控制台，在集群列表中，找到目标集群，单击集群 ID 或**操作**列的**管理**，进入集群管理页面。
2. 在集群管理页面，选择**实例列表**页，找到需要重启的实例，单击**重启**。



3. 在弹出的对话框，确认待重启实例的信息，单击**确定**。
4. 待该实例状态从**重启中**恢复为**运行中**，即完成实例重启。



删除实例

最近更新时间：2025-01-15 10:06:42

当实例不再使用，可以对实例进行删除。本文为您介绍如何删除集群下的实例。

> **注意：**
> - 读写实例删除后数据将无法找回，备份文件会同步删除，无法在云上进行数据恢复，请提前做好备份文件的转存。
> - 实例删除后 IP 资源同时释放，删除前请确认无业务需对该实例进行访问。
> - 直接删除集群时，整个集群下的所有实例会全部被删除。
> - 只有在完成读写节点的删除后，数据才会被删除，删除只读实例只会删除计算资源。

操作场景

- 集群的实例形态为预置资源时，支持删除集群下的只读实例，当集群下全部只读实例被删除后，才可以删除该集群下的读写实例，删除读写实例后集群也被删除。
- 集群的实例形态为 Serverless，架构为集群版时，不支持单独删除读写实例或只读实例，只读实例的减少需参考 只读实例管理 来操作；架构为单节点版时，删除读写实例意味着整集群删除。

生命周期

- 包年包月实例删除后，实例被移入回收站保留7天，期间实例无法访问。如您想恢复该实例，可在回收站进行续费恢复。
- 按量计费/Serverless 实例删除后，实例被移入回收站保留3天，期间实例无法访问。如您想恢复该实例，可在回收站进行续费恢复。

> **说明：**
> 对于 Serverless 集群而言，在回收站进行恢复的场景仅针对读写实例，通过 只读实例管理 操作减少的只读实例不会在回收站隔离，而是直接下线。

操作步骤

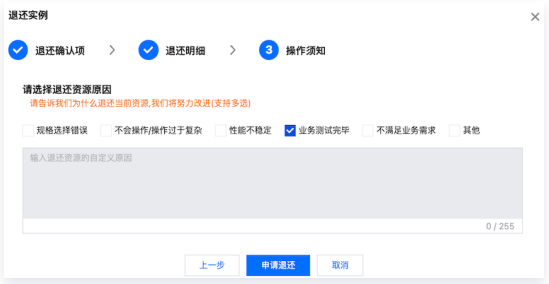
在集群列表页，根据实际使用的视图模式进行操作：

页签视图

1. 登录 TDSQL-C MySQL 版控制台，在左侧集群列表，单击目标集群，进入集群管理页。
2. 在集群详情下，找到目标实例，单击**退还/退款**。



3. 在弹出的对话框，完成**退还确认项** > **退还明细** > **操作须知**的阅读或勾选后，单击**申请退还**。



列表视图

1. 登录 TDSQL-C MySQL 版控制台，在集群列表中，找到需要删除实例的集群，单击集群 ID 或**操作**列的**管理**，进入集群管理页面。
2. 在集群管理页面，选择**实例列表**页，找到需要删除的读写实例或只读实例，选择**更多 > 退还/退款**。



3. 在弹出的对话框，完成**退还确认项 > 退还明细 > 操作须知**的阅读或勾选后，单击**申请退还**。



恢复实例

最近更新时间：2024-06-12 15:53:32

本文为您介绍如何通过控制台恢复已隔离状态的实例。

操作场景

若由于误操作、实例到期、欠费等导致的实例被删除暂存在回收站当中，可以在回收站中将实例进行恢复使用。

> **注意：**
> - 实例恢复后，所有的配置均不会发生变化。
> - 实例短时间内无法连续被销毁，恢复。
> - 对于 Serverless 集群而言，在回收站进行恢复的场景仅针对读写实例，通过 只读实例管理 操作减少的只读实例不会在回收站隔离，而是直接下线。

操作步骤

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在左侧导航选择**回收站**，选择地域，在回收站列表找到待恢复的实例，在**操作**列单击**恢复**。

> **注意：**
> - 必须恢复了同一个集群下的读写实例后，才能恢复该集群下的只读实例。
> - 您需要在销毁期限内进行实例恢复，否则无法恢复实例。
> - 操作列的**立即释放**为立即下线该实例，选择此操作后会不可进行实例恢复，需谨慎操作。

3. 在弹出的对话框，选择续费时长，确认实例信息后，单击**确定**。



4. 恢复完成后，可在实例列表中查看到该实例，实例状态为**运行中**。

任务列表

最近更新时间：2025-03-18 17:38:12

您可以通过 TDSQL-C MySQL 版控制台查看所执行的任务详情，方便您了解对集群和集群下的实例进行操作的任务进度。本文为您介绍如何通过控制台查看任务列表。

任务类型

通过任务列表，您可查看的任务类型包括：集群创建、集群回档、集群克隆、整集群回档至新集群、创建整集群回档用新集群、库表回档至新集群、创建库表回档用新集群、实例升降级、存储配置调整、参数修改、重启、手动备份、自动备份、版本升级、新增备可用区、修改备可用区、删除备可用区、切换可用区、修改备可用区属性、修改 binlog 设置、创建数据库代理、数据库代理升级、关闭数据库代理、数据库代理配置连接池、数据库代理节点重新负载、数据库代理内核小版本升级、配置读写分离、开启密码强度、修改密码强度、关闭密码强度、拷贝密码强度、开通审计、修改审计、关闭审计、数据库代理创建连接、数据库代理修改连接、数据库代理删除连接、一键迁移、连接地址切换、强制重启、开启 CLS 投递、停止 CLS 投递、删除 CLS 投递、数据加密、切换 Serverless、只读实例配置、Serverless 配置、只读实例切换 Serverless、集群暂停、集群启动、实例暂停、实例启动、隔离实例、添加实例、开启 SSL、关闭 SSL、集群销毁、开启外网、关闭外网等。

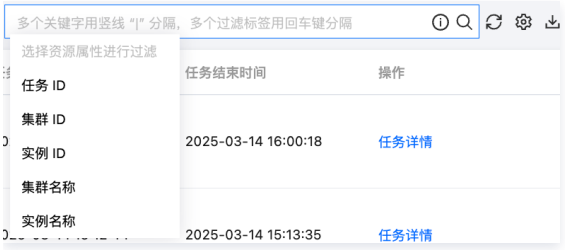
查看任务列表

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在左侧导航栏单击**任务列表**进入任务列表界面。
3. 在上方选择对应地域。
4. 您可以在任务列表直接查找或检索关键字查询对应任务并查询任务详情。



检索关键字

在任务列表，您可以通过搜索框检索关键字快速查找到目标任务，搜索框内支持按照任务 ID、集群 ID、实例 ID、集群名称、实例名称的资源属性来进行搜索，多个关键字用“|”分隔，多个过滤标签用回车键分隔。



下载任务数据

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在左侧导航栏单击**任务列表**进入任务列表界面。
3. 在上方选择对应地域。
4. 单击检索框后的下载按钮，可下载当前页的数据或者当前查询条件下的数据。

查看任务详情

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在左侧导航栏单击**任务列表**进入任务列表界面。
3. 在上方选择对应地域。
4. 找到需要查询任务详情的任务项，在**操作**列单击**任务详情**。

版权所有：腾讯云计算（北京）有限责任公司


任务 ID
任务类型
集群 ID / 名称
实例 ID / 名称
任务执行进度
任务执行状态
任务开始时间
任务结束时间
操作
重启
100 %
执行成功
2022-01-23 17:21:47
2022-01-23 17:21:57
任务详情


任务详情 - 重启
任务 ID
集群 ID / 名称
实例 ID / 名称
开始时间
2022-01-23 17:21:47
结束时间
2022-01-23 17:21:57
任务进度
100%
任务状态
执行成功
提示信息
ok
关闭

配置变更
调整计算配置

最近更新时间：2025-01-16 10:57:43

当实例性能无法满足业务变化或实例性能过剩时，可对实例进行变配操作，以满足业务诉求或节省成本。
本文为您介绍实例形态为预置资源的集群如何调整计算配置，实例形态为 Serverless 的集群调整配置请参见 调整配置 。

> **说明：**
> 若集群的实例形态为预置资源且开通了数据库代理，则在调整计算配置页面，可以选择是否对数据库代理的配置进行调整。

变配能力

TDSQL-C MySQL 版采用计算和存储分离的架构，所有计算节点共享一份数据。在跨机变配的场景下，完全无需搬迁数据，所以能够支撑秒级的配置升降级能力。
当实例变配时，会自动拉起一个高规格的新主实例。当完成建立之后，自动将新的主实例升级为主库，旧的主库网络断开即达到了变配的目的。

操作步骤

步骤一：在集群列表页，根据实际使用的视图模式操作，进入配置调整页面：

页签视图

1. 登录 TDSQL-C MySQL 版控制台，在左侧集群列表，单击目标集群，进入集群管理页。
2. 在集群详情下，找到目标实例，单击**调整配置**进入配置调整页面。



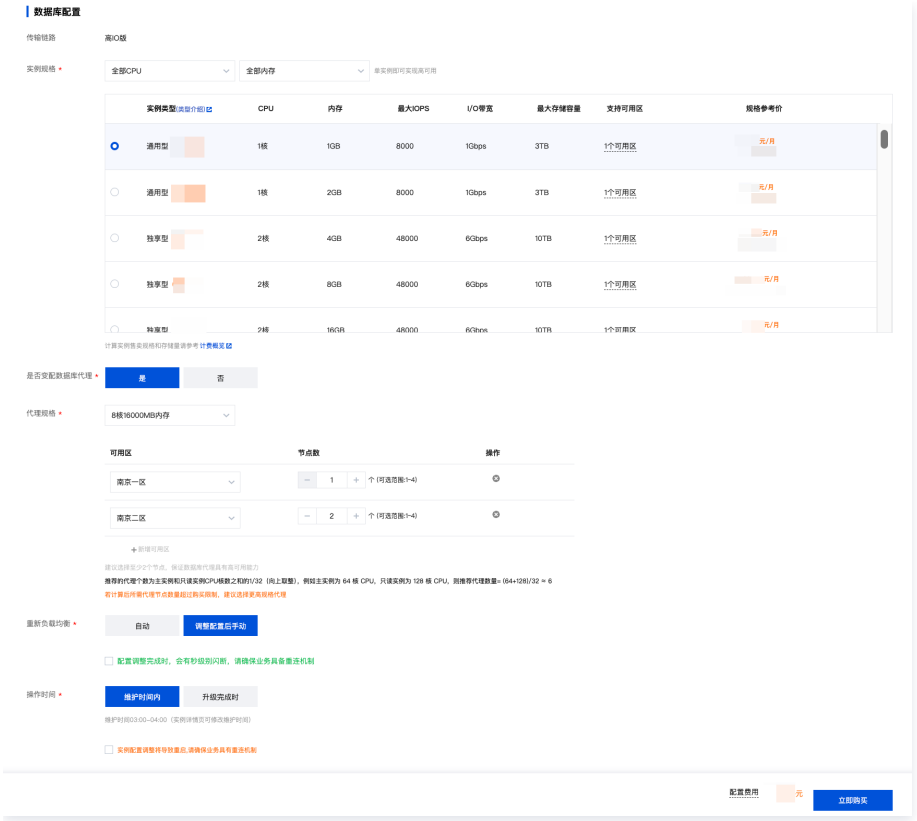
列表视图

1. 登录 TDSQL-C MySQL 版控制台，在集群列表中，找到需要调整配置的集群，单击集群 ID 或**操作**列的**管理**，进入集群管理页面。
2. 在集群管理页面，选择**实例列表**页，找到想要变更配置的实例，选择**更多 > 调整配置**。



步骤二：在实例配置调整页面中，选择需要的配置，选择操作时间，单击**立即购买**。

> **注意：**
> 实例配置调整会出现秒级闪断，请确保业务具有重连机制。



- 实例规格：选择需要调整的 CPU 和内存。
- 是否变配数据库代理：选择是否调整数据库代理节点的配置，关于调整数据库代理节点各项配置的影响说明，请参见 调整数据库代理配置 。此选项仅当集群的实例形态为预置资源且开通了数据库代理时才会显示。
- 操作时间：
  - 选择**升级完成时**，则会在完成配置变更后，立即触发切换闪断动作。
  - 选择**维护时间内**，则会在实例的**维护窗口期**内进行闪断切换。修改维护窗口期可参考 修改实例维护时间 。

调整存储空间

最近更新时间：2024-07-02 17:05:31

当集群存储空间无法满足业务需求时，可对集群的存储空间进行变配操作，以满足业务需求。
本文为您介绍实例形态为预置资源的集群如何调整存储空间，实例形态为 Serverless 的集群存储空间为按量计费，无须预购存储空间，但是对应的算力配置有对应的存储空间上限，具体可参见 服务算力配置，若您需要调整算力配置来获取更大的存储空间上限，可参见 修改集群的算力配置。

说明：

- 实例形态为预置资源的集群，若计算计费模式和存储计费模式均为按量计费，则无需对存储空间进行扩容，您可使用的存储空间最大容量为该集群读写实例的计算规格所对应的存储空间上限，若要使用超过此计算规格上限的存储空间，您可对该读写实例的计算规格进行升级，除此以外，您还可以通过 按量转包年包月，将集群的计费模式调整为包年包月后进行存储空间的变更，但需注意，计费模式暂不支持从包年包月回退为按量计费。详细计算规格和对应最大存储空间请参见 产品规格。
- 实例形态为预置资源的集群，若计算计费模式为包年包月，存储计费模式为按量计费，由于存储计费模式支持由按量计费转为包年包月，因此您可以在存储计费模式转为包年包月的同时调整存储空间。
- 存储计费模式为包年包月时，您可通过控制台调整集群的存储空间，若要调整超过此计算规格上限的存储空间，您可对读写实例进行计算规格升级。详细计算规格和对应最大存储空间请参见 产品规格。
- 存储计费模式为包年包月时，变更存储空间的有效计费时间为变更存储时到您的集群到期的时间。

操作步骤

说明：

- 存储预付费规格变更需大于已使用存储容量。
- 存储计费模式修改不会影响现有业务。

场景一：存储计费模式为按量计费，需使用超过当前计算规格上限的存储空间
存储计费模式为按量计费时，无需对存储空间进行扩容，您可使用的存储空间最大容量为对应读写实例的计算规格对应的存储空间上限，若要使用超过此计算规格上限的存储空间，您可对该读写实例进行计算规格升级，步骤请参考 调整计算配置。详细计算规格和对应最大存储空间请参见 产品规格。
场景二：存储计费模式为按量计费，需转为包年包月并指定存储空间
在集群列表页，根据实际使用的视图模式进行操作：

页签视图

1. 登录 TDSQL-C MySQL 版控制台，在左侧集群列表，单击目标集群，进入集群管理页。
2. 在集群管理页，计费模式后，单击编辑图标。



3. 在弹出的设置窗口下，选择包年包月计费模式，然后选择储存空间大小，单击**确定**完成支付即可将存储计费模式变为包年包月并指定存储空间。



列表视图

1. 登录 TDSQL-C MySQL 版控制台，在集群列表中，找到需要调整存储空间的集群，单击集群 ID 或**操作**列的**管理**，进入集群管理页面。
2. 在**集群详情** > **配置信息** > **数据库存储使用量**后，单击图标 。

说明：

您也可以在集群详情页下的**计费信息** > **存储计费**后，单击图标 进行调整。

3. 在弹出的设置窗口下，选择包年包月计费模式，然后选择储存空间大小，单击**确定**完成支付。



场景三：存储计费模式为包年包月，需调整存储空间

在集群列表页，根据实际使用的视图模式进行操作：

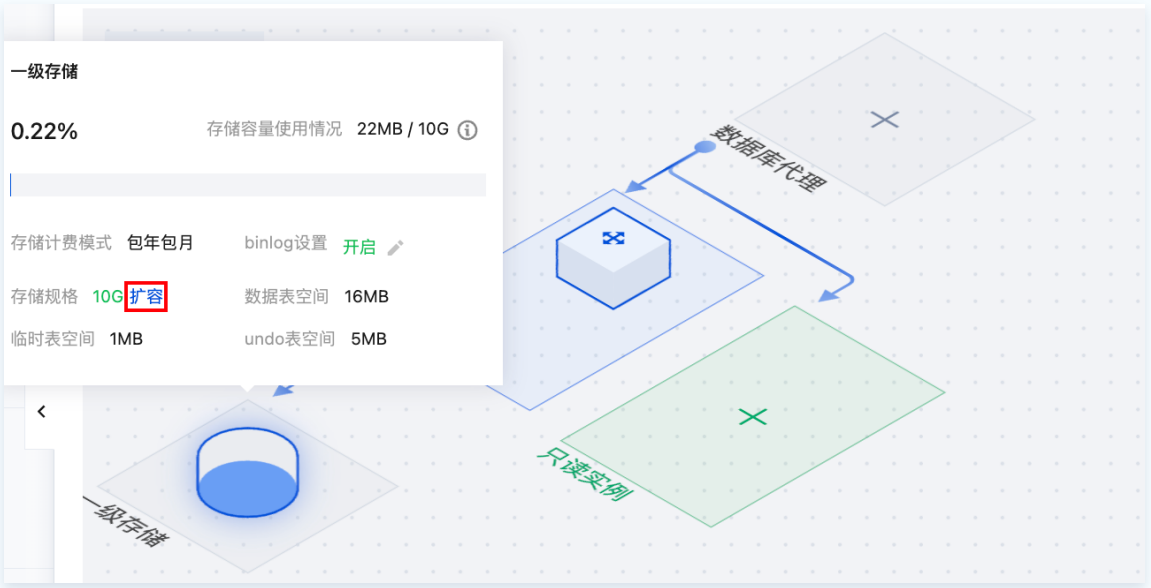
页签视图

1. 登录 TDSQL-C MySQL 版控制台，在左侧集群列表，单击目标集群，进入集群管理页。
2. 在集群管理页，计费模式后，单击编辑图标。



说明：

您也可以在集群详情页下方的拓扑图中，将鼠标移到存储板块，然后单击存储规格后的**扩容**进行调整。



3. 在弹出的设置窗口下，选择储存空间大小，单击**确定**完成支付。



列表视图

1. 登录 TDSQL-C MySQL 版控制台，在集群列表中，找到需要调整存储空间的集群，单击集群 ID 或**操作**列的**管理**，进入集群管理页面。
2. 在**集群详情** > **配置信息** > **数据库存储使用量**后，单击图标 。

> **说明：**
>
> 您也可以在集群详情页下的**计费信息** > **存储计费**后，单击图标 进行调整。

3. 在弹出的设置窗口下，选择储存空间大小，单击**确定**完成支付。



升级内核小版本

最近更新时间：2025-06-20 20:43:41

TDSQL-C MySQL 版会陆续发布新的数据库内核小版本，通过升级数据库内核小版本，可以使得您的数据库实例获得性能提升、安全性增强、新功能支持、兼容性改进和错误修复等升级。本文为您介绍通过控制台升级数据库内核小版本的操作和说明。

> **说明：**
> 如需了解各个数据库内核小版本的功能详情，请参见 内核版本更新动态 。

操作场景

当系统发布新的数据库内核小版本时，或者您当前集群的数据库内核小版本过旧，需要升级到新的数据库内核小版本时，您可以通过控制台进行手动升级。

> **说明：**
> - 如果后台程序出现 bug，或者系统发现安全隐患，系统将通过站内信、短信等方式推送升级通知，TDSQL-C MySQL 版会在指定时间进行系统升级和修复。
> - 当发生触发集群迁移的操作（如升降配集群规格、扩缩容磁盘容量、数据库版本升级等）时，系统不会主动将您的集群升级至最新内核小版本，如需升级，您可手动升级。

升级规则

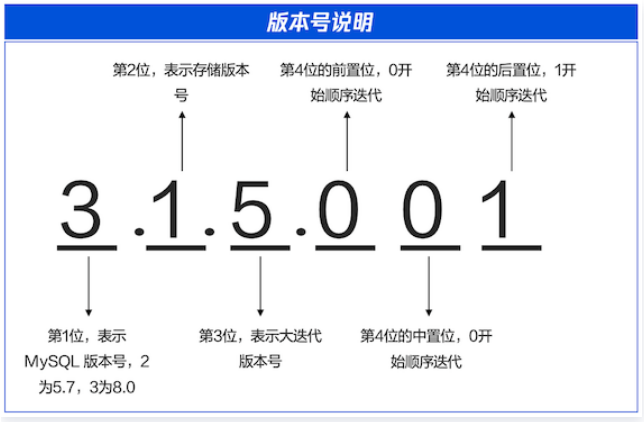
- 为保证数据复制一致性，待升级数据库内核小版本的集群下的关联实例（读写实例、只读实例）会一起进行内核小版本升级。
- 数据库内核小版本升级会额外拉起计算节点进行升级，不涉及数据搬迁，可以实现快速升级。

注意事项

- 数据库内核小版本升级切换阶段有秒级闪断，建议选择在维护时间内或业务低峰期切换，减少对业务的影响。
- 单个实例的表数量超过100万后，可能会造成升级失败，同时也会影响数据库监控，请合理规范表的数量，控制单个实例表数量不超过100万。
- 内核小版本升级后无法降级。

四位小版本号说明

TDSQL-C MySQL 版的数据库内核小版本全新支持了四位小版本号。版本号由原来的三位更新为四位，可避免发生被动升级，保障更新版本时实例的稳定运行，同时在大版本内（例如3.1.15这种三位大版本）系统可推出多个附带新能力的小版本，帮助您更好的体验产品新能力。
四位小版本号中，每一位的描述，如下图所示。



- 标准版：表示大版本的第一个小版本，比如大版本为2.1.12，则第一个数据库内核小版本号为2.1.12.001。
- 尝鲜版：表示大版本的最新小版本，比如大版本为2.1.12，则最新数据库内核小版本为2.1.12.003或2.1.12.004。尝鲜版本提供多种灰度中的功能。

升级内核小版本

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在集群列表页，根据您实际使用的视图模式进行操作，进入数据库内核小版本升级窗口。

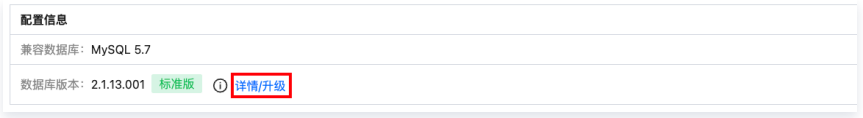
页签视图

1. 在左侧集群列表，单击**目标集群**，进入集群管理页。
2. 在集群管理页，数据库版本后，单击**详情/升级**。



列表视图

1. 在集群列表，找到目标集群，单击**集群 ID** 或**操作**列的**管理**，进入集群详情页。
2. 在集群详情页的**配置信息** > **数据库版本**后，单击**详情/升级**。



3. 在弹出的侧边栏，单击**集群立即升级**。
4. 在弹窗下选择切换时间和要升级的目标内核小版本，勾选“已阅读并同意数据库版本升级说明”，单击**确定**。



切换时间

- 维护时间内：升级完成后，会在实例的维护时间内进行切换。修改维护窗口期可参考 设置实例维护时间 。
- 升级完成时：升级完成后立即触发切换。

5. 如需了解升级任务进度，您可前往 任务列表 进行查询。
6. 当实例运行状态更新为“运行中”时，即升级完成。

实例形态管理
实例形态简介

最近更新时间：2024-12-06 14:38:43

TDSQL-C MySQL 版支持两种数据库实例形态，一种是预置资源，一种是 Serverless。本文为您介绍 TDSQL-C MySQL 版的实例形态定义，两种实例形态的对比以及转换实例形态的方法。

实例形态定义



预置资源

实例形态为预置资源的实例，表示需预先分配固定规格。目前一个预置资源的集群下，拥有1个读写实例和1 - 15个只读实例，对于只读实例类型而言，一个预置资源的集群下可以既挂载预置资源的只读实例，又挂载 Serverless 的只读实例，并且支持挂载 LibraDB 只读分析引擎。

Serverless

实例形态为 Serverless 的实例，表示无需预先分配固定的规格，按实际使用量付费，并可根据业务负载自动扩缩容，可以实现资源的最大化利用和成本的最优化控制。目前一个 Serverless 的集群下，拥有1个读写实例，如果是 Serverless 单节点版，则该集群下无只读实例，如果是 Serverless 集群版，则该集群下可以挂载1 - 15个只读实例，只读实例的类型仅支持配置 Serverless 的只读实例。

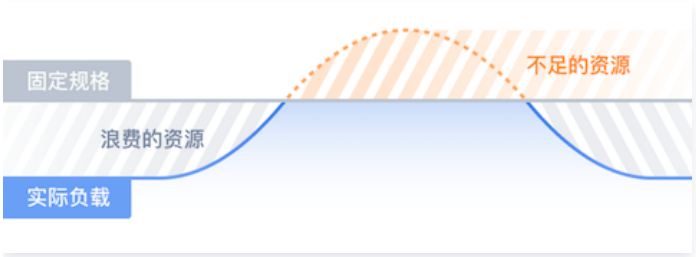
实例形态对比

适用场景

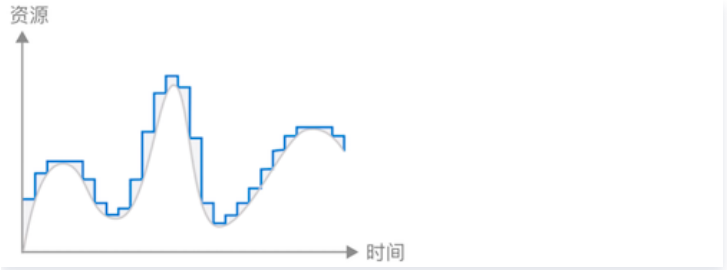
预置资源（未启用 Serverless）集群	预置资源（启用 Serverless）集群	Serverless 集群
• 业务负载比较稳定。 • 固定实例规格能满足业务压力变化。 • 经常需要快速回档的游戏项目。 • PB 级数据存储访问场景。 • 写入 QPS 要求较高的场景。 • 对主从延迟较敏感的场景。	• 有明显业务波峰波谷的场景。 • 不确定负载的场景。 • 希望保留已有的预置资源实例形态，同时又能够满足业务波动的场景。	• 期望降低运维成本、提升运维效率的场景。 • 业务峰值偏波动的场景。 • 低频使用数据库的场景，例如开发、测试环境。

业务负载波动时的资源利用

预置资源为固定规格，实际业务负载波动较大时，如未手动对实例进行升降配，可能出现资源浪费（波谷时期）和资源不足（波峰时期）。



Serverless 具备动态弹性扩缩容能力，使用资源（CCU 和存储）可随业务需求量及时调整，总体资源浪费较少，可提升资源利用率，根据配置的算力区间，可在波峰时期快速扩容，充分满足业务需求，在波谷时期，自动缩容，有效降低使用成本。

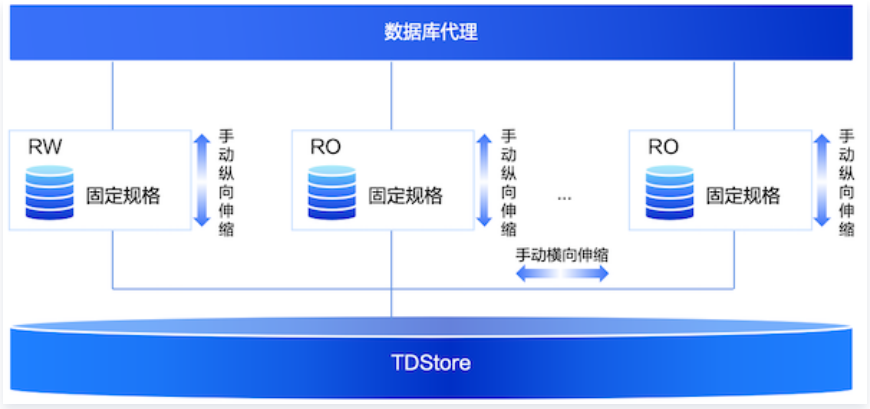


预置资源和 Serverless 相比，从贴合业务负载以及成本角度来看，在业务波动较大的场景下，Serverless 实例形态更贴合业务需求。



架构形式

预置资源（未启用 Serverless）集群



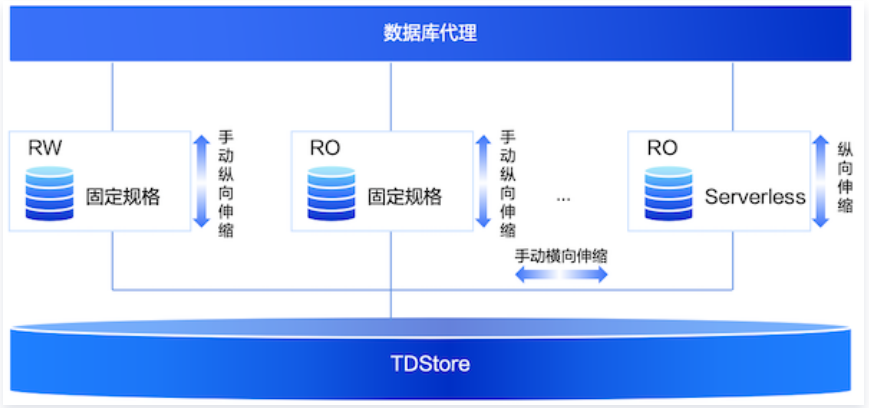
计算

读写实例（RW）和只读实例（RO）为预先配置的固定规格，可通过手动的方式，对实例的计算规格、实例数量进行扩缩容配置。

存储

- 包年包月的存储空间，由您根据业务情况自主选择大小，支持扩缩容。
- 按量计费存储空间，由您所选的计算规格决定其上限，可通过升级计算规格来获取更大的存储空间上限。

预置资源（启用 Serverless）集群



计算

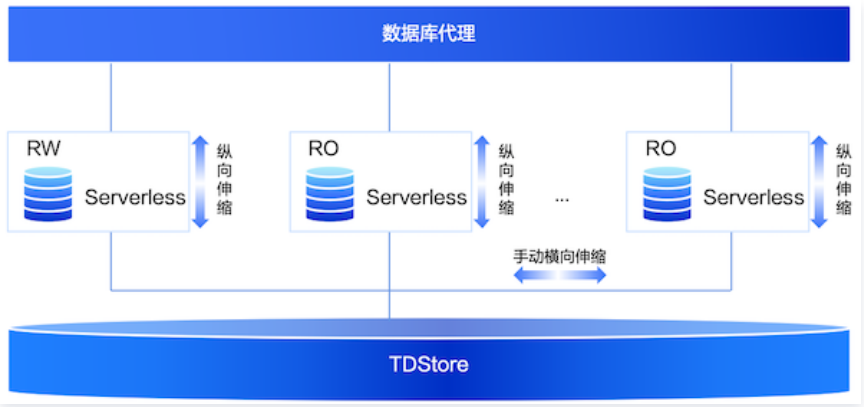
读写实例（RW）为预先配置的固定规格，可通过手动的方式，对实例的计算规格进行扩缩容配置。只读实例（RO）可以包含预置资源的只读实例，以及 Serverless 只读实例。其中，预置资源部分不随业务压力的变化而改变，Serverless 部分随业务负载变化而弹性扩缩（纵向），每当 Serverless 只读节点随业务波动扩缩容时，CCU 算力也会随之增大和减小。建议在第一次设置弹性范围时，最小容量配置为0.25 CCU，最大容量选择较高的值。

存储

- 包年包月的存储空间，由您根据业务情况自主选择大小，支持扩缩容。

- 按量计费存储空间，由您所选的计算规格决定其上限，可通过升级计算规格来获取更大的存储空间上限。

Serverless 集群



计算

读写实例（RW）和只读实例（RO）均为 Serverless 形态，均可随业务负载变化而弹性扩缩（纵向），每当 Serverless 只读节点随业务波动扩缩容时，CCU 算力也会随之增大和减小。您可以手动修改 Serverless 只读实例的数量和算力区间。

存储

Serverless 集群存储空间计费方式为按量计费，创建时无需选择存储空间，按照实际数据量所占的存储空间大小进行收费，每个算力区间对应的最大存储空间上限不同，具体可参见 服务算力配置 。

部署及功能

对比项	预置资源集群	Serverless 集群
购买类型	一个集群内包含1个读写实例和最多15个只读实例	• 单节点版：1个读写实例 • 集群版：1个读写实例和最多15个只读实例
存储引擎	InnoDB	InnoDB
内核	• TXSQL • LibraDB	TXSQL
版本	• MySQL 5.7 • MySQL 8.0	• MySQL 5.7 • MySQL 8.0
实例规格	CPU 和内存，单实例最高88核710GB	算力区间，CCU 量，最小算力为0.25，最大算力为64
自动备份	默认保留7天，最高可设置保留1830天	默认保留7天，最高可设置保留1830天
手动备份	支持	支持
备份文件形式	• 逻辑备份 • 快照备份	• 逻辑备份 • 快照备份
回档	支持	支持
实例生命周期管理	支持	支持
集群管理	支持	支持
监控与告警	支持	支持
账号管理	支持	支持
数据库管理	支持	支持

版权所有：腾讯云计算（北京）有限责任公司

数据库代理	支持	支持
最大建表数	不对创建数据库的数量和创建表的数量进行限制，理论上有足够的空间，就可以创建更多的库与表	不对创建数据库的数量和创建表的数量进行限制，理论上有足够的空间，就可以创建更多的库与表
读写分离	支持	支持

详细功能差异对比，请参见 产品功能列表 。

费用

- 实例形态为预置资源时，计费方式支持包年包月和按量计费，计费项包含计算、存储。如果启用 Serverless，挂载了 Serverless 只读实例，则 Serverless 只读实例的计费方式为 Serverless，则集群下的总费用包含固定规格部分的费用以及 Serverless 的费用。
- 实例形态为 Serverless 时，计费方式为 Serverless，计费项包含计算、存储。
- 详细产品价格请参见 产品价格 和 计费概述 。

如何创建这两种实例形态的集群

可通过购买页创建不同实例形态的集群：

- 创建实例形态为预置资源的集群，操作方法请参见 创建集群 。
- 创建实例形态为 Serverless 的集群，操作方法请参见 创建 Serverless 版集群 。

已创建的集群支持转换实例形态

> **说明：**
> 下表为您列举实例形态的转换场景和操作指引，目前一个预置资源的集群下，可以挂载预置资源和 Serverless 这两种类型的只读实例，其整体的实例类型在控制台仍属于预置资源集群。

实例形态	转换场景	操作指引
预置资源	预置资源集群转为 Serverless 集群。	• 按量转 Serverless • 包年包月转 Serverless
	预置资源集群启用 Serverless。 **说明：** 启用 Serverless 后，表示一个预置资源的集群下，可同时挂载预置资源只读节点和 Serverless 只读节点。	预置资源集群启用或关闭 Serverless
	预置资源实例转 Serverless。 **说明：** 表示将预置资源集群下的一个预置资源只读节点转换为 Serverless 只读节点，转换前后实例 IP 地址、实例名称、实例 ID 不变，此转换场景为实例维度的转换。	预置资源实例转 Serverless
Serverless	Serverless 不支持转换为预置资源。	-

预置资源集群启用或关闭 Serverless

最近更新时间：2024-12-04 20:49:22

实例形态为预置资源的集群，支持启用或关闭 Serverless。启用 Serverless，表示预置资源的集群开启了同时挂载预置资源只读节点和 Serverless 只读节点的能力，因 Serverless 服务按实际计算和存储资源使用量收取费用，可实现自动启停、自动扩缩容等特性，因此，启用 Serverless 能帮助您更好的管理预置资源集群各节点在业务上的应用，更精细的控制成本。本文为您介绍为预置资源集群启用或关闭 Serverless 的操作方法。

前提条件

已创建实例形态为预置资源的集群，操作请参见 创建集群 。

注意事项

- 预置资源集群启用 Serverless 后，预置资源只读节点使用的是预置资源集群的参数模板，支持参数公式，Serverless 只读节点使用的是 Serverless 的参数模板，不支持参数公式。
- 预置资源集群启用 Serverless 后，变更 Serverless 只读节点的参数时，若该参数是预置资源节点（包含读写和只读节点）不支持的参数，则不支持同时变更预置资源节点的参数。
- 预置资源集群启用 Serverless 后，变更预置资源节点（包含读写和只读节点）的参数时，若该参数是 Serverless 只读节点不支持的参数，则不支持同时变更 Serverless 只读节点的参数。

操作步骤

为预置资源集群启用 Serverless

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在左侧集群列表，单击目标集群，进入集群管理页。
3. 单击右上角的 **Serverless 配置**。



4. 在弹窗下，打开 Serverless 启用状态后的按钮，完成如下配置，单击**确定**。



参数	说明
只读节点算力配置	• 个数：选择 Serverless 只读节点的个数，取值范围是1 – 15个，且预置资源只读节点数 + Serverless 只读节点数 ≤ 15。 • 算力区间：配置只读节点的算力区间，不支持单个节点单独配置。
自动暂停	• 勾选自动暂停，需要设定自动暂停时间，默认为1小时。数据库在该时间内没有连接和 CPU 使用时，将自动暂停，暂停后计算不计费，存储仍然按实际使用量计费。

- 未勾选自动暂停，数据库会保持持续运行，在没有连接和 CPU 使用时，按您配置的最小 CCU 算力进行计费。

> **说明：**
> 若集群开启了数据库代理，则不支持设置自动暂停。

修改 Serverless 配置

为预置资源集群启用 Serverless 后，如需修改有关 Serverless 只读节点的配置，您可在集群管理页，单击右上角的 **Serverless 配置**，然后在弹窗下进行 Serverless 只读节点数量、算力配置修改。

关闭 Serverless

> **说明：**
> - 关闭 Serverless 操作会一键销毁该集群下的 Serverless 只读节点，请谨慎操作。
> - 被销毁的 Serverless 只读节点会进入回收站，在回收站期间，若对应集群重新启用 Serverless，则该集群选择挂载的 Serverless 只读节点会优先从回收站进行恢复。若恢复数量小于该集群在回收站的 Serverless 只读节点的数量，则随机进行恢复；若恢复数量大于该集群在回收站的 Serverless 只读节点的数量，且系统判断该恢复后该集群只读节点总数小于15时，会优先恢复该集群在回收站的全部 Serverless 只读节点。
> - 关闭 Serverless 后，新增预置资源的只读节点时，若该集群在回收站的 Serverless 只读节点数量加预置资源只读节点数量小于15，则可以正常添加；若大于15，则会出现不可添加预置资源只读节点的情况，需先在回收站下线 Serverless 只读节点后再重试。

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在左侧集群列表，单击目标集群，进入集群管理页。
3. 单击右上角的 **Serverless 配置**。
4. 在弹窗下，关闭 Serverless 启用状态后的按钮，单击**确定**，关闭 Serverless 并销毁实例。



预置资源实例转 Serverless

最近更新时间：2025-03-21 16:25:03

实例形态为预置资源的集群，支持将集群下预置资源的只读实例/节点转为 Serverless 只读节点。本文为您介绍将预置资源集群下的预置资源只读节点转换为 Serverless 只读节点的操作方法。

前提条件

- 已创建实例形态为预置资源的集群，操作请参见 创建集群 。
- 集群下存在至少1个预置资源的只读实例，创建预置资源的只读实例请参见 创建只读实例 。

注意事项

- 预置资源实例转 Serverless 后，预置资源只读节点使用的是预置资源集群的参数模板，支持参数公式，Serverless 只读节点使用的是 Serverless 的参数模板，不支持参数公式。
- 预置资源实例转 Serverless 后，变更 Serverless 只读节点的参数时，若该参数是预置资源节点（包含读写和只读节点）不支持的参数，则不支持同时变更预置资源节点的参数。
- 预置资源实例转 Serverless 后，变更预置资源节点（包含读写和只读节点）的参数时，若该参数是 Serverless 只读节点不支持的参数，则不支持同时变更 Serverless 只读节点的参数。

使用限制

- 若集群下的预置资源只读实例已开启 并行查询，则该实例不支持转 Serverless，如需转 Serverless，请关闭并行查询能力后再进行转化。
- 若集群下的预置资源只读实例已开启 列存索引，则该实例不支持转 Serverless，如需转 Serverless，请关闭列存索引能力后再进行转化。

操作步骤

说明：

- 转化后，参数设置将重置为默认参数模板（例如 max_connections 参数值会变成80）。
- 若预置资源集群下没有 Serverless 只读节点，则将预置资源只读实例转为 Serverless 时，可以修改算力配置；若预置资源集群下已有 Serverless 只读节点，要将其他预置资源只读实例转为 Serverless 时，不支持修改算力配置。
- 转化时实例会有瞬时闪断，请确保业务具备重连机制。
- 转化后的节点 IP 地址、实例名称、实例 ID 不变。
- 转化后，无法从 Serverless 只读节点转回预置资源只读节点。

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在左侧集群列表，单击**目标集群**，进入集群管理页。
3. 在实例信息下找到需要转 Serverless 的只读实例，在其**配置**后，单击转 **Serverless**。



4. 在弹窗下，完成如下配置，勾选阅读事项，单击**确定**。

参数	说明
算力配置	配置算力区间。
自动暂停	• 勾选自动暂停，需要设定自动暂停时间，默认为1小时。数据库在该时间内没有连接和 CPU 使用时，将自动暂停，暂停后计算不计费，存储仍然按实际使用量计费。 • 未勾选自动暂停，数据库会保持持续运行，在没有连接和 CPU 使用时，按您配置的最小 CCU 算力进行计费。

	说明： 若集群开启了数据库代理，则不支持设置自动暂停。

集群管理

集群管理简介

最近更新时间：2025-01-14 17:27:22

本文为您介绍 TDSQL-C MySQL 集群概念、集群管理操作及如何查看集群列表。

集群概述

集群是包含读写实例和只读实例的一组网络资源。TDSQL-C MySQL 版最基本的管理单元为集群，一个集群中最多只能有一个读写实例，最多可以创建15个只读实例。

集群管理概览

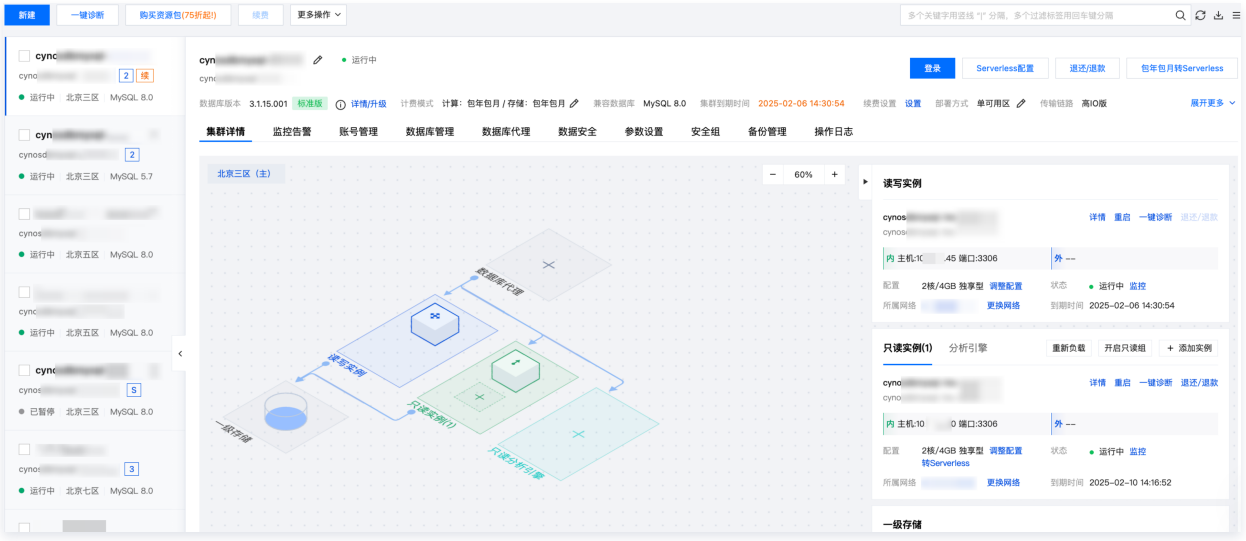
集群管理项	说明	操作方法
修改集群名	• 您可对集群名进行修改 • 集群名需满足的格式要求：长度小于60，由中文、英文字母、“-”、“_”、“.”	修改集群名
修改集群项目	您可对集群所属项目进行设置和修改	修改集群项目
删除集群	请确定此集群您不再使用，再执行删除	删除集群
恢复集群	• 您可在误删或其他需要恢复的情况下恢复已隔离状态的集群 • 回收站会显示该集群可恢复的期限	恢复集群

查看集群列表

在集群列表页，根据实际使用的视图模式进行操作：

页签视图

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在上方选择地域，选择引擎版本为 MySQL，即可查询对应地域下的集群列表信息。



列表视图

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在上方选择地域，选择引擎版本为 MySQL，即可查询对应地域下的集群列表信息。


TDSQL-C - 集群列表
北京 47
其他地域 30
MySQL
PostgreSQL
新功能速递
列表视图
异常告警
用户指南
尊敬的用户，为实现小规格实例平滑变配能力，TDSQL-C MySQL 版将于2024年1月16日--31日期间，对 table_open_cache_instances、innodb_buffer_pool_instances 的参数值类型进行变更，影响范围为参数模版。本次变更不更改实例的参数运行数值，只更改数值类型，对集群性能无影响，变更后参数将不会随着规格变动而更改，请您及时关注集群状态，由此给您带来的不便敬请谅解。点击了解
新建
一键诊断
续费
更多操作
多个关键字用竖线 "|" 分隔，多个过滤标签用回车键分隔
集群 ID / 集群名
集群状态
兼容数据库
可用区
读写地址
只读地址
数据库代理地址
计费模式
所属项目
操作
运行中
MySQL 8.0
北京三区
(内) 10.
(外) 未开启
已开启
(内) 10
3306
计算:包年包月
存储:按量计费
默认项目
登录 管理 更多

修改集群名

最近更新时间：2025-01-15 10:06:43

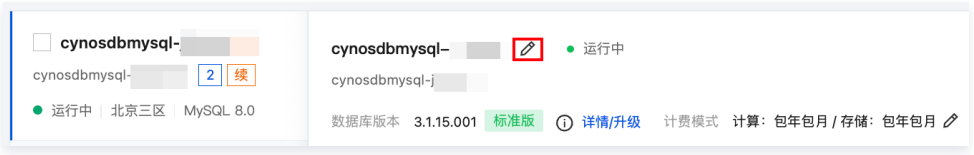
您可以通过集群名对 TDSQL-C MySQL 版集群进行区分和管理，默认集群名和集群 ID 保持一致，也可以通过控制台对集群名字进行修改。

操作步骤

在集群列表页面，根据实际使用的视图模式进行操作：

页签视图

1. 登录 TDSQL-C MySQL 版控制台，在左侧集群列表，单击目标集群，进入集群管理页。
2. 在集群管理页，单击集群名后的编辑图标。



3. 在弹出的对话框，填写需要修改的集群名称，单击**确定**。

> **说明：**
> 集群名仅支持长度小于60的中文，英文，数字，-(中横线)，_(下划线)，.(英文标点)。

列表视图

在集群列表中修改集群名

1. 登录 TDSQL-C MySQL 版控制台，在集群列表中，找到需要的集群，单击集群名后的编辑图标。



2. 在弹出的对话框，填写需要修改的集群名称，单击**确定**。

> **说明：**
> 集群名仅支持长度小于60的中文，英文，数字，-(中横线)，_(下划线)，.(英文标点)。

在集群详情中修改集群名

1. 登录 TDSQL-C MySQL 版控制台，在集群列表中，找到需要的集群，单击集群 ID，进入集群详情页面。
2. 在集群详情页面，单击集群名后的编辑图标。



3. 在弹出的对话框，填写需要修改的集群名称，单击**确定**。

> **说明：**
> 集群名仅支持长度小于60的中文，英文，数字，-(中横线)，_(下划线)，.(英文标点)。

修改集群项目

最近更新时间：2025-01-15 10:06:43

TDSQL-C MySQL 版支持将集群分配至不同的项目进行管理。
本文介绍通过控制台修改集群所属项目。

操作步骤

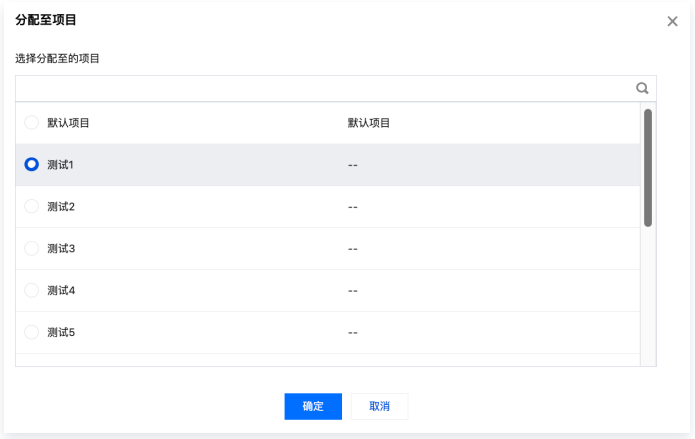
在集群列表页面，根据实际使用的视图模式进行操作：

页签视图

1. 登录 TDSQL-C MySQL 版控制台，在左侧集群列表，单击目标集群，进入集群管理页。
2. 在集群管理页，所属项目后，单击**调整**。



3. 在弹出的对话框，选择新分配至的项目，单击**确定**。

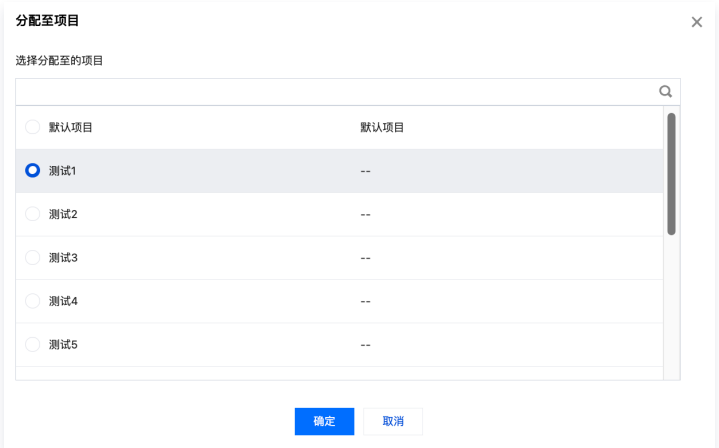


列表视图

1. 登录 TDSQL-C MySQL 版控制台。
2. 在集群列表，找到目标集群，单击**集群 ID** 或操作列的**管理**，进入集群详情页面。
3. 在集群详情页>基本信息里，找到**所属项目**，单击其后的**转至其他项目**。



4. 在弹出的对话框，选择新分配至的项目，单击**确定**。



删除集群

最近更新时间：2024-06-12 15:53:32

确认业务不再使用 TDSQL-C MySQL 版集群后，可以对集群进行删除。
本文介绍如何通过控制台删除集群。

注意：

- 删除集群后，此集群下的所有实例（包含读写实例和所有只读实例）均会被自动删除。
- 对于按量计费集群，删除集群将自动停止计费。
- 对于包年包月集群，在未到期时删除集群将自动按照按量计费价格与使用时长重新计算集群中所有实例的费用，根据已付费金额对产生的费用进行扣减后退还。

操作场景

删除整个集群。

生命周期说明

- 包年包月实例删除后，实例被移入回收站保留7天，期间实例无法访问。如需恢复该实例，可在回收站进行续费恢复。
- 按量计费/Serverless 实例删除后，实例被移入回收站保留3天，期间实例无法访问。如需恢复该实例，可在回收站进行续费恢复。

说明：
对于 Serverless 集群而言，在回收站进行续费恢复的场景仅针对读写实例，通过 只读实例管理 操作减少的只读实例不会在回收站隔离，而是直接下线。

操作步骤

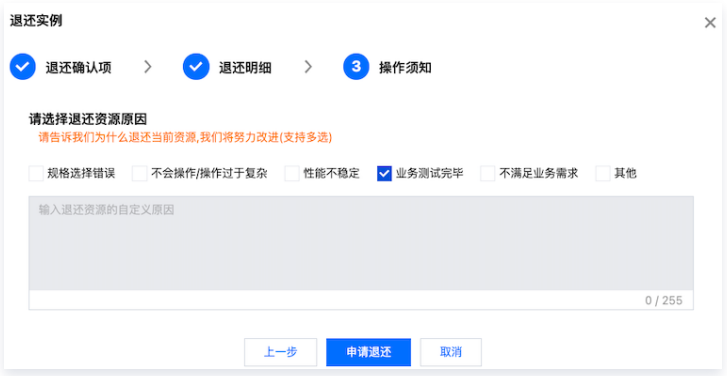
在集群列表页面，根据实际使用的视图模式进行操作：

页签视图

1. 登录 TDSQL-C MySQL 版控制台，在左侧集群列表，单击目标集群，进入集群管理页。
2. 在集群管理页右侧，单击**退还/退款**。



3. 在弹出的对话框，完成**退还确认项** > **退还明细** > **操作须知**的阅读或勾选后，单击**申请退还**。



列表视图

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在集群列表中，找到目标集群，在**操作**列选择**更多 > 退还/退款**。



3. 在弹出的对话框，完成**退还确认项 > 退还明细 > 操作须知**的阅读或勾选后，单击**申请退还**。
4. 删除集群后，集群下的实例将自动放入回收站中。在回收站中的实例无法访问与管理，也不会产生费用，隔离期间可将实例从回收站中进行恢复使用。

恢复集群

最近更新时间：2024-06-12 15:53:32

本文介绍如何通过控制台恢复已隔离的集群。

操作场景

若因为误操作、实例到期、欠费等导致集群被删除暂存在回收站时，您可以在回收站中恢复该集群下的所有读写实例和只读实例。

> **注意：**
> - 集群下只读实例和读写实例恢复后，所有的配置均不会发生变化。
> - 集群无法连续被删除和恢复。
> - 对于 Serverless 集群而言，在回收站进行恢复的场景仅针对读写实例，通过 只读实例管理 操作减少的只读实例不会在回收站隔离，而是直接下线。

操作步骤

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在左侧导航选择**回收站**，选择地域，在回收站列表找到待恢复的集群下的读写实例和只读实例，在读写实例的**操作**列单击**恢复**。



3. 在只读实例的**操作**列单击**恢复**。

> **说明：**
> - 必须先恢复该集群下的读写实例后，才能恢复其只读实例。
> - 您需要在销毁期限内进行恢复，否则无法恢复。
> - 操作列的**立即释放**为立即下线该实例，选择此操作后不可进行实例恢复，请谨慎操作。

4. 恢复完成后，可在集群列表中查看到此集群，集群状态为**运行中**。

只读实例管理
只读实例简介

最近更新时间：2024-12-04 20:49:22

在对数据库有少量写请求，但有大量读请求的应用场景下，单个实例可能无法承受读取压力，甚至对业务产生影响。为了实现读取能力的弹性扩展，分担数据库压力，您可以创建一个或多个只读实例，利用只读实例满足大量的数据库读取需求，增加应用的吞吐量。

前提条件

兼容数据库版本为 MySQL 5.7 或 MySQL 8.0（8.0的内核小版本需为3.1.2及以上）。

计费

- **计算计费模式**
  - 实例形态为预置资源时：若集群下的读写实例为包年包月，则对应新扩展预置资源只读节点的计算计费模式也为包年包月；若集群下的读写实例为按量计费，则对应新扩展预置资源只读节点的计算计费模式也为按量计费；若集群通过启用 Serverless 新扩展 Serverless 只读节点，或者将预置资源只读节点转为 Serverless 只读节点，则新的 Serverless 只读节点计费模式为 Serverless。
  - 实例形态为 Serverless 时：集群新增加的只读节点计费模式为 Serverless。
- **存储计费模式**

  购买时无需预购存储空间，TDSQL-C MySQL 版按每小时存储实际使用量计费，只读实例的存储使用的是整个集群的存储。

只读实例配置

- **地域和可用区**：与读写实例在同一地域，同一可用区。
- **兼容数据库**：与读写实例一致。
- **规格**：可以与读写实例不一致，并且可以随时更改。建议只读实例规格不小于读写实例规格，否则易导致只读实例延迟高、负载高等现象。
- **实例名**：支持长度小于60的中文、英文、数字、`-`、`_`、`.`。
- **网络类型**：与读写实例一致。
- **所属项目**：与主实例一致。
- **监控与告警**：提供丰富的系统性能指标的监控视图，如内存使用量、QPS、TPS、最大连接数等。

功能限制

- **可创建的只读实例数量**：15个。
- **数据库管理**：不支持创建和删除数据库。
- **账号管理**：不支持创建和删除账号，不支持为账号授权以及修改账号密码功能。

创建只读实例

- 实例形态为预置资源的操作请参见 创建只读实例 。
- 为实例形态为预置资源的集群启用 Serverless 来增加 Serverless 只读节点的操作请参见 预置资源集群启用或关闭 Serverless 。
- 实例形态为 Serverless 的操作请参见 只读实例管理 。

版权所有：腾讯云计算（北京）有限责任公司

创建只读实例

最近更新时间：2024-12-04 20:49:22

本文介绍实例形态为预置资源的集群通过 TDSQL-C MySQL 版控制台创建预置资源的只读实例/节点。
如需为实例形态为预置资源的集群启用 Serverless 来增加 Serverless 只读节点的操作请参见 预置资源集群启用或关闭 Serverless 。
如需了解实例形态为 Serverless 的集群的相关操作，请参见 只读实例管理 。

操作场景

TDSQL-C MySQL 版支持用户为一个集群创建一个或多个只读实例，用以支持读写分离和一写多读应用场景，增加只读实例可以显著提升数据库集群的读负载能力。

- TDSQL-C MySQL 版集群支持两种实例类型：读写实例，只读实例。其中，根据实例形态不同，只读实例分为预置资源只读实例、Serverless 只读实例。
- TDSQL-C MySQL 版集群默认提供读写内网地址和只读内网地址，用户可以通过集群只读内网地址访问所有的只读实例。只读实例创建完成后，通过集群只读内网地址访问新只读实例，访问请求会自动转发到该只读实例。
- 只读实例费用与读写实例相同，请参见 产品价格 。

> **说明：**
> 如何访问 TDSQL-C MySQL 版，请参见 连接 TDSQL-C MySQL 版集群 。

注意事项

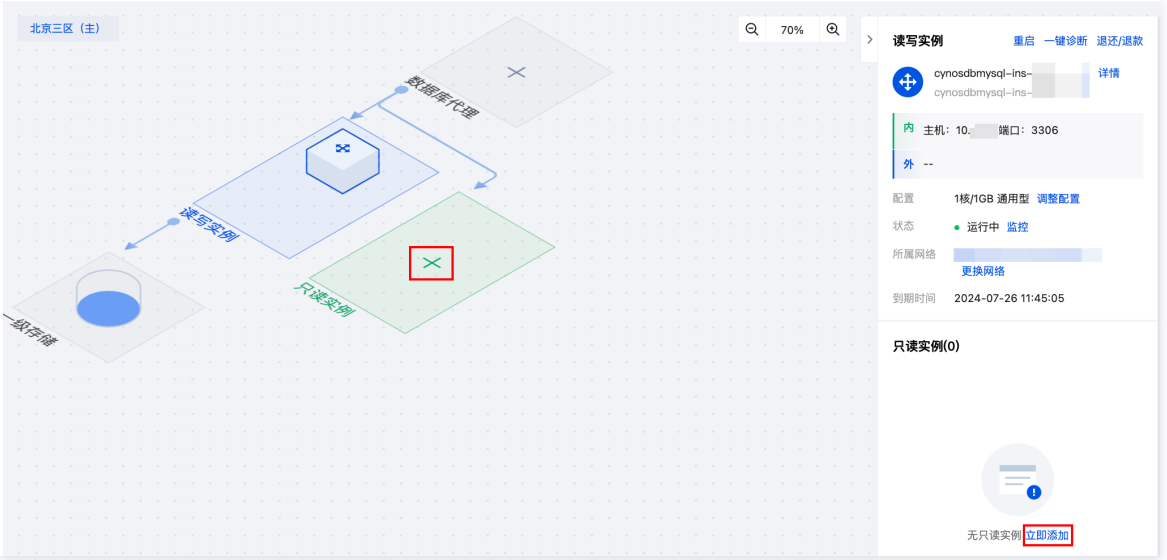
- 只读实例与读写实例共享同一个存储，无需维护账号与数据库。
- 只读实例无需通过 binlog 与读写实例保持同步，无需复制搬迁数据，通常只需秒级时间即可完成只读实例的创建。
- 只读实例与读写实例之间的时延通常在毫秒级，可通过监控与告警页的只读实例时延监控指标进行查看。
- 只读实例规格可以与读写实例不一致，但为了方便用户根据负载情况调整配置，建议各只读实例的规格保持一致。
- 包年包月集群添加包年包月只读实例时，新添加实例的到期时间将与集群的到期时间保持一致。

操作步骤

在集群列表页面，根据实际使用的视图模式进行操作：

页签视图

1. 登录 TDSQL-C MySQL 版控制台 ，在左侧集群列表，单击目标集群，进入集群管理页。
2. 在集群详情下，在拓扑图中的只读实例板块单击“+”，或者在右侧只读实例信息下单击**立即添加**。



3. 跳转至购买页，选择只读实例的相应配置，确认无误后，单击**立即购买**创建只读实例，创建成功后可在集群的实例列表页查看新添加的只读实例。

列表视图

1. 登录 TDSQL-C MySQL 版控制台，在集群列表中，单击**集群 ID/集群名**或**操作**列的**管理**，进入集群管理页面。
2. 在集群管理页面，选择**实例列表**页，选择**只读实例** > **添加只读实例**，进入只读实例购买页。



3. 在购买页，选择只读实例的相应配置，确认无误后，单击**立即购买**创建只读实例，创建成功后可在集群的实例列表页查看新添加的只读实例。

数据库代理
数据库代理概述

最近更新时间：2024-10-12 17:12:18

本文为您介绍 TDSQL-C MySQL 版数据库代理服务的特性、适用场景。
数据库代理是位于云数据库服务和应用服务之间的网络代理服务，用于代理应用服务访问数据库时的所有请求。
数据库代理访问地址独立于原有的数据库访问地址，通过数据库代理地址的请求，全部通过代理集群中转访问数据库的主从节点，进行读写分离，将读请求转发至只读实例，降低主库的负载。

> **说明**
> 数据库代理目前处于公测阶段，支持地域陆续开放中，如您的集群所在地域暂无数据库代理选项即暂未开放，敬请见谅。

数据库代理特性

- **高稳定性**
  采用集群架构部署，多节点保证故障平滑转移。
- **强隔离性**
  数据库代理使用独立资源为当前实例提供代理服务（各代理资源独立，不共享资源）。
- **超高性能**
  每个代理每秒最高可以处理10万左右的请求。
- **扩容方便快捷**
  支持1个 - 60个代理节点动态扩展（公测期间仅支持6个节点）。
- **完备的性能监控**
  提供了读/写请求数、CPU、内存等性能指标的秒级监控，可以根据 监控数据 及业务规划调整代理个数。
- **热加载**
  主实例发生切换、变配、只读实例增减等情况，数据库代理可动态热加载配置，不会出现网络中断或重启。
- **支持 自动读写分离**
  通过开启数据库代理的读写分离功能，可以有效降低主实例的读负载，通过添加只读实例来提供数据库集群的水平扩展能力，并且可以帮助用户实现自助读写分离，降低用户自行拆分业务读写请求的复杂度，尤其适合大量读负载的情况。
  开通数据库代理的读写分离功能后，应用程序中只需配置一个代理连接地址，该地址会自动实现读写分离，将读请求发送至只读实例，将写请求发送至主实例。即使增加或删除只读实例，也无需调整应用程序的设置。

适用场景

- 大量短连接为主的业务，性能不足。
- 业务使用多个只读实例，在应用程序侧手动读写分离，维护成本和风险较高。
- 连接数过多导致实例负载过高。

版权所有：腾讯云计算（北京）有限责任公司

注意事项

最近更新时间：2025-01-14 14:36:12

本文介绍 TDSQL-C MySQL 版数据库代理的注意事项。

- 使用代理连接地址时，如果没有启用事务拆分，事务请求都会路由到主实例。
- 数据库代理配置支持跨可用区，可选择的可用区数量与当前地域可选可用区数量相关，最多支持选择四个可用区，当可选可用区数量为一个时，表示当前地域可选可用区数量仅有一个。
- 可设置多个数据库代理访问地址，其数量与数据库代理节点相同，即有多少个节点，则支持设置多少个数据库代理访问地址。
- 使用代理连接地址进行读写分离时，不保证非事务读的一致性，业务上有读一致性需求可以封装到事务中，或者使用 Hint 语法。
- 使用代理连接地址时，show processlist 会将所有节点的结果合并后返回。
- 对于 prepare 语句，数据库代理会先将 prepare 发送到所有节点，当后续的 execute 请求到来时，根据 prepare 的语句类型来决定 execute 的路由。如 prepare 了一个写语句，execute 时会发到主库，而如果 prepare 的是事务外的读语句，则会发送到只读实例。
- 业务连接到达数据库代理后，代理会连接到读写实例和所有配置的只读实例，数据库代理本身并没有最大连接数的限制，连接数的限制主要由后端数据库实例的最大连接数决定（读写实例和只读实例最大连接数的最小值会影响业务性能）。
- 开启数据库代理后，新增只读实例或只读实例重启，只有新的连接的请求才会路由到新的只读实例或重启的只读实例，可以通过概览或 性能监控 查看各代理节点的性能指标，若发现各代理节点连接数不均衡的现象，可通过重新负载均衡打散连接。
- Serverless 服务只有在关闭自动启停能力时，可开启数据库代理功能。
- 只读分析引擎加入到数据库代理中，默认的权重为 0，且不可修改。只能通过 `/*to server 实例 ID*/` 的语法将 SQL 转发至只读分析引擎。

版权所有：腾讯云计算（北京）有限责任公司

数据库代理内核特性
内核小版本更新说明

最近更新时间：2024-07-02 17:05:31

本文为您介绍 TDSQL-C MySQL 版数据库代理的版本更新说明。

查看数据库代理版本

在集群列表页面，根据实际使用的视图模式进行查看：

页签视图

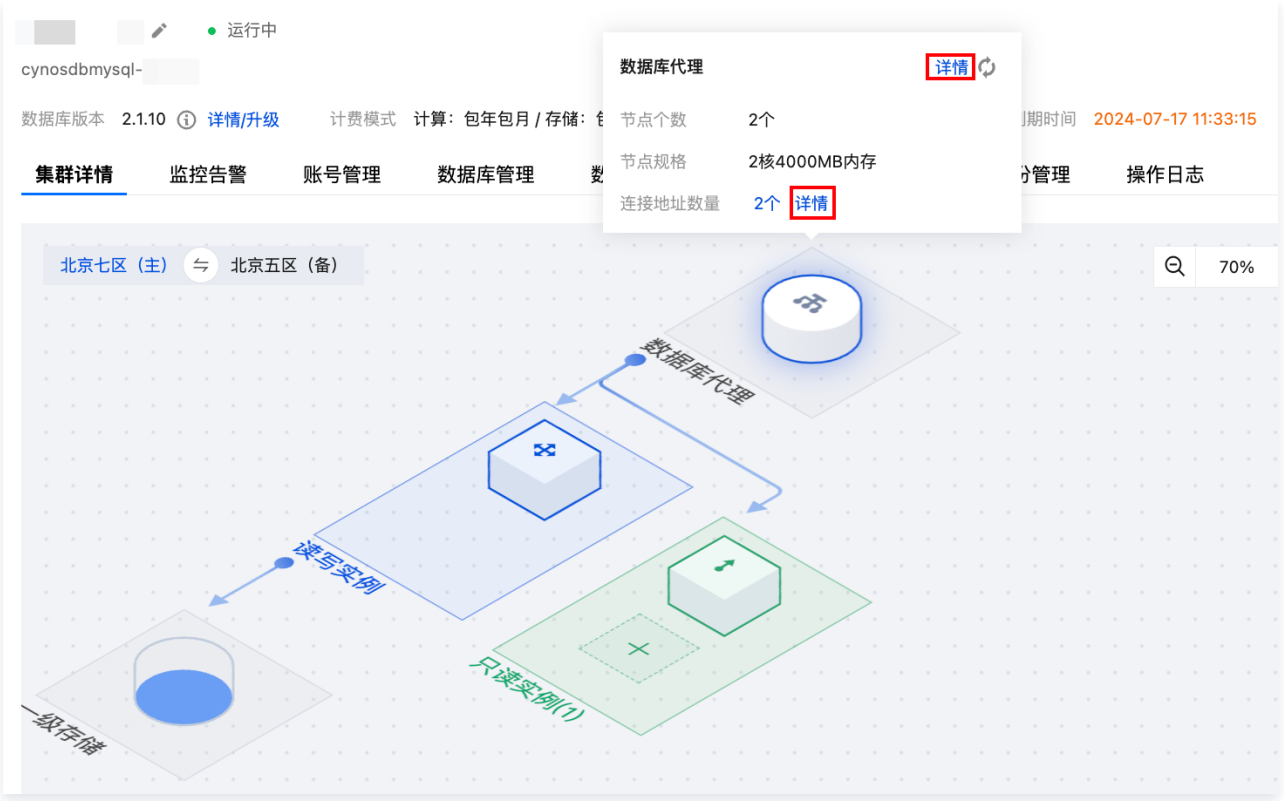
方法一：

您可在集群管理页下，单击**数据库代理**，在**概览 > 基本信息 > 代理版本**处查看。



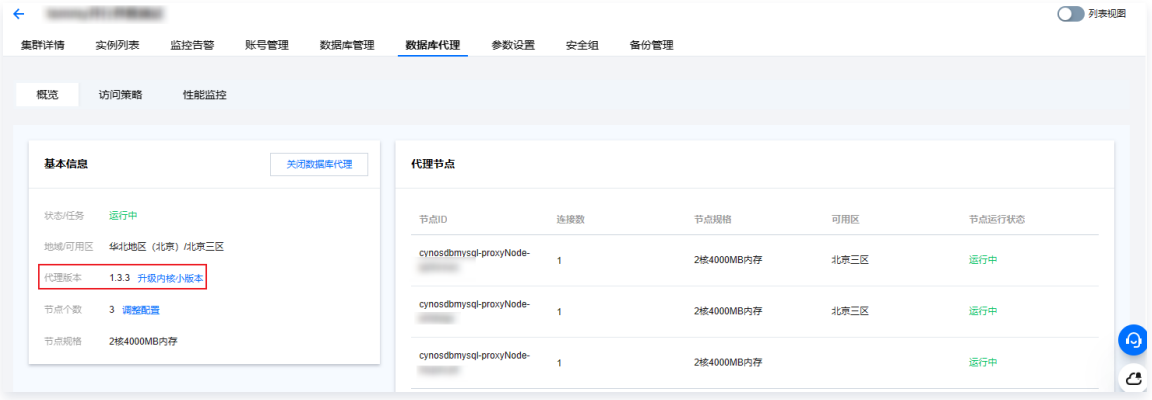
方法二：

您也可以在集群管理页的集群详情下，在拓扑图中，鼠标移动到数据库代理板块，然后单击**详情**，跳转进入数据库代理页，然后在**概览 > 基本信息 > 代理版本**处查看。



列表视图

您可通过实例的 数据库代理 页的**概览** > **代理版本**处查看。



版本说明

说明：

- 数据库代理内核版本更新说明请参见 内核版本更新动态 – 数据库代理内核版本更新说明 。
- 如不满足 TDSQL-C MySQL 版内核版本要求，可先升级数据库内核版本，详细操作请参见 升级内核小版本 。

升级数据库代理内核小版本

最近更新时间：2025-06-27 16:17:11

TDSQL-C MySQL 版会不定期更新数据库代理的版本，新的版本可能涉及功能更新、性能优化或问题修复，您可以查看数据库代理版本更新动态，并结合实际业务评估是否需要升级。本文为您介绍通过控制台手动升级数据库代理版本的操作及说明。

> **说明：**
> 如需使用滚动升级能力，请先 提交工单 申请使用。

前提条件

- 已 开通数据库代理 。
- 当前实例的数据库代理版本不是最新版。

升级方案

您可以根据业务实际情况，选择升级切换时间以实现更平滑的升级。

- 维护时间内：可以选择在实例维护时间内升级，设置实例维护时间请参见 修改实例维护时间 。
- 升级完成时：支持在升级完成时立即切换至所选择的数据库代理版本。
- 滚动升级：拉起新的数据库代理版本节点，拉起后，新连接将通过新数据库代理版本节点访问数据库，旧数据库代理版本节点将不再接受新的连接请求，可设置切换等待时间，当到达切换时间后，旧的数据库代理版本节点才会被回收，同时切断旧数据库代理版本节点上的存量旧连接。

注意事项

- 升级数据库代理版本会出现秒级别的闪断。请您尽量在业务低峰期执行升级操作，并确保您的应用具备自动重连机制。
- 升级数据库代理版本是默认所有代理节点同时升级，若部分代理节点状态异常，则无法升级。
- 仅支持升级数据库代理版本，不支持降级。

升级过程中可执行任务说明

在升级数据库代理版本过程中，当**状态/任务**为**内核小版本升级中**时，您可以对数据库代理执行的操作为：修改数据库代理的网络、修改数据库代理内网地址、修改数据库代理地址的备注。您不可以对数据库代理执行的操作为：调整配置、关闭数据库代理、升级内核小版本、重新负载均衡、修改网络、修改 IP，如需执行，请立即完成升级任务后再执行。



升级数据库代理版本

1. 登录 TDSQL-C MySQL 版控制台，在上方选择地域，然后单击**目标集群 ID**，进入集群管理页。
2. 在集群管理页，选择**数据库代理**页。
3. 在数据库代理页的**概览** > **基本信息** > **代理版本**后，单击**升级内核小版本**。



4. 在弹出的对话框，选择需升级的目标版本，选择升级切换时间，确认无误后，单击**确定**。



参数	操作说明
当前版本	显示实例当前的数据库代理版本。
目标版本	在此项选择要升级的目标数据库代理版本。有关各版本的更新说明，请参见 数据库代理版本更新动态 。
切换时间	选择升级的切换时间，支持选择维护时间内、升级完成时、滚动升级三种。 • **维护时间内**：选择维护时间内，会在下一个维护时间内完成升级切换。 • **升级完成时**：选择升级完成时，会在版本升级完成时立即进行切换。 • **滚动升级**：选择滚动升级，需要设置切换等待时间，默认为10秒，取值范围：10 – 3600秒，到达切换等待时间后，系统回收旧版本节点。
升级涉及的相关节点	在此项下面，会展示本次升级涉及的数据库代理节点信息。

取消执行

在升级数据库代理版本过程中，如果切换时间选择为**滚动升级**但还未到达切换等待的时间，此时支持根据实际情况取消本次任务的执行，取消执行后，本次数据库代理版本不会升级。

1. 在数据库代理页的**概览** > **基本信息** > **状态/任务**后单击**取消执行**。



2. 在弹窗中，单击**确定**。

立即执行

> **说明：**
> 操作立即执行后，会立即回收旧数据库代理版本节点，旧数据库代理版本节点上的存量连接会立即断开。

在升级数据库代理版本过程中，如果切换时间选择为**维护时间内**但还未到达维护时间，或者切换时间选择为**滚动升级**但还未到达切换等待的时间，此时如果您需要立即执行升级，切换数据库代理版本，可通过以下操作执行。

1. 在数据库代理页的**概览** > **基本信息** > **状态/任务**后单击**立即执行**。
2. 在弹窗中，单击**确定**。

相关 API

API	描述
升级数据库代理版本	接口（UpgradeCDBProxyVersion）用于升级数据库代理版本。

管理数据库代理
开通数据库代理

最近更新时间：2025-01-14 14:36:12

本文为您介绍如何通过 TDSQL-C MySQL 版控制台开启数据库代理。
数据库代理是位于云数据库服务和应用服务之间的网络代理服务，用于代理应用服务访问数据库时的所有请求。数据库代理提供自动读写分离、连接池、连接保持、一致性级别设置等高级功能，具有高可用、高性能、可运维、简单易用等特点。

注意事项

- 数据库代理目前支持的地域：
  - 北京（除一、二、四区）、上海（除一区）、广州（除一、二区）、成都、重庆、南京、香港（除一区）。
  - 东京（除一区）、弗吉尼亚（除一区）、硅谷（除一区）、首尔（除一区）、新加坡（除一、二区）。
- 数据库代理目前支持的版本：MySQL 5.7（内核小版本须2.0.19及以上），MySQL 8.0（内核小版本须3.1.5及以上），升级集群内核小版本会同时升级所关联的读写实例、只读实例，请参见 升级内核小版本 。
- 只读分析引擎所有版本均支持数据库代理。

操作步骤

1. 登录 TDSQL-C MySQL 版控制台，在集群列表，选择需要开启代理的集群，单击集群 ID 或**操作**列的**管理**，进入集群管理页面。
2. 在集群管理页面，选择**数据库代理**页，单击**立即开启**。



3. 在弹出的对话框，完成如下配置，单击**确定**。



参数	说明
网络类型	选择数据库代理的网络，仅支持私有网络 VPC。
代理规格	支持选择规格为2核4000MB内存、4核8000MB内存、8核16000MB内存。
可用区及节点个数	1. 选择数据库代理可用区，支持单击新增可用区来多选，可选择的可用区数量与当前地域可选可用区数量相关。 2. 选择节点个数，推荐的代理节点个数为主实例和只读实例 CPU 核数的之和的1/8（向上取整），例如主实例为4核 CPU，只读实例为8核 CPU，则推荐代理数量 = (4 + 8) / 8 ≈ 2，当前版本最多支持选择4个数据库代理节点，建议选择至少2个节点，保证数据库代理具有高可用能力。 3. 若您的集群为多可用区部署，为保证数据库代理高可用性，推荐您在主备可用区均配置数据库代理节点，且节点数量为一致。 **⚠ 注意：** • 如果所选数据库代理与主实例不在同一可用区，通过数据库代理连接时，写入性能可能会下降。 • 若计算推荐节点个数后所需代理节点数量超过购买限制，建议选择更高规格代理。
安全组	可根据需要选择已有安全组或新建安全组。 **⚠ 注意：** 访问数据库代理需要开通配置安全策略，放通内网访问端口（当前内网端口为：3306），具体详见 安全组配置 。
备注	非必填项，可为要开通的数据库代理服务进行备注。



4. 开通成功后，可在数据库代理页查看基本信息，管理代理节点，也可在连接地址项对数据库代理地址、网络类型修改和添加备注。

> **说明：**
> - 您可通过查看代理节点列表的**连接数**或查看各代理节点的性能监控，来判断各节点是否出现访问不均衡现象。若出现各代理节点连接数不均衡的现象，可单击连接地址中的**重新负载均衡**打散连接。
> - 重新负载均衡会触发代理节点重启，重启过程中会有短暂服务不可用现象，建议在低峰期重启服务，请确保业务具备重连机制。

修改或删除连接地址

最近更新时间：2024-07-02 17:05:31

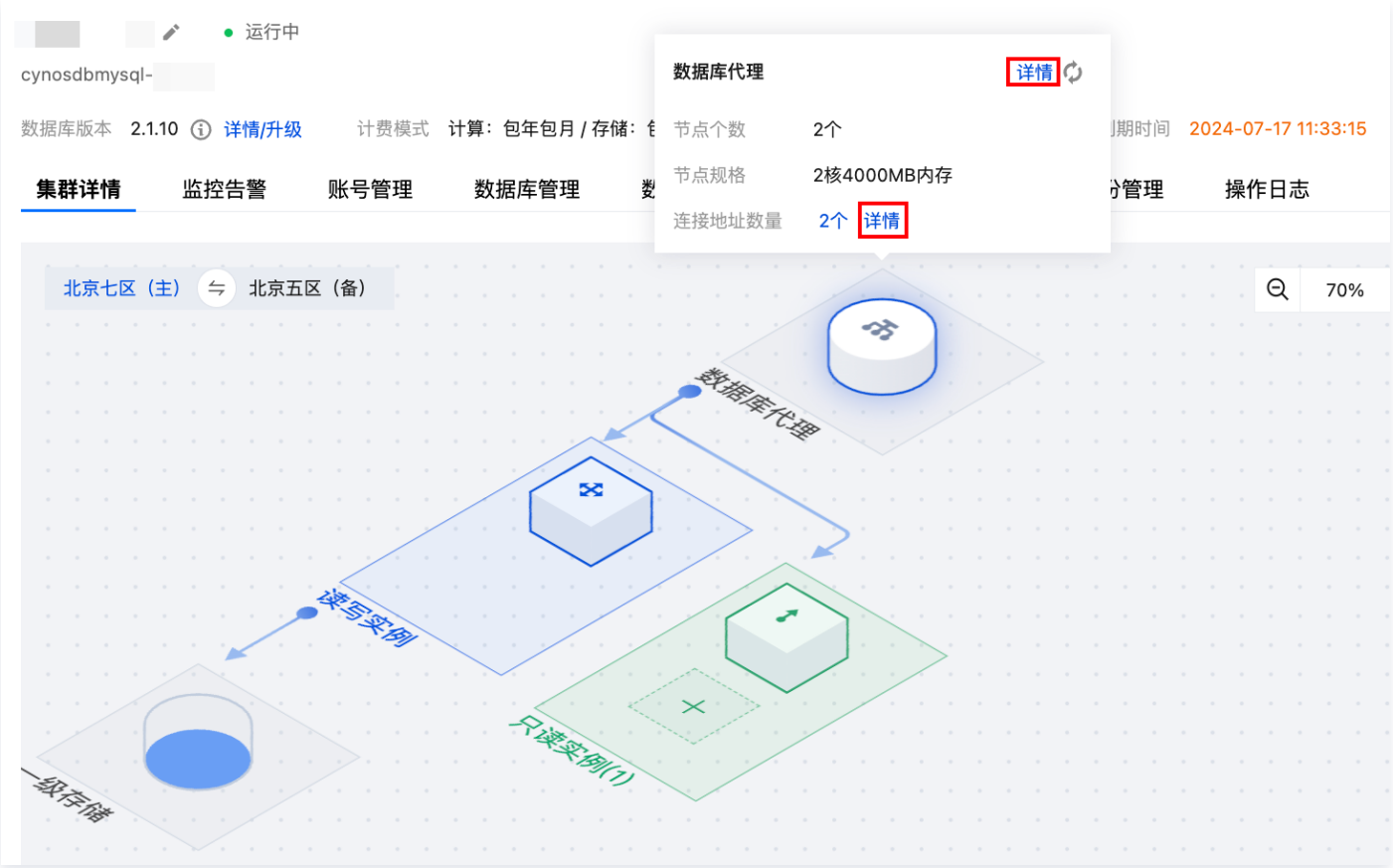
本文为您介绍如何通过 TDSQL-C MySQL 版控制台修改或删除数据库代理连接地址。

前提条件

已 开通数据库代理 。

修改数据库代理内网地址

1. 登录 TDSQL-C MySQL 版控制台，选择集群所属的**地域**。
2. 在集群列表，选择已开启代理的集群，单击集群 ID，进入集群管理页面。
3. 在**集群管理** > **集群详情**下的拓扑图中，鼠标移动到数据库代理板块，然后单击**详情**进入数据库代理管理页面。或在**集群管理**页面选择**数据库代理**，直接进入数据库代理管理页面。



4. 在数据库代理管理页面中，找到**连接地址**，在**内网访问地址**下找到目标地址，单击 图标。



5. 在弹出的对话框中，修改内网地址后，单击**确定**。

> **注意：**
> 修改代理内网地址会影响正在访问的数据库业务，建议在低峰期修改，请确保业务具备重连机制。



参数	说明
内网地址	您可在内网地址可选段范围自定义内网地址。
自定义端口	您可自定义端口值，取值范围为1024 - 65535。
旧 IP 地址回收时间	回收时间系统默认为0小时，表示修改后会立即回收旧 IP 地址，随即不可通过旧 IP 地址访问数据库代理。此时间支持自定义，范围是：0 - 168小时。

删除数据库代理内网地址

> **说明：**
> 当数据库代理下有多个代理连接地址时，支持删除不需要的连接地址，但只有一个地址时不支持删除。

1. 登录 TDSQL-C MySQL 版控制台，选择实例所属的**地域**。
2. 在集群列表，选择已开启代理的集群，单击集群 ID，进入集群管理页面。
3. 在**集群管理** > **集群详情**下的拓扑图中，鼠标移动到数据库代理板块，然后单击**详情**进入数据库代理管理页面。或在**集群管理**页面直选择**数据库代理**，直接进入数据库代理管理页面。
4. 在数据库代理管理页面中，找到**连接地址**，在**内网访问地址**下找到目标地址，单击其操作列的**关闭**。



5. 在弹出的对话框单击**确定**。

设置单个或多个连接地址

最近更新时间：2025-06-19 11:14:31

本文为您介绍如何通过 TDSQL-C MySQL 版控制台设置单个或多个数据库代理连接地址。
数据库代理访问地址独立于原有的数据库访问地址，通过数据库代理地址的请求全部通过代理集群中转访问数据库的主从节点，进行读写分离，将读请求转发至只读实例，降低主库的负载。

前提条件

已 开通数据库代理 。

设置单个数据库代理连接地址

TDSQL-C MySQL 版在开通数据库代理后默认提供一个数据库代理连接地址，即开通数据库代理后已具备单个数据库代理连接地址，您可对其进行策略配置并挂载只读实例。

1. 登录 TDSQL-C MySQL 版控制台，在集群列表，选择已开启代理的集群，单击**集群 ID**，进入集群管理页面。
2. 在**集群管理 > 集群详情**下的拓扑图中，鼠标移动到数据库代理板块，然后单击**详情**进入数据库代理管理页面。或在集群管理页面直接选择**数据库代理**页进入**数据库代理**管理页面。
3. 在连接地址下找到目标访问地址，单击其操作列的调整配置。



4. 在跳转的窗口下，完成具体策略的配置，单击**确定**。
步骤1：配置策略



参数	说明
读写属性	选择此代理访问地址的读写属性，支持选择读写分离或只读，若选择读写分离，则自动开启读写分离功能。
接入模式	支持均衡分配和就近访问两种接入模式来设置客户端到数据库代理的连接链路，详情请参考 接入模式 。
一致性设置	在读写分离属性下提供了最终一致性、会话一致性和全局一致性三种一致性级别，满足您在不同场景下对一致性的要求，详情请参考 一致性级别 。
连接池状态	连接池功能主要用于减少短连接业务频繁建立新连接带来的实例负载。此项开启，可选择支持的连接池类型，目前默认仅支持会话级连接池。

连接池阈值	设置连接池阈值，可选范围：0-300秒。
事务拆分	设置是否开启，开启后，在一个事务中拆分读和写到不同的实例上去执行，读请求转发到只读实例，降低主实例负载。
故障转移（读写属性为读写分离）	设置是否开启，开启后，数据库代理出现故障时，连接地址将会路由到主实例。
自动添加只读实例	设置是否开启，开启后，若您购买新的只读实例或者只读分析引擎，会自动添加到数据库代理中。 当读权重为系统自动分配时，新购只读实例按照规格大小默认权重分配。 当读权重为自定义时，新购只读实例默认加入时权重为0，可通过数据库代理页，连接地址下的调整配置来修改。

步骤2：配置权重

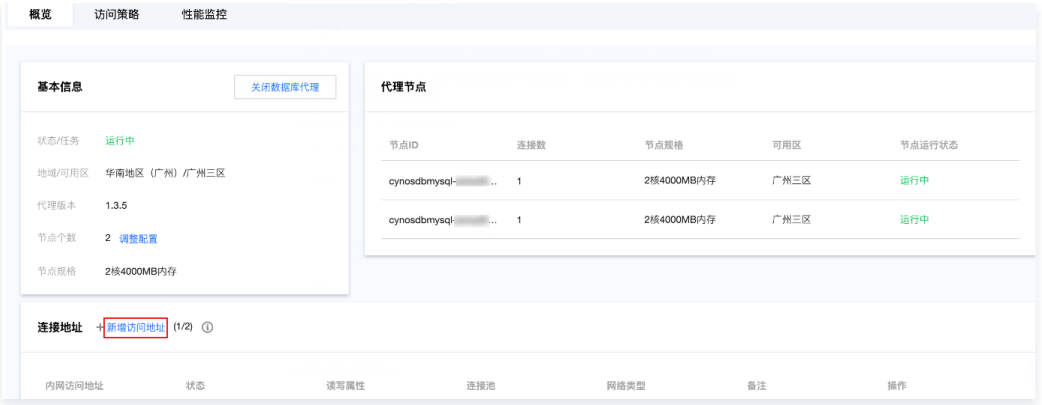


参数	说明
读权重分配	数据库代理需要将读请求分发给不同的实例。通过为各个实例分配不同的读权重，数据库代理可以根据实例的权重值来决定将请求路由到哪个实例。这里支持选择系统自动分配或自定义。 若选择自定义分配，则支持对不同实例访问数据库的权重进行分别配置，以下是读权重分配时，启用和权重相关的说明： • 启用：开启启用按钮，表示所选实例会被分配连接，关闭启用按钮，表示所选实例不会被分配连接。 • 权重：权重值的范围为0 - 100，设置为0表示所选实例会被分配连接且权重最小。数值越大，权重越大，可以接收到更多的读请求。设置完读权重分配，读请求会按比例路由到对应实例。例如分别为实例1、实例2、实例3分配读权重值为3、2、1，表示来源于数据库代理连接地址的读请求，会按照3:2:1的比例分别路由到实例1、实例2、实例3。 **说明：** • 此处的权重是面向读请求（非事务）权重的分配策略。 • 只读分析引擎不支持权重修改，仅支持使用/*to server 实例 ID*/ 的 hint 语法进行强制指定转发执行。

设置多个数据库代理连接地址

说明：
- 数据库代理连接地址的上限为7个。
- 开通数据库代理时已默认创建一个连接地址。

1. 登录 TDSQL-C MySQL 版控制台，在集群列表，选择已开启代理的集群，单击**集群 ID**，进入集群管理页面。
2. 在**集群管理** > **集群详情**下的拓扑图中，鼠标移动到数据库代理板块，然后单击**详情**进入数据库代理管理页面。或在集群管理页面直接选择**数据库代理**页进入数据库代理管理页面。
3. 在数据库代理管理页面，在**连接地址**后单击**新增访问地址**。



4. **在新建连接窗口，完成如下配置，单击确定。**

步骤1：网络配置



参数	说明
网络类型	选择数据库代理的网络，仅支持私有网络 VPC，支持选择自动分配 IP 或指定 IP。
安全组	默认选择的安全组与主实例保持一致，也可根据需要选择已有安全组或新建安全组，支持多选安全组。 **注意：** 访问数据库代理需要开通配置安全策略，放通内网访问端口（当前内网端口为：3306），具体详见 安全组配置 。
备注	非必填项，可为新增数据库代理连接地址进行备注。

步骤2：配置策略



参数	说明
读写属性	选择此代理访问地址的读写属性，支持选择读写分离或只读，若选择读写分离，则自动开启读写分离功能。
接入模式	支持均衡分配和就近访问两种接入模式，来设置客户端到数据库代理的连接链路，详情请参考 接入模式 。
一致性设置	在读写分离属性下提供了最终一致性、会话一致性和全局一致性三种一致性级别，满足您在不同场景下对一致性的要求，详情请参考 一致性级别 。
连接池状态	连接池功能主要用于减少短连接业务频繁建立新连接带来的实例负载。此项开启，可选择支持的连接池类型，目前默认仅支持会话级连接池。
连接池阈值	设置连接池阈值，可选范围：0 – 300秒。
事务拆分	设置是否开启，开启后，在一个事务中拆分读和写到不同的实例上去执行，读请求转发到只读实例，降低主实例负载。
故障转移（读写属性为读写分离）	设置是否开启，开启后，数据库代理出现故障时，连接地址将会路由到主实例。
自动添加只读实例	设置是否开启，开启后，若您购买新的只读实例，会自动添加到数据库代理中。 当读权重为系统自动分配时，新购只读实例按照规格大小默认权重分配。 当读权重为自定义时，新购只读实例默认加入时权重为0，可通过数据库代理页，连接地址下的调整配置来修改。

步骤3：配置权重



参数	说明
读权重分配	数据库代理需要将读请求分发给不同的实例。通过为各个实例分配不同的读权重，数据库代理可以根据实例的权重值来决定将请求路由到哪个实例。这里支持选择系统自动分配或自定义。 若选择自定义分配，则支持对不同实例访问数据库的权重进行分别配置，以下是读权重分配时，启用和权重相关的说明： • 启用：开启启用按钮，表示所选实例会被分配连接，关闭启用按钮，表示所选实例不会被分配连接。

- 权重：权重值的范围为0 - 100，设置为0表示所选实例会被分配连接且权重最小。数值越大，权重越大，可以接收到更多的读请求。设置完读权重分配，读请求会按比例路由到对应实例。例如分别为实例1、实例2、实例3分配读权重值为3、2、1，表示来源于数据库代理连接地址的读请求，会按照3:2:1的比例分别路由到实例1、实例2、实例3。

> **说明：**
> - 此处的权重是面向读请求（非事务）权重的分配策略。
> - 只读分析引擎不支持权重修改，仅支持使用/*to server 实例 ID*/ 的 hint 语法进行强制指定转发执行。

查看和修改访问策略

最近更新时间：2025-01-14 17:27:22

TDSQL-C MySQL 版开启数据库代理后，默认添加一个数据库代理的连接地址，之后您可新增连接地址，通过使用不同的连接地址来实现不同的业务逻辑，可创建连接地址的数量与数据库代理节点数相同。您可通过控制台查看和修改数据库代理连接地址的访问策略。

前提条件

已 开通数据库代理 。

查看访问策略

1. 登录 TDSQL-C MySQL 版控制台，在集群列表，选择已开启代理的集群，单击**集群 ID**，进入集群管理页面。
2. 在**集群管理** > **集群详情**下的拓扑图中，鼠标移动到数据库代理板块，然后单击**详情**进入数据库代理管理页面。或在集群管理页面直接选择**数据库代理**页进入数据库代理管理页面。
3. 在数据库代理管理页，选择**数据库代理** > **访问策略**。



修改访问策略

1. 登录 TDSQL-C MySQL 版控制台，在集群列表，选择已开启代理的集群，单击**集群 ID**，进入集群管理页面。
2. 在**集群管理** > **集群详情**下的拓扑图中，鼠标移动到数据库代理板块，然后单击**详情**进入数据库代理管理页面。或在集群管理页面直接选择**数据库代理**页进入数据库代理管理页面。
3. 在数据库代理管理页，选择**访问策略**，找到目标访问策略，单击**配置调整**。

> **说明：**
> 您也可以在**数据库代理** > **概览** > **连接地址**下找到目标访问地址，单击其**操作**列的**调整配置**。

4. 在跳转的窗口下，修改具体策略的配置，单击**确定**。

步骤1：配置策略



参数	说明
读写属性	选择此代理访问地址的读写属性，支持选择读写分离或只读，若选择读写分离，则自动开启读写分离功能。
接入模式	支持均衡分配和就近访问两种接入模式来设置客户端到数据库代理的连接链路，详情请参考 接入模式 。
一致性设置	在读写分离属性下提供了最终一致性、会话一致性和全局一致性三种一致性级别，满足您在不同场景下对一致性的要求，详情请参考 一致性级别 。
连接池状态	连接池功能主要用于减少短连接业务频繁建立新连接带来的实例负载。此项开启，可选择支持的连接池类型，目前默认仅支持会话级连接池。
连接池阈值	设置连接池阈值，可选范围：0 – 300秒。
事务拆分	设置是否开启，开启后，在一个事务中拆分读和写到不同的实例上去执行，读请求转发到只读实例，降低主实例负载。
故障转移（读写属性为读写分离）	设置是否开启，开启后，数据库代理出现故障时，连接地址将会路由到主实例。
自动添加只读实例	设置是否开启，开启后，若您购买新的只读实例，会自动添加到数据库代理中。 当读权重为系统自动分配时，新购只读实例按照规格大小默认权重分配。 当读权重为自定义时，新购只读实例默认加入时权重为0，可通过数据库代理页，连接地址下的调整配置来修改。

步骤2：配置权重



参数	说明
读权重分配	数据库代理需要将读请求分发给不同的实例。通过为各个实例分配不同的读权重，数据库代理可以根据实例的权重值来决定将请求路由到哪个实例。这里支持选择系统自动分配或自定义。若选择自定义分配，则支持对不同实例访问数据库的权重进行分别配置，以下是读权重分配时，启用和权重相关的说明： • 启用：开启启用按钮，表示所选实例会被分配连接，关闭启用按钮，表示所选实例不会被分配连接。 • 权重：权重值的范围为0 – 100（只读分析引擎实例的权重默认为0且不可修改），设置为0表示所选实例会被分配连接且权重最小。数值越大，权重越大，可以接收到更多的读请求。设置完读权重分配，读请求会按比例路由到对应实例。例如分别为实例1、实例2、实例3分配读权重值为3、2、1，表示来源于数据库代理连接地址的读请求，会按照3:2:1的比例分别路由到实例1、实例2、实例3。 **说明：** • 此处的权重是面向读请求（非事务）权重的分配策略。 • 设置权重的操作不会引起连接闪断。 • 设置不启用可能会引起少量连接受影响，如需避免影响，建议先调整对应实例/节点的读权重为0，再设置不启用。

调整数据库代理配置

最近更新时间：2024-07-02 17:05:31

数据库代理功能支持调整配置，您可通过控制台对数据库代理的代理规格、节点个数进行调整。

前提条件

已 开通数据库代理 。

注意事项

- 若调整配置操作时，数据库代理版本存在更新，则在调整配置过程中数据库代理版本会自动完成升级。
- 若所选数据库代理与主实例不在同一可用区，通过数据库代理连接时，写入性能可能会下降。
- 若计算后所需代理节点数量超过购买限制，建议选择更高规格代理。
  - 代理规格为2核4000MB内存时，推荐的代理个数为主实例和只读实例 CPU 核数之和的1/8（向上取整）。
  - 代理规格为4核8000MB内存时，推荐的代理个数为主实例和只读实例 CPU 核数之和的1/16（向上取整）。
  - 代理规格为8核16000MB内存时，推荐的代理个数为主实例和只读实例 CPU 核数之和的1/32（向上取整）。

影响说明

配置变更会出现不同的影响，部分变更不会导致连接闪断，部分会导致闪断，具体可变更项和对应影响如下表。

场景一：代理规格变更，可用区、节点个数不变。

代理规格	可用区	节点个数	切换时间	影响
升级或降级	不变更	不变更	维护时间内	会有秒级别闪断，请确保业务具备重连机制。
			升级完成时	

场景二：节点个数变更，代理规格、可用区不变。

代理规格	可用区	节点个数	负载均衡方式	切换时间	影响
不变更	不变更	增加	自动	维护时间内	会有秒级别闪断，请确保业务具备重连机制。
				升级完成时	
			手动	维护时间内	不会导致连接闪断。
				升级完成时	
		减少	自动	维护时间内	会有秒级别闪断，请确保业务具备重连机制。
				升级完成时	
			手动	维护时间内	
				升级完成时	

场景三：代理规格不变，可用区、节点个数组合变更。

代理规格	可用区	节点个数	负载均衡方式	切换时间	影响
不变更	增加	增加	自动	维护时间内	会有秒级别闪断，请确保业务具备重连机制。
				升级完成时	
			手动	维护时间内	不会导致连接闪断。
				升级完成时	
		减少	自动	维护时间内	会有秒级别闪断，请确保业务具备重连机制。

				升级完成时
			手动	维护时间内
				升级完成时
	减少	增加	自动	维护时间内
				升级完成时
			手动	维护时间内
				升级完成时
		减少	自动	维护时间内
				升级完成时
			手动	维护时间内
				升级完成时
	更换	增加	自动	维护时间内
				升级完成时
			手动	维护时间内
				升级完成时
		减少	自动	维护时间内
				升级完成时
			手动	维护时间内
				升级完成时

操作步骤

1. 登录 TDSQL-C MySQL 版控制台，在集群列表，选择已开启代理的集群，单击**集群 ID**，进入集群管理页面。
2. 在**集群管理** > **集群详情**下的拓扑图中，鼠标移动到数据库代理板块，然后单击**详情**进入数据库代理管理页面。或在集群管理页面直接选择**数据库代理** tab 页进入数据库代理管理页面。
3. 在数据库代理页的**概览**页，在**基本信息** > **节点个数**后单击**调整配置**。



4. 在弹出的对话框，根据需要修改数据库代理的相关配置（代理规格、可用区及节点个数），单击**确定**。



5. 待**数据库代理**页的**基本信息**下，任务状态由**升级中**变为**运行中/等待升级**，即可完成。

> **说明：**
> - 若切换时间选择在升级完成时切换，则配置调整完成后，系统自动切换为新配置。
> - 若切换时间选择在维护时间内切换，则配置调整完成后，系统会在您设定的维护时间段内进行切换。
> - 若您选择维护时间内切换，但因业务需要提前切换，您可在配置调整完成后，在数据库代理页的**概览** > **基本信息** > **状态 / 任务**升级待切换后单击**立即完成**。
> - 若重新负载均衡选择在调整配置后手动进行，则配置调整完成后，可在数据库代理页的**概览** > **连接地址**下对应手动操作重新负载均衡。

切换数据库代理网络

最近更新时间：2025-02-08 16:27:02

本文为您介绍如何通过 TDSQL-C MySQL 版控制台修改数据库代理的网络。

前提条件

已 开通数据库代理 。

注意事项

- 更换网络会导致该实例数据库代理 IP 变化，最长保留时间支持设置168小时，到设置的时间后，旧的访问 IP 会失效，请及时修改客户端程序。
- 若旧 IP 地址的回收时间设置为0小时，更换网络后会立即回收旧 IP 地址。
- 只能选择 TDSQL-C MySQL 版集群所在地域下的 VPC 网络，但不限制子网可用区的选择，并可查看子网地址范围。

操作步骤

1. 登录 TDSQL-C MySQL 版控制台，在集群列表，选择已开启代理的集群，单击**集群 ID**，进入集群管理页面。
2. 在**集群管理** > **集群详情**下的拓扑图中，鼠标移动到数据库代理板块，然后单击**详情**进入数据库代理管理页面。或在集群管理页面直接选择**数据库代理** tab 页进入数据库代理管理页面。
3. 在数据库代理页的**概览** > **连接地址** > **网络类型**下，单击编辑图标。



4. 在弹出的对话框，选择新网络配置，单击**确定**。
   - 选择网络，可选择集群所在地域和可用区内的 VPC 网络与子网。
   - 旧 IP 地址回收时间，默认为0小时，可设置范围为0 – 168小时。
5. 更换网络成功后，可在**连接地址**下查询变更后的网络。

查看数据库代理监控

最近更新时间：2025-02-13 16:46:02

本文为您介绍如何通过 TDSQL-C MySQL 版控制台查看数据库代理节点监控。

前提条件

已 开通数据库代理 。

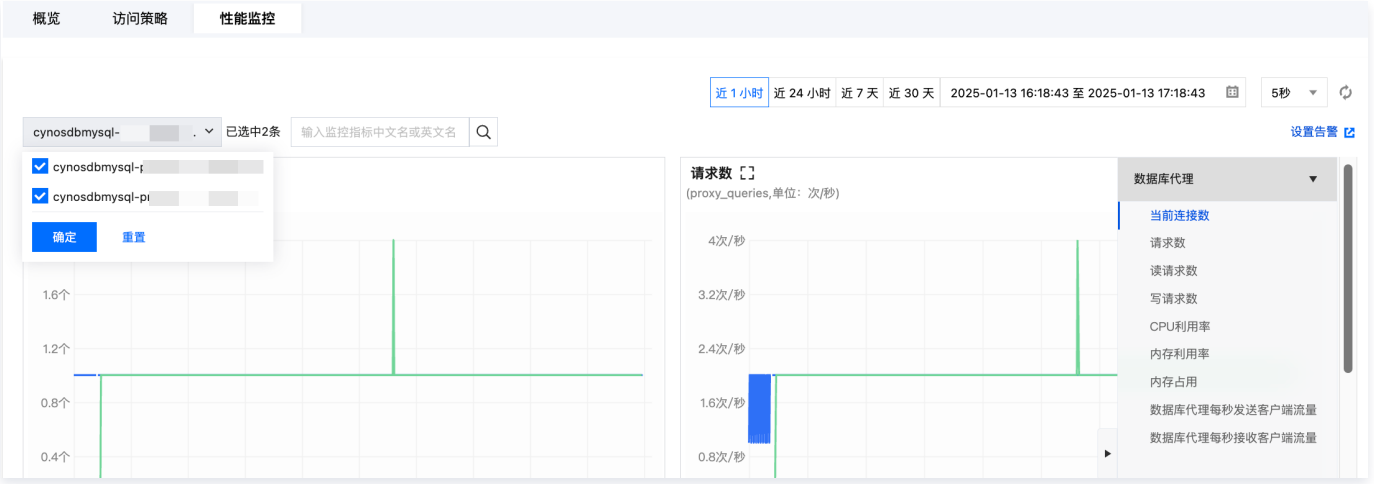
数据库代理节点监控指标

> **说明：**
> 通过 API 调用获取监控数据时，调用参数项使用的监控指标名称请以下表中的“可调用指标名称”列为准。

监控指标名称	可调用指标名称	单位
当前连接数	ProxyCurrentConnections	个
请求数	ProxyQueries	次/秒
读请求数	ProxyRouteSlave	次/秒
写请求数	ProxyRouteMaster	次/秒
CPU 利用率	ProxyCpuUseRate	%
内存利用率	ProxyMemoryUseRate	%
内存占用	ProxyMemoryUsage	MB
数据库代理每秒发送客户端流量	ProxyBytesSent	MB/s
数据库代理每秒接收客户端流量	ProxyBytesReceived	MB/s

操作步骤

1. 登录 TDSQL-C MySQL 版控制台，在集群列表，选择已开启代理的集群，单击**集群 ID**，进入集群管理页面。
2. 在**集群管理** > **集群详情**下的拓扑图中，鼠标移动到数据库代理板块，然后单击**详情**进入数据库代理管理页面。或在集群管理页面直接选择**数据库代理**进入数据库代理管理页面。
3. 在**数据库代理**页选择**性能监控**，单击节点名称可选择查看各代理节点监控。



重新负载均衡

最近更新时间：2025-01-14 17:27:22

开通数据库代理后，您可通过查看代理节点列表的连接数或查看各代理节点的性能监控，来判断各节点是否出现访问不均衡现象。若业务存在大量长连接，增加数据库代理节点也可能会出现新的节点负载不均衡，若出现各代理节点连接数不均衡的现象，可单击重新负载均衡打散连接。本文介绍通过控制台手动操作重新负载均衡。

前提条件

已 开通数据库代理 。

操作步骤

1. 登录 TDSQL-C MySQL 版控制台，在集群列表，选择已开启代理的集群，单击**集群 ID**，进入集群管理页面。
2. 在**集群管理** > **集群详情**下的拓扑图中，鼠标移动到数据库代理板块，然后单击**详情**进入数据库代理管理页面。或在集群管理页面直接选择**数据库代理**页签进入数据库代理管理页面。
3. 在数据库代理管理页选择**概览**，在连接地址下找到目标访问地址，在其**操作**列单击**重新负载均衡**。



4. 在弹出的对话框中单击**确定**。

> **说明：**
> 重新负载均衡操作会导致连接到该地址的会话断开，过程中会有短暂服务不可用现象，建议在低峰期重启服务，请确保业务具备重连机制。

关闭数据库代理

最近更新时间：2025-01-14 17:27:22

本文为您介绍如何通过 TDSQL-C MySQL 版控制台关闭数据库代理。

注意：

- 关闭数据库代理前，请确认您的客户端没有通过数据库代理地址访问数据库，关闭数据库后，产生的数据会被删除，不可恢复。
- 关闭数据库代理后，数据库代理地址会不可用，但读写实例地址不受影响。
- 关闭后如果再次开通，数据库代理连接的地址会改变。

前提条件

已 开通数据库代理 。

操作步骤

1. 登录 TDSQL-C MySQL 版控制台，在集群列表，选择已开启代理的集群，单击**集群 ID**，进入集群管理页面。
2. 在**集群管理** > **集群详情**下的拓扑图中，鼠标移动到数据库代理板块，然后单击**详情**进入数据库代理管理页面。或在集群管理页面直接选择**数据库代理**进入数据库代理管理页面。
3. 在数据库代理页选择**概览** > **基本信息**里的**关闭数据库代理**。



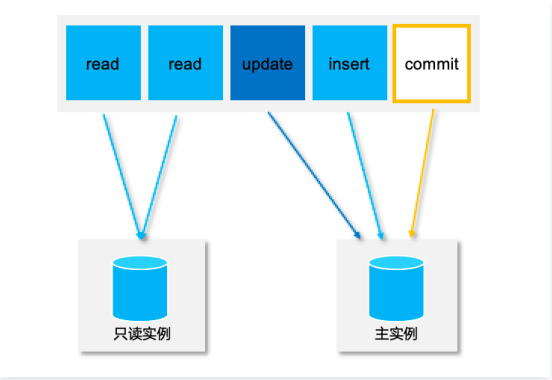
4. 在弹出的对话框，确认无误后，单击**确定**。

事务拆分功能

最近更新时间：2025-06-06 16:29:22

TDSQL-C MySQL 版的数据库代理提供事务拆分功能，能够将一个事务中读和写操作拆分到不同的实例上去执行，读请求转发到只读实例，降低主实例负载。

说明：
- 开启或关闭事务拆分后仅对新连接生效。
- 只读分析引擎无法支持此功能。



背景信息

默认情况下，TDSQL-C MySQL 版数据库代理会将事务内的所有请求都发送到主实例以保障事务的正确性，但是某些框架会将所有请求封装到非自动提交的事务中，导致主实例负载过大。此时您可以使用事务拆分功能。
您可通过对数据库代理访问地址进行配置调整，开启或关闭事务拆分能力。

前提条件

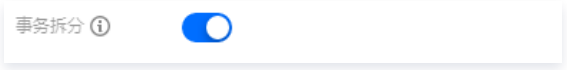
已 开通数据库代理 。

操作步骤

1. 登录 TDSQL-C MySQL 版控制台，在集群列表，选择已开启代理的集群，单击**集群 ID**，进入集群管理页面。
2. 在**集群管理** > **集群详情**下的拓扑图中，鼠标移动到数据库代理板块，然后单击**详情**进入数据库代理管理页面。或在集群管理页面直接选择**数据库代理**页签进入数据库代理管理页面。
3. 在数据库代理管理页，选择**访问策略**，找到目标访问策略，单击**配置调整**。

   说明：
   您也可以在**数据库代理** > **概览** > **连接地址**下找到目标访问地址，单击其**操作**列的**调整配置**。

4. 在调整配置窗口，根据实际需要，将事务拆分后的按钮打开或关闭，单击**确定**。



防闪断功能

最近更新时间：2025-07-02 16:34:42

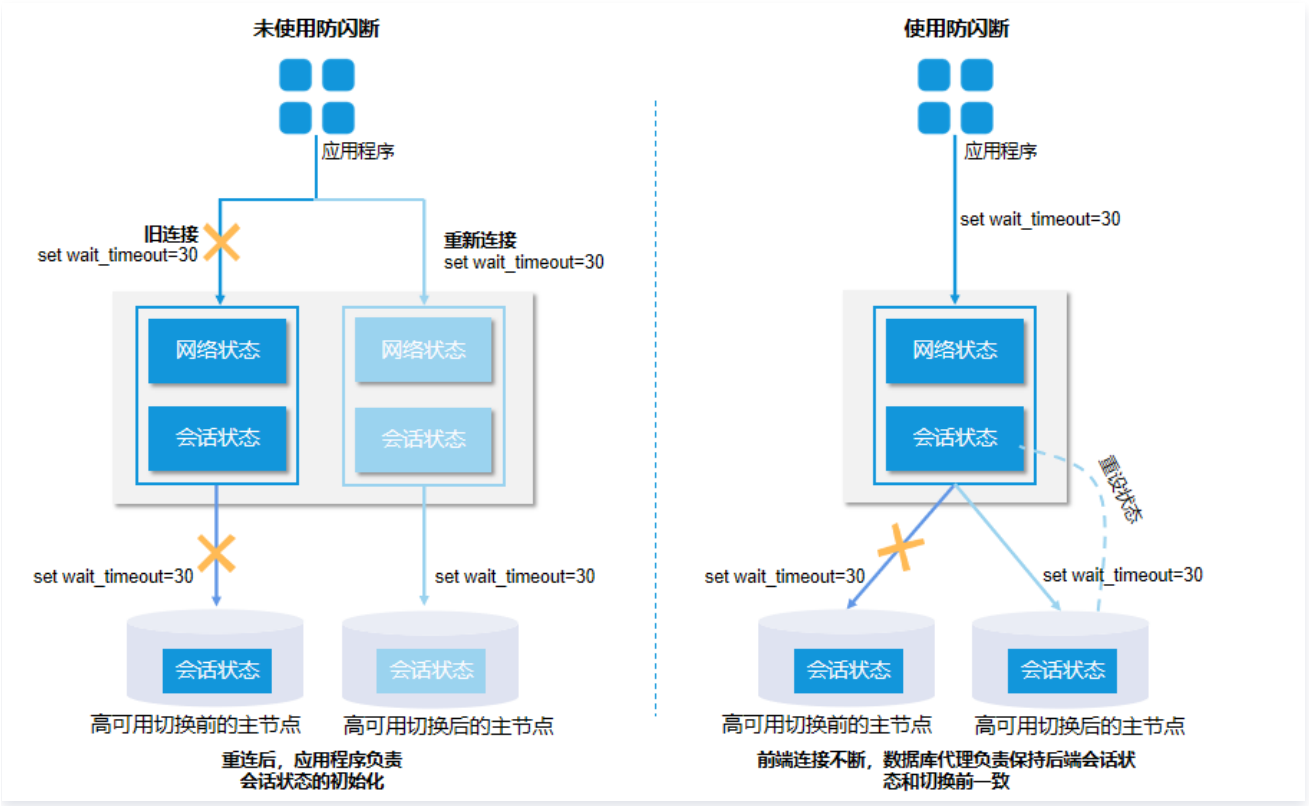
本文介绍 TDSQL-C MySQL 版数据库代理的防闪断功能。

背景信息

数据库实例在运维过程中，总会根据需要，进行相应的调整，如变更配置、计划内 HA 切换、计划内重启等，这些行为可能会中断用户会话，导致连接闪断、新建连接短暂失败等问题。TDSQL-C MySQL 版数据库代理提供防闪断能力，在数据库实例进行有损切换、转移时，可以提供无损的应用连续性，避免连接和事务的中断。

实现原理

防闪断能力，使用了 TDSQL-C MySQL 版的 session track 功能，当感知到计划内的有损行为时，数据库代理会与切换前的主节点断开连接，将客户端到数据库代理上的连接，恢复至切换后主节点的连接上，通过 session track 能力将会话相关的系统变量、用户变量、字符集编码信息转移至新的后端连接上，实现对应用程序端无损切换。



注意事项

- 当语句中使用了临时表（因临时表和每个会话相关联），会无法恢复连接，直接报错。
- 数据库代理具备防闪断能力，需要更新数据库代理内核版本至1.3.1或以上。
- 防闪断能力会自动将超过3秒的事务中止。
- 当连接切换时，如果数据库代理正在从数据库接收结果报文，但是主备切换导致报文仅传输了部分数据，此时，防闪断无法继续保持连接。
- 如果在防闪断期间有正在使用的预编译语句（Prepared Statement），则防闪断无法保持连接。

性能测试

以下为您介绍 TDSQL-C MySQL 版数据库代理防闪断能力的性能测试。

测试环境

- 地域/可用区：北京 – 北京七区。
- 客户端：S5.8XLARGE64(标准型S5，32核64GB)。

- 客户端操作系统：CentOS 8.2 64位。
- 网络：云服务器 CVM 和 TDSQL-C MySQL 版实例网络类型均为私有网络（VPC）且在同一子网。

测试的 TDSQL-C MySQL 版实例信息如下。

- 实例类型：独享型。
- 实例版本：MySQL 8.0。
- 实例规格：4核32GB。
- 参数模板：默认模板。

测试工具

本次性能测试的工具为 sysbench。sysbench 是一个跨平台且支持多线程的模块化基准测试工具，用于评估系统在运行高负载的数据库时相关核心参数的性能表现。可绕过复杂的数据库基准设置，甚至在没有安装数据库的前提下，快速了解数据库系统的性能。

测试方法

在不同运维操作场景中，测试 TDSQL-C MySQL 版实例的连接防闪断的连接保活率（即执行运维操作前后连接未闪断的比例）。

测试结果

在以下运维操作场景中，TDSQL-C MySQL 版实例通过数据库代理的防闪断能力保持了100%的连接保活率。

运维场景	保活率
内核小版本升级	100%
调整实例规格	100%

接入模式

最近更新时间：2024-12-04 20:49:22

本文为您介绍 TDSQL-C MySQL 版数据库代理的接入模式。

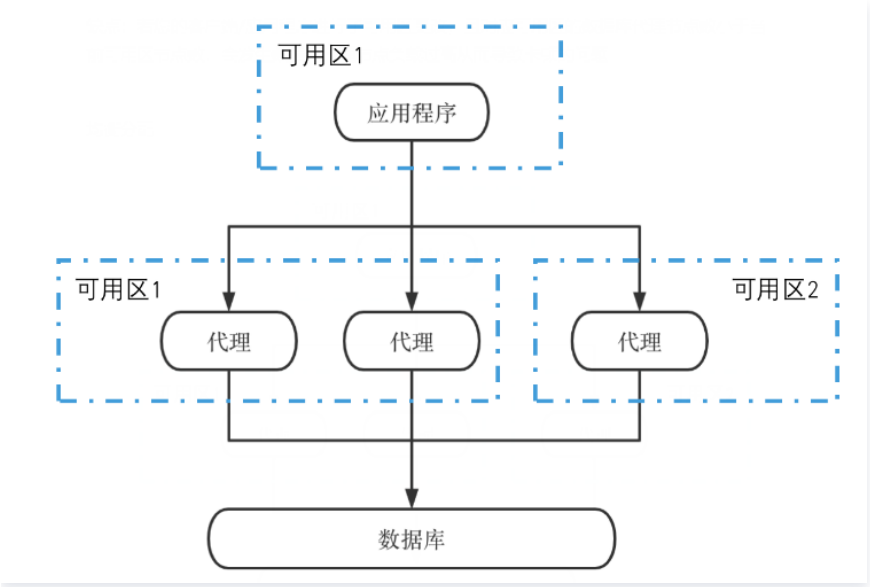
前提条件

已 开通数据库代理 。

简介

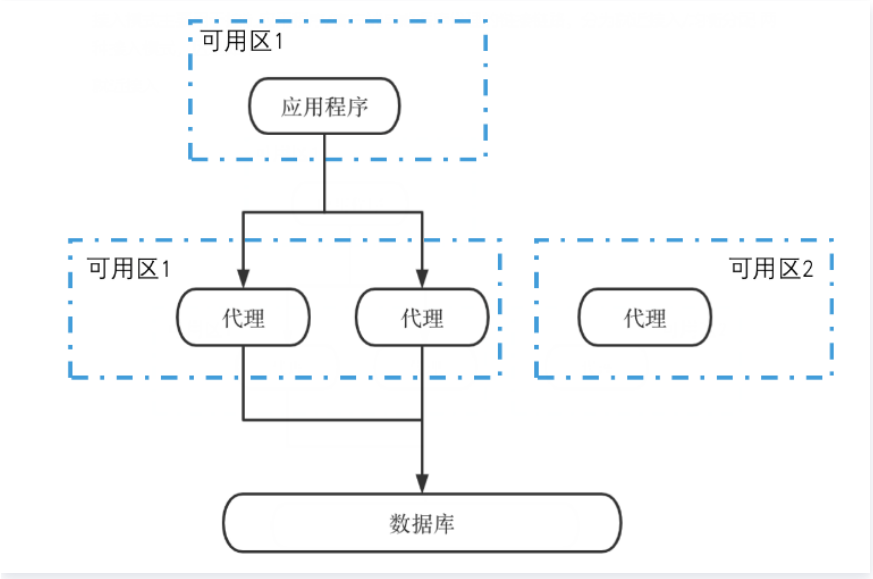
接入模式主要用于控制应用程序或客户端与数据库代理的连接链路，分为**均衡分配**、**就近接入**两种接入模式，以下分别为您介绍两种接入模式的规则和优缺点。

均衡分配



- **规则**：均衡分配接入模式下，应用程序将与所有的数据库代理节点连接，若数据库代理存在多个可用区节点，则从应用程序到数据库代理的所有连接会被均衡的分配到所有可用区的数据库代理节点上来访问数据库。
- **优点**：流量均衡分配，不存在单一节点负载过高的问题。
- **缺点**：若存在不同可用区的代理节点，容易出现访问路径较远导致请求延迟过高的问题。

就近接入



- **规则**：就近接入模式下，应用程序将与同一可用区或访问路径最近的数据库代理节点连接，若数据库代理存在多个可用区节点，应用程序仍然选择访问路径最近的数据库代理节点进行连接。
- **优点**：访问延迟低，速度快。
- **缺点**：若您的应用程序或客户端进行了可用区更改，且更改后可用区的数据库代理节点数小于更改前可用区的数据库代理节点数，则会出现数据库代理节点负载过高导致卡顿的问题。

修改接入模式

在修改访问策略时可以对接入模式进行修改，详细步骤您可参见 查看和修改访问策略 。

实例故障转发

最近更新时间：2025-04-22 16:40:32

当加入到数据库代理中的实例因异常导致无法使用时，数据库代理将通过实例故障转发策略将 SQL 转发到其他实例执行，或者丢弃 SQL 并输出报错信息。本文为您介绍实例故障转发策略的设置和原理。

背景信息

数据库实例在实际运行过程中可能会遇到各类意外导致的实例不可用，例如：服务器断电、重启、网络中断等。TDSQL-C MySQL 版数据库代理提供实例故障转发能力，避免在实例出现异常时，用户的 SQL 执行出错导致业务不可使用的情况发生。

前提条件

已 开通数据库代理 。

转发策略

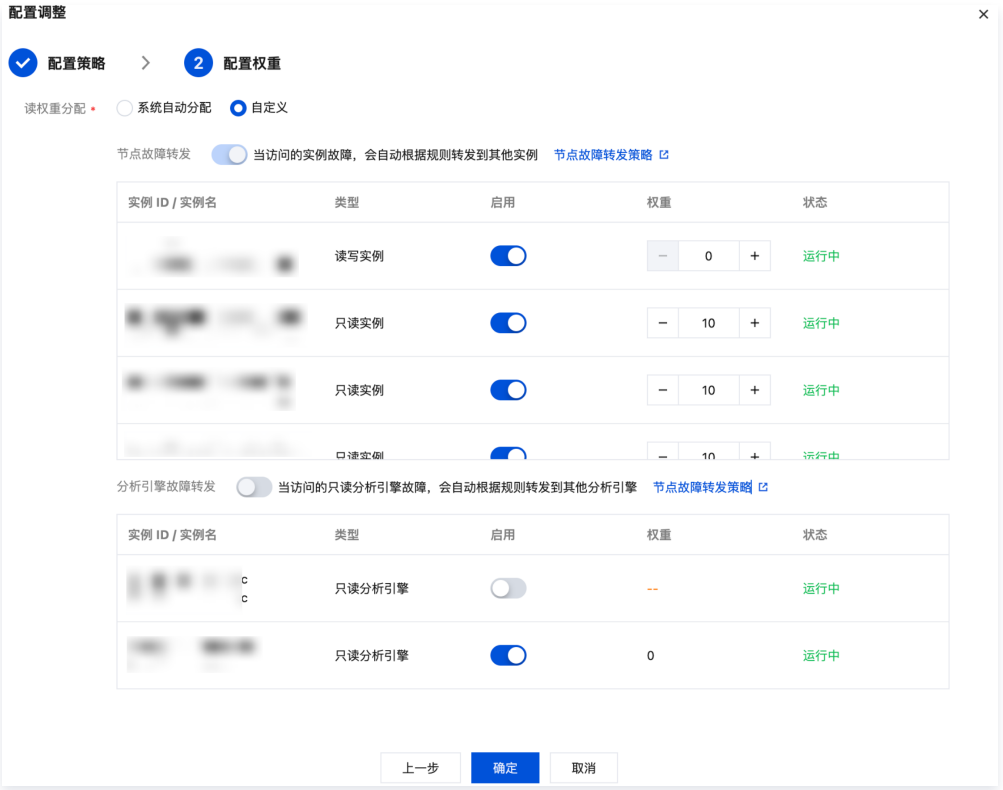
当前数据库代理支持的实例故障转发策略如下：

- 当**只读实例**与**读写实例**出现故障时，数据库代理会默认将 SELECT 执行语句发送至正常的实例中执行。
- 当**读写实例**出现故障时，数据库代理会将 DML 或 DDL 语句执行报错。
- 当数据库代理下的**只读分析引擎**出现故障时，数据库代理会将 SQL 转发至其他正常的只读分析引擎中执行。
- 当数据库代理下的所有**只读分析引擎**全部出现故障时，系统将会返回执行报错。提示“no backend server”的报错信息。

> **注意：**
> 以上实例故障转发策略暂时不支持修改。只读实例与读写实例故障转发参数默认为打开状态，只读分析引擎故障转发参数默认为关闭状态。

操作步骤

1. 登录 TDSQL-C MySQL 版控制台，在集群列表，选择已开启数据库代理的集群，单击**集群 ID**，进入集群管理页面。
2. 在**集群管理 > 集群详情**下的拓扑图中，鼠标移动到**数据库代理**板块，然后单击**详情**进入数据库代理管理页面。或在集群管理页面直接选择**数据库代理**页签进入数据库代理管理页面。
3. 在数据库代理管理页选择**概览**，在连接地址下找到需要修改的访问地址，在其**操作**列中单击**调整配置**，进入到**调整配置**页面，单击**下一步**即可查看不同类型的实例故障转发策略。



自动读写分离功能
自动读写分离介绍

最近更新时间：2025-01-14 14:36:12

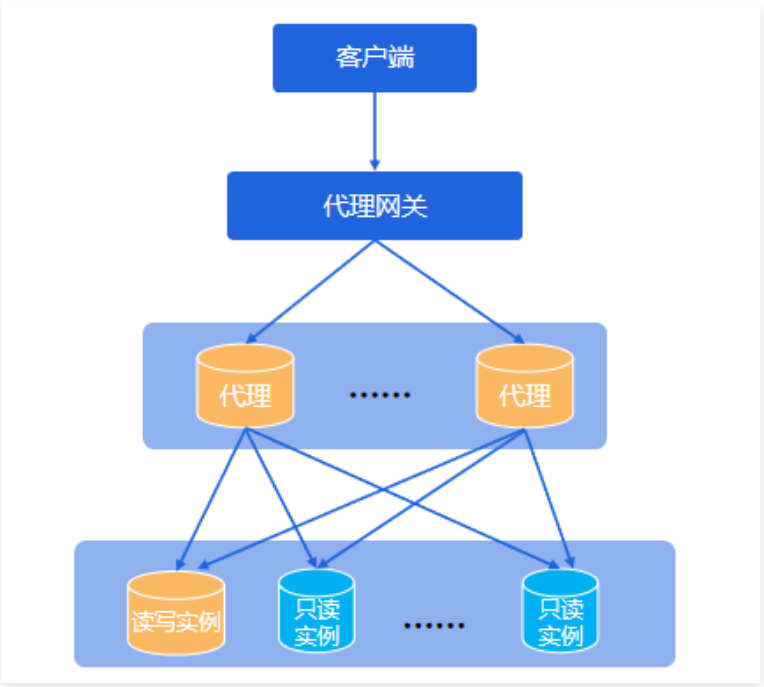
版权所有：腾讯云计算（北京）有限责任公司

本文主要介绍 TDSQL-C MySQL 版数据库代理服务的自动读写分离、功能优势、读写分离路由规则。

自动读写分离

目前，大量现网用户的业务场景中存在读多写少、业务负载无法预测等情况，在有大量读请求的应用场景下，单个实例可能无法承受读取压力，甚至会对业务产生影响。为了实现读取能力的弹性扩展，分担数据库压力，可以创建一个或多个只读实例，利用只读实例满足大量的数据库读取需求。但此类解决方案需要业务侧支持读写分离改造，其代码的健壮性决定了业务读写分离的质量，对用户的技术要求较高，而且灵活性和可扩展性较差。
故创建只读实例后，可以通过开通数据库代理，配置访问地址策略，然后在应用程序中配置数据库代理地址，使写请求自动转发到主实例，读请求自动转发到各个只读实例。数据库代理除了具备自动读写分离功能，也能为其他业务痛点提供更好的解决方案，具体如下：

- **负载无法预测或波动不规律/负载“洪峰”现象明显的场景**
  在互联网业务场景中，业务负载和访问压力往往是不可预估、不平稳的，会出现频繁的波动和“洪峰”。如果业务侧大量使用短连接访问数据库，那么在这些场景中很容易产生大量的新建连接，即数据库与应用程序间的连接数，容易随着业务访问压力的频繁变动而波动。
  数据库代理的连接管理，使用户可以通过有效重用数据库连接，来适当扩展处理不可预测工作负载的应用程序连接问题。首先，使多个应用程序连接可以共享数据库连接，以有效利用数据库资源；其次，允许用户通过调节打开的数据库连接数，来保持数据库性能；最后，可删除无法使用的应用程序请求，以保留应用程序的整体性能和可用性。
- **应用程序与数据库频繁建立和断开连接的场景**
  基于无服务器、PHP 或 Ruby on Rails 等技术构建的应用程序，可能会频繁打开和关闭数据库连接来满足应用程序请求。
  数据库代理可以帮助用户维护数据库连接池，以避免对数据库计算和用于建立新连接的内存造成不必要的压力。
- **数据库访问连接长时间处于空闲状态并未释放的场景**
  SaaS 或电子商务行业中的传统应用程序，可能会使数据库连接保持空闲状态，以最大程度减少用户重新参与的响应时间。用户可以使用数据库代理来保留空闲连接，仅根据需要建立数据库连接，而不是为支持大多数空闲连接过度调大阈值或升级为更高配的数据库。
- **提高数据库 PaaS 服务故障转移场景下的平滑和稳定**
  使用数据库代理，用户可以构建可透明容忍数据库故障转移（主动/被动）的应用程序，而无需编写复杂的故障处理代码。数据库代理自动将读流量路由到新的数据库实例，同时保留应用程序连接。



功能优势

- 自助读写分离，统一访问地址。
- 原生链路支持，提升性能，减少维护成本。
- 可设权重和阈值，可供用户灵活选择。
- 具备故障转移能力，即使数据库代理故障请求也能正常访问主库。
- 读写实例发生切换、变配、只读实例增减等情况，数据库代理可动态热加载配置，不会出现网络中断或重启。

功能说明

TDSQL-C MySQL 版数据库代理服务具备自动读写分离能力，并支持为读写实例和多个只读实例做读权重的分配，需注意，此处的权重是面向读请求（非事务）权重的分配策略。

读写分离路由规则

发送到主实例

- CREATE、ALTER、DROP、RENAME 等 DDL 语句。
- INSERT、UPDATE、DELETE 等 DML 语句。
- SELECT FOR UPDATE 语句。
- 临时表相关语句。
- 部分系统函数调用（如 last_insert_id()）以及所有自定义函数调用。
- LOCK 相关语句。
- 开启事务后的语句（包括 set autocommit=0）。
- 存储过程。
- 用“;”连接的多语句。
- KILL（SQL 语句中的 KILL，非命令 KILL）。
- 所有对用户变量的查询和更改。

发送到只读实例

事务外的读（SELECT）语句。

> **注意：**
> 只读分析引擎不支持读写分离。

发送到所有实例

- show processlist 语句。
- 所有系统变量的更改（SET 命令）。
- USE 命令。

设置数据库代理读写属性

最近更新时间：2024-07-02 17:05:32

创建只读实例后，通过购买数据库代理，配置连接地址策略，在应用程序中配置数据库代理地址，就可以使写请求自动转发到主实例，读请求自动转发到各个只读实例。本文介绍通过控制台开启读写分离。

前提条件

- 实例为读写实例。
- 已 开通数据库代理 。
- 已 创建只读实例 。

操作步骤

1. 登录 TDSQL-C MySQL 版控制台，在集群列表，选择已开启代理的集群，单击**集群 ID**，进入集群管理页面。
2. 在**集群管理** > **集群详情**下的拓扑图中，鼠标移动到数据库代理板块，然后单击**详情**进入数据库代理管理页面。或在集群管理页面直接选择**数据库代理**页进入数据库代理管理页面。
3. 选择**概览**，在连接地址下找到目标访问地址，在其操作列单击**调整配置**。



4. 在调整配置页面，选择对此访问连接的读写属性，根据实际需要完成其他设置和读权重分配，单击**确定**。





说明：
此处的权重是面向读请求（非事务）权重的分配策略。

一致性级别

最近更新时间：2024-12-17 14:09:42

TDSQL-C MySQL 版的自动读写分离功能，会在 TDSQL-C MySQL 版与应用程序间建立好连接，进而解析发送进来的每一条 SQL，如果是 CREATE、ALTER、DROP、RENAME 等语句则直接发往读写实例，如果是事务外的读（SELECT）语句则发送到只读实例，从而实现了读写分离。但是当数据库负载很高，如大批量插入数据的时候，延迟会非常严重，导致无法从只读节点中读取最新数据。
当读写实例有数据更新后，相关的更新会应用到只读实例，其中数据同步的延迟时间与写入压力有关。TDSQL-C MySQL 版通过提供不同的一致性级别，来保证业务访问数据库的数据一致性要求。
TDSQL-C MySQL 版提供了以下三种一致性级别，满足您在不同场景下对一致性的要求：

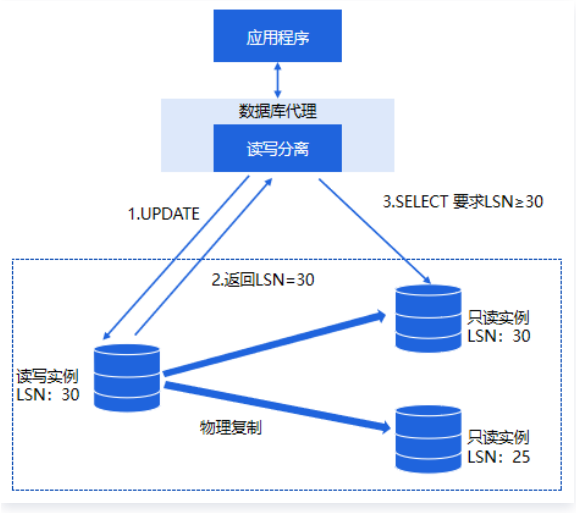
- 最终一致性
- 会话一致性
- 全局一致性

最终一致性

功能简介
TDSQL-C MySQL 版的数据库代理实现了自动读写分离功能，在自动读写分离场景下默认提供数据的最终一致性，从而保证只读实例对已改变写的数据的读取，最终都能取得已更新的数据，但不完全保证能立即取得。已更新的数据主从复制延迟会导致从不同节点查询到的结果不同。
适用场景
如需要减轻读写实例压力，让尽量多的读请求路由到只读实例，对一致性要求不是很高的场景您可以选择最终一致性。

会话一致性

功能简介
有些场景要求一致性较高，而最终一致性会导致查询的结果有所不同，通常需要对业务进行拆分，将一致性要求高的请求直接发往读写实例，而可以接受最终一致性的请求则通过读写分离发往只读实例。这既会增加读写实例的压力，影响读写分离的效果，又会增加应用开发的负担。
为解决上述问题，TDSQL-C MySQL 版提供会话一致性。会话一致性保证了同一个会话内，一定能够查询到读请求执行前已更新的数据，确保了数据单调性。
在 TDSQL-C MySQL 版的链路中间层做读写分离的同时，中间层会追溯各个节点已经应用的日志位点，即日志序号（Log Sequence Number，简称 LSN）。同时每次数据更新时 TDSQL-C MySQL 版会记录此次更新的位点为 Session LSN。当有新请求到来时， TDSQL-C MySQL 版会比较 Session LSN 和当前各个实例的 LSN，仅将请求发往 LSN 大于或等于 Session LSN 的实例，从而保证了会话一致性。



在上述场景中，当更新完成后，返回客户端结果时复制也在同步进行，而当下一个读请求到来时，读写实例和只读实例之间的数据复制有可能已经完成。且大多数应用场景都是读多写少，所以在该机制下保证了会话一致性。
适用场景
适用于对一致性有较高要求的场景，TDSQL-C MySQL 版的一致性级别越高，对主库的压力越大，集群性能也越低。推荐使用会话一致性，该级别对性能影响很小而且能满足绝大多数应用场景的需求。

> **注意：**
> 开启会话一致性后，如果读写实例与只读实例复制时延较大，各个读节点 LSN 均小于 Session LSN，会导致 SELECT 请求发送到读写实例，从而增大读写实例的压力，整个集群的读写性能会有一定的降低。

全局一致性

功能简介

某些场景对一致性要求极高，除了会话内部有逻辑上的因果依赖关系，会话之间也存在依赖关系，例如在使用连接池的场景下，同一个线程的请求有可能通过不同连接发送出去。对数据库来说这些请求属于不同会话，但是业务逻辑上这些请求有前后依赖关系，此时会话一致性便无法保证查询结果的一致性。因此 TDSQL-C MySQL 版提供了全局一致性来解决该问题。

适用场景

适用于对一致性要求极高的场景，当主从延迟较高时，使用全局一致性可能会导致更多的请求被路由到读写实例，造成读写实例压力增大，业务延迟也可能增加。因此建议在读多写少的场景下选择全局一致性。

设置一致性级别

> **说明：**
> 设置一致性级别需先设置数据库代理的读写属性为读写分离，在 设置数据库代理读写属性 的步骤中即可设置会话一致性级别，如需修改一致性级别，请参考以下步骤。

1. 登录 TDSQL-C MySQL 版控制台，在集群列表，选择已开启代理的集群，单击**集群 ID**，进入集群管理页面。
2. 在**集群管理 > 集群详情**下的拓扑图中，鼠标移动到数据库代理板块，然后单击**详情**进入数据库代理管理页面。或在集群管理页面直接选择**数据库代理**页进入数据库代理管理页面。
3. 在连接地址下找到目标访问地址，在其操作列单击**调整配置**。



4. 在配置窗口下，选择所需的一致性级别，单击**确定**。



> **说明：**
> - 若一致性设置为会话一致性或全局一致性，您需对读写实例和只读实例之间的数据复制设置超时时间，取值范围：0 – 1000000毫秒，设置为0，则表示若只读实例出现延迟，导致一致性策略不满足，请求将一直等待，设置为非0，如设置1000毫秒，则表示若只读实例出现延迟，超过设置的1000毫秒后，请求将由系统自动转发到主实例。
> - 温馨提示，当前同一云账号下多个数据库代理实例不支持单独设置各自的超时时间，多个数据库代理实例的超时时间只能统一生效一份，单个数据库代理实例修改超时时间，会导致多个数据库代理实例的超时时间都统一修改。

连接池功能
连接池概述

最近更新时间：2024-10-12 16:33:21

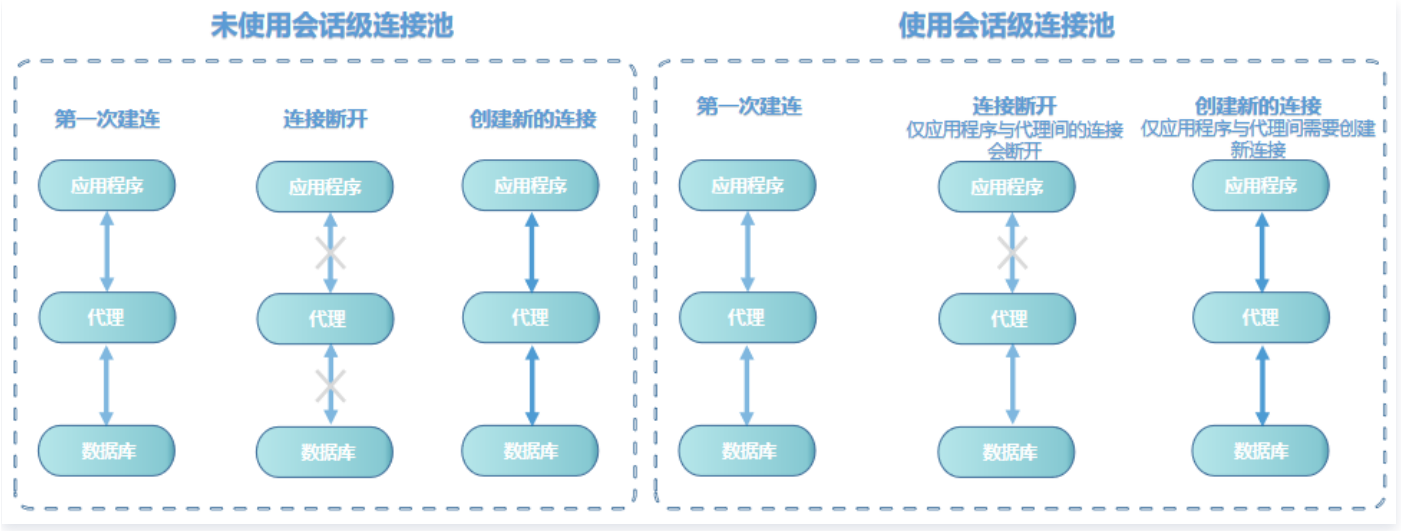
TDSQL-C MySQL 版数据库代理支持连接池功能，目前支持的数据库代理连接池功能为会话级连接池，能有效解决短连接业务频繁建立新连接导致实例负载过高的问题。本文为您介绍会话级连接池功能。

前提条件

已 开通数据库代理 。

背景信息

会话级连接池



会话级连接池适用于短连接场景。

会话级连接池主要用于减少短连接业务频繁建立新连接带来的实例负载。当某客户端连接断开时，系统会判断当前连接是否为闲置连接，如果是闲置连接，系统会将该连接放到代理的连接池中并保留一小段时间（系统默认为5秒，支持 设置连接保留阈值 ）。

当客户端重新发起新连接时，若连接池中有可用的连接，则可直接使用，从而减少与数据库的建连开销。如果连接池内没有可用的闲置连接，则走正常连接流程，与数据库重新建立新连接。

> **说明**
> - 会话级连接池并不能减少数据库的并发连接数，而是通过降低应用与数据库建立连接的频率来减少 TDSQL-C MySQL 版主线程的开销，更好地处理业务请求。但连接池里的闲置连接会短暂占用您的连接数。
> - 会话级连接池不能解决由于存在大量慢 SQL，导致的连接堆积问题，您需要先解决慢 SQL 问题。

注意事项

- 目前连接池功能不支持同一账号对不同 IP 有不同权限，这可能会导致连接复用时权限出错。例如 mt@test123 设置了 database_a 的权限，而 mt@test456 没有 database_a 的权限，开通连接池可能会导致权限错误问题。
- 连接池功能是指数据库代理的连接池功能，不影响客户端的连接池功能，如果您的客户端已经支持连接池，则可以不使用数据库代理的连接池功能。

设置会话级连接池

最近更新时间：2025-01-14 17:27:22

本文为您介绍如何设置会话级连接池功能。

开启连接池功能

1. 登录 TDSQL-C MySQL 版控制台，在集群列表，选择已开启代理的集群，单击**集群 ID**，进入集群管理页面。
2. 在**集群管理** > **集群详情**下的拓扑图中，鼠标移动到数据库代理板块，然后单击**详情**进入数据库代理管理页面。或在集群管理页面直接选择**数据库代理**页进入数据库代理管理页面。
3. 在数据库代理管理页，选择**访问策略**，找到目标访问策略，单击**配置调整**。

> **说明：**
> 您也可以在**数据库代理** > **概览** > **连接地址**下找到目标访问地址，单击其**操作**列的**调整配置**。

4. 在跳转的窗口下，打开连接池状态后的按钮，根据实际需要完成其他项的设置后单击**确定**。



- **连接池状态**：设置开启连接池。
- **连接池类型**：目前支持会话级连接池。
- **连接保留阈值**：闲置连接放到代理的连接池中保留的时间阈值，设置范围为：0秒 – 300秒。

关闭连接池功能

1. 登录 TDSQL-C MySQL 版控制台，在集群列表，选择已开启代理的集群，单击**集群 ID**，进入集群管理页面。
2. 在**集群管理** > **集群详情**下的拓扑图中，鼠标移动到数据库代理板块，然后单击**详情**进入数据库代理管理页面。或在集群管理页面直接选择**数据库代理**页进入数据库代理管理页面。
3. 在数据库代理管理页，选择**访问策略**，找到目标访问策略，单击**配置调整**。

> **说明：**
> 您也可以在**数据库代理** > **概览** > **连接地址**下找到目标访问地址，单击其**操作**列的**调整配置**。

4. 在跳转的窗口下，关闭连接池状态后的按钮，根据实际需要完成其他项的设置后单击**确定**。

其他功能
Hint 语法使用

最近更新时间：2025-06-20 15:27:01

本文主要介绍在数据库代理上如何使用 Hint 语法。
使用 Hint 语法可以强制 SQL 请求在指定的实例上执行，Hint 的路由优先级最高，Hint 不受一致性、事务的约束，使用前请合理评估业务场景是否需要。

> **注意：**
> - 使用 TDSQL-C MySQL 版命令行进行连接并使用 Hint 语句时，需要在命令中增加 -c 选项，否则 Hint 会被命令行工具过滤。
>   例如（通过内网建立连接时）：为 `mysql -h hostname -u username -p -c` 。
> - 通过数据库代理使用 Hint 语法时，数据库代理版本大于等于1.3.3的版本，支持 prepare。

目前支持四种 Hint：

- 指定到读写实例执行：

```
-- Hint语句
/* to master */
或
/*FORCE_MASTER*/
```

```
-- 示例
-- 强制在读写实例上执行查询
SELECT /* to master */ * FROM users WHERE user_id = 1;

-- 强制在读写实例上执行更新
UPDATE /* FORCE_MASTER */ orders SET status = 'shipped' WHERE order_id = 1001;
```

- 指定到只读实例执行：

```
-- Hint语句
/* to slave */
或
/*FORCE_SLAVE*/
```

```
-- 示例
-- 强制在只读实例执行查询
SELECT /* to slave */ * FROM products WHERE category = 'electronics';

-- 强制在只读实例执行计数查询
SELECT /* FORCE_SLAVE */ COUNT(*) FROM transactions WHERE status = 'completed';
```

- 指定到只读分析引擎执行：

> **注意：**
> 只有数据库代理版本为1.4.4及以上的版本才支持此执行方式。如果存在多个只读分析引擎，则将基于权重规则负载均衡。

```
-- 强制在只读分析引擎执行查询
SELECT /* to ap */ * FROM products WHERE category = 'electronics';
```

- 指定某个具体实例执行：

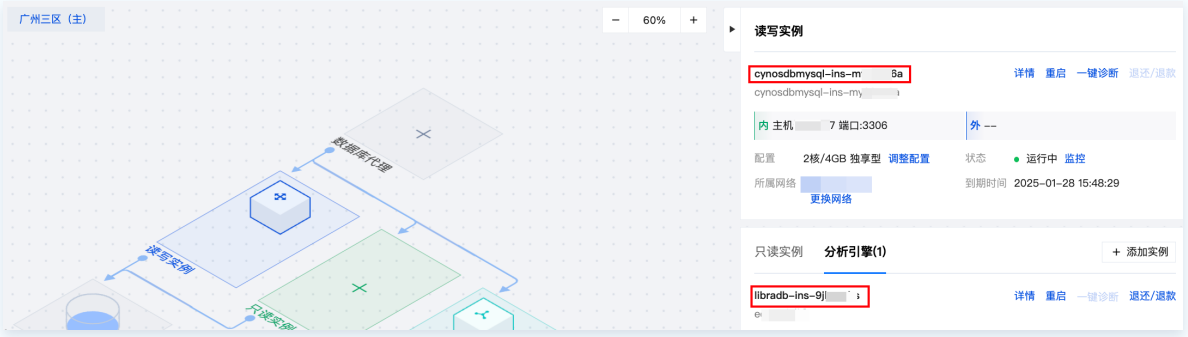
```
-- Hint语句
```

```
/* to server server_name */
```

```
-- 示例
-- 强制在特定实例 test_ro_1 上执行查询
SELECT /* to server test_ro_1 */ * FROM inventory WHERE product_id = 2002;
```

说明：

上述语句中的 server_name 为实例 ID，您可在集群详情页面右侧实例信息内（页签视图模式），获取目标实例的实例 ID，如下图所示。



账号管理
账号简介

最近更新时间：2024-05-13 15:05:51

在使用 TDSQL-C MySQL 版之前，您需要具备2个账号，腾讯云控制台账号和数据库账号，分别用于访问控制台和您的数据库。本文为您介绍腾讯云控制台账号以及数据库账号相关信息。

腾讯云控制台云账号

在开始使用腾讯云服务之前，您需要先注册一个腾讯云账号。拥有腾讯云账号后，您可以在腾讯云网站、控制台登录。详细步骤请参见 注册腾讯云账号 。

数据库账号

TDSQL-C MySQL 版集群创建之后，系统会保留默认的账号，除此之外，为方便分配管理数据库，您也可以根据业务需要创建除了默认账号以外的业务账号。

默认账号说明

系统默认账号	主机	MySQL 5.7	MySQL 8.0	说明
root	%	✓	✓	管理员账号，具有所有权限，可执行所有操作。
mysql.sys	localhost	✓	✓	用于管理和访问系统自带 sys 库的账号，通过 sys 库可以快速了解系统的元数据信息。
mysql.session	localhost	-	✓	内部插件访问服务器的账号。
mysql.infoschema	localhost	-	✓	用于管理和访问系统自带 information_schema 库的账号。

警告：
为避免造成数据库问题，建议不要删除默认账号。

非默认账号

除系统默认创建的账号外，您可根据业务的需要，通过控制台创建其他的业务账号。详细请参见 创建账号 。

控制台账号
注册腾讯云账号

最近更新时间：2024-12-04 20:49:22

在开始使用腾讯云服务之前，您需要先注册一个腾讯云账号。拥有账号后，您可以在腾讯云网站、控制台登录，从而选购和使用 TDSQL-C MySQL 版等产品。

步骤1：前往注册页面

单击前往 注册页面 。

步骤2：注册账号

腾讯云注册方式包括以下几种：

注册方式	描述
微信扫码快速注册	使用您的微信扫码快速注册腾讯云，后续您可以使用微信扫码登录腾讯云。
邮箱注册	使用邮箱注册腾讯云，方便企业客户维护账号。
QQ 注册	使用您已有的账号注册腾讯云，直接用账号快捷登录腾讯云。
微信公众号注册	使用您已有的公众号注册腾讯云。
企业微信注册	使用您已有的企业微信注册腾讯云。

如需了解详细的注册方法，请参见 注册腾讯云 。

后续操作

实名认证

实名认证

最近更新时间：2024-10-12 10:13:52

当您完成腾讯云账号注册后，您需要完成实名认证或完善行业信息。

> **警告**
> 实名认证直接影响账号和资源的归属，如果企业用户使用个人信息进行实名认证，后续出现人员变动或账号纠纷时，可能会影响企业用户的业务，甚至造成经济损失。在进行实名认证之前，请确认您在腾讯云购买和使用的资源是属于个人还是企业。

认证方式

- 个人实名认证
- 企业实名认证

数据库账号
创建账号

最近更新时间：2025-05-26 09:28:11

除系统默认创建的账号外，您可以根据业务的需要，通过控制台创建其他的业务账号。

操作步骤

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在上方选择地域，根据实际控制台视图模式，对应操作。

页签视图

在左侧集群列表，单击**目标集群**，进入集群管理页。



列表视图

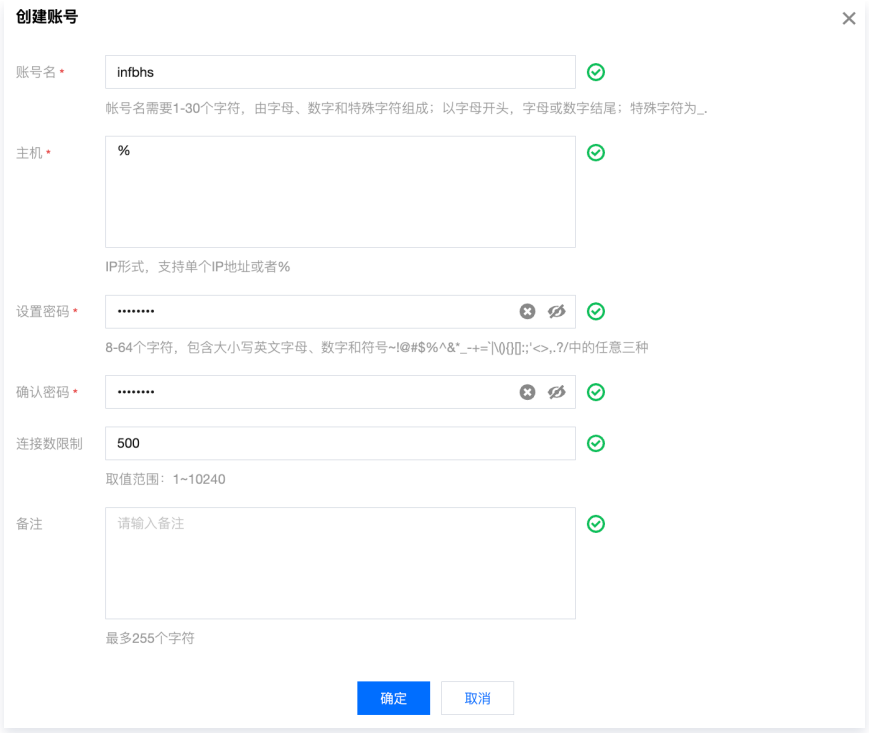
在左侧集群列表找到目标集群，单击**集群 ID** 或**操作**列的**管理**，进入集群管理页。



3. 在集群管理页面，选择**账号管理**页，单击**创建账号**。



4. 在创建账号窗口下，设置如下信息配置，单击**确定**。



参数	说明
账号名	数据库账号名需要1个 - 30个字符，由字母、数字和特殊字符组成；以字母开头，字母或数字结尾；特殊字符为_
主机	指定主机地址访问数据库，支持 IP 形式，也支持填入%（表示不做 IP 范围限制） 示例1：填入%，表示不做 IP 范围限制，即允许所有 IP 地址的客户端使用该账号访问数据库 示例2：填入10.5.10.%，表示允许 IP 范围在10.5.10.%内的客户端使用该账号访问数据库
设置密码	密码长度范围为8个 - 64个字符，包含大小写英文字母、数字和符号~!@#$%^&*_-+=`|\(){}[]:;'<>,.?/中的任意三种
确认密码	再次输入账号密码，两次密码保持一致
连接数限制	输入限制该账号的连接数限制，不填写则不进行额外限制（均会受到最大连接数限制），取值范围：1 - 10240
备注	为账号添加备注，最多可输入255个字符

5. 数据库账号创建成功后，可在账号列表中，对其进行管理（修改权限、重置密码、修改账号设置等）。

> **注意：**
> 开启只读分析引擎后，系统会默认为数据库创建一个名为“txsqlPlugins_analytics_repl”的账号，此账号为只读分析引擎从读写实例拉取数据使用，请不要删除此账号。

自定义密码强度
简介

最近更新时间：2024-10-12 15:42:51

本文为您介绍 TDSQL-C MySQL 版自定义密码强度功能。

功能概述

密码是数据库安全最重要的一道防线，随着国家等保三级要求的推出，数据库对密码强度的要求也越来越高。TDSQL-C MySQL 版支持自定义密码强度功能，保障数据库的安全性，满足企业等保安全规范要求。
您可以通过控制台设置该功能，对控制台以及数据库所有与密码相关的操作进行强度限制，保护用户的密码安全，预防密码泄露等安全隐患，自定义密码强度功能支持的设置项如下。

支持功能设置	说明
大小写英文字母对数	默认值为1，取值范围1 - 50。
数字个数	默认值为1，取值范围1 - 50。
特殊字符个数	默认值为1，取值范围1 - 50。
最小密码长度	默认值为8，取值范围8 - 256。
违禁词字典	当密码检查强度等级为 STRONG 时，此选项可以设置，违规词内容输入要求：单词长度4 - 100，须为英文大小写字母。
密码检查强度等级	支持选择检查强度为 MEDIUM 或 STRONG。 • MEDIUM：该设置下插件检查长度、数字、大小写、特殊符号。 • STRONG：该设置下插件检查长度、数字、大小写、特殊符号及违规词字典。
修改参数设置	支持修改功能参数，灵活调整密码强度设置。
参数同步	提供设置参数的同步与批量关闭功能，方便用户实现设置一次即可同时应用到多个集群的多集群操作。

开启自定义密码强度功能后，将对创建账号、重置密码和账户克隆功能的密码设定进行限制，开启后在这三个功能中所设定的密码必须符合您所设定的密码强度要求。
在连接数据库使用命令行操作时，自定义密码强度功能开启后，将对所有涉及密码设定操作的语句进行限制，如 create user、alter user、set password 等，使用该类语句对账户进行密码设定或修改时，所设定的密码必须符合您所设定的密码强度要求。

前提条件

已创建 TDSQL-C MySQL 版集群。

版本限制

自定义密码强度功能支持版本为：

- MySQL 5.7 内核版本 2.0.21 及以上或 2.1.7 及以上。
- MySQL 8.0 内核版本 3.1.7 及以上。

需升级至以上内核版本后使用此功能，升级操作请参见 升级内核小版本 。

互斥任务

集群处于隔离中、小版本升级、参数修改、集群回档、集群创建等任务状态时，与自定义密码强度的任务互斥，任务将按排列顺序执行操作。

相关操作

- 如子账号需要使用自定义密码强度功能，请参见 授权子用户使用自定义密码强度 。
- 如需开启和关闭此功能，请参见 开启和关闭自定义密码强度 。
- 开启自定义密码强度功能后，如需修改自定义密码强度、修改具体参数及设置参数同步，请参见 修改参数设置和参数同步 。

版权所有：腾讯云计算（北京）有限责任公司

授权子用户使用功能

最近更新时间：2024-05-13 14:51:21

本文为您介绍通过主账号为子用户授权使用自定义密码强度功能。

操作场景

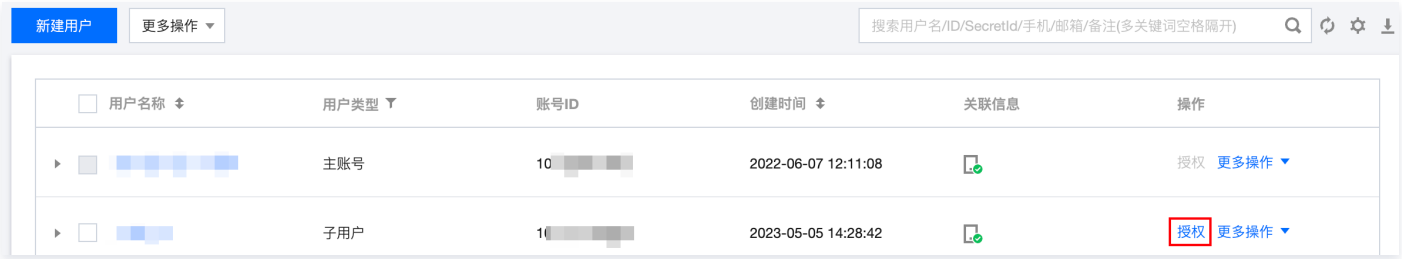
您的主账号具有自定义密码强度功能的全部权限，无需额外设置，在默认情况下，子用户是没有该功能权限的，因此您需要创建策略来允许子用户使用自定义密码强度功能。

访问管理（Cloud Access Management，CAM）是腾讯云提供的一套 Web 服务，主要用于帮助用户安全管理腾讯云账号下资源的访问权限。通过 CAM，您可以创建、管理和销毁用户（组），并通过身份管理和策略管理控制指定用户可以使用的腾讯云资源。

当您使用 CAM 的时候，可以将策略与一个用户或一组用户关联起来，策略能够授权或者拒绝用户使用指定资源完成指定任务。有关 CAM 策略的更多基本信息，请参见 策略语法 。

操作步骤

1. 以主账号身份登录 访问管理控制台 ，在用户列表选择对应子用户，单击**授权**。



2. 请根据账号需要设置下述权限。需要注意的是，无论该子用户是否需要被赋予控制该功能的权限，均建议为您的子用户添加 `cynosdb:DescribeClusterPasswordComplexity` ，方便子用户在控制台查看主账号所设定的密码强度的详细设置，进行合规的密码设定。

相关 CAM 权限说明

权限名称	功能
cynosdb:DescribeClusterPasswordComplexity	显示自定义密码强度的详细设置与开关情况
cynosdb:CloseClusterPasswordComplexity	关闭与批量关闭自定义密码强度功能的权限
cynosdb:CopyClusterPasswordComplexity	同步自定义密码强度到该账号下控制的其他集群的权限
cynosdb:ModifyClusterPasswordComplexity	修改自定义密码强度功能详细设置的权限
cynosdb:OpenClusterPasswordComplexity	开启自定义密码强度功能的权限

版权所有：腾讯云计算（北京）有限责任公司

开启和关闭自定义密码强度

最近更新时间：2024-10-12 17:40:45

使用主账号操作，或者使用被赋予对应 CAM 权限后的子用户操作时，您可使用自定义密码强度功能。本文为您介绍如何开启和关闭自定义密码强度。

版本限制

自定义密码强度功能支持版本为：

- MySQL 5.7 内核版本 2.0.21 及以上或 2.1.7 及以上。
- MySQL 8.0 内核版本 3.1.7 及以上。

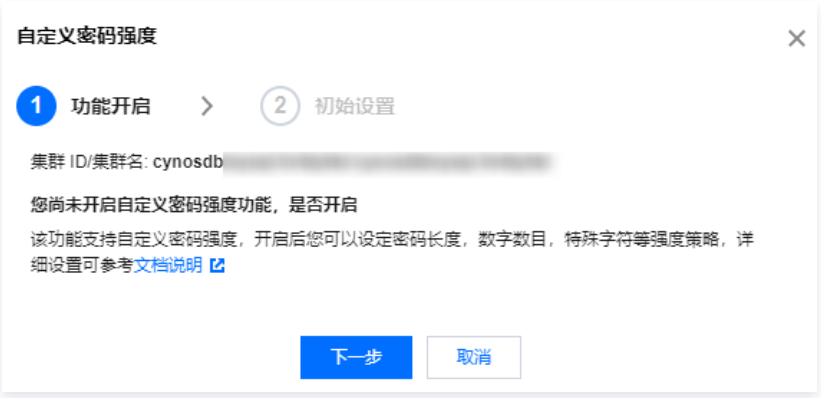
需升级至以上内核版本后使用此功能，升级操作请参见 升级内核小版本 。

开启自定义密码强度

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在上方选择地域，在集群列表中，找到目标集群，单击集群 ID，进入集群管理页面。
3. 在集群管理页选择**账号管理**，在右侧开启“自定义密码强度”功能按钮。

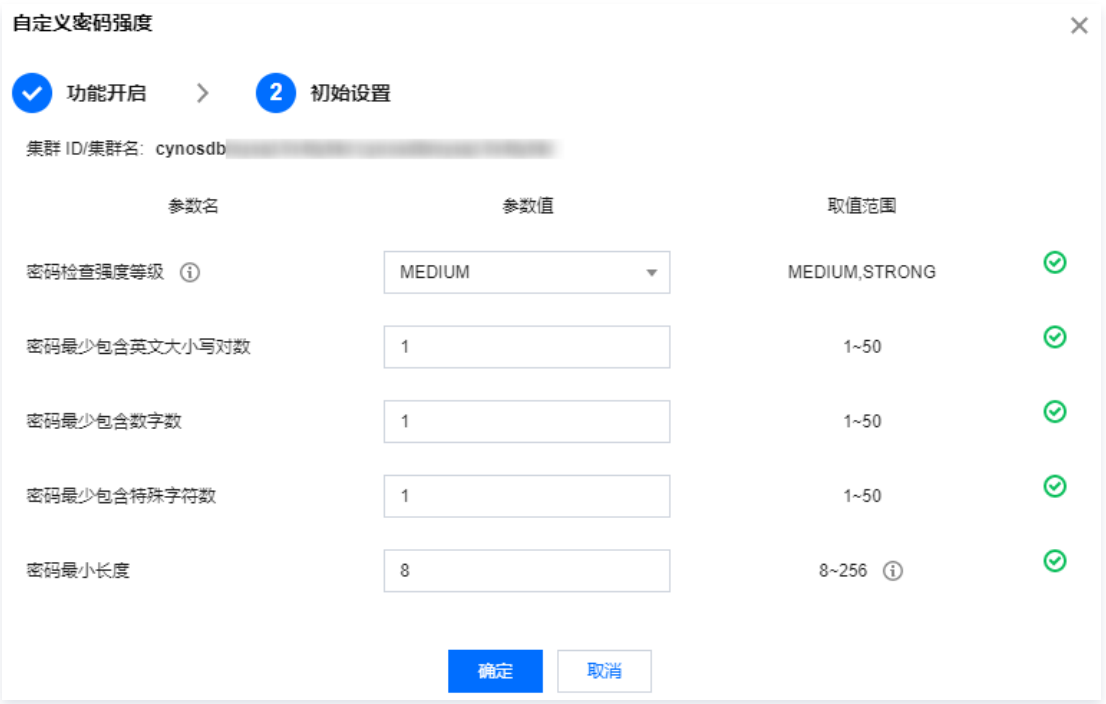


4. 在弹出的窗口下，单击**下一步**进入功能初始设置。



5. 完成如下设置后，单击**确定**。
   自定义密码强度初始设置中，随密码检查强度等级的不同，呈现两种场景。

场景一：密码检查强度等级为 MEDIUM。



参数	说明
密码检查强度等级	支持选择检查强度为 MEDIUM 或 STRONG。 MEDIUM：该设置下此功能检查长度、数字、大小写、特殊符号。 STRONG：该设置下此功能检查长度、数字、大小写、特殊符号及违规词字典。
密码最少包含英文大小写对数	密码最少要包含的大写字母个数和小写字母个数，此参数值代表一对大小写字母，如设定为2时，密码至少需要包含2个大写字母与2个小写字母，默认值为1，取值范围1 - 50。
密码最少包含数字数	密码最少要包含的阿拉伯数字个数，默认值为1，取值范围1 - 50。
密码最少包含特殊字符数	密码最少要包含的特殊字符个数，默认值为1，取值范围1 - 50。
密码最小长度	密码最少需要设定的长度，默认值为8，取值范围8 - 256，密码最小长度 = 数字个数参数 + 特殊字符参数 + 2 * 大小写字母数参数，但为保证您密码的安全，密码最小长度需大于等于8，若其余参数设置导致最小长度大于8，则按最新取值范围显示为主。

场景二：密码检查强度等级为 STRONG。



参数	说明
密码检查强度等级	支持选择检查强度为 MEDIUM 或 STRONG。 MEDIUM：该设置下此功能检查长度、数字、大小写、特殊符号。 STRONG：该设置下此功能检查长度、数字、大小写、特殊符号及违规词字典。
密码最少包含英文大小写对数	密码最少要包含的大写字母个数和小写字母个数，此参数值代表一对大小写字母，如设定为2时，密码至少需要包含2个大写字母与2个小写字母，默认值为1，取值范围1 – 50。
密码最少包含数字数	密码最少要包含的阿拉伯数字个数，默认值为1，取值范围1 – 50。
密码最少包含特殊字符数	密码最少要包含的特殊字符个数，默认值为1，取值范围1 – 50。
密码最小长度	密码最少需要设定的长度，默认值为8，取值范围8 – 256，密码最小长度 = 数字个数参数 + 特殊字符参数 + 2 * 大小写字母数参数，但为保证您密码的安全，密码最小长度需大于等于8，若其余参数设置导致最小长度大于8，则按最新取值范围显示为主。
违规词字典	当密码检查强度等级为 STRONG 时，此选项可以设置，可单击字典设置跳转操作。违规词内容输入要求：单词长度4 – 100，须为英文大小写字母。设置完成后，密码设置校验时将对密码中是否有完整的违禁词进行检查，若存在违禁词，无论大小写，将不予通过。

关闭自定义密码强度

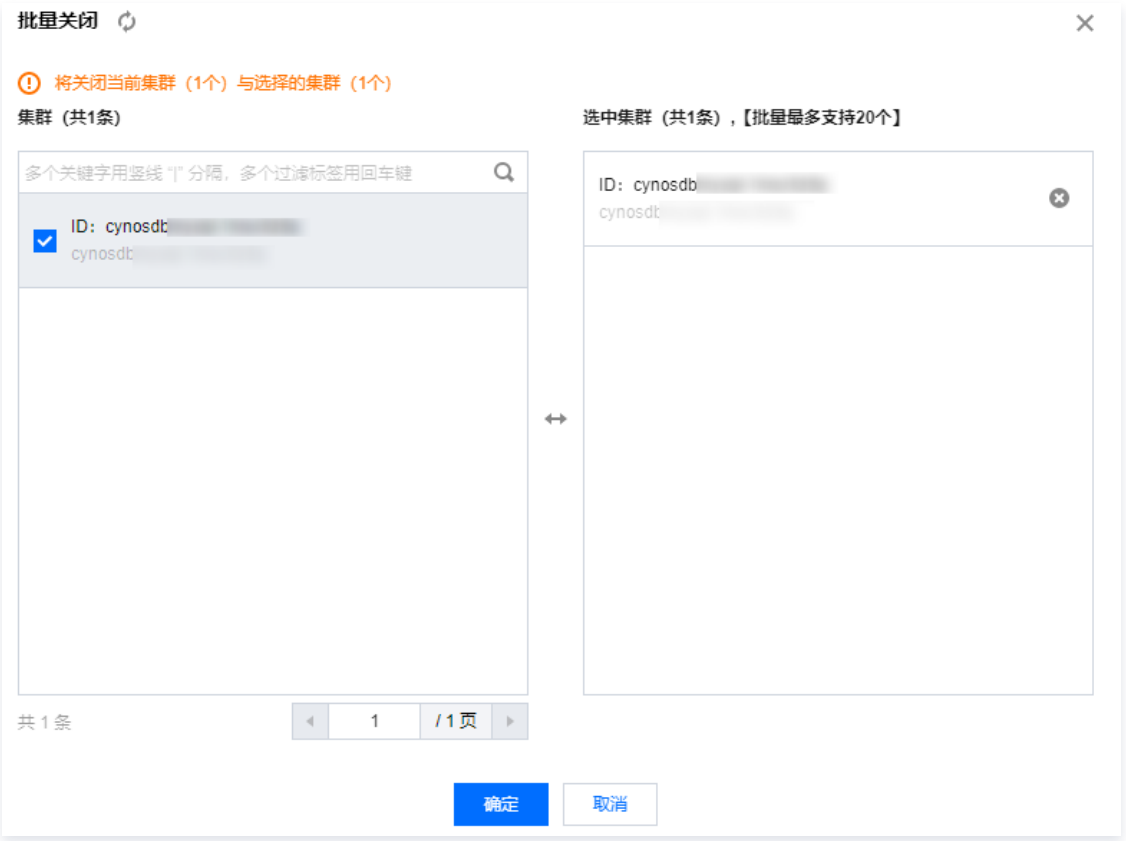
1. 登录 TDSQL-C MySQL 版控制台 。
2. 在上方选择地域，在集群列表中，找到目标集群，单击集群 ID，进入集群管理页面。
3. 在集群管理页选择**账号管理**，在右侧关闭“自定义密码强度”功能按钮。



4. 在弹出的对话框内单击**确定**，将关闭当前集群的自定义密码强度功能。



5. 在弹出的对话框内单击**批量关闭**，可以跳转批量选择关闭开启自定义密码强度功能的其他集群。



说明

- 关闭后您所设置的参数将自动恢复至默认值，再次开启需重新设置参数。
- 批量关闭操作最多支持选中20个集群。

修改参数设置和参数同步

最近更新时间：2024-10-14 10:43:41

开启自定义密码强度功能后，支持修改自定义密码强度的参数，支持将当前集群下此功能设置的参数同步至同一账号下管理的其它集群。本文为您介绍修改自定义密码强度参数和设置参数同步。

前提条件

已 开启自定义密码强度 。

修改自定义密码强度参数

> **说明：**
> 密码检查强度等级分为 MEDIUM 和 STRONG，其中强度设置为 STRONG 时才会出现参数“违规词字典”。

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在上方选择地域，在集群列表中，找到目标集群，单击集群 ID，进入集群管理页面。
3. 在集群管理页选择**账号管理**，单击右侧的**强度设置**。



4. 在弹窗里单击**下一步**，进入参数设置页面。
5. 在跳转的参数修改界面，设置新参数配置，单击**确定**。



设置参数同步

> **说明：**
> 若您使用的是子账号登录控制台，需确保该子账号有被授权使用自定义密码强度的参数同步功能，详细授权功能名称和操作请参见 授权子用户使用功能 。

为集群开启自定义密码强度功能后，支持通过参数同步功能，将当前集群所设置的自定义密码强度参数包括违禁词字典，同步至该腾讯云账号下管理的同地域的其他集群，若同步的集群未开启该功能，则会进行功能开启并同步参数。

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在上方选择地域，在集群列表中，找到目标集群，单击集群 ID，进入集群管理页面。

3. 在集群管理页选择**账号管理**，单击右侧的**强度设置**。



4. 在跳转的参数修改界面里，单击**参数同步**。



5. 在参数同步界面，勾选需要同步的集群（支持多选或取消已选择的集群），单击**确定**。



重置密码

最近更新时间：2025-01-15 10:06:43

本文为您介绍如何通过控制台重置数据库管理员账号的密码。

操作步骤

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在上方选择地域，根据实际控制台视图模式，对应操作。

页签视图

在左侧集群列表，单击**目标集群**，进入集群管理页。



列表视图

在左侧集群列表找到目标集群，单击**集群 ID** 或**操作**列的**管理**，进入集群管理页。



3. 在集群管理页面，选择**账号管理**页，单击**操作**列的**重置密码**。



4. 在弹出的对话框，输入新密码，单击**确定**。



说明：

> 密码可以输入8个 - 64个字符，包含大小写英文字母、数字和符号 `~!@#$%^&*_-+=|\(){}[]:;'<>,.?/` 中的任意三种。

修改账号权限

最近更新时间：2025-01-15 10:06:43

您可以在控制台授权管理已有的数据库账号，可以对数据库账号授予全局特权或者对象级特权，同时您也可以取消其授权。

账号权限说明

权限	说明
ALTER	修改数据库中表的结构，包括修改表的字段、增加表字段、删除表字段、增加和删除表的索引等操作的权限。
ALTER ROUTINE	更改存储过程和函数等例程的权限。
CREATE	创建新的数据库、表、视图、存储过程、函数等对象的权限。
CREATE ROUTINE	创建存储过程和函数等例程对象的权限。
CREATE TEMPORARY TABLES	创建临时表的权限。
CREATE USER	此权限可以创建新用户并为其分配相应的权限。
CREATE VIEW	创建视图对象的权限。
DELETE	能够从指定的表中删除数据的权限。
DROP	删除数据库、表、视图、存储过程、函数等对象的权限。
EVENT	创建、修改和删除事件的权限。
EXECUTE	此权限可以让用户执行已经存在的存储过程和函数等对象。
GRANT OPTION	为其他用户授予自身的某个权限。
INDEX	创建和删除索引的权限。
INSERT	向表中插入/写入新的数据的权限。
LOCK TABLES	锁定指定的数据表以进行读写操作的权限。
PROCESS	查看当前所有正在执行的线程和进程的信息的权限。
REFERENCES	在当前数据库中创建或删除外键约束的权限。
RELOAD	重新加载系统的一些配置文件、重启或关闭数据库等操作的权限。
REPLICATION CLIENT	查看和管理复制进程状态和参数的权限。
REPLICATION SLAVE	从 Master 服务器复制数据的权限。
SELECT	从指定的表中查询数据的权限。
SHOW DATABASES	显示当前数据库服务器中的所有数据库的权限。
SHOW VIEW	显示视图的权限。
TRIGGER	创建和管理触发器的权限。
UPDATE	更新指定表的数据的权限。

操作步骤

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在上方选择地域，根据实际控制台视图模式，对应操作。

页签视图
在左侧集群列表，单击**目标集群**，进入集群管理页。



列表视图

在左侧集群列表找到目标集群，单击集群 ID 或操作列的管理，进入集群管理页。

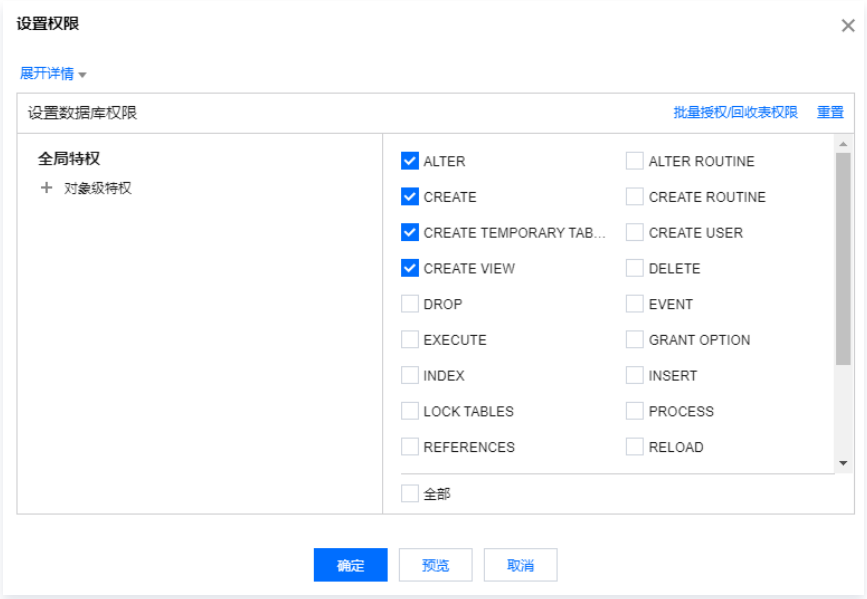


3. 在集群管理页面，选择账号管理页，单击操作列的修改权限。



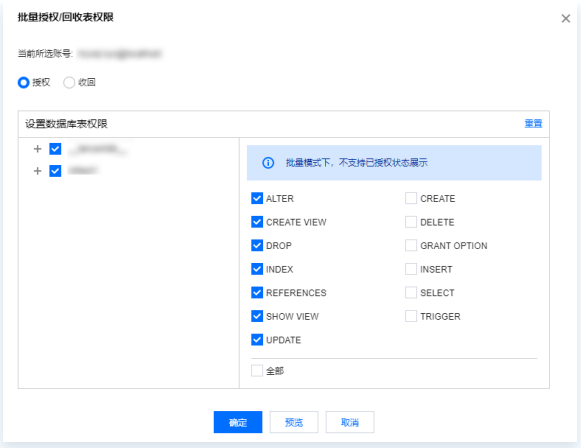
4. 在弹出的对话框，选中或者取消需要授予的权限，单击确定。

- 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。

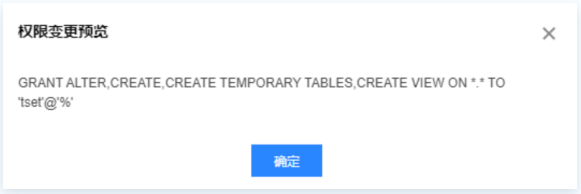


说明：

- 如果您需要批量授权或回收库表的权限，可单击**批量授权/回收表权限**，在弹窗下，选择**授权**或**收回**，然后选择需授权的多个库表以及所需权限，单击**确定**。



- 如果您想要预览账号权限的变更，您可以单击**预览**查看。



修改授权访问的主机地址

最近更新时间：2025-01-15 10:06:43

您可以通过控制台修改数据库账号所授权的主机地址，来限制对数据库的访问，进而提升数据库的访问安全。

操作步骤

> **说明：**
> root 账号不支持修改主机地址。

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在上方选择地域，根据实际控制台视图模式，对应操作。

页签视图

在左侧集群列表，单击**目标集群**，进入集群管理页。



列表视图

在左侧集群列表找到目标集群，单击**集群 ID** 或**操作**列的**管理**，进入集群管理页。



3. 在集群管理页面，选择**账号管理**页，在**操作**列选择**更多 > 修改主机**。



4. 在弹出的对话框，输入新主机地址，单击**确定**。

> **说明：**
> 主机地址支持单个 IP 形式的地址，也支持填入%（表示不做 IP 范围限制）。
> - 示例1：填入%，表示不做 IP 范围限制，即允许所有 IP 地址的客户端使用该账号访问数据库。
> - 示例2：填入10.5.10.%，表示允许 IP 范围在10.5.10.%内的客户端使用该账号访问数据库。

版权所有：腾讯云计算（北京）有限责任公司

修改账号连接数

最近更新时间：2025-01-14 17:27:22

TDSQL-C MySQL 版支持修改账号连接数限制（每个账号到数据库的最大连接数）。您可以在控制台修改账号的连接数限制，防止单个账号耗尽所有 TDSQL-C MySQL 版的连接。
本文为您介绍如何通过控制台修改数据库账号连接数。

操作步骤

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在上方选择地域，根据实际控制台视图模式，对应操作。

页签视图

在左侧集群列表，单击目标集群，进入集群管理页。



列表视图

在左侧集群列表找到目标集群，单击**集群 ID** 或**操作**列的**管理**，进入集群管理页。



3. 单击**账号管理**，在账号列表找到需要修改连接数的账号，在其**操作**列选择**更多** > **修改连接数**。



4. 在修改连接数弹窗里，输入**新连接数限制**，单击**确定**。



说明：
连接数取值范围：1 – 10240，若不填写连接数，会受到最大连接数10240的限制。

修改账号备注

最近更新时间：2025-01-14 17:27:22

您可以通过备注账号信息，实现对账号资源的分配和管理，本文为您介绍如何通过控制台修改数据库账号的备注信息。

操作步骤

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在上方选择地域，根据实际控制台视图模式，对应操作。

页签视图

在左侧集群列表，单击**目标集群**，进入集群管理页。



列表视图

在左侧集群列表找到目标集群，单击**集群 ID** 或**操作**列的**管理**，进入集群管理页。



3. 单击**账号管理**，在账号列表找到需修改备注的账号，单击其备注项后的编辑图标。



4. 在修改备注弹窗里，输入备注信息（最多可输入255个字符）后单击**确定**。



克隆账号

最近更新时间：2025-01-15 10:06:43

本文为您介绍通过控制台，对账号进行克隆操作，可直接继承原账号的账号名、主机、备注信息以及账号权限信息。

操作步骤

1. 登录 TDSQL-C MySQL 版控制台 。
2. 选择地域，根据实际控制台视图模式，对应操作。

页签视图

在左侧集群列表，单击**目标集群**，进入集群管理页。



列表视图

在左侧集群列表找到目标集群，单击**集群 ID** 或**操作**列的**管理**，进入集群管理页。



3. 在集群管理页，单击**账号管理**，在账号列表找到需克隆的账号，在其**操作**列选择**更多>克隆**。



4. 在克隆账号弹窗里，确认账号信息和输入密码后，单击**确定**。


克隆账号
账号名 *
tset
帐号名需要1-16个字符，由字母、数字和特殊字符组成；以字母开头，字母或数字结尾；特殊字符为_.
主机 *
%
1. IP形式，支持填入%
2. 多个主机以分隔符分隔，分隔符支持,,|换行符和空格
设置密码 *
••••••••
8-64个字符，包含大小写英文字母、数字和符号~!@#$%^&*_-+=`|\(){}[]:;'<>,.?/中的任意三种
确认密码 *
••••••••
连接数限制
100
取值范围：1~10240
备注
请输入备注
最多255个字符
确定
取消

删除账号

最近更新时间：2025-01-15 10:06:43

若您需要对已创建的数据库账号进行停用，可以通过控制台删除数据库账号。

> **注意：**
> - 数据库账号删除后不可恢复，请谨慎操作。
> - 避免因账号误删除而影响业务的正常使用，请确认即将删除的数据库账号不再对任何应用程序提供使用。

操作步骤

> **说明：**
> root 账号不支持删除。

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在上方选择地域，根据实际控制台视图模式，对应操作。

页签视图

在左侧集群列表，单击**目标集群**，进入集群管理页。



列表视图

在左侧集群列表找到目标集群，单击**集群 ID** 或**操作**列的**管理**，进入集群管理页。



3. 在集群管理页面，选择**账号管理**页，在账号列表找到需要删除的账号，在其**操作**列选择**更多 > 删除账号**。



4. 在弹出的对话框，确认无误后，单击**确定**。

数据库管理
创建数据库

最近更新时间：2024-10-14 10:43:41

TDSQL-C MySQL 版支持通过控制台的数据库管理页创建数据库，并支持对数据库授权账号访问。本文为您介绍数据库账号权限以及如何通过控制台创建数据库。

操作步骤

1. 登录 TDSQL-C MySQL 版控制台，在集群列表，找到目标集群，单击集群 ID 或**操作**列的**管理**，进入集群管理页面。
2. 在集群管理页面，选择**数据库管理**页，单击**创建数据库**。



3. 在弹出的对话框，配置如下参数后，单击**确定**。



参数	说明
数据库名	输入数据库名。由大小写字母、数字、中划线(-)、下划线(_)组成，字母开头，字母或是数字结尾，最长64字符。
支持字符集	设置数据库支持的字符集。字符集介绍可参见 MySQL 官方文档 。
排序规则	设置数据库的排序规则。排序规则介绍可参见 MySQL 官方文档 。
账号授权	单击添加，然后选择授权账号、授权给账号的权限、主机信息，也支持删除该授权列记录。
备注	为新建的数据库备注。最多支持输入256个字符。

账号权限授权明细

授权账号对所创建的数据库的操作权限包括只读、只写、DML、DDL 和只读 + 索引，对应权限明细及授权 SQL 语句如下表所示。

权限	授权明细	授权 SQL 语句
只读	SELECT LOCK TABLES SHOW VIEW	``GRANT SELECT, LOCK TABLES, SHOW VIEW ON `库`.* TO '账号'@'%'``
读写	ALL PRIVILEGES	``GRANT ALL PRIVILEGES ON `库`.* TO '账号'@'%'``
DML	SELECT INSERT UPDATE DELETE CREATE TEMPORARY TABLES LOCK TABLES EXECUTE SHOW VIEW EVENT TRIGGER	``GRANT SELECT, INSERT, UPDATE, DELETE, CREATE TEMPORARY TABLES, LOCK TABLES, EXECUTE, SHOW VIEW, EVENT, TRIGGER ON `库`.* TO '账号'@'%'``
DDL	CREATE DROP INDEX ALTER CREATE TEMPORARY TABLES LOCK TABLES CREATE VIEW SHOW VIEW CREATE ROUTINE ALTER ROUTINE	``GRANT CREATE, DROP, INDEX, ALTER, CREATE TEMPORARY TABLES, LOCK TABLES, CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE ON `库`.* TO '账号'@'%'``
只读 + 索引	SELECT INDEX LOCK TABLES SHOW VIEW	``GRANT SELECT, INDEX, LOCK TABLES, SHOW VIEW ON `库`.* TO '账号'@'%'``

修改数据库账号权限

最近更新时间：2025-01-14 14:36:12

TDSQL-C MySQL 版支持授权和修改数据库的账号权限，由于 root 账号默认拥有 TDSQL-C MySQL 版集群下所有数据库的读写权限，所以不支持对 TDSQL-C MySQL 版集群下的数据库进行 root 账号的授权修改，您可以在数据库管理页面对已创建的数据库，授权其他账号对其进行读写、只读等的权限。

前提条件

- TDSQL-C MySQL 版集群下有除了 root 账号以外的账号，操作方法请参见 创建账号 。
- TDSQL-C MySQL 版集群下有创建至少一个数据库，操作方法请参见 创建数据库 。

操作步骤

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在上方选择地域，根据实际控制台视图模式，对应操作。

页签视图

在左侧集群列表，单击**目标集群**，进入集群管理页。



列表视图

在左侧集群列表找到目标集群，单击**集群 ID** 或**操作**列的**管理**，进入集群管理页。



3. 在集群管理页面，选择**数据库管理**页，在目标数据库的**操作**列单击**修改**。



4. 在弹出的对话框，修改授权账号的权限，单击**确定**。



说明：

如需了解账户权限的具体授权明细，您可参见 账号授权明细 。

删除数据库

最近更新时间：2025-01-15 10:06:43

TDSQL-C MySQL 版集群支持通过控制台的数据库管理页删除已创建的数据库。

前提条件

TDSQL-C MySQL 版集群下有创建至少一个数据库，操作方法请参见 创建数据库 。

操作步骤

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在上方选择地域，根据实际控制台视图模式，对应操作。

页签视图

在左侧集群列表，单击**目标集群**，进入集群管理页。



列表视图

在左侧集群列表找到目标集群，单击**集群 ID** 或**操作**列的**管理**，进入集群管理页。



3. 在集群管理页面，选择**数据库管理**页，找到需要的数据库，单击其**操作**列的**删除**。



说明：

支持批量删除数据库，勾选多个数据库，在列表上方选择**更多操作 > 批量删除**即可。



4. 在弹出的对话框，确认无误后，单击**确定**。

数据库管理工具（DMC）
DMC 简介

最近更新时间：2024-10-12 11:35:34

DMC（Database Management Center）是腾讯云的一款数据库管理工具，提供库表级操作、实时监控、实例会话管理、SQL 窗口、数据管理为一体的数据库管理服务。

功能介绍

- **数据管理**
  提供数据库的库表增删、结构修改以及表数据维护等数据库操作。
- **数据库实例会话管理**
  提供多维度的实例会话信息展示，方便用户查看并进行操作。
- **数据库监控**
  提供数据库状态信息、InnoDB 存储引擎行操作、数据库连接以及流量相关的多维度监控。
- **SQL 窗口**
  为用户提供 SQL 编译器，支持常用 SQL 模板以及自定义 SQL 保存，支持 SQL 结果展示与导出。

工具优势

- **易操作**
  DMC 为用户提供零门槛的数据库管理服务，用户可通过操作界面方便地进行新建库表以及其他的数据库操作。
- **可视化**
  DMC 无需单独安装驱动程序或应用程序，只需登录到 DMC 控制台，即可通过可视化的管理页面对数据库进行多维度的管理、监控和操作。
- **实时数据库状态监测**
  DMC 可以实时采集数据库实例数据，包括 CPU 占用率、连接数以及存储空间使用量，为用户直观地展示数据库实例的运行状态。
- **实时数据库性能监控**
  DMC 提供数据库性能秒级监控，包含 MySQL Status Information、InnoDB Row Operation、Threads、NetWork 相关的多项数据，用户可实时监控各项性能波动。此外，DMC 支持会话的分类统计并选择 kill 实例会话，让用户快速定位异常来源进行性能优化。

版权所有：腾讯云计算（北京）有限责任公司

DMC 管理

最近更新时间：2024-07-02 17:05:32

本文为您介绍 DMC 数据库管理控制台的新建库表、库管理、实例会话、数据导入导出、表数据可视化编辑等功能。

新建库

1. 通过如下三种登录方式之一，登录数据库管理控制台。
   - 直接登录 DMC 控制台 。
   - 或在 TDSQL-C MySQL 版 集群列表单击**操作**列的**登录**（列表视图）。
   - 或在 TDSQL-C MySQL 版 集群列表下单击目标集群，进入集群管理页，在集群管理页右上方单击**登录**（页签视图）。
2. 在导航栏选择**新建** > **新建库** > **新建数据库**。



3. 在弹出的对话框，对新建的库进行相关配置，单击**确定**即可完成创建。



参数	说明
数据库名称	输入数据库名。由大小写字母、数字、中划线(-)、下划线(_)组成，字母开头，字母或是数字结尾，最长64字符。
字符集	选择字符集。具体支持的字符集请以控制台为准。
排序规则	选择字符集后会默认选择排序规则，具体排序规则请以控制台为准。

新建表

1. 通过如下三种登录方式之一，登录数据库管理控制台。
   - 直接登录 DMC 控制台 。
   - 或在 TDSQL-C MySQL 版 集群列表单击**操作**列的**登录**（列表视图）。
   - 或在 TDSQL-C MySQL 版 集群列表下单击目标集群，进入集群管理页，在集群管理页右上方单击**登录**（页签视图）。
2. 在左侧先选择需要创建表的数据库，然后在导航栏选择**新建** > **新建表**。

版权所有：腾讯云计算（北京）有限责任公司

3. 在新建表的页面，根据需要完成相关配置后，单击**提交**。

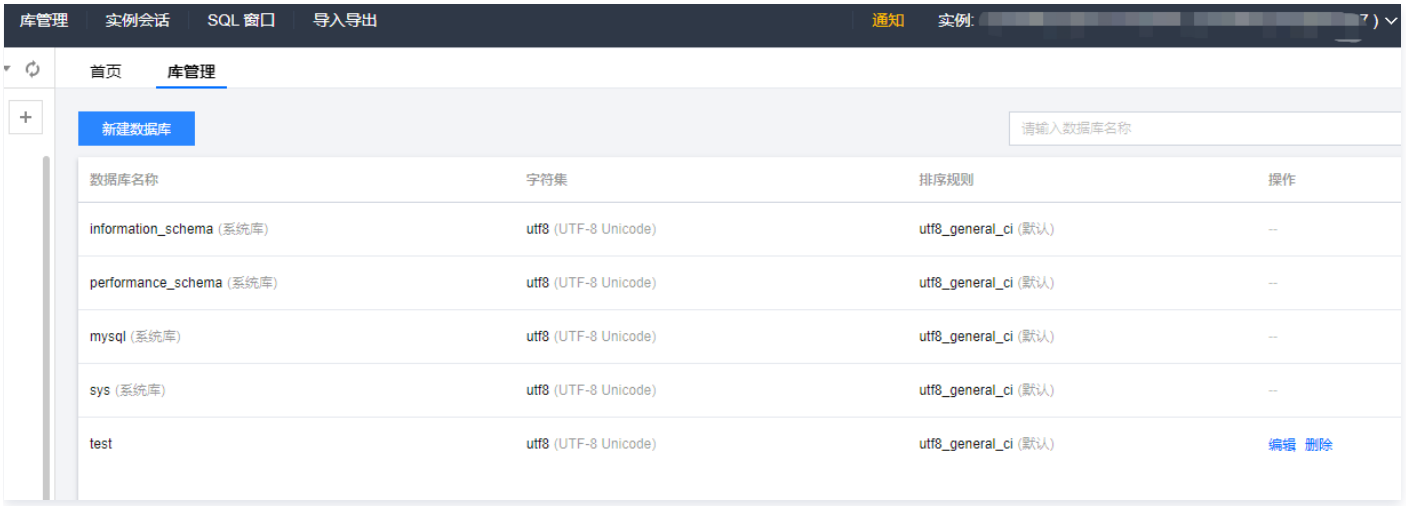


参数	说明
基本信息	在基本信息页，您需要设置表名、添加备注（非必选项）、选择存储引擎（默认 Innodb，其他引擎请以实际控制台为准）、设置字符集、设置校验规则、设置行格式、设置平均行长度、设置最小行、设置最大行。
列信息	在列信息页，您可以通过新增、删除、插入、上移、下移操作来完成表列信息的设置。
索引	在索引页，您可以新增索引，它可以帮助快速定位到包含特定数据值的数据行，具体支持的索引类型和索引方式请以控制台为准。
外键	在外键页，您可以新增外键，外键用于建立表与表之间关联关系的约束。外键约束指定了一个表中的列，该列的值必须与另一个表中的某个列的值相匹配。这个被匹配的列通常是另一个表的主键。
分区	在分区页，您可以通过将表数据分成多个分区，来提高查询性能、简化数据维护和管理

说明
字符集、排序规则介绍可参见 官方文档 。

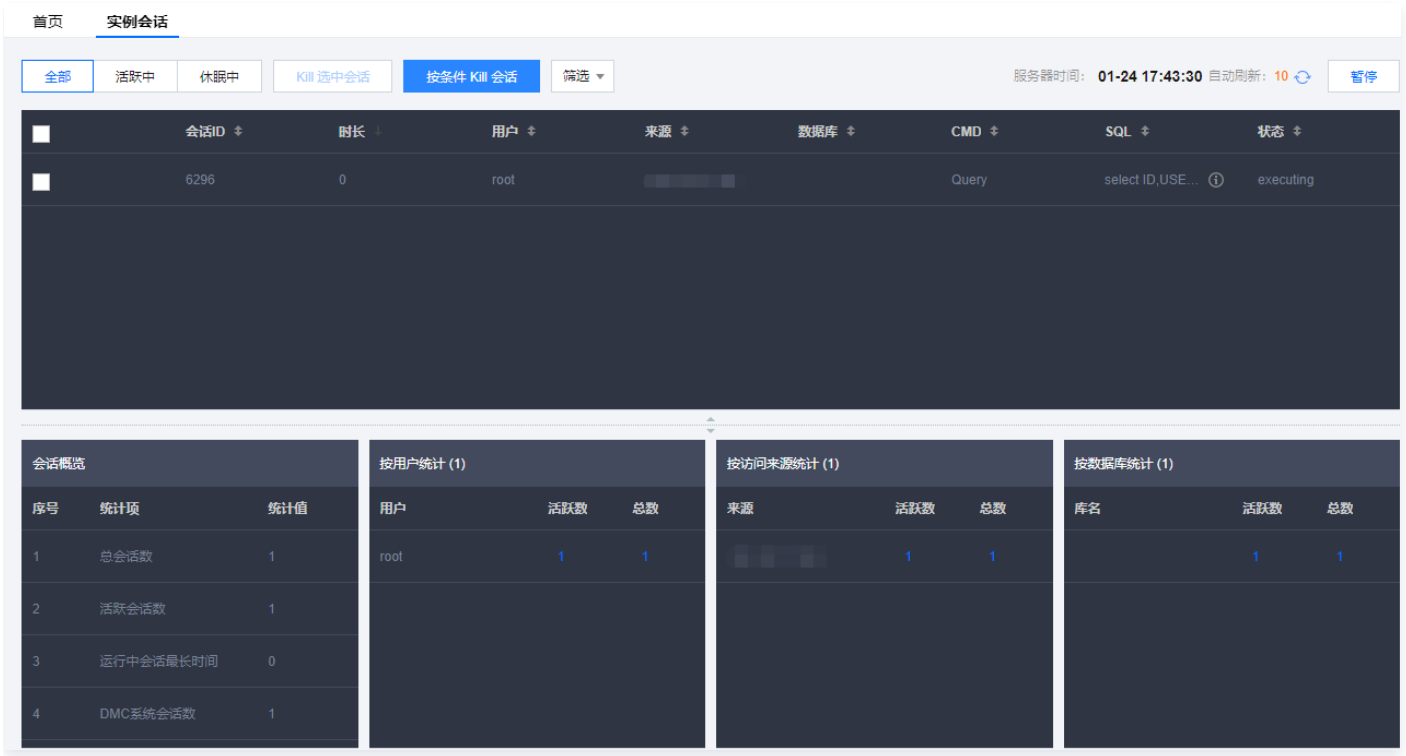
库管理

登录 DMC 控制台，在导航栏单击**库管理**，进入数据库管理页面，用户可新建、编辑、删除数据库。



实例会话

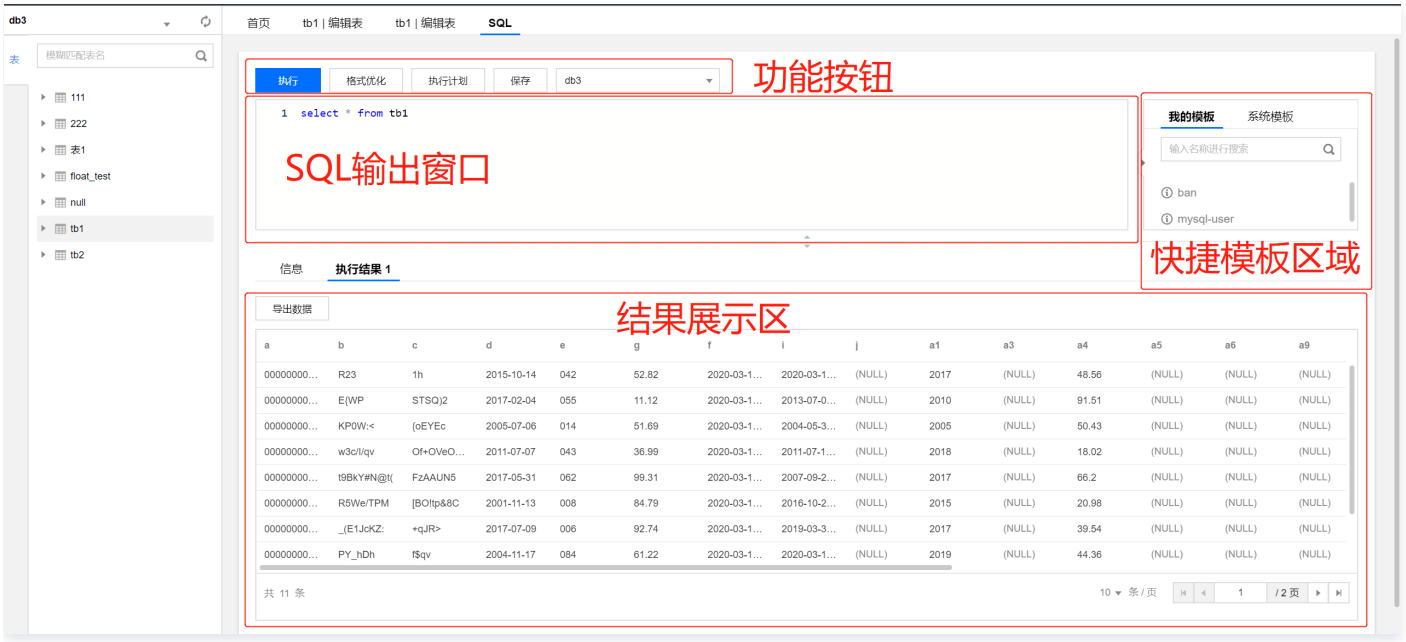
登录 DMC 控制台，在导航栏单击**实例会话**，进入实例会话页面，用户可查看当前数据库中所有实例的会话详细信息，以及按照会话概览、用户、访问来源和数据库四个不同维度的信息展示。
DMC 提供 kill 会话的功能，方便用户对会话进行管理。



SQL 窗口

登录 DMC 控制台，单击顶部导航中的 **SQL 窗口**，或者左侧栏表格操作菜单中的 SQL 操作进入 SQL 窗口页面。SQL 窗口支持如下功能：

- SQL 命令执行及结果查看
- SQL 格式优化
- 查看 SQL 命令执行计划
- 常用 SQL 保存
- 模板 SQL
- SQL 结果导出



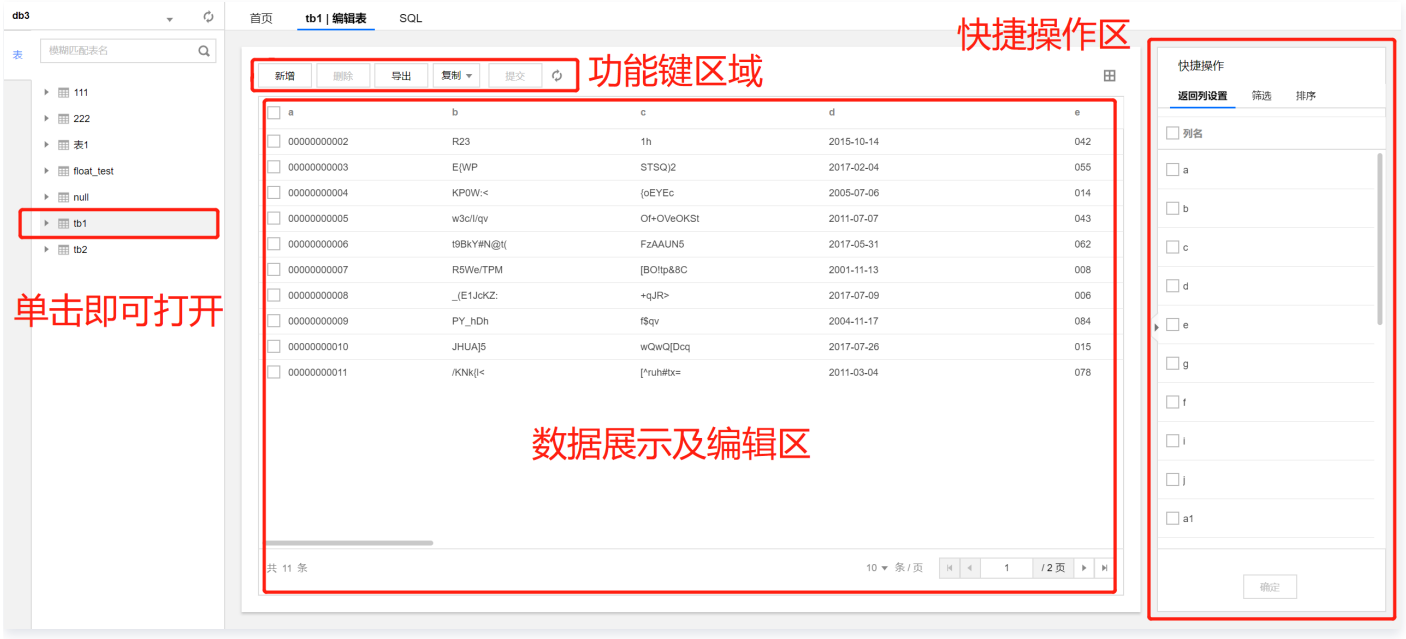
导入导出

登录 DMC 控制台，在导航栏选择**导入导出 > 数据导入**或**数据导出**，可对数据库进行数据导入导出操作。



表数据可视化编辑

DMC for TDSQL-C MySQL 版增加了对数据增删改的支持。用户可在左侧栏单击数据表，对表数据进行批量的增、删、改操作，修改完成后，在快捷操作栏单击确定预览本次修改的 SQL 语句，二次确认后将批量执行修改。



参数配置
参数概述

最近更新时间：2024-10-10 14:05:43

系统变量（system variable）也称参数，是用于管理和控制数据库的相关配置，本文为您介绍参数相关说明。

使用场景

TDSQL-C MySQL 版提供了丰富的参数供您用于优化数据库的性能，您可直接通过控制台，通过设置参数，对格式类、权限类、功能开关、字符集、执行条件、缓存大小、时间/数量/大小限制、对表的操作方式等进行修改默认值，使得调整后数据库的参数更加适配您的业务场景对数据库的需求。

参数分类

TDSQL-C MySQL 版提供的参数分为全局参数（global）和非全局参数（session），全局参数针对整个集群，设置的参数值将对集群下所有实例生效，非全局参数针对集群下的某个实例，设置的参数值只会对当前实例生效，但设置后也支持将该参数值同步至其他实例。

支持的参数操作

- TDSQL-C MySQL 版提供了多种参数模板，在创建集群时，您可直接使用默认了参数值的参数模板，保障 TDSQL-C MySQL 版在兼顾稳定性的同时提升数据库的性能。
- 支持根据实际业务需要，单个或批量调整参数值，灵活适配各种场景。
- 支持新建模板，以默认参数模板为基础，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。
- 支持从参数设置保存为模板，应用到其他集群。
- 部分参数支持设置公式化参数值，使用公式设置的参数值会适应规格的变更而自动变化，使数据库始终保持最优或最稳定的状态。
- 支持全局参数和非全局参数。

相关文档

- 如需了解实例参数、单个/批量修改参数、通过导入参数模板修改参数、通过导入参数配置文件修改参数、导出参数配置文件、查询参数修改记录等，请参见 设置实例参数 。
- 如需了解新建参数模板、复制参数模板、修改参数模板的参数值、导入/导出/删除参数模板等，请参见 使用参数模板 。
- 如需了解参数公式化能力和支持的具体参数，请参见 参数公式 。
- 如需获取参数设置的相关建议，请参见 参数配置建议 。

版权所有：腾讯云计算（北京）有限责任公司

设置实例参数

最近更新时间：2024-10-12 09:28:11

您可以通过控制台查看参数及对参数的修改记录，也可以根据业务需要调整部分参数值，本文为您介绍通过控制台修改参数的相关操作。

注意事项

- 为保证实例的稳定，控制台仅开放部分参数的修改，控制台的参数配置页面展示的参数即为可以修改的参数。
- 如果修改的参数需要重启实例才生效，系统会提示您是否重启，建议您在业务低峰期操作，并确保应用程序具有重连机制。
- 参数任务：修改参数任务详情可通过任务列表查询。
  - 如参数任务暂未执行，可撤销。
  - 如该集群存在参数修改任务，在未完成时，再次修改参数会导致修改失败。
- 全局参数需要在读写实例上进行修改，修改后将应用至集群下的所有实例。非全局参数可以在集群下的实例上单独设置并生效。

修改单个参数

1. 登录 TDSQL-C MySQL 版控制台，在集群列表，找到目标集群，单击集群 ID，进入集群管理页面。
2. 在集群管理页面，选择参数设置页，选择需要修改参数的实例 ID（全局参数需要在读写实例进行设置，设置后对集群整体生效，非全局参数可以在任一实例单独设置）。



> **注意：**
> TDSQL-C MySQL 版对参数新增了**是否全局参数**字段，方便您快速区分哪些参数对集群整体生效，哪些参数可以对不同实例设置不同参数值。全局参数设置的参数值将对集群下所有实例生效，非全局参数的参数值设置后只会对当前实例生效，但您可以将该值同步至其他实例。

3. 在参数列表找到目标参数所在行，在**参数运行值**列，单击 修改参数值。

> **说明：**
> 您也可通过参数设置页右侧检索框搜索参数名快速找到需要修改的参数。
>
> 
>

4. 根据参数可修改值列的提示，输入目标参数值，单击 保存，单击 可取消操作。

> **说明：**
> 若参数值前显示 % 图标，表明当前参数支持整型和公式化两种设置方式。公式化设置后参数值将随 cpu /内存等规格自动调整（公式仅支持修改数值变量，详情请参考 参数公式）。
>
> 
>

5. 单击 后，在弹出的对话框确认无误后，选择执行时间，单击**确定**。

   执行时间：

- 选择立即执行时，则会在确认后，立即触发修改。
- 选择维护时间内修改时，则会在实例的维护时间内进行修改。修改维护窗口期可参考 设置实例维护时间 。

注意：

参数修改确认时，系统会提示您是否会重启数据库实例，您可选择立即执行或维护时间内执行。

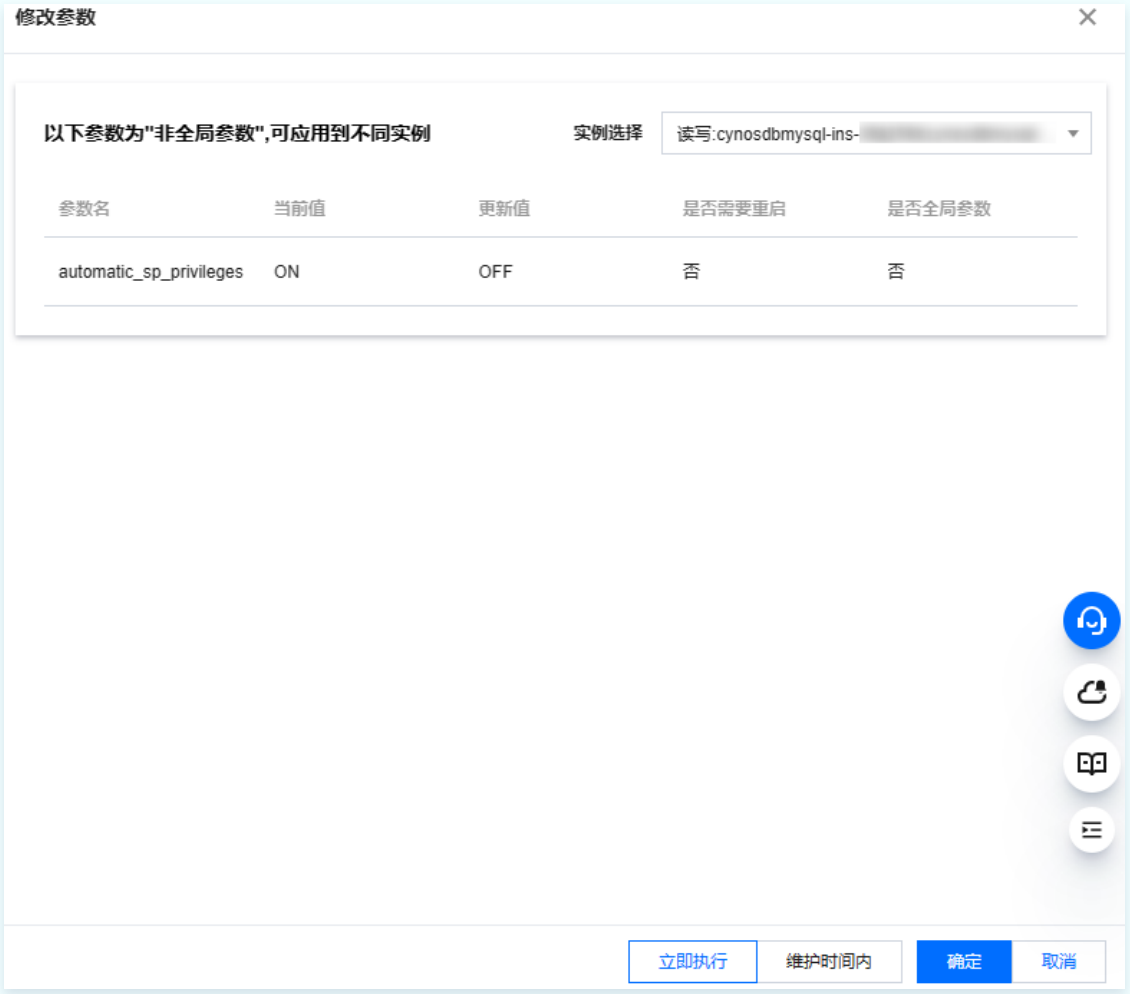
修改非重启项界面如下：



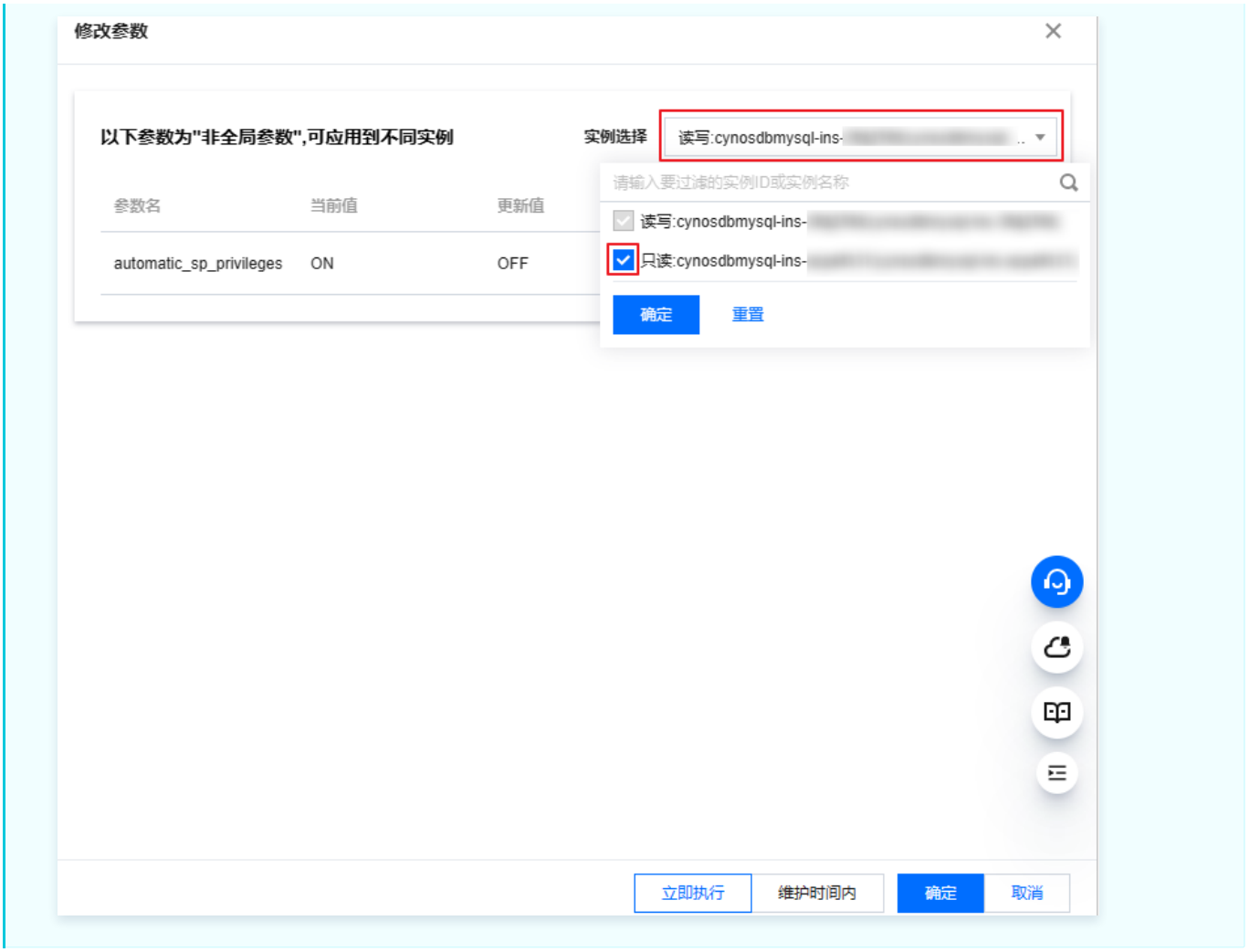
修改重启项提示如下：



若修改参数包含非全局参数，将提示您是否将该参数同步应用至其他实例，您可选择不同步，单独对当前实例设置该参数值。



或者在下拉框中选择要同步设置的实例 ID，选中后，该非全局参数值将同步应用至选中的实例。

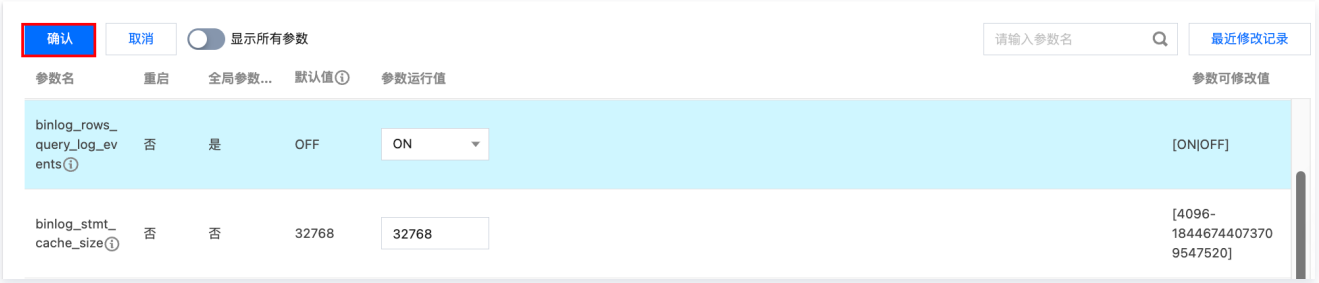


批量修改参数

1. 登录 TDSQL-C MySQL 版控制台，在集群列表，找到目标集群，单击集群 ID，进入集群管理页面。
2. 在集群管理页面，选择参数设置页，选择需要修改参数的实例 ID（全局参数需要在读写实例进行设置，设置后对集群整体生效，非全局参数可以在任一实例单独设置）。



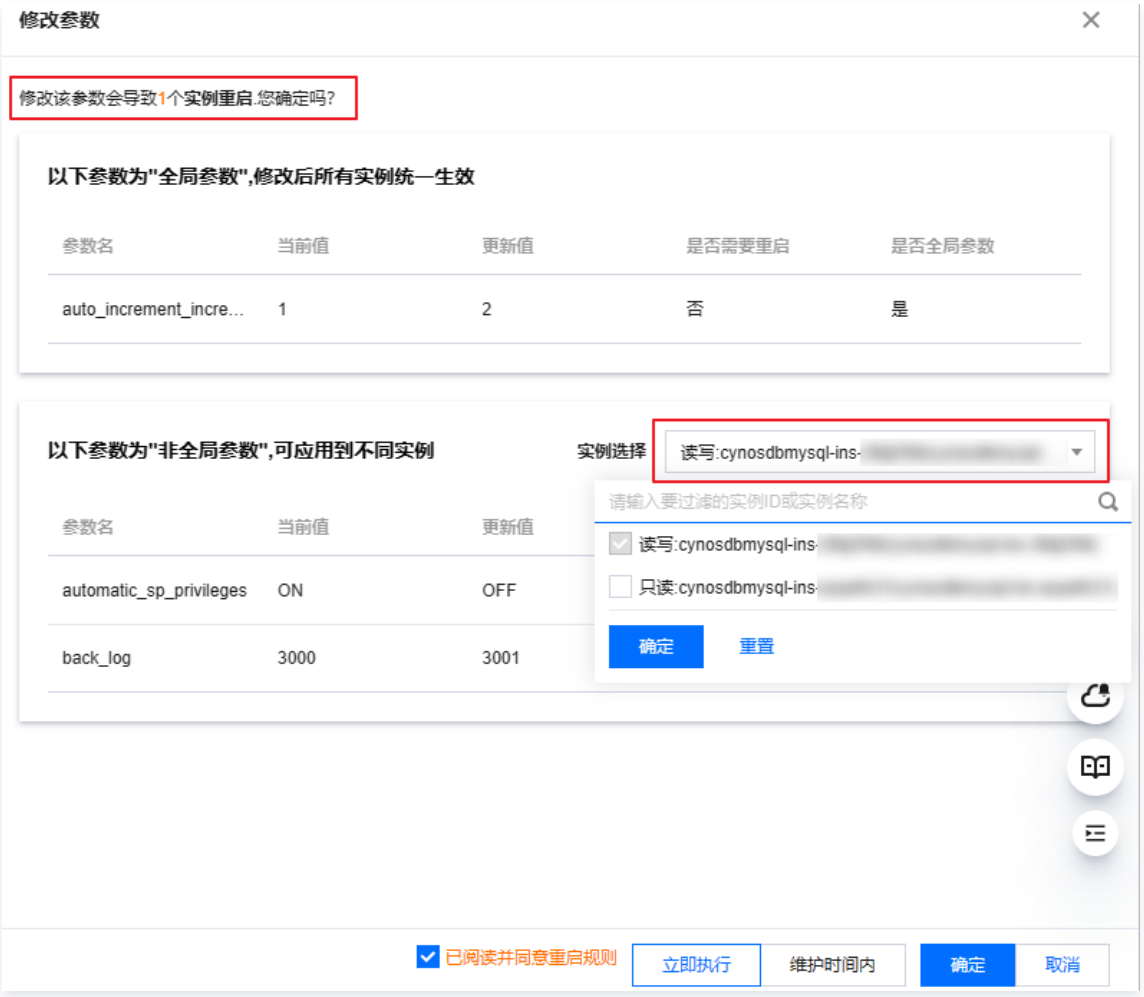
3. 单击**修改参数**。
4. 在参数运行值列，选择需要修改的参数进行修改，确认无误后，单击**确认**。



说明：

> 修改参数时或修改后，单击“显示所有参数”前的按钮，可仅显示已进行修改的参数。

5. 在弹出的对话框，系统会提示您是否会重启数据库实例，若存在非全局参数，可以选择应用至其他实例，确认无误后，选择立即执行或维护时间内执行，然后单击**确定**。



通过导入参数模板修改参数

1. 登录 TDSQL-C MySQL 版控制台，在集群列表，找到需要的集群，单击集群 ID，进入集群管理页面。
2. 在集群管理页面，选择参数设置页，选择目标实例 ID，单击**从模板导入**。



3. 在弹出的对话框，选择参数模板，单击**确定**。
4. 确认参数后，单击左上角的**确认**。
5. 在弹出的对话框，系统会提示您是否会重启数据库实例，若存在非全局参数，可以选择应用至其他实例，确认无误后，选择立即执行或维护时间内执行，然后单击**确定**。

通过导入参数配置文件修改参数

> **注意：**
> 参数配置文件不支持导入公式。

1. 登录 TDSQL-C MySQL 版控制台，在集群列表，找到需要的集群，单击集群 ID，进入集群管理页面。
2. 在集群管理页面，选择参数设置页，选择目标实例 ID，单击**导入参数**。



3. 在弹出的对话框，选择参数文件上传后，单击**导入并覆盖原有参数**。



4. 确认参数后，单击左上角的**确认**。
5. 在弹出的对话框，系统会提示您是否会重启数据库实例，若存在非全局参数，可以选择应用至其他实例，确认无误后，选择立即执行或维护时间内执行，然后单击**确定**。

导出参数配置文件

> **注意：**
> 当前导出参数配置文件时，公式化参数值将自动转换为整数值导出。

1. 登录 TDSQL-C MySQL 版控制台，在集群列表，找到需要的集群，单击集群 ID，进入集群管理页面。
2. 在集群管理页面，选择参数设置页，选择目标实例 ID，单击**导出参数**，导出文件直接会保存到本地。



查询参数修改记录

1. 登录 TDSQL-C MySQL 版控制台，在集群列表，找到需要的集群，单击集群 ID，进入集群管理页面。
2. 在集群管理页面，选择**参数设置**页，在右侧单击**最近修改记录**。



3. 在跳转的页面，您可在右侧选择需要查看最近修改记录的实例 ID，然后查看其参数修改记录，可查看的字段信息包括（参数名称、修改前参数值、修改后参数值、修改状态、修改时间）。



使用参数模板

最近更新时间：2024-05-13 15:12:21

TDSQL-C MySQL 版除了提供多种系统参数模板供您选择外，您也可以创建自定义参数模板，根据自己的业务场景，实现批量参数设置。
您可以使用数据库参数模板管理数据库引擎的参数配置。数据库参数组就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。
参数模板支持如下功能，用户可登录控制台，在左侧边栏选择**参数模板**页查看参数：

- 支持系统默认参数模板，参数默认值保证了 TDSQL-C MySQL 版在兼顾稳定性的同时提升数据库性能。
- 支持新建模板，以默认参数模板为基础，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。
- 支持从参数设置保存为模板。
- 参数模板的部分参数支持设置公式化参数值。

> ⚠ **注意：**
> - 已使用参数模板的数据库实例并不会随着参数模板更新而同步更新，需要手动批量更新数据库实例。
> - 如果您需要将新的参数应用到批量实例上，可通过批量参数设置时导入模板来重新应用。

新建参数模板

当您想使用自己的数据库参数模板时，只需创建一个新的数据库参数模板，修改所需的参数并应用于您的数据库，就可以使用新的数据库参数模板。

1. 登录 TDSQL-C MySQL 版控制台，在左侧导航选择**参数模板**页面，单击**创建模板**。



2. 在弹出的对话框，配置如下参数，单击**确定**。



参数	说明
模板名称	输入参数模板的名称，在参数模板名称需具有唯一性。
模板描述	输入参数模板的简要说明。
集群形态	选择集群所属的形态，包年包月/按量付费集群归属为通用，serverless 为单独形态。
数据库版本	选择您需要的数据库版本。

3. 创建完成后，您可以在模板详情页进行修改、导入、导出参数等操作。

复制参数模板

当您已创建一个数据库参数模板，并且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。

方式一：通过已有参数模板复制参数模板

1. 登录 TDSQL-C MySQL 版控制台，在左侧导航选择**参数模板**页面，找到目标参数模板，单击其**操作**列的**查看修改**，进入模板详情页。
2. 在模板详情页，单击上方的**另存为模板**。
3. 在弹出的对话框，配置如下参数：
   - **模板名称**：输入参数模板的名称，在参数模板名称需具有唯一性。
   - **模板描述**：输入参数模板的简要说明。
4. 确认无误后，单击**确定**即可将当前参数模板另存为新的参数模板，完成复制操作。

方式二：通过实例的参数设置功能复制参数模板

1. 登录 TDSQL-C MySQL 版控制台，在集群列表，找到目标集群，单击集群 ID，进入集群管理页面。
2. 在集群管理页面，选择**参数设置**页，单击**另存为模板**。
3. 在弹出的对话框，配置如下参数：
   - **模板名称**：输入参数模板的名称，在参数模板名称需具有唯一性。
   - **模板描述**：输入参数模板的简要说明。
4. 确认无误后，单击**确定**即可将当前参数模板另存为新的参数模板，完成复制操作。

修改参数模板的参数值

1. 登录 TDSQL-C MySQL 版控制台，在左侧导航选择**参数模板**页面，找到目标参数模板，单击其**操作**列的**查看修改**，进入模板详情页。
2. 在模板详情页，单击**修改参数**、或在**参数运行值**列，单击 可修改参数值。

> **注意：**
> 选择**导入参数**时，在从本地文件导入页中，选择参数配置文件时，需要注意配置文件的格式与 MySQL 数据库服务器的配置文件格式一致或者使用导出参数的文件模板，否则提示导入不成功。

导入参数模板

1. 登录 TDSQL-C MySQL 版控制台，在左侧导航选择**参数模板**页面，找到目标参数模板，单击其**操作**列的**查看修改**，进入模板详情页。
2. 在模板详情页，单击**导入参数**。

> **注意：**
> 选择**导入参数**时，在从本地文件导入页中，选择参数配置文件时，需要注意配置文件的格式与 MySQL 数据库服务器的配置文件格式一致或者使用导出参数的文件模板，否则提示导入不成功。

3. 在弹出的对话框，选择文件，单击**导入并覆盖原有参数**。

导出参数模板

> **注意：**
> 当前导出参数模板时，公式化参数值将自动转换为整数值导出。

方式一

1. 登录 TDSQL-C MySQL 版控制台，在左侧导航选择**参数模板**页面。
2. 在参数模板列表，在目标模板的**操作**列，单击**导出**即可。

方式二

1. 登录 TDSQL-C MySQL 版控制台，在左侧导航选择**参数模板**页面，单击**操作**列的**查看修改**，进入模板详情页。
2. 在模板详情页，在上方单击**导出参数**。

删除参数模板

当您创建了多余的参数模板或者确定该参数模板已无使用价值，可通过删除参数模板功能删除不再使用的参数模板。

1. 登录 TDSQL-C MySQL 版控制台，在左侧导航选择**参数模板**页面。
2. 在参数模板列表，在目标模板的**操作**列，单击**删除**。

3. 在弹出的对话框，单击**确定**。

参数公式

最近更新时间：2024-10-14 10:43:41

版权所有：腾讯云计算（北京）有限责任公司

TDSQL-C MySQL 版部分参数新增了公式化参数值能力，方便参数更智能地适配数据库，当实例规格发生变更时，使用公式设置的参数值会适应规格的变更而自动变化，使数据库始终保持最优或最稳定的状态。

注意事项

- 本次参数公式仅支持参数值为数值类型的参数，其余数据类型的参数值不支持公式化。
- 参数设置公式化数值后将随着实例规格的变更而变化，若您设置的公式计算出的参数值超过该参数限制的取值范围（最小值到最大值），则该参数取值会按照参数范围最近的边界值选取，即，若低于最小值，则参数取范围最小值，若高于最大值，则参数取范围最大值。
  示例：
  - 某参数根据设置的公式计算出的参数值为7，而该参数限制的取值范围为1 – 6，则参数取值为6。
  - 某参数根据设置的公式计算出的参数值为5，而该参数限制的取值范围为6 – 10，则参数取值为6。
- 导出为配置文件/从配置文件导入，均不支持公式化参数值，导出时将自动调整为整数值型。
- 为保证数据库可用性，目前仅支持部分参数进行公式化设置，更多参数敬请期待后续迭代。

参数公式说明

参数公式化组成	名称	说明
变量	DBinitMemory	实例规格的内存大小，整数型，单位为MB。
	DBInitCpu	实例规格的 CPU 核数，整数型，单位为核。
运算符	除法运算符（/）	用被除数除以除数，返回整数型商。如果计算结果为小数，会截断取整数部分。
	乘法运算符（*）	两个乘数相乘，返回整数型积。如果计算结果为小数，会截断取整数部分。
函数	MIN（）	返回整数型或者参数公式列表中最小的值。
	MAX（）	返回整数型或者参数公式列表中最大的值。

示例：
若公式为 MAX(DBInitCpu/2,4)，则表明参数值为该实例的 CPU 除以2，与4进行对比，返回最大值。

支持参数公式的参数

以下参数为当前版本支持的参数公式，默认公式的数值部分均可自定义修改，您可根据业务需求自定义调整。

参数	说明	默认公式
binlog_cache_size	在事务期间，用于保存更改的二进制日志的内存缓冲区的大小。	MIN(DBInitMemory/4000 * 32768,2097152)
max_heap_table_size	此变量为设置 MEMORY 允许用户创建的表增长到的最大大小。	MIN(DBInitMemory/1000 * 4194304,134217728)
innodb_buffer_pool_size	缓冲池的大小（以字节为单位），InnoDB 缓存表和索引数据的内存区域。	min((DBInitMemory – 500), DBInitMemory*3/4)*1000000
innodb_read_io_threads	InnoDB 中用于读操作的 I/O 线程数。	MAX(DBInitCpu/2,4)
innodb_write_io_threads	InnoDB 中用于写操作的 I/O 线程数。	MAX(DBInitCpu/2,4)
join_buffer_size	用于普通索引扫描、范围索引扫描和执行全表扫描的表连接的缓冲区的最小大小。	MIN(DBInitMemory*128,262144)
max_connections	最大连接数。	MIN(DBInitMemory/4+500,100000)
table_definition_cache	打开的表缓存实例的数量。	MAX(DBInitMemory*512/1000,2048)
table_open_cache	表描述符缓存大小，可减少文件打开/关闭次数。	MIN(MAX(DBInitMemory*512/1000,2048), 65536)

版权所有：腾讯云计算（北京）有限责任公司

thread_pool_size	该参数设置线程池中线程组的数量，默认值时表示线程组数与 CPU 数量一致。	MIN(DBInitCpu,64)
thread_cache_size	应该在缓存中保留多少线程以供重用。	MIN(DBInitMemory/125+8,512)
tmp_table_size	内部内存临时表的最大大小。	MIN(DBInitMemory/1000*4194304,134217728)

参数配置建议

最近更新时间：2024-10-12 17:01:55

TDSQL-C MySQL 版已在官方的默认值基础上进行了优化，但基于客户不同的业务场景，在购买实例后，建议根据您的业务场景对如下参数进行合理的配置：

character_set_server

- 默认值：UTF8
- 是否需要重启：是
- 作用：用于配置 MySQL 服务器的默认字符集。TDSQL-C MySQL 版提供4种字符集，分别为 LATIN1、UTF8、GBK、UTF8MB4：
- LATIN1 支持英文字符，一个字符占用一个字节。
- UTF8 包含全世界所有国家需要用到的字符，是国际编码，通用性强，一个字符占用三个字节。
- GBK 的文字编码是用双字节来表示的，即不论中、英文字符均使用双字节来表示。
- UTF8MB4 作为 UTF8 的超集，完全向下兼容，一个字符占用四个字节，且支持 emoji 表情。
- 建议：购买实例后，根据业务所需要支持的数据格式选择适合的字符集，确保客户端与服务器端设置相同的字符集，避免因字符集设置不正确而引发乱码的问题和不必要的重启操作。

lower_case_table_names

- 默认值：0
- 是否需要重启：是
- 作用：创建数据库及表时，存储与查询时是否大小写敏感。该参数可以设置的值为0、1，默认的参数值为0，表示创建数据库及表时，存储与查询均区分大小写，反之则不做区分。
- 建议：数据库 MySQL 默认大小写敏感，请根据您的业务需求及使用习惯进行合理的配置。

sql_mode

- 默认值：

```
NO_ENGINE_SUBSTITUTION（5.6版本），ONLY_FULL_GROUP_BY、STRICT_TRANS_TABLES、NO_ZERO_IN_DATE、
NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO、NO_AUTO_CREATE_USER、NO_ENGINE_SUBSTITUTION（5.7版本）
```

- 是否需要重启：否
- 作用：MySQL 可以运行在不同 sql 模式，sql 模式定义了 mysql 应该支持的 sql 语法、数据校验等。
- 5.6版本的默认参数值为 `NO_ENGINE_SUBSTITUTION`，表示使用的存储引擎被禁用或未编译则抛出错误。
- 5.7、8.0版本的默认参数值为

  ```
  ONLY_FULL_GROUP_BY、STRICT_TRANS_TABLES、NO_ZERO_IN_DATE、NO_ZERO_DATE、ERROR_FOR_DIVISION_BY_ZERO、
  NO_AUTO_CREATE_USER、NO_ENGINE_SUBSTITUTION
  ```

  其中：
  - `ONLY_FULL_GROUP_BY` 表示在 GROUP BY 聚合操作时，如果在 SELECT 中的列、HAVING 或者 ORDER BY 子句的列，必须是 GROUP BY 中出现或者依赖于 GROUP BY 列的函数列。
  - `STRICT_TRANS_TABLES` 为启用严格模式；NO_ZERO_IN_DATE 是否允许日期中的月份和日包含 0，且受是否开启严格模式的影响。
  - `NO_ZERO_DATE` 数据库不允许插入零日期，且受是否开启严格模式的影响。
  - `ERROR_FOR_DIVISION_BY_ZERO` 在严格模式下，INSERT 或 UPDATE 过程中，如果数据被零除，则产生错误而非警告，而非严格模式下，数据被零除时 MySQL 返回 NULL。
  - `NO_AUTO_CREATE_USER` 禁止 GRANT 创建密码为空的用户。
  - `NO_ENGINE_SUBSTITUTION` 使用的存储引擎被禁用或者未编译则抛出错误。
- 建议：由于不同的 SQL 模式支持不同的 SQL 语法，建议根据您的业务场景及开发习惯进行合理的配置。

long_query_time

- 默认值：10
- 是否需要重启：否
- 作用：用于指定慢查询的界定时间，默认值为10s。当某个查询执行时间为10s及以上，该查询的执行情况会记录于慢日志中，便于过后对慢查询进行分析。

版权所有：腾讯云计算（北京）有限责任公司

- 建议：基于客户业务场景及性能敏感度不同，建议根据各自业务场景设置合理的值，以便事后进行性能分析。

多可用区部署
多可用区部署概述

最近更新时间：2025-06-23 18:26:21

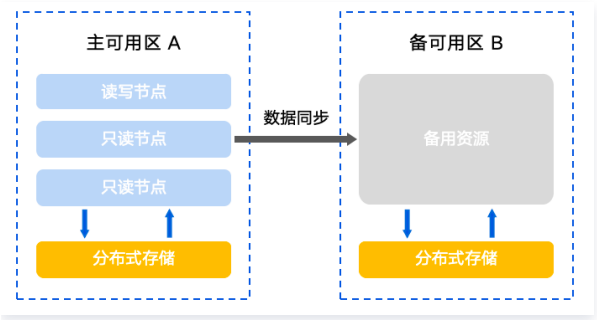
TDSQL-C MySQL 版集群实例形态为预置资源或者 Serverless，均支持多可用区部署，相比单可用区部署，多可用区部署的方式具备更高的容灾能力，可以保护数据库，以防数据库实例发生故障或可用区中断，可以抵御机房级别的故障。多可用区部署为数据库实例提供高可用性和故障转移支持。多可用区是在单可用区的级别上，将同一地域的多个单可用区组合成的物理区域。本文为您介绍实例形态为预置资源的集群的多可用区部署，如需了解实例形态为 Serverless 的多可用区部署，请参见 多可用区部署 。

> **说明：**
> TDSQL-C MySQL 版的备可用区为容灾使用，不对外提供访问。

前提条件

- 集群所在的地域需要包含两个及以上的可用区。
- 目标可用区拥有足够计算资源。
- 数据库版本要求：
  - 数据库版本5.7需满足内核小版本不低于2.0.15。
  - 数据库版本8.0需满足内核小版本不低于3.0.1。

多可用区部署架构



支持的地域和可用区

- 目前此功能为公测期间，暂时仅支持如下表所示的地域和可用区。
- 此功能会逐渐扩充支持地域和可用区。
- 如因业务需要，您可 提交工单 反馈其他地域和可用区的部署需求。

支持地域	支持主可用区	支持备可用区
北京	北京三区	北京五区
	北京五区	北京七区
	北京六区	北京七区
		北京八区
	北京七区	北京五区
广州	广州三区	广州四区
	广州四区	广州六区
	广州六区	广州四区
		广州七区
	广州七区	广州四区
上海	上海二区	上海四区
	上海四区	上海二区
	上海五区	上海四区

中国香港	香港一区	香港二区
	香港二区	香港三区
新加坡	新加坡二区	新加坡四区
	新加坡三区	新加坡四区
	新加坡四区	新加坡三区
硅谷	硅谷二区	硅谷一区
法兰克福	法兰克福一区	法兰克福二区
	法兰克福二区	法兰克福一区
弗吉尼亚	弗吉尼亚一区	弗吉尼亚二区
	弗吉尼亚二区	弗吉尼亚一区
东京	东京一区	东京二区
	东京二区	东京一区
雅加达	雅加达二区	雅加达一区

如何实现多可用区架构

关于多可用区架构，可通过控制台新建集群实现，存量的单可用区集群也会升级至多可用区集群，该升级通过在线迁移数据的方式自动完成，对您的业务无任何影响。详情请参见 设置多可用区部署 。

多可用区部署费用说明

多可用区部署功能暂时不需要支付额外费用。

> **说明：**
> 当前单可用区集群也可免费升级为至多可用区集群。

多可用区信息展示

在集群列表页面，根据实际使用的视图模式进行展示：

页签视图

1. 在 集群列表 页左侧单击集群，进入集群管理页。
2. 在集群管理页部署方式处，以及集群详情的拓扑图里可以分别看到该集群的部署方式以及主备可用区。



列表视图

1. 集群列表 页展示集群可用区信息（展示主可用区）。



2. 在集群详情页的**基本信息**和**可用性信息**处可查看集群部署分布的可用区。



TDSQL-C MySQL 版支持的数据复制方式

一、异步复制

应用发起数据更新（含 insert、update、delete 操作）请求，Master 在执行完更新操作后立即向应用程序返回响应，然后 Master 再向 Slave 复制数据。数据更新过程中 Master 不需要等待 Slave 的响应，因此异步复制的数据库实例通常具有较高的性能，且 Slave 不可用并不影响 Master 对外提供服务。但因数据并非实时同步到 Slave，而 Master 在 Slave 有延迟的情况下发生故障则有较小概率会引起数据不一致。

二、半同步复制

应用发起数据更新（含 insert、update、delete 操作）请求，Master 在执行完更新操作后立即向 Slave 复制数据，Slave **接收到数据并写到 relay log 中（无需执行）**后才向 Master 返回成功信息，Master 必须在接受到 Slave 的成功信息后再向应用程序返回响应。
仅在数据复制发生异常（Slave 节点不可用或者数据复制所用网络发生异常）的情况下，Master 会暂停（默认10秒左右）对应用的响应，将复制方式降为异步复制。当数据复制恢复正常，将恢复为半同步复制。

三、强同步复制

应用发起数据更新（含 insert、update、delete 操作）请求，Master 在执行完更新操作后立即向 Slave 复制数据，**Slave 接收到数据并写到 relay log 中（执行完成）**后才向 Master 返回成功信息，Master 必须在接受到 Slave 的成功信息后再向应用程序返回响应。
在数据复制发生异常（Slave 节点不可用或者数据复制所用网络发生异常）的情况下，**复制方式均不会发生降级**，为保障数据一致性，此时 Master 会暂停对应用的响应，直至异常结束。

版权所有：腾讯云计算（北京）有限责任公司

设置多可用区部署

最近更新时间：2025-04-11 17:41:54

TDSQL-C MySQL 版支持在创建集群时选择主可用区和备可用区，如果在创建集群时选择了单可用区部署，也支持在创建后设置为多可用区部署，本文为您介绍通过购买页和控制台设置多可用区部署。

操作场景

TDSQL-C MySQL 版支持同地域下跨多个可用区部署，相对单可用区部署，多可用区部署具有更高的可用性和容灾能力。
多可用区部署为 TDSQL-C MySQL 版集群提供增强的可用性和持久性。当您预置多可用区数据库集群时，TDSQL-C MySQL 版会自动创建主数据库实例，并将数据同步复制到其他可用区中的备用实例。每个可用区在其独立的、物理上显著不同的基础设施中运行，并已设计为具备高可靠性。如果发生基础设施故障，可自动故障转移至备用实例中，以便您能够在故障转移结束后立即恢复数据库操作。由于故障转移后数据库实例的终端节点维持不变，因此应用程序可在无需手动管理干预的情况下恢复数据库操作。

创建集群时在购买页设置多可用区部署

1. 登录 TDSQL-C MySQL 版控制台，在集群列表，单击**新建**。
2. 在购买页的**数据库配置**选项，选择对应地域，在下方会显示可设置的主可用区以及备可用区区域，根据需要进行选择。
3. 数据库配置选择完毕后，单击**下一步**，进入**基础信息**和**高级配置**选项设置。
4. 设置完成并确定无误后，单击**立即购买**。
5. 购买完成后，可在对应的视图下，查看可用区信息。

页签视图

购买完成后，返回集群列表，待新创建的集群状态变更为**运行中**，可通过集群管理页里，集群详情下的拓扑图查看可用区信息。



列表视图

购买完成后，返回集群列表，待新创建的集群状态变更为**运行中**，可通过集群列表页或单击**集群 ID** 进入**集群详情**页的**可用性信息**，查询可用区。



创建集群后在控制台设置多可用区部署（前提：创建时为单可用区部署）

1. 在集群列表页面，根据实际使用的视图模式操作，进入修改部署方式界面。

页签视图

1. 登录 TDSQL-C MySQL 版控制台，在左侧集群列表，单击目标集群，进入集群管理页。
2. 在集群管理页，部署方式后单击编辑图标。



列表视图

1. 登录 TDSQL-C MySQL 版控制台，在集群列表，单击**集群 ID** 或**操作**列的**管理**，进入集群详情页面。
2. 在集群详情页的**可用性信息**模块，单击**部署方式**后的**修改**，即可对可用区进行修改更换。



2. 在弹出的对话框，完成如下配置，单击**确定**。

> **说明：**
> 目前仅支持对备可用区进行修改，主可用区暂不支持修改变更。



参数	说明
多可用区部署	选择“是”，表示将集群从单可用区设置为多可用区。
备可用区	选择备库的可用区，具体地域支持选择的可用区以实际**修改部署方式**窗口为准。
binlog 复制方式	选择 binlog 复制的方式，默认为异步复制，支持异步复制、半同步复制、强同步复制三种模式。了解复制方式，请参见 多可用区部署概述 。

> **说明：**
> 若选择 binlog 复制方式为强同步复制，需注意强同步复制会对实例的性能会产生影响。

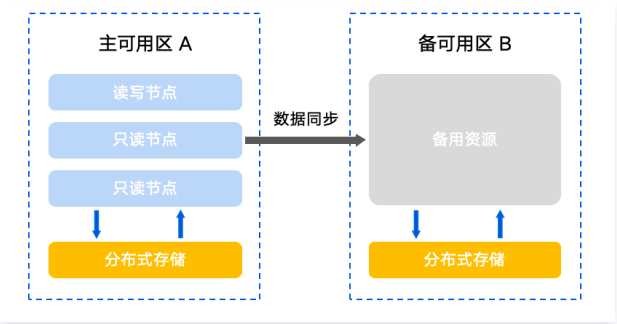
切换可用区

最近更新时间：2024-07-02 17:05:32

TDSQL-C MySQL 版支持切换集群的主可用区和备可用区，即在集群出现故障时，可将备用数据库切换为主数据库，以保证系统的可用性和数据的完整性，本文为您介绍通过控制台进行可用区切换。

使用场景

可用区切换功能适用于灾难恢复、设备故障等场景。



注意事项

- 主备可用区切换过程中，会出现2秒 – 5秒左右的连接闪断。建议您在业务低谷期执行切换操作并确保应用具备重连机制。
- 如果目标可用区是备可用区，则不需要迁移数据。系统只需要切换数据库计算节点，因此可以达到比较快的跨机房切换效果，该操作常用于容灾演练。
- 集群整体发生主备切换，即集群下包括读写节点、只读节点均发生切换（不支持仅部分节点切换）。
- 切换后读写地址、只读地址均不发生改变。

操作步骤

自动切换说明

本功能支持在发生故障的情况下，集群的主可用区和备可用区可自动进行切换，以确保您的业务能正常运行，保护数据库，除此之外，您还可以参考如下步骤进行手动切换可用区。

手动切换可用区

如您因业务需要对集群主备可用区立即切换或指定时间切换，可进行手动切换可用区操作。
在集群列表页面，根据实际使用的视图模式进行操作：

页签视图

1. 登录 TDSQL-C MySQL 版控制台，在左侧集群列表，单击目标集群，进入集群管理页。
2. 在集群管理页，集群详情下拓扑图中的主备可用区中间，单击切换图标，切换主备可用区。



3. 在弹出的对话框，选择**立即切换**或**维护时间内切换**，勾选“可用区切换时，会有秒级别的连接闪断，请确保业务具备重连机制”，单击**确定**。

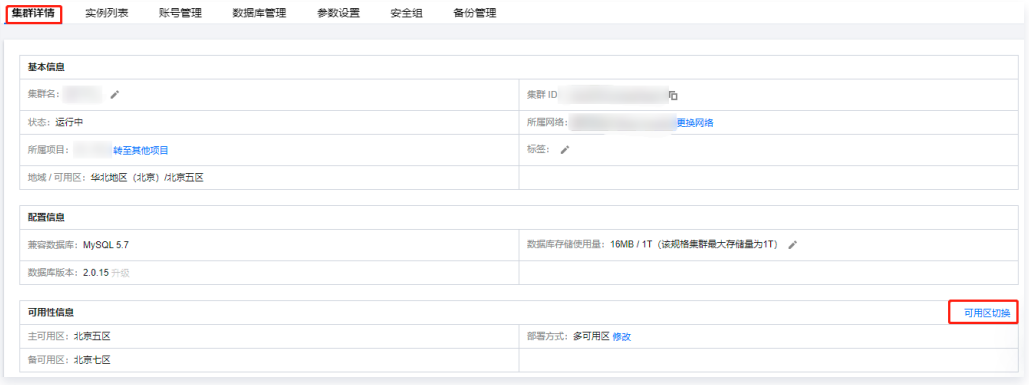
> **说明：**
> - 可用区切换时，会有秒级别的连接闪断，请确保业务具备重连机制。
> - 选择**实例列表**页，单击对应实例 ID，可进入实例详情页，在**维护信息**后单击编辑可设置修改维护的时间。



4. 待集群状态为**运行中**时，可在**集群详情**下的架构图中查看切换的可用区。

列表视图

1. 登录 TDSQL-C MySQL 版控制台，在集群列表，单击**集群 ID** 或**操作**列的**管理**，进入集群详情页。
2. 在集群详情页的**可用性信息**模块，单击**可用区切换**。



3. 在弹出的对话框，选择**立即切换**或**维护时间内切换**，勾选“可用区切换时，会有秒级别的连接闪断，请确保业务具备重连机制”，单击**确定**。

> **说明：**
> - 可用区切换时，会有秒级别的连接闪断，请确保业务具备重连机制。
> - 选择**实例列表**页，单击对应实例 ID，可进入实例详情页，在**维护信息**后单击编辑可设置修改维护的时间。



4. 待集群状态为**运行中**时，可在**集群详情页**可用性信息处查看切换的可用区。



常见问题

需要多长时间才能完成可用区的切换?

若您选择的目标可用区是备可用区，则更换主可用区时不需要迁移数据。系统只需要切换数据库计算节点，因此可以达到比较快的跨机房切换效果。

集群整体发生切换后读写地址是否变更?

集群整体发生切换时，集群下的读写节点、只读节点均发生切换，切换后读写地址、只读地址均不发生改变。

手动切换/自动切换后，如果对实例进行升级、升配等操作，是否会影响或者重置主备可用区信息?

切换后不会影响和重置主备可用区信息。

变更可用区部署

最近更新时间：2024-04-12 09:21:41

本文为您介绍如何通过 TDSQL-C MySQL 版控制台升级单可用区为多可用区，或从多可用区降级为单可用区。

版权所有：腾讯云计算（北京）有限责任公司

升级单可用区为多可用区

在集群列表页面，根据实际使用的视图模式进行操作：

页签视图

1. 登录 TDSQL-C MySQL 版控制台，在左侧集群列表，单击目标集群，进入集群管理页。
2. 在集群管理页，部署方式后单击。
3. 在弹出的对话框，**多可用区部署**选择**是**，选择备可用区和 binlog 复制方式，单击**确定**。

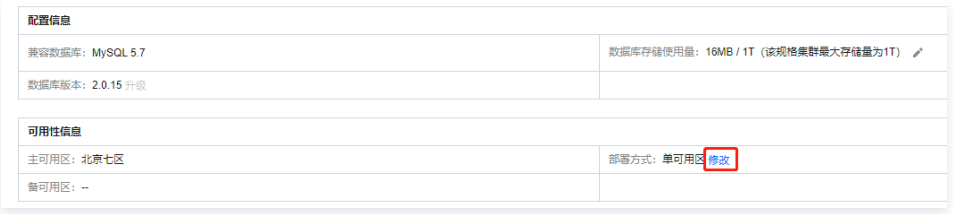


参数	说明
多可用区部署	选择“是”，表示将集群从单可用区设置为多可用区。
备可用区	选择备库的可用区，具体地域支持选择的可用区以实际**修改部署方式**窗口为准。
binlog 复制方式	选择 binlog 复制的方式，默认为异步复制，支持异步复制、半同步复制、强同步复制三种模式。 **说明：** 需注意强同步复制会对实例的性能会产生影响。

4. 待集群状态为**运行中**时，可在集群详情下的架构图中查看主备可用区信息。

列表视图

1. 登录 TDSQL-C MySQL 版控制台，在集群列表，单击集群 ID 或**操作**列的**管理**，进入集群详情页。
2. 在集群详情页的**可用性信息**模块，单击**部署方式**后的**修改**。



3. 在弹出的对话框，**多可用区部署**选择**是**，选择备可用区和 binlog 复制方式，单击**确定**。



参数	说明
多可用区部署	选择“是”，表示将集群从单可用区设置为多可用区。
备可用区	选择备库的可用区，具体地域支持选择的可用区以实际**修改部署方式**窗口为准。
binlog 复制方式	选择 binlog 复制的方式，默认为异步复制，支持异步复制、半同步复制、强同步复制三种模式。 **说明：** 需注意强同步复制会对实例的性能会产生影响。

4. 待集群状态为**运行中**时，可在集群详情页查看主备可用区信息。

降级多可用区为单可用区

在集群列表页面，根据实际使用的视图模式进行操作：

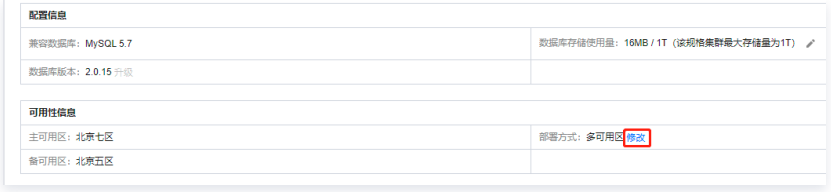
页签视图

1. 登录 TDSQL-C MySQL 版控制台，在左侧集群列表，单击目标集群，进入集群管理页。
2. 在集群管理页，部署方式后单击。
3. 在弹出的对话框，**多可用区部署**选择**否**，单击**确定**。

> **说明：**
> 由单可用区修改为多可用区或多可用区修改为单可用区，以上操作仅支持修改“备可用区”，即主可用区不会变更。

列表视图

1. 登录 TDSQL-C MySQL 版控制台，在集群列表，单击集群 ID 或**操作**列的**管理**，进入集群详情页。
2. 在集群详情页的**可用性信息**模块，单击**部署方式**后的**修改**。



3. 在弹出的对话框，**多可用区部署**选择**否**，单击**确定**。

> **说明：**

> 由单可用区修改为多可用区或多可用区修改为单可用区，以上操作仅支持修改“备可用区”，即主可用区不会变更。

常见问题

集群的可用区信息可以在哪里查询到?

列表视图

- 在集群列表，可用区列可查询。
- 在集群详情页，基本信息，地域/可用区项可以查询。
- 在集群详情页，可用性信息项可以查询。

页签视图

- 在集群详情下的架构图中可查询。

在修改可用区操作中，集群能否保持正常运行?

- 修改可用区任务开始和修改过程中，不会影响只读实例以及读写实例的正常访问。
- 修改可用区后，数据库连接地址（集群读写地址和只读地址）不变，因此您仍然可通过数据库连接地址正常访问集群。

可用区迁移

最近更新时间：2025-05-29 16:04:22

当需要对集群进行业务区域转移时，您可以通过可用区迁移功能将 TDSQL-C MySQL 版的主可用区迁移至同地域的其他可用区。可用区迁移后，集群的属性、配置和连接地址均不会改变。本文为您介绍通过控制台进行可用区迁移的操作。

前提条件

- 集群状态为运行中。
- 集群所在地域存在多个可用区。
- 集群的实例形态为预置资源。

功能说明

- 可用区迁移后，集群的 VIP 地址保持不变。
- 集群下存在只读实例时，也支持使用可用区迁移功能。
- 无论集群采用单可用区部署还是多可用区部署，均支持使用可用区迁移功能。
- 切换可用区时会受到短暂影响，请确保应用具备自动重连机制。

使用限制

- 开启 SSL 加密的集群不支持可用区迁移，如需进行可用区迁移，请先关闭 SSL 加密后再进行迁移。
- 开启 TDE 加密的集群不支持可用区迁移。
- binlog 设置关闭状态，不支持可用区迁移。
- 实例形态为 Serverless 的集群不支持可用区迁移。
- 可用区迁移时，若目标可用区库存不足，则不支持迁移，请更换其他可用区进行迁移。
- 开启数据库代理的集群不支持可用区迁移，如需进行可用区迁移，请先关闭数据库代理后再进行迁移。
- 可用区迁移操作过程中的多可用区部署设置，可选择的主备可用区请参见 支持的地域和可用区 。

操作步骤

1. 在集群列表页面，根据实际使用的视图模式操作，进入迁移可用区设置。

页签视图

1. 登录 TDSQL-C MySQL 版控制台，在左侧集群列表，单击**目标集群**，进入集群管理页。
2. 在集群管理页右侧，单击**可用区迁移**。



列表视图

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在集群列表找到目标集群，单击其**操作**列的**更多** > **可用区迁移**。

2. 在**可用区迁移**弹窗下，完成如下配置，单击**确定**。



参数	说明
当前地域/可用区	默认显示当前所属地域/可用区。
目标可用区	选择要迁移至的目标可用区，所属地域不变。 **说明：** • 若要进行可用区迁移，则目标可用区的选择和原可用区不允许一致。 • 若选择目标可用区后，提示“当前可用区不支持迁移”，则表示当前可用区库存不足，请更换其他可用区进行迁移。
多可用区部署	• 选择“是”，表示将集群设置为多可用区部署。 • 选择“否”，表示将集群设置为单可用区部署。
目标备可用区	当**多可用区部署**选项选择为“是”时，需要选择目标备可用区，表示将集群设置为多可用区部署，主可用区和备可用区为不同的可用区。
数据校验延迟阈值	当**目标可用区**选项（即主可用区）变更时，此设置才会出现。阈值为1秒 – 10秒的整数。 **说明：** 数据一致性校验过程中可能产生延迟，需设置数据校验延迟阈值，当延迟超过设定值时，数据库一致性校验会暂停，直到恢复至指定阈值以下后会继续进行数据库一致性校验任务。该阈值设置越小，可能导致升级完成时间过长。
切换时间	选择可用区迁移的切换时间。 • 维护时间内：表示在实例的维护时间内进行切换，设置实例维护时间请参见 修改实例维护时间 。 • 升级完成时：表示可用区迁移升级完成时立即进行切换。

全球数据库
概述

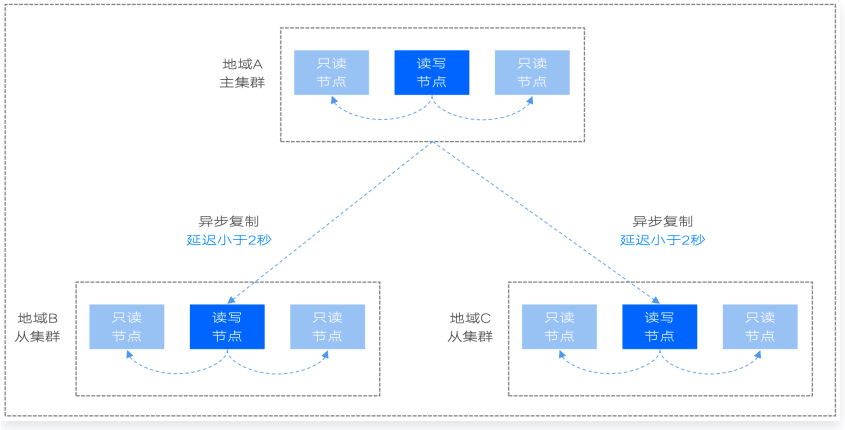
最近更新时间：2025-06-03 14:56:42

TDSQL-C MySQL 版支持全球数据库，可帮助企业构建高可用、低延迟、强一致的全球数据库架构。本文为您介绍全球数据库功能。

> **说明：**
> 全球数据库功能当前处于内测阶段，如需体验请 提交工单 申请。

什么是全球数据库

全球数据库（Global Database，简称 GD）是由分布于多个地域的多个 TDSQL-C MySQL 版集群构成的网络。在该网络中，所有集群的数据保持同步，且每个集群均可提供读服务，而写请求将被转发至主集群进行处理。



功能背景

随着企业数字化转型加速，越来越多的用户将核心业务系统迁移上云。在此过程中，用户数据量呈现指数级增长，业务逻辑也日趋复杂化。尽管 TDSQL-C MySQL 版凭借其云原生架构已能满足大部分业务场景的需求，但在全球化运营场景下，基于单可用区的传统部署模式逐渐显露出局限性，当用户需要在全球多个地域部署业务系统时，单可用区架构往往难以满足跨地域的实时查询需求。
针对这一挑战，传统方案通常采用第三方数据同步工具，通过在不同地域的集群间建立连接实现数据同步。然而，这类方案在实际应用中面临诸多痛点：首先，数据同步速度受网络带宽和工具性能限制，难以满足低延迟业务场景；其次，引入第三方工具会增加额外的采购和维护成本；更重要的是，此类方案在数据实时性和一致性方面存在技术瓶颈，难以适应对数据时效性要求严苛的核心业务系统。
为彻底解决上述问题，TDSQL-C MySQL 版推出 GD 功能。该功能专为全球化业务场景设计，通过云原生架构实现跨地域数据协同，从根本上突破传统方案的局限性，为企业构建高性能、高可用的全球数据库体系提供全栈解决方案。

功能优势

- **无需修改业务代码**：可以直接从同城部署扩展到多地部署。
- **跨地域读写分离、就近读取**：GD 内读请求被直接发送到本地集群，写请求转发至主集群。
- **配置灵活**：主、从集群拥有相互独立的配置，例如集群规格和参数值等，均可以拥有独立配置。
- **跨地域低延时同步**：通过多通道物理复制，高压力下全球同步延迟小于2秒。

应用场景

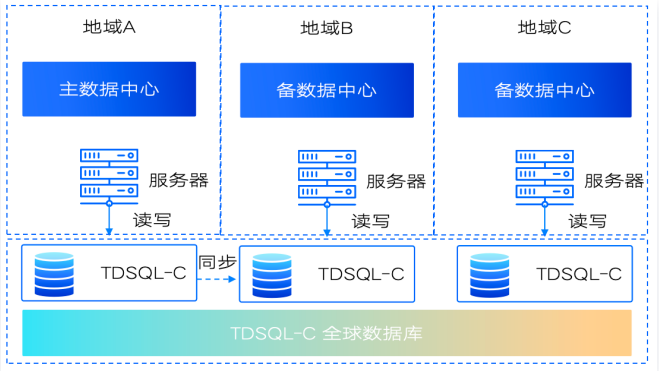
GD 功能支持的典型应用场景是异地多活。下面为您介绍异地多活场景的业务架构和部署流程。

异地多活场景业务架构

在传统网络架构中，若业务系统采用单地域数据库部署（主库位于单一地域），其他地域的应用需通过跨地域访问该数据库。受网络延迟和带宽波动影响，长距离数据交互会导致查询响应时间显著增加，甚至出现超时错误，直接影响终端用户体验。使用全球数据库，可以打通全球数据库下的数据，这样全球地域都能访问同一个数据库，读请求尽量发往本地集群，而写请求会被转发到主集群进行处理。通过全球数据库的跨地域低延迟同步、跨地域读写分离、本地就近读取等特性，可以确保各地域的应用访问数据库时的延迟小于2秒。

- **典型行业**：游戏、跨境电商、金融、新零售（门店）等。

- **业务架构说明：**
  - 各地域的应用程序均直接就近读写各地域的数据库，写请求会被转发到主集群进行处理。
  - 在 GD 中，主集群和从集群均提供独立的集群地址。您可以根据应用所在的地域，连接就近的集群地址以实现访问。
  - 地域 B、地域 C 的集群规格无需与地域 A 保持一致，可以灵活选择。



异地多活场景部署流程

1. 创建一个 GD，将已有集群作为 GD 中的主集群。详细操作步骤，请参见 创建全球数据库 。
2. 在 GD 中，添加新集群作为从集群。详细操作步骤，请参见 添加从集群 。
3. 连接 GD。详细操作步骤，请参见 连接全球数据库 。

支持形态及版本说明

- 支持实例形态：预置资源。实例形态为 Serverless 的集群暂不支持。
- 支持版本：内核小版本为 TDSQL-C MySQL 版8.0 3.1.16.100及之后的版本。升级内核小版本请参见 升级内核小版本 。

功能计费说明

当前 GD 功能的跨地域传输流量为免费使用，您仅需支付每个集群的费用，了解集群费用请参见 产品价格 。

功能限制说明

- 此功能当前支持的数据库版本仅为 TDSQL-C MySQL 8.0。
- 一个 GD 包含1个主集群和最多2个从集群。
- 主集群开启 GD 前，需开启数据库代理，从集群默认会开启数据库代理，且 GD 功能开启期间，数据库代理不支持关闭。
- 主集群和从集群共用一份全局参数。
- 开启 GD 功能，需要读写实例规格在4核及以上。

支持地域及可用区

地域	可用区
上海	上海二区
中国香港	香港一区
法兰克福	法兰克福一区
新加坡	新加坡四区

相关文档

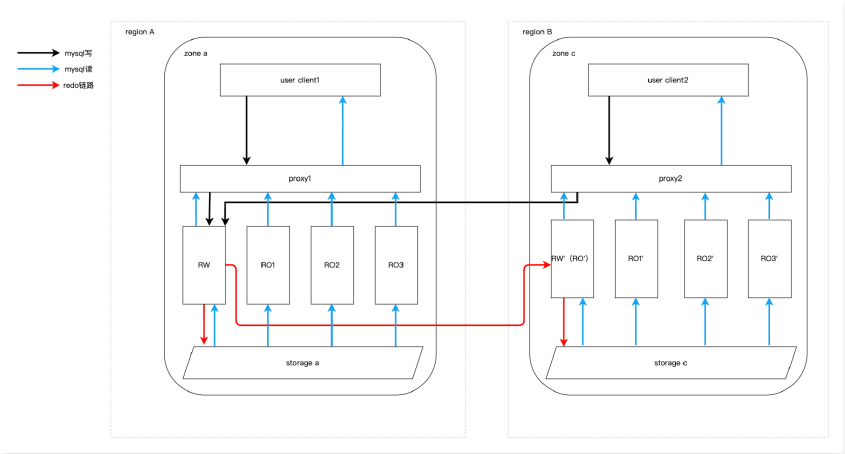
- 技术架构
- 使用功能
- 全球数据库实践教程

技术架构

最近更新时间：2025-06-03 14:56:42

本文为您介绍全球数据库（Global Database，简称 GD）的技术架构。

技术架构图



- 主集群与从集群允许部署在不同地域。
- 读取数据采用就近读取原则，优先从本地域读取数据。
- 写入数据时，主集群会直接写入本地存储，从集群通过数据库代理转发写入主集群。

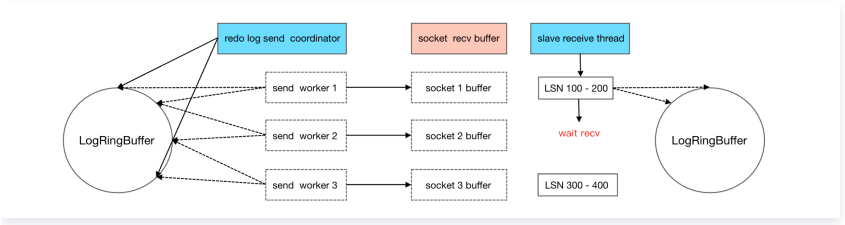
功能难点解析及优化

在全球数据库架构中，各地域的主从集群间采用 Redo Log 机制实现数据同步复制，以确保强一致性和高可用性。为了保障从集群的数据延迟严格控制在2秒以内，功能必须具备两项核心能力：

- **超高的日志传输速度**：通过高性能跨地域数据传输通道，确保海量 Redo Log 能快速同步至异地集群。
- **高延迟网络下的带宽优化**：在长距离、高延迟的跨地域网络环境下，能提升有效吞吐量，避免网络拥塞影响同步效率。

技术方案

并行乱序日志传输。主集群将日志分为三个 Worker 线程并行乱序发送，多条网络连接提高日志传输吞吐，优化网络传输带宽瓶颈，从集群按顺序接受日志进行回放，生成数据。



优化效果

延迟由1分钟降低至2秒以内。

使用功能
创建全球数据库

最近更新时间：2025-06-03 15:14:51

本文为您介绍通过控制台创建全球数据库（Global Database，简称 GD）的相关操作及说明。

> **说明：**
> 全球数据库功能当前处于内测阶段，如需体验请 提交工单 申请。

前提条件

已 创建集群 作为全 GD 的主集群，主集群在创建过程中需开启数据库代理，且读写实例规格在4核及以上。

注意事项

- 在创建从集群的过程中默认会开启数据库代理。
- GD 功能开启期间，GD 内主集群和从集群的数据库代理不支持关闭。

支持形态及版本说明

- 支持实例形态：预置资源。实例形态为 Serverless 的集群暂不支持。
- 支持版本：内核小版本为 TDSQL-C MySQL 版8.0 3.1.16.100及之后的版本，升级至此版本当前需 提交工单 获取解决方案。升级内核小版本请参见 升级内核小版本 。

功能计费说明

当前 GD 功能的跨地域传输流量为免费使用，您仅需支付每个集群的费用，了解集群费用请参见 产品价格 。

功能限制说明

- 此功能当前支持的数据库版本仅为 TDSQL-C MySQL 8.0。
- 一个 GD 包含1个主集群和最多2个从集群。
- 主集群和从集群共用一份全局参数。
- 开启 GD 功能，需要读写实例规格在4核及以上。
- GD 的主集群和从集群均不支持 Serverless 配置功能、不支持转 Serverless 操作。

支持地域及可用区

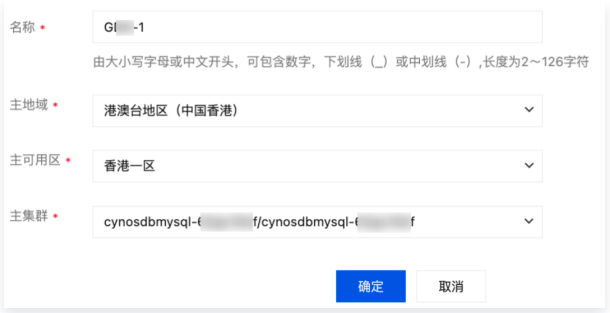
支持地域及可用区情况请参见 概述 。

创建全球数据库

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在左侧导航栏中单击**全球数据库**。
3. 在界面中单击**立即创建**。



4. 在弹窗中完成如下配置，单击**确定**。



参数	说明
名称	GD 的名称，建议配置具有业务意义的名称（后续支持修改），便于识别。输入格式：由大小写字母或中文开头，可包含数字，下划线（_）或中划线（-），长度为2～126字符。
主地域	选择主集群所在的地域。 **说明：** 请先确认要选择哪个集群作为主集群，并在此选择该集群所在的地域。
主可用区	选择主集群所在地域下的主可用区。
主集群	选择一个已有集群作为 GD 的主集群。 **说明：** 主集群需已开启数据库代理，且读写实例规格在4核及以上。

5. （可选）全球数据库创建成功后，您可以添加从集群，操作步骤请参见 管理从集群 。

管理从集群

最近更新时间：2025-06-03 14:56:42

本文为您介绍创建全球数据库（Global Database，简称 GD）后，对 GD 下的集群进行管理和变更配置说明。

前提条件

已 创建全球数据库 。

注意事项

- 一个 GD 包含1个主集群和最多2个从集群。
- 主集群和从集群的数据库版本一致，均为 TDSQL-C MySQL 8.0。
- 一个集群只能属于一个 GD。
- 从集群只能新建，不能将已经创建好的集群添加为从集群。
- 从集群的实例形态默认仅支持预置资源。
- GD 中的从集群的计算实例（读写）的规格要求至少为4核及以上，计算实例（只读）的规格无核数限制。
- 若 GD 中的主集群的存储计费模式为包年包月，则从集群的存储计费模式支持选择包年包月或按量计费，若 GD 中的主集群的存储计费模式为按量计费，则从集群的存储计费模式仅支持选择按量计费。

从集群计费说明

当前 GD 功能的跨地域传输流量为免费使用，您仅需支付每个集群的费用，了解集群费用请参见 产品价格 。

配置变更说明

由于 GD 下的集群会进行数据同步，为保证从集群数据同步的完整性和一致性，同一 GD 下从集群的存储空间不能小于主集群的存储空间，因此在对 GD 下集群进行配置变更时有较多操作注意点。详细操作及说明请参见 配置变更 。

添加从集群

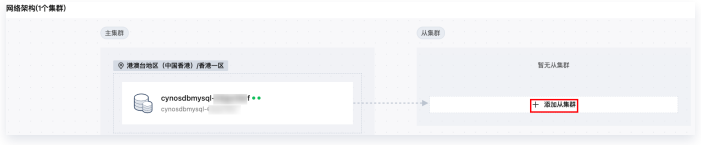
方法一、在主集群管理页添加从集群

1. 登录 TDSQL-C MySQL 版控制台 。
2. 根据实际控制台视图模式，对应操作。

页签视图
在左侧集群列表，单击目标主集群，进入集群管理页。

列表视图
在左侧集群列表找到目标主集群，单击**集群 ID** 或**操作**列的**管理**，进入集群管理页。

3. 在集群管理页，选择**全球数据库**，在网络架构下单击**添加从集群**。



4. 在跳转的“创建全球数据库从集群”页面中，根据需要选择从集群的配置，单击**立即购买**。

> **说明：**
> - 在“创建全球数据库从集群”页面中的详细步骤可以参考 创建集群 ，可选择的地域及可用区请以实际页面为准。
> - 在创建从集群时，从集群的数据库代理必须开启。

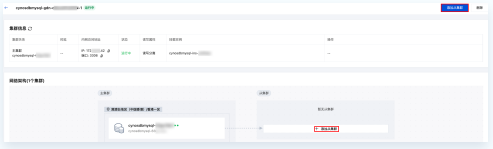


5. 等待从集群发货完成后，即完成添加从集群操作。

方法二、在全球数据库下添加从集群

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在左侧导航栏中单击**全球数据库**。
3. 找到目标全球数据库，在其操作列单击**添加从集群**。



> **说明：**
> 在全球数据库的详情页内，也有**添加从集群**的入口。
>
> 

4. 在跳转的“创建全球数据库从集群”页面中，根据需要选择从集群的配置，单击**立即购买**。

> **说明：**
> - 在“创建全球数据库从集群”页面中的详细步骤可以参考 创建集群 ，可选择的地域及可用区请以实际页面为准。
> - 在创建从集群时，从集群的数据库代理必须开启。

5. 等待从集群发货完成后，即完成添加从集群操作。

移除从集群

> **注意：**
> - 仅当 GD 下有从集群存在，才支持移除从集群操作。
> - 移除从集群过程中，GD 下所有集群（包括正在脱离的从集群）的连接地址仍然可用，您仍然可以通过集群的连接地址正常访问数据库。
> - 主集群无法从 GD 中移除。
> - 移除从集群后，该从集群不会被删除，而是会中断与主集群的数据同步，同时系统将设置该集群为独立集群。
> - 从 GD 移除后，该集群不能再作为从集群加入 GD，请谨慎操作。

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在左侧导航栏中单击**全球数据库**。
3. 找到目标全球数据库，在其操作列单击**移除从集群**。

> **说明：**
> 如果您当前在主集群或者从集群管理页面下的全球数据库页，您可单击网络架构下对应从集群下的**脱离**，来移除从集群。
>
> 
>

4. 在弹窗中，单击**确定**即可。

配置变更

最近更新时间：2025-06-03 14:56:42

由于全球数据库（Global Database，简称 GD）下的集群会进行数据同步，为保证从集群数据同步的完整性和一致性，同一 GD 下从集群的存储空间不能小于主集群的存储空间，因此在对 GD 下集群进行配置变更时有较多操作注意点。本文为您介绍全球数据库（GD）中集群配置变更的操作及说明。

前提条件

- 已 创建全球数据库 。
- 已为 GD 添加从集群，操作参见 管理从集群 。

存储变配策略

主集群存储空间计费模式	从集群存储空间计费模式	变配策略说明
包年包月	包年包月	从集群存储空间 ≥ 主集群存储空间
包年包月	按量计费	从集群计算规格对应的存储空间上限 ≥ 主集群计算规格对应的存储空间上限
按量计费	按量计费	从集群计算规格对应的存储空间上限 ≥ 主集群计算规格对应的存储空间上限

变配说明

- 如果因主集群读写实例变更计算规格而影响到存储变化，则从集群读写实例的计算规格需要同步调整，调整至满足上表中的变配策略即可。
- 对主集群读写实例进行配置变更时，若存在从集群读写实例存储空间小于主集群读写实例存储空间的情况，则需在从集群读写实例全部变配成功后，主集群读写实例的配置变更才可以操作。

场景一：主集群存储空间计费模式为包年包月，从集群存储空间计费模式为包年包月

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在左侧导航栏中单击**全球数据库**。
3. 找到目标全球数据库，在“集群列表”字段下单击要变更配置的主集群 ID 快捷跳转至对应主集群管理页。
4. 在右侧读写实例下单击**调整配置**。
5. 在调整配置页面完成配置变更，单击**立即购买**。

参数	说明
实例类型	选择所需实例类型，支持通用型和独享型。
实例规格	选择所需实例规格。
变配数据库代理	选择是否变配数据库代理。 • 选择是：需对代理规格，节点数和可用区进行调整。 • 选择否：表示不变配数据库代理。
操作时间	选择变配成功的切换时间，支持选择维护时间内或升级完成时。
平滑变配	选择是否开启平滑变配，开启平滑变配时，变配不会发生闪断，升降配即时生效，特殊情况下若发生闪断将停止变配。
存储空间	设置存储空间。 • 如果主集群存储空间小于或等于从集群存储空间，则可以操作变配。 • 如果主集群存储空间大于从集群存储空间，则不可以操作变配。 

说明：

主集群存储空间计费模式为包年包月，从集群存储空间计费模式为包年包月时，在**计费模式**后单击编辑图标，设置存储空间大于从集群存储空间时，需先在 GD 信息列表中单击**变配**，将从集群存储空间先调整成目标大小，再调整主集群存储空间。

计费模式后编辑图标位置如下图。

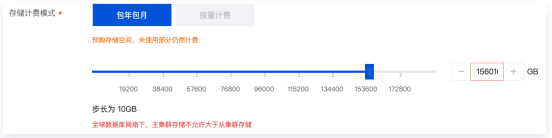


单击编辑图标后，从集群变配位置如下图。



场景二：主集群存储空间计费模式为包年包月，从集群存储空间计费模式为按量计费

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在左侧导航栏中单击**全球数据库**。
3. 找到目标全球数据库，在“集群列表”字段下单击要变更配置的主集群 ID 快捷跳转至对应主集群管理页。
4. 在右侧读写实例下单击**调整配置**。
5. 在调整配置页面完成配置变更，单击**立即购买**。

参数	说明
实例类型	选择所需实例类型，支持通用型和独享型。
实例规格	选择所需实例规格。
变配数据库代理	选择是否变配数据库代理。 • 选择是：需对代理规格，节点数和可用区进行调整。 • 选择否：表示不变配数据库代理。
操作时间	选择变配成功的切换时间，支持选择维护时间内或升级完成时。
平滑变配	选择是否开启平滑变配，开启平滑变配时，变配不会发生闪断，升降配即时生效，特殊情况下若发生闪断将停止变配。
存储空间	设置存储空间。 • 如果所选实例规格对应的存储空间上限小于或等于从集群实例规格对应的存储空间上限，则可以操作变配。 • 如果所选实例规格对应的存储空间上限大于从集群实例规格对应的存储空间上限，则不可以操作变配。如果需要调整至更大的存储空间，则需要先将从集群实例规格进行升配，再调整主集群的实例规格，关于实例规格和对应存储空间上限数据可参见 产品规格 。 

说明：

主集群存储空间计费模式为包年包月，从集群存储空间计费模式为按量计费时，如果将从集群存储空间计费模式变更为包年包月，则从集群存储空间需大于等于主集群存储空间才能操作。



场景三：主集群存储空间计费模式为按量计费，从集群存储空间计费模式为按量计费

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在左侧导航栏中单击**全球数据库**。
3. 找到目标全球数据库，在“集群列表”字段下单击要变更配置的主集群 ID 快捷跳转至对应主集群管理页。
4. 在右侧读写实例下单击**调整配置**。
5. 在调整配置页面完成配置变更，单击**立即购买**。

参数	说明
实例类型	选择所需实例类型，支持通用型和独享型。
实例规格	选择所需实例规格。 • 如果所选实例规格对应的存储空间上限小于或等于从集群实例规格对应的存储空间上限，则所选实例规格支持变配。 • 如果所选实例规格对应的存储空间上限大于从集群实例规格对应的存储空间上限，则不支持继续变配操作。关于实例规格和对应存储空间上限数据可参见 产品规格 。
变配数据库代理	选择是否变配数据库代理。 • 选择是：需对代理规格，节点数和可用区进行调整。 • 选择否：表示不变配数据库代理。
操作时间	选择变配成功的切换时间，支持选择维护时间内或升级完成时。
平滑变配	选择是否开启平滑变配，开启平滑变配时，变配不会发生闪断，升降配即时生效，特殊情况下若发生闪断将停止变配。
存储空间	存储空间默认为按量计费。

连接全球数据库

最近更新时间：2025-06-03 14:56:42

本文为您介绍查看全球数据库（Global Database，简称 GD）的集群地址并连接 GD 的相关操作。

前提条件

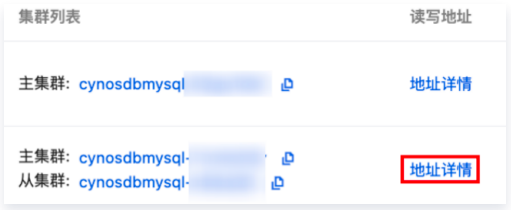
已 创建全球数据库 。

GD 连接地址说明

在 GD 架构中，主集群和每个从集群都提供了独立的集群地址，您可基于应用所在地域，选择连接地理位置最近的集群地址，即可快速接入 GD，系统支持自动读写分离机制，即读请求通常情况下会发送至本地集群，写请求则统一转发至主集群执行。

查看集群地址

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在左侧导航栏中单击**全球数据库**。
3. 找到目标全球数据库，在字段“读写地址”下单击**地址详情**。



4. 在弹出的侧边栏中，即可查看集群地址详情。



连接全球数据库

各个地域的应用可以使用本地集群的集群地址连接到 GD，GD 会自动进行读写分离。连接全球数据库的方法和连接集群的方法一致，详细请参见 连接集群 。

删除全球数据库

最近更新时间：2025-06-03 14:56:42

本文为您介绍删除全球数据库（Global Database Network，简称 GD）的操作及说明。

前提条件

已 创建全球数据库 。

注意事项

- GD 删除后，所有集群将断开关联，请谨慎处理。
- GD 删除后，主集群会变成独立集群，和普通预置资源集群具备一样的能力，从集群会隔离至回收站中，如需继续使用，可以在隔离期间 恢复实例 。
- GD 删除后，连接 GD 的应用程序将无法访问数据库，请及时修改应用程序的相关连接地址。

操作步骤

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在左侧导航栏中单击**全球数据库**。
3. 在列表中找到需要删除的全球数据库，单击其操作列的**删除**。



4. 在弹窗中，检查并确认集群信息，单击**确定**。

全球数据库实践教程

最近更新时间：2025-06-03 14:56:42

本文为您介绍跨地域部署全球数据库（Global Database，简称 GD）的完整流程和操作步骤。

部署背景

某电商企业业务覆盖多个地域（主要在中国香港，扩展地域为新加坡、法兰克福），随着业务的快速发展，原有的本地数据库架构在性能和业务连续性方面面临巨大挑战，不同地区用户访问延迟高，区域内的业务数据难以实时同步。现在需要通过全球数据库打通不同地域间的数据，使得全部地域都能访问同一个数据库，且读请求就近发往本地集群（即业务所在地域的集群），而写请求会被转发到主集群进行处理。

部署流程

1. 创建 GD 。
2. 添加从集群 。
3. 连接 GD 。

操作步骤

创建 GD

1. 在中国香港地域创建一个预置资源集群作为 GD 的主集群（若在该地域已有集群则无需再次创建），操作步骤参见 创建集群 。
2. 创建一个 GD，操作步骤参见 创建全球数据库 。

添加从集群

在 GD 中添加2个从集群，地域分别选择为新加坡、法兰克福，组成一个完整的 GD，操作步骤请参见 管理从集群 。

连接 GD

GD 创建成功后，您可以通过连接就近 TDSQL-C MyQL 版集群的连接地址，从而连接 GD 进行管理，详细操作步骤，请参见 连接全球数据库 。

版权所有：腾讯云计算（北京）有限责任公司

备份与恢复
备份与回档概述

最近更新时间：2024-08-20 10:05:31

数据是企业的核心资产。随着业务发展，企业数据呈现出规模化、爆炸式的增长，业务应用要求实时、在线的快速处理，对于数据库运维人员来说，保护企业核心数据的任务越来越有挑战性，例如数据误删除、相关系统漏洞和病毒、硬件故障，甚至自然灾害都可能造成数据的丢失，因此，备份和回档是数据库非常重要的功能。

备份概述

TDSQL-C MySQL 版支持数据备份和 binlog 备份，通过一个完整的数据备份以及后续一段时间的 binlog 备份，就可以将整个 TDSQL-C MySQL 版集群或特定库表恢复到任意时间点。
您可在备份管理页查询数据备份列表的字段信息，包括：文件名/备注名、备份时间点、开始时间、结束时间、备份类型、备份方式、备份大小、备份地域、状态以及操作等；可查询 binlog 备份列表的字段信息，包括：备份 ID、文件名、binlog 数据开始时间、binlog 数据结束时间、备份大小以及操作等，方便您对历史备份进行查询和管理。
备份与回档的控制台界面如下：



数据备份

数据备份即将某个时间点上集群的全量数据生成一个备份文件，TDSQL-C MySQL 版数据备份支持逻辑备份和快照备份两种备份类型，逻辑备份为全量备份，快照备份包含全量备份和增量备份。

- **逻辑备份**：仅支持全量备份，指的是保存数据库的逻辑结构和内容，是以 SQL 语句的方式来表示数据库的结构和数据，是数据库对象级备份，备份内容是表、索引、存储过程等数据库对象，其优势主要体现在能支持进行更精细的库、表级别备份，对数据库性能影响较小，但相对备份速度较慢，占用体积较大。
- **快照备份**：采用写时重定向技术 ROW（Redirect-On-Write），通过对存储层磁盘创建快照的方式进行备份，其优势主要体现在备份速度快，可秒级完成，整个过程计算层业务无感知，相对占用体积小。
- **全量备份**：对某　个时间点上的所有数据进行的　个完全拷贝。
- **增量备份**：在上一次备份的基础上，备份新增或改变的文件。

备份类型对比表

备份类型		优势/劣势	对象	方式	下载	删除
逻辑备份	全量备份	优势：精细的库、表级别备份，对数据库性能影响较小 劣势：会对数据库加锁，备份速度较慢，占用体积较大	表、索引、存储过程等数据库对象 整个集群	手动	✓	✓
				自动	✓	✓
快照备份	全量备份	优势：备份速度为秒级，业务无感知，占用体积小 劣势：备份不支持下载	整个集群	手动	×	×
				自动	×	×
	增量备份			自动	×	×

binlog 备份

备份类型		优势/劣势	对象	方式	下载	删除
binlog 备份	增量备份	优势：记录增量数据，可通过 binlog 备份恢复到任一时间点 劣势：开启 binlog 备份会降低实例写性能	整个集群	自动	✓	x

binlog 备份即在某个时间点上将集群的全量数据记录生成备份文件之后，产生的增量数据。TDSQL-C MySQL 版在执行大事务或大量 DML 时会产生较多的 binlog，此 binlog 备份会被上传至云端存储，然后控制台的 binlog 备份列表会显示上传到云端的 binlog 备份，binlog 备份用于存储操作日志。

版权所有：腾讯云计算（北京）有限责任公司

TDSQL-C MySQL 版不再依赖 binlog 备份进行回档，而采用 redo log 进行回档，在关闭 binlog 的情况下仍可正常回档到任意时间点，关闭 binlog 可以对实例的写性能有30%以上的提升。

> **说明：**
> 目前 TDSQL-C MySQL 版的网络架构正在逐步升级中，升级完成后将开放关闭 binlog 能力。

跨地域备份

TDSQL-C MySQL 版提供跨地域备份，支持将备份文件放到另一个地域存储，帮助用户提升监管与容灾恢复能力，同时提升数据的可靠性。目前自动逻辑备份、binlog 备份、手动逻辑备份（整个集群）类型支持使用跨地域备份能力。

注意事项

- 手动备份任务限制频率为1小时1次，自动备份任务根据用户设置的备份开始时间进行，系统默认每天进行一次自动备份。
- 备份方式中，手动备份的逻辑备份文件支持在备份列表手动删除，释放掉备份空间，避免空间浪费和占用，没有手动删除的前提下，备份文件会保留在备份列表下，为避免浪费，请定期手动删除不需要的部分。
- 备份方式中，自动备份产生的快照备份文件不支持手动删除，可通过设置自动备份保留时间来实现到期后自动删除；自动备份产生的逻辑备份文件支持手动删除，没有手动删除的前提下，备份文件会保留在备份列表下，为避免浪费，请定期手动删除不需要的部分。
- binlog 备份大小可在日志备份列表查询，binlog 备份的总大小为每个 binlog 备份文件大小之和。

回档概述

TDSQL-C MySQL 版支持通过回档功能来恢复数据到指定的时间，可以最大程度地减少系统损失。
TDSQL-C MySQL 版支持库表级别回档到原集群、库表级别回档至新集群以及整集群级别回档至新集群，用户可以根据业务选择不同的回档方式。

> **说明：**
> 库级别的回档，不会导入的对象类型包含：外键约束、check 约束、触发器、存储过程、用户视图、Event。

回档方式

- 按备份文件回档：将集群恢复到任一备份文件的数据文件状态，备份文件选择范围由用户设置的数据备份保留时长决定。
- 按时间点回档：将集群恢复到任意时间点，时间点的选择范围由用户设置的 binlog 备份保留时长决定。

版权所有：腾讯云计算（北京）有限责任公司

备份与回档费用说明

最近更新时间：2024-10-12 17:10:19

本文为您介绍备份与回档的费用说明。

计费

- **备份存储空间**：备份文件会占用一定的存储空间。备份方式包括自动备份和手动备份，备份对象包括 binlog 备份和数据备份，以上占用的空间均会算入此计费项。

  > **说明**
  > 备份空间暂时免费，随着业务发展和功能逐渐完善，未来可能实行商业化计费，我们将另行通知。

- **回档**：TDSQL-C MySQL 版支持根据备份文件或时间点回档到原集群或回档到新集群，其中回档到原集群不涉及新增计费，回档到新集群，新集群在回档成功之后开始计费，费用详情请参见 产品价格 。

执行备份
自动备份

最近更新时间：2025-01-15 10:06:43

TDSQL-C MySQL 版支持自动备份和手动备份两种方式来备份数据，自动备份中快照备份为默认开启，逻辑备份需要手动开启，您也可设置自动备份的备份策略，以达到长期保留备份的效果，保证数据的安全性。
集群创建之初，TDSQL-C MySQL 版根据日志生成速率和写入量等因素生成数据备份（自动备份中的快照备份），默认情况下每2日进行一次备份，如果写入量大于4G且距离上次备份超过24小时则立即进行备份。后续您也可以根据业务需求，在备份管理页开启自动备份中的逻辑备份，同时也支持对自动备份中逻辑备份的保留时间进行设置。
本文为您介绍通过控制台进行自动备份设置。

> **说明：**
> - 为保证备份系统的稳定，**当前系统每小时仅允许发起一次备份**。
> - 如果您的集群存储空间使用大于2TB，则不建议使用逻辑备份，否则会存在备份失败的风险。

注意事项

- 自动备份产生的快照备份不支持删除，您可以通过自动备份设置中的**数据备份保留**选项，来设置自动快照备份文件的保留时长，控制备份文件的生命周期，以此达到删除的效果。
- 您可在数据备份列表下查看**备份时间点**字段来确定该备份文件中数据镜像的确切时间点。

操作步骤

自动快照备份设置

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在上方选择地域，根据实际控制台视图模式，对应操作。

页签视图

在左侧集群列表，单击目标集群，进入集群管理页。



列表视图

在左侧集群列表找到目标集群，单击**集群 ID** 或**操作**列的**管理**，进入集群管理页。



3. 在集群管理页面，选择**备份管理**页，单击**自动备份设置**。



4. 在弹出的对话框，完成下列配置，单击**确定**。

参数	说明
数据备份保留	数据备份可以保留7天 - 1830天。
开始时间	默认情况下每2日进行一次备份，如果写入量大于4G且距离上次备份超过24小时则立即进行备份。
逻辑备份	若您仅需设置自动备份中的快照备份，则无需开启并设置逻辑备份。

开启自动逻辑备份

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在上方选择地域，根据实际控制台视图模式，对应操作。

页签视图

在左侧集群列表，单击目标集群，进入集群管理页。

列表视图

在左侧集群列表找到目标集群，单击**集群 ID** 或**操作**列的**管理**，进入集群管理页。



3. 在集群管理页面，选择**备份管理**页，单击**自动备份设置**。



4. 在逻辑备份下单击**开启并设置**。
5. 在弹出的对话框，完成下列配置，单击**确定**。



参数	说明
开始时间	设置自动逻辑备份每天的开始时间，默认为02:00-06:00。
自动备份周期	默认为每天。 **说明：** 当前版本仅支持自动备份周期为每天。由于逻辑备份速率限制，部分数据量较大的集群备份周期预计延长至1天以上（若当天逻辑备份未完成，则会在此逻辑备份完成日期的下一天才进行下一次自动逻辑备份）。
跨地域备份	选择是否开启跨地域备份。 • 开启后，可将自动逻辑备份文件保存到其他地域。详细说明请参见 跨地域备份 。 • 未开启，自动逻辑备份保留地域默认为当前集群所在地域。
备份地域	开启跨地域备份时，可选择备份存放的地域，最多支持设置三个地域。
自动备份保留	备份可以保留7天 – 1830天。

6. 开启并设置成功后，在右下角单击**确定**完成逻辑备份设置。
7. 您也可以更改备份设置或者关闭备份设置。



- 更改备份设置：可以更改开始时间与调整跨地域备份设置（跨地域备份地域修改、跨地域备份能力开启或关闭）。
- 关闭备份设置：支持关闭逻辑备份和自动逻辑备份的跨地域备份功能，关闭后，备份文件将持续保留至备份周期终止。

8. 自动逻辑备份的文件名是自动生成，您可在备份列表单击备注名后的编辑图标进行设置和修改。



设置窗口如下：



手动备份

最近更新时间：2025-01-15 10:06:43

TDSQL-C MySQL 版支持自动备份和手动备份两种方式来备份数据。本文为您介绍如何通过控制台创建手动备份。

说明：

- 为保证备份系统的稳定，**当前系统每小时仅允许发起一次备份**。
- 手动备份可选择快照备份对整个集群进行备份，也可以选择逻辑备份对整个集群或者特定的库/表进行备份。
- 手动备份中的逻辑备份可在备份列表手动删除，释放掉备份空间，避免空间浪费和占用，快照备份不支持手动删除。
- 手动备份中逻辑备份（整个集群）支持开启跨地域备份功能，可将手动逻辑备份（整个集群）文件存储至其他地域。
- 如果您的集群存储空间使用大于2TB，则不建议使用逻辑备份，否则会存在备份失败的风险。

操作步骤

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在上方选择地域，在集群列表，找到需要的集群，单击集群 ID 或**操作**列的**管理**，进入集群管理页面。
3. 在集群管理页面，选择**备份管理**页，单击**手动备份**。

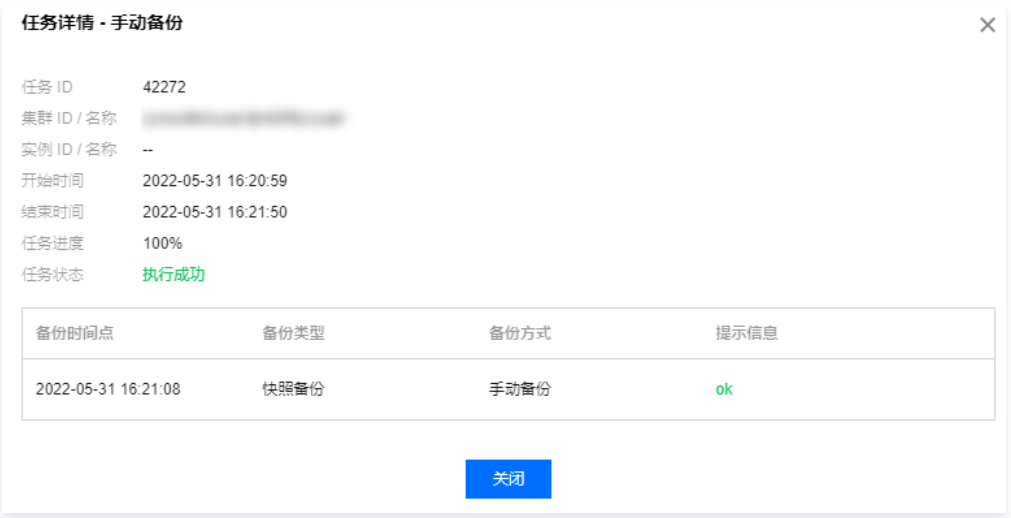


4. 在弹出的对话框，选择备份类型和对象，设置备份文件备注名，单击**确定**创建备份任务。



参数	说明
备份类型	支持快照备份和逻辑备份。 **注意：** 逻辑备份会对数据库加锁，在手动备份期间会影响数据库的使用，建议在业务低峰期进行逻辑备份操作。
选择对象	快照备份针对整个集群进行备份，逻辑备份支持备份整个集群或指定库表。
备注名	手动备份创建时支持对备份文件设置备注名（仅支持数字，英文大小写字母、中文以及特殊字符-_./()[]（）+=：;@且长度不能超过60）。
跨地域备份	**手动逻辑备份（整个集群）**可以选择是否开启跨地域备份。 • 开启后，可将手动逻辑备份（整个集群）文件保存到其他地域。 • 未开启，手动逻辑备份保留地域默认为当前集群所在地域。
备份地域	开启跨地域备份时，可选择手动逻辑备份（整个集群）文件存放的地域，最多支持选择三个不同地域。

5. 创建备份任务后可通过左侧导航栏进入**任务列表**查询新发起的备份任务进度，单击**操作**列的**备份详情**可查看具体执行情况。



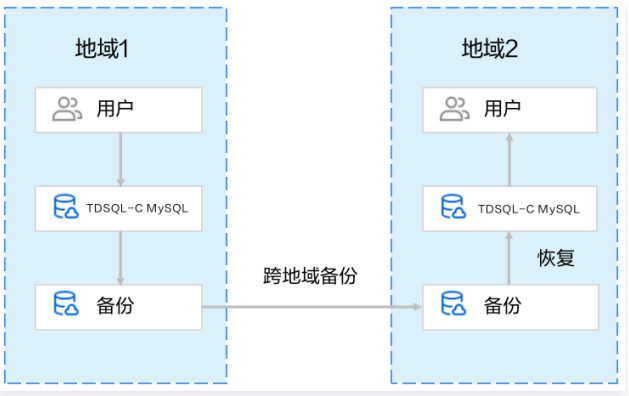
跨地域备份

最近更新时间：2025-01-14 14:36:12

云原生数据库 TDSQL-C MySQL 版提供跨地域备份功能，支持将备份文件放到另一个地域存储，帮助用户提升监管与容灾恢复能力，同时提升数据的可靠性，目前 TDSQL-C MySQL 版集群支持为自动逻辑备份、binlog 备份以及手动逻辑备份（整个集群）进行跨地域备份设置。
本文介绍跨地域备份功能。

背景信息

数据是企业运营的重要组成，信息化带来便利的同时，电子数据、存储信息极易毁损、丢失的特点也暴露了出来。而在自然灾害、系统故障、员工误操作和病毒感染面前，任意一次事故就有可能导致企业的运营中断，甚至带来灾难性损失。因此，确保核心数据的安全完整性，是每个企业必须考虑的。云原生数据库 TDSQL-C MySQL 版提供跨地域备份功能，用于将备份文件存储于其他地域，达到最大程度避免自然灾害、系统故障等导致的数据损毁，跨地域备份可实现数据库恢复的高可用性，满足数据可用性及安全性、异地备份恢复、异地容灾、长期归档数据、监管等需求。



跨地域备份限制说明

当前仅支持高 IO 版集群在自动逻辑备份、binlog 备份、手动逻辑备份（整个集群）使用跨地域备份能力。

跨地域备份说明

- 跨地域备份可恢复数据到集群所在地域或备份所在地域。
- 开启跨地域备份后，不会影响本地产生的备份，两者同时存在。
- 本地自动备份完成后就会触发跨地域备份，即将自动备份转储到跨地域备份存储设备上。

计费说明

跨地域备份当前处于公测阶段，目前免费使用。

跨地域逻辑备份和本地自动备份的区别

对比项	跨地域逻辑备份	本地自动备份
是否默认开启	默认关闭，需要手动开启	默认开启
备份存储地域	备份存储在指定异地地域（最多指定三个）	备份存储在集群所在地域
备份恢复	可以恢复数据到： • 原集群 • 目的地域的新集群	可以恢复到： • 原集群 • 当前地域的新集群

支持地域

支持跨地域备份功能的地域为 TDSQL-C MySQL 版全部地域。

开启自动逻辑备份的跨地域备份

> ⚠ 注意：

建议在开启自动逻辑备份的跨地域备份后，同步开启对应地域的 binlog 跨地域备份，方便您在对应地域回档时可以执行按时间点进行回档。

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在上方选择地域，根据实际控制台视图模式，对应操作。

页签视图

在左侧集群列表，单击目标集群，进入集群管理页。



列表视图

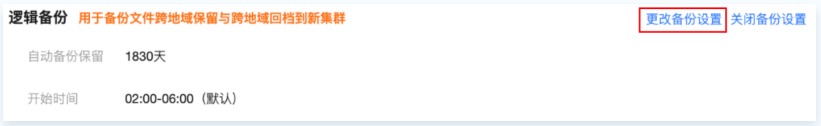
在左侧集群列表找到目标集群，单击**集群 ID** 或**操作**列的**管理**，进入集群管理页。



3. 在集群管理页面，选择**备份管理**页，单击**自动备份设置**。



4. 在逻辑备份下，单击**开启并设置**。

> **说明：**
> 若您已开启自动逻辑备份，但未开启自动逻辑备份的跨地域备份功能，请单击**更改备份设置**。
>
> 
>

5. 在弹出的对话框，完成下列配置，单击**确定**。



参数	说明
开始时间	设置自动逻辑备份每天的开始时间，默认为02:00-06:00。
自动备份周期	默认为每天。 **说明：** 当前版本仅支持自动备份周期为每天。由于逻辑备份速率限制，部分数据量较大的集群备份周期预计延长至1天以上（若当天逻辑备份未完成，则会在此逻辑备份完成日期的下一天才进行下一次自动逻辑备份）。
跨地域备份	选择开启跨地域备份。
备份地域	选择备份存放的地域，最多支持选择三个不同地域。
自动备份保留	备份可以保留7天 - 1830天。

开启 binlog 备份的跨地域备份

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在上方选择地域，根据实际控制台视图模式，对应操作。

页签视图

在左侧集群列表，单击目标集群，进入集群管理页。

列表视图

在左侧集群列表找到目标集群，单击**集群 ID** 或**操作**列的**管理**，进入集群管理页。



3. 在集群管理页面，选择**备份管理**页，点击**日志备份设置**。



4. 在弹出的对话框，完成下列配置，单击**确定**。



参数	说明
binlog 设置	默认开启。
binlog 备份保留	binlog 备份保留天数：7天 – 1830天，且不得小于数据备份保留天数：7天。
跨地域备份	选择是否开启跨地域备份。 • 开启后，可将 binlog 备份保存到其他地域。 • 未开启，binlog 备份保留地域默认为当前集群所在地域。
备份地域	开启跨地域备份时，可选择备份存放的地域，最多支持选择三个不同地域。

开启手动逻辑备份的跨地域备份

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在上方选择地域，根据实际控制台视图模式，对应操作。

页签视图

在左侧集群列表，单击目标集群，进入集群管理页。

列表视图

在左侧集群列表找到目标集群，单击**集群 ID** 或**操作**列的**管理**，进入集群管理页。



3. 在集群管理页面，选择**备份管理**页，单击**手动备份**。



4. 在弹出的对话框，完成下列配置，单击**确定**创建备份任务。



参数	说明
备份类型	选择逻辑备份。 **注意：** 逻辑备份会对数据库加锁，在手动备份期间会影响数据库的使用，建议在业务低峰期进行逻辑备份操作。
选择对象	在手动备份中，跨地域备份能力暂时仅支持选择对整个集群进行跨地域备份。
备注名	手动备份创建时支持对备份文件设置备注名（仅支持数字，英文大小写字母、中文以及特殊字符-_./()[]（）+=：;@且长度不能超过60）。
跨地域备份	选择开启跨地域备份，开启后，可将手动逻辑备份保存到其他地域。
备份地域	选择手动逻辑备份存放的地域，最多支持选择三个不同地域。

管理备份
查看备份列表

最近更新时间：2024-10-12 19:19:12

集群创建之初，TDSQL-C MySQL 版会按照默认的备份设置（根据日志生成速率等因素，7*24不间断进行快照备份，快照备份文件生成周期间隔6～48小时不等，默认保留7天）自动备份数据，您也可以通过控制台对集群进行自动备份设置或进行手动备份，备份文件及相关信息可通过备份列表进行查阅。
本文为您介绍通过控制台查看备份列表的方法。

操作步骤

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在上方选择地域，找到所需集群，单击**集群 ID** 或**操作**列的**管理**，进入集群管理页。
3. 在集群管理页，选择**备份管理**页，在数据备份列表或 binlog 备份列表中可查看每一个备份任务的相关信息。



- 数据备份列表信息：文件名/备注名、备份时间点、开始时间、结束时间、备份类型、备份方式、备份大小、备份地域、状态以及操作等。
- binlog 备份列表信息：文件名、binlog 数据开始时间、binlog 数据结束时间、备份大小以及操作等。
- 支持按全部、今天、昨天、近7天、近30天以及自定义时间范围筛选备份文件。
- 右侧搜索框支持根据文件名、备注名、备份方式、备份类型的资源属性搜索备份文件。



热点问题

超出备份保留时间的备份还可以下载或还原吗?

到期后的备份文件（包含数据备份和 binlog 备份）会自动删除，无法进行下载还原。

- 您可在 TDSQL-C MySQL 版控制台 下载备份文件，支持下载的备份文件为逻辑备份。
- 您也可以在控制台进行手动备份。

> **说明**
> 手动备份中的逻辑备份支持手动删除，快照备份不支持删除。

备份可以手动删除吗?

- 自动备份中的快照备份无法手动删除，到期后会自动删除，默认保留7天，您也可以结合业务需要，对备份保留时间进行设置，以便合理使用备份空间，详细请参见 设置备份保留时间 。
- 手动备份中的快照备份无法删除。
- 自动备份中的逻辑备份和手动备份中的逻辑备份支持在 TDSQL-C MySQL 版控制台的备份列表手动删除。

如何减少备份空间开销?

- 删除不再使用逻辑备份，操作请参见 删除备份 。

版权所有：腾讯云计算（北京）有限责任公司

- 根据业务需要，合理缩短自动备份保留时间，实现定期删除自动备份产生的文件，详细请参见 设置备份保留时间 。

设置备份保留时间

最近更新时间：2025-01-16 10:57:43

TDSQL-C MySQL 版的备份包含数据备份和 binlog 备份，备份方式分为自动备份和手动备份，备份地域支持本地域备份和跨地域备份，设置备份保留时间的操作场景为：为自动备份中快照备份设置备份保留时间、为自动备份中逻辑备份在本地域或跨地域设置备份保留时间、为 binlog 设置备份保留时间。
本文为您介绍通过控制台设置备份保留时间。

操作步骤

设置自动备份快照备份保留时间

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在上方选择地域，根据实际控制台视图模式，对应操作。

页签视图

在左侧集群列表，单击**目标集群**，进入集群管理页。



列表视图

在左侧集群列表找到目标集群，单击**集群 ID** 或**操作**列的**管理**，进入集群管理页。



3. 在集群管理页面，选择**备份管理**页，单击**自动备份设置**。



4. 在自动备份设置窗口，可以设置快照备份的数据库保留时间，单击**确定**。



设置自动备份逻辑备份保留时间

1. 按照 步骤1-3 进入自动备份设置页面。
2. 在自动备份设置窗口，逻辑备份下单击**开启并设置**。
3. 在弹出的对话框，完成下列配置，单击**确定**。



参数	说明
开始时间	设置自动逻辑备份每天的开始时间，默认为02:00-06:00。
自动备份周期	默认为每天。 **说明：** 当前版本仅支持自动备份周期为每天。由于逻辑备份速率限制，部分数据量较大的集群备份周期预计延长至1天以上（若当天逻辑备份未完成，则会在此逻辑备份完成日期的下一天才进行下一次自动逻辑备份）。
跨地域备份	选择是否开启跨地域备份。 • 开启后，可将自动逻辑备份文件保存到其他地域。详细说明请参见 跨地域备份 。 • 未开启，自动逻辑备份保留地域默认为当前集群所在地域。
备份地域	开启跨地域备份时，可选择备份存放的地域，最多支持设置三个地域。
自动备份保留	备份可以保留7天 – 1830天。自动备份逻辑备份文件（本地域）的保留时间与自动备份逻辑备份文件（跨地域）的保留时间一致。

4. 开启并设置成功后，在右下角单击**确定**完成逻辑备份设置。

设置 binlog 备份保留时间

说明：
- binlog 备份保留天数：7~1830天，且不得小于数据备份保留天数。
- 若为 binlog 开启了跨地域备份，则本地域 binlog 备份文件保留时间和跨地域 binlog 备份文件保留时间一致。

1. 按照 步骤1-2 进入集群管理页。
2. 在集群管理页，选择**备份管理**页，单击**日志备份设置**。
3. 设置 binlog 备份保留天数，单击**确定**。



设置备份下载规则

最近更新时间：2025-06-12 15:17:32

TDSQL-C MySQL 版集群的备份文件默认可以通过公网或内网进行下载，同时 TDSQL-C MySQL 版提供备份下载条件的限制，请结合实际需要设置备份下载规则，提升安全性和管理效率。

使用场景

本功能是集群维度，若您需要针对某个集群设置备份下载规则来限制此集群备份文件下载的条件，您可使用此功能。

功能说明

- 外网下载为默认开启，开启时，在外网和内网均可以下载备份。
- 外网下载关闭时，需设置内网下载条件（IP、VPC）。

设置备份下载规则

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在上方选择地域，根据实际控制台视图模式，对应操作。

页签视图
在左侧集群列表，单击**目标集群**，进入集群管理页。

列表视图
在左侧集群列表找到目标集群，单击**集群 ID** 或**操作**列的**管理**，进入集群管理页。

3. 在集群管理页面，选择**备份管理** > **下载设置**，单击**编辑**。



4. 在弹出页面，根据需要进行配置，单击**确定**。



参数	说明
外网下载	根据需要开启或关闭外网下载。 • 开启：外网下载开启时，无法对下载条件进行设置。 • 关闭：外网下载关闭时，可以对内网下载条件进行设置，设置条件包含 IP、VPC。
下载条件	设置下载条件。 • 当条件对应的条件值为空时，代表不使用该条件进行限制。 • IP 条件支持 IP 地址作为条件值，条件操作符可以选择**包含**或者**不包含**，多个 IP 地址之间使用中文逗号分隔。 • VPC 条件支持当前地域下的私有网络作为条件值，条件操作符为**包含**，可勾选多个私有网络。

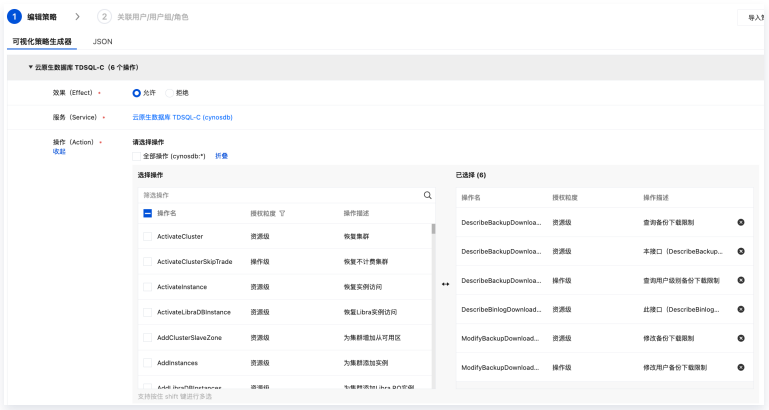
5. 配置完成后，即可查看生效的条件。

版权所有：腾讯云计算（北京）有限责任公司

授权子账号使用设置备份下载规则

默认情况下，子账号没有设置数据库备份下载规则的权利。因此用户就需要创建策略来允许子账号设置备份下载规则。

1. 登录 访问管理控制台 。
2. 在左侧导航栏，单击**策略**。
3. 选择**新建自定义策略 > 按策略生成器创建**，配置对应的策略参数，单击**下一步**。



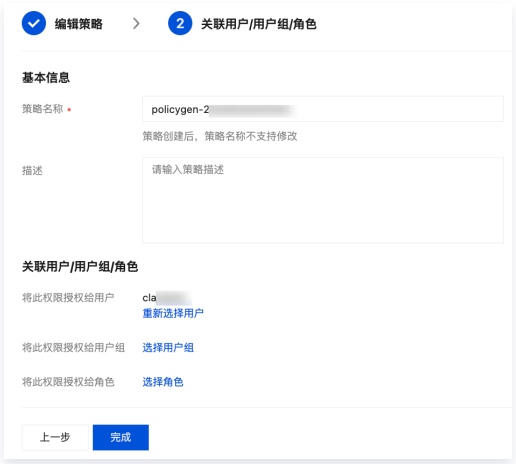
资源级接口

- 效果：选择允许。
- 服务：云原生数据库 TDSQL-C（cynosdb）。
- 操作：选择 DescribeBackupDownloadRestriction、DescribeBackupDownloadUrl、DescribeBinlogDownloadUrl、ModifyBackupDownloadRestriction。
- 资源：可以选择指定实例，也可选择全部资源。

操作级接口

- 效果：选择允许。
- 服务：云原生数据库 TDSQL-C（cynosdb）。
- 操作：选择 DescribeBackupDownloadUserRestriction、ModifyBackupDownloadUserRestriction。
- 资源：无法指定实例，选择全部资源即可。

4. 输入策略基本信息，关联用户/用户组/角色后，单击**完成**。



下载和解压备份

最近更新时间：2025-01-15 10:06:43

TDSQL-C MySQL 版控制台提供备份的文件列表，并可以下载逻辑备份文件，通过下载的备份文件可将数据库恢复到自建或其他云厂商提供的数据库。

支持下载/解压/删除的文件类型

分类	备份类型		方式	下载	下载后是否需解压	删除
数据备份	逻辑备份	全量备份	手动	✓	✓	✓
			自动	✓	✓	✓
binlog 备份	binlog 备份	增量备份	自动	✓	✓	x

下载数据备份文件

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在上方选择地域，根据实际控制台视图模式，对应操作。

页签视图

在左侧集群列表，单击目标集群，进入集群管理页。



列表视图

在左侧集群列表找到目标集群，单击**集群 ID** 或**操作**列的**管理**，进入集群管理页。



3. 在集群管理页，选择**备份管理** > **数据备份列表**查看对应备份文件。



4. 在数据备份列表的**操作**列，单击**下载**。
5. 在弹出的对话框，**点击复制**文件的下载地址，运用 wget 命令进行高速下载或单击**本地下载**。

> **说明：**
> - 复制下载地址后 登录到 CVM（Linux 系统）中运用 wget 命令下载。
> - 如果集群和 CVM 属于同地域，那么无论是跨 VPC 还是同 VPC，都支持运用 wget 命令进行内网高速下载。
> - 如果集群与 CVM 属于跨地域，将不支持内网高速下载，CVM 必须开启公网 IP 才支持运用 wget 命令进行下载。
> - 下载地址的有效期为12小时，超过12小时后，需要重新进入下载页面获取新生成的下载地址。
> - 使用 wget 下载时需要对 URL 添加英文引号。

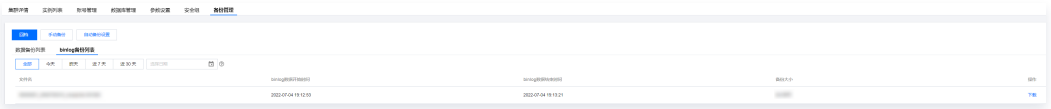
版权所有：腾讯云计算（北京）有限责任公司

- wget 命令格式： wget -c "<备份文件下载地址>" -O <自定义名称>.xb



下载 binlog 备份文件

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在上方选择地域，找到所需集群，单击**集群 ID** 或**操作**列的**管理**，进入集群管理页。
3. 在集群管理页，选择**备份管理** > **binlog 备份列表**查看对应备份文件。



4. 在数据备份列表的**操作**列，单击**下载**。
5. 在弹出的对话框，**点击复制**文件的下载地址，可运用 wget 命令进行高速下载或单击**本地下载**。

说明：

- 复制下载地址后 登录到 CVM（Linux 系统）中运用 wget 命令下载。
- 如果集群和 CVM 属于同地域，那么无论是跨 VPC 还是同 VPC，都支持运用 wget 命令进行内网高速下载。
- 如果集群与 CVM 属于跨地域，将不支持内网高速下载，CVM 必须开启公网 IP 才支持运用 wget 命令进行下载。
- 下载地址的有效期为12小时，超过12小时后，需要重新进入下载页面获取新生成的下载地址。
- 使用 wget 下载时需要对 URL 添加英文引号。
- wget 命令格式： wget -c "<备份文件下载地址>" -O <自定义名称>.xb



解压文件

为节约存储空间，TDSQL-C MySQL 版的数据/binlog 备份文件，都会先经过 qpress 压缩，后经过 xbstream 打包（xbstream 为 Percona 的一种打包/解包工具），即下载的备份文件，不能直接导入到目标数据库，需进行解包/解压文件后才能将文件导入目标数据库。

解包备份文件

使用 xbstream 解包备份文件。

说明：

xbstream 工具下载地址请参见 Percona XtraBackup 官网，请选择 Percona XtraBackup 2.4.6 及以上的版本，安装介绍请参见 Percona XtraBackup 2.4 。

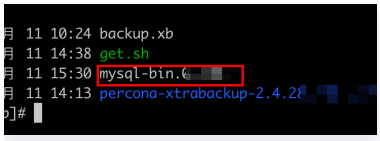
```
xbstream -x -C /data/mysql < test.xb
```

说明：

- /data/mysql 为文件解压的目录，您可根据实际情况替换为实际路径。
- test.xb 替换为您的备份文件。

解包结果如下图所示：





解压备份文件

1. 通过如下命令下载 qpress 工具。

```
wget -d --user-agent="Mozilla/5.0 (Windows NT x.y; rv:10.0) Gecko/20100101 Firefox/10.0" https://docs-tencentdb-1256569818.cos.ap-guangzhou.myqcloud.com/qpress-11-linux-x64.tar
```

说明：
若 wget 下载提示错误，您可单击 下载 qpress 工具 下载到本地后，再将 qpress 工具上传至 Linux 云服务器，请参见 通过 SCP 上传文件到 Linux 云服务器 。

2. 通过如下命令解出 qpress 二进制文件。

```
tar -xf qpress-11-linux-x64.tar -C /usr/local/bin
source /etc/profile
```

3. 使用 qpress 解压备份文件。

```
qpress -d /<备份文件所在目录>/<备份文件> <解压目录>/
示例：
qpress -d /backup/data_backup_286***_2024***.sql.qp sql/
```

说明：
备份解压后为 logbin 文件形式。

- <备份文件所在目录>：替换为对应解包备份文件所在目录，命令中的尖括号<>需要删除。
- <备份文件>：替换为对应解包备份文件名，命令中的尖括号<>需要删除，只保留实际的参数。
- <解压目录>：替换为您希望将文件解压到的目标路径，紧跟在解包备份文件名后面，命令中的尖括号<>需要删除。

删除备份

最近更新时间：2024-10-12 19:19:13

TDSQL-C MySQL 版支持删除备份，以节省备份空间。手动备份和自动备份中产生的逻辑备份文件支持手动删除，自动备份中的快照备份文件不支持手动删除，但可以通过 设置备份保留时间 来调整该备份的保留生命周期。
本文为您介绍通过控制台删除手动备份文件。

操作步骤

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在上方选择地域，找到需要删除备份文件的集群，单击集群 ID 或**操作**列的**管理**，进入集群管理页。
3. 在集群管理页面，选择**备份管理**页，选择数据备份列表或日志备份列表，在备份列表找到需删除的逻辑备份文件，单击**操作**列的**删除**。



4. 在弹出的对话框，单击**确定**。



> **注意**
> 该备份文件选择删除后，无法恢复，请谨慎操作。

回档
使用逻辑备份恢复数据
通过 Linux 恢复数据

最近更新时间：2025-02-19 11:16:33

本文为您介绍通过 Linux 使用逻辑备份文件进行数据恢复。

操作场景

说明：
- 为节约存储空间，TDSQL-C MySQL 版的备份文件，都会先经过 qpress 压缩，后经过 xbstream 打包（xbstream 为 Percona 的一种打包/解包工具）。
- 如果您的集群存储空间使用大于2TB，则不建议使用逻辑备份，否则会存在备份失败的风险。

TDSQL-C MySQL 版支持逻辑备份方式，用户可通过控制台手动备份来生成逻辑备份文件，并支持下载获取整集群/部分库表的逻辑备份文件，本文为您介绍通过 Linux 平台，使用逻辑备份文件进行数据恢复。

操作步骤

步骤1：下载备份文件

1. 登录 TDSQL-C MySQL 版控制台，根据实际控制台视图模式，对应操作。

页签视图

在左侧集群列表，单击**目标集群**，进入集群管理页。

列表视图

在左侧集群列表找到目标集群，单击**集群 ID** 或**操作**列的**管理**，进入集群管理页。



2. 在集群管理页面，选择**备份管理** > **数据备份列表**，找到需要下载的备份，在**操作**列单击**下载**。



3. 在弹出的对话框，复制下载地址，并 登录到 CVM（Linux 系统）中，运用 wget 命令进行高速下载。

说明：
- 复制下载地址后 登录到 CVM（Linux 系统）中运用 wget 命令下载。
- 如果集群和 CVM 属于同地域，那么无论是跨 VPC 还是同 VPC，都支持运用 wget 命令进行内网高速下载。
- 如果集群与 CVM 属于跨地域，将不支持内网高速下载，CVM 必须开启公网 IP 才支持运用 wget 命令进行下载。

- 您也可以选择**本地下载**直接下载，但耗时较多。

wget 命令格式如下：

```
wget -c "<备份文件下载地址>" -O <自定义名称>.xb
```



步骤2：解包备份文件

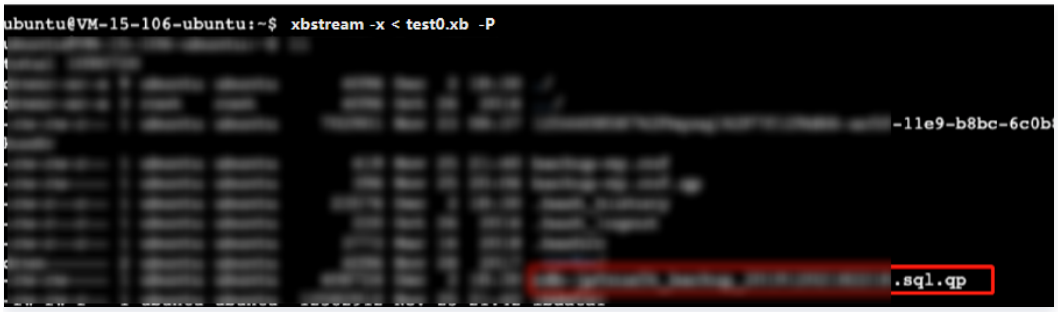
使用 xbstream 解包备份文件。

说明：
xbstream 工具下载地址请参见 Percona XtraBackup 官网，请选择 Percona XtraBackup 2.4.6 及以上的版本，安装介绍请参见 Percona XtraBackup 2.4 。

```
xbstream -x < test0.xb  -P
```

说明：
`test0.xb` 替换为您的备份文件名。

解包结果如下图所示：



步骤3：解压备份文件

1. 通过如下命令下载 qpress 工具。

```
wget -d --user-agent="Mozilla/5.0 (Windows NT x.y; rv:10.0) Gecko/20100101 Firefox/10.0" https://docs-tencentdb-1256569818.cos.ap-guangzhou.myqcloud.com/qpress-11-linux-x64.tar
```

说明：
若 wget 下载提示错误，您可单击 下载 qpress 工具 下载到本地后，再将 qpress 工具上传至 Linux 云服务器，请参见 通过 SCP 上传文件到 Linux 云服务器 。

2. 通过如下命令解出 qpress 二进制文件。

```
tar -xf qpress-11-linux-x64.tar -C /usr/local/bin
source /etc/profile
```

3. 使用 qpress 解压备份文件。

```
qpress -d /<备份文件所在目录>/<备份文件> <解压目录>/
示例：
qpress -d /backup/data_backup_286***_2024***.sql.qp sql/
```

- <备份文件所在目录>：替换为对应解包备份文件所在目录，命令中的尖括号<>需要删除。
- <备份文件>：替换为对应解包备份文件名，命令中的尖括号<>需要删除，只保留实际的参数。
- <解压目录>：替换为您希望将文件解压到的目标路径，紧跟在解包备份文件名后面，命令中的尖括号<>需要删除。

步骤4：导入备份至目标数据库

执行如下命令导入 sql 文件至目标数据库：

```
mysql -u<账户名> -P<端口> -h<目标数据库内网地址> -p < <通过 qpress 实际解压出的 sql 文件>
```

通过控制台恢复数据
库表级回档至原集群

最近更新时间：2025-02-19 11:16:33

TDSQL-C MySQL 版支持使用逻辑备份来恢复数据，其控制台提供了便捷的操作入口，您可通过控制台轻松实现数据的快速恢复。使用逻辑备份时，支持库表级回档至原集群和 整集群回档至新集群，本文为您介绍通过控制台使用逻辑备份指定库表回档至原集群。

功能限制

库级别的回档，不会导入的对象类型包含：外键约束、check 约束、触发器、存储过程、用户视图、Event。

回档模式

- 按备份文件回档：将集群恢复到某一个逻辑备份文件的数据集状态，备份文件选择范围由用户设置的数据备份保留时长决定。
- 按时间点回档：将集群恢复到任意时间点，时间点的选择范围由用户设置的日志保留时长决定。

前提条件

- 原集群下已存在可用的逻辑备份，要生成逻辑备份，可参见 执行备份 各文档中的生成逻辑备份的步骤。
- 集群状态为**运行中**。

操作步骤

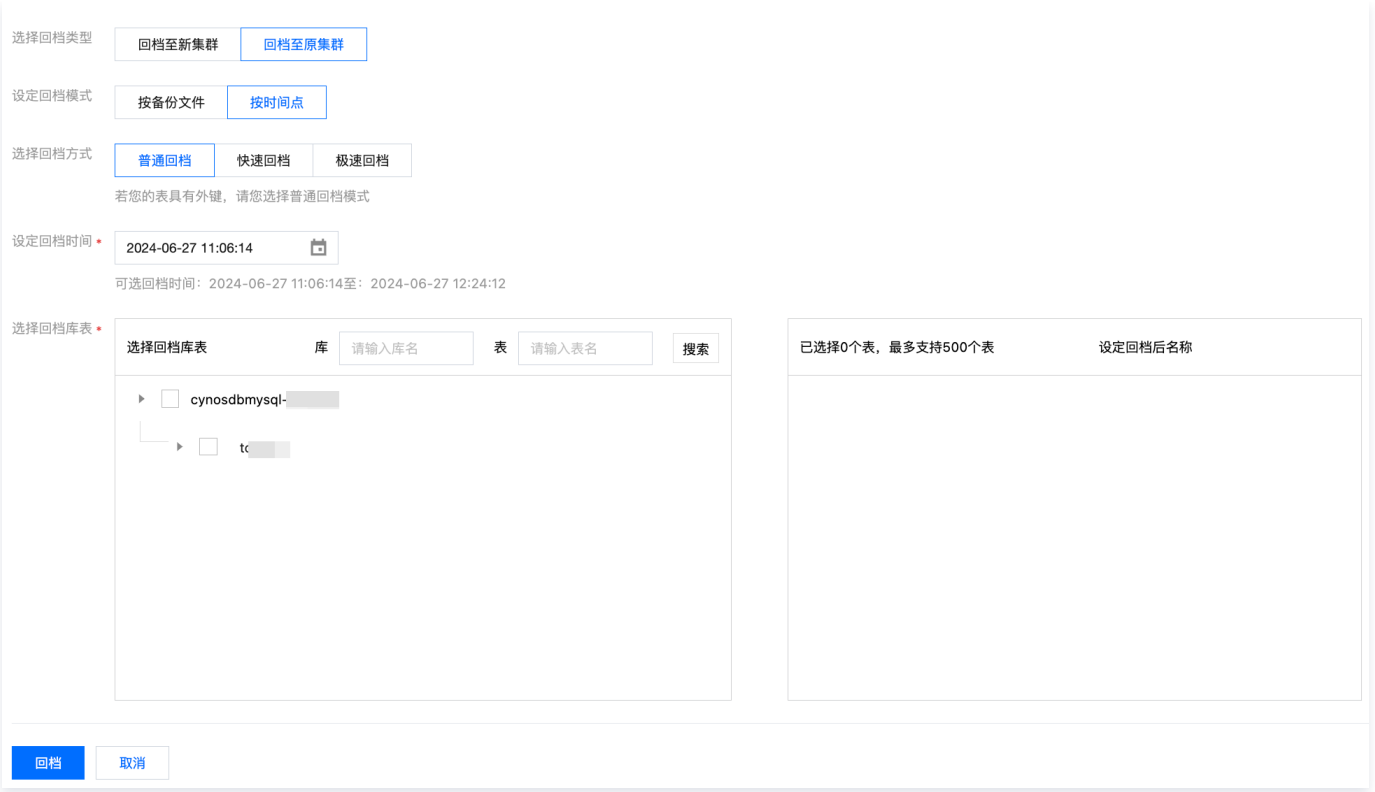
1. 登录 TDSQL-C MySQL 版控制台，在集群列表，单击**集群 ID**，进入集群管理页面。
2. 在集群管理页面，选择**备份管理**页，单击**回档**。



3. 在回档页面，完成如下配置，单击**回档**。

> **说明：**
> - 库表回档需明确指定所需回档的库表，若无法确定所有涉及的库表，建议 整集群回档至新集群，确认后再将数据通过 DTS 迁回原集群。
> - 若需回档的库表在指定回档的时间点不存在，库表回档会失败。
> - 若需回档库表已不存在或被误删，需要先登录数据库并创建库表，再使用控制台进行回档操作。
> - 若指定回档库表存在主外键约束，请确保回档过程中关联的库表存在，否则库表回档会失败。
> - 单次操作最多支持回档500个库或表。

版权所有：腾讯云计算（北京）有限责任公司



参数	说明
选择回档类型	选择回档至原集群。
设定回档模式	支持选择按备份文件或按时间点回档。
选择备份文件	若回档模式设定为按备份文件，则需选择已有的用于回档的逻辑备份文件，然后选择需要回档的库表即可。
选择回档方式	若回档模式设定为按时间点，则需选择回档方式。 • 普通回档：若您的表具有外键，请您选择普通回档模式。 • 快速回档：导入该实例的全量备份，再对选中的库、表进行回档。该回档模式无限制，但回档速度较慢。 • 极速回档：极速回档模式下不支持具有外键的表，若您的表存在外键，请您选择普通回档模式。
设定回档时间	若回档模式设定为按时间点，则需选择需要回档的时间点。
选择回档库表	选择回档的库表，支持按库名和表名进行搜索，支持对选中的库重命名，如未重命名，系统将默认命名新库为“原库名_bak”。

4. 在弹出的对话框，确认无误后，单击**确定**发起回档任务。
5. 任务提交后，可单击**查看回档任务**或在任务列表中查看回档进度和任务详情。



6. 回档完成后，可在原集群中看到回档后的新库表，并进行进一步操作。

整集群回档至新集群

最近更新时间：2025-02-19 11:16:33

TDSQL-C MySQL 版支持使用逻辑备份来恢复数据，其控制台提供了便捷的操作入口，您可通过控制台轻松实现数据的快速恢复。使用逻辑备份时，支持 库表级回档至原集群 和整集群回档至新集群，本文为您介绍通过控制台使用逻辑备份将整集群回档至新集群。

说明：

- 原集群的逻辑备份也支持跨地域保存，若您希望将整集群在其他地域恢复成新集群，您可以通过跨地域备份来回档，这需要您先开启并生成跨地域备份，操作请参见 跨地域备份 。
- 新回档的集群在回档成功后才开始进行计费，计费介绍与费用详情请参见 计费概述 。
- 整集群回档至新集群时，系统会根据您的选择创建一个新集群，新集群经过验证后，您可再将数据通过 DTS 迁回原集群，这使得您可以继续使用原集群的 VIP 地址；或者您可以直接使用新集群，但您需要在应用程序更换新集群的 VIP 地址，并根据实际情况选择是否删除原集群。

功能限制

通过控制台使用逻辑备份将整集群回档至新集群时，不支持交换原集群和新集群的 VIP 地址。

回档模式

- 按备份文件回档：将集群恢复到某一个逻辑备份文件的数据集状态，备份文件选择范围由用户设置的数据备份保留时长决定。
- 按时间点回档：将集群恢复到任意时间点，时间点的选择范围由用户设置的日志保留时长决定。

前提条件

- 原集群下已存在可用的逻辑备份，要生成逻辑备份，可参见 执行备份 各文档中的生成逻辑备份的步骤。
- 集群状态为**运行中**。

操作步骤

1. 登录 TDSQL-C MySQL 版控制台，在集群列表，单击**集群 ID**，进入集群管理页面。
2. 在集群管理页面，选择**备份管理**页，单击**回档**。



3. 在回档页面，完成如下配置，单击**前往购买页回档集群**。



参数	说明
选择回档类型	选择回档至新集群。

选择恢复范围	整集群回档。
地域	选择新集群的地域，支持选择在原集群所在地域新建集群，也支持选择跨地域新建集群，跨地域的选择范围由您的集群下存在的跨地域备份文件决定，了解跨地域备份，请参见 跨地域备份 。
可用区	选择新集群对应地域的可用区。
设定回档模式	支持选择按备份文件或按时间点回档。
选择备份文件	若回档模式设定为按备份文件，则需选择已有的用于回档的逻辑备份文件。
设定回档时间	若回档模式设定为按时间点，则需选择回档的时间点。

4. 在弹出的购买页，根据所需数据库形态，完成对应**数据库配置**，单击**下一步**。

> **说明：**
> 若您的新集群要创建为实例形态是 Serverless 的集群，则需确保原集群和新集群的地域及可用区均支持实例形态为 Serverless，否则将只能创建实例形态为预置资源的集群，具体请以购买页是否支持为准。

- **实例形态选择预置资源**
  - 实例形态：预置资源。
  - 回档模式：支持按备份文件、按时间点回档。
    - 按备份文件：可从指定的备份集中恢复出新集群，选择范围基于备份完成的时间和保留时长。
    - 按时间点：可从具体某个时间点恢复出新集群，时间点选择范围基于备份保留时长。
  - 选择恢复范围：整集群回档。
  - 地域：未开启跨地域备份时的地域默认和被回档集群所在地域一致，开启跨地域备份并且已产生跨地域备份文件时，支持选择备份文件所在地的地域进行回档。开启跨地域备份请参见 跨地域备份 。
  - 主可用区：未开启跨地域备份时可用区和被回档集群所在可用区一致，开启跨地域备份并且已产生跨地域备份文件时，支持选择备份文件所在地所支持的可用区进行回档。开启跨地域备份请参见 跨地域备份 。
  - 传输链路：高 IO 版。
  - 网络：选择私有网络。
  - 数据库版本：和被回档集群一致。
  - 实例类型：支持通用型、独享型。
  - 计算实例（读写）：选择读写实例的计算规格，仅支持配置一个读写实例。
  - 计算实例（只读）：选择只读实例的计算规格和数量。
  - 计算计费模式：支持包年包月、按量计费。
  - 存储计费模式：默认按量计费。
- **实例形态选择 Serverless**
  - 实例形态：Serverless。
  - 回档模式：支持按备份文件、按时间点回档。
    - 按备份文件：可从指定的备份集中恢复出新集群，选择范围基于备份完成的时间和保留时长。
    - 按时间点：可从具体某个时间点恢复出新集群，时间点选择范围基于备份保留时长。
  - 选择恢复范围：整集群回档。
  - 地域：未开启跨地域备份时的地域默认和被回档集群所在地域一致，开启跨地域备份并且已产生跨地域备份文件时，支持选择备份文件所在地的地域进行回档。开启跨地域备份请参见 跨地域备份 。
  - 可用区：未开启跨地域备份时可用区和被回档集群所在可用区一致，开启跨地域备份并且已产生跨地域备份文件时，支持选择备份文件所在地所支持的可用区进行回档。开启跨地域备份请参见 跨地域备份 。
  - 传输链路：高 IO 版。
  - 网络：选择私有网络。
  - 数据库版本：和被回档集群一致。
  - Serverless 架构：支持选择单节点版、集群版。
  - 算力配置：选择算力配置 CCU（TDSQL-C Compute Unit）上下限，实例会根据选择资源范围自动进行弹性扩缩容。CCU（TDSQL-C Compute Unit）为 Serverless 的计算计费单位，一个 CCU 近似等于1个 CPU 和 2GB 内存的计算资源，每个计费周期的 CCU 使用数量

为：数据库所使用的 CPU 核数 与 内存大小的1/2 二者中取最大值。算力配置可参考 服务算力配置 。

- 自动暂停：配置实例自动暂停时间，在设定时间内无连接访问数据库会自动暂停实例，实例暂停后计算将不再计费。
- 计算计费模式：支持选择按量计费或资源包。

> **说明：**
> 计算资源包，用来优先抵扣按量计费产品中实际产生的用量，资源包额度消耗完毕后，会继续按照按量计费模式计费。计算资源包会按照每秒实际使用的 CCU 量进行抵扣，相比于按量付费模式更为划算，使用更为灵活。

- 计算资源包（计算计费模式选择资源包时此项显示）：绑定资源包，可绑定当前账户下，在有效期内的，集群所选地域的所有计算资源包。若无可用资源包，可先 购买资源包 。
- 存储计费模式：支持选择按量计费或资源包。

> **说明：**
> 存储资源包，用来优先抵扣按量计费产品中实际产生的用量，资源包额度消耗完毕后，会继续按照按量计费模式计费。存储资源包会按照每小时实际使用的存储量进行抵扣，相比于按量付费模式更为划算，使用更为灵活。

- 存储资源包（存储计费模式选择资源包时此项显示）：绑定资源包，可绑定当前账户下，在有效期内的，集群所选地域的所有存储资源包。若无可用资源包，可先 购买资源包 。

5. 在跳转页面配置基本信息和高级配置，包括集群名称，默认字符集，安全组，参数模板及项目等，确认无误后单击**立即购买**。
6. 返回集群列表，可看到通过回档新建的集群，待状态为**运行中**即可正常使用。

使用快照备份恢复数据
库表级回档至原集群

最近更新时间：2025-02-19 11:16:33

TDSQL-C MySQL 版支持使用快照备份来恢复数据，其控制台提供了便捷的操作入口，您可通过控制台轻松实现数据的快速恢复。使用快照备份时，支持库表级回档至原集群、整集群回档至新集群，本文为您介绍通过控制台使用快照备份指定库表回档至原集群。

功能限制

库级别的回档，不会导入的对象类型包含：外键约束、check 约束、触发器、存储过程、用户视图、Event。

回档方式

- 按备份文件回档：将集群恢复到某一个快照备份文件的数据集状态，备份文件选择范围由用户设置的数据备份保留时长决定。
- 按时间点回档：将集群恢复到任意时间点，时间点的选择范围由用户设置的日志保留时长决定。

前提条件

- 原集群下已存在可用的快照备份，要生成快照备份，可参见 执行备份 各文档中的生成快照备份的步骤。
- 集群状态为**运行中**。

操作步骤

1. 登录 TDSQL-C MySQL 版控制台，在集群列表，单击**集群 ID**，进入集群管理页面。
2. 在集群管理页面，选择**备份管理**页，单击**回档**。



3. 在回档页面，完成如下配置，单击**回档**。

> **说明：**
> - 库表回档需明确指定所需回档的库表，若无法确定所有涉及的库表，建议 整集群级别回档至新集群，确认后再将数据通过 DTS 迁回源集群。
> - 若需回档的库表在指定回档的时间点不存在，库表回档会失败。
> - 若需回档库表已不存在或被误删，需要先登录数据库并创建库表，再使用控制台进行回档操作。
> - 若指定回档库表存在主外键约束，请确保回档过程中关联的库表存在，否则库表回档会失败。
> - 单次操作最多支持回档500个库或表。



参数	说明
选择回档类型	选择回档至原集群。
设定回档模式	支持选择按备份文件或按时间点回档。
选择备份文件	若回档模式设定为按备份文件，则需选择已有的用于回档的快照备份文件，然后选择回档的库表即可。
选择回档方式	若回档模式设定为按时间点，则需选择回档方式。 • 普通回档：若您的表具有外键，请您选择普通回档模式。 • 快速回档：导入该实例的全量备份，再对选中的库、表进行回档。该回档模式无限制，但回档速度较慢。 • 极速回档：极速回档模式下不支持具有外键的表，若您的表存在外键，请您选择普通回档模式。
设定回档时间	若回档模式设定为按时间点，则需选择需要回档的时间点，时间点的选择范围由您设置的日志保留时长决定。
选择回档库表	选择回档的库表，支持按库名和表名进行搜索，支持对选中的库重命名，如未重命名，系统将默认命名新库为“原库名_bak”。

4. 在弹出的对话框，确认无误后，单击**确定**发起回档任务。
5. 任务提交后，可单击**查看回档任务**或在任务列表中查看回档进度和任务详情。



6. 回档完成后，可在原集群中看到回档后的新库表，并进行进一步操作。

集群回档优化方案

最近更新时间：2025-02-19 11:16:33

通常情况下，您可以使用快照备份通过 库表级回档至原集群 或 整集群回档至新集群 方式来恢复数据。但如果您的集群存储使用量较大（超过1TB），且需要回档的库表较大（需回档的库表占比超过存储使用量的70%），则建议您使用本文推荐的功能进行数据的快速恢复。

> **说明：**
> - 使用本文方法进行回档时，此功能会将原集群与新建集群进行建连，通过 binlog 来同步非回档的库表，以保证除了指定回档的库表处于回档时间，其余的库表均可处于最新数据状态。
> - 使用本文方法进行回档时，当原集群与新建集群数据追平时，若一致性校验通过，则允许进行 VIP 切换操作。若切换了 VIP，则会终止数据同步，同时原集群与新建集群进行内网 VIP 交换；若取消切换 VIP，则会终止数据同步，原集群与新建集群的 VIP 不交换。

操作场景

集群回档优化方案适用于大存储集群针对某单一库表的回档，如果集群存储使用量大于1TB，且需要回档的库表较大（需回档的库表占比超过存储使用量的70%），则可以使用此功能快速恢复数据。

使用限制

- 此功能仅支持开启 binlog 的集群使用。
- 此功能仅支持回档至和原集群同地域。
- 使用此功能恢复数据时，新集群的实例形态仅支持选择为预置资源。
- 仅库表级回档至新集群可以和原集群交换 VIP 地址，整集群回档至新集群不支持。

集群 VIP 地址切换能力说明

库表级回档至新集群时，系统会根据您的选择创建一个新集群，新集群经过验证后，您可在系统提示的时间内手动切换原集群与新集群的读写实例、只读实例（若有）的 VIP 地址，无需在应用程序切换新 VIP 地址，以便更快的恢复业务对数据库的访问及操作。若超过系统提示的时间未进行 VIP 地址切换，则视为放弃切换 VIP 地址，回档完成后您可以在应用程序切换新 VIP 地址来连接新集群使用，或者将数据通过 DTS 迁回原集群，继续使用旧集群。

回档方式

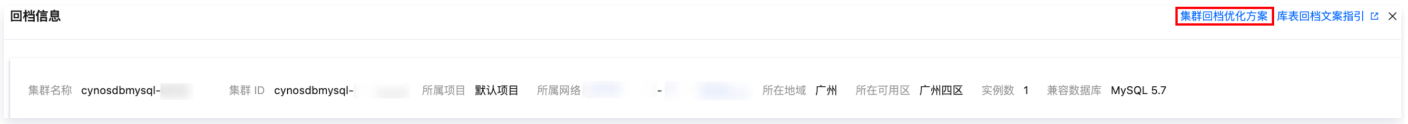
- 按备份文件回档：将集群恢复到某一个快照备份文件的数据集状态，备份文件选择范围由用户设置的数据备份保留时长决定。
- 按时间点回档：将集群恢复到任意时间点，时间点的选择范围由用户设置的日志保留时长决定。

前提条件

- 集群的实例形态为预置资源。
- 原集群下已存在可用的快照备份，要生成快照备份，可参见 执行备份 各文档中的生成快照备份的步骤。
- 集群状态为**运行中**。

操作步骤

1. 登录 TDSQL-C MySQL 版控制台，在集群列表，单击**集群 ID**，进入集群管理页面。
2. 在集群管理页面，选择**备份管理**页，单击**回档**。
3. 在回档信息右上角，单击**集群回档优化方案**。



4. 在回档弹窗下，完成如下配置，单击**前往购买页回档集群**。



参数	说明
地域	默认和原集群一致。
可用区	默认和原集群一致。
设定回档模式	支持按备份文件、按时间点回档。
选择备份文件	若回档模式设定为按备份文件，则需选择已有的用于回档的快照备份文件。
设定回档时间	若回档模式设定为按时间点，则需选择回档的时间点。
选择回档库表	选择要回档的具体库或表，回档后默认命名为原库表名称。

5. 跳转至购买页，完成如下配置后，单击**下一步**。

参数	说明
实例形态	这里选择预置资源。
回档模式	支持按备份文件、按时间点回档。
选择备份文件	若回档模式设定为按备份文件，则需选择已有的用于回档的快照备份文件。
设定回档时间	若回档模式设定为按时间点，则需选择回档的时间点。
选择恢复范围	这里选择库表级回档。
选择回档库表	选择要回档的具体库或表，回档后默认命名为原库表名称。
地域	默认和原集群一致，不支持调整。
主可用区	默认和原集群一致，不支持调整。
传输链路	默认为高 IO 版。
网络	选择私有网络。
数据库版本	默认和原集群的数据库版本一致。
实例类型	支持通用型、独享型。

计算实例（读写）	选择读写实例的计算规格，仅支持配置一个读写实例。
计算实例（只读）	若原集群无只读实例，此项不需要设置。 若原集群有只读实例，此项需要选择保留的只读实例。只读实例的数量和规格默认与原集群中只读实例一致，支持删除只读实例，支持恢复此步骤中被删除的只读实例，不支持在此时增加只读实例，如需增加只读实例，您可在回档完成后再新建只读实例。 **说明：** 当只读实例数量小于原集群只读实例数量时，只读 VIP 地址将无法完全对应迁移。
计算计费模式	支持包年包月、按量计费。
存储计费模式	默认按量计费。

6. 在跳转页面配置基本信息和高级配置，包括集群名称，默认字符集，安全组，参数模板及项目等，确认无误后单击**立即购买**。
7. 返回集群列表，找到通过回档新建的集群，单击**集群 ID** 进去新集群的管理页。
8. 在**集群详情** > **读写实例** > **读写地址**后，单击**切换详情**，您需要在系统提示的时间内完成切换，否则将视为放弃切换。



9. 进入连接地址切换弹窗，核对连接地址后，根据需要进行如下操作。



   - 确认切换：单击**确认切换**，会与原集群连接中断，过程中会有1 – 5秒闪断，原集群和新集群将交换 VIP 地址。
   - 放弃切换：单击**放弃切换**，会与原集群连接中断，原集群和新集群不交换 VIP 地址，操作后当前集群的 VIP 地址将允许写操作。
   - 取消：单击**取消**，将关闭弹窗，您可在系统提示的时间内再次选择切换。
10. 上述操作执行完成后，即库表级别回档至新集群完成。

附录：回档实时进度查询

当您执行了库表级别回档至新集群操作，您可以通过任务列表查询任务“库表回档至新集群”、“创建库表回档用新集群”的实时进度。

1. 登录 TDSQL-C MySQL 版控制台。
2. 在左侧导航栏单击**任务列表**进入任务列表界面。
3. 在上方选择对应地域。
4. 找到需要查询任务详情的任务项（“库表回档至新集群”、“创建库表回档用新集群”），在**操作**列单击**任务详情**后进行进度查看。

整集群回档至新集群

最近更新时间：2025-02-19 11:16:33

TDSQL-C MySQL 版支持使用快照备份来恢复数据，其控制台提供了便捷的操作入口，您可通过控制台轻松实现数据的快速恢复。使用快照备份时，支持 库表级回档至原集群 、整集群回档至新集群，本文为您介绍通过控制台使用快照备份将整集群回档至新集群。

说明：

- 新回档的集群在回档成功后才开始进行计费，计费介绍与费用详情请参见 计费概述 。
- 整集群回档至新集群时，系统会根据您的选择创建一个新集群，新集群经过验证后，您可再将数据通过 DTS 迁回原集群，这使得您可以继续使用原集群的 VIP 地址；或者您可以直接使用新集群，但您需要在应用程序更换新集群的 VIP 地址，并根据实际情况选择是否删除原集群。

功能限制

通过控制台使用快照备份将整集群回档至新集群时，不支持交换原集群和新集群的 VIP 地址。

回档方式

- 按备份文件回档：将集群恢复到某一个快照备份文件的数据集状态，备份文件选择范围由用户设置的数据备份保留时长决定。
- 按时间点回档：将集群恢复到任意时间点，时间点的选择范围由用户设置的日志保留时长决定。

前提条件

- 原集群下已存在可用的快照备份，要生成快照备份，可参见 执行备份 各文档中的生成快照备份的步骤。
- 集群状态为**运行中**。

操作步骤

1. 登录 TDSQL-C MySQL 版控制台，在集群列表，单击**集群 ID**，进入集群管理页面。
2. 在集群管理页面，选择**备份管理**页，单击**回档**。



3. 在回档页面，完成如下配置，单击**前往购买页回档集群**。



参数	说明
选择回档类型	选择回档至新集群。
选择恢复范围	整集群回档。

地域	选择新集群的地域，若您希望通过快照备份进行回档，这里需要选择和原集群一致的地域。
可用区	选择新集群对应地域的可用区。
设定回档模式	支持选择按备份文件或按时间点回档。
选择备份文件	若回档模式设定为按备份文件，则需选择已有的用于回档的快照备份文件。
设定回档时间	若回档模式设定为按时间点，则需选择回档的时间点。

4. 在弹出的购买页，根据所需数据库形态，完成对应**数据库配置**，单击**下一步**。

> **说明：**
> 若您的新集群要创建为实例形态是 Serverless 的集群，则需确保原集群和新集群的地域及可用区均支持实例形态为 Serverless，否则将只能创建实例形态为预置资源的集群，具体请以购买页是否支持为准。

- **实例形态选择预置资源**
  - 实例形态：预置资源。
  - 回档模式：支持按备份文件、按时间点回档。
    - 按备份文件：可从指定的备份集中恢复出新集群，选择范围基于备份完成的时间和保留时长。
    - 按时间点：可从具体某个时间点恢复出新集群，时间点选择范围基于备份保留时长。
  - 选择恢复范围：整集群回档。
  - 地域：选择新集群的地域，若您希望通过快照备份进行回档，这里需要选择和原集群一致的地域。
  - 主可用区：选择新集群对应地域的可用区。
  - 传输链路：高 IO 版。
  - 网络：选择私有网络。
  - 数据库版本：和被回档集群一致。
  - 实例类型：支持通用型、独享型。
  - 计算实例（读写）：选择读写实例的计算规格，仅支持配置一个读写实例。
  - 计算实例（只读）：选择只读实例的计算规格和数量。
  - 计算计费模式：支持包年包月、按量计费。
  - 存储计费模式：默认按量计费。
- **实例形态选择 Serverless**
  - 实例形态：Serverless。
  - 回档模式：支持按备份文件、按时间点回档。
    - 按备份文件：可从指定的备份集中恢复出新集群，选择范围基于备份完成的时间和保留时长。
    - 按时间点：可从具体某个时间点恢复出新集群，时间点选择范围基于备份保留时长。
  - 选择恢复范围：整集群回档。
  - 地域：选择新集群的地域，若您希望通过快照备份进行回档，这里需要选择和原集群一致的地域。
  - 可用区：选择新集群对应地域的可用区。
  - 传输链路：高 IO 版。
  - 网络：选择私有网络。
  - 数据库版本：和被回档集群一致。
  - Serverless 架构：支持选择单节点版、集群版。
  - 算力配置：选择算力配置 CCU（TDSQL-C Compute Unit）上下限，实例会根据选择资源范围自动进行弹性扩缩容。CCU（TDSQL-C Compute Unit）为 Serverless 的计算计费单位，一个 CCU 近似等于1个 CPU 和 2GB 内存的计算资源，每个计费周期的 CCU 使用数量为：数据库所使用的 CPU 核数 与 内存大小的1/2 二者中取最大值。算力配置可参考 服务算力配置 。
  - 自动暂停：配置实例自动暂停时间，在设定时间内无连接访问数据库会自动暂停实例，实例暂停后计算将不再计费。
  - 计算计费模式：支持选择按量计费或资源包。

> **说明：**

> 计算资源包，用来优先抵扣按量计费产品中实际产生的用量，资源包额度消耗完毕后，会继续按照按量计费模式计费。计算资源包会按照每秒实际使用的 CCU 量进行抵扣，相比于按量付费模式更为划算，使用更为灵活。

- 计算资源包（计算计费模式选择资源包时此项显示）：绑定资源包，可绑定当前账户下，在有效期内的，集群所选地域的所有计算资源包。若无可用资源包，可先 购买资源包 。
- 存储计费模式：支持选择按量计费或资源包。

> **说明：**
> 存储资源包，用来优先抵扣按量计费产品中实际产生的用量，资源包额度消耗完毕后，会继续按照按量计费模式计费。存储资源包会按照每小时实际使用的存储量进行抵扣，相比于按量付费模式更为划算，使用更为灵活。

- 存储资源包（存储计费模式选择资源包时此项显示）：绑定资源包，可绑定当前账户下，在有效期内的，集群所选地域的所有存储资源包。若无可用资源包，可先 购买资源包 。

5. 在跳转页面配置基本信息和高级配置，包括集群名称，默认字符集，安全组，参数模板及项目等，确认无误后单击**立即购买**。
6. 返回集群列表，可看到通过回档新建的集群，待状态变为**运行中**即可正常使用。

操作日志
慢日志概述

最近更新时间：2024-10-14 10:43:41

概述

慢日志也称慢 SQL，是 TDSQL-C MySQL 版提供的一种日志记录，用以记录数据库进程中响应时间超过阈值的 SQL 语句，具体指运行时间超过参数 long_query_time 值的 SQL，则会被记录到慢日志中。

参数

TDSQL-C MySQL 版参数 long_query_time 的值默认为1，表示运行1秒以上的 SQL 语句会被记录到慢日志，此参数支持用户手动调整，范围是：0.000000 - 3600.000000，调整参数请参见 设置实例参数 。

目的

通过此参数设置，我们可以查找出哪些 SQL 语句的执行效率低，以便进行优化。通俗的说，TDSQL-C MySQL 版慢日志是排查问题的 SQL 语句，以及检查当前 TDSQL-C MySQL 版性能的一个重要功能。

相关操作

操作项	操作场景	如何操作
预防慢日志产生	通过 DBbrain 获取优化建议，优化 SQL 语句。用于业务优化慢 SQL的场景。	SQL 优化
设置慢日志相关参数	通过设置参数 long_query_time 的值，记录慢日志。	设置实例参数
查看慢日志监控告警	通过实例监控指标（慢查询数），了解实例慢查询情况，通过设置告警来获取相关通知。	告警策略
查询和下载慢日志明细	在操作日志页，可查询慢日志信息包含：执行时间、SQL 语句、客户端地址、用户名、数据库、执行时长（秒）、加锁时长（秒）、解析行数、返回行数，也支持 CSV 和原生格式进行下载。	查询和下载慢日志明细
分析优化慢日志	通过 DBbrain 可查看完整的 SQL 模板、SQL 样例以及优化建议和说明，用以分析和优化 SQL。	慢 SQL 分析

版权所有：腾讯云计算（北京）有限责任公司

查询和下载慢日志明细

最近更新时间：2025-05-26 09:28:11

慢日志是用来记录在 TDSQL-C MySQL 版中对读写实例和只读实例执行时间超过指定时间的查询语句。通过查询慢日志明细，可以查找出哪些查询语句的执行效率低，以便进行优化。TDSQL-C MySQL 版支持下载慢日志明细文件，便于分析和作出对应优化。
本文为您介绍通过控制台查询和下载慢日志明细。

查询慢日志

1. 在集群列表页面，根据实际使用的视图模式，参考如下步骤，进入集群管理页。

页签视图

登录 TDSQL-C MySQL 版控制台，在左侧集群列表，单击**目标集群**，进入集群管理页。

列表视图

1. 登录 TDSQL-C MySQL 版。
2. 在上方选择地域，找到需要查询或下载慢日志的集群，单击**集群 ID** 或**操作**列的**管理**，进入集群管理页。

2. 在集群管理页，选择**操作日志**页，选择目标实例后，单击**慢日志明细**。



- 在慢日志明细列表，可查询全部、今天、昨天、近7天、近30天或自定义时间范围内的慢日志明细。
- 支持检索关键字段（客户端地址、用户名、数据库名）来快速过滤出目标慢日志明细，支持导出列表文件。
- 在慢日志明细列表，可查看明细信息，具体字段请参见 慢日志字段 。

说明：

- 远程读取次数、远程读取的字节数、远程读取所花费的时间（纳秒）、远程写入次数、远程写入的字节数、远程写入所花费的时间（纳秒）、事务提交延迟（微秒）这7个字段需要内核版本为 TXSQL 8.0，内核小版本为3.1.12或以上才能查看。如需展示在慢日志列表，您可单击右侧的设置图标，然后勾选所需查看的字段，单击**确认**。



- 通过**导出**功能，下载慢日志明细时，下载的明细里会包含所有字段信息，即使在设置里未勾选所有字段。

慢日志字段

> **注意：**
> 如果实例开启了只读分析引擎，则在初次加载数据到只读分析引擎时，会从读写实例拉取数据，从而产生慢 SQL。请忽略“txsqlPlugins_analytics_repl”账号产生的慢 SQL 信息。

中文名称	英文名称	描述
执行时间	Execution Time	执行的时间，示例：2024/9/22 23:56:12。
SQL 语句	SQL Statement	执行的 SQL 语句，示例：DROP DATABASE IF EXISTS db1。
客户端地址	Client Address	执行操作的客户端地址，示例：172.*.*.10。
用户名	Username	执行操作的用户名，示例：root。
数据库	Database	数据库名，示例：mysql。
执行时长（秒）	Duration (sec)	SQL 的执行时间，单位：秒。
加锁时长（秒）	Lock Time (sec)	指在执行某个操作过程中，从开始等待锁到实际获得锁并完成操作所花费的总时间，单位：秒。
解析行数	Parsed Rows	指在执行查询或操作时，成功解析并处理的行数。
返回行数	Returned Rows	指在执行查询或操作后，实际返回给客户端的行数。
远程读取次数	Number of remote reading times	指在执行查询或操作过程中，从存储层读取数据的次数。
远程读取的字节数	Number of remote read bytes	指在执行查询或操作过程中，从存储层读取的数据量，以字节为单位。
远程读取所花费的时间（纳秒）	Time spent on remote reading (microseconds)	指在执行查询或操作过程中，从存储层读取数据所耗费的时间，单位：纳秒。
远程写入次数	Number of remote writing times	指在执行写入操作过程中，向存储层写入数据的次数。
远程写入的字节数	Number of remote written bytes	指在执行查询或操作过程中，向存储层写入的数据量，以字节为单位。
远程写入所花费的时间（纳秒）	Time spent on remote writing (microseconds)	指在执行查询或操作过程中，向存储层写入数据所耗费的时间，单位：纳秒。
事务提交延迟（微秒）	Transaction submission delay (μs)	指在执行事务提交操作时，从开始提交事务到实际完成提交所经历的时间，单位：微秒。

下载慢日志明细

在集群列表页面，根据实际使用的视图模式进行操作：

页签视图

1. 登录 TDSQL-C MySQL 版控制台，在左侧集群列表，单击**目标集群**，进入集群管理页。
2. 在集群管理页，选择**操作日志**页，选择目标实例后，单击**慢日志明细**。
3. 通过筛选时间或检索关键字段，可查询需要的慢日志明细，然后单击**导出**。



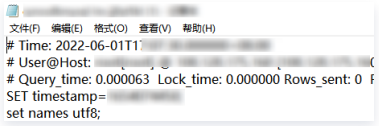
4. 在弹出的下载窗口，选择文件格式，单击**确定**进行本地下载。



- 您可将筛选或检索出的结果导出，导出格式支持 CSV 及原生格式（开源分析工具可识别）。
  - CSV 格式导出，可快捷检查和优化，方便使用。导出信息示例如下：

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
执行时间	sql 语句	客户端地址	用户名	数据库	执行时长(s	加锁时长(s	解析行数	返回行数	远程读取次	远程读取总	远程读取页	远程写入次	远程写入总	远程写入页	事务提交延迟（微秒）	
2024/9/22 23:56	select sleep	172.	root	h	3.000158	0	1	1	0	0	0	0	0	0	0	
2024/9/22 23:55	DROP DA	172.	root		6.162792	0.000838	0	0	0	0	0	0	0	0	0	
2024/9/22 3:42	select sleep	172.	root	h	3.000136	0	1	1	0	0	0	0	0	0	0	

  - 原生格式导出，能支持开源分析工具识别。导出信息示例如下：



- 一次最多只能导出2000条记录，超出部分请通过缩小时间范围的方式导出。

列表视图

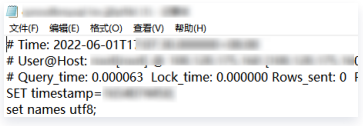
1. 登录 TDSQL-C MySQL 版。
2. 在上方选择地域，找到需要查询或下载慢日志的集群，单击**集群 ID** 或**操作**列的**管理**，进入集群管理页。
3. 在集群管理页，选择**操作日志**页，选择目标实例后，单击**慢日志明细**。
4. 通过筛选时间或检索关键字段，可查询需要的慢日志明细，然后单击**导出**。
5. 在弹出的下载窗口，选择文件格式，单击**确定**进行本地下载。



- 您可将筛选或检索出的结果导出，导出格式支持 CSV 及原生格式（开源分析工具可识别）。
  - CSV 格式导出，可快捷检查和优化，方便使用。导出信息示例如下：

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
执行时间	sql 语句	客户端地址	用户名	数据库	执行时长(s	加锁时长(s	解析行数	返回行数	远程读取次	远程读取总	远程读取页	远程写入次	远程写入总	远程写入页	事务提交延迟（微秒）	
2024/9/22 23:56	select sleep	172.	root	h	3.000158	0	1	1	0	0	0	0	0	0	0	
2024/9/22 23:55	DROP DA	172.	root		6.162792	0.000838	0	0	0	0	0	0	0	0	0	
2024/9/22 3:42	select sleep	172.	root	h	3.000136	0	1	1	0	0	0	0	0	0	0	

  - 原生格式导出，能支持开源分析工具识别。导出信息示例如下：



- 一次最多只能导出2000条记录，超出部分请通过缩小时间范围的方式导出。

错误日志

最近更新时间：2024-10-23 11:13:21

TDSQL-C MySQL 版提供错误日志记录，本文为您介绍错误日志的定义，以及查询和下载错误日志的操作方法。

定义

错误日志是 TDSQL-C MySQL 版提供的一种日志记录，用于记录数据库运行过程中，因操作、执行 SQL、系统运行错误等原因产生的警告、错误和其他重要事件的日志，可以帮助数据库管理员找出业务系统或数据库出现问题的原因。

查询错误日志

TDSQL-C MySQL 版在集群管理页的**操作日志**页面下，提供错误日志的查询能力。

1. 在集群列表页面，根据实际使用的视图模式，参考如下步骤，进入集群管理页。

页签视图

登录 TDSQL-C MySQL 版控制台，在左侧集群列表，单击目标集群，进入集群管理页。

列表视图

1. 登录 TDSQL-C MySQL 版。
2. 在上方选择地域，找到需要查询错误日志的集群，单击**集群 ID** 或**操作**列的**管理**，进入集群管理页。

2. 在集群管理页，选择**操作日志**页，选择目标实例后，单击**错误日志明细**。



- 在错误明细列表，可查询自定义时间范围内的错误日志明细，最长支持查询的时间范围是1个月。
- 日志等级：error，warning，note，支持全选。
- 支持检索关键词来快速过滤出目标错误日志明细，支持导出列表文件。

错误日志字段

中文名称	英文名称	描述
采集时间	Collection Time	指错误信息被捕捉并记录到日志文件中的时间戳，示例：2024-09-30 11:09:52。
日志内容	Log Content	指记录的具体错误信息、警告或其他重要事件的详细描述。
日志等级	Log Level	用于标识日志消息重要性或严重程度的一个分类。

下载错误日志明细

说明：

- 根据页面筛选结果，导出错误日志明细，即导出到本地的错误日志明细为您筛选后的内容。
- 一次最多只能导出2000条记录，超出部分请通过缩小时间范围的方式导出。

1. 在集群列表页面，根据实际使用的视图模式，参考如下步骤，进入集群管理页。

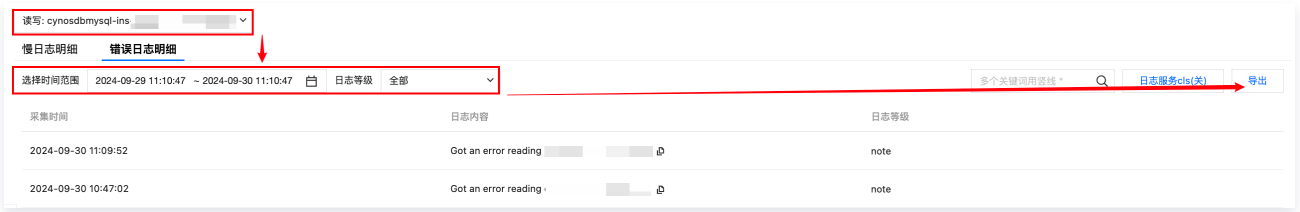
页签视图

登录 TDSQL-C MySQL 版控制台，在左侧集群列表，单击目标集群，进入集群管理页。

列表视图

1. 登录 TDSQL-C MySQL 版。
2. 在上方选择地域，找到需要下载错误日志的集群，单击**集群 ID** 或**操作**列的**管理**，进入集群管理页。

2. 在集群管理页，选择**操作日志**页，选择目标实例后，单击**错误日志明细**。
3. 选择目标实例，筛选需要的时间范围和日志等级，然后单击**导出**。



4. 在弹出的下载窗口，选择文件格式，单击**确定**进行本地下载。



日志服务 CLS

最近更新时间：2024-10-23 11:42:01

TDSQL-C MySQL 版提供日志服务（CLS）功能，可采集来源为 TDSQL-C MySQL 版实例的慢日志或错误日志数据，并投递至日志服务（Cloud Log Service，CLS）进行分析，实现从日志采集，日志存储到日志检索等全方位的日志服务，以便快速监控和定位业务问题。本文为您介绍如何通过控制台开启或关闭日志服务（CLS）功能。

> **说明：**
> 本功能陆续发布中，如您的实例所在地域暂未支持日志服务 CLS，请 提交工单 反馈。

前提条件

使用该功能前，请确保您已开通 日志服务 CLS 。

慢日志字段值说明

字段值	类型	说明
TIMESTAMP	-	CLS 的保留字段，代表 志产 时间。
instance_id	String	数据库实例 ID，例如 cynosdbmysql-ins-xxxx。
db_name	String	数据库名。
rows_examined	Long	扫描的行数。
rows_sent	Long	返回的行数。
query_time	Double	执 耗时（单位 s）。
lock_time	Double	锁等待耗时（单位 s）。
user_host	String	客户端信息。
user_name	String	客户端连接数据库实例所使用的 户名。
query_sql	String	慢 志 SQL。

开启日志服务（CLS）

1. 在控制台根据实际使用的视图模式进入集群的操作日志页面。

页签视图

1. 登录 TDSQL-C MySQL 版控制台，在左侧集群列表，单击目标集群，进入集群管理页。
2. 在集群管理页，选择**操作日志**页。

列表视图

1. 登录 TDSQL-C MySQL 版，在上方选择地域，找到目标集群，单击**集群 ID** 或**操作**列的**管理**，进入集群管理页。
2. 在集群管理页，选择**操作日志**页。



2. 在操作日志页面，选择需要开启日志投递服务的实例，进行对应操作。

- 如需开启慢日志投递，请选择**慢日志明细**，再单击**日志服务 CLS（关）**。
- 如需开启错误日志投递，请选择**错误日志明细**，再单击**日志服务 CLS（关）**。



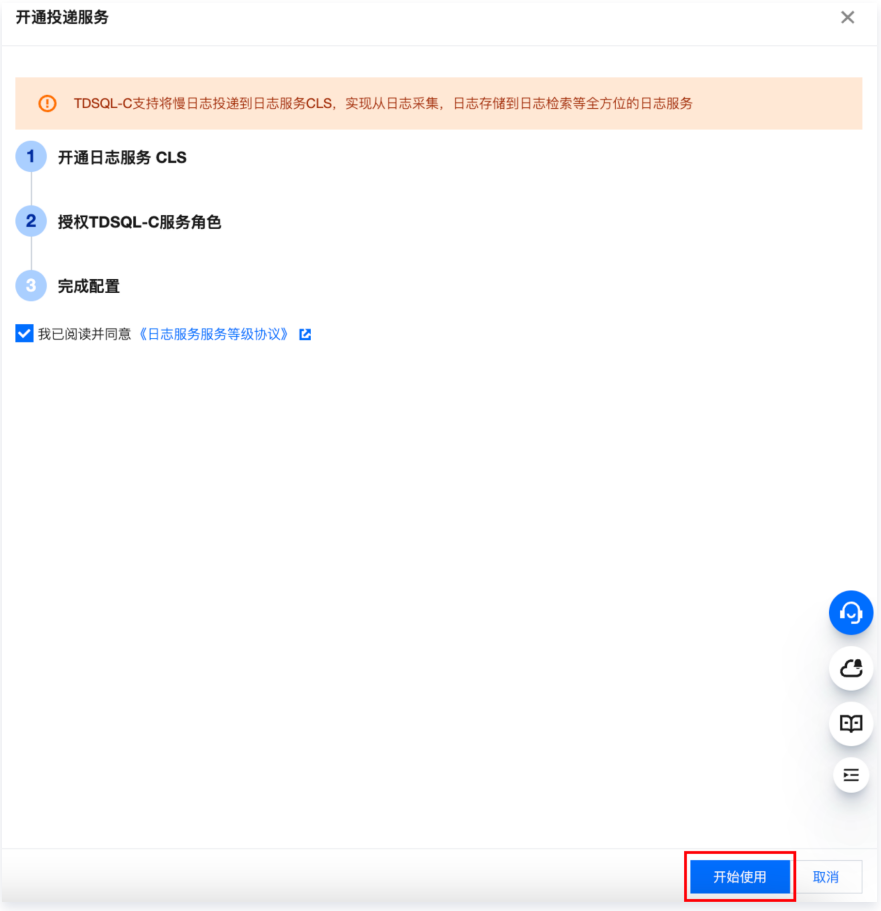
3. 在**开通投递服务**弹窗下，单击**前往开通**，为 TDSQL-C 授权服务角色（如已授权，可跳过步骤3 – 步骤5）。



4. 在跳转的**角色管理**页面，单击**同意授权**。



5. 返回**开通投递服务**弹窗，阅读并勾选《日志服务服务等级协议》，单击**开始使用**。



6. 单击**添加投递日志集**，进行日志投递配置。



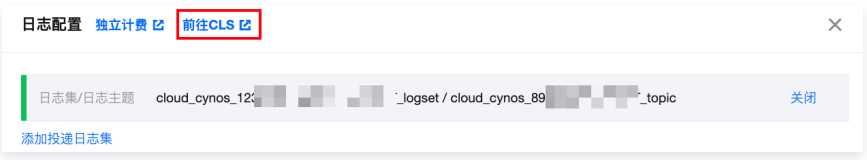
7. 在弹窗中完成如下配置，单击**立即开启**。



参数	说明
地域	选择日志集投递的地域。
日志集操作	日志集是对日志主题的分类，方便您管理日志主题。支持选择已有日志集或者创建日志集。
日志集	• 选择已有日志集：可在搜索框筛选已有的日志集，作为日志投递的归类。 • 创建日志集：可新建日志集作为日志投递的归类，创建日志集的命名格式为 **cloud_cynos_自定义_logset**，其中自定义部分仅支持英文字母、数字、下划线、短横线。
日志主题操作	日志主题是日志数据进行采集、存储、检索和分析的基本单元。支持选择已有日志主题或者创建日志主题。
日志主题	• 选择已有日志主题：仅当日志集操作选择已有日志集时，此项可设置。可在搜索框筛选所选日志集下的日志主题。 • 创建日志主题：在所选日志集下创建新的日志主题，创建日志主题的命名格式为 **cloud_cynos_自定义_topic**，其中自定义部分仅支持英文字母、数字、下划线、短横线。 **说明：** 您可对日志主题进行管理，详细请参见 管理日志主题 。

8. 成功开启日志服务后，您可在日志配置弹窗下看到创建的**日志集/日志主题**，单击**前往 CLS**，可跳转至 日志服务控制台 进行后续分析管理。

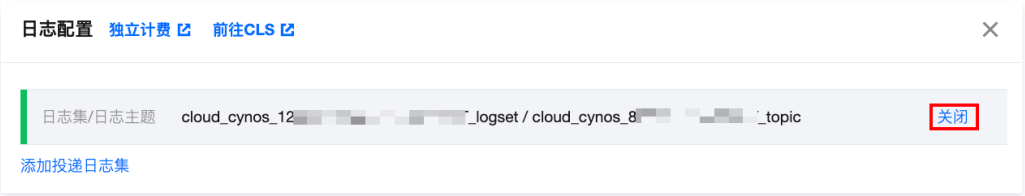
说明：
目前，在开启日志服务后进行日志投递时，默认是未开启索引的，而索引配置是使用日志服务（CLS）进行检索分析的必要条件，只有开启索引才能对日志进行检索分析。如需开启索引，请参见 索引配置（CLS）。



关闭日志服务（CLS）

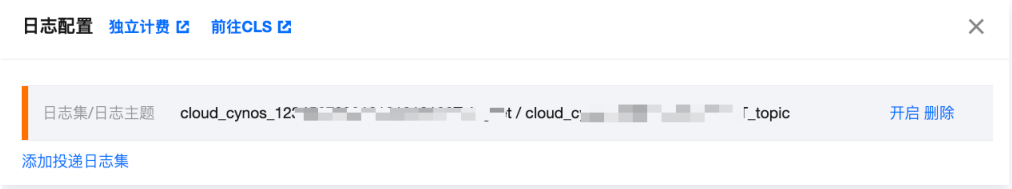
> **说明：**
> 关闭后将停止向对应主题投递日志，已投递日志将保留，若需要删除日志，请前往 日志服务控制台 删除。

1. 参考 开启日志服务（CLS）> 步骤1 进入操作日志页面。
2. 在操作日志页面，选择需要关闭日志投递服务的实例，如果需要关闭的是慢日志的日志投递，请单击**慢日志明细 > 日志服务 CLS（开）**，如果需要关闭的是错误日志的日志投递，请单击**错误日志明细 > 日志服务 CLS（开）**。
3. 在日志配置弹窗下，找到目标**日志集/日志主题**，单击**关闭**。



4. 在弹窗下单击**确定**。

日志配置页面功能说明



功能项	说明
开启	对已创建的日志投递配置（关闭状态）进行重新启用，启用后将向 CLS 侧重新投递日志，期间将持续收取日志存储费用。
删除	对已创建的日志投递配置（关闭状态）进行删除，删除投递后历史日志仍持续存储在日志主题中，直至过期，期间将持续收取日志存储费用，若不希望持续存储历史日志可前往 日志服务控制台 删除日志主题。
添加投递日志集	新建投递日志集，操作步骤参考 开启日志服务（CLS）> 步骤6 – 步骤8 。

相关文档

- 日志服务计费概述
- 日志集
- 管理日志主题
- 仪表盘
- 数据加工
- 检索分析

迁移数据
一键迁移（从云数据库 MySQL 迁移至 TDSQL-C MySQL 版）
简介

最近更新时间：2024-08-01 16:04:12

本文为您介绍一键迁移功能简介。

背景

云原生数据库 TDSQL-C MySQL 版100%兼容云数据库 MySQL，具备超百万级 QPS 数据吞吐量，支持最高 PB 级海量存储，保证数据安全可靠，在同等规格情况下，TDSQL-C MySQL 版价格更实惠，性能更强劲。
当前已支持在控制台，通过全量 + 增量数据进行一键迁移，适用于从云数据库 MySQL 迁移至云原生数据库 TDSQL-C MySQL 版的场景，可实现读写实例不停机平滑迁移，且保留原有的 IP 地址、监控及参数等各项设置。
迁移源端：云数据库 MySQL。
迁移目标端：TDSQL-C MySQL 版。

使用限制

- 仅支持云数据库 MySQL 5.7、8.0的双节点版和三节点版一键迁移至 TDSQL-C MySQL 版。

> **说明：**
> 一键迁移无内核小版本限制，迁移至 TDSQL-C MySQL 版后，该集群将自动升级为最新的内核小版本。

- 目前仅支持迁移云数据库 MySQL 的主实例和只读组，灾备实例迁移正在开发中，敬请期待。
- 目前仅支持迁移同地域同可用区的实例。
- 目前仅支持迁移相同计费模式的实例，如果迁移源端某个云数据库 MySQL 主实例的计费模式为包年包月，则其只读实例计费模式也需要为包年包月，或者主实例和只读实例都为按量计费才支持一键迁移，调整实例的计费模式请参见 按量转包年包月 。

功能优势

- 功能完全免费，无需 DTS 工具。
- 迁移过程数据0丢失。
- 支持增量迁移。
- 支持在线热迁移，迁移过程仅会闪断一次（即当连接地址从云数据库 MySQL 切换至 TDSQL-C MySQL 版时）。
- 迁移后可保留数据库原连接地址，无需应用程序修改任何连接配置即可切换至 TDSQL-C MySQL 版。
- 无需手动退费再新购，连接地址切换时后台自动操作计费流程，省去繁琐步骤。

迁移规格对照表

当前一键迁移支持的云数据库 MySQL 实例规格与迁移后 TDSQL-C MySQL 版对应的规格如下：

> **注意：**
> MySQL 实例规格与 TDSQL-C MySQL 版支持的实例规格略有差异，具体实例规格迁移对比请以控制台一键迁移时的对比为准。



MySQL 双节点版规格	MySQL 三节点版规格	TDSQL-C MySQL 版规格

1核1000MB	1核1000MB	1核1GB
1核2000MB	1核2000MB	1核2GB
2核4000MB	2核4000MB	2核4GB
2核16000MB	2核16000MB	2核16GB
4核8000MB	4核8000MB	4核8GB
4核16000MB	4核16000MB	4核16GB
4核24000MB	4核24000MB	4核24GB
4核32000MB	4核32000MB	4核32GB
8核16000MB	8核16000MB	8核16GB
8核32000MB	8核32000MB	8核32GB
8核48000MB	8核48000MB	8核48GB
8核64000MB	8核64000MB	8核64GB
12核48000MB	12核48000MB	12核48GB
12核72000MB	12核72000MB	12核72GB
12核96000MB	12核96000MB	12核96GB
16核32000MB	16核32000MB	16核64GB
16核64000MB	16核64000MB	16核64GB
16核96000MB	16核96000MB	16核96GB
16核128000MB	16核128000MB	16核128GB
24核96000MB	24核96000MB	24核96GB
24核128000MB	24核128000MB	24核144GB
24核144000MB	24核144000MB	24核144GB
24核192000MB	24核192000MB	24核192GB
24核244000MB	24核244000MB	32核256GB
24核256000MB	24核256000MB	32核256GB
32核128000MB	32核128000MB	32核128GB
32核192000MB	32核192000MB	32核192GB
32核256000MB	32核256000MB	32核256GB
48核192000MB	48核192000MB	48核192GB
48核288000MB	48核288000MB	48核288GB
48核384000MB	48核384000MB	48核384GB
48核488000MB	48核488000MB	48核488GB
64核256000MB	64核256000MB	64核256GB
64核384000MB	64核384000MB	64核384GB
64核512000MB	64核512000MB	88核710GB

版权所有：腾讯云计算（北京）有限责任公司

80核690000MB	80核690000MB	88核710GB
90核720000MB	90核720000MB	88核710GB

备份策略

- 目前 TDSQL-C MySQL 版备份空间暂不收费，您可免费使用备份功能。
- 目前 TDSQL-C MySQL 版支持逻辑备份和快照备份，迁移至 TDSQL-C MySQL 版后默认采用自动快照备份策略进行备份，快照备份具有备份时用户无感知、秒级恢复速度等优势，您也可以在迁移后根据需要进行逻辑备份，操作请参见 开启自动逻辑备份 以及 手动备份 。

> **说明：**
> 备份周期规则：自动备份中的快照备份，默认情况下每2日进行一次备份，如果写入量大于4G且距离上次备份超过24小时则立即进行备份。后续您也可以根据业务需求，在备份管理页开启自动备份中的逻辑备份，同时也支持对自动备份中逻辑备份的保留时间进行设置。

- 目前 TDSQL-C MySQL 版支持设置备份文件保留周期功能，如需回档，您可按照备份文件或者按时间点进行回档。

执行一键迁移

最近更新时间：2024-10-23 11:23:21

本文为您介绍通过控制台，从云数据库 MySQL 一键迁移至云原生数据库 TDSQL-C MySQL 版。

> **说明：**
> 当前版本的一键迁移功能，暂时仅支持同地域同可用区的迁移。

前提条件

- 迁移源端云数据库 MySQL 版本为 MySQL 5.7、8.0 的双节点版或三节点版。
- 迁移源端的云数据库 MySQL 实例没有灾备实例。
- 迁移源端的云数据库 MySQL 实例下如果有只读实例，则主实例和只读实例的计费模式需一致，调整实例的计费模式请参见 按量转包年包月 。

操作步骤

使用一键迁移，将根据 迁移规格对照表 创建一个规格对应的 TDSQL-C MySQL 版集群，并进行数据库迁移，在迁移完成后，需进行连接地址切换，详细操作流程如下。

一、开始迁移

1. 登录 云数据库 MySQL 控制台 。
2. 在实例列表勾选1个需要迁移的 MySQL 实例，单击上方的**一键升级**。



> **说明：**
> - 在实例列表，在字段**计费模式**下也有**一键升级**的按钮。
> - 单击**实例 ID** 进入实例详情页，在右上方也有**一键升级**的按钮。

3. 在弹窗下选择**迁移至 TDSQL-C MySQL**，单击**立即升级**。
4. 进入迁移界面后，预览当前 MySQL 实例与迁移后的新 TDSQL-C MySQL 版集群的对比信息。



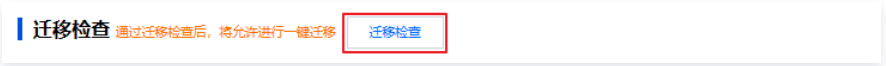
字段	说明
实例类型	读写实例或只读实例，迁移前后实例类型不变。
规格	对比迁移前后的实例规格配置。
可用区	对比迁移前后的可用区部署情况。
配置费用	对比迁移前后的配置费用。
到期时间	对比迁移前后实例的到期时间。

> **说明：**
> - 针对迁移源端为包年包月计费模式的云数据库 MySQL 实例，迁移后，TDSQL-C MySQL 版的到期时间计算方式有如下说明：云数据库 MySQL 当前时间进行退款后，由于退款机制为按使用量退款，而非按使用时间退款，所以退款较少，TDSQL-C MySQL 版将利用这部分差额

进行集群购买，差额/对应规格价格，即为 TDSQL-C MySQL 版集群到期时间。

- 源端云数据库 MySQL 进行退款时，退款机制为按使用量退款，若系统核算退款差额不足以购买新的 TDSQL-C MySQL 版集群，则无法使用一键迁移。
- 若源端云数据库 MySQL 剩余金额无法支持一键迁移，建议您手动 购买 TDSQL-C MySQL 版集群，然后进行数据导入。

5. 在迁移界面，单击**迁移检查**，系统会对需要迁移的云数据库 MySQL 实例与迁移到 TDSQL-C MySQL 版后实例的信息进行检查，您可根据检查结果对应操作，若迁移源端实例下存在只读组，则支持选择是否迁移只读组。



检查项如下

检查项	说明
可用区	TDSQL-C MySQL 版暂不支持部分可用区，如果迁移源端云数据库 MySQL 实例部署在不支持的可用区，则暂时无法迁移。
TDSQL-C 规格库存	后台判定新的集群规格库存是否满足，若满足，则可迁移。
InnoDB 引擎	TDSQL-C MySQL 版仅支持 InnoDB 引擎迁移，若需要迁移的 MySQL 实例为 RocksDB 引擎，则无法迁移。
非 InnoDB 表单	TDSQL-C MySQL 版仅支持 InnoDB 引擎迁移，若需要迁移的 MySQL 实例下存在非 InnoDB 表单，则无法迁移。
云数据库 MySQL 版本	迁移源端云数据库 MySQL 版本须为 MySQL 5.7、8.0 的双节点版或三节点版，若不是，则无法迁移。
外网开通	如需要迁移的 MySQL 实例开启了外网，则迁移过程中源端 MySQL 实例的外网无法关闭，迁移后目标端 TDSQL-C MySQL 版集群的外网状态为未开启，如迁移完成后需要使用源端的 MySQL 实例，可从 MySQL 侧回收站恢复实例，实例恢复后外网状态为未开启。
具有灾备实例	如需要迁移的 MySQL 实例具有灾备实例，则无法迁移，当前版本无法迁移灾备实例。
数据加密	如需要迁移的 MySQL 实例开启了数据加密功能，则无法迁移，TDSQL-C MySQL 版暂不支持数据加密。
数据库代理	无论迁移源端云数据库 MySQL 实例是否开启数据库代理，当前版本均支持迁移。若迁移源端实例开启了数据库代理，则一键迁移至 TDSQL-C MySQL 版后，新集群也默认开启了数据库代理，但需注意，迁移后，数据库代理的内核小版本为当前最新支持的版本，数据库代理的相关访问策略和配置不会保留，您需要重新手动设置，详细请参见 查看和修改访问策略 。
排序规则	后台会判定字符集的排序规则是否适配，若不适配则不支持迁移。
具有不同可用区实例	后台会判定迁移源端云数据库 MySQL 实例的主实例和只读实例是否在同一可用区，如在同一可用区，则支持迁移，若跨可用区，则不支持迁移。
计算计费模式	迁移源端云数据库 MySQL 实例的主实例和只读实例计费模式需一致，若不一致则不支持迁移。
IP 库存检查	后台会检查 IP 库存是否充足，若不足，则不支持迁移。
自定义密码复杂度	如需要迁移的 MySQL 实例开启了自定义密码复杂度功能，则无法迁移，建议关闭，允许一键修复。
SSL	如需要迁移的 MySQL 实例开启了 SSL，则无法迁移，建议关闭，允许一键修复。

是否迁移只读组

若迁移源端实例下存在只读组，此项才会显示。默认开启按钮，如无需迁移只读组，您也可以手动关闭按钮。



情形一：迁移检查不通过

若迁移检查不通过，将无法进行迁移，部分选项（自定义密码复杂度、SSL）提供**一键修复**功能，您可单击**一键修复**，修复成功后可重新执行一键迁移，部分选项不支持**一键修复**，例如迁移源端实例开启了数据加密，则无法一键修复，无法进行迁移。

说明：

单击**下载报告**，可下载迁移检查结果到本地。

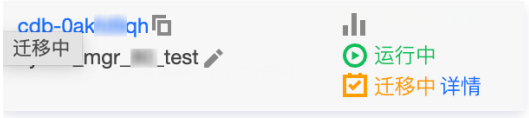


情形二：迁移检查通过

若迁移检查通过，您可单击**一键迁移**开始迁移。

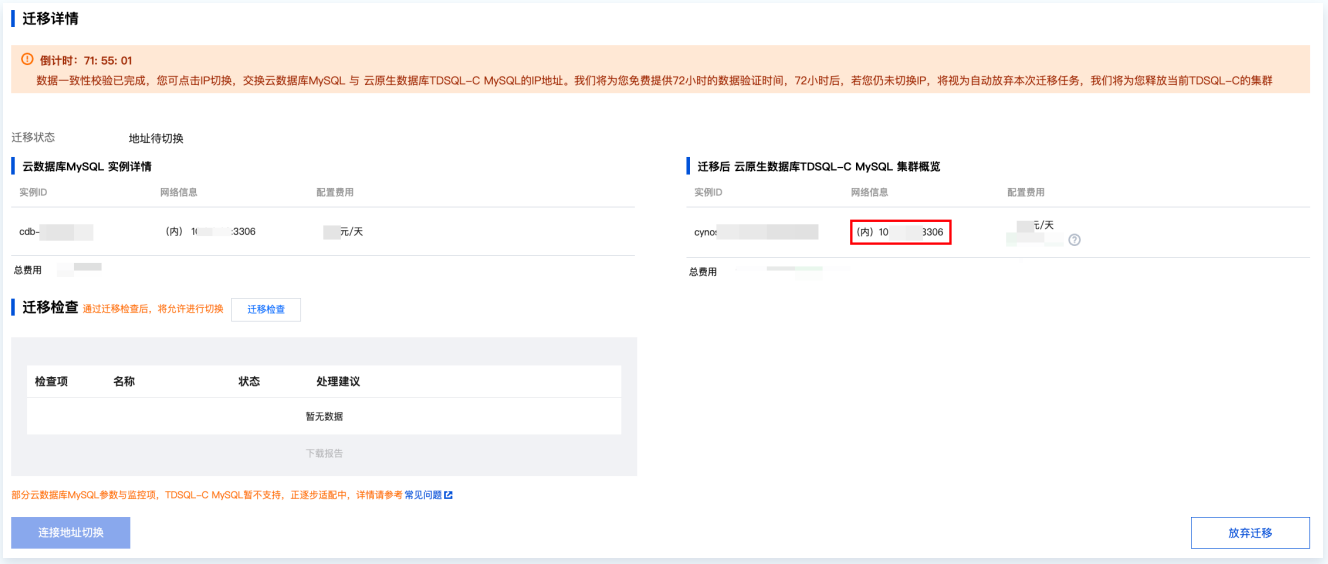
二、迁移中

一键迁移将采用迁移源端云数据库 MySQL 实例最近一次全量备份文件，在 TDSQL-C MySQL 版新建集群并持续进行数据同步，迁移期间云数据库 MySQL 实例显示迁移状态为**迁移中**，在 TDSQL-C MySQL 版集群列表可以看到新建的目标集群，迁移状态也为**迁移中**。



说明：

在 TDSQL-C MySQL 版集群列表，如果新建的目标集群状态由**迁移中**变为**待切换**，则表示数据同步已完成，此时您可单击**待切换**或**详情**，进入连接地址切换页面，可根据页面提供的目标集群连接地址，连接目标集群自行进行数据查看和校验，确认无误后可按如下步骤进行连接地址切换。



连接地址切换

1. 在 TDSQL-C MySQL 版集群列表，找到新建的集群，在集群状态下，单击**详情**。
2. 在跳转的页面下，单击**迁移检查**后，再单击**连接地址切换**，然后在弹窗下单击**确认切换**。

说明：

- 切换会造成秒级闪断，需要确保业务具有重连机制。
- 点击地址连接切换后，IP 地址交换，数据将写入 TDSQL-C MySQL 版新建的集群。
- 地址待切换状态将存在**72小时**，若超时未操作，则视为放弃迁移，系统将自动取消迁移流程。

三、迁移完成

连接地址切换后，视为迁移完成，您的 TDSQL-C MySQL 版集群将获得云数据库 MySQL 实例的 IP 地址，您的业务将切换至新集群下。源端云数据库 MySQL 实例将进入回收站进行隔离，如有需要，您可以进入云数据库 MySQL 的回收站进行恢复。后续您可登录 TDSQL-C MySQL 版控制台 对迁移后的新集群进行管理。

四、迁移回滚

> **说明：**
> 本版本的一键迁移暂不支持回滚功能，请在连接地址切换前做好业务检查。若后续需要重新使用迁移源端的云数据库 MySQL 实例，需在有效期内进入云数据库 MySQL 侧回收站进行恢复，**恢复后连接地址将改变，需重新适配应用程序**。

常见问题

最近更新时间：2024-03-29 11:48:21

本文为您介绍一键迁移功能相关的常见问题。

1、一键迁移后，迁移源端云数据库 MySQL 支持的部分高级功能参数在迁移目标端 TDSQL-C MySQL 版上也支持吗？

暂不全部支持，TDSQL-C MySQL 版将逐步进行适配。迁移后，TDSQL-C MySQL 版具体是否存在云数据库 MySQL 支持的高级功能参数，请参见下表。

MySQL 5.7 版本

MySQL 侧参数名	MySQL 侧参数值	TDSQL-C MySQL 侧参数是否存在
cdb_recycle_bin_enabled	OFF	否
information_schema_stats_expiry	86400	否
binlog_row_event_max_size	8192	否
innodb_backquery_history_limit	8000000	否
innodb_ddl_threads	4	否
innodb_parallel_ddl	OFF	否
innodb_adaptive_hash_index	OFF	否
cdb_plan_cache	OFF	否
innodb_table_drop_mode	sync_drop	否
cdb_recycle_bin_retention	604800	否
slave_net_timeout	120	否
innodb_backquery_enable	OFF	否
innodb_fast_ddl	OFF	否
max_length_for_sort_data	1024	是
slave_parallel_type	LOGICAL_CLOCK	否
cdb_optimize_large_trans_binlog	OFF	否
innodb_fast_ahi_cleanup_for_drop_table	OFF	否
txsql_parallel_exchange_buffer_size	1048576	否
innodb_temp_data_file_path	ibtmp1:12M:autoextend	否
slave_rows_search_algorithms	TABLE_SCAN,INDEX_SCAN,HASH_SCAN	否
cdb_recycle_scheduler_interval	0	否
innodb_ddl_buffer_size	10485760	否
cdb_kill_idle_trans_timeout	0	是
collation_server	utf8_tolower_ci	是
delay_key_write	ON	否

版权所有：腾讯云计算（北京）有限责任公司

innodb_async_truncate_size	128	否
cdb_more_gtid_feature_supported	OFF	否
cdb_opt_outline_enabled	OFF	否
innodb_txsql_parallel_partitions_per_worker	13	否
cdb_kill_user_extra	root@%	是
slave_parallel_workers	0	否
innodb_backquery_window	900	否

MySQL 8.0 版本

MySQL 侧参数名	MySQL 侧参数值	TDSQL-C MySQL 侧参数是否存在
cdb_recycle_bin_enabled	OFF	否
information_schema_stats_expiry	86400	否
binlog_row_event_max_size	8192	否
innodb_backquery_history_limit	8000000	是
innodb_ddl_threads	4	否
innodb_parallel_ddl	OFF	否
innodb_adaptive_hash_index	OFF	否
cdb_plan_cache	OFF	否
innodb_table_drop_mode	sync_drop	否
cdb_recycle_bin_retention	604800	否
slave_net_timeout	120	否
innodb_backquery_enable	OFF	是
innodb_fast_ddl	OFF	否
max_length_for_sort_data	1024	否
slave_parallel_type	LOGICAL_CLOCK	否
cdb_optimize_large_trans_binlog	OFF	否
innodb_fast_ahi_cleanup_for_drop_table	OFF	否
txsql_parallel_exchange_buffer_size	1048576	否
innodb_temp_data_file_path	ibtmp1:12M:autoextend	否
slave_rows_search_algorithms	TABLE_SCAN,INDEX_SCAN,HASH_SCAN	否
cdb_recycle_scheduler_interval	0	否
innodb_ddl_buffer_size	10485760	否

版权所有：腾讯云计算（北京）有限责任公司

cdb_kill_idle_trans_timeout	0	是
collation_server	utf8_tolower_ci	是
delay_key_write	ON	否
innodb_async_truncate_size	128	否
cdb_more_gtid_feature_supported	OFF	否
cdb_opt_outline_enabled	OFF	否
innodb_txsql_parallel_partitions_per_worker	13	否
cdb_kill_user_extra	root@%	是
slave_parallel_workers	0	否
innodb_backquery_window	900	是

2、一键迁移后，迁移源端云数据库 MySQL IOPS 监控项在迁移目标端 TDSQL-C MySQL 版上支持吗?

暂不支持，TDSQL-C MySQL 版将逐步进行适配。

3、如果迁移源端云数据库 MySQL 实例部署的可用区是 TDSQL-C MySQL 版不支持的，如何继续一键迁移?

如果迁移源端云数据库 MySQL 实例下仅有主实例，或者其主实例和只读实例在同一可用区，遇到如上问题时，您可以首先在 TDSQL-C MySQL 版购买页查询与迁移源端实例同地域的可用区有哪些，如广州四区、广州六区，然后在云数据库 MySQL 控制台将目标实例 迁移可用区 至广州四区或广州六区，最后重新尝试一键迁移即可。

使用 DTS 服务迁移

最近更新时间：2024-10-12 17:07:31

TDSQL-C MySQL 版可使用 数据传输服务 DTS 的数据迁移功能导入数据至 TDSQL-C MySQL 版，详情请参见 数据迁移 。

版权所有：腾讯云计算（北京）有限责任公司

通过命令行工具迁移

最近更新时间：2024-10-12 16:59:08

TDSQL-C MySQL 版支持通过命令行工具迁移数据。

命令行工具迁移数据

1. 使用 MySQL 命令行工具 mysqldump 生成待导入的 SQL 文件，方式如下：

> **注意**
> 使用 mysqldump 导出的数据文件必须兼容所购买的云原生数据库 TDSQL-C（兼容 MySQL 版）的 SQL 规范，可登录云数据库通过 `select version();` 获取相应的 MySQL 版本信息。生成的 SQL 文件名称允许英文/数字/下划线，但不能包含 “test” 字符。

```
shell > mysqldump [options] db_name [tbl_name ...] > bak_pathname
```

其中，options 为导出选项，db_name 为数据库名称，tbl_name 为表名称，bak_pathname 为导出路径名。
更多 mysqldump 导出数据说明，请参见 MySQL 官方手册 。

2. 还原数据库时，可以通过 MySQL 命令行工具进行还原，方式如下：

```
shell > mysql -h hostname -P port -u username -p < bak_pathname
```

其中，hostname 为还原数据的目标主机，port 为目标主机的端口，username 为目标主机的数据库用户名，bak_pathname 为备份文件的完整路径。

Windows 迁移数据

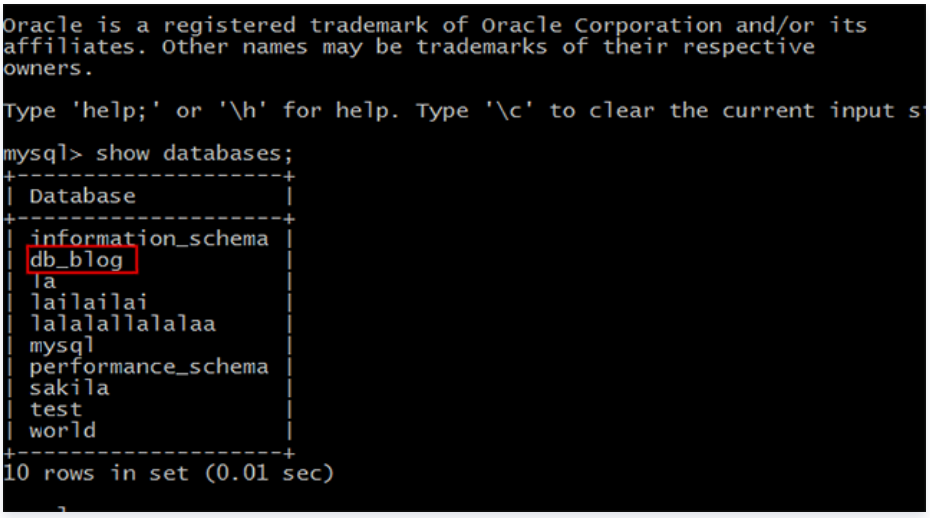
1. 使用 Windows 系统的 mysqldump 工具 dump 出数据，具体请参考 命令行工具迁移数据 中的描述。

> **注意**
> 请保证源和目标数据库版本一致，mysqldump 工具版本一致，源和目标数据库字符集一致。您可以通过参数 ``--default-character-set`` 来指定字符集。

2. 进入命令提示符，通过 MySQL 命令行工具进行还原。



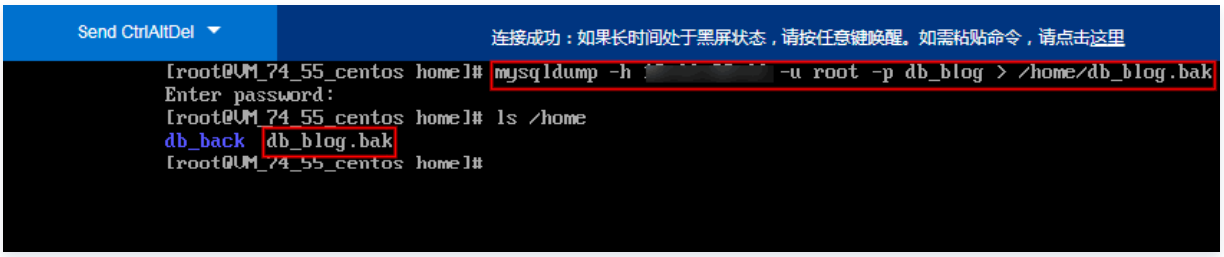
3. 登录 MySQL 数据库可以查看到备份的数据库已经还原到服务器内。



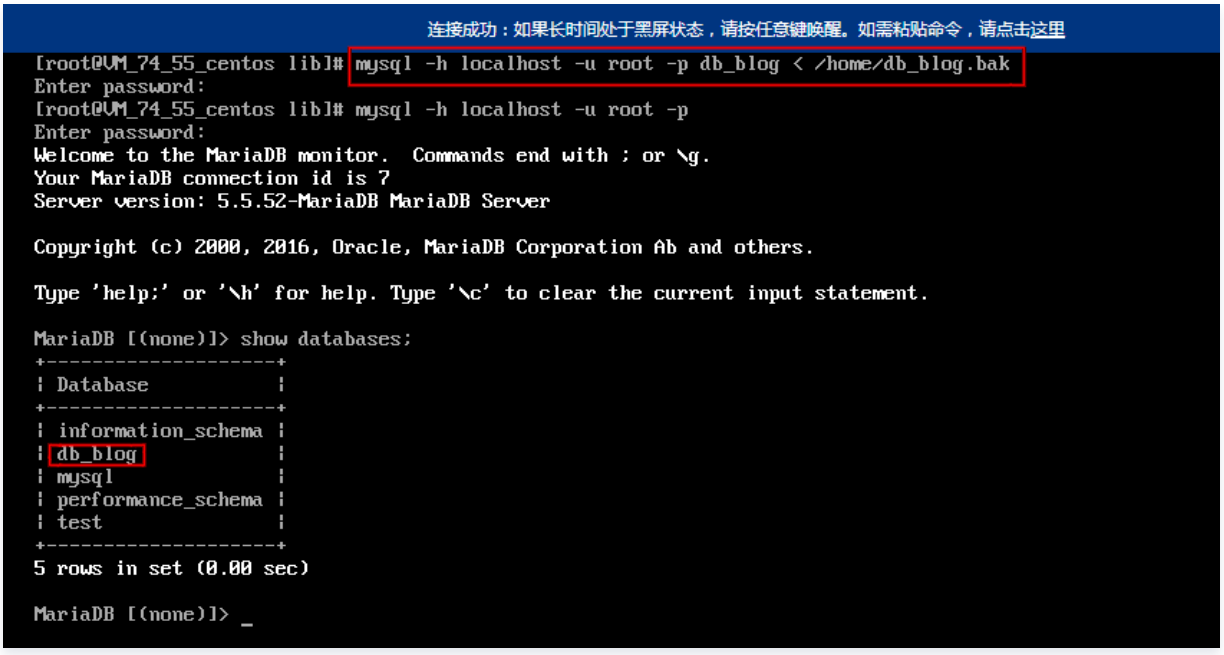
Linux 系统迁移数据（通过 CVM 主机）

CVM 主机访问数据库请参见 访问 TDSQL-C 集群。

1. 以云数据库上的 db_blog 数据库为例。登录 CVM 主机，使用 MySQL 命令行工具 mysqldump 生成待导入的 SQL 文件。



2. 通过 MySQL 命令行工具进行还原，本例将数据还原到 CVM 服务器上。可以查看到备份的数据库已导入到目标服务器对应的数据库中。



导入数据文件字符集编码问题

1. 如果云数据库导入数据文件没有指定字符集编码，以云数据库设置的字符集编码执行。
2. 如果导入数据文件中有指定的字符集编码，则以指定的字符集编码执行。
3. 如果导入的数据文件的字符集编码与云数据库当前字符集编码不同，会造成乱码。

更多字符集编码问题，请参见 使用限制 的字符集说明。

版权所有：腾讯云计算（北京）有限责任公司

并行查询
简介
背景

最近更新时间：2024-10-11 11:33:40

TDSQL-C MySQL 版支持并行查询能力，帮助提升对大表、计算量较大的一类查询的查询效率。本文介绍并行查询功能的背景。

功能背景

传统基于 MySQL 的关系型数据库在进行云原生改造，部署到云上后，仍存在存储冗余，弹性负载能力不强的情况，但 TDSQL-C MySQL 版利用“日志即存储”的架构，将计算与存储分离，各计算节点通过分布式文件系统共享底层存储，降低了用户成本，同时实现了无状态计算节点，使得计算资源在秒级时间内完成弹性扩展与故障恢复。
TDSQL-C MySQL 版针对于传统基于 MySQL 的关系型数据库在计算性能、存储能力、容灾能力和弹性扩展能力上的痛点问题已经解决并有所突破，但仍然存在以下痛点问题：

- 随着互联网的发展，云原生数据库对存储能力和存储量级的提升逐渐增强，TDSQL-C MySQL 版也已经支持到 PB 级数据量，大数据量的表单出现的频次越来越高，线上部分用户的表单已达到 TB 级别，对于大表查询能力，现有的技术瓶颈导致 SQL 语句响应过慢，影响业务流程。
- 现行的市场环境中，业务上出现越来越多的 AP 查询，例如报表统计或者其他分析查询，这些查询虽然不多，但通常要处理比较大的数据量且对查询时间要求很高。一定的数据分析能力，异构数据处理能力开始成为标配能力。

以上两个痛点问题的产生原因主要是：在 MySQL 生态里，各开源发行版只支持传统的单线程查询处理模式，即单条 SQL 处理涉及到的解析、优化和执行等阶段，都是在一个线程（称为用户线程）中完成的，这种技术实现模式无法充分利用现代多核 CPU 与大内存的硬件资源，导致一定程度的资源浪费。
因此，需要简化复杂分析的使用并且提升分析性能，基于同一份数据，调动多核服务于大查询（查询内并行），无疑是查询加速和降本增效的重要措施。这就是**并行查询（Parallel Query，PQ）**，构建 HTAP 产品形态的技术手段之一 。

功能优势

- **零成本性能提升**：内核能力升级、无需付费支付额外附加成本，将充分调动您的实例 CPU 计算能力，加快语句响应速度，计算性能大幅提升。
- **透明级流程监控**：提供多种监控项对并行查询过程进行监控，保证每个异常状态均可实时掌握，保护集群稳定运行。
- **常用语句全面支持**：兼容大部分常用 SQL 语句，支撑多种业务场景，保证业务流畅加速。
- **灵活参数设置**：提供多种参数帮助您控制并行查询的启停条件，让查询更智能，灵活适配您的业务场景，无需改造即可使用该能力。

版权所有：腾讯云计算（北京）有限责任公司

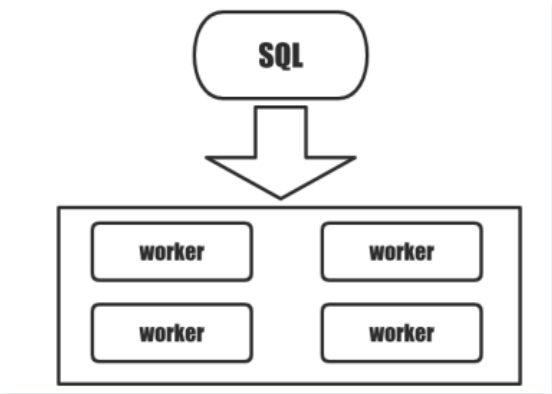
实现原理

最近更新时间：2024-10-11 16:18:31

本文介绍并行查询功能的实现原理。

并行原理

TDSQL-C MySQL 版推出并行查询（Parallel Query）能力，将完整数据分区下推到不同的线程上，利用多个线程进行并行计算，并将结果汇总到用户线程上，并返回给用户，提升查询效率。



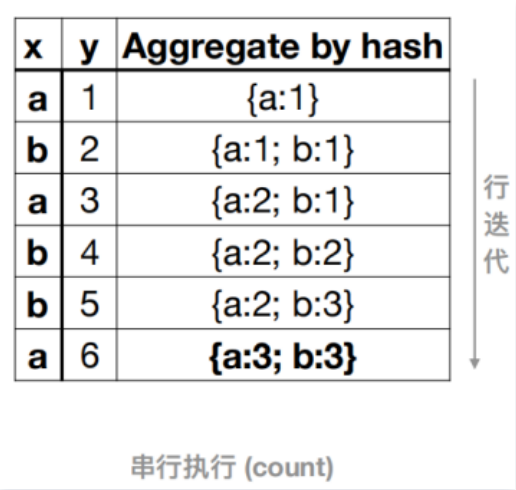
下面用一个简单的例子介绍并行查询的基本原理。

> **说明**
> MySQL/InnoDB 物理存储为段页式，分区单位为页，这里用行演示分区的概念。

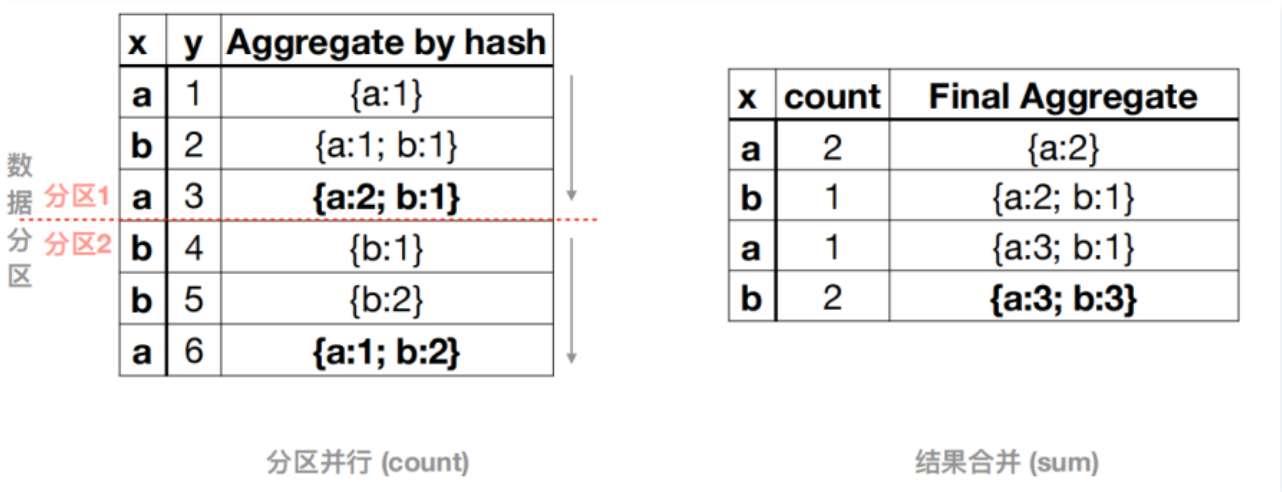
给定一个表 t 和如下分组聚合查询。

```
select x, count(*) from t group by x;
```

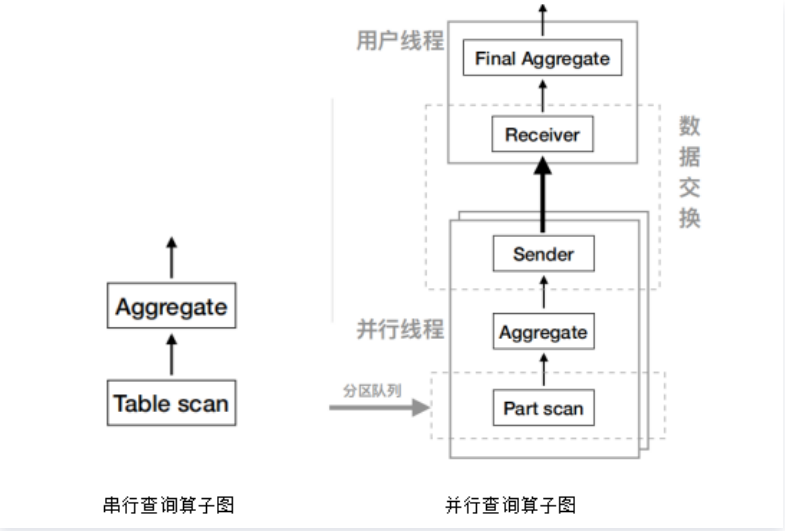
哈希聚合算法（迭代求值）过程如下表所示。算法迭代每一行，更新分组聚合状态。当所有数据行迭代结束时，就得到了分组聚合结果。假设聚合状态更新操作是恒定的，那么算法时间复杂度是 O(n) 的。



如果用 k 个线程来加速这个查询（这里 k = 2），那么最好先将数据表划分成 k * p 个分区（这里 p = 1）。这样，每个线程可以处理 p 个分区，产生一个局部结果。显然，这些局部结果并不是最终结果，还需要进行合并处理。合并操作需要放在同一个用户线程里执行，才能获得正确的最终结果。



由上述示例我们可以将串行查询模式与并行查询模式的运算用算子图表示为：

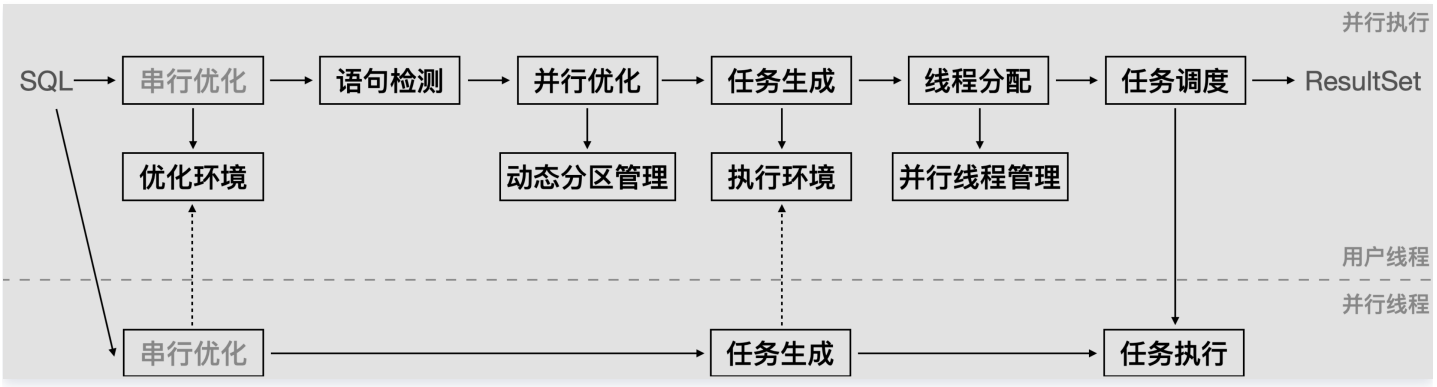


通过算子图可看到，在并行查询中数据会根据并行度，被拆分为数个不重叠的分区，这一过程被称为数据分区。
在数据分区完成后，原始计划中的特殊运算也会进行拆分，此过程为任务拆分。全部拆分完成后数据会被多个工作线程（并行线程）扫描并执行，工作线程会将各部分结果通过数据交换算子聚集到用户线程，这一过程依赖于数据交换。
数据交换完成后，由用户线程完成聚集操作，汇总结果，将完整结果输出。其中，用户线程负责数据分区和任务分拆，同时充当协调者角色（也称为协调线程），协调多个工作线程并行地执行子任务。用户线程还负责合并最终结果，返回给用户。工作线程执行并行子任务，并通过数据通道交换中间结果。
那么，可以总结并行查询的几个核心要素为：

- **数据分区**：将原始表数据划分成不重叠的分区，并支持按分区读数据。
- **任务拆分**：将原始计划中的特殊运算拆分成“局部-整体”两段运算，除此之外，还要插入数据交换算子，支持跨线程数据传递。
- **数据交换**：支持在不同线程间传递数据。

并行流程

TDSQL-C MySQL 版基于上述原理，实现了整套并行查询计划，将串行处理流程扩展为了并行处理流程，如下图所示。



一条 SQL 语句，在 MySQL 传统的串行流程中为：一条 SQL 语句先进行串行优化，输出串行执行计划，之后执行迭代式模型输出结果，整个过程效率不高。为实现并行查询能力，TDSQL-C MySQL 版全新设计了整套 SQL 语句处理流程：

1. 基于并行查询原理我们可知，整个过程被划分为用户线程和工作线程（并行线程）两部分，所有的流程均在这两类线程中进行。在一条 SQL 语句开始执行后，用户线程会在优化器中根据所设定的参数值对这条语句进行分析，生成对应的执行计划。SQL 优化环境可以简单理解为一个高度抽象的确定性计算模块，其输入是 SQL 和优化环境，输出是执行计划。对于相同的输入，就会有相同的优化路径，产生相同的输出，保证结果的准确性。
2. 生成执行计划后，会进入语句检测阶段，此时计算层会检测该语句是否符合执行并行查询的标准，语句层面检测包括动态查询，数据隔离级别是 RC 或 RR，数据处理量是否足够多代价，执行计划层面检测迭代算子和函数是否可以并行，若不符合要求，该语句会回到串行执行；若符合标准则执行并行查询，进入并行优化阶段。具体支持的语句请参见 并行查询支持的语句场景 。
3. 在并行优化阶段，计算层会根据需求去选择对哪个表进行数据拆分，并对聚合或排序等操作进行任务拆分，方便各 worker 并行执行。在优化阶段，由于涉及数据划分，为保证划分到每个线程的数据足够均衡，TDSQL-C MySQL 版引入动态分区管理能力，保证一个线程执行多个任务，最大程度避免数据倾斜。
4. 在完成上述一系列流程后，计算层将生成并行查询任务，通过任务副本的方式下推至工作线程，并根据参数设置的值对工作线程数量进行分配与限制，工作线程此时开始执行并行查询操作，各线程并行执行，得到结果后将数据上推至用户线程进行聚合处理，并返回给用户，至此，一条语句完整的并行查询流程执行完毕。

整体流程可以总结为，用户线程收到 SQL 后，经解析、校验和优化等常规步骤，产生串行执行计划，同时搜集优化过程依赖的各种信息（优化环境）。然后，分析串行执行计划（语句检测，实际上是检测算子树、执行环境和优化代价），决定是否启动并行优化。并行优化时将串行算子树划分成粗粒度任务，选定并行表（动态分区）和任务间数据交换算法，并构造任务依赖图。此时，用户线程准备就绪，从可用线程管理器里申请到足够的工作线程，就可以开始调度执行，工作线程完成后，根据并行查询原理，数据通过数据交换至用户线程进行聚合处理，返回给用户完整结果。

在以上流程中，TDSQL-C MySQL 版设置了多种参数方便用户对并行查询能力进行调整，控制语句的执行代价与并行查询造成的资源负载等，并辅以多种监控指标，实时监控与并行查询有关的多项信息，详情请参见 开启或关闭并行查询 。

支持的语句场景和受限场景

最近更新时间：2025-04-11 17:41:54

本文介绍并行查询能力支持的语句场景和受限场景。

兼容语句场景

TDSQL-C MySQL 版已经实现了具备如下特征的 SQL 语句的并行查询处理，并在逐渐完善更多的功能场景。

功能场景	支持的 TXSQL 引擎内核版本	说明
单表扫描	TXSQL 8.0 3.1.15.001及以上 	支持全表扫描、索引扫描、索引范围扫描、索引 REF 查询等扫描类型的正序、逆序扫描。
多表连接		支持 Nested Loop Join 算法以及 Semi Join、Anti Join、Outer Join 等连接类型。
数据类型		支持带多种数据类型的查询，包括整型数据、字符型数据、浮点型数据、时间类型数据、以及（有运行时大小限制的）溢出类型数据。
普通运算符和函数		普通运算符和函数原则上不限。
聚合函数		支持 COUNT/SUM/AVG/MIN/MAX/STD/VARIANCE/VAR_SAMP/STDDEV_SAMP。
UNION/UNION ALL		支持 UNION/UNION ALL 查询。
EXPLAIN		支持 traditional（默认格式）、json 和 tree 三种 EXPLAIN 格式。
PS 模式并行执行		支持 PS 模式并行执行，PREPARE STATEMENT 让语句仅进行一次语法分析而可以被多次执行。经过 Prepare 预处理的语句只需要修改变量值即可使用，带来性能上的提升。
Hash join(in memory) 并行执行		支持 Hash join(in memory) 并行执行。
子查询 derived table 单独并行执行		支持子查询 derived table 单独并行执行，独立于主查询执行。
ROLL UP 并行执行		支持 ROLL UP 并行执行。
EXPLAIN ANALYZE 并行执行		支持 EXPLAIN ANALYZE 并行执行。
分区表作为并行查询并行表		支持分区表作为并行查询并行表。
全局聚合优化		支持全局聚合优化。
having 条件下推并行		支持 having 条件下推并行。

对应语句的性能提升

SF=100，Dop=16设置下，加速比如下。

- SUM/AVG/COUNT 等聚合函数。

```
SELECT l_returnflag, l_linestatus,
Sum(l_quantity) AS sum_qty,
Sum(l_extendedprice) AS sum_base_price,
Sum(l_extendedprice * (1 - l_discount)) AS sum_disc_price,
Sum(l_extendedprice * (1 - l_discount) * (1 + l_tax)) AS sum_charge,
Avg(l_quantity) AS avg_qty,
```

```
Avg(l_extendedprice) AS avg_price,
Avg(l_discount) AS avg_disc,
Count(*) AS count_order FROM
lineitem WHERE l_shipdate <= date '1998-12-01' - INTERVAL '93' day
GROUP BY l_returnflag, l_linestatus ORDER BY l_returnflag, l_linestatus ;
```

未开启并行查询执行时间为1376.96秒，开启并行查询执行时间为107.25秒，加速倍数可以达到12.84倍。

- order by/group by 分组/排序语句。

```
SELECT l_returnflag, l_linestatus,
Sum(l_quantity) AS sum_qty,
Sum(l_extendedprice) AS sum_base_price,
Sum(l_extendedprice * (1 - l_discount)) AS sum_disc_price,
Sum(l_extendedprice * (1 - l_discount) * (1 + l_tax)) AS sum_charge,
Avg(l_quantity) AS avg_qty,
Avg(l_extendedprice) AS avg_price,
Avg(l_discount) AS avg_disc,
Count(*) AS count_order FROM
lineitem WHERE l_shipdate <= date '1998-12-01' - INTERVAL '93' day
GROUP BY l_returnflag, l_linestatus ORDER BY l_returnflag, l_linestatus ;
```

未开启并行查询执行时间为1376.96秒，开启并行查询执行时间为107.25秒，加速倍数可以达到12.84倍。

- Join/Between/In 语句

```
select
sum(l_extendedprice* (1 - l_discount)) as revenue
from
lineitem,
part
where
(
p_partkey = l_partkey
and p_brand = 'Brand#12'
and p_container in ('SM CASE', 'SM BOX', 'SM PACK', 'SM PKG')
and l_quantity >= 6
and l_quantity <= 6 + 10
and p_size between 1 and 5
and l_shipmode in ('AIR', 'AIR REG')
and l_shipinstruct = 'DELIVER IN PERSON'
)
or
(
p_partkey = l_partkey
and p_brand = 'Brand#13'
and p_container in ('MED BAG', 'MED BOX', 'MED PKG', 'MED PACK')
and l_quantity >= 10 and l_quantity <= 10 + 10
and p_size between 1 and 10
and l_shipmode in ('AIR', 'AIR REG')
and l_shipinstruct = 'DELIVER IN PERSON'
)
or
(
p_partkey = l_partkey
and p_brand = 'Brand#24'
and p_container in ('LG CASE', 'LG BOX', 'LG PACK', 'LG PKG')
and l_quantity >= 21
and l_quantity <= 21 + 10
and p_size between 1 and 15
and l_shipmode in ('AIR', 'AIR REG')
```

版权所有：腾讯云计算（北京）有限责任公司

```
and l_shipinstruct = 'DELIVER IN PERSON'
);
```

未开启并行时查询时间为20.55秒，开启并行查询后加速时间为1.87秒，加速倍数可以达到11倍。

受限场景

TDSQL-C MySQL 版并行查询能力不支持的场景如下。

限制项	限制说明
语句兼容性限制	非查询语句不支持并行查询，包括 INSERT ... SELECT 和 REPLACE ... SELECT。
	stored program 中的查询语句无法并行。
	串行化隔离级别事务内的查询语句无法并行。
	加锁读语句无法并行，如 select for update/share lock。
	CTE 无法并行。
表/索引兼容性限制	查询表为系统表/临时表/非 Innodb 表时无法并行。
	空间索引无法并行。
	全文索引无法并行。
	扫描方式为 index_merge 的表无法并行。
表达式/Field 兼容性限制	包含 Generated Column、BIT 和 GEOMETRY 字段的 SQL 查询无法并行。 **说明：** 包含 BLOB、TEXT、JSON 字段的 SQL 查询在 TXSQL 8.0引擎内核版本大于等于3.1.14时支持并行，前提条件是将参数 txsql_parallel_lob_enabled 的运行值设置为 ON，操作方法参考 设置实例参数 。
	BIT_AND、BIT_OR、BIT_XOR 类型的聚合函数无法并行。
	aggregation(distinct)，如 SUM(DISTINCT)、COUNT(DISTINCT) 等聚合函数无法并行。
	GIS 相关函数（如 SP_WITHIN_FUNC、ST_DISTANCE 等）无法并行。
	用户自定义函数无法并行。
	json 相关的函数无法并行（如 json_length，json_type，JSON_ARRAYAGG 等）。
	XML 相关函数无法并行（xml_str）。
	用户锁相关的函数无法并行（is_free_lock，is_used_lock，release_lock， release_all_locks，get_lock）。
	sleep 函数、random 函数、GROUP_CONCAT 函数、set_user_var 函数、weight_string 函数无法并行。
	窗口函数无法并行。

除了通过 并行查询兼容语句场景 可以查询到语句是否被执行并行查询外，您还可以通过查看并行查询执行计划与查看线程工作状态查询，详情参见 查看并行查询 。

设置并行查询
开启或关闭并行查询

最近更新时间：2025-03-18 17:38:12

TDSQL-C MySQL 版支持并行查询能力，您可通过控制台或命令行调整相关参数，为实例开启或关闭并行查询功能。

前提条件

- 数据库版本：TDSQL-C MySQL 版8.0内核版本3.1.8及以上。
- 使用 新增支持参数能力 需内核版本 TXSQL 8.0 3.1.15及以上。

参数说明

> **说明：**
> 读写实例与只读实例均支持开启并行查询功能，但实例 CPU 核数需大于等于4。

您可通过控制台或命令行调整参数 txsql_max_parallel_worker_threads 和 txsql_parallel_degree 不为0，来开启当前实例的并行查询功能。参数的相关信息和具体设置建议如下。

参数信息

参数	类型	作用域	默认值	取值范围	说明
txsql_max_parallel_worker_threads	Integer	Global	0	0 – 1024	实例节点可用于并行查询的线程资源总数，设置为0则无并行线程可用，视为关闭并行查询功能。
txsql_parallel_degree	Integer	Global Session	4	0 – 1024	单条语句并行查询时可用的最大线程数（默认并行度）。设置为0时视为关闭并行查询功能。

设置建议

- 并行度规格限制：txsql_parallel_degree 参数的数值表明单条语句并行查询使用的最大线程数，即并行查询的默认并行度，建议并行度不要超过实例 CPU 核数的二分之一。为保证稳定性，CPU 核数小于4的小规格集群将禁用并行查询功能，您将无法在控制台或使用命令行调整并行查询相关参数。
- SQL 语句在执行并行查询时将默认使用 txsql_parallel_degree 所设置的并行度，但用户可通过 hint 语句调整单条 SQL 语句的并行查询并行度，详细说明请参见 hint 语句控制 。
- txsql_max_parallel_worker_threads 参数的值表明并行查询中实例可用于并行查询的线程数，txsql_max_parallel_worker_threads / txsql_parallel_degree 的值表明最多同时有多少条 SQL 语句可以执行并行查询。
- txsql_max_parallel_worker_threads 与 txsql_parallel_degree 共同控制并行查询功能的开启与关闭，当任意一个参数设置为0时，表示关闭并行查询功能。

TDSQL-C MySQL 版也提供了多种参数对并行查询的执行条件进行设置，方便您对业务进行个性化适配，保证业务稳定运行。设置后，TDSQL-C MySQL 版将会对语句的执行代价，表的行数，单条语句执行并行计划时所使用的内存等条件进行判断，确认每一条 SQL 语句是否允许执行并行查询。相关参数说明如下：

参数	类型	作用域	默认值	取值范围	说明
txsql_parallel_cost_threshold	Numeric	Global Session	50000	0 – 9223372036854775808	并行执行代价阈值，只有执行代价高于阈值的语句才会进行并行查询。
txsql_parallel_table_record_threshold	Integer	Global Session	5000	0 – 9223372036854775807	并行表行数阈值，只有行数高于阈值的表才能被选为并行表。
txsql_optimizer_context_max_mem_size	Integer	Global	8388608	0 – 9223372036854775807	单条语句可申请的并行查询计划环境最大内存限制。
txsql_parallel_execution_max_lob_size	Integer	Global Session	65536	128 – 9223372036854775807	单个溢出字段的内存限制。

> **注意：**
> - 并行查询等所有参数均无需重启实例，设置后即时生效。

- 标有 session 作用域的参数表明该参数支持对单条语句进行设置。

通过控制台开启或关闭并行查询

您可通过 TDSQL-C MySQL 版控制台进入集群参数设置页面通过设置相关参数开启或关闭功能。

- 设置 txsql_max_parallel_worker_threads 和 txsql_parallel_degree 不为0表示开启并行查询能力。
- 设置 txsql_max_parallel_worker_threads 和 txsql_parallel_degree 任意一个为0表示关闭并行查询能力。

也可设置相关执行条件参数。控制台参数设置页面如下图所示。详细操作方法请参考 设置实例参数 。



通过 hint 语句对单条 SQL 语句指定并行执行方式

TDSQL-C MySQL 版支持对单条 SQL 语句进行指定并行执行方式。对单条 SQL 语句进行设置时，将采用 hint 语句的方式进行操作，详细方法请参照 hint 语句控制 。

附录：新增参数说明

参数	变量类型	作用域	默认值	取值范围	说明
txsql_parallel_partition_verbose	Boolean	Global Session	OFF	ON/OFF	启用时 EXPLAIN ANALYZE 打印更详细的分区信息。
txsql_parallel_limit_enabled	Boolean	Global Session	ON	ON/OFF	启用时支持带 LIMIT 限定的有序数据流（无需额外排序）并行。
txsql_parallel_rollup_pushdown_threshold	Integer	Global Session	4	0 – 100	ROLL UP 下推算法的启用阈值。当 GROUP BY 字段数量大于此阈值时并行计算。
txsql_parallel_force_scan_enabled	Boolean	Global Session	OFF	ON/OFF	全表扫描或全索引扫描查询并行开关。启用时这两类查询才能并行。
txsql_parallel_force_range_enabled	Boolean	Global Session	ON	ON/OFF	索引范围扫描（RANGE/REF）并行开关。启用时这两类查询才能并行。
txsql_parallel_subselect_cost_threshold	Numeric	Global Session	50000	0 – DBL_MAX	并行执行代价阈值。只有估计执行代价高于阈值的子查询和 derived table 才能并行执行。
txsql_parallel_partition_threshold	Integer	Global Session	10000	0 – ULONG_MAX	Nested Loop Join 并行内表 NDV 阈值。只有连接字段唯一值数量高于该阈值的表才能作为并行表。
txsql_parallel_force_probe_enabled	Boolean	Global Session	OFF	ON/OFF	Hash Join 的 probe 表并行阈值。当 hash table 大于 join buffer size 时考虑 probe 表并行。
txsql_parallel_derived_enabled	Boolean	Global Session	ON	ON/OFF	启用时支持 derived table 单独并行。

相关文档

- 查看并行查询
- 并行查询指标

hint 语句控制

最近更新时间：2024-10-10 15:19:47

TDSQL-C MySQL 版支持通过调整相关参数开启或关闭并行查询功能，通过控制台可实现对整个 SQL 语句开启或关闭并行查询能力、设置执行条件参数，也支持使用 hint 语句对单条 SQL 语句进行指定并行执行方式。

> **说明**
> hint 语句可以指定 SQL 语句是否执行，并对指定 SQL 语句可以应用 session 级参数。hint 语句同时支持查询指定的并行表。

hint 语句使用范例

功能	命令行	说明
开启并行查询	`SELECT /*+PARALLEL(x)*/ ... FROM ...;`	x 需大于0，x 表示该条 SQL 语句所使用的并行查询并行度。
关闭并行查询	`SELECT /*+PARALLEL(x)*/ ... FROM ...;`	x 设置为0，表示关闭并行查询能力。

指定并行表	可通过以下两种方式指定允许哪些表执行或不执行并行查询计划：
	• 通过 PARALLEL 可指定表执行并行查询计划
	`SELECT /*+PARALLEL(t)*/ ... FROM ...;`
	• 通过 NO_PARALLEL 可以指定表禁止执行并行查询计划
	`SELECT /*+NO_PARALLEL(t)*/ ... FROM ...;`
		t 为表的名称。

同时指定并行表与并行查询并行度	`SELECT /*+PARALLEL(t x)*/ * ... FROM ...;`	x 需大于0，x 表示该条 SQL 语句所使用的并行查询并行度，t 为表的名称。
通过 hint 语句设置 session 级参数，仅对指定 SQL 语句生效	`SELECT /*+SET_VAR(var=n)*/ * ... FROM ...;`	var 为支持 session 作用域的并行查询参数。

hint 语句使用场景示例

场景一：
`select /*+PARALLEL()*/ * FROM t1, t2;`
强制并行度为 txsql_parallel_degree 所设置的数值（默认并行度）执行并行查询，当语句不符合并行查询执行条件时，将回退为串行查询。
场景二：
`select /*+PARALLEL(4)*/ * FROM t1, t2;`
无论系统默认并行度数值为多少，强制该条语句使用并行度为4执行并行查询，设置该条语句的 txsql_parallel_degree = 4 ，当语句不符合并行查询执行条件时，将回退为串行查询。
场景三：
`select /*+PARALLEL(t1)*/ * FROM t1, t2;`
选择 t1 表执行并行查询，并行度为系统默认并行度，当 t1 表小于 txsql_parallel_table_record_threshold 所设置的值时，将回退为串行查询。
场景四：
`select /*+PARALLEL(t1 8)*/ * FROM t1, t2;`
选择 t1 表执行并行查询，并行度为8，当 t1 表小于 txsql_parallel_table_record_threshold 所设置的值时，将回退为串行查询。
场景五：
`select /*+NO_PARALLEL(t1)*/ * FROM t1, t2;`
选择 t1 表禁止执行并行查询，当 t1 表大于 txsql_parallel_table_record_threshold 所设置的值时，将回退为串行查询。
场景六：
`select /*+SET_VAR(txsql_parallel_degree=8)*/ * FROM t1, t2;`
无论系统默认并行度数值为多少，强制该条语句使用并行度为8执行并行查询，设置该条语句的 txsql_parallel_degree = 8。
场景七：

```
select /*+SET_VAR(txsql_parallel_cost_threshold=1000)*/ * FROM t1, t2
```

设置该条语句的 txsql_parallel_cost_threshold=1000，当该条语句的执行代价大于1000时，即可使用并行查询。

场景八：

```
select /*+SET_VAR(txsql_optimizer_context_max_mem_size=500000)*/ * FROM t1, t2
```

设置单条语句的 txsql_optimizer_context_max_mem_size=500000，该条语句可申请的并行查询计划环境最大内存限制调整为500000。

相关文档

- 开启或关闭并行查询
- 查看并行查询
- 并行查询指标

查看并行查询

最近更新时间：2024-10-23 11:46:41

TDSQL-C MySQL 版支持查看并行查询的执行计划，以及查看线程中哪些线程在执行并行查询计划。您可清晰了解到并行查询是如何在数据库中稳定生效，也可在并行查询执行过程中遇到问题时，帮助快速定位问题。
本文为您介绍查看并行查询的两种常用方法。

方法一：使用 EXPLAIN 语句

创建表示例：

```
CREATE TABLE lineitem (
  L_ORDERKEY INTEGER NOT NULL,
  L_PARTKEY INTEGER NOT NULL,
  L_SUPPKEY INTEGER NOT NULL,
  L_LINENUMBER INTEGER NOT NULL,
  L_QUANTITY DECIMAL(15, 2) NOT NULL,
  L_EXTENDEDPRICE DECIMAL (15, 2) NOT NULL,
  L_DISCOUNT DECIMAL(15, 2) NOT NULL,
  L_TAX DECIMAL(15, 2) NOT NULL,
  L_RETURNFLAG CHAR(1) NOT NULL,
  L_LINESTATUS CHAR(1) NOT NULL,
  L_SHIPDATE DATE NOT NULL,
  L_COMMITDATE DATE NOT NULL,
  L_RECEIPTDATE DATE NOT NULL,
  L_SHIPINSTRUCT CHAR(25) NOT NULL,
  L_SHIPMODE CHAR(10) NOT NULL,
  L_COMMENT VARCHAR(44) NOT NULL
  );
```

插入的数据集来自 TPC-H。
示例 SQL 语句：

```
SELECT l_returnflag, l_linestatus, sum(l_quantity) as sum_qty
FROM lineitem
WHERE l_shipdate <= '1998-09-02'
GROUP BY l_returnflag, l_linestatus
ORDER BY l_returnflag, l_linestatus;
```

本示例为 TPC-H Q1 的简化形式，是典型的报表运算。
执行计划打印语句（EXPLAIN）：

```
EXPLAIN  SELECT l_returnflag, l_linestatus, sum(l_quantity) as sum_qty
FROM lineitem
WHERE l_shipdate <= '1998-09-02'
GROUP BY l_returnflag, l_linestatus
ORDER BY l_returnflag, l_linestatus;
```

查询结果：

```
MySQL [tpch100g]> explain SELECT l_returnflag, l_linestatus, sum(l_quantity) as sum_qty FROM lineitem
WHERE l_shipdate <= '1998-09-02' GROUP BY l_returnflag, l_linestatus ORDER BY l_returnflag, l_linestatus;
+----+-------------+-------------+------------+------+---------------+------+---------+------+-----------
+----------+-----------------------------------------------------------+
| id | select_type | table       | partitions | type | possible_keys | key  | key_len | ref  | rows      |
filtered | Extra                                                     |
```

```
+----+-------------+-------------+------------+------+---------------+------+---------+------+-----------
+----------+-----------------------------------------------------------------+
|  1 | SIMPLE      | lineitem    | NULL       | ALL  | i_l_shipdate  | NULL | NULL    | NULL | 593184480 |
50.00 | Parallel scan (4 workers); Using where; Using temporary   |
|  1 | SIMPLE      | <sender1>   | NULL       | ALL  | NULL          | NULL | NULL    | NULL |         0 |
0.00 | Send to (<receiver1>)                                      |
|  1 | SIMPLE      | <receiver1> | NULL       | ALL  | NULL          | NULL | NULL    | NULL |         0 |
0.00 | Receive from (<sender1>); Using temporary; Using filesort |
+----+-------------+-------------+------------+------+---------------+------+---------+------+-----------
+----------+-----------------------------------------------------------------+
3 rows in set, 1 warning (0.00 sec)
```

树状执行计划打印语句（EXPLAIN format=tree）：

```
EXPLAIN format=tree SELECT l_returnflag, l_linestatus, sum(l_quantity) as sum_qty
FROM lineitem
WHERE l_shipdate <= '1998-09-02'
GROUP BY l_returnflag, l_linestatus
ORDER BY l_returnflag, l_linestatus;
```

查询结果:

```
MySQL [tpch100g]> explain format=tree SELECT l_returnflag, l_linestatus, sum(l_quantity) as sum_qty FROM
lineitem WHERE l_shipdate <= '1998-09-02' GROUP BY l_returnflag, l_linestatus ORDER BY l_returnflag,
l_linestatus\G
*************************** 1. row ***************************
EXPLAIN: -> Sort: lineitem.L_RETURNFLAG, lineitem.L_LINESTATUS
    -> Table scan on <temporary>
        -> Final Aggregate using temporary table
            -> PX Receiver (slice: 0; workers: 1)
                -> PX Sender (slice: 1; workers: 4)
                    -> Table scan on <temporary>
                        -> Aggregate using temporary table
                            -> Filter: (lineitem.L_SHIPDATE <= DATE'1998-09-02')  (cost=65449341.10
rows=296592240)
                                -> Parallel table scan on lineitem  (cost=65449341.10 rows=593184480)

1 row in set (0.00 sec)
```

由上述结果可以看出：

- **并行查询计划将语句分布给了4个工作线程进行运算。**
- **将聚合运算拆分为了上下段，用户线程和并行线程分别执行。**
- **对 lineitem 表采用了并行扫描算子。**
- **实例中树状执行计划打印（EXPLAIN format=tree query）相较于传统执行计划打印（EXPLAIN）效果更好。**

方法二：线程列表查看

show processlist 命令的输出结果显示了有哪些线程在运行，不仅可以查看当前所有的连接数，还可以查看当前的连接状态帮助识别出有问题的查询语句等。基于 show processlist 命令，TDSQL-C MySQL 版自研了 show parallel processlist 语句，帮助您过滤线程中非并行查询的线程，使用该命令行后，将只展示与并行查询有关的线程。

示例 SQL 语句：

```
SELECT l_returnflag, l_linestatus, sum(l_quantity) as sum_qty
FROM lineitem
WHERE l_shipdate <= '1998-09-02'
GROUP BY l_returnflag, l_linestatus
ORDER BY l_returnflag, l_linestatus;
```

本示例为 TPC-H Q1 的简化形式，是典型的报表运算。

show processlist 查询结果：

```
mysql> show processlist;
+--------+-------------+-----------------+-----------+---------+-------+------------+----------------------
-----------------------------------------------------------------------------------+
| Id     | User        | Host            | db        | Command | Time  | State      | Info
|
+--------+-------------+-----------------+-----------+---------+-------+------------+----------------------
-----------------------------------------------------------------------------------+
|      7 | tencentroot | 127.0.0.1:49238 | NULL      | Sleep   |     0 |            | NULL
|
|     11 | tencentroot | 127.0.0.1:49262 | NULL      | Sleep   |     0 |            | NULL
|
|     13 | tencentroot | 127.0.0.1:49288 | NULL      | Sleep   |     1 |            | NULL
|
| 237062 | tencentroot | localhost       | tpch100g  | Query   |    24 | Scheduling | SELECT l_returnflag,
l_linestatus, sum(l_quantity) as sum_qty FROM lineitem WHERE l_shipdate <= '199 |
| 237107 | tencentroot | localhost       | NULL      | Query   |     0 | init       | show processlist
|
+--------+-------------+-----------------+-----------+---------+-------+------------+----------------------
-----------------------------------------------------------------------------------+
6 rows in set (0.00 sec)
```

show parallel processlist 查询结果：

```
mysql> show parallel processlist;
+--------+-------------+-----------+----------+---------+------+-------------+----------------------------
-------------------------------------------------------------------------+
| Id     | User        | Host      | db       | Command | Time | State       | Info
|
+--------+-------------+-----------+----------+---------+------+-------------+----------------------------
-------------------------------------------------------------------------+
| 237062 | tencentroot | localhost | tpch100g | Query   |   18 | Scheduling  | SELECT l_returnflag,
l_linestatus, sum(l_quantity) as sum_qty FROM lineitem WHERE l_shipdate <= '199 |
| 237110 |             |           |          | Task    |   18 | Task runing | connection 237062, worker
0, task 1                                                                       |
| 237111 |             |           |          | Task    |   18 | Task runing | connection 237062, worker
1, task 1                                                                       |
| 237112 |             |           |          | Task    |   18 | Task runing | connection 237062, worker
2, task 1                                                                       |
| 237113 |             |           |          | Task    |   18 | Task runing | connection 237062, worker
3, task 1                                                                       |
+--------+-------------+-----------+----------+---------+------+-------------+----------------------------
-------------------------------------------------------------------------+
5 rows in set (0.00 sec)
```

由上述结果可以看出：

- 上述查询由并行计划分布给四个 work 线程进行执行：user 仅有一行有显示，表明 ID 237062 为用户线程，将 SQL 语句执行计划下推至下面四个 work 线程中进行，通过 info 列可看到，这四个工作线程均在执行 task1。
- 每个线程均可查询出来，精准进行定位。
- show parallel processlist 相较于 show processlist 可以精准查询到所有进行并行查询的线程，不被其余线程影响。

相关文档

- 开启或关闭并行查询
- hint 语句控制
- 并行查询指标

并行查询指标

最近更新时间：2024-08-02 16:34:01

TDSQL-C MySQL 版设置了多种监控指标帮助您实时监控数据库中有关并行查询的相关操作。本文为您介绍并行查询的相关监控指标。

前提条件

数据库版本：TDSQL-C MySQL 版8.0内核版本3.1.8及以上。

并行查询相关监控指标

监控项（中文）	监控项（英文）	单位	说明	数据聚合方式
当前并行查询线程数	txsql_parallel_threads_currently_used	个	并行查询当前使用的线程数量。	MAX
并行查询错误数	txsql_parallel_stmt_error	个	并行查询报错的语句数量。	SUM
已执行并行查询数	txsql_parallel_stmt_executed	个	已执行的并行查询语句数量。	SUM
回滚串行查询数	txsql_parallel_stmt_fallback	个	并行查询回滚到串行查询的语句数量。	SUM

监控项实践教程

下面列举部分特殊场景问题需对应查看监控项和解决方法。

示例1：

当前并行查询线程数始终等于 txsql_max_parallel_worker_threads 所设置的值。

解决方法：

示例表明线程数始终处于满载状态，建议在 CPU 使用率未达到高负载的情况下，增大参数 txsql_max_parallel_worker_threads 的值。

示例2：

并行查询错误数增加。

解决方法：

查询监控项 CPU 使用率与内存使用率，若负载明显变高，建议降低 txsql_parallel_degree 值为 CPU 核数的1/4。

示例3：

回滚串行查询数增加。

解决方法：

表明当前执行 SQL 不满足执行并行查询条件，建议增大线程资源总数 txsql_max_parallel_worker_threads 或增大单条语句可申请的并行查询计划环境最大内存限制 txsql_optimizer_context_max_mem_size，保证 SQL 语句可以使用并行查询。

版权所有：腾讯云计算（北京）有限责任公司

性能测试报告

最近更新时间：2024-10-10 11:11:26

本文为您展示 TDSQL-C MySQL 版并行查询的性能测试报告。

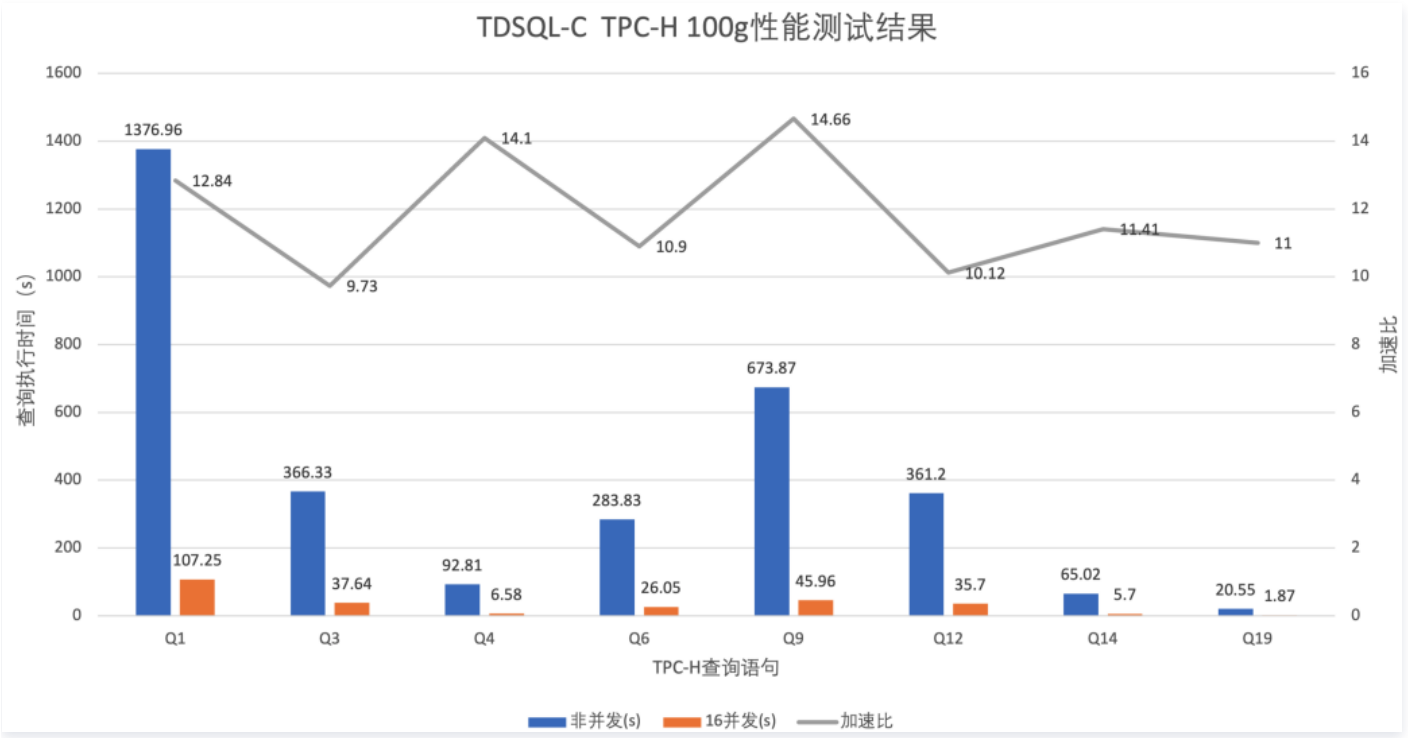
标准性能测试报告

- 数据集：TPC-H
- 测试数据量：100GB
- 并行线程数：16
- Query：Q1、Q3、Q4、Q6、Q9、Q12、Q14、Q19 性能加速比

执行时间表

Query	非并行执行时间（s）	并行查询16线程（s）	加速倍数
Q1	1376.96	197.25	12.84
Q3	366.33	37.64	9.73
Q4	92.81	6.58	14.1
Q6	283.83	26.05	10.9
Q9	673.87	45.96	14.66
Q12	361.2	35.7	10.12
Q14	65.02	5.7	11.41
Q19	20.55	1.87	11.0

执行时间加速比



结论

利用并行查询能力，调动多核计算资源，可大幅缩短大查询响应时间。

列存索引 CSI
简介
功能简介

最近更新时间：2024-05-20 17:44:11

背景

随着越来越多的用户将核心系统上云，用户数据量不断增加，业务逻辑也变得越来越复杂。虽然 TDSQL-C MySQL 版的云原生架构可以较好地处理业务的事务请求，但是对于用户多样化的查询分析需求，基于行存的查询并不能总是满足用户的响应时间要求。一种常见的解决方案是采用数据同步工具，将事务系统的数据复制到独立的分析系统中，并将用户的查询分析路由到分析系统中，然而，这种解决方案存在一些缺点，如数据同步和独立分析集群的额外成本，同步的实时性和一致性等问题。
为了解决这类问题，TDSQL-C MySQL 版推出了 **CSI（Column store Index）列存索引**功能。该功能使用列式数据格式存储、检索和管理数据的技术，可实现更好的查询性能和更高的数据压缩率。

功能优势

相比传统的基于行的存储方式，列存索引使用基于列的数据存储和查询处理，可以实现一个量级的查询性能提升。此外，列存储更有利于数据压缩，相比行存储，可以降低近一个量级的存储空间。

支持版本

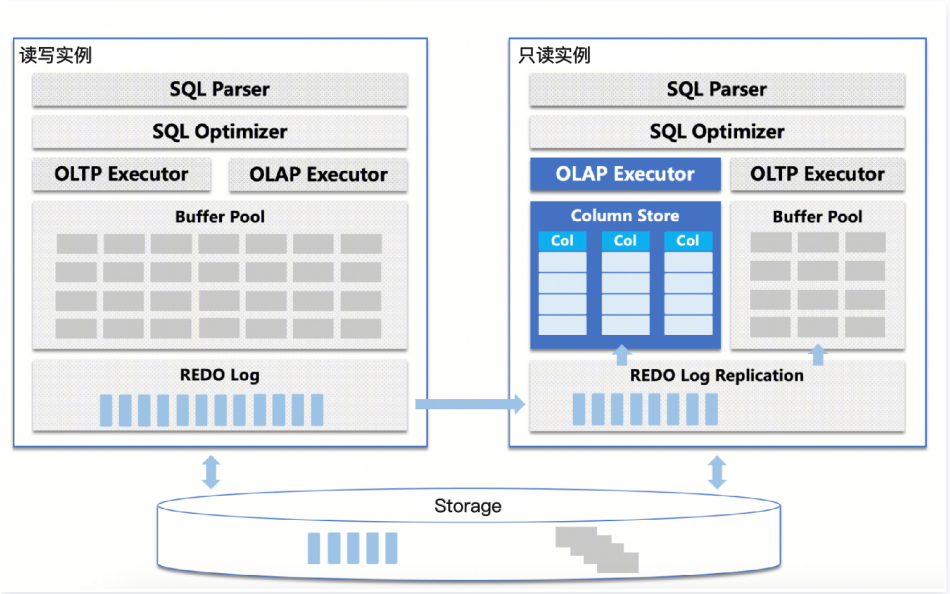
内核版本 TDSQL-C MySQL 版8.0 3.1.14及以上。

> **说明：**
> 针对只读实例而言，符合版本要求的情况下，4核以上的只读实例才可以开启列存索引功能。

应用场景

- 在线报表，数据看板等对在线数据有实时数据分析需求的场景。
- 大数据量分析型查询场景。

技术原理



TDSQL-C MySQL 版的列存索引能力，基于如下3个关键技术点实现：

1. 行列数据的混合存储

   在 TDSQL-C MySQL 版中，数据默认以行式存储。为了更好地支持分析查询，列式存储是更为合适的选择。在统一的架构上，我们采用了为行存表创建列存索引的方式，达到数据混合存储的目的。

2. 行列混合计划的生成与调度执行

   TDSQL-C MySQL 版基于优化器的统计信息、代价模型，让列存索引适配当前行存优化器模型，并参与 CBO（代价优化器）的搜索空间。进而执行计划片断下压，基于片断执行混合调度。

3. 高效计算

   由于数据是以列式单独存储于数据块中的，计算时只读取所需要的列数，从而可以大大减少 IO 层的开销，特别是大宽表的场景。并且，列式数据可以按更紧凑的方式存储在内存中，使用列存索引 CSI 计算时，各个算子尽量按批处理多行的列数据，充分发挥 CPU 的 cache 亲和力，从而可以提高计算效率。

版权所有：腾讯云计算（北京）有限责任公司

支持的语句和受限场景

最近更新时间：2024-05-20 17:44:11

本文为您介绍 CSI（Column store Index）列存索引功能支持的语句和受限场景。

前提条件

内核版本为 TDSQL-C MySQL 版8.0 3.1.14及以上。

> **说明：**
> 针对只读实例而言，符合版本要求的情况下，4核以上的只读实例才可以开启列存索引功能。

支持的语句

TDSQL-C MySQL 版已经实现了具备如下特征的 SQL 语句的计算下压查询处理，并在逐渐完善更多的功能场景。

功能场景	说明
单表扫描	对于单表扫描，支持全表扫描、索引扫描、索引范围扫描等扫描类型的正序、逆序扫描。
多表 join	对于多表 Join，支持 Nested loop Join、HashJoin、Sort Merge Join 等多种 Join 算法。
子查询	对于子查询，支持非关联子查询。
数据类型	对于数据类型，支持带多种数据类型的查询，包括整型数据、字符型数据、浮点型数据、时间型数据等。
运算公式	支持算术表达式运算（+、-、*、%、/、|、&），条件表达式运算（<、<=、>、>=、<>、between/and、In 等），逻辑运算（or、and、not等），一般函数（字符函数、整型函数、时间函数等），聚合函数（count/sum/avg/min/max 等）。
查看下压执行计划	支持 Explain 查看下压执行计划，但只有在 Explain format=tree 中才能显示 COLUMNSTORE 标记。

功能限制

TDSQL-C MySQL 版将持续迭代 CSI 查询的能力，当前无法使用 CSI 查询的场景如下：

创建 CSI 的限制

- 同一张表只能创建一个 CSI 索引，若数据库内核版本为3.1.13及以上，则单个索引最多可包含256个字段；若数据库内核版本为3.1.13以下，则单个索引最多可包含16个字段。

  > **说明：**
  > 只要对应字段包含在 CSI 索引中，均可命中索引，无需满足前缀匹配条件。

- 不支持使用 Generated Column、BLOB、TEXT、JSON、BIT 和 GEOMETRY 来创建 CSI。
- 不支持使用 COPY 算法创建或删除 CSI，或先删除再加 CSI。
- 非 InnoDB 表不支持创建 CSI。

对 CSI 表的 DDL 限制

- 不支持对 CSI 表进行分区操作。

SELECT 使用 CSI 的限制

- 不支持 DML 中的 SELECT，包括 INSERT ... SELECT/REPLACE ... SELECT。
- 不支持 CTE。
- 不支持窗口函数。
- 不支持 with rollup。
- 不支持 UNION。
- 不支持 ReadCommit 外的隔离级别的事务。

版权所有：腾讯云计算（北京）有限责任公司

- 不支持加锁读语句，例如 SELECT FOR UPATE/SHARE LOCK。
- 不支持 Stored Procedure。
- 不支持 MODE_PAD_CHAR_TO_FULL_LENGTH/MODE_INVALID_DATES的SQL_MODE。
- 不支持 GIS 相关函数（如 SP_WITHIN_FUNC、st_distance 等）。
- 不支持用户自定义函数。
- 不支持 XML 相关函数（xml_str）。
- 不支持 lock 相关的函数（is_free_lock、is_used_lock、release_lock、release_all_locks、get_lock）。

注意事项

CSI 查询与普通查询执行结果存在一定的不兼容，包括以下几点。

- **结果集无序**

  当不指定 ORDER 序时，由于列存的组织方式与 BTree 的组织方式不同，返回的结果集可能与 BTR 执行的结果序不一致。实际上这符合 ANSI 的标准，但是用户对于结果集顺序敏感时，建议指定排序规则来获取与串行一致的结果。
- **精度与截断问题**

  使用 CSI 时，由于部分数据类型的存储和计算精度与 MySQL 不完全一致，因此可能会导致浮点部分产生精度差别或者产生截断，导致最终结果有细微的差别。
- **错误或告警提示次数可能会增多**

  对于在串行执行中出现错误/告警提示的查询，当使用 CSI 时采用并行执行的情况下，每个工作线程可能都会提示错误/告警，导致总体错误/告警提示数增多。
- **资源影响**

  CSI 的 buffer pool 会占系统内存空间。
  CSI 的并行执行时，相比串行查询的单线程模式，并行查询会消耗更多的线程、内存和 IO 资源。

版权所有：腾讯云计算（北京）有限责任公司

列存索引使用方式
基本设置

最近更新时间：2025-01-14 14:36:12

本文为您介绍列存索引 CSI 的相关操作。

前提条件

内核版本为 TDSQL-C MySQL 版8.0 3.1.14及以上。

> **说明：**
> 针对只读实例而言，符合版本要求的情况下，4核以上的只读实例才可以开启列存索引功能。

开启或关闭 CSI

1. 在集群列表页，根据实际使用的视图模式进行进入实例详情页。

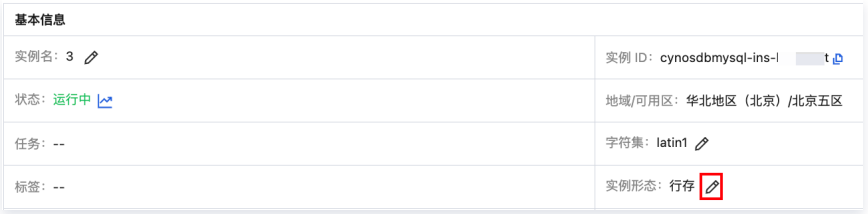
页签视图

1. 登录 TDSQL-C MySQL 版控制台，在左侧集群列表，单击**目标集群**，进入集群管理页。
2. 在集群详情下，找到目标实例，单击**实例 ID** 后的**详情**，进入实例详情页。

列表视图

1. 登录 TDSQL-C MySQL 版控制台，在集群列表，找到需要修改字符集的集群，单击**集群 ID**，进入集群管理页面。
2. 在集群管理页面，选择**实例列表**页，找到要开启或关闭列存索引的只读实例，单击**实例 ID**，进入实例详情页。

2. 在实例详情页的**实例形态**后，单击修改图标。



3. 在弹窗下，选择操作时间，勾选“**在操作过程中，会有秒级别闪断，请确保业务具备重连机制**”，单击**确定**。

操作时间

- **立即执行**：立即执行实例形态的切换。
- **维护时间内**：在您设置的实例维护时间内执行，修改维护时间请参见 修改实例维护时间 。

> **说明：**
> - 实例形态由**行存**调整为**行列混存**，表示**开启**列存索引 CSI。
> - 实例形态**行列混存**调整为**行存**，表示**关闭**列存索引 CSI。

创建 CSI

开启列存索引 CSI 后，建表相关的创建 CSI 命令如下：

1. 创建表时创建 CSI

```
CREATE TABLE table_name (col1, col2,... COLUMNSTORE INDEX [index_name] [(col1, col2,...)]);
```

2. 创建表后创建 CSI

```
CREATE COLUMNSTORE INDEX [index_name] ON table_name[(col1, col2,...)];
ALTER TABLE table_name ADD COLUMNSTORE INDEX [index_name] [(col1, col2,...)];
```

删除 CSI

开启列存索引 CSI 后，删除列存索引 CSI 的命令如下：

```
ALTER TABLE table_name DROP INDEX index_name;
```

重命名 CSI

开启列存索引 CSI 后，重命名列存索引 CSI 的命令如下：

```
ALTER TABLE table_name RENAME index old_index_name to new_index_name;
```

列存索引 CSI HINT 语句

1. 强制执行行存执行/列存执行。
   - 强制执行行存执行

     ```
     SELECT a FROM t IGNORE INDEX (csi);
     ```

   - 强制执行列存执行

     ```
     SELECT a FROM t FORCE INDEX (csi);
     ```

2. 同时使用 HINT 执行并行查询与列存索引。

   ```
   SELECT /*+PARALLEL(2)*/ a FROM t FORCE INDEX (csi);
   ```

创建表和列存索引示例

```
CREATE TABLE t (a int, columnstore index csi (a));
INSERT INTO t VALUES (0), (1), (2);
SHOW CREATE TABLE t;
SHOW INDEX FROM t;
```

执行结果如下：

```
MySQL [test]> CREATE TABLE t (a int, columnstore index csi (a));
Query OK, 0 rows affected (0.01 sec)
MySQL [test]> INSERT INTO t VALUES (0), (1), (2);
Query OK, 3 rows affected (0.01 sec) Records: 3  Duplicates: 0  Warnings: 0
MySQL [test]> SHOW CREATE TABLE t;
+-------+-----------------------------------------------------------------------------------------------------
--------------+
| Table | Create Table
|
+-------+-----------------------------------------------------------------------------------------------------
--------------+
| t     | CREATE TABLE `t` (   `a` int DEFAULT NULL,   COLUMNSTORE KEY `csi` (`a`) ) ENGINE=InnoDB DEFAULT
CHARSET=utf8 |
+-------+-----------------------------------------------------------------------------------------------------
--------------+
```

```
MySQL [test]> SHOW INDEX FROM t;
+-------+------------+----------+--------------+-------------+-----------+-------------+----------+-------
-+------+------------+---------+---------------+---------+------------+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed
| Null | Index_type   | Comment | Index_comment | Visible | Expression |
+-------+------------+----------+--------------+-------------+-----------+-------------+----------+-------
-+------+------------+---------+---------------+---------+------------+
| t     |          1 | csi      |            1 | a           | NULL      |           1 |     NULL |   NULL
| YES   | COLUMNSTORE |          |               | YES     | NULL       |
+-------+------------+----------+--------------+-------------+-----------+-------------+----------+-------
-+------+------------+---------+---------------+---------+------------+
1 row in set (0.00 sec)
```

INDEX HINT 使用

1. 强制语句使用列存索引。

```
SELECT a FROM t FORCE INDEX (csi);
EXPLAIN FORMAT=TREE SELECT a FROM t FORCE INDEX (csi);
```

执行结果：

```
MySQL [test]> SELECT a FROM t FORCE INDEX (csi);
+------+
| a    |
+------+
|    0 |
|    1 |
|    2 |
+------+
3 rows in set (0.00 sec)
MySQL [test]> EXPLAIN FORMAT=TREE SELECT a FROM t FORCE INDEX (csi);
+-------------------------------------------------------------------+
| EXPLAIN                                                           |
+-------------------------------------------------------------------+
| -> COLUMNSTORE Index scan on t using csi  (cost=1.30 rows=3)  |
+-------------------------------------------------------------------+
1 row in set (0.00 sec)
```

2. 强制语句不使用列存索引（行存执行）。

```
SELECT a FROM t IGNORE INDEX (csi);
EXPLAIN FORMAT=TREE SELECT a FROM t IGNORE INDEX (csi);
```

执行结果：

```
MySQL [test]> SELECT a FROM t IGNORE INDEX (csi);
+------+
| a    |
+------+
|    0 |
|    1 |
|    2 |
+------+
3 rows in set (0.00 sec)
MySQL [test]> EXPLAIN FORMAT=TREE SELECT a FROM t IGNORE INDEX (csi);
+----------------------------------------+
| EXPLAIN                                |
```

```
+-----------------------------------------+
| -> Table scan on t  (cost=0.55 rows=3)  |
+-----------------------------------------+
1 row in set (0.00 sec)
```

查看 CSI 索引创建情况

- show create table TABLE

> **说明：**
> 默认不显示 COLUMNSTORE 前缀，需要指定开关 columnstore_display_in_show_create=1，才会显示。

- show index from TABLE
- explain format=tree

> **说明：**
> 开启列 CSI 后，要查看 CSI 索引创建情况，也可通过 explain format=tree 查看执行计划算子是否有 COLUMNSTORE 前缀（有则表示算子采用了列式执行），来得知该算子是否使用列式执行。只有指定 format=tree 时，才显示 COLUMNSTORE 前缀，不指定格式则默认不显示 COLUMNSTORE。

参数配置与监控项

最近更新时间：2025-02-24 20:18:12

本文为您介绍列存索引 CSI 的参数配置和监控项。

前提条件

内核版本为 TDSQL-C MySQL 版8.0 3.1.14及以上。

> **说明：**
> 针对只读实例而言，符合版本要求的情况下，4核以上的只读实例才可以开启列存索引功能。

参数

参数名	类型	默认值	取值范围	说明
columnstore_buffer_pool_size	int	1024	0, INT_MAX	cstore 总内存，CSI 的 buffer pool 大小，单位为 MB。
columnstore_scan_pushdown_rows_threshold	int	100000	0, INT_MAX	若一个表的索引表中的数据行数少于此阈值，则该表的数据不会通过列存访问。此参数为非全局参数，只对开启 CSI 功能的实例生效。

列存索引 CSI 监控项

中文指标名	英文指标名	单位	聚合方式	含义
列存索引执行语句数	Innodb_csi_select_pushdown	个	SUM	已执行的列存查询语句数量。
列存索引回退语句数	Innodb_csi_select_fallback	个	SUM	列存索引回退语句数量。
列存索引同步延时	innodb_csi_lag_seconds	秒	MAX	rw 与 CSI 实例同步延时。
列索引存储使用量	Innodb_csi_disk_usage	GB	MAX	列索引存储使用量。
创建 CSI 失败数	txsql_csi_fail	个	SUM	创建列存索引失败个数。

版权所有：腾讯云计算（北京）有限责任公司

性能测试报告

最近更新时间：2024-10-12 17:11:42

本文为您介绍 TDSQL-C MySQL 版列存索引 CSI 的性能测试报告。

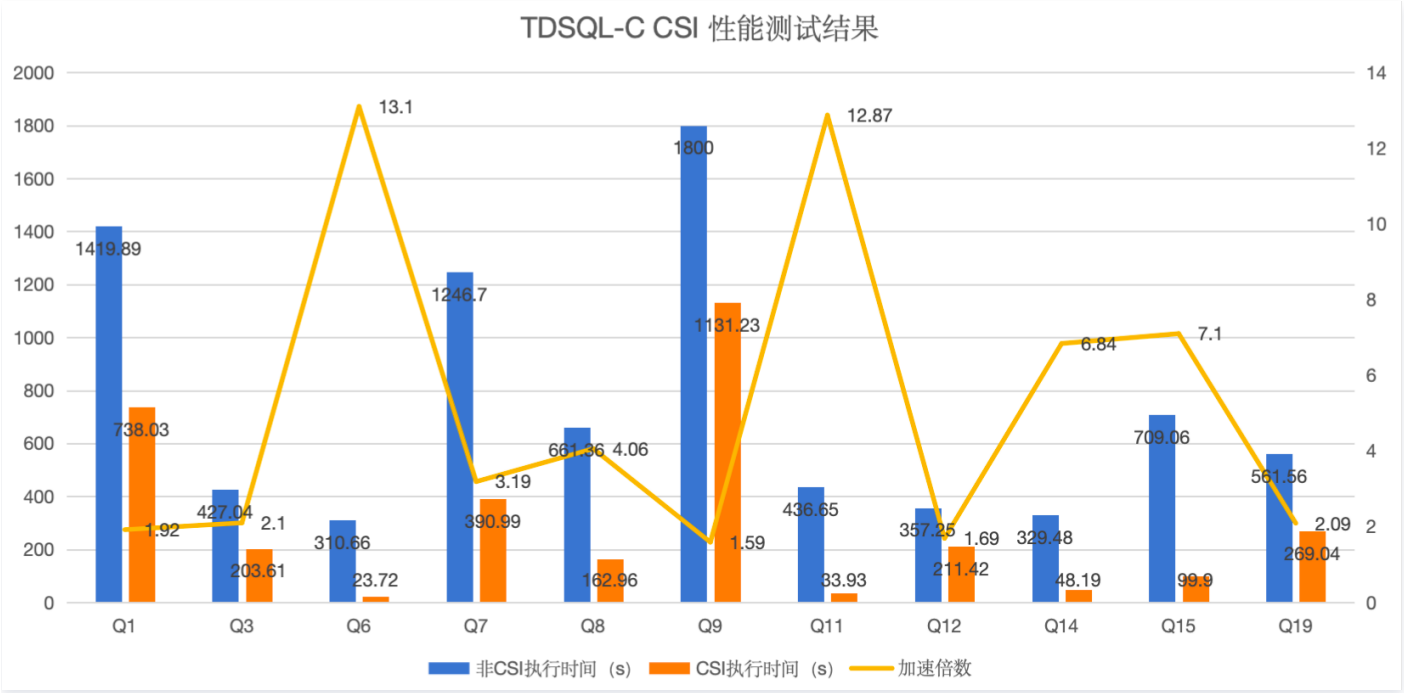
标准性能测试报告

- 数据集：TPC-H。
- 测试数据量：100GB。
- 实例规格：32核256GB。
- Query：Q1、Q3、Q6、Q7、Q8、Q9、Q11、Q12、Q14、Q15、Q19性能加速比。

执行时间表

Query	非 CSI 执行时间（s）	CSI 执行时间（s）	加速倍数
Q1	1419.89	738.03	1.92
Q3	427.04	203.61	2.10
Q6	310.66	23.72	13.10
Q7	1246.7	390.99	3.19
Q8	661.36	162.96	4.06
Q9	1800	1131.23	1.59
Q11	436.65	33.93	12.87
Q12	357.25	211.42	1.69
Q14	329.48	48.19	6.84
Q15	709.06	99.90	7.10
Q19	561.56	269.04	2.09

执行时间加速比



版权所有：腾讯云计算（北京）有限责任公司

结论

使用列存索引 CSI，可实现更好的查询性能，大大缩短查询时间。

分析引擎

版权所有：腾讯云计算（北京）有限责任公司

简介

最近更新时间：2025-07-02 15:03:42

本文为您介绍 TDSQL-C MySQL 版基于 LibraDB 引擎实现的全新功能：分析引擎。

> **说明：**
> - 目前仅支持**只读**的分析引擎实例，所有数据均通过 Zero-ETL 的能力同步而来。
> - TDSQL-C MySQL 版支持两种实例引擎（或者说集群类型）：事务集群与分析集群。事务集群可通过 开启只读分析引擎 配合 TXSQL 读写实例提供 HTAP 能力，分析集群中仅存在**只读分析引擎**实例，并通过 Zero-ETL 能力将配置的数据源数据同步至分析引擎。

背景

TDSQL-C MySQL 版作为云原生数据库产品，在支撑高并发、强一致、企业级数据库特性上做了非常多的优化工作。其基于 TXSQL 引擎支持了高性能的在线事务处理能力。但是，除了利用数据库服务于高 QPS 的在线事务外，很多业务系统还需要对数据进行挖掘，利用数据库进行数据分析，以帮助企业进行更好的业务决策，推动业务的迭代创新，快速适应市场环境的变化。
传统数据库为了支持高性能的在线事务处理能力，并且保证业务查询的稳定性，通常在存储结构上选择了行式存储，在执行模型上选择了火山模型，这使得其无法高效的服务分析类查询。当然，部分业务为了能够支持事务以及分析的混合负载选择了“传统数据库 + 数据仓库”的解决方案，但是这种方案为客户带来了高昂的维护成本，需要客户自己构建数据库到数据仓库的 ETL 工具，并且数据的实时性以及一致性也得不到很好的满足。
因此，腾讯云原生数据库 TDSQL-C MySQL 版全新支持了分析引擎功能，为用户提供高效且实时的数据分析服务。

什么是分析引擎

分析引擎是 TDSQL-C MySQL 版针对实时数据分析与复杂查询场景推出的全新功能。其可插拔式的引擎设计使得创建和销毁过程变得灵活便捷，同时能够为用户提供海量数据的处理能力和高效实时的复杂分析能力。

支持地域和可用区

分析引擎支持的地域和可用区请参见 地域和可用区 。

功能优势

- **高速分析引擎 LibraDB**

  LibraDB 引擎拥有高效的复杂查询分析性能，让您的业务分析系统可以高效地从海量数据库中及时获取到有用的信息。LibraDB 引擎支持向量化引擎、大规模并行执行等针对分析类查询的加速特性，无论是在超大表的多表 JOIN、数据聚合和排序，还是复杂嵌套 SQL 等查询场景，LibraDB 引擎都能提供出色的性能体验。
- **可插拔式分析引擎**

  LibraDB 引擎兼容 MySQL 协议和语法，用户无需修改业务逻辑，可以直接将复杂的查询语句放置于 LibraDB 中执行，同时亦可根据业务的实际情况选择是否开启分析引擎，并且在无需分析加速的时候，可随时关闭分析引擎，以达到控制成本的目的。
- **实时列存数据加载能力**

  通过 LibraDB 引擎内置的数据同步组件，可快速将 TDSQL-C MySQL 版中的存量数据加载至分析引擎中。同时在完成数据加载后，亦可实时同步在读写实例中对数据的所有变更，使行列数据变得实时一致。另外，针对传统列式存储在高并发数据更新与删除的场景中数据变更低效的问题，LibraDB 引擎也提供了在高并发数据更新场景下的列式存储能力，可以支撑实时的数据同步，以达到0延时的效果。
- **指定数据加载能力**

  作为传统的只读实例而言，所有的主库数据均需要同步到从库中。但是对于分析引擎而言，可以支持指定对象加载到分析引擎中，而不是一定要求所有的对象。用户可以指定需要使用分析引擎加速的库表，或者有数据分析价值的库表加载到分析引擎中进行数据分析，灵活控制分析引擎所占用的磁盘大小。
- **多源数据合并能力**

  在分析集群的产品形态中可通过添加多个数据源端，将多个数据库中的数据汇聚同步到分析引擎中，以实现库表合并能力，提供高效实时的数据分析服务。
- **超高数据压缩率**

  基于列式存储的结构，提供超高数据扫描性能的同时，还能够同时提供平均4 – 5倍的压缩率，大幅度降低存储的成本。
- **完善的云上托管能力**

  无需运维复杂的 ETL 逻辑，无需关心数据库的后端运维，通过全托管的产品设计，让您获得开箱即用的数据分析能力体验。同时，通过全面监控功能，从 TXSQL 到分析引擎，从链路层到存储层，精心筛选出核心指标，为您去繁化简，让您能迅速通过关键指标了解实例健康情况，为业务系统的使用提供有效的优化指引。除此之外，您还可以自定义阈值告警，提前防范可能出现的异常情况。

适用场景

分析引擎旨在为用户提供实时的高性能数据分析，以帮助用户解决自行构建 ETL 工具同步数据的复杂运维难题。通过分析引擎的功能，用户可以轻松地一键化地构建数据分析实例，以作为业务决策的依据，充分发挥数据的价值。

- **报表分析、实时看板**

  面向企业内部分析和管理者的报表系统，可以及时、实时地查看到业务系统线上的运行情况；或者面向业务运营的数据分析业务。此类业务场景中的查询 SQL 复杂，且查询模式不固定，需要较高的吞吐，同时线上数据较多，使用分析引擎可满足业务在此类场景下的实时性与高性能。
- **用户画像、行为分析**

  在广告业务、游戏运营场景下，经常会需要针对用户行为、用户画像进行深度分析，分析后的结果用于实时的经营决策。此类场景数据量大、需要及时返回数据分析结果，查询 QPS 较高，使用分析引擎，用户可以在此类场景中快速获得需要分析的数据，用于用户行为分析，以作为精准推送相关业务的决策依据。
- **实时数据仓库**

  在电商大促订单的数据分析、物流行业的运单分析、金融行业的绩效分析、指标计算、直播质量分析、广告推送分析、智能驾驶舱、探针分析场景下亦可使用分析引擎以获取超高的复杂查询性能。
- **大数据对账、批量计算**

  在一些在线业务中，特别是与金钱有关的业务场景中，需要定时对数据进行统计合并的对账计算，在传统的行式数据中进行数据批量计算对账效率低，资源消耗高，无法快速的达到业务预期。通过借助分析引擎的超强并发计算能力，可以以极高的效率来完成业务需求。
- **汇聚分析**

  在一些在线业务中，需要将多个业务模块的数据进行汇聚查询，或者在分库分表场景，需要将分散的数据进行合并。通过分析集群的能力，可以轻松地将数据合并汇聚并进行配置。无需自建复杂的同步链路，可高效的实现业务需求。

版权所有：腾讯云计算（北京）有限责任公司

功能限制与兼容性说明
兼容性与使用限制

最近更新时间：2025-07-02 16:34:42

本文为您介绍只读分析引擎功能的使用限制和兼容性说明。

> **说明：**
> 除本文中提到的支持场景，只读分析引擎均不支持其他特殊场景。

产品架构与可用性差异

TDSQL-C MySQL 版的读写实例和只读实例使用 存算分离架构 实现，在弹性性能与扩展性上具备天然优势。但是只读分析引擎与读写实例和普通的只读实例不同。为了确保在复杂查询场景下的性能，只读分析引擎使用的是存储与计算集中架构，其数据是存储于计算节点所在的同一服务器的本地磁盘中。基于此区别需要注意存在如下差异点：

- 只读分析引擎扩缩容需要搬迁数据，故在扩缩容场景下无法与读写实例和普通只读实例的效率保持一致。(当前版本暂不支持扩缩容)
- 只读分析引擎在单节点场景下，并不具备高可用能力，若节点存在故障则会导致此只读分析引擎不可用。若需要只读分析引擎能够为您提供连续服务，请申请多个只读分析引擎。

语法限制

在只读分析引擎中，只能执行只读的查询语句，无法对数据进行任何变更操作，包括 DDL 和 DML 操作。

- 在只读分析引擎中，仅支持 SELECT 查询语句。而 SELECT 语句中依然存在少量关键字与语法不支持，详细请见 SELECT 语句说明 。
- 在只读分析引擎中，支持 CTE 语法和窗口函数。但使用 CTE 语法和窗口函数依然存在部分限制，详细请见 CTE 语法使用说明 与 窗口函数使用说明 。
- 只读分析引擎，暂时不支持全文检索语法。
- 如 TDSQL-C MySQL 集群开启了只读分析引擎，则在“读写实例”中进行某些数据变更操作会导致只读分析引擎的数据加载出现异常，详细说明请参考 数据加载限制 。

表限制

只读分析引擎在不同版本中建议的最多表数量存在不同。

- **1.2404.x**：最多表数量为2000个。
- **2.2410.x**：最多表数量为20000个。

值限制

在只读分析引擎中，默认普通列的值大小限制为16MB。若存在列值超过16MB的表，则在尝试将其加载到分析引擎时会报错并暂停。此时，您可以 提交申请 来修改普通列支持的最大值大小。

SQL_MODE

和 MySQL 类似，只读分析引擎支持通过 `SET [ SESSION | GLOBAL ] sql_mode='modes'` 语句设置 SQL 模式来设置全局或者会话级别的 SQL Mode。也可以通过 `SELECT @@sql_mode` 来查询当前 SQL 的 SQL Mode。
只读分析引擎支持如下常见的 MySQL 系统 SQL_MODE，未提到的 SQL_MODE 均不支持。但需要注意的是虽然只读分析引擎支持这些 SQL_MODE，但是部分 SQL_MODE 在只读分析引擎中并不适用，如 NO_AUTO_CREATE_USER、NO_ENGINE_SUBSTITUTION 等。

名称	含义
PIPES_AS_CONCAT	将 || 视为字符串连接操作符（+）（同 CONCAT()），而不视为 OR。
ANSI_QUOTES	将 " 视为识别符，如果启用 ANSI_QUOTES，只有单引号内的会被认为是 String Literals，双引号被解释为识别符，因此不能用双引号来引用字符串。
IGNORE_SPACE	若开启该模式，系统忽略空格。例如：“user”和“user ”是相同的。
ONLY_FULL_GROUP_BY	如果未被聚合函数处理或未被 GROUP BY 的列，出现在 SELECT、HAVING、ORDER BY 中，此 SQL 不合法。
NO_UNSIGNED_SUBTRACTIO	在减运算中，如果某个操作数没有符号，不要将结果标记为 UNSIGNED（支持）。

版权所有：腾讯云计算（北京）有限责任公司

N	
NO_BACKSLASH_ESCAPES	若启用该模式，\ 反斜杠符号仅代表它自己。
STRICT_TRANS_TABLES	对于事务存储引擎启用严格模式，insert 非法值之后，回滚整条语句。
STRICT_ALL_TABLES	对于事务型表，写入非法值之后，回滚整个事务语句。
NO_ZERO_IN_DATE	在严格模式时，不接受月或日部分为0的日期。如果使用 IGNORE 选项，我们为类似的日期插入“0000-00-00”。在非严格模式时，可以接受该日期，但会生成警告。
NO_ZERO_DATE	在严格模式时，不要将“0000-00-00”作为合法日期。您仍然可以用 IGNORE 选项插入0日期。在非严格模式时，可以接受该日期，但会生成警告。
ALLOW_INVALID_DATES	不检查全部日期的合法性，仅检查月份值是否在1至12之间，以及日期值是否在1到31之间，仅适用于 DATE 和 DATETIME 列，TIMESTAMP 列需要全部检查其合法性。
ERROR_FOR_DIVISION_BY_ZERO	启用该模式，在 INSERT 或 UPDATE 过程中，被除数为0值时，系统产生错误。若未启用该模式，被除数为0值时，系统产生警告，并用 NULL 代替。
REAL_AS_FLOAT	将 REAL 视为 FLOAT 的同义词，而不是 DOUBLE 的同义词。
NO_DIR_IN_CREATE	创建表时，忽视所有 INDEX DIRECTORY 和 DATA DIRECTORY 指令，该选项仅对从复制服务器有用。
NO_AUTO_CREATE_USER	防止 GRANT 自动创建新用户，但指定密码除外（但是在只读分析引擎中没有实际作用）。
NO_ENGINE_SUBSTITUTION	如果需要的存储引擎被禁用或未编译，可以防止自动替换存储引擎（但是在只读分析引擎中没有实际作用）。

字符集和排序规则

字符集（character set）是符号与编码的集合。只读分析引擎中的默认字符集是 utf8mb4。
排序规则（collation）是在字符集中比较字符以及字符排序顺序的规则。例如，在二进制排序规则中，比较 A 和 a 的结果是不一样的。
目前只读分析引擎支持的字符集和排序规则如下表：

字符集	描述	支持的排序规则	最大大小
utf8	UTF-8 Unicode	utf8_bin	3
utf8mb4	UTF-8 Unicode	utf8mb4_bin	4

> **注意：**
> 当读写实例中的对象采用其他字符集时，对于数据加载至只读分析引擎不受任何影响，但是某一些特殊字符在只读分析引擎中进行查询时将存在异常，同时也会因为排序规则的不同导致排序结果不一致的情况出现。

其他行为说明

在只读分析引擎中执行 `SELECT …… GROUP BY expr` 的返回结果与 MySQL 8.0 保持一致，默认不排序，与 MySQL 5.7 会有一定区别，MySQL 5.7 会默认排序。因此，无论是在 MySQL 5.7 版本还是 MySQL 8.0 版本中构建的只读分析引擎，都是如此的逻辑。

数据加载限制

最近更新时间：2025-05-07 11:50:52

由于只读分析引擎通过列式存储构建数据，因此在某些特殊的 MySQL 使用场景下，会存在部分场景不支持的情况。如下所示：

- **无主键且无唯一键的表支持说明**

  在1.2404.x 版本中：当表没有主键又没有唯一键时，表将无法加载到只读分析引擎中。这要求表必须包含主键或者唯一键。在 LibraDB 引擎中，将默认使用表主键或唯一键进行列式数据构建。同时在1.2404.x 版本中，不支持任何形式的对表主键进行修改的 DDL 语句。若在 TDSQL-C MySQL 版中对主键进行修改，则只读分析引擎的此表将停止加载并不可查询。若需要恢复使用，则需要将此表移除加载，然后再重新进行数据加载。

  在2.2410.x 版本中：表在无主键或无唯一键时也能加载到只读分析引擎中。同时也支持表主键的变更操作。但依然存在部分特殊场景不支持，不支持的场景如下：

  - 表仅存在 time、date、timestamp、datetime、float、double 这些字段，无其他字段类型。
  - 当对表进行了主键的 DDL，导致表中仅存在如上字段类型，无其他字段类型时。

- **使用 float、double 字段类型作为主键的表不支持加载到只读分析引擎**

  float、double 字段为浮点字段类型，不支持这两种字段类型作为主键的表将数据加载到只读分析引擎中。

- **存储过程、自定义函数、触发器、外键约束、event、索引均不会在只读分析引擎加载**

  不支持在列式存储中构建以上特殊对象。

- **存在字段为空间类型的表无法加载到只读分析引擎**

  存在空间字段类型的表无法加载到只读分析引擎。

- **不支持临时表加载到只读分析引擎**

  由于临时表数据变更不记录日志，因此无法加载临时表的数据到只读分析引擎中。

- **加载到分析引擎中的表被 rename 后的行为**

  当某一个表被加载到分析引擎中后，使用 rename 语句将表改名后，被 rename 后的表也将在分析引擎中自动加载。此时再创建一张与被 rename 的表的同名表，此新建的表亦会自动加载到分析引擎。如表 A 已经在只读分析引擎中加载为列存，此时将表 A 改名为表 B，则在只读分析引擎中，此表也会被改为表 B。同时若在读写实例中又新增一张表为表 A，则表 A 同样会自动加载到只读分析引擎中。

- **不支持的表创建操作**

  当对整库或者整实例进行预加载到分析引擎时，在 TDSQL-C MySQL 版读写实例中执行 `create table … select` 语句，此时此表将无法自动加载到分析引擎，若想要此表加载到分析引擎，则需要在表完成创建后将此表从分析引擎加载列表中移除，再通过控制台操作将此表加载到分析引擎。

- **不支持主键字段为超长数字类型的表加载为列存**

  在1.2404.x 版本中，不支持主键为 Decimal 超出128长度的表加载为列存。
  在2.2410.x 版本中，不支持主键为 Decimal 超出256长度的表加载为列存。

- **不支持列级权限**

  只读分析引擎默认会同步读写实例中的所有用户关于对象的查询权限，但是不会同步列级权限。故无法在只读分析引擎中实现对列级权限的管控。

- **不支持的数据类型**

  只读分析引擎存在部分 数据类型 不支持，若对象存在不支持的数据类型，则此表将无法加载到分析引擎中。

- **不支持的表结构**

  只读分析引擎不支持生成列语法，无论是虚拟列还是物理列。当表定义语句中包含生成列语法，会导致数据加载中断，如存在此类场景，请将表从只读分析引擎的表加载列表中移除。

- **视图的加载**

  若需要加载视图，请将视图所在的库直接加载到只读分析引擎中，在此库中的视图对象将自动加载到分析引擎。若无需整库加载到分析引擎，又想要在只读分析引擎中使用视图，请 提交工单 处理。

- **不支持的字段类型转换**

  在只读分析引擎中存在某一些字段类型的转换不支持，若在 TDSQL-C MySQL 版的读写实例中进行表类型的转换，可能会导致只读分析引擎的数据加载任务中止，所有表加载状态变为暂停中。详细的类型转换支持情况请参考 类型转换函数支持说明 。

  > **注意：**
  > 若在 TDSQL-C MySQL 版中对表的字段类型进行修改，而此类型修改在只读分析引擎中不支持，则所有表加载状态均会暂停。此时若需要恢复使用，则需要将此表移除加载后，再重新对此表进行数据加载。

- **分区表的 DDL 同步说明**

  在1.2404.x 版本中：分区表默认可以加载到只读分析引擎中并且支持查询。但是无法支持对分区表的分区执行相关 DDL 的同步。例如：重建分区、对分区进行 OPTIMIZE、修复分区、CHECK 分区、交换分区、删除分区、合并分区等操作。同时在只读分析引擎中也不支持单独针对某个子分区进行查询。

> **注意：**
> 当在 TDSQL-C MySQL 版的读写实例中对分区表进行 drop 子分区、truncate 分区或执行 exchange 分区时，此表在只读分析引擎中将不可查询。若需要恢复使用，则需要将此表移除加载，然后再重新进行数据加载。

在2.2410.x 版本中：支持将分区表加载到只读分析引擎中，并且支持在源端对分区表进行如下 DDL 变更。

- 支持 Range/List 分区表的 Drop Partition/Subpartition template。
- 支持 Range/List 分区表的 Add Partition/Subpartition template。
- 在无函数场景下，支持的分区数据类型有：Uint8、Uint16、Uint32、Uint64、Int8、Int16、Int32、Int64。
- 支持的分区函数有：year、month、day。这几个函数支持的数据类型有：DATE、DATETIME、TIMESTAMP。

版权所有：腾讯云计算（北京）有限责任公司

函数与运算符说明
位运算支持说明

最近更新时间：2024-09-27 09:29:31

本文介绍在只读分析引擎中对于位运算的支持情况，针对输入参数中的类型限制进行了列举。如存在文档中未提到的入参与使用方法则均不支持。

位运算支持类型

& 按位与

语法： `value1 & value2`
value1支持的数据类型：int，bigint，double，decimal，varchar，char。
value2支持的数据类型：int，bigint，double，decimal，varchar，char。

| 按位或

语法： `value1 | value2`
value1支持的数据类型：int，bigint，double，decimal，varchar，char。
value2支持的数据类型：int，bigint，double，decimal，varchar，char。

~ 按位取反

语法： `~value1`
value1支持的数据类型：int，bigint，double，decimal，varchar，char。

^ 按位异或

语法： `value1 ^ value2`
value1支持的数据类型：int，bigint，double，decimal，varchar，char。
value2支持的数据类型：int，bigint，double，decimal，varchar，char。

版权所有：腾讯云计算（北京）有限责任公司

四则运算支持说明

最近更新时间：2025-03-07 11:30:43

目前只读分析引擎支持大部分 MySQL 的四则运算逻辑，但在部分场景下仍然存在限制和不兼容。本文介绍在只读分析引擎中四则运算的支持情况，针对输入参数中的类型限制进行了列举。如存在文档中未提到的入参与使用方法则均不支持。

四则运算使用限制

- **Decimal 类型的除法运算**：结果的小数位数会基于被除数的小数位数 + 4进行截断。如 Decimal(9, 1) / Decimal(9, 1)，则结果的小数位数会基于5进行截断。
- **Decimal 的加、减、乘、除运算**：若结果的总长度超过65，则会报错提示“out of range”。因为在只读分析引擎中 Decimal 支持的数据类型长度最大为65。
- **任意数字类型与浮点数类型（float、double）进行数学运算**：均会存在精度误差问题，核心原因是 Double 等浮点类型本身就不是严格精度的字段类型。在 MySQL 的实践中也不建议数字类型与浮点类型进行数学计算。
- **非数值类型的四则运算**：涉及到隐式转换，受限于只读分析引擎的 CAST 函数支持情况，在某些场景下，如果 CAST 不支持，相关四则运算也不支持。详细的 CAST 支持情况，请参见 类型转换函数支持说明 。

四则运算支持类型

+ 加运算符

语法： `op1 + op2`
op1支持值类型：int，bigint，float，double，decimal。
op2支持值类型：int，bigint，float，double，decimal。

– 减运算符

语法： `op1 - op2`
op1支持值类型：int，bigint，float，double，decimal。
op2支持值类型：int，bigint，float，double，decimal。

* 乘运算符

语法： `op1  op2`
op1支持值类型：int，bigint，float，double，decimal。
op2支持值类型：int，bigint，float，double，decimal。

/ 除以运算符

语法： `op1 / op2`
op1支持值类型：int，bigint，float，double，decimal。
op2支持值类型：int，bigint，float，double，decimal。

% 取余运算符

语法： `op1 % op2`
op1支持值类型：int，bigint，float，double，decimal。
op2支持值类型：int，bigint，float，double，decimal。

比较运算支持说明

最近更新时间：2024-10-23 11:50:21

目前只读分析引擎支持大部分 MySQL 的比较运算逻辑，但在部分场景下仍然存在限制和不兼容。本文介绍在只读分析引擎中比较运算的支持情况，针对输入参数中的类型限制进行了列举。如存在文档中未提到的入参与使用方法则均不支持。

比较运算支持类型

>= 大于等于比较

语法： `op1 >= op2`
op1支持值类型：int，bigint，float，double，decimal，char，varchar，date，datetime。
op2支持值类型：int，bigint，float，double，decimal，char，varchar，date，datetime。

<= 小于等于比较

语法： `op1 <= op2`
op1支持值类型：int，bigint，float，double，decimal，char，varchar，date，datetime。
op2支持值类型：int，bigint，float，double，decimal，char，varchar，date，datetime。

= 等值比较

语法： `op1 = op2`
op1支持值类型：int，bigint，float，double，decimal，char，varchar，date，datetime。
op2支持值类型：int，bigint，float，double，decimal，char，varchar，date，datetime。

!= 不等比较

语法： `op1 != op2`
op1支持值类型：int，bigint，float，double，decimal，char，varchar，date，datetime。
op2支持值类型：int，bigint，float，double，decimal，char，varchar，date，datetime。

< 小于比较

语法： `op1 < op2`
op1支持值类型：int，bigint，float，double，decimal，char，varchar，date，datetime。
op2支持值类型：int，bigint，float，double，decimal，char，varchar，date，datetime。

> 大于比较

语法： `op1 > op2`
op1支持值类型：int，bigint，float，double，decimal，char，varchar，date，datetime。
op2支持值类型：int，bigint，float，double，decimal，char，varchar，date，datetime。

版权所有：腾讯云计算（北京）有限责任公司

逻辑运算支持说明

最近更新时间：2024-10-08 10:12:31

目前只读分析引擎支持大部分 MySQL 的逻辑运算，但在部分场景下仍然存在限制和不兼容。本文介绍在只读分析引擎中逻辑运算的支持情况，针对输入参数中的类型限制进行了列举。如存在文档中未提到的入参与使用方法则均不支持。

逻辑运算支持类型

and 运算

语法： `op1 and op2`
op1支持值类型：仅支持 bool 数据类型。
op2支持值类型：仅支持 bool 数据类型。

or 运算

语法： `op1 or op2`
op1支持值类型：仅支持 bool 数据类型。
op2支持值类型：仅支持 bool 数据类型。

not 运算

语法： `not op1`
op1支持值类型：仅支持 bool 数据类型。

case 运算

语法： `case when search_condition then statement_list else statement_list end`
search_condition 支持值类型：仅支持 bool 数据类型。
statement_list 支持值类型：int，bigint，double，decimal，varchar，char。

if 运算

语法： `if(expr1,expr2,expr3)`
expr1支持值类型：仅支持 bool 数据类型。
expr2支持值类型：int，bigint，double，decimal，varchar，char。
expr3支持值类型：int，bigint，double，decimal，varchar，char。

ifnull 运算

语法： `ifnull(expr1, expr2)`
expr1支持值类型：int，bigint，double，decimal，varchar，char。
expr2支持值类型：int，bigint，double，decimal，varchar，char。

isnull 运算

语法： `isnull(expr1)`
expr1支持值类型：int，bigint，double，decimal，varchar，char。

in 运算

- **语法：** `expr IN (value,...)`
  expr 支持值类型：int，bigint，double，decimal，varchar，char。
  value 支持值类型：int，bigint，double，decimal，varchar，char。

  > ⚠ **注意：**
  > 在多 value 值的场景下，value 只能为常量，不能为变量或者字段名，如 columnA in (columnB,varA)，只能为 column in ('1','2','3') 此类的使用方法。

- **语法：** `expr IN (sub_query)`
  expr 支持值类型：int，bigint，double，decimal，varchar，char。

sub_query 支持值类型：支持子查询、单一字段。

like 运算

语法： `expr like pattern`
expr 支持值类型：int，bigint，double，decimal，varchar，char。
pattern 支持值类型：int，bigint，double，decimal，varchar，char。

> **注意：**
> pattern 只支持常量，不支持代入变量或者字段名。

coalesce 运算

语法： `coalesce(value,...)`
value 支持值类型：int，bigint，double，decimal，varchar，char。

字符串函数支持说明

最近更新时间：2025-03-18 17:38:12

目前只读分析引擎支持大部分 MySQL 的字符串函数，但在部分场景下仍然存在限制和不兼容。本文介绍在只读分析引擎中字符串函数的支持列表，并针对输入参数中的类型限制进行了列举。如存在文档中未提到的入参与函数则均不支持。

字符串函数支持类型

substr 函数

字符串截取函数，有着以下几种使用方法。

- **语法：** `substr(str, start)`
  str 支持值类型：支持 char，varchar 数据类型。
  start 支持值类型：bigint，double，decimal。
- **语法：** `substr(str from start)`
  str 支持值类型：支持 char，varchar 数据类型。
  start 支持值类型：bigint，double，decimal。
- **语法：** `substr(string, start, length)`
  str 支持值类型：支持 char，varchar 数据类型。
  start 支持值类型：bigint，double，decimal。
  length 支持值类型：bigint，double，decimal。
- **语法：** `substr(string from start for length)`
  str 支持值类型：支持 char，varchar 数据类型。
  start 支持值类型：bigint，double，decimal。
  length 支持值类型：bigint，double，decimal。

char_length 函数

字符串长度判断函数，返回字符串的字符数。
语法： `char_length(str)`
str 支持值类型：支持 char，varchar 数据类型。

replace 函数

字符串替换函数。
语法： `replace(str, from_str, to_str)`
str 支持值类型：支持 char，varchar 数据类型。
from_str 支持值类型：支持 char，varchar 数据类型。
to_str 支持值类型：支持 char，varchar 数据类型。

concat 函数

字符串连接函数。
语法： `concat(str1, str2, ...)`
str1支持值类型：支持 char，varchar 数据类型。
str2支持值类型：支持 char，varchar 数据类型。

concat_ws 函数

带分隔符的字符串连接函数。
语法： `concat_ws(str1, str2, ...)`
str1支持值类型：支持 char，varchar 数据类型。
str2支持值类型：支持 char，varchar 数据类型。

left 函数

从字符串的第一个字符开始返回指定个数的字符。
语法： `left(str, len)`
str1支持值类型：支持 char，varchar 数据类型。

版权所有：腾讯云计算（北京）有限责任公司

len 支持值类型：支持 int，bigint，double，decimal 数据类型。

right 函数

从字符串的最后一个字符开始返回指定个数的字符。
语法： `right(str, len)`
str1支持值类型：支持 char，varchar 数据类型。
len 支持值类型：支持 int，bigint，double，decimal 数据类型。

ascii 函数

把字符转换成 ascii 码值的函数。
语法： `ascii(str)`
str1支持值类型：支持 char，varchar 数据类型。

length 函数

字符串长度判断函数，返回字符串的字节数。
语法： `length(str)`
str1支持值类型：支持 char，varchar 数据类型。

trim 函数

清除字符串前后的不必要字符的函数。
语法： `trim([{BOTH | LEADING | TRAILING} [remstr] FROM] str)`
remstr 支持值类型：支持 char，varchar 数据类型。
str 支持值类型：支持 char，varchar 数据类型。

ltrim 函数

清除字符串前的不必要字符。
语法： `ltrim(str)`
str 支持值类型：支持 char，varchar 数据类型。

rtrim 函数

清除字符串后的不必要字符。
语法： `rtrim(str)`
str 支持值类型：支持 char，varchar 数据类型。

position 函数

返回与子字符串的位置对应的函数。
语法： `position(substr IN str)`
str 支持值类型：支持 char，varchar 数据类型。
substr 支持值类型：支持 char，varchar 数据类型。

instr 函数

查找一个字符串在另一个字符串中首次出现的位置。
语法： `instr(str, substr)`
str 支持值类型：支持 char，varchar 数据类型。
substr 支持值类型：支持 char，varchar 数据类型。

locate 函数

返回字符串中第一次出现的子字符串的位置。
语法： `locate(str, substr)`
str 支持值类型：支持 char，varchar 数据类型。
substr 支持值类型：支持 char，varchar 数据类型。

lower 函数

返回字符串的小写字符串。

版权所有：腾讯云计算（北京）有限责任公司

语法： `lower(str)`
str 支持值类型：支持 char，varchar 数据类型。

ucase 函数

将给定字符串中的所有小写字母转换为大写字母。
语法： `ucase(str)`
str 支持值类型：支持 char，varchar 数据类型。

upper 函数

将给定字符串中的所有小写字母转换为大写字母。
语法： `upper(str)`
str 支持值类型：支持 char，varchar 数据类型。

substring_index 函数

支持反向的字符串截取函数。
语法： `substring_index(str, delim, count)`
str 支持值类型：支持 char，varchar 数据类型。
delim 支持值类型：支持 char，varchar 数据类型，只能是常量，不接受类似于字段值的变量值。
count 支持值类型：支持 int，bigint，double，decimal 数据类型，支持负数，只能是常量，不接受类似于字段值的变量值。

lpad 函数

对字符串开头进行填充以达到指定的长度。
语法： `lpad(str, len, padstr)`
str 支持值类型：支持 char，varchar 数据类型。
len 支持值类型：支持 int，bigint，double，decimal 数据类型，只能是常量，不接受类似于字段值的变量值。
padstr 支持值类型：支持 char，varchar 数据类型。

rpad 函数

对字符串末尾进行填充以达到指定的长度。
语法： `rpad(str, len, padstr)`
str 支持值类型：支持 char，varchar 数据类型。
len 支持值类型：支持 int，bigint，double，decimal 数据类型，只能是常量。
padstr 支持值类型：支持 char，varchar 数据类型。

strcmp 函数

比较两个字符串并根据比较结果返回整数。
语法： `strcmp(expr1, expr2)`
expr1支持值类型：支持 char，varchar 数据类型。
expr2支持值类型：支持 char，varchar 数据类型。

quote 函数

在查询中引用包含特殊字符的字符串的函数。
语法： `quote(str)`
str 支持值类型：支持 char，varchar 数据类型。

find_in_set 函数

找到在字符串中的对应字符的位置。目前仅支持 strlist 为字符串类型，不支持 set 类型。
语法： `find_in_set(str,strlist)`
str 要查询的字符串。
strlist 字段名参数以“,”分隔，例如：(1,2,6,8)。

数学运算函数支持说明

最近更新时间：2024-10-08 10:08:21

目前只读分析引擎支持大部分 MySQL 的数学运算函数，但在部分场景下仍然存在限制和不兼容。本文介绍在只读分析引擎中数学运算函数的支持列表，并针对输入参数中的类型限制进行了列举。如存在文档中未提到的入参与函数则均不支持。

数学运算函数支持类型

round 函数

四舍五入函数。
语法： `round(x)`
x 支持值类型：支持 decimal 数据类型。

abs 函数

求绝对值函数。
语法： `abs(x)`
x 支持值类型：支持 int，bigint，double，decimal 数据类型。

floor 函数

向下取整函数。
语法： `floor(x)`
x 支持值类型：支持 int，bigint，double，decimal 数据类型。

ceil 函数

向上取整函数。
语法： `ceil(x)`
x 支持值类型：支持 int，bigint，double，decimal 数据类型。

ceiling 函数

向上舍入函数。
语法： `ceiling(x)`
x 支持值类型：支持 int，bigint，double，decimal 数据类型。

sqrt 函数

平方根函数。
语法： `sqrt(x)`
x 支持值类型：支持 int，bigint，double，decimal 数据类型。

log 函数

对数函数。
语法： `log(x)`
x 支持值类型：支持 int，bigint，double，decimal 数据类型。

log2函数

以2为底的对数函数。
语法： `log2(x)`
x 支持值类型：支持 int，bigint，double，decimal 数据类型。

log10函数

以10为底的对数函数。
语法： `log10(x)`
x 支持值类型：支持 int，bigint，double，decimal 数据类型。

ln 函数

返回数字的自然对数的函数。
语法： `ln(x)`
x 支持值类型：支持 int，bigint，double，decimal 数据类型。

exp 函数

返回 e 的 x 次幂的函数。
语法： `exp(x)`
x 支持值类型：支持 int，bigint，double，decimal 数据类型。

pow 函数

计算 x 的 y 次方函数。
语法： `pow(x, y)`
x 支持值类型：支持 int，bigint，double，decimal 数据类型。
y 支持值类型：支持 int，bigint，double，decimal 数据类型。

sign 函数

用以判断实数的正负号。
语法： `sign(x)`
x 支持值类型：支持 int，bigint，double，decimal 数据类型。

radians 函数

将度数转换为弧度。
语法： `radians(x)`
x 支持值类型：支持 int，bigint，double，decimal 数据类型。

degrees 函数

弧度转换为度。
语法： `degrees(x)`
x 支持值类型：支持 int，bigint，double，decimal 数据类型。

conv 函数

将一个数字从一个数字基系统转换为另一个，并将结果作为字符串值返回。
语法： `conv(n, from_base, to_base)`
n 支持值类型：支持 int，bigint，double，decimal 数据类型。
from_base 支持值类型：支持 int，bigint，double，decimal 数据类型。
to_base 支持值类型：支持 int 数据类型。

crc32函数

循环冗余校验函数。
语法： `crc32(expr)`
expr 支持值类型：支持 int，bigint，double，decimal 数据类型。

greatest 函数

返回 N 个值中的最大值。
语法： `greatest(value1, value2, ...)`
value1支持值类型：支持 int，bigint，double，decimal 数据类型。
value2支持值类型：支持 int，bigint，double，decimal 数据类型。

least 函数

返回 N 个值中的最小值。
语法： `least(value1, value2, ...)`
value1支持值类型：支持 int，bigint，double，decimal 数据类型。
value2支持值类型：支持 int，bigint，double，decimal 数据类型。

版权所有：腾讯云计算（北京）有限责任公司

日期函数支持说明

最近更新时间：2025-03-18 17:38:12

目前只读分析引擎支持大部分 MySQL 的日期函数，但在部分场景下仍然存在限制和不兼容。本文介绍在只读分析引擎中日期函数的支持列表，并针对输入参数中的类型限制进行了列举。如存在文档中未提到的入参与函数则均不支持。

日期函数支持类型

datediff 函数

两个日期之间有多少个指定的时间间隔函数。
语法： `datediff(date1, date2)`
date1支持值类型：支持 date，datetime 数据类型。
date2支持值类型：支持 date，datetime 数据类型。

year 函数

计算年份函数。
语法： `year(date1)`
date1支持值类型：支持 date，datetime，char，varchar 数据类型。

month 函数

计算月份函数。
语法： `month(date1)`
date1支持值类型：支持 date，datetime，char，varchar 数据类型。

day 函数

计算当天号数函数。
语法： `day(date1)`
date1支持值类型：支持 date，datetime，char，varchar 数据类型。

hour 函数

计算小时函数。
语法： `hour(date1)`
date1支持值类型：支持 date，datetime 数据类型。

minute 函数

计算分钟函数。
语法： `minute(date1)`
date1支持值类型：支持 date，datetime 数据类型。

second 函数

计算秒函数。
语法： `second(date1)`
date1支持值类型：支持 date，datetime 数据类型。

microsecond 函数

返回时间/日期时间的微秒部分。
语法： `microsecond(date1)`
date1支持值类型：支持 date，datetime 数据类型。

quarter 函数

返回给定日期值（1到4之间的数字）的一年中的季度。
语法： `quarter(date1)`
date1支持值类型：支持 date，datetime，varchar，char 数据类型。

版权所有：腾讯云计算（北京）有限责任公司

date 函数

日期函数。
语法： `date(date1)`
date1支持值类型：支持 date，datetime，char，varchar 数据类型。

sysdate 函数

返回系统时间函数。
语法： `sysdate()`

date_add 函数

日期相加函数。
语法： `date_add(date1, interval expr type)`
date1支持值类型：支持 date，datetime 数据类型。
expr 支持值类型：支持 int，long，double，decimal。
type 支持值类型：支持 DAY，DAY 为关键字。

date_sub 函数

日期相减函数。
语法： `date_sub(date1, interval expr type)`
date1支持值类型：支持 date，datetime 数据类型。
expr 支持值类型：支持 int，long，double，decimal。
type 支持值类型：支持 DAY，DAY 为关键字。

dayofyear 函数

返回给定日期是当年的第几天的函数。
语法： `dayofyear(date1)`
date1支持值类型：支持 date，datetime，char，varchar 数据类型。

dayofmonth 函数

返回给定日期是当月的哪一天的函数。
语法： `dayofmonth(date1)`
date1支持值类型：支持 date，datetime，char，varchar 数据类型。

dayofweek 函数

返回给定日期是当周的哪一天的函数。
语法： `dayofweek(date1)`
date1支持值类型：支持 date，datetime，char，varchar 数据类型。

dayname 函数

返回指定日期对应的星期的名称。
语法： `dayname(date1)`
date1支持值类型：支持 date，datetime，char，varchar 数据类型。

monthname 函数

返回指定日期的月份的名称。
语法： `monthname(date1)`
date1支持值类型：支持 date，datetime，char，varchar 数据类型。

to_seconds 函数

返回指定的日期/日期时间从0年开始的秒数。
语法： `to_seconds(date1)`
date1支持值类型：支持 date，datetime，char，varchar 数据类型。

last_day 函数

版权所有：腾讯云计算（北京）有限责任公司

返回月份中的最后一天的函数。
语法： `last_day(date1)`
date1支持值类型：支持 date，datetime，char，varchar 数据类型。

sleep 函数

在执行时暂停几秒。
语法： `sleep(N)`
N 支持值类型：int 数据类型。

now 函数

获取当前时间函数。
语法： `NOW()`
获取当前的时间，无需入参。

current_timestamp 函数

获取当前时间函数，等同于 NOW 函数。
语法： `CURRENT_TIMESTAMP()`
获取当前的时间，无需入参。

curdate 函数

获取当前日期函数，等同于 CURRENT_DATE 函数。
语法： `CURDATE()`
获取当前的日期，无需入参。

current_date 函数

获取当前日期函数，等同于 CURDATE 函数。
语法： `CURRENT_DATE()`
获取当前的时间，无需入参。

current_time 函数

获取当前时间函数。
语法： `CURRENT_TIME()`
获取当前的时间，无需入参。

curtime 函数

获取当前时间函数，等同于 CURRENT_TIME 函数。
语法： `CURTIME()`
获取当前的时间，无需入参。

from_unixtime 函数

将时间戳转换为时间格式函数。
语法： `FROM_UNIXTIME(unix_timestamp[,format])`
unix_timestamp 支持 int 类型数字。
format 支持的格式如下面 date_format 函数 的表格示例。若 format 为空，则默认为 %Y-%m-%d %H:%i:%s 格式。

unix_timestamp 函数

基于日期时间将其转换为时间戳的函数。
语法： `UNIX_TIMESTAMP([date])`
date 支持 DATE 标准的日期类型。

> **注意：**
> 仅1.2404.17.0及以后的版本支持 FROM_UNIXTIME 函数与 UNIX_TIMESTAMP 函数。

weekday 函数

获取参数值所对应的星期。
语法： `weekday(date)`
获取一个日期对应的星期数，入参支持 Date 或 Datetime 值，返回结果范围：0 – 6，分别对应：星期日 – 星期六。如果日期为 NULL 或 0000-00-00，函数则返回 NULL。

date_format 函数

日期格式化函数。
语法： `date_format(date1, format)`
date1支持值类型：支持 date，datetime，char，varchar 数据类型，不支持 date1包含 null 场景。format 规定日期/时间的输出格式。可以使用的格式如下表。

格式	描述
%a	缩写星期名。
%b	缩写月名。
%c	月，数值。
%D	带有英文前缀的月中的天。
%d	月的天，数值（00 – 31）。
%e	月的天，数值（0 – 31）。
%f	微秒。
%H	小时（00 – 23）。
%h	小时（01 – 12）。
%I	小时（01 – 12）。
%i	分钟，数值（00 – 59）。
%j	年的天（001 – 366）。
%k	小时（0 – 23）。
%l	小时（1 – 12）。
%M	月名。
%m	月，数值（00 –12）。
%p	AM 或 PM。
%r	时间，12 – 小时（hh:mm:ss AM 或 PM）。
%S	秒（00 – 59）。
%s	秒（00 – 59）。
%T	时间，24 – 小时（hh:mm:ss）。
%U	周（00 – 53）星期日是一周的第一天。
%u	周（00 – 53）星期一是一周的第一天。
%V	周（01 – 53）星期日是一周的第一天，与 %X 使用。
%v	周（01 – 53）星期一是一周的第一天，与 %x 使用。
%W	星期名。
%w	周的天（0 = 星期日，6 = 星期六）。

版权所有：腾讯云计算（北京）有限责任公司

%X	年，其中的星期日是周的第一天，4位，与 %V 使用。
%x	年，其中的星期一是周的第一天，4位，与 %v 使用。
%Y	年，4位。
%y	年，2位。

版权所有：腾讯云计算（北京）有限责任公司

聚合函数支持说明

最近更新时间：2025-05-21 17:47:22

目前只读分析引擎支持大部分 MySQL 的聚合函数，但在部分场景下仍然存在限制和不兼容。本文介绍在只读分析引擎中聚合函数的支持列表，并针对输入参数中的类型限制进行了列举。如存在文档中未提到的入参与函数则均不支持。

聚合函数支持类型

count 函数

统计行数。
语法： `count(val)`
val 支持值类型：int，bigint，double，decimal，varchar，char。

sum 函数

统计求和。
语法： `sum(val)`
val 支持值类型：int，bigint，double，decimal，varchar，char。

avg 函数

统计平均数。
语法： `avg(val)`
val 支持值类型：int，bigint，double，decimal，varchar，char。

max 函数

获取最大值。
语法： `max(val)`
val 支持值类型：int，bigint，double，decimal，varchar，char。

min 函数

获取最小值。
语法： `min(val)`
val 支持值类型：int，bigint，double，decimal，varchar，char。

group_concat 函数

将同一组内多个相同字段的值合并成一个字符串。
语法： `group_concat(val)`
val 支持值类型：int，bigint，double，decimal，varchar，char。

any_value 函数

返回分组中任意一个值。
语法： `any_value(val)`
val 支持值类型：int，bigint，double，decimal，varchar，char。

版权所有：腾讯云计算（北京）有限责任公司

JSON 函数

最近更新时间：2025-02-11 16:41:53

在2.2410.x 及之后的版本中，只读分析引擎新增支持符合 RFC 7159 标准的 JSON（JavaScript Object Notation）数据类型。用户可以保存半结构化的 JSON 数据并可以访问和修改 JSON 文档中的数据。
存储在 JSON 列中的 JSON 文档可以被转换为便于快速读取和访问的格式。当服务器必须读取以二进制格式存储的 JSON 值时，不需要从文本中解析该值。
除了 JSON 数据类型，还有一组 SQL 函数可用于对 JSON 值进行创建、搜索等操作。详细支持的函数如下：

JSON_OBJECT 函数

创建 JSON 对象。
语法： `JSON_OBJECT(key1, value1, key2, value2…)`
根据入参值创建一个 JSON 对象。

JSON_REMOVE 函数

从指定 JSON 对象中删除对应的 path。
语法： `JSON_REMOVE(json_doc, path[, path] ...)`
json_doc 是 JSON 字段，path 是路径。支持多个路径。

JSON_EXTRACT 函数

从 JSON 字段中提取指定路径的元素。如果指定的路径不存在，则会返回 NULL。可指定多个 path，匹配到的多个值会以数组形式返回。
语法： `JSON_EXTRACT(json_doc, path[, path] ...)` 与 `column-> path` 同义。
json_doc 是 JSON 字段，path 是路径。

JSON_UNQUOTE 函数

将 JSON 转义成字符串输出。
语法： `JSON_UNQUOTE(column -> path)` 与 `column->> path` 同义。
json_doc 是 JSON 字段，path 是路径。

JSON_CONTAINS 函数

指定 path 是否包含指定数据。若包含则返回1，若不包含则返回0，如果有参数为 NULL 或者 path 不存在，则返回 NULL。
语法： `JSON_CONTAINS(json_doc, candidate[, path])`
json_doc 是 JSON 字段，candidate 为判断值，path 是路径。

JSON_CONTAINS_PATH 函数

检查是否存在指定路径，是否满足一个或者所有。若存在则返回1，若不存在则返回0，如果有参数为 NULL，则返回 NULL。
语法： `JSON_CONTAINS_PATH(json_doc, one_or_all, path[, path] …)`
json_doc 是 JSON 字段，one_or_all 中 one 表示查询到一个即返回，all 表示查询所有后才返回，path 是路径。支持多个路径。

类型转换

- 支持 JSON 到 INT、UINT、FLOAT、STRING、DATE、DATETIME、DURATION 的转换。
- 支持将 INT、UINT、FLOAT、STRING、DATE、DATETIME、DURATION 转换为 JSON。

版权所有：腾讯云计算（北京）有限责任公司

类型转换函数支持说明

最近更新时间：2025-02-11 16:41:53

在只读分析引擎中，执行 SQL 运算时会出现一些字段类型不匹配导致的隐式转换，或者显式指定 CAST 函数时的类型转换的场景。只读分析引擎在不同版本中对其支持的情况有所区别，本文将详细列举类型转换的支持情况。

> **注意：**
> 如果在 SQL 表达式中存在不同类型的值进行比较或者计算，则会触发隐式转换。在使用过程中会出现的所有隐式转换逻辑均受制于如下表所示的转换支持情况。如果存在不支持隐式转换的值进行比较或者计算等运算场景，则会提示不支持此类转换报错。

语法： `CAST (expression AS data_type)`

在1.2404.x 版本的只读分析引擎中，详细的转换类型支持情况如下表所示。

源/目标	Binary	Char	Date	DateTime	Decimal	Double	Float	Real	Signed	Time	Unsigned	Year
Bigint	✓	✓	×	×	✓	✓	×	✓	✓	×	✓	×
Binary	✓	×	×	×	✓	✓	×	✓	✓	×	✓	×
Date	✓	×	✓	✓	×	×	×	×	×	✓	×	×
Datetime	✓	×	✓	✓	×	×	×	×	×	✓	×	×
Decimal	✓	✓	×	×	✓	✓	×	✓	✓	×	✓	×
Double	✓	✓	×	×	✓	✓	×	✓	×	×	×	×
Float	✓	✓	×	×	✓	×	✓	✓	✓	×	✓	×
Real	✓	✓	×	×	✓	✓	×	✓	×	×	×	×
Signed	✓	✓	×	×	✓	✓	×	✓	✓	×	✓	×
Time	×	×	×	×	×	×	×	×	×	×	×	×
Unsigned	✓	✓	×	×	✓	✓	×	✓	✓	×	✓	×
Year	✓	✓	×	×	✓	✓	×	✓	✓	×	✓	×
Varchar	✓	✓	×	×	✓	✓	×	✓	✓	×	✓	×

在2.2410.x 版本的只读分析引擎中，详细的转换类型支持情况如下表所示。

源/目标	Int/Bigint	Binary	Char	Date	DateTime	Decimal	Double	Float	Real	Signed	Time	Unsigned	Year
Int/Bigint	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×
Binary	✓	✓	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	×
Date	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×
Datetime	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×
Decimal	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×
Double	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×

Float	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×
Real	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×
Signed	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×
Time	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×
Unsigned	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×
Year	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×
Varchar	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×

版权所有：腾讯云计算（北京）有限责任公司

数据类型

最近更新时间：2025-02-11 16:41:53

本文为您详细列举在只读分析引擎中支持的数据类型。

数字数据类型

类型分类	类型	支持情况与说明
数字数据类型	TINYINT	支持
	SMALLINT	支持
	MEDIUMINT	支持
	INT	支持
	BIGINT	支持
	DECIMAL、NUMERIC	支持
	FLOAT	支持
	DOUBLE	支持
	BIT	支持 **注意：** 只读分析引擎中的 BIT 数据类型是通过转换为无符号的 INT 类型进行支持的，故只读分析引擎仅支持将 BIT 类型数据用于一些数字判断的场景。
日期和时间数据类型	DATE	支持
	DATETIMESTAMP	支持
	TIMESTAMP	支持
	TIME	支持
	YEAR	支持
字符数据类型	CHAR	支持
	VARCHAR	支持
	BINARY	支持
	BLOB	支持
	TEXT	支持
	ENUM	支持 **注意：** 只读分析引擎中的 ENUM 与 SET 数据类型是将其转换为 String 类型进行支持的，故无法使用相关表达式或专用函数，例如：&、find_in_set。
	SET	
空间数据类型	GEOMETRY	不支持
	POINT	不支持

	LINESTRING	不支持
	POLYGON	不支持
JSON 数据类型	JSON	支持 **注意：** 在只读分析引擎内核版本2.2410.1.0及之后的版本才支持 JSON 类型，1.2404.x 版本不支持 JSON 数据类型。

只读分析引擎管理
开启只读分析引擎

最近更新时间：2025-04-27 17:58:22

目前只读分析引擎实例仅支持在实例形态为预置资源的集群中创建，不支持在实例形态为 Serverless 的集群中创建。本文为您介绍如何为 TDSQL-C MySQL 版创建只读分析引擎实例。

前提条件

- 准备好实例形态为预置资源的集群，请参见 创建集群 。
- 集群状态为运行中。

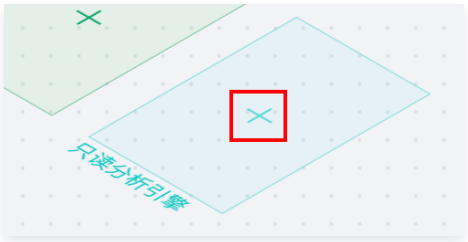
操作步骤

> **说明：**
> 目前仅支持在已经创建好的实例形态为预置资源的集群下，进行只读分析引擎的创建，不支持在新建集群的时候进行只读分析引擎的创建。

1. 登录 TDSQL-C MySQL 版控制台，参考以下不同视图模式的操作，进入创建只读分析引擎的页面。

页签视图

1. 在左侧集群列表，单击**目标集群**，进入集群管理页。
2. 在集群拓扑图中，单击只读分析引擎上面的“+”进入创建只读分析引擎的页面。



列表视图

1. 在集群列表，找到目标集群，单击**集群 ID**，进入集群管理页面。
2. 在集群管理页面，选择**实例列表**页，单击**只读分析引擎** > **添加只读分析引擎**进入创建只读分析引擎的页面。



2. 只读分析引擎的购买页面与只读实例的购买页面基本一致，可以在引擎处自由选择。详细的填写参数信息如下表所示：

参数	说明
计费模式	在读写实例计算规格的计费模式为包年包月的场景下，只读分析引擎的计费模式支持选择为包年包月或按量计费；若读写实例计算规格的计费模式为按量计费，则只读分析引擎的计费模式仅支持按量计费。
实例引擎	选择当前只读实例的引擎，目前支持 InnoDB 与 LibraDB 引擎，这里选择 LibraDB 引擎。
版本号	选择只读分析引擎实例的版本号，不同版本号的更新说明请参见 LibraDB 引擎内核版本更新动态 。
地域	选择只读分析引擎实例所属的地域。目前仅支持与 TDSQL-C MySQL 版读写实例处于同一地域。

版权所有：腾讯云计算（北京）有限责任公司

可用区	选择只读分析引擎实例所属的可用区。目前支持与 TDSQL-C MySQL 版集群下读写实例或其他只读实例不在同一可用区。得益于 VPC 的跨可用区能力实现，若在跨可用区访问的场景下，网络开销小于等于100ms，具体可选的可用区请以实际页面显示为准。
网络	默认 VPC 与 TDSQL-C MySQL 版集群下的读写实例保持一致，子网为指定可用区的可选子网。同时也支持与读写实例不在同一 VPC 网络内。
安全组	可为只读分析引擎实例选择和对应 TDSQL-C MySQL 版集群不同的安全组，以控制实例访问策略。
自定义端口	可以自行设定访问只读分析引擎实例的端口号，默认为2000，端口取值范围：1024 - 65534。
数据加载方式	指只读分析引擎加载对应 TDSQL-C MySQL 版集群下读写实例中数据的方式。默认为无对象。 • 全加载：指将读写实例中的所有数据对象在只读分析引擎中进行加载（不支持的对象直接跳过）。 • 部分对象：仅加载选中的对象到只读分析引擎中，选择部分对象后需要在下方选择具体加载对象。 • 无对象：所有对象均不加载，仅创建一个空的只读分析引擎，后续需要用户自行设置。
节点数	选择只读分析引擎实例下的节点数量，支持选择1 - 3个。温馨提示，单节点无高可用能力，正式环境建议选择2个及以上节点数。
实例规格	选择只读分析引擎的实例规格。不同的实例规格可以提供不同的性能体验，由于在数据分析或者在复杂 SQL 执行时会消耗较多的计算资源，故只读分析引擎的实例规格都相对较大，同时在高并发数据写入场景下，只读分析引擎若需要确保主从数据一致性，也会消耗一定的资源，请根据业务实际情况进行规格的选择。
硬盘类型	目前仅支持高 IO 型 SSD 硬盘。
硬盘规格	由于只读分析引擎实例当前未使用 TDSQL-C MySQL 版集群的共享存储，故需要单独购买磁盘空间用于存放列存数据，因此，此处硬盘空间需要单独选择，也需要单独进行计费，最小硬盘规格为100GB，最大硬盘规格上限跟所选实例规格有关，请以实际购买页面为准。
实例名	自定义实例名称，支持**创建后命名**或**立即命名**。
数量	根据业务实际情况选择购买数量，最多支持购买6个只读分析引擎。

3. **在完成以上信息配置后，单击立即购买，完成购买后，即可返回集群列表，待实例状态显示为运行中，即可正常使用。**

访问只读分析引擎

最近更新时间：2025-03-26 17:03:42

当“只读分析引擎”开启完成，即可通过任意 MySQL 数据库客户端/JDBC/ODBC 访问至只读分析引擎中。本文将详细列举几种常见的访问方法。

通过只读分析引擎的地址进行访问

您可选择如下任意一种连接方法，通过只读分析引擎的地址进行访问。

连接方法	说明
通过控制台提供的数据库管理平台 DMC 连接集群	通过 DMC 连接集群，简单易用，DMC 提供库表级操作、实时监控、实例会话管理、SQL 窗口、数据管理为一体的数据库管理服务。 **说明：** 仅新版本 DMC 数据库管理平台可以指定访问只读分析引擎。新版本 DMC 数据库管理平台登录地址请参见 登录地址 。
使用内网或外网地址通过 Linux 云服务器连接集群	使用 Linux 云服务器的开发者，可以参考此连接方法。
使用内网或外网地址通过 Windows 云服务器连接集群	使用 Windows 云服务器的开发者，可以参考此连接方法。

通过云联网地址访问数据库

通过云联网地址访问数据库的详细操作可参考 云联网连接 。

通过数据库代理转发 SQL 的方式进行访问

您可以使用任意 MySQL 数据库客户端，通过数据库代理访问至数据库代理中所加入的只读分析引擎。
您可以参考文档 开通数据库代理 开启分析引擎代理。**目前仅1.3.15及最新的数据库代理支持添加只读分析引擎**。如您的数据库代理内核版本低于1.3.15，则无法将只读分析引擎添加至此数据库代理。如需要添加，请参考 升级数据库代理内核小版本 ，将数据库代理版本升级至1.3.15或最新的版本。

- 在1.4.4版本之前的数据库代理仅支持通过 Hint 语法使用 中的 `/* to server 实例 ID */` 语法将 SQL 转发至此数据库代理中的只读分析引擎中执行。
- 在1.4.4版本以及之后的数据库代理支持通过 `/*to ap*/` 的 Hint 语法将 SQL 转发至此数据库代理中的只读分析引擎中执行。如果存在多个只读分析引擎实例，则会根据只读分析引擎的权重进行 SQL 的负载均衡。

注意：

- 如果 `/*to server 实例 ID*/` 语法和 `/*to ap*/` 语法中指定的实例 ID 不存在或只读分析引擎不存在。SQL 将默认转发至当前数据库代理中的存在的只读实例或者读写实例中执行。
- 请注意，如果使用 MySQL 客户端访问至数据库代理。请在连接参数中添加“-c”参数，否则 SQL 中添加的 hint 无法生效。

```
# mysql -h172.16.0.12 -P3306 -utest -p -c -A

mysql> /* to server libradb-ins-hw0af2zi */ show databases;
+--------------------+
| Database           |
+--------------------+
| INFORMATION_SCHEMA |
| Test1              |
| Test2              |
| __cdb_recycle_bin__ |
| haha               |
| libra_system       |
| test               |
+--------------------+
7 rows in set (0.01 sec)
```

在 1.4.4 即以后的版本中支持使用/*to ap*/语法。

```
mysql> select /* to ap */ * from haha.t;
+----+------+
| id | haha |
+----+------+
|  1 |    1 |
|  2 |    2 |
+----+------+
2 rows in set (0.01 sec)
```

支持在 SQL 语句中间添加 hint 的执行方法。

```
mysql> select /* to server libradb-ins-hw0af2zi */  * from haha.t;
+----+------+
| id | haha |
+----+------+
|  1 |    1 |
|  2 |    2 |
+----+------+
2 rows in set (0.01 sec)
```

也支持在添加指定路由 hint 的语法同时添加 SQL 执行计划的语法。

```
mysql> select /* to server libradb-ins-hw0af2zi */  /*+ set_var(libra_enable_late_materialization=1) */ *
from haha.t;
+----+------+
| id | haha |
+----+------+
|  1 |    1 |
|  2 |    2 |
+----+------+
2 rows in set (0.01 sec)
```

基本操作

最近更新时间：2025-01-20 16:21:32

只读分析引擎实例的相关操作与其他的只读实例一致，您可以直接对只读分析引擎实例进行单独操作。本文为您介绍 TDSQL-C MySQL 版集群下的只读分析引擎实例支持的操作。

前提条件

- 准备好只读分析引擎实例，请参见 开启只读分析引擎 。
- 实例状态为运行中。

相关操作指引

查看实例信息

1. 登录 TDSQL-C MySQL 版控制台，在左侧集群列表，单击**目标集群**，进入集群管理页。
2. 在集群详情下找到只读分析引擎实例，可查看只读分析引擎的简略信息。
3. 在集群详情页右侧实例信息下，找到只读分析引擎实例，单击**详情**，即可进入到只读分析引擎实例的详情页面，查看实例详细信息。



修改实例名

和读写实例、只读实例的方法一致，请参考 修改实例名 。

修改安全组

和读写实例、只读实例的方法一致，只是实例类型选择只读分析引擎实例，请参考 为云数据库匹配安全组 。

修改 IP 和端口

和读写实例、只读实例的方法一致，请参考 修改内网地址 。

更改网络

和读写实例、只读实例的方法一致，请参考 更换网络 。

开启或关闭外网

和读写实例、只读实例的方法一致，请参考 开启或关闭外网 。

> **说明：**
> - 开启外网需要授权为内网地址绑定一个外网地址，将其暴露在外网会存在一定的安全风险，因此需要对外授权确认后才可开启。
> - 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。

修改维护时间

和读写实例、只读实例的方法一致，请参考 修改实例维护时间 。

删除和恢复实例

当实例不再使用，可以对实例进行删除，操作请参考 删除实例 。只读分析引擎实例删除后，实例无法访问，但是数据依然会持续从读写节点加载到只读分析引擎。
包年包月实例删除后，实例被移入回收站保留7天，期间实例无法访问。如果您想恢复该实例，可在回收站进行续费恢复，操作请参考 恢复实例 。
按量计费实例删除后，实例被移入回收站保留3天，期间实例无法访问。如您想恢复该实例，可在回收站进行续费恢复，操作请参考 恢复实例 。

调整配置

最近更新时间：2025-04-27 17:58:22

当实例性能无法满足业务变化或实例性能过剩时，可对实例进行调整配置操作，以满足业务诉求或节省成本。

变配能力

TDSQL-C MySQL 版只读分析引擎实例的计算与存储并未分离，所以当遇到资源扩容、磁盘缩容时可能会出现跨服务器变配的场景。当实例变配时，会自动拉起一个新规格的实例节点，待节点数据同步后，自动将新的节点与旧的节点进行访问路由切换，最后完成变配操作。

操作步骤

> **注意：**
> 只读分析引擎实例目前默认采用从集群下的读写实例拉取数据的方式进行节点创建，故变配操作时，建议在业务低峰期进行。

1. 登录 TDSQL-C MySQL 版控制台，参考以下不同视图模式的操作，进入调整配置页面。

页签视图

1. 在左侧集群列表，单击**目标集群**，进入集群管理页。
2. 在集群详情下，在右侧的实例信息内找到目标只读分析引擎实例，单击**调整配置**。



列表视图

1. 在集群列表中，单击**目标集群 ID**，进入集群管理页。
2. 在集群管理页，单击**实例列表 > 只读分析引擎**，然后在列表中找到目标只读分析引擎实例，单击其操作列的**更多 > 调整配置**，进入调整配置页面。



2. 在调整配置页面中，根据需要调整实例的计算规格和硬盘大小，选择操作时间后，单击**立即购买**。



> **说明：**
> 调整配置会产生一次业务闪断，请根据实际业务需要选择变配切换的操作时间。
> - 选择升级完成时，会在完成配置变更后，立即触发闪断切换。
> - 选择维护时间内，会在实例的维护时间内进行闪断切换。修改实例维护时间可参见 修改实例维护时间 。

多节点高可用

最近更新时间：2025-06-27 16:17:11

数据库高可用（High Availability, HA）旨在通过冗余设计和故障自动恢复机制，确保系统在部分节点失效时仍能持续提供服务。只读分析引擎实例通过多节点部署的方式实现高可用，通过将数据库实例分布在多个物理节点上，并结合数据同步、故障检测和流量切换等机制，保障业务连续性。本文为您介绍只读分析引擎实例实现高可用的原理以及相关操作和管理。

只读分析引擎实例高可用实现原理

冗余架构

一个只读分析引擎实例下支持部署1 - 3个节点，当节点数量为2个或3个时，可以形成对等集群（Cluster）。当只读分析引擎实例下的某一个节点发生故障时，系统可自动将 SQL 流量转发至“存活”的节点，即转发至该只读分析引擎实例下其他正常运行的节点，避免单点故障导致业务不可用。

故障检测与切换（Failover）

- **心跳机制**：在只读分析引擎实例的节点之间，通过周期性心跳检测来探测节点的“存活”状态，若节点超时未响应（检测间隔时间为3秒），则会触发故障判定。
- **自动切换**：当周期性心跳检测到节点故障时，系统将再次对节点进行两次无间隔时间的“存活”探测（探测时间约10秒）。如果持续探活节点失败，则系统将在1 - 5秒内将故障节点移出只读分析引擎实例，然后等待故障节点恢复或者自动重建后再加入到该只读分析引擎实例中。

负载均衡

只读分析引擎实例中的多个节点使用负载均衡的形式为您提供服务。所有节点都可以正常承载业务流量，提升业务的并发执行性能。

> **注意：**
> 目前节点的负载均衡能力是连接级别的负载均衡。如果业务均为长连接，则建议在完成多节点设置后，全部重新访问数据库，以使连接更加均衡。

计费说明

只读分析引擎实例根据节点数量的不同，分为单节点、多节点，且多节点的定价与单节点的定价不同，详细说明请参见 产品价格 。

调整节点数量

> **注意：**
> 只读分析引擎实例目前默认采用从集群下的读写实例拉取数据的方式进行节点重建，故新增节点操作时，建议在业务低峰期进行。

1. 登录 TDSQL-C MySQL 版控制台，参考以下不同视图模式的操作，进入只读分析引擎实例详情页面。

页签视图

1. 在左侧集群列表，单击**目标集群**，进入集群管理页。
2. 在集群详情下的拓扑图中，单击只读分析引擎，并单击**详情**进入只读分析引擎实例的详情页面。



列表视图

1. 在集群列表中，单击**目标集群 ID**，进入集群管理页。
2. 在集群管理页，单击**实例列表** > **只读分析引擎**，然后单击目标只读分析引擎实例的实例 ID，进入实例详情页面。



2. 在实例详情页的**节点信息**后，单击**节点变更**。
3. 在弹窗下，根据需要新增或删除节点，确认费用后单击**确定**。



- 新增节点：如需增加节点，请单击**新增节点**，可为当前只读分析引擎实例新增1个节点，一个只读分析引擎实例下最多支持3个节点。
- 删除：如需删除节点，请单击**删除**，若操作失误删除了错误的节点，可在单击**确定**之前，单击**恢复**。

> **说明：**
> - 节点的规格和可用区与只读分析引擎实例一致。
> - 一个只读分析引擎实例下至少需保留1个节点，不支持全部删除。

4. 完成操作后即可返回集群列表，当节点状态显示为“运行中”时，即可正常使用。

监控告警

最近更新时间：2025-04-27 17:58:22

通过 TDSQL-C MySQL 版的监控功能可以完整的查看只读分析引擎实例的运行信息。查看监控指标有多种方法，如通过 TDSQL-C MySQL 版控制台查看、云 API 查看、腾讯云可观测平台的 dashboard 查看。本文为您介绍只读分析引擎实例支持的监控指标。

前提条件

- 准备好只读分析引擎实例，请参见 开启只读分析引擎 。
- 实例状态为运行中。

查看监控

只读分析引擎实例查看监控的方法与 TXSQL 引擎实例的操作方法基本一致，详细操作请参考 查看监控 。

> **注意：**
> 因为只读分析引擎实例的监控指标与读写实例和普通只读实例的指标不一致，所以不支持只读分析引擎实例与读写实例和只读实例在一个监控图表中联合查看。

支持的监控指标

类别	监控指标（中文）	监控指标（英文）	单位	数据聚合方式
分析引擎汇总监控	分析引擎 QPS	LibraDBQps	Count/s	max
	分析引擎总连接数	LibraDBConnections	Count	max/last
	分析引擎错误请求数	LibraDBErrorQueries	Count	sum
	分析引擎活跃连接数	LibraDBActiveConnections	Count	max/last
	分析引擎返回结果集大小	LibraDBResultSize	MBytes	sum
	分析引擎数据延迟时间	LibraDBLatency	s	max
	分析引擎每秒同步处理行数	LibraDBSinkRowsPerSec	Count/s	max/avg
	分析引擎全量加载时每秒写行数	LibraDBDumpRowsPerSec	Count/s	max/avg
	分析引擎全量加载时每秒写数据大小	LibraDBDumpSizePerSec	MBytes/s	max/avg
	分析引擎平均响应时间	LibraDBDuration	Ms	avg
分析引擎节点监控	分析引擎 CPU 使用率	LibraDBCpuUseRate	%	last
	分析引擎内存使用率	LibraDBMemUseRate	%	last
	分析引擎存储使用率	LibraDBStorageUseRate	%	last
	分析引擎存储使用量	LibraDBStorageUsed	MBytes	last
	分析引擎读吞吐	LibraDBReadThroughput	MBytes/s	max
	分析引擎写吞吐	LibraDBWriteThroughput	MBytes/s	max
	分析引擎网络入流量	LibraDBBytesReceived	MBytes	max/avg
	分析引擎网络出流量	LibraDBBytesSent	MBytes	max/avg
	分析引擎 CompactRowset 数量	LibraDBCompactRowsets	Count	last

	分析引擎接收到的 MppTask 数	LibraDBMPPTasks	Count	sum
	分析引擎正在运行的 MppTask 数	LibraDBRunningMPPTasks	Count	last
	分析引擎数据文件大小	LibraDBDatafileSize	MBytes	last
	分析引擎缓存使用量	LibraDBCached	MBytes	last
	分析引擎缓存命中率	LibraDBCacheHitsRate	%	avg
	分析引擎节点延迟时间	LibraDBNodeLatency	s	max
	分析引擎节点每秒同步处理行数	LibraDBNodeSinkRowsPerSec	Count/s	max
	分析引擎节点全量加载时每秒写行数	LibraDBNodeDumpRowsPerSec	Count/s	max
	分析引擎节点全量加载时每秒写数据大小	LibraDBNodeDumpSizePerSec	MBytes/s	max

告警设置

只读分析引擎实例的告警配置方法与 TDSQL-C MySQL 版读写实例和只读实例的方法一致，详细操作指导可参考 设置告警流程简介 。

管理参数

最近更新时间：2025-05-26 09:28:11

本文为您介绍只读分析引擎实例支持修改的参数。

前提条件

- 准备好只读分析引擎实例，请参见 开启只读分析引擎 。
- 实例状态为运行中。

管理参数

支持对只读分析引擎实例的参数进行管理，包括修改参数、参数模板的应用、查看参数修改记录等，操作方法与 TXSQL 引擎的只读实例基本一致，请参见 设置实例参数 。

参数列表

下表为只读分析引擎实例支持修改的参数列表与参数修改建议：

常用参数

参数名	默认值	是否重启	参数修改建议
sql_mode	与读写实例保持一致	否	此参数控制 SQL 在只读分析引擎的执行行为，参数值的行为逻辑与 MySQL 保持一致。但存在部分 SQL_MODE 不兼容或可配置但无法生效，详情请参考文档：兼容性与使用限制 的 SQL_MODE。
mpp-time-out	3600	是	此参数控制 SQL 在只读分析引擎中执行的超时时间。如 SQL 执行超出此时间，则会超时报错。可根据业务实际情况进行此参数的调整。
max_connections	1024	是	此参数控制只读分析引擎的最大连接数，超出此连接数配置时，新增的连接将会报错。建议最大连接数根据业务实际的要求进行配置。因只读分析引擎主要承载业务中的复杂 SQL，也不建议新增过多链接，维持到200以下即可。
max_threads	16	否	此参数控制 SQL 在只读分析引擎中最大可使用的线程数。默认与实例规格的最大 CPU 保持一致，建议修改为实例规格的 CPU/2。
libratree_enable_check_disk_space_available	ON	是	此参数控制是否开启磁盘空间保留功能。默认控制用户只能使用到当前磁盘实例规格的90%。达到90%的磁盘空间后，会暂停只读分析引擎的数据同步与加载。
libratree_reserve_space_pcnt	10	是	此参数控制磁盘空间保留比例，默认值为10，即保留10%的空间不被数据占用。参数值设置范围：0 - 100。
libra_preserve_order_for_pagination	OFF	否	此参数控制是否全局开启分页保序功能，详细功能介绍请参考 分页保序功能 。

场景参数：

参数名	默认值	是否重启	参数修改建议
replication_sink_parallelism	16	否	修改此参数可提升在数据增量加载时的并发数量。可提升增量数据加载的速度，但会增加对读写实例的压力。请在读写实例资源比较富余的时候调整此参数。
replication_dumper_split_chunk_parallel	1	否	通过此参数可控制拆分 chunk 的并发度。并发度越高，全量加载的速度越快。但是调高此值会增加对读写实例的压力。请在读写实例资源比较富余的时候调整此参数。
replication_dumper_table_parallel	4	否	通过此参数可控制同时启动数据全量加载的表数量。在读写实例的资源充足的情况下，同时启动的表数量越多，全量数据加载的速度越快。请在读写实例资源比较富余的时候调整此参数。

版权所有：腾讯云计算（北京）有限责任公司

replication_dumper_insert_parallel	4	否	通过此参数可控制在全量数据加载时写入只读分析引擎的并发度。在只读分析引擎资源充足的情况下，写入并发数量越高，全量数据加载的速度越快。请在“只读分析引擎”资源较富余的时候调整此参数。
replication_dumper_select_parallel	16	否	通过此参数可控制在全量数据加载时有几个并发同时读取表数据。在读写实例的资源充足的情况下，同时启动的并发数量越多，全量加载的速度就越快。请在读写实例资源比较富余的时候调整此参数。
replication_dumper_chunk_size	10000	否	每一次加载全量数据时从读写实例读取的 chunk 大小。chunk 越大，全量数据加载的速度越快，但是对读写实例的资源消耗也更大一些。请根据读写实例的资源使用实际情况调整此参数。
libra_runtime_filter_type	MIN_MAX,BLOOM_FILTER,IN_FILTER	否	控制 Runtime Filter 功能可使用的 Runtime Filter 类型。详细介绍可参考：Runtime Filter 使用手册 。
libra_enable_runtime_filter	ON	否	控制是否开启/关闭 Runtime Filter 功能。详细介绍可参考：Runtime Filter 使用手册 。
libra_enable_late_materialization	ON	否	控制是否开启/关闭延迟物化功能。详细介绍可参考 延迟物化 。
null_for_aggregation_by_empty_set	ON	否	此参数控制在当 Group By 遇到空结果集时候，是否返回空值。如果参数开启，则就会对空值进行 Group By。如果参数关闭，则会跳过空值。

数据加载
加载数据到只读分析引擎

最近更新时间：2025-04-28 17:48:12

本文为您介绍如何加载数据到只读分析引擎中，以供用户进行数据分析查询。

> **说明：**
> 目前在只读分析引擎中仅支持加载库、表、视图。

前提条件

- 准备好实例形态为预置资源的集群，请参见 创建集群 。
- 集群状态为运行中。

操作步骤

目前只读分析引擎支持多种方式加载数据，详细的操作方法如下。

> **注意：**
> - 开启只读分析引擎，会默认为数据库创建一个名为“txsqlPlugins_analytics_repl”的账号，此账号为只读分析引擎从读写实例拉取数据使用。请不要删除此账号。
> - 在初次加载数据到只读分析引擎时，会从读写实例拉取数据，从而产生慢 SQL。请忽略“txsqlPlugins_analytics_repl”账号产生的慢 SQL 信息。

方法1：创建只读分析引擎时加载数据

在创建只读分析引擎实例的时候，可以选择数据加载的方式。目前支持3种选项，分别是全加载、部分对象以及无对象，三种不同的数据加载方式对应不同的加载方案。

1. 数据加载方式为全加载。

   全加载模式下，TDSQL-C MySQL 版集群中的所有库表都将自动同步至只读分析引擎中，由于可能存在部分库表对象不满足加载条件的情况，这些不支持同步的对象会自动被忽略，不会加载到只读分析引擎中。
2. 数据加载方式为部分对象。

   若数据加载方式选择为部分对象，则需要在创建只读分析引擎实例时，指定需要加载到分析引擎中的对象。若存在不满足加载条件的对象，会忽略加载。若视图引用的表没有被加载到只读分析引擎，则在执行视图加载时，会报错提示表不存在。
3. 数据加载方式为无对象。

   若数据加载方式选择为无对象，则指在创建只读分析引擎实例时不加载任何对象。需要用户自行在实例运行过程中去手动指定需要加载的对象，请参见 方式2：手动加载 。

方法2：手动加载

若在创建只读分析引擎实例创建时选择无对象，或者需要修改加载到只读分析引擎的对象信息，可使用如下方法。

通过控制台加载对象

1. 登录 TDSQL-C MySQL 版控制台，在左侧集群列表，单击目标集群，进入集群管理页。
2. 在集群详情页下，找到需要修改加载对象的只读分析引擎实例，单击**详情**，进入只读分析引擎实例详情页面。
3. 单击**修改对象**，在修改对象对话框中，**选择**或者**取消**需要加载到只读分析引擎的对象。


详情信息
实例详情
基本信息
实例名
rc
实例 ID
libradb
地域/可用区
华北地区（北京）/北京六区
计费方式
按量计费
只读内网地址
主机:
6 端口: 2000
只读外网地址
开启
实例规格
通用型 4核/16GB
硬盘规格
高IO型SSD盘 100GB存储空间
分析引擎版本
1.2404.7
创建时间
2024-09-05 09:25:28
所属网络
更换网络
维护周期
星期一、星期二、星期三、星期四、星期五、星期六、星期日
维护时间
03:00-04:00
对象信息
修改对象
数据库名
对象名
状态
数据加载情况
增量阶段延迟
其他信息

查看数据加载状态

最近更新时间：2025-02-11 16:41:53

操作场景

加载数据到只读分析引擎会存在两个不同的阶段，首先是数据全量加载阶段，其次是实时数据加载阶段。当数据加载到只读分析引擎中时，只有在完成数据对象构建后才能进行数据查询。您可以使用如下的方法查看数据表加载的状态。

前提条件

- 准备好实例形态为预置资源的集群，请参见 创建集群 。
- 集群状态为运行中。
- 已 加载数据到只读分析引擎 。

操作步骤

方法1：控制台查看数据加载状态

1. 登录 TDSQL-C MySQL 版控制台，在左侧集群列表，单击**目标集群**，进入集群管理页。
2. 在集群详情页下，找到需要修改加载对象的只读分析引擎实例，单击**详情**，进入只读分析引擎实例详情页面。
3. 在只读分析引擎实例的详情页面中，可查看到对象信息，可通过对象信息查看到数据加载状态。



参数	说明
数据库名	代表此对象所属的数据库名称。支持通过数据库名进行筛选。 **注意：** 如果加载到只读分析引擎的数据库较多时，筛选数据库可能会导致控制台加载时间过长。
对象名	代表加载到只读分析引擎的表名，视图同样可以加载到只读分析引擎，但是在此处不会显示视图信息。如需查看视图，请使用 MySQL 命令进行查看。 当加载的对象较多时，可在控制台中通过对象名称的关键字进行搜索。
状态	状态包括“数据加载中”、“实时增量中”、“同步异常”。
数据加载情况	若对象正在数据加载中，则数据加载情况会显示当前表的加载进度。
增量阶段延迟	若对象正在实时增量中，则会显示此对象的同步延时信息。
其他信息	若数据加载出现异常，则其他信息中会显示此对象加载异常的相关信息。

方法2：通过 SQL 语句查看数据加载状态

```
show replication tables;
或者
select * from information_schema.libra_table_status;
```

在只读分析引擎中，执行此命令后可查看当前所有在加载中的数据对象的情况。可支持通过 where 条件过滤。
字段显示信息如下图所示：

```
MySQL [(none)]>  select * from INFORMATION_SCHEMA.LIBRA_TABLE_STATUS;
+---------------+------------+------------+-----------+-------------------+------+---------+
| DATABASE_NAME | TABLE_NAME | REPLICA_ID | AVAILABLE | REPLICATION_STEP  | LAG  | MESSAGE |
+---------------+------------+------------+-----------+-------------------+------+---------+
| perf_1        | sbtest3    | r1         |         1 | Change Propagation |    0 |         |
| perf_1        | sbtest5    | r1         |         1 | Change Propagation |    0 |         |
| perf_1        | sbtest10   | r1         |         1 | Change Propagation |    0 |         |
| perf_1        | sbtest8    | r1         |         1 | Change Propagation |    0 |         |
| perf_1        | sbtest2    | r1         |         1 | Change Propagation |    0 |         |
| perf_1        | sbtest1    | r1         |         1 | Change Propagation |    0 |         |
| perf_1        | sbtest7    | r1         |         1 | Change Propagation |    0 |         |
| perf_1        | sbtest9    | r1         |         1 | Change Propagation |    0 |         |
| perf_1        | sbtest6    | r1         |         1 | Change Propagation |    0 |         |
| perf_1        | sbtest4    | r1         |         1 | Change Propagation |    0 |         |
+---------------+------------+------------+-----------+-------------------+------+---------+
10 rows in set (0.00 sec)
```

版权所有：腾讯云计算（北京）有限责任公司

控制数据加载速度

最近更新时间：2025-01-20 16:21:32

操作场景

在进行数据全量加载的时候，会对 TDSQL-C MySQL 版的读写实例造成一定的资源开销，特别是在数据量较大的集群中做一次性的数据加载时，此操作针对一些资源比较紧张的实例，在一定程度上会影响读写实例的性能稳定性。可通过如下几种方式去控制在数据加载过程中对 TDSQL-C MySQL 版读写实例的影响，降低对源端的资源消耗。但是**此类操作会导致数据从行存加载为列式存储的效率降低**，所以请根据业务实际情况进行调整。

前提条件

- 准备好实例形态为预置资源的集群，请参见 创建集群 。
- 集群状态为运行中。
- 已 加载数据到只读分析引擎 。

控制数据加载速度

- **调整全量数据加载场景下，单表数据读出并发数量**

  通过调整参数功能，修改 replication_dumper_select_parallel 参数。此参数代表同时可对同一个对象进行数据读出的并发度。此值越大，对表的数据读取范围更大，在资源充足场景下，加载效率则会更高。如果资源不足，建议将此值设置得更小。操作请参见 设置实例参数 。
- **调整全量数据加载场景下，单表数据读出行数**

  通过调整参数功能，修改 replication_dumper_chunk_size 参数。此参数代表每一次对同一个对象进行数据单次读出的数据行数。此值越大，代表每一次读出的数据越多。在资源充足场景下，加载效率则会越高。如果资源不足，建议将此值设置得更小。操作请参见 设置实例参数 。
- **调整全量数据加载场景下，总的数据读出并发**

  通过调整参数功能，修改 replication_dumper_table_parallel 参数。此参数代表同时可对多少张表同时进行数据读出，此值越大，同时可读出数据的表数量越多，在资源充足的情况下，数据加载的效率就更快。操作请参见 设置实例参数 。

版权所有：腾讯云计算（北京）有限责任公司

取消部分对象加载到只读分析引擎

最近更新时间：2025-05-21 17:47:22

操作场景

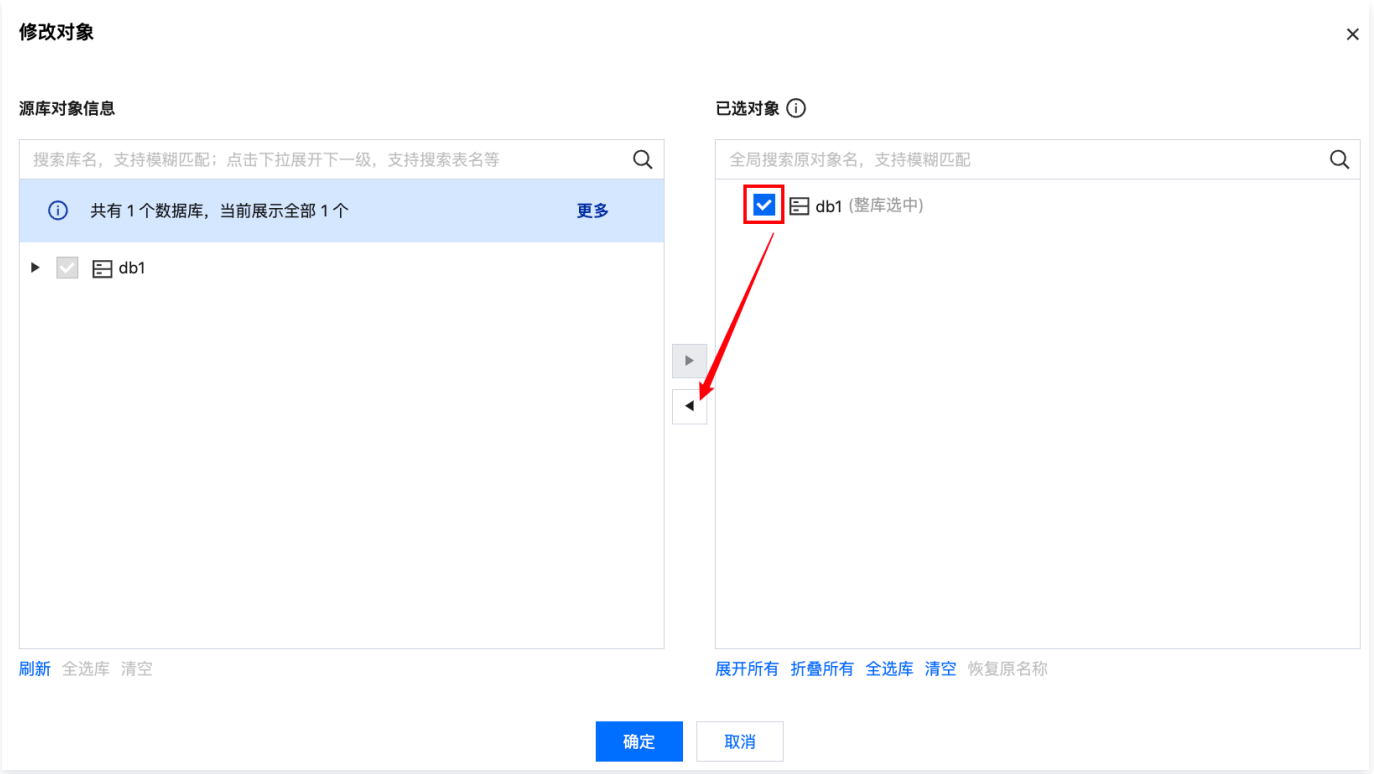
在加载数据到只读分析引擎的过程中，需要从只读分析引擎中取消某一些对象加载的场景。

前提条件

- 准备好实例形态为预置资源的集群，请参见 创建集群 。
- 集群状态为运行中。
- 已 加载数据到只读分析引擎 。

操作步骤

1. 登录 TDSQL-C MySQL 版控制台，在左侧集群列表，单击目标集群，进入集群管理页。
2. 在集群详情页下，找到需要修改加载对象的只读分析引擎实例，单击**详情**，进入只读分析引擎实例详情页面。
3. 单击**修改对象**，在修改对象对话框中，在右侧的同步对象中取消已经加载到只读分析引擎的对象。



执行查询
收集统计信息

最近更新时间：2025-06-27 16:17:11

对于只读分析引擎而言，想要 SQL 执行结果好，则需要对表对象进行数据的基本统计信息收集。完成了统计信息收集的表，系统才会更精准的为用户执行的 SQL 进行合理的执行计划推荐。

统计信息概述

统计信息（Statistics）是关于数据库对象的数据特征和分布的信息。数据库可以利用统计信息来计算出更准确的计划代价，优化计划的选择。统计信息的类型包含表的行数、列的不同值个数（number of distinct，简称 NDV）、列的直方图等信息。统计信息的收集涉及对数据库对象的扫描与获取，并将所收集的统计信息存储于系统的数据字典，此外，系统还会在本地维护统计信息的缓存，以提升优化器访问统计信息的效率。同时，在收集使用统计信息时，需要注意以下两点：

- 统计信息的准确性对于优化器的决策非常重要。因此，定期更新和维护统计信息是保持数据库性能的关键。
- 统计信息的收集可能会对数据库的性能产生一定的影响，因为它涉及扫描和分析数据库对象。因此，在收集统计信息时需要权衡性能和统计信息的准确性。

总之，统计信息在数据库管理系统中扮演着至关重要的角色，它们提供了关于数据库对象的关键信息，协助优化器作出更为明智的决策，进而提升查询性能及整体数据库系统的效能。

> **说明：**
> 在2.2410.1.0及之后的版本中支持了随机采样功能，该功能可以按照 TableScan 算子指定的采样率进行随机采样。同时可支持基于 Block 级别的采样，并且可以根据设置的比例进行统计信息收集，能够极大地降低在收集统计信息过程中因全表扫描而产生的开销。

表的统计信息查询

```
SHOW TABLE STATS [ShowLikeOrWhere];
```

查询结果字段说明如下：

列名	说明
db_name	数据库的名字。
table_name	表的名字。
partition_name	分区的名字。
update_time	收集更新时间。
modify_count	表涉及修改的行数。
row_count	表的总行数。
sample_size	收集基础统计信息采样的行数。

查询 test 库中 t1 的表的统计信息如下示例：

```
SHOW TABLE STATS WHERE db_name='test' AND table_name='t1';
```

统计信息收集

目前只读分析引擎支持两种不同的统计信息收集方式：

- **自动收集**
  在2.2410.1.0及之后的版本中，只读分析引擎内核支持了根据数据加载为列存时进行动态采样和统计信息自动收集的能力。用户无需手动收集即可生成对应库表的统计信息。

> **注意：**

> 默认自动收集统计信息不包含表的直方图信息，如需收集表的直方图信息，请进行手动收集。

- **手动收集**

  如果需要手动获取当前表的最新统计信息，可以手动通过 MySQL 客户端登录至只读分析引擎实例中，执行如下语句即可收集指定表对象的统计信息：

```
ANALYZE TABLE <表名> [ ALL COLUMNS| COLUMNS IdentList [WITH NUM BUCKETS]];
```

- ALL COLUMNS 指定所有列都按照指定行为收集统计信息，比如指定所有列都收集直方图。
- 不指定 ALL COLUMNS，则使用默认收集策略，即收集列的基础统计信息，不包含直方图信息。
- COLUMNS IdentList 代表需要收集指定列的相关统计信息。如 `COLUMN c1,c2,c3` 。
- NUM BUCKETS 指定等高直方图或者 TOPN 收集的桶个数，其中范围为[1，1024]，其中指定桶个数为1则表示不收集等高直方图或者 TOPN。

使用示例

- 使用默认策略收集 test 库中 t1 的统计信息：

```
ANALYZE TABLE t1;
```

- 指定收集 test 库中 t1 的所有列的直方图统计信息：

```
ANALYZE TABLE t1 ALL COLUMNS WITH 256 BUCKETS;
```

- 指定收集 test 库中 t1 的 c1，c2列统计信息直方图：

```
ANALYZE TABLE t1 COLUMNS c1,c2 WITH 256 BUCKETS;
```

手动指定 database 级别的统计信息收集

如果表的数量较多时，可以直接对 database 进行统计信息收集。收集策略的含义同表级相同，收集原则是尽可能的将整个 database 下的表都收集完成，如果某个表收集失败，不会影响其他表的收集。

```
ANALYZE DATABASE DBNameList [ ALL COLUMNS| COLUMNS IdentList [WITH NUM BUCKETS]];
```

SELECT 语句说明

最近更新时间：2024-08-06 16:49:11

本文为您介绍在只读分析引擎中，支持使用的查询语句及相关说明。

对象权限说明

在只读分析引擎中，只有已经完全加载到分析引擎中的对象可以进行数据查询。查询数据需要拥有对应数据表的 SELECT 权限，若无权限，请通过 TDSQL-C MySQL 版账号管理进行对象权限赋予，操作请参见 修改账号权限 。

SELECT 语句

在只读分析引擎中主要支持 SELECT 语句进行数据查询。

```
SELECT
    [ALL | DISTINCT | DISTINCTROW ]
        select_expr [, select_expr] ...
    [FROM table_references
    [WHERE where_condition]
    [GROUP BY {col_name | expr | position}]
    [HAVING where_condition]
    [WINDOW window_name AS (window_spec)
        [, window_name AS (window_spec)] ...]
    [ORDER BY {col_name | expr | position}
      [ASC | DESC]]
    [LIMIT {[offset,] row_count | row_count OFFSET offset}]
```

如上 SELECT 语句块为只读分析引擎中支持的 SELECT 语句，下文描述介绍了 MySQL 的完整的 SELECT 语句段中**在只读分析引擎中不支持的语句单元**：

- HIGH_PRIORITY 表示设置执行 SELECT 语句的优先级，这在只读分析引擎中不会生效。但若 SQL 语句中包含此关键字，也不会影响执行结果。
- STRAIGHT_JOIN 表示强制优化器按照子句中列出的顺序进行表连接，但是在只读分析引擎中完全基于只读分析引擎自身的列式执行优化器进行执行计划的选择，故此关键字在只读分析引擎中不会生效。
- SQL_SMALL_RESULT、SQL_BIT_RESULT、SQL_BUFFER_RESULT、SQL_NO_CACHE、SQL_CALC_FOUND_ROWS 这几个查询选项也不会在只读分析引擎中生效，但不会影响查询语句的执行。
- PARTITION partition_list 子句在只读分析引擎中不支持，且此选项会导致查询执行失败。
- 不支持带 WITH ROLLUP 语法。如果 SQL 语句中包含此关键字，则执行查询会报错。
- 不支持 FOR UPDATE | Lock IN SHARE MODE 等子句语法，若执行 SQL 中包含此类子句，则不会生效，SQL 执行正常。
- 不支持 into option。若语句中包含此关键字，则执行会报错。

版权所有：腾讯云计算（北京）有限责任公司

CTE 语法使用说明

最近更新时间：2025-02-11 16:41:53

TDSQL-C MySQL 版在5.7版本中无法支持 CTE 语法。仅8.0版本后的行存引擎才支持 CTE 语法，但对于只读分析引擎而言，无论是5.7版本还是8.0版本，在只读分析引擎中均能正常支持 CTE 语法。

支持情况

- **1.2404.x**
  在1.2404.x 版本中，只读分析引擎仅支持 Non-Recursive CTE，使用时需要添加 `Hint /*+ MERGE() */`，否则可能无法生成正确的执行计划。在后续版本中，只读分析引擎将逐步支持 Recursive CTE，并且会优化 CTE 的执行性能。
- **2.2410.x**
  在2.2410.x 版本中，只读分析引擎仍然仅支持 Non-Recursive CTE。但是当前版本执行 CTE 语法时已经无需添加 `Hint /*+ MERGE() */` 即可正常执行。同时在2.2410.x 版本中，也支持了流式执行的方式，大幅度优化了 CTE 在执行时的性能。

CTE 简介

公共表达式（Common Table Expressions，CTE）是 SQL 标准的一部分，通常被称为“WITH 子句”。CTE 在 SQL：1999标准首次引入，旨在提供一种简洁而强大的方法来定义临时结果集。这些结果集可以在单个 SQL 语句中被多次引用，大大提高了查询的可读性和可维护性。
WITH 子句使用示例：

```
-- Start defining CTE.
WITH CustomerCTE AS (
    SELECT customer_id, first_name, last_name, email_address
    FROM customer
)
-- End defining CTE.
SELECT   *
FROM CustomerCTE; -- Reference the CTE.
```

CTE 的优势

在复杂的 SQL 查询中，CTE 提供了许多优势。

- **简化查询**：CTE 可以组织和简化复杂的 SQL 语句，提高可维护性。例如，在需要多次引用相同子查询的情况下，CTE 可以避免重复代码，从而使查询更加清晰。
- **提高代码可读性**：使用有意义的名称来表示中间结果，使 SQL 更易理解。
- **避免重复查询**：CTE 允许定义临时的结果集，这些结果集可以在一条 SQL 中多次引用，从而避免了相同操作的重复执行。
- **递归查询**：CTE 支持递归查询，能够处理层次结构数据（如员工组织结构）的查询需求。在处理树状结构数据时非常有用。

语法结构

CTE 的语法结构如下。

```
with_clause:
    WITH [RECURSIVE]
        cte_name [(col_name [, col_name] ...)] AS (subquery)
        [, cte_name [(col_name [, col_name] ...)] AS (subquery)] ...
```

参数项	描述
WITH 关键字	表示 CTE 定义的开始。
[RECURSIVE]	可选关键字，如果包含 RECURSIVE，表示允许在 CTE 中引用自身，用于创建递归查询。
cte_name	为 CTE 指定的名称，可以在后续的查询中被引用。
[(col_name [，col_name] ...)]	可选的列名列表，为 CTE 的结果集指定列名。如果省略，将使用子查询中的列名。

版权所有：腾讯云计算（北京）有限责任公司

AS (subquery)	CTE 内部的子查询，定义 CTE 的内容。
逗号和额外的 CTEs	在一个 WITH 子句中，可以定义多个 CTEs，用逗号分隔。每个额外的 CTE 都遵循相同的结构：cte_name [(col_name ...)] AS (subquery)。

非递归 CTE（Non-Recursive CTE）

在非递归 CTE 中，CTE 仅引用其他表或者之前定义的 CTE，而不会引用自身。它适用于多步查询的分解，通过中间层次计算逐步构建最终查询结果。

```
WITH
    cte1 AS (SELECT * FROM t1, t2),
    cte2 AS (SELECT i1, i2 FROM cte1 WHERE i3 > 10)
    cte3 AS (SELECT * FROM cte2, t3 WHERE cte2.i1 = t3.i1)
SELECT * FROM cte3;
```

递归 CTE（Recursive CTE）

在递归 CTE 中，CTE 会引用自身。Recursive CTE 常用于处理树结构或图结构的数据查询，例如计算阶乘，生成序列或遍历层级关系。
Recursive CTE 由 Seed Part Subquery，Union Type，Recursive Part Subquery 三部分组成。Seed Part Subquery 不引用自身，Recursive Part Subquery 一定引用自身。

```
WITH RECURSIVE cte(n, fact) AS (
    SELECT 0, 1 -- Seed Part Subquery
    UNION ALL -- Union Type
    SELECT n + 1, (n + 1) * fact FROM cte WHERE n < 5 -- Recursive Part Subquery
)
SELECT n, fact FROM cte;
```

示例

- **计算阶乘**

  ```
  WITH RECURSIVE cte(n, fact) AS (
      SELECT 0, 1
      UNION ALL
      SELECT n + 1, (n + 1) * fact FROM cte WHERE n < 5
  )
  SELECT n, fact FROM cte;
  ```

  在这个示例中，递归部分 `UNION ALL SELECT n + 1, (n + 1) * fact FROM cte WHERE n < 5` 会重复调用自身，直到 n 达到5，递归部分输出空行时，结束递归。

- **遍历树结构**

  假设我们有一张表 employees，包含员工的层级关系，其中 id 是员工的唯一标识，name 是员工姓名，manager_id 是该员工的上级员工 id。

  ```
  CREATE TABLE employees (
      id INT PRIMARY KEY,
      name VARCHAR(100),
      manager_id INT
  );

  INSERT INTO employees (id, name, manager_id) VALUES
  (1, 'CEO', NULL),
  (2, 'Manager 1', 1),
  (3, 'Manager 2', 1),
  (4, 'Employee 1', 2),
  (5, 'Employee 2', 2),
  (6, 'Employee 3', 3);
  ```

  递归 CTE 用于遍历员工层次结构，从上到下获取所有下属：

```
WITH RECURSIVE employee_hierarchy AS (
    -- 基础情况：从 CEO 开始
    SELECT
        id,
        name,
        manager_id,
        1 AS level
    FROM employees
    WHERE manager_id IS NULL

    UNION ALL

    -- 递归情况：找到每个员工的下属
    SELECT
        e.id,
        e.name,
        e.manager_id,
        eh.level + 1
    FROM employees e
    INNER JOIN employee_hierarchy eh ON eh.id = e.manager_id
)
SELECT id, name, manager_id, level
FROM employee_hierarchy
ORDER BY level, manager_id;

-- Result
┌────────┬──────────────┬────────────┬────────┐
│  id    │     name     │ manager_id │ level  │
│ int32  │   varchar    │   int32    │ int32  │
├────────┼──────────────┼────────────┼────────┤
│      1 │ CEO          │            │      1 │
│      2 │ Manager 1    │          1 │      2 │
│      3 │ Manager 2    │          1 │      2 │
│      4 │ Employee 1   │          2 │      3 │
│      5 │ Employee 2   │          2 │      3 │
│      6 │ Employee 3   │          3 │      3 │
└────────┴──────────────┴────────────┴────────┘
```

基础 CTE

```
WITH CustomerCTE AS (
    SELECT customer_id, first_name, last_name, email_address
    FROM customer
)
SELECT /*+ MERGE() */ *
FROM CustomerCTE;
```

多个 CTE

```
WITH
CTE1 AS (
    SELECT customer_id, first_name, last_name, email_address
    FROM customer
),
CTE2 AS (
    SELECT ss_item_sk, ss_customer_sk, ss_sold_date_sk, ss_sales_price
    FROM store_sales
)
```

```
SELECT /*+ MERGE() */ CTE1.first_name, CTE1.last_name, CTE2.ss_sales_price
FROM CTE1
JOIN CTE2 ON CTE1.customer_id = CTE2.ss_customer_sk;
```

- 定义了两个 CTE：CTE1 和 CTE2。
- 在最终查询中使用这两个 CTE 的结果集进行 Join 操作。

执行结果：

```
+------------+-----------+----------------+
| first_name | last_name | ss_sales_price |
+------------+-----------+----------------+
| John       | Doe       |          45.99 |
| Jane       | Smith     |          32.50 |
| Michael    | Johnson   |          78.25 |
| Emily      | Brown     |          19.99 |
| David      | Wilson    |          55.00 |
| John       | Doe       |          67.75 |
| Jane       | Smith     |          22.99 |
| Michael    | Johnson   |          41.50 |
| Emily      | Brown     |          89.99 |
| David      | Wilson    |          33.25 |
+------------+-----------+----------------+
10 rows in set (0.12 sec)
```

嵌套 CTE

```
WITH SalesSummary AS (
    SELECT ss_customer_sk, SUM(ss_net_paid) AS total_spent
    FROM store_sales
    GROUP BY ss_customer_sk
),
TopCustomers AS (
    SELECT ss_customer_sk, total_spent
    FROM SalesSummary
    WHERE total_spent > 1000  -- 假设设置一个阈值，例如消费超过 1000 的顾客
),
CustomerDetails AS (
    SELECT c.customer_id, c.first_name, c.last_name, tc.total_spent
    FROM customer c
    JOIN TopCustomers tc ON c.customer_id = tc.ss_customer_sk
)
SELECT /*+ MERGE() */ *
FROM CustomerDetails;
```

- SalesSummary 计算每个顾客的总消费。
- TopCustomers 从 SalesSummary 结果集中筛选出消费超过1000的顾客。
- CustomerDetails 将 customer 表中的顾客信息与 TopCustomers 中的结果集进行连接。
- 最终的 SELECT 查询从 CustomerDetails 中提取所有数据。

执行结果：

```
+-------------+------------+-----------+--------------+
| customer_id | first_name | last_name | total_spent  |
+-------------+------------+-----------+--------------+
|        1001 | John       | Doe       |      1523.75 |
|        1002 | Jane       | Smith     |      2105.50 |
|        1003 | Michael    | Johnson   |      1789.99 |
|        1004 | Emily      | Brown     |      1650.25 |
```

```
|         1005 | David      | Wilson    |       1875.00 |
|         1006 | Sarah      | Davis     |       2250.75 |
|         1007 | Robert     | Taylor    |       1955.50 |
|         1008 | Jennifer   | Anderson  |       1725.25 |
|         1009 | William    | Thomas    |       2015.00 |
|         1010 | Lisa       | Jackson   |       1890.75 |
+--------------+------------+-----------+---------------+
10 rows in set (0.15 sec)
```

窗口函数使用说明

最近更新时间：2025-04-09 21:31:31

窗口函数也叫 OLAP 函数（Online Analytical Processing，联机分析处理），可以对数据进行实时分析处理，对于一个分析型数据库而言非常重要。只读分析引擎中窗口函数的使用方法与 MySQL 8.0 基本一致。在 MySQL 5.7 版本中开启的只读分析引擎依然可以使用窗口函数与开窗语法。
具体的开窗使用语法如下：

```
SELECT
    <WindowFunction> OVER (PARTITION BY <expr_list> ORDER BY <expr_list> [ASC / DESC] [<WindowFrame>])
FROM
    tb_test_window;
```

SQL 示例：

```
select studentid,departmentid,classid,math,
row_number() over(partition by departmentid,classid order by math) as row_num
from student_scores;
```

此示例为：对分组 departmentid，classid 内的数据按 math 排序。

WindowFrame

Frame	ROWS 场景	RANGE 场景
CURRENT ROW	当前行。	与当前行相同的所有行。
UNBOUNDED PRECEDING	到第一行。	到第一行。
UNBOUNDED FOLLOWING	到最后一行。	到最后一行。
<N> PRECEDING	前 N 行。	到大于等于 OrderBy 列值减去<N>得到的值对应的行。
<N> FOLLOWING	后 N 行。	到小于等于 OrderBy 列值加上<N>得到的值对应的行。

- ROWS | RANGE <Frame>。
- ROWS | RANGE BETWEEN <Frame> AND <Frame>。

WindowFunction

目前在开窗语法中支持使用的窗口函数如下表所示：

函数名称	功能描述	函数参数	支持类型
ROW_NUMBER()	对每个分区内数据标记行号。	-	-
RANK()	对每个分区内数据非密集型排序。	-	-
DENSE_RANK()	对每个分区内数据密集型排序。	-	-
LEAD(<expr>，<offset>，<default>)	计算当前行的后<offset>行的值，没有满足行则返回<default>。	[必选] <expr>：计算列。	所有类型（三参数除了 Time 类型）。
		[可选] <offset>：当前行向后偏移行，缺省为1。	数值类型。
		[可选] <default>：不满足计算行默认返回值，缺省返回 NULL。	与 <expr> 类型一致。

LAG(<expr>，<offset>，<default>)	计算当前行的前<offset>行的值，没有满足行则返回<default>。	[必选] <expr>：计算列。	所有类型（三参数除了 Time 类型）。
		[可选] <offset>：当前行向前偏移行，缺省为1。	数值类型。
		[可选] <default>：不满足计算行默认返回值，缺省返回 NULL。	与 <expr> 类型一致。
FIRST_VALUE(<expr>)	计算分区窗口内第一个值。	[必选] <expr>：计算列。	所有类型。
LAST_VALUE(<expr>)	计算分区窗口内最后一个值。	[必选] <expr>：计算列。	所有类型。
MIN(<expr>)	计算分区窗口OrderBy列最小值对应行的<expr>值。	[必选] <expr>：计算列。	所有类型。
MAX(<expr>)	计算分区窗口OrderBy列最大值对应行的<expr>值。	[必选] <expr>：计算列。	所有类型。
COUNT(<expr>)	计算分区窗口内数据总行数。	[必选] <expr>：计算列。	所有类型。
SUM(<expr>)	计算分区窗口内数据总和。	[必选] <expr>：计算列。	数值类型。
AVG(<expr>)	计算分区窗口内数据平均值。	[必选] <expr>：计算列。	数值类型。

- 数值类型：int，bigint，float，double，decimal。
- 字符类型：char，varchar。
- 时间类型：date，time，datetime，timestamp。

详细案例

案例建表语句：

```
drop table if exists test.tb_window;
create table test.tb_window (c1 int not null primary key, c2 int, c3 int);
create table test.tb_window (c1 Int32, c2 Nullable(Int32), c3 Nullable(Int32)) engine = LibraTree order by
(c1);
insert into test.tb_window values (1, 1, 1), (2, 1, 1), (3, 1, 2), (4, 1, 4), (5, 1, 6), (6, 1, 6);
```

ROWS 关键字

> **说明：**
> 该关键字是按照行统计窗口大小，并对窗口内数据进行计算。

```
-- 案例语句
mysql> select c2, c3, COUNT(c1) over (partition by c2 order by c3 ROWS BETWEEN CURRENT ROW AND 2
FOLLOWING) cn from test.tb_window;
+----+----+----+
| c2 | c3 | cn |
+----+----+----+
|  1 |  1 |  3 |  -- 窗口行索引范围：current -> 之后2行 [0 ~ 2]
|  1 |  1 |  3 |  -- 窗口行索引范围：current -> 之后2行 [1 ~ 3]
|  1 |  2 |  3 |  -- 窗口行索引范围：current -> 之后2行 [2 ~ 4]
|  1 |  4 |  3 |  -- 窗口行索引范围：current -> 之后2行 [3 ~ 5]
|  1 |  6 |  2 |  -- 窗口行索引范围：current -> 之后1行 [4 ~ 5] （后边只有一行）
|  1 |  6 |  1 |  -- 窗口行索引范围：current （后边没有数据）
+----+----+----+
6 rows in set (0.06 sec)
```

RANGE 关键字

> **说明：**
> 该关键字是按照值统计窗口大小，并对窗口内数据进行计算，例如以下 SQL 案例是对 C3列当前行值加2后找到对应行位置，到当前行之间为一个窗口。

```
-- 案例语句
mysql> select c2, c3, COUNT(c1) over (partition by c2 order by c3 RANGE BETWEEN CURRENT ROW AND 2
FOLLOWING) cn from test.tb_window;
+----+----+----+
| c2 | c3 | cn |
+----+----+----+
|  1 |  1 |  3 |  -- 窗口行索引范围：current -> 3对应的行索引之间的行数据 [0 ~ 2]
|  1 |  1 |  3 |
|  1 |  2 |  2 |  -- 窗口行索引范围：current -> 4对应的行索引之间的行数据 [2 ~ 3]
|  1 |  4 |  3 |  -- 窗口行索引范围：current -> 6对应的行索引之间的行数据 [3 ~ 5]
|  1 |  6 |  2 |  -- 窗口行索引范围：current -> 8对应的行索引之间的行数据 [4 ~ 5]
|  1 |  6 |  2 |
+----+----+----+
6 rows in set (0.06 sec)
```

ROW_NUMBER

> **说明：**
> 该函数是对分区内数据进行编号处理，该函数不受 <WindowFrame> 限制。

```
-- 案例语句
mysql> select c2, c3, ROW_NUMBER() over (partition by c2 order by c3) rn from test.tb_window;
+----+----+------+
| c2 | c3 | rn   |
+----+----+------+
|  1 |  1 |    1 |
|  1 |  1 |    2 |
|  1 |  2 |    3 |
|  1 |  4 |    4 |
|  1 |  6 |    5 |
|  1 |  6 |    6 |
+----+----+------+
6 rows in set (0.04 sec)
```

RANK & DENSE_RANK

- RANK 函数：对分区内部数据进行非密集型排名，该函数不受 <WindowFrame> 限制。
- DENSE_RANK 函数：对分区内部数据进行密集型排名，该函数不受 <WindowFrame> 限制。

```
-- 案例语句
select
    c2, c3,
    RANK() over (partition by c2 order by c3) rk,
    DENSE_RANK() over (partition by c2 order by c3) drk
from test.tb_window;
+------+------+------+------+
| c2   | c3   | rk   | drk  |
+------+------+------+------+
|    1 |    1 |    1 |    1 |
|    1 |    1 |    1 |    1 |
|    1 |    2 |    3 |    2 |
|    1 |    4 |    4 |    3 |
|    1 |    6 |    5 |    4 |
|    1 |    6 |    5 |    4 |
```

```
+------+------+------+------+
6 rows in set (0.05 sec)
```

LEAD & LAG

一、1个参数场景

- LEAD(<expr>)函数：计算分区当前行的后一行数据，没有后一行默认补 NULL，该函数不受 <WindowFrame> 限制。
- LAG(<expr>)函数：计算分区当前行的前一行数据，没有前一行默认补 NULL，该函数不受 <WindowFrame> 限制。

```
-- 案例语句
mysql> select
    c2, c3,
    LEAD(c3) over (partition by c2 order by c3) ld,
    LAG(c3) over (partition by c2 order by c3) lg
from test.tb_window;
+------+------+------+------+
| c2   | c3   | ld   | lg   |
+------+------+------+------+
|    1 |    1 |    1 | NULL |
|    1 |    1 |    2 |    1 |
|    1 |    2 |    4 |    1 |
|    1 |    4 |    6 |    2 |
|    1 |    6 |    6 |    4 |
|    1 |    6 | NULL |    6 |
+------+------+------+------+
6 rows in set (0.11 sec)
```

二、2个参数场景

- LEAD(<expr>, <offset>)函数：计算分区当前行的后 <offset> 行数据，没有后 <offset> 行默认补 NULL，该函数不受 <WindowFrame> 限制。
- LAG(<expr>, <offset>)函数：计算分区当前行的前 <offset> 行数据，没有前 <offset> 行默认补 NULL，该函数不受 <WindowFrame> 限制。

```
-- 案例语句
mysql> select
    c2, c3,
    LEAD(c3, 2) over (partition by c2 order by c3) ld,
    LAG(c3, 2) over (partition by c2 order by c3) lg
from test.tb_window;
+------+------+------+------+
| c2   | c3   | ld   | lg   |
+------+------+------+------+
|    1 |    1 |    2 | NULL |
|    1 |    1 |    4 | NULL |
|    1 |    2 |    6 |    1 |
|    1 |    4 |    6 |    1 |
|    1 |    6 | NULL |    2 |
|    1 |    6 | NULL |    4 |
+------+------+------+------+
6 rows in set (0.07 sec)
```

三、3个参数场景

- LEAD(<expr>, <offset>, <default>>)函数：计算分区当前行的后 <offset> 行数据，没有后 <offset> 行补 <default>，该函数不受 <WindowFrame> 限制。
- LAG(<expr>, <offset>, <default>>)函数：计算分区当前行的前 <offset> 行数据，没有前 <offset> 行补 <default>，该函数不受 <WindowFrame> 限制。

```
-- 案例语句
```

```
mysql> select
    c2, c3,
    LEAD(c3, 2, 1000) over (partition by c2 order by c3) ld,
    LAG(c3, 2, 1000) over (partition by c2 order by c3) lg
from test.tb_window;
+------+------+------+------+
| c2   | c3   | ld   | lg   |
+------+------+------+------+
|    1 |    1 |    2 | 1000 |
|    1 |    1 |    4 | 1000 |
|    1 |    2 |    6 |    1 |
|    1 |    4 |    6 |    1 |
|    1 |    6 | 1000 |    2 |
|    1 |    6 | 1000 |    4 |
+------+------+------+------+
6 rows in set (0.10 sec)
```

FIRST_VALUE & LAST_VALUE

- FIRST_VALUE(<expr>)函数：计算分区内窗口的第一个值（如果 OrderBy c3，且 c3列第一个值存在重复数据，那么 first_value(c4)，场景结果可能是不稳定结果）。
- LAST_VALUE(<expr>)函数：计算分区内窗口的最后一个值（如果 OrderBy c3，且 c3列最后一个值存在重复数据，那么 last_value(c4)，场景结果可能是不稳定结果）。

```
-- 案例语句
mysql> select
    c2, c3,
    FIRST_VALUE(c3) over (partition by c2 order by c3 ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED
FOLLOWING) fv,
    LAST_VALUE(c3) over (partition by c2 order by c3 ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED
FOLLOWING) lv
from test.tb_window;
+------+------+------+------+
| c2   | c3   | fv   | lv   |
+------+------+------+------+
|    1 |    1 |    1 |    6 |
|    1 |    1 |    1 |    6 |
|    1 |    2 |    1 |    6 |
|    1 |    4 |    1 |    6 |
|    1 |    6 |    1 |    6 |
|    1 |    6 |    1 |    6 |
+------+------+------+------+
6 rows in set (0.07 sec)
```

MIN & MAX

- MIN(<expr>)函数：计算分区内窗口的最小值。
- MAX(<expr>)函数：计算分区内窗口的最大值。

```
-- 案例语句
mysql> select
    c2, c3,
    MIN(c3) over (partition by c2 order by c3 ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING)
mi,
    MAX(c3) over (partition by c2 order by c3 ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) ma
from test.tb_window;
+------+------+------+------+
| c2   | c3   | mi   | ma   |
+------+------+------+------+
```

```
|    1 |    1 |    1 |    6 |
|    1 |    1 |    1 |    6 |
|    1 |    2 |    1 |    6 |
|    1 |    4 |    1 |    6 |
|    1 |    6 |    1 |    6 |
|    1 |    6 |    1 |    6 |
+------+------+------+------+
6 rows in set (0.07 sec)
```

COUNT

> **说明：**
> 计算分区内窗口的数据总行数。

```
-- 案例语句
mysql> select c2, c3, COUNT(c3) over (partition by c2 order by c3 ROWS BETWEEN UNBOUNDED PRECEDING AND
UNBOUNDED FOLLOWING) c from test.tb_window;
+------+------+----+
| c2   | c3   | c  |
+------+------+----+
|    1 |    1 |  6 |
|    1 |    1 |  6 |
|    1 |    2 |  6 |
|    1 |    4 |  6 |
|    1 |    6 |  6 |
|    1 |    6 |  6 |
+------+------+----+
6 rows in set (0.04 sec)
```

SUM

> **说明：**
> 计算分区内窗口的数据总和。

```
-- 案例语句
mysql> select c2, c3, SUM(c3) over (partition by c2 order by c3 ROWS BETWEEN UNBOUNDED PRECEDING AND
UNBOUNDED FOLLOWING) s from test.tb_window;
+------+------+------+
| c2   | c3   | s    |
+------+------+------+
|    1 |    1 |   20 |
|    1 |    1 |   20 |
|    1 |    2 |   20 |
|    1 |    4 |   20 |
|    1 |    6 |   20 |
|    1 |    6 |   20 |
+------+------+------+
6 rows in set (0.06 sec)
```

AVG

> **说明：**
> 计算分区内窗口的数据平均值。

```
-- 案例语句
```

```
mysql> select c2, c3, AVG(c3) over (partition by c2 order by c3 ROWS BETWEEN UNBOUNDED PRECEDING AND
UNBOUNDED FOLLOWING) a from test.tb_window;
+------+------+--------+
| c2   | c3   | a      |
+------+------+--------+
|    1 |    1 | 3.3333 |
|    1 |    1 | 3.3333 |
|    1 |    2 | 3.3333 |
|    1 |    4 | 3.3333 |
|    1 |    6 | 3.3333 |
|    1 |    6 | 3.3333 |
+------+------+--------+
6 rows in set (0.06 sec)
```

版权所有：腾讯云计算（北京）有限责任公司

执行 SQL 并查看执行计划

最近更新时间：2024-10-23 11:08:52

只读分析引擎兼容 MySQL 协议与语法。用户可通过 MySQL 客户端或开发语言中的 MySQL 连接器访问到只读分析引擎中执行 SQL 语句。
只读分析引擎是一个只读实例，故可执行的 SQL 语句主要为 SELECT 语句，无法执行 DML 和 DDL 语句。同时查询也严格受限于对象权限，若访问账户不拥有某一对象的 SELECT 权限，则亦无法进行此对象的查看。
在执行 SQL 时，可直接使用 MySQL 的语法进行语法编辑，然后到只读分析引擎中执行。但是时常也会遇到执行结果不及预期的情况，此时则可以通过 EXPLAIN 功能查看执行计划。

EXPLAIN 功能及使用

使用 EXPLAIN 可查看只读分析引擎中，执行查询语句时选用的执行计划，这是经过内部优化器进行多个阶段优化后，最终选出来的最优查询计划。查看执行计划时不会真正的执行 SQL，仅会输出执行计划。
EXPLAIN 示例如下：

```
EXPLAIN <SELECT_STATMENT>

explain select * from t1 left join t2 on t1.id = t2.id;
```

返回结果如下：

```
+-------------------------------------------------------------------------------+
| query plan                                                                    |
+-------------------------------------------------------------------------------+
| ============================================                                  |
| |ID|OPERATOR                 |NAME|EST.ROWS|                                  |
| --------------------------------------------                                  |
| |0 |HASH JOIN                |    |12500.00|                                  |
| |1 |├─TABLE FULL SCAN (B)    |t2  |10000.00|                                  |
| |2 |└─TABLE FULL SCAN (P)    |t1  |10000.00|                                  |
| --------------------------------------------                                  |
| Details:                                                                      |
| ------------------------------------                                          |
|   0 - JOIN TYPE: left outer join, EQUAL: [eq(singleton.t1.id, singleton.t2.id)] |
|   1 - STORAGE: LIBRASTORE, BLOCK OFFSET: 1                                    |
|   2 - STORAGE: LIBRASTORE, BLOCK OFFSET: 1                                    |
+-------------------------------------------------------------------------------+
```

EXPLAIN 结果解读

EXPLAIN 的返回结果包含以下字段：

- ID 为算子在计划中展示的编号，从0开始往下逐级递增，方便用户用于查看 detail。
- OPERATOR 为算子名称，代表着 SQL 执行过程中每一步进行的操作。
- NAME 显示在查询中所访问到的表、分区等信息。
- EST.ROWS 显示只读分析引擎根据统计信息估算出来的每个算子预计会处理的行数，如果没有统计信息，则会基于一个默认值进行计算显示。

算子介绍

算子是为了返回查询结果而执行的特定步骤，下表为只读分析引擎目前支持的各类算子名称及其作用介绍：

算子名称	算子介绍
SORT	SORT 算子用于对输入的数据进行排序。
TOPN	如果 ORDER BY 语句块后面还有 LIMIT 语句，优化器会进一步优化执行计划，生成 TOP-N SORT 算子，采用堆排序来选择 TOP-N 的数据。
LIMIT	LIMIT 算子用于限制数据输出的行数，这与 MySQL 的 LIMIT 算子功能相同。

版权所有：腾讯云计算（北京）有限责任公司

FILTER	FILTER 算子用于对数据按指定谓词条件进行过滤操作，通常出现在 WHERE/HAVING/ON 子句中。
HASH JOIN	利用 HASH JOIN 来进行大数据集的 JOIN 操作。优化器使用两个数据集中表，较小的那个在内存中根据 JOIN 条件构建 HASH 表，之后数据库扫描更大的数据集，探测 HASH 表找到满足关联条件的行。
COLUMN READ	延迟物化算子，只读分析引擎支持下推部分过滤条件到 TableScan 算子，即先扫描过滤条件相关的列数据，过滤得到符合条件的行后，再扫描这些行的其他列数据，继续后续计算，从而减少 IO 扫描和数据处理的计算量。
TABLE FULL SCAN	对目标表进行全表扫描。
UNION	UNION 算子用于将两个查询的结果集进行并集运算。
WINDOW FUNCTION	WINDOW FUNCTION 算子用于实现 SQL 中的分析函数（也叫窗口函数），计算窗口内的相关行的结果，窗口函数每组可以返回多行，组内每一行都是基于窗口的逻辑计算的结果。
HASH GROUP BY	GROUP BY 算子主要用于在 SQL 中进行分组聚合计算操作，对于普通的聚合函数（SUM/MAX/MIN/AVG/COUNT/STDDEV）都是通过分配 GROUP BY 算子来完成。
PROJECTION	投影算子，对应 SQL 语句中的 SELECT 列表，功能是将每一条输入数据映射成新的输出数据。
EXCHANGE RECEIVER	数据接收算子，用于在执行 MPP 查询各个计算节点交换数据时，接收数据用。
EXCHANGE SENDER	数据发送算子，用于在执行 MPP 查询各个计算节点交换数据时，发送数据用。

Detail 信息

除了上述返回结果表格中展示的信息以外，可以看到在下方还会展示一个 Detail 项，Detail 项中展示了每个算子所带有的一些额外的信息，其格式呈 n – detail info，其中 n 代表的是每个算子的 ID，后面则是每个算子的具体信息。下表列出了每个算子会展示出的具体信息。

算子名称	Detail 信息
SORT	ORDER BY KEY：代表排序算子所采用的排序键。
TOPN	ORDER BY KEY：代表排序算子所采用的排序键。 OFFSET：Limit 操作所指定的偏移量。 COUNT：Limit 操作所指定的保留行数。
LIMIT	OFFSET：Limit 操作所指定的偏移量。 COUNT：Limit 操作所指定的保留行数。
FILTER	CONDITIONS：代表 FILTER 算子在过滤数据时所采用的谓词条件。
HASH JOIN	JOIN TYPE：当前 JOIN 的类型，如：inner join、left outer join、semi join 等。 NON EQUAL：当前 JOIN 是否是笛卡尔积，如果没有这个字段，则代表其不是笛卡尔积。 EQUAL：显示两个表进行 JOIN 时采用的等值条件。 OTHER COND：显示两个表进行 JOIN 时的非等值条件。 此外可以看到上文中 EXPLAIN 结果中 ID 为1和2的算子名称后面分别跟了（B）和（P）标志，这两个标志标记了在进行 HASH JOIN 操作时的 Build 与 Probe 端，B 代表 Build，P 代表 Probe。
COLUMN READ	COLUMN READ：延迟物化所读取的列名字。
TABLE FULL SCAN	STORAGE：所读取的底层存储类型，目前支持两种类型，为 LIBRASTORE、TDSQL。 BLOCK OFFSET：当前表在整条 SQL 所处的 Query Block 编号，用于辅助 Hint 使用。
UNION	无。
WINDOW FUNCTION	WINDOW FUCN DESCS：窗口函数名称。 PARTITION BY：分区键。 ORDER BY：排序所采用的排序键。 FRAME：窗口函数的窗口定义。
HASH GROUP BY	GROUP BY：执行聚合函数时指定的 group by key。 AGG FUNCS：SQL 中指定的聚合函数。
PROJECTION	EXPRS：PROJECTION 算子执行的表达式列表，主要是执行的 cast、还有各种 scalarfunc。

版权所有：腾讯云计算（北京）有限责任公司

EXCHANGE RECEIVER	无。
EXCHANGE SENDER	ExchangeType：交换数据采用的方式： • PASS：将多个节点的数据发送往一个节点。 • BCJ：将一个节点的数据广播到多个节点。例如 JOIN 时将 build 表的数据广播到各个节点，以便执行 JOIN 操作。 • HASH：将数据通过 HASH 函数散列后分发到各个节点。例如用于 JOIN 时将左右两表数据打散重新分布。 • HASH (BY PARTITION)：当两表 JOIN，其中一个表的 join key 是 partition key 时，将另一个表按照 join key 是 partition key 的表分布方式 shuffle 出去。

除了上述表中算子的基本信息外，当开启了 runtime filter 算子且计划中有 HASH JOIN 时，HASH JOIN 算子两侧可能带有以下两种特殊的 detail 信息。

- Probe Runtime Filters：当算子上有这个信息时，代表当前算子 apply 了 runtime filter 以过滤冗余数据。
- Build Runtime Filters：此信息只会出现在 JOIN 算子上，代表当前 JOIN 算子的 build 侧生成了一个 runtime fileter，以提前过滤 probe 侧冗余数据，具体 runtime filter 的使用、调优以及显示信息的具体作用，可参考 Runtime Filter 使用手册 。

EXPLAIN 执行后，不同算子返回不同的信息。您可以使用 Optimizer Hints 来控制优化器的行为，以此控制物理算子的选择。
例如 `/*+ HASH_JOIN_PROBE(t1) */` 表示优化器将强制采用 t1表作为 hash join 中的 probe 表。

特色功能
加速 ETL 回写

最近更新时间：2025-01-16 10:11:52

TDSQL-C MySQL 版可通过只读分析引擎加速数据查询并回写至读写实例，使用此功能，您可以在只读分析引擎中使用列存加速 `INSERT…SELECT…` 语句中的 SELECT 语句查询，并将结果通过内部网络将数据传回读写实例并写入目标表。详细的功能描述请参考 加速 ETL 回写 。

使用场景

> **注意：**
> - 因为只读分析引擎为异步复制模式，所以当只读分析引擎存在延迟时查询的结果与读写实例中存在一定的时差。此功能仅适用于对数据延迟不敏感的场景。
> - 当前仅支持 `INSERT...SELECT...` 加速，暂不支持 `UPDATE...SELECT...` 以及 `DELETE...SELECT...` 等场景加速。

推荐在以下场景使用此功能：当查询条件复杂、SQL 语句执行时间较长但查询结果集的数据量较小时。在这种场景下，使用此功能可以显著提高性能，因为只读分析引擎会加速 SELECT 查询的执行效率。
使用此功能并不是在所有场景下都能带来性能收益，在某些场景下性能可能会下降。例如：

- `INSERT…SELECT…` 语句中的 SELECT 查询比较简单时，从只读分析引擎读取数据并回写到读写实例会带来额外的网络开销。对比直接从读写实例读取数据的优势并不明显。
- `INSERT…SELECT…` 语句中的 SELECT 查询结果集数据量大的情况下，此时主要的性能瓶颈在于结果集通过网络传输并写入读写实例的过程。此场景无法使用此功能来优化性能。

前提条件

- 只读分析引擎实例和读写实例的运行状态均为正常运行中。
- 执行的用户需拥有对应对象的 INSERT 权限。
- 仅支持 `INSERT…SELECT…` 的 SQL 执行场景。

功能参数说明

参数名	参数说明	默认值	参数值
libra_etl_to_rw	控制是否执行回写功能。	off	• on：开启回写能力，表示执行回写。 • off：关闭回写能力。当值为 off 时，执行 `INSERT…SELECT…` 语句时会报错权限不足。
libra_concurrent_etl	控制是否在回写读写实例时开启并发写入。	off	• on: 开启并发回写。并发回写能提升结果回写效率，但是不能保证多线程间写入的事务一致性。 • off：关闭并发回写。设置为此值时，回写至读写实例的数据将在一个事务中完成。
libra_etl_concurrency	控制回写读写实例的并发线程数。只有在 libra_concurrent_etl 设置为 on 时才会生效。	1	取值范围：[1 – 读写实例 CPU 核数]，可设置值为整数，最大值为读写实例的 CPU 核数。 **注意：** 使用多并发回写时执行速度更快，但是并发数量过高会导致读写实例无法承受其写入压力而使得实例性能受损。

执行示例

1. 使用客户端工具登录至只读分析引擎。例如使用 MySQL 客户端通过数据库账号密码登录至只读分析引擎。
2. 执行如下 SQL，即可开启加速 ETL 回写功能（通过只读分析引擎回写读写实例功能）。
   通过在 seesion 中设置参数执行：

```
set libra_etl_to_rw=on;
set libra_concurrent_etl=on;
set libra_etl_concurrency=4;
INSERT INTO database1.table1 select a.t1,b.t2,a.t2,b.t3 from a,b where a.t1=b.t1 and a.t5='x';
set libra_concurrent_etl=off;
set libra_etl_to_rw=off;
```

通过在 SQL 中添加 Hint 参数执行：

```
INSERT /*+ SET_VAR(libra_etl_to_rw=ON) SET_VAR(libra_concurrent_etl=ON)
SET_VAR(libra_etl_concurrency=4)*/ INTO database1.table1 select a.t1,b.t2,a.t2,b.t3 from a,b where
a.t1=b.t1 and a.t5='x';
```

> ⚠ **注意：**
> 如果使用 mysql 客户端执行“添加 Hint 参数的方法执行”，需要在 MySQL 客户端登录只读分析引擎时添加 -c 参数，否则 hint 将不会生效。

性能对比

测试环境

- 读写实例：16核64GB。
- 只读分析引擎：16核64GB。

测试数据

使用 TPC-H 10GB数据集。

测试复杂 SQL 及结果

```
set libra_etl_to_rw=on;
set libra_concurrent_etl=off;
set libra_etl_concurrency=1;

create table t1 (
    supp_nation VARCHAR(50),
    cust_nation VARCHAR(50),
    l_year INT,
    revenue DECIMAL
);

INSERT INTO t1 SELECT
supp_nation, cust_nation, l_year, sum(volume) as revenue
from (select n1.n_name as supp_nation, n2.n_name as cust_nation,
             extract(year from l_shipdate) as l_year,
             l_extendedprice * (1 - l_discount) as volume
      from supplier, lineitem, orders, customer, nation n1, nation n2
      where s_suppkey = l_suppkey
        and o_orderkey = l_orderkey
        and c_custkey = o_custkey
        and s_nationkey = n1.n_nationkey
        and c_nationkey = n2.n_nationkey
        and ((n1.n_name = 'JAPAN' and n2.n_name = 'INDIA')
              or (n1.n_name = 'INDIA' and n2.n_name = 'JAPAN'))
        and l_shipdate between '1995-01-01' and '1996-12-31') as shipping
group by supp_nation, cust_nation, l_year
order by supp_nation, cust_nation, l_year;
```

查询结果为4行数据时，测试结果如下（单位：秒）。

只读实例 SELECT 的时间	未开启回写功能 在读写实例执行 INSERT…SELECT… 的时间	开启回写功能 在只读分析引擎执行 INSERT…SELECT… 的时间
0.57	232.77	0.61

```
set libra_etl_to_rw=on;
set libra_concurrent_etl=off;
set libra_etl_concurrency=1;

CREATE TABLE t2
 (
    p_brand VARCHAR(10),
    p_type VARCHAR(25),
    p_size INTEGER,
    supplier_cnt INTEGER
);

INSERT INTO t2 SELECT
    p_brand,
    p_type,
    p_size,
    count(distinct ps_suppkey) as supplier_cnt
from
    partsupp,
    part
where
    p_partkey = ps_partkey
    and p_brand <> 'Brand#45'
    and p_type not like 'MEDIUM POLISHED%'
    and p_size in (49, 14, 23, 45, 19, 3, 36, 9)
    and ps_suppkey not in (
        select
            s_suppkey
        from
            supplier
        where
            s_comment like '%Customer%Complaints%'
    )
group by
    p_brand,
    p_type,
    p_size
order by
    supplier_cnt desc,
    p_brand,
    p_type,
    p_size;
```

查询结果为27840行数据时，测试结果如下（单位：秒）。

只读实例 SELECT 的时间	未开启回写功能 在读写实例执行 INSERT…SELECT… 的时间	开启回写功能 在只读分析引擎执行 INSERT…SELECT… 的时间
0.15	8.77	0.61

测试简单 SQL 及结果

```
set libra_etl_to_rw=on;
set libra_concurrent_etl=on;
set libra_etl_concurrency=8;
```

```
CREATE TABLE IF NOT EXISTS lineitem ( L_ORDERKEY INTEGER NOT NULL,
L_PARTKEY INTEGER NOT NULL,
L_SUPPKEY INTEGER NOT NULL,
L_LINENUMBER INTEGER NOT NULL,
L_QUANTITY DECIMAL(15,2) NOT NULL,
L_EXTENDEDPRICE DECIMAL(15,2) NOT NULL,
L_DISCOUNT DECIMAL(15,2) NOT NULL,
L_TAX DECIMAL(15,2) NOT NULL,
L_RETURNFLAG CHAR(1) NOT NULL,
L_LINESTATUS CHAR(1) NOT NULL,
L_SHIPDATE DATE NOT NULL,
L_COMMITDATE DATE NOT NULL,
L_RECEIPTDATE DATE NOT NULL,
L_SHIPINSTRUCT CHAR(25) NOT NULL,
L_SHIPMODE CHAR(10) NOT NULL,
L_COMMENT VARCHAR(44) NOT NULL , primary key(L_ORDERKEY, L_LINENUMBER));

INSERT INTO lineitem_t SELECT
    *
from
    lineitem;
```

查询结果为59986051行数据时，测试结果如下（单位：秒）。

执行并发	未开启回写功能 在读写实例执行 INSERT…SELECT… 的时间	开启回写功能 在只读分析引擎执行 INSERT…SELECT… 的时间
不开启并发	622.7	1441.02
并发数为8		259.20
并发数为16		181.79
并发数为32		185.64

分页保序功能

最近更新时间：2025-04-15 15:35:22

分页保序功能是只读分析引擎在并行执行数据查询时，因 order by 字段存在重复值导致分页查询结果不稳定的解决方法。本文介绍如何使用分页保序功能。详细的功能介绍，请参考 分页保序能力 。

保序场景

- 无 order by 操作符，仅仅有 limit 的场景。
- order by 字段存在重复值的场景。
- 子查询中包含排序，但是外层查询中未进行排序的场景。

使用说明

- **默认开启全局分页保序功能**

  使用 参数设置 功能，修改只读分析引擎的参数“libra_preserve_order_for_pagination”。此参数默认值为 OFF，修改为 ON 则对整实例开启了分页保序能力。

  > ⚠ **注意：**
  > 因为分页保序功能会默认对查询的输出结果添加隐式排序，所以输出结果集较大的场景下会导致性能损耗。建议在 SQL 执行中对需要此功能的 SQL 指定显式开启分页保序功能。

- **在 session 会话中开启分页保序功能**

  在访问至只读分析引擎后，可显示指定会话级的参数设置。在当前会话开启/关闭分页保序功能。

```
mysql> set libra_preserve_order_for_pagination=on;
mysql> set libra_preserve_order_for_pagination=off;
```

- **在 SQL 执行中指定使用分页保序功能**

  在 SQL 执行的时候，添加 hint 语法显式指定此 SQL 使用分页保序能力。

  > ⚠ **注意：**
  > 如果使用 mysql 客户端访问只读分析引擎，请添加 -c 参数。否则 SQL 中显示指定分页保序无法生效。

```
mysql>  select /*+ SET_VAR(libra_preserve_order_for_pagination=ON) */ t1.c1, t1.c2 from t1, t2 where
t1.c1 = t2.c1 order by t1.c1 limit 10,10;
```

性能优化
通过 HINT 优化 SQL 执行

最近更新时间：2025-01-20 15:31:22

什么是 Hint

优化器一般会为用户 SQL 选择最优的执行计划，但是在某些场景下，例如统计信息估计误差、代价模型拟合偏差等，优化器生成的执行计划可能就不是最优的。这个时候用户可以通过 Hint 机制指导优化器生成更优的执行计划。
Hints 不区分大小写，通过 /*+ ... */ 注释的形式跟在 SELECT 关键字后面，多个 Hint 通过空格或逗号间隔开。以下是一个 Hint 使用示例。

```
SELECT /*+ [hint_text] [hin_text]... */ * FROM ....
```

Hint 生效范围

Hint 以 Query Block 为单位生效，在查询语句中，每一个查询 Block 都会有一个 QB_NAME (Query Block Name)，只读分析引擎会按照 @sel_1、@sel_2 的方式，从左到右依次为每个查询 Block 生成 QB_NAME。以下面的 SQL 为例。

```
SELECT * FROM (SELECT * FROM t) t1, (SELECT * FROM t) t2;
```

这条 SQL 中包含了三个 Query Block，最外面一层 SELECT 所在的查询块的名字为 sel_1，两个 SELECT 子查询的名字依次为 sel_2 和 sel_3，编号依次递增。Hint 中使用 QB_NAME 即可控制 Hint 的作用范围以及 Hint 的作用对象。如果在 Hint 中没有显式指定 QB_NAME，则 Hint 作用范围为当前 Hint 所在的 Query Block，示例如下。

```
SELECT /*+ HASH_JOIN_PROBE(@sel_2 t1) */ * FROM (SELECT t1.a, t1.b FROM t t1, t t2 WHERE t1.a = t2.a) t1,
t t3 WHERE t1.b = t3.b;
SELECT /*+ HASH_JOIN_PROBE(t1@sel_2) */ * FROM (SELECT t1.a, t1.b FROM t t1, t t2 WHERE t1.a = t2.a) t1, t
t3 WHERE t1.b = t3.b;
```

以上两条 SQL 使用了两种不同方法在 Hint 中指定 QB_NAME，第一条 SQL 通过在 Hint 的第一个参数指定 QB_NAME 并用空格与其他参数间隔开。第二条 SQL 通过在参数中后面加上 QB_NAME 来指定 Hint 的生效范围。

支持的 Hint 概览

只读分析引擎 Hint 相关参数名称、语义和语法如下表所示。

名称	语法	语义
SHUFFLE_JOIN	SHUFFLE_JOIN([QB_NAME] tbl1_name,tbl2_name …)	指定 JOIN 操作采用 shuffle 方式分发数据。
BROADCAST_JOIN	BROADCAST_JOIN([QB_NAME] tbl1_name,tbl2_name …)	指定 JOIN 操作采用 broadcast 方式分发数据。
HASH_JOIN_BUILD	HASH_JOIN_BUILD([QB_NAME] tbl1_name,tbl2_name …)	指定 HASH JOIN 操作中的 Build 表。
HASH_JOIN_PROBE	HASH_JOIN_PROBE([QB_NAME] tbl1_name,tbl2_name …)	指定 HASH JOIN 操作中的 Probe 表。
LEADING	LEADING([QB_NAME] tbl1_name,tbl2_name …)	指定 JOIN 操作的 Join Order。
SET_VAR	SET_VAR(setting_name = value)	在 SQL 级别设置系统参数。
NO_PX_JOIN_FILTER_ID/PX_JOIN_FILTER_ID	NO_PX_JOIN_FILTER_ID(rf_id1,rf_id2…)/ PX_JOIN_FILTER_ID(rf_id1,rf_id2…)	控制 runtime filter 的开启与关闭。

只读分析引擎 Hint 语法详解

SHUFFLE_JOIN(t1_name, t2_name ...)

使用说明

SHUFFLE_JOIN(t1_name, t2_name ...)，用于控制只读分析引擎优化器在进行 JOIN 操作的时候采用 Shuffle Join 算法，将左右两个表数据进行打散重新分布，再进行 JOIN 操作返回结果。

参考示例

```
EXPLAIN SELECT /*+ SHUFFLE_JOIN(t1) */ * FROM t1, t2 WHERE t1.id = t2.id;
```

此条 SQL 中 t1 与 t2 表进行 JOIN 操作，通过 SHUFFLE_JOIN Hint 指定了 Join 数据分发方式为 Shuffle Join。最终计划如下图所示，可以看到 Details 中的第2行和第5行的 EXCHANGE TYPE 已经变成了 HASH，代表采用了 Hash Shuffle。

```
mysql> explain SELECT /*+ SHUFFLE_JOIN(t1) */ * FROM t1, t2 WHERE t1.id = t2.id;
+-------------------------------------------------------------------------------+
| query plan                                                                    |
+-------------------------------------------------------------------------------+
| ============================================                                  |
| |ID|OPERATOR                   |NAME|EST.ROWS|                                |
| --------------------------------------------                                  |
| |0 |HASH JOIN                  |    |12500.00|                                |
| |1 | ├─EXCHANGE RECEIVER (B)   |    |10000.00|                                |
| |2 | └─EXCHANGE SENDER         |    |10000.00|                                |
| |3 |   └─TABLE FULL SCAN       |t1  |10000.00|                                |
| |4 | └─EXCHANGE RECEIVER (P)   |    |10000.00|                                |
| |5 |   └─EXCHANGE SENDER       |    |10000.00|                                |
| |6 |     └─TABLE FULL SCAN     |t2  |10000.00|                                |
| --------------------------------------------                                  |
| Details:                                                                      |
| ------------------------------------                                          |
|   0 - JOIN TYPE: inner join, EQUAL: [eq(starcat.t1.id, starcat.t2.id)]        |
|   2 - EXCHANGE TYPE: HASH, Hash Cols: [name: starcat.t1.id, collate: N/A]     |
|   3 - STORAGE: LIBRASTORE, BLOCK OFFSET: 1                                    |
|   5 - EXCHANGE TYPE: HASH, Hash Cols: [name: starcat.t2.id, collate: N/A]     |
|   6 - STORAGE: TDSQL, BLOCK OFFSET: 1                                         |
+-------------------------------------------------------------------------------+
```

除了指定单表外，还可以指定 JOIN 中间结果进行数据重分布。

```
EXPLAIN SELECT /*+ SHUFFLE_JOIN((t1@sel_2,t2@sel_2)) */ * FROM (SELECT t1.a, t1.b FROM t t1, t t2 WHERE
t1.a = t2.a) t1, t t3 WHERE t1.b = t3.b;
```

如上所示，通过将 t1 表与 t3 表用括号括起来，并指定好每个表的 QB_NAME，即可指定 t1 与 t3 表 JOIN 的中间结果与 t2 表 JOIN 时采用 Shuffle Join。

```
+-----------------------------------------------------------------------------------------------------+
| query plan                                                                                          |
+-----------------------------------------------------------------------------------------------------+
| ================================================                                                    |
| |ID|OPERATOR                          |NAME|EST.ROWS|                                              |
| ------------------------------------------------                                                    |
| |0 |PROJECTION                        |    |15625.00|                                              |
| |1 | └─HASH JOIN                      |    |15625.00|                                              |
| |2 |   ├─EXCHANGE RECEIVER (B)        |    |10000.00|                                              |
| |3 |   │ └─EXCHANGE SENDER            |    |10000.00|                                              |
| |4 |   │   └─TABLE FULL SCAN          |t3  |10000.00|                                              |
| |5 |   └─EXCHANGE RECEIVER (P)        |    |12500.00|                                              |
| |6 |     └─EXCHANGE SENDER            |    |12500.00|                                              |
| |7 |       └─HASH JOIN                |    |12500.00|                                              |
| |8 |         ├─EXCHANGE RECEIVER (B)  |    |10000.00|                                              |
| |9 |         │ └─EXCHANGE SENDER      |    |10000.00|                                              |
| |10|         │   └─TABLE FULL SCAN    |t1  |10000.00|                                              |
| |11|         └─TABLE FULL SCAN (P)    |t2  |10000.00|                                              |
| ------------------------------------------------                                                    |
| Details:                                                                                            |
| ------------------------------------                                                                |
|   0 - EXPRS: [starcat.t.a, starcat.t.b, starcat.t.pk, starcat.t.id, starcat.t.a, starcat.t.b]       |
|   1 - JOIN TYPE: inner join, EQUAL: [eq(starcat.t.b, starcat.t.b)]                                  |
|   3 - EXCHANGE TYPE: HASH, Hash Cols: [name: starcat.t.b, collate: N/A]                             |
|   4 - STORAGE: LIBRASTORE, BLOCK OFFSET: 1                                                          |
|   6 - EXCHANGE TYPE: HASH, Hash Cols: [name: starcat.t.b, collate: N/A]                             |
|   7 - JOIN TYPE: inner join, EQUAL: [eq(starcat.t.a, starcat.t.a)]                                  |
|   9 - EXCHANGE TYPE: BCJ                                                                            |
|   10 - STORAGE: LIBRASTORE, BLOCK OFFSET: 2                                                         |
|   11 - STORAGE: LIBRASTORE, BLOCK OFFSET: 2                                                         |
+-----------------------------------------------------------------------------------------------------+
```

版权所有：腾讯云计算（北京）有限责任公司

> **注意：**
> 此 Hint 只在生成分布式计划时生效，在单机计划时无效。

BROADCAST_JOIN(t1_name [, tl_name ...])

使用说明

BROADCAST_JOIN(t1_name,t2_name...)，用于控制只读分析引擎优化器在进行 JOIN 操作时，采用 Broadcast Join 算法，将指定表数据广播到所有节点上，进行 JOIN 操作并返回结果。

参考示例

```
EXPLAIN SELECT /*+ BROADCAST_JOIN(t1) */ * FROM t1, t2 WHERE t1.id = t2.id;
```

此条 SQL 中 t1与 t2表进行 JOIN 操作，通过 BROADCAST_JOIN Hint 指定了 JOIN 方式为 Broadcast Join。最终计划如下图所示，可以看到 Details 中的第2行和第5行的 EXCHANGE TYPE 已经变成了 BCJ，代表采用了 Broadcast。

```
+-------------------------------------------------------------------------+
| query plan                                                              |
+-------------------------------------------------------------------------+
| ============================================                            |
| |ID|OPERATOR                   |NAME|EST.ROWS|                         |
| --------------------------------------------                            |
| |0 |HASH JOIN                  |    |12500.00|                         |
| |1 |├─EXCHANGE RECEIVER (B)    |    |10000.00|                         |
| |2 |│  └─EXCHANGE SENDER        |    |10000.00|                         |
| |3 |│    └─TABLE FULL SCAN      |t1  |10000.00|                         |
| |4 |└─TABLE FULL SCAN (P)       |t2  |10000.00|                         |
| --------------------------------------------                            |
| Details:                                                                |
| ------------------------------------                                    |
|   0 - JOIN TYPE: inner join, EQUAL: [eq(starcat.t1.id, starcat.t2.id)] |
|   2 - EXCHANGE TYPE: BCJ                                                |
|   3 - STORAGE: LIBRASTORE, BLOCK OFFSET: 1                              |
|   4 - STORAGE: LIBRASTORE, BLOCK OFFSET: 1                              |
+-------------------------------------------------------------------------+
```

除了指定单表外，还可以指定 JOIN 中间结果进行数据广播。

```
EXPLAIN SELECT /*+ BROADCAST_JOIN((t1@sel_2,t3@sel_1)) */ * FROM (SELECT t1.a, t1.b FROM t t1, t t2 WHERE
t1.a = t2.a) t1, t t3 WHERE t1.b = t3.b;
```

如上所示，通过将 t1 表与 t3 表用括号括起来，并指定好每个表的 QB_NAME 即可指定 t1 与 t3 表 JOIN 的中间结果与 t2 表 JOIN 时采用 Broadcast Join，结果如下图所示。

```
+-----------------------------------------------------------------------------------------------------------+
| query plan                                                                                                |
+-----------------------------------------------------------------------------------------------------------+
| ================================================                                                          |
| |ID|OPERATOR                        |NAME|EST.ROWS|                                                      |
| ------------------------------------------------                                                          |
| |0 |PROJECTION                      |    |15625.00|                                                      |
| |1 |└HASH JOIN                      |    |15625.00|                                                      |
| |2 |  ├EXCHANGE RECEIVER (B)        |    |10000.00|                                                      |
| |3 |  │└EXCHANGE SENDER            |    |10000.00|                                                      |
| |4 |  │  └TABLE FULL SCAN          |t3  |10000.00|                                                      |
| |5 |  └HASH JOIN (P)                |    |12500.00|                                                      |
| |6 |    ├EXCHANGE RECEIVER (B)      |    |10000.00|                                                      |
| |7 |    │└EXCHANGE SENDER          |    |10000.00|                                                      |
| |8 |    │  └TABLE FULL SCAN        |t1  |10000.00|                                                      |
| |9 |    └TABLE FULL SCAN (P)        |t2  |10000.00|                                                      |
| ------------------------------------------------                                                          |
| Details:                                                                                                  |
| ----------------------------------                                                                        |
|   0 - EXPRS: [starcat.t.a, starcat.t.b, starcat.t.pk, starcat.t.id, starcat.t.a, starcat.t.b]             |
|   1 - JOIN TYPE: inner join, EQUAL: [eq(starcat.t.b, starcat.t.b)]                                        |
|   3 - EXCHANGE TYPE: BCJ                                                                                  |
|   4 - STORAGE: LIBRASTORE, BLOCK OFFSET: 1                                                                |
|   5 - JOIN TYPE: inner join, EQUAL: [eq(starcat.t.a, starcat.t.a)]                                        |
|   7 - EXCHANGE TYPE: BCJ                                                                                  |
|   8 - STORAGE: LIBRASTORE, BLOCK OFFSET: 2                                                                |
|   9 - STORAGE: LIBRASTORE, BLOCK OFFSET: 2                                                                |
+-----------------------------------------------------------------------------------------------------------+
```

> **注意：**
> - 此 Hint 只在生成分布式计划时生效，在单机计划时无效。
> - 只读分析引擎优化器会选择 Hash Join 的 Build 端作为 Broadcast 的广播表，若需要进行调整，可结合 HASH_JOIN_BUILD 一起进行调整。

HASH_JOIN_BUILD(t1_name,t2_name…)

使用说明

HASH_JOIN_BUILD(t1_name,t2_name ...)，用于控制只读分析引擎优化器对指定表使用 HASH JOIN 算法，同时将指定表作为 HASH JOIN 算法的 Build 端，即用指定表来构建哈希表。

参考示例

```
EXPLAIN SELECT /*+ HASH_JOIN_BUILD(t2)*/ * FROM t t1, t t2 WHERE t1.a = t2.a;
```

此条 SQL 指定了 t2 表作为 HASH JOIN 中的 Build 表，最终计划如下图所示。

```
+-----------------------------------------------------------------------+
| query plan                                                            |
+-----------------------------------------------------------------------+
| ========================================                              |
| |ID|OPERATOR                |NAME|EST.ROWS|                          |
| ----------------------------------------                              |
| |0 |HASH JOIN               |    |12500.00|                          |
| |1 |├TABLE FULL SCAN (B)    |t2  |10000.00|                          |
| |2 |└TABLE FULL SCAN (P)    |t1  |10000.00|                          |
| ----------------------------------------                              |
| Details:                                                              |
| ----------------------------------                                    |
|   0 - JOIN TYPE: inner join, EQUAL: [eq(singleton.t.a, singleton.t.a)] |
|   1 - STORAGE: LIBRASTORE, BLOCK OFFSET: 1                            |
|   2 - STORAGE: LIBRASTORE, BLOCK OFFSET: 1                            |
+-----------------------------------------------------------------------+
```

除了指定单个表外，还可以指定 JOIN 中间结果作为 BUILD 端，示例如下。

```
EXPLAIN SELECT /*+ HASH_JOIN_BUILD((t1@sel_2,t3@sel_1)) */ * FROM (SELECT t1.a, t1.b FROM t t1, t t2 WHERE
t1.a = t2.a) t1, t t3 WHERE t1.b = t3.b;
```

如上所示，通过将 t1 表与 t2 表用括号括起来，并指定好每个表的 QB_NAME 即可。

```
+-----------------------------------------------------------------------------------------------------------------+
| query plan                                                                                                      |
+-----------------------------------------------------------------------------------------------------------------+
| ================================================                                                                |
| |ID|OPERATOR                        |NAME|EST.ROWS|                                                            |
| ------------------------------------------------                                                                |
| |0 |PROJECTION                      |    |15625.00|                                                            |
| |1 |└HASH JOIN                      |    |15625.00|                                                            |
| |2 |  ├HASH JOIN (B)                |    |12500.00|                                                            |
| |3 |  │ ├TABLE FULL SCAN (B)        |t1  |10000.00|                                                            |
| |4 |  │ └TABLE FULL SCAN (P)        |t2  |10000.00|                                                            |
| |5 |  └TABLE FULL SCAN (P)          |t3  |10000.00|                                                            |
| ------------------------------------------------                                                                |
| Details:                                                                                                        |
| ------------------------------------                                                                            |
|   0 - EXPRS: [singleton.t.a, singleton.t.b, singleton.t.pk, singleton.t.id, singleton.t.a, singleton.t.b]       |
|   1 - JOIN TYPE: inner join, EQUAL: [eq(singleton.t.b, singleton.t.b)]                                          |
|   2 - JOIN TYPE: inner join, EQUAL: [eq(singleton.t.a, singleton.t.a)]                                          |
|   3 - STORAGE: LIBRASTORE, BLOCK OFFSET: 2                                                                      |
|   4 - STORAGE: LIBRASTORE, BLOCK OFFSET: 2                                                                      |
|   5 - STORAGE: LIBRASTORE, BLOCK OFFSET: 1                                                                      |
+-----------------------------------------------------------------------------------------------------------------+
```

HASH_JOIN_PROBE(t1_name,t2_name…)

使用说明

HASH_JOIN_PROBE(t1_name,t2_name ...)，用于控制优化器对指定表使用 HASH JOIN 算法，同时将指定表作为 HASH JOIN 算法的 Probe 端，即用指定表作为 HASH JOIN 的探测（Probe）表。

参考示例

```
EXPLAIN SELECT /*+ HASH_JOIN_PROBE(t2)*/ * FROM t t1, t t2 WHERE t1.a = t2.a;
```

此条 SQL 指定了 t2 表作为 HASH JOIN 中的 Probe 表，最终计划如下图所示。

```
+-------------------------------------------------------------------------+
| query plan                                                              |
+-------------------------------------------------------------------------+
| ========================================                                |
| |ID|OPERATOR                |NAME|EST.ROWS|                             |
| ----------------------------------------                                |
| |0 |HASH JOIN               |    |12500.00|                             |
| |1 |├TABLE FULL SCAN (B)    |t1  |10000.00|                             |
| |2 |└TABLE FULL SCAN (P)    |t2  |10000.00|                             |
| ----------------------------------------                                |
| Details:                                                                |
| ------------------------------------                                    |
|   0 - JOIN TYPE: inner join, EQUAL: [eq(singleton.t.a, singleton.t.a)]  |
|   1 - STORAGE: LIBRASTORE, BLOCK OFFSET: 1                              |
|   2 - STORAGE: LIBRASTORE, BLOCK OFFSET: 1                              |
+-------------------------------------------------------------------------+
```

除了指定单个表外，还可以指定 JOIN 中间结果作为 Probe 端，示例如下。

```
EXPLAIN SELECT /*+ HASH_JOIN_PROBE((t1@sel_2,t3@sel_1)) */ * FROM (SELECT t1.a, t1.b FROM t t1, t t2 WHERE
t1.a = t2.a) t1, t t3 WHERE t1.b = t3.b;
```

如上所示，通过将 t1 表与 t2 表用括号括起来，并指定好每个表的 QB_NAME 即可，最终计划如下图所示。

```
+-----------------------------------------------------------------------------------------------------------------------------------------------+
| query plan                                                                                                                                    |
+-----------------------------------------------------------------------------------------------------------------------------------------------+
| ============================================                                                                                                  |
| |ID|OPERATOR                      |NAME|EST.ROWS|                                                                                                  |
| --------------------------------------------                                                                                                  |
| |0 |PROJECTION                    |    |15625.00|                                                                                                  |
| |1 |└HASH JOIN                    |    |15625.00|                                                                                                  |
| |2 |  ├TABLE FULL SCAN (B)        |t3  |10000.00|                                                                                                  |
| |3 |  └HASH JOIN (P)              |    |12500.00|                                                                                                  |
| |4 |    ├TABLE FULL SCAN (B)      |t1  |10000.00|                                                                                                  |
| |5 |    └TABLE FULL SCAN (P)      |t2  |10000.00|                                                                                                  |
| --------------------------------------------                                                                                                  |
| Details:                                                                                                                                      |
| ------------------------------------                                                                                                          |
|   0 - EXPRS: [singleton.t.a, singleton.t.b, singleton.t.pk, singleton.t.id, singleton.t.a, singleton.t.b]                                     |
|   1 - JOIN TYPE: inner join, EQUAL: [eq(singleton.t.b, singleton.t.b)]                                                                        |
|   2 - STORAGE: LIBRASTORE, BLOCK OFFSET: 1                                                                                                    |
|   3 - JOIN TYPE: inner join, EQUAL: [eq(singleton.t.a, singleton.t.a)]                                                                        |
|   4 - STORAGE: LIBRASTORE, BLOCK OFFSET: 2                                                                                                    |
|   5 - STORAGE: LIBRASTORE, BLOCK OFFSET: 2                                                                                                    |
+-----------------------------------------------------------------------------------------------------------------------------------------------+
```

LEADING(t1_name,t2_name ···)

使用说明

Leading(t1_name,t2_name...)，用于控制优化器在 Join Reorder 阶段生成的 Join Order，优化器会按照 Leading Hint 中出现的顺序来确定 Join Order。

参考示例

```
EXPLAIN SELECT /*+ LEADING(t1,t3,t2,t4)*/ * FROM t1,t2,t3,t4 WHERE t1.a = t2.a and t2.a = t3.a and t3.a =
t4.a;
```

此条 SQL 通过 LEADING Hint 显示指定了 t1 表首先与 t3 表进行 JOIN 操作，接着与 t2 表进行 JOIN，最终与 t4 表进行 JOIN，最终计划如下图所示。

```
+-----------------------------------------------------------------------------------------------------------------------------------------------+
| query plan                                                                                                                                    |
+-----------------------------------------------------------------------------------------------------------------------------------------------+
| ================================================                                                                                              |
| |ID|OPERATOR                         |NAME|EST.ROWS     |                                                                                              |
| ------------------------------------------------                                                                                              |
| |0 |PROJECTION                       |    |156250000.00|                                                                                              |
| |1 |└HASH JOIN                       |    |156250000.00|                                                                                              |
| |2 |  ├TABLE FULL SCAN (B)           |t4  |10000.00     |                                                                                              |
| |3 |  └HASH JOIN (P)                 |    |125000000.00|                                                                                              |
| |4 |    ├TABLE FULL SCAN (B)         |t2  |10000.00     |                                                                                              |
| |5 |    └HASH JOIN (P)               |    |100000000.00|                                                                                              |
| |6 |      ├TABLE FULL SCAN (B)       |t1  |10000.00     |                                                                                              |
| |7 |      └TABLE FULL SCAN (P)       |t3  |10000.00     |                                                                                              |
| ------------------------------------------------                                                                                              |
| Details:                                                                                                                                      |
| ------------------------------------                                                                                                          |
|   0 - EXPRS: [singleton.t1.pk, singleton.t1.id, singleton.t1.a, singleton.t1.b, singleton.t2.pk, singleton.t2.id, singleton.t2.a, singleton.t2.b, singleton.t3.pk,  |
|       singleton.t3.id, singleton.t3.a, singleton.t3.b, singleton.t4.pk, singleton.t4.id, singleton.t4.a, singleton.t4.b]                      |
|   1 - JOIN TYPE: inner join, EQUAL: [eq(singleton.t3.a, singleton.t4.a)]                                                                      |
|   2 - STORAGE: LIBRASTORE, BLOCK OFFSET: 1                                                                                                    |
|   3 - JOIN TYPE: inner join, EQUAL: [eq(singleton.t1.a, singleton.t2.a) eq(singleton.t3.a, singleton.t2.a)]                                   |
|   4 - STORAGE: LIBRASTORE, BLOCK OFFSET: 1                                                                                                    |
|   5 - NON EQUAL: CARTESIAN ,JOIN TYPE: inner join                                                                                             |
|   6 - STORAGE: LIBRASTORE, BLOCK OFFSET: 1                                                                                                    |
|   7 - STORAGE: LIBRASTORE, BLOCK OFFSET: 1                                                                                                    |
+-----------------------------------------------------------------------------------------------------------------------------------------------+
```

上述方法指定 JOIN 顺序只能生成 Left Deep Tree，只读分析引擎还为 LEADING 提供了一种进阶语法，可以通过括号配合生成 Bushy Tree，示例如下。

```
EXPLAIN SELECT /*+ LEADING((t1,t3),(t2,t4))*/ * FROM t1,t2,t3,t4 WHERE t1.a = t2.a and t2.a = t3.a and
t3.a = t4.a;
```

此条 SQL 的 LEADING Hint 首先通过将 (t1,t3) 控制优化器先进行 t1 表和 t3 表的 JOIN，之后将通过 (t2,t4) 控制优化器进行 t2 表与 t4 表的 JOIN，最后通过 ((t1,t3),(t2,t4)) 控制优化器将上述两个 JOIN 操作的结果进行 JOIN，最终执行计划如下图所示。

版权所有：腾讯云计算（北京）有限责任公司

```
+-----------------------
```

```
+---------+------+-----------------------------------------------------------------------------------------
-----------------------------------------------------------------+
| Level   | Code | Message
|
+---------+------+-----------------------------------------------------------------------------------------
-----------------------------------------------------------------+
| Warning | 1815 | There are no matching table names for (t1) in optimizer hint /*+ SHUFFLE_JOIN(t1) */ or
/*+ SHUFFLE_JOIN(t1) */. Maybe you can use the table alias name |
+---------+------+-----------------------------------------------------------------------------------------
-----------------------------------------------------------------+
1 row in set (0.00 sec)
```

QB_NAME 未指定/指定错误，导致 Hint 不生效

对于有多个 QB_NAME 的查询，如果 Hint 没有写在目标表的 Query Block 中，则需要显示的在 Hint 中指定 QB_NAME，若没有指定，则可能发生 Query Hint 不生效的问题。例如：

```
SELECT /*+ HASH_JOIN_PROBE(t2) */ * FROM (SELECT t1.a, t1.b FROM t t1, t t2 WHERE t1.a = t2.a) t1, t t3
WHERE t1.b = t3.b;
```

由于 Hint 中的 t1表没有显示指定 QB_NAME，导致 Hint 没有生效，Warning 信息如下。

```
mysql> show warnings;
+---------+------+-----------------------------------------------------------------------------------------
-----------------------------------------+
| Level   | Code | Message
|
+---------+------+-----------------------------------------------------------------------------------------
-----------------------------------------+
| Warning | 1815 | There are no matching table names for (t2) in optimizer hint /*+ HASH_JOIN_PROBE(t2)
*/. Maybe you can use the table alias name |
+---------+------+-----------------------------------------------------------------------------------------
-----------------------------------------+
1 row in set (0.00 sec)
```

此时可通过关键字查询 SQL 计划，确认每个表所在 Query Block 的 QB_NAME。

```
+-----------------------------------------------------------------------------------------------------+
| query plan                                                                                          |
+-----------------------------------------------------------------------------------------------------+
| ================================================                                                  |
| |ID|OPERATOR                    |NAME|EST.ROWS|                                                    |
| ------------------------------------------------                                                    |
| |0 |PROJECTION                  |    |15625.00|                                                    |
| |1 |└─HASH JOIN                 |    |15625.00|                                                    |
| |2 |  ├─TABLE FULL SCAN (B)     |t3  |10000.00|                                                    |
| |3 |  └─HASH JOIN (P)           |    |12500.00|                                                    |
| |4 |    ├─TABLE FULL SCAN (B)   |t1  |10000.00|                                                    |
| |5 |    └─TABLE FULL SCAN (P)   |t2  |10000.00|                                                    |
| ------------------------------------------------                                                    |
| Details:                                                                                            |
| -------------------------------------                                                               |
|   0 - EXPRS: [singleton.t.a, singleton.t.b, singleton.t.pk, singleton.t.id, singleton.t.a, singleton.t.b] |
|   1 - JOIN TYPE: inner join, EQUAL: [eq(singleton.t.b, singleton.t.b)]                              |
|   2 - STORAGE: LIBRASTORE, BLOCK OFFSET: 1                                                          |
|   3 - JOIN TYPE: inner join, EQUAL: [eq(singleton.t.a, singleton.t.a)]                              |
|   4 - STORAGE: LIBRASTORE, BLOCK OFFSET: 1                                                          |
|   5 - STORAGE: LIBRASTORE, BLOCK OFFSET: 2                                                          |
+-----------------------------------------------------------------------------------------------------+
```

Hint 位置不正确导致不生效

如果没有按照 Optimizer Hints 语法将 Hint 正确地放在指定关键字的后面，它将无法生效。例如：

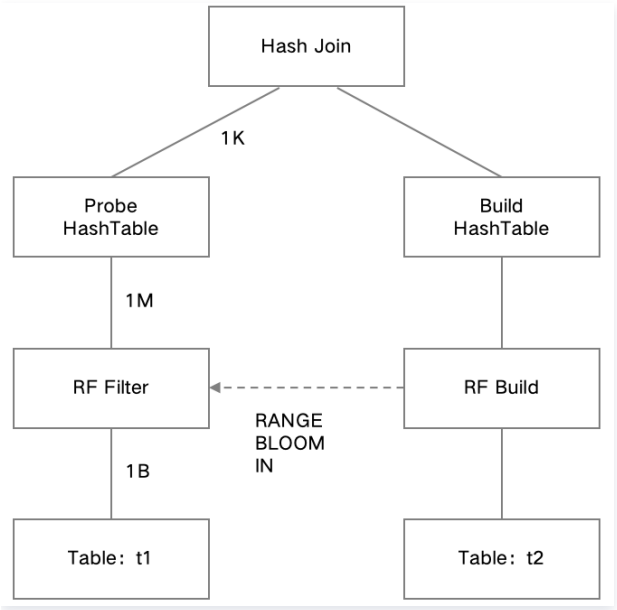
```
MySQL> SELECT * /*+ SET_VAR(max_threads = 64)) */ FROM t;
```

```
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL
server version for the right syntax to use line 1 column 42 near "/*+ SET_VAR(max_threads = 64)) */ FROM
t"
```

Runtime Filter 使用手册

最近更新时间：2024-08-02 16:34:02

HASH JOIN 是一种数据库中常见的数据库连接算法，它通过使用哈希表来加速连接过程，通常其分为 Build 和 Probe 两个阶段，当 Probe 阶段数据量比较大且输出的数据比较少时，可以开启 Runtime Filter 来提前过滤一部分数据从而提升性能。

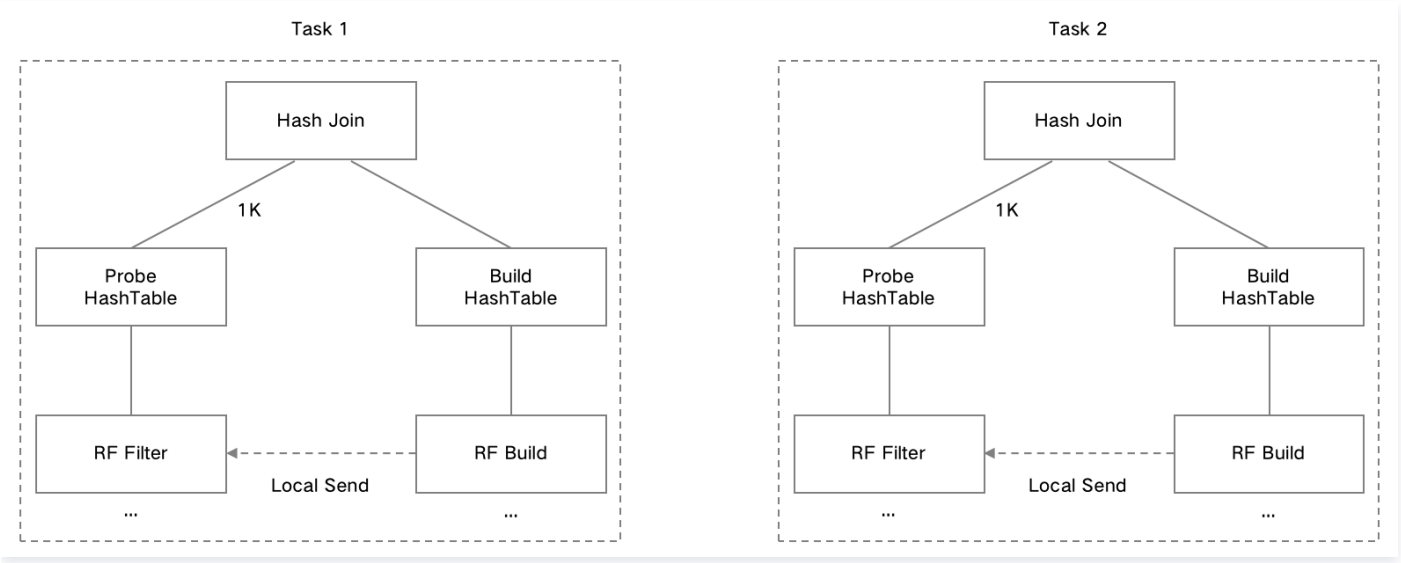


只读分析引擎的 Runtime Filter 由两部分构成，分别是 RF Build 和 RF Filter。RF Build 是在 HASH JOIN 的 Build 端，用来构建 Runtime Filter。RF Filter则在对应 HASH JOIN 表的 Probe 端的 TableScan 上，用于提前过滤数据以提升性能。

Runtime Filter 类型

Local Runtime Filter

Local Runtime Filter 通常用于数据 JOIN 的时候没有被 Shuffle 的场景。此时，当前构建节点的 Runtime Filter 已经可以满足 Probe 的需要，就无须网络传输，直接将 Runtime Filter 的数据传递给 Probe 侧即可使用。

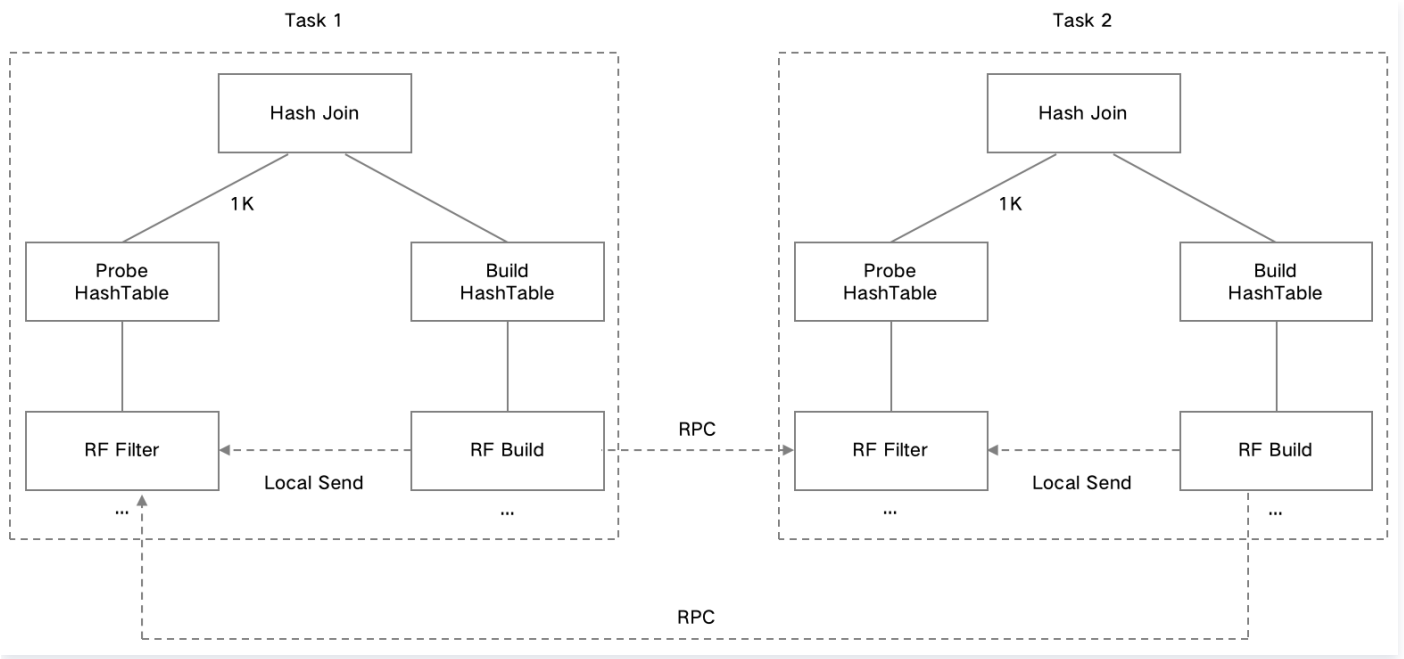


如上图所示，当我们 JOIN 时，Build 表没有被 Shuffle 发送的时候，那么在同一个计划下的 Runtime Filter Build 算子将自己的数据构建发送给当前计划中对应的 Filter Probe 部分。

Global Runtime Filter

当 JOIN 数据被 Shuffle 到不同节点上进行构建时，这时候当前构建节点所构建的 Runtime Filter 不能够满足过滤要求。这种情况下就需要接受其他节点的 Runtime Filter，将所有构建节点的 Runtime Filter 合并完成之后，再进行使用。

版权所有：腾讯云计算（北京）有限责任公司



当我们 JOIN 时，Build 表的数据是被 Shuffle 出去的时候，当前计划中的 Runtime FIlter Build 算子构建的数据就不是完整的，Runtime Filter 就必须接收除了当前计划的算子构建的数据以及其他相同计划构建的数据进行合并后才能使用。

过滤器类型

在过滤器算法的选择上，通常会按照数据分布的情况选择以下一个或者多个过滤算法。

- **Bloom Filter**

  Bloom Filter 作为一个经典的过滤器算法，能够通过几个 HASH 函数来决定数据是否存在。Runtime Filter 中布隆过滤器的大小通常取决于数据的 NDV。当然，布隆过滤器会出现一些误判（False Negative），将需要过滤的数据没有正确过滤，但是这些数据在 JOIN 的 Probe 阶段也会被过滤掉。
- **MIN_MAX Filter**

  MIN_MAX Filter 收集 Build 端数据的最大值和最小值，在过滤的时候就判断数据是否在这个区间内，如果不在就需要过滤掉。当 Build 端的数据是按照某一个区间进行分布的时候，MIN_MAX Filter 的过滤效果会比较好。
- **IN Filter**

  IN Filter 针对的是 NDV 值比较小的场景，这种场景下直接将这个列所有的值发送到探测端进行匹配。

只读分析引擎中的 Runtime Filter

开启或关闭 Runtime Filter

默认情况下，只读分析引擎的 Runtime Filter 处于开启状态，可以使用下列开关开启或关闭。

```
mysql> set libra_enable_runtime_filter=ON;
mysql> set libra_enable_runtime_filter=OFF;
```

开启之后，优化器会对 JOIN 进行评估，如果符合条件的情况就会自动启用 Runtime Filter。
如果需要强制所有的 JOIN 都开启 Runtime Filter，可以在上述参数的基础之上配置如下参数。

```
mysql>SET libra_enable_cost_based_runtime_filter=OFF;
```

Runtime Filter 计划

如下所示，这是一个 Local RF 的计划，在 JOIN 上分配了三种 Runtime Filter，这个场景中 HASH JOIN 的 Build 侧和 Probe 侧之间没有数据的重分布。

```
mysql> explain select * from t1, t2 where t1.v1 = t2.v1;
+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+
| query plan                                                                                                                                                                                    |
+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+
| ============================================                                                                                                                                                  |
| |ID|OPERATOR                    |NAME|EST.ROWS|                                                                                                                                                                  |
| --------------------------------------------                                                                                                                                                  |
| |0 |HASH JOIN                   |    |12487.50|                                                                                                                                                                  |
| |1 | ├─EXCHANGE RECEIVER (B)    |    |9990.00 |                                                                                                                                                                  |
| |2 | │ └─EXCHANGE SENDER        |    |9990.00 |                                                                                                                                                                  |
| |3 | │   └─FILTER               |    |9990.00 |                                                                                                                                                                  |
| |4 | │     └─TABLE FULL SCAN    |t1  |10000.00|                                                                                                                                                                  |
| |5 | └─FILTER (P)               |    |9990.00 |                                                                                                                                                                  |
| |6 |   └─TABLE FULL SCAN        |t2  |10000.00|                                                                                                                                                                  |
| --------------------------------------------                                                                                                                                                  |
| Details:                                                                                                                                                                                      |
| ------------------------------------                                                                                                                                                          |
|   0 - JOIN TYPE: inner join, EQUAL:[eq(test.t1.v1, test.t2.v1)], Build Runtime Filters:  [0 [MIN_MAX,BLOOM_FILTER,IN_FILTER]  [ NDV: 7992]  <- test.t1.v1]                                   |
|   2 - EXCHANGE TYPE: BCJ                                                                                                                                                                      |
|   3 - CONDITIONS: [not(isnull(test.t1.v1))]                                                                                                                                                   |
|   4 - STORAGE: LIBRASTORE                                                                                                                                                                     |
|   5 - CONDITIONS: [not(isnull(test.t2.v1))]                                                                                                                                                   |
|   6 - STORAGE: LIBRASTORE, Probe Runtime Filters: [0 [MIN_MAX,BLOOM_FILTER,IN_FILTER]  [ NDV: 8000]  [ SELECTIVITY: 0.001000]  -> test.t2.v1 Mode: LOCAL]                                   |
+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+
19 rows in set (0.00 sec)
```

而下面所示的计划则是 Global RF，在 Build 侧和 Probe 侧之间存在数据的重分布，RF 可以在数据被网络传输之前提前过滤，可以减少网络传输和后续的 JOIN 开销从而获得性能提升。

```
mysql> explain select /*+ shuffle_join(t2)*/ * from t1, t2 where t1.v1 = t2.v1;
+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+
| query plan                                                                                                                                                                                    |
+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+
| ============================================                                                                                                                                                  |
| |ID|OPERATOR                    |NAME|EST.ROWS|                                                                                                                                                                  |
| --------------------------------------------                                                                                                                                                  |
| |0 |HASH JOIN                   |    |12487.50|                                                                                                                                                                  |
| |1 | ├─EXCHANGE RECEIVER (B)    |    |9990.00 |                                                                                                                                                                  |
| |2 | │ └─EXCHANGE SENDER        |    |9990.00 |                                                                                                                                                                  |
| |3 | │   └─FILTER               |    |9990.00 |                                                                                                                                                                  |
| |4 | │     └─TABLE FULL SCAN    |t1  |10000.00|                                                                                                                                                                  |
| |5 | └─EXCHANGE RECEIVER (P)    |    |9990.00 |                                                                                                                                                                  |
| |6 |   └─EXCHANGE SENDER        |    |9990.00 |                                                                                                                                                                  |
| |7 |     └─FILTER               |    |9990.00 |                                                                                                                                                                  |
| |8 |       └─TABLE FULL SCAN    |t2  |10000.00|                                                                                                                                                                  |
| --------------------------------------------                                                                                                                                                  |
| Details:                                                                                                                                                                                      |
| ------------------------------------                                                                                                                                                          |
|   0 - JOIN TYPE: inner join, EQUAL:[eq(test.t1.v1, test.t2.v1)], Build Runtime Filters:  [0 [MIN_MAX,BLOOM_FILTER,IN_FILTER]  [ NDV: 7992]  <- test.t1.v1]                                   |
|   2 - EXCHANGE TYPE: HASH, Hash Cols: [name: test.t1.v1, collate: N/A]                                                                                                                        |
|   3 - CONDITIONS: [not(isnull(test.t1.v1))]                                                                                                                                                   |
|   4 - STORAGE: LIBRASTORE                                                                                                                                                                     |
|   6 - EXCHANGE TYPE: HASH, Hash Cols: [name: test.t2.v1, collate: N/A]                                                                                                                        |
|   7 - CONDITIONS: [not(isnull(test.t2.v1))]                                                                                                                                                   |
|   8 - STORAGE: LIBRASTORE, Probe Runtime Filters: [0 [MIN_MAX,BLOOM_FILTER,IN_FILTER]  [ NDV: 8000]  [ SELECTIVITY: 0.001000]  -> test.t2.v1 Mode: GLOBAL]                                   |
+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+
22 rows in set (0.00 sec)
```

调整 Runtime Filter 参数

Runtime Filter 可以对以下参数进行调整。

- libra_enable_runtime_filter 表示是否打开 Runtime Filter。

属性	描述
参数类型	BOOL。
默认值	ON。
取值范围	ON：打开 Runtime Filter。 OFF：关闭 Runtime Filter。
作用域	Global & Session。
支持 SET_VAR Hint	是。

- libra_runtime_filter_type 表示设置可以分配的 Runtime Filter 类型。

属性	描述
参数类型	VARCHAR。
默认值	MIN_MAX，BLOOM_FILTER，IN_FILTER。
取值范围	BLOOM_FILTER：构建 JOIN BUILD 端 JOIN KEY 的 Bloom Filter 来进行 Probe 端的数据过滤。 MIN_MAX：构建 JOIN BUILD 端 JOIN KEY 的最大最小值来进行 Probe 端的数据过滤。 IN：构建 JOIN BUILD 端 JOIN KEY 的值列表来进行 Probe 端的数据过滤。 空字符串：表示关闭 Runtime Filter 功能。
作用域	Global & Session。
支持 SET_VAR Hint	是。

- libra_enable_cost_based_runtime_filter 表示是否开启基于代价的 Runtime Filter 分配，关闭后默认会生成所有的 Runtime Filter。

属性	描述
参数类型	BOOL。
默认值	ON。
取值范围	ON：开启基于代价的 Runtime Filter 分配。 OFF：关闭基于代价的 Runtime Filter 分配。
作用域	Global & Session。
支持 SET_VAR Hint	是。

- libra_max_in_runtime_filter_ndv 表示在基于代价的 Runtime Filter 中，生成 IN 类型的 Runtime Filter 时，BUILD 端最大的 NDV（Number of Distinct Values）值。

属性	描述
参数类型	INT。
默认值	1024。
取值范围	0 – MaxValue。
作用域	Global & Session。
支持 SET_VAR Hint	是。

延迟物化

最近更新时间：2024-10-31 16:56:11

什么是延迟物化

在数据库执行 SQL 查询时，若无法命中主键或索引键，将不得不进行全表扫描（TableScan），这种全量数据扫描的代价通常较高。为了优化这一过程，可以采用延迟物化（Lazy Materialization）技术，该技术通过推迟数据的物化，直到查询执行阶段才进行必要的计算和存储，进而提升查询性能和系统响应速度。在只读分析引擎执行 SQL 查询的过程中，首先会读取谓词列，并利用 Filter 算子进行计算，从而得到过滤后的结果。随后，只读分析引擎会根据这些过滤结果物化其他需要读取的列数据。这种方法有效减少了非谓词列在大量过滤数据上的读取量，从而提高了扫描速度。

延迟物化的优势

- 在只读分析引擎中数据是被压缩过的，物化的过程就必须对数据进行解压，通过延迟物化技术可以减少解压的数据范围，从而减轻解压缩所带来的 CPU 开销。
- 当 SQL 查询涉及多个列时，早期物化可能会导致数据库读取并组合所有列的数据，尽管其中一些列可能并不会用于最终结果的查询。相比之下，延迟物化策略则推迟了数据的组合过程，使得数据库仅读取和处理那些真正需要的列数据。这种做法有效减少了不必要的 I/O 操作，从而提高了查询效率。
- 延迟物化可以让过滤、聚合等操作更加高效，因为这些操作只需要处理列数据，而不需要处理整行数据。列存储的特性（如数据压缩、批量处理）能够更好地发挥作用，从而加快查询执行速度。

延迟物化相关参数

延迟物化功能可以通过参数 libra_enable_late_materialization 进行开关，参数值为 ON，表示已经开启延时物化功能。

属性	描述
参数类型	BOOL。
默认值	ON。
取值范围	ON：打开延迟物化功能。 OFF：关闭延迟物化功能。
作用域	Global & Session。
支持 SET_VAR Hint	是。

```
#在 session 级别关闭延迟物化
set libra_enable_late_materialization=off;
#在 session 级别开启延迟物化
set libra_enable_late_materialization=on;
```

延迟物化示例

如下图所示，当开启延迟物化时，可以从执行计划清晰的看到 COLUMN READ 算子。此算子即为延迟物化功能生效的示例。

```
mysql> explain select * from t where c1 = 1 and c2 = 1;
+-----------------------------------------------------------------------+
| query plan                                                            |
+-----------------------------------------------------------------------+
| ========================================                              |
| |ID|OPERATOR             |NAME|EST.ROWS|                             |
| ----------------------------------------                              |
| |0 |FILTER               |    |0.00    |                             |
| |1 |└─TABLE FULL SCAN    |t   |10000.00|                             |
| ----------------------------------------                              |
| Details:                                                              |
| ------------------------------------                                  |
|   0 - CONDITIONS: [eq(starcat.t.c1, 1), eq(starcat.t.c2, 1)]          |
|   1 - STORAGE: LIBRASTORE; BLOCK OFFSET: 1                            |
+-----------------------------------------------------------------------+
10 rows in set (0.00 sec)

mysql> set libra_enable_late_materialization = on;
Query OK, 0 rows affected (0.00 sec)

mysql> explain select * from t where c1 = 1 and c2 = 1;
+-----------------------------------------------------+
| query plan                                          |
+-----------------------------------------------------+
| ========================================            |
| |ID|OPERATOR                 |NAME|EST.ROWS|      |
| ----------------------------------------            |
| |0 |FILTER                   |    |0.00    |      |
| |1 |└─COLUMN READ            |t   |10000.00|      |
| |2 |  └─FILTER               |    |1.00    |      |
| |3 |    └─TABLE FULL SCAN    |t   |10000.00|      |
| ----------------------------------------            |
| Details:                                            |
| ------------------------------------                |
|   0 - CONDITIONS: [eq(starcat.t.c2, 1)]             |
|   1 - COLUMN READ:[c2,c3,c4,c5]                     |
|   2 - CONDITIONS: [eq(starcat.t.c1, 1)]             |
|   3 - STORAGE: LIBRASTORE; BLOCK OFFSET: 1          |
+-----------------------------------------------------+
14 rows in set (0.01 sec)
```

如何在 HINT 中设置延迟物化

通过 SET_VAR Hint 可以在单条 SQL 语句中指定开启或关闭延迟物化，示例如下。

```
select /*+ set_var(libra_enable_late_materialization=1)*/ * from t where c1=1 and c2=1;
```

```
mysql> explain select /*+ set_var(libra_enable_late_materialization=1)*/ * from t where c1 = 1 and c2 = 1;
+-----------------------------------------------------+
| query plan                                          |
+-----------------------------------------------------+
| ========================================            |
| |ID|OPERATOR                 |NAME|EST.ROWS|      |
| ----------------------------------------            |
| |0 |FILTER                   |    |0.00    |      |
| |1 |└─COLUMN READ            |t   |10000.00|      |
| |2 |  └─FILTER               |    |1.00    |      |
| |3 |    └─TABLE FULL SCAN    |t   |10000.00|      |
| ----------------------------------------            |
| Details:                                            |
| ------------------------------------                |
|   0 - CONDITIONS: [eq(starcat.t.c2, 1)]             |
|   1 - COLUMN READ:[c2,c3,c4,c5]                     |
|   2 - CONDITIONS: [eq(starcat.t.c1, 1)]             |
|   3 - STORAGE: LIBRASTORE; BLOCK OFFSET: 1          |
+-----------------------------------------------------+
14 rows in set (0.00 sec)
```

Merge Join

最近更新时间：2025-02-11 16:41:53

什么是 Merge Join

Merge Join 一般又叫做 Sort Megre Join，这是多表连接查询中常用的一种 Join 方式。具体而言，它会先将关联表的关联列各自做排序，然后从各自的排序表中抽取数据，并将这些数据与另一个排序表中的数据进行匹配。
因为 Merge Join 需要做更多的排序，所以消耗的资源也更多。通常来讲，能够使用 Merge Join 的地方，Hash Join 都可以发挥更好的性能，即 Hash Join 的效果都比 Merge Join 要好。但如果连接键已经有序，在执行 Merge Join 时则不需要再排序了，这时 Merge Join 的性能会优于 Hash Join。Merge Join 的性能优势在此时就会得到有效的体现。
LibraDB 引擎在之前版本中，仅支持 Hash Join，而在2.2410.1.0及之后的版本中，列存引擎也已支持了基于主键的 Merge Join 能力。

适用场景

Join 键为两表的主键的场景下进行等价关联（A.a=B.a）有着明显效果。

使用方法

目前仅支持通过指定 Hint 的方式使用 Merge Join，如下命令所示。

```
select /*+ merge_join(lineitem,customer) */ * from orders o,lineitem l where o.o_orderkey=l.l_orderkey;
```

列存二级索引

最近更新时间：2025-03-18 17:38:12

索引介绍

索引是数据库查询加速的一项重要能力。为了充分满足不同用户的查询需求、提升整体数据库性能，只读分析引擎全面支持了基于列存的二级索引能力。目前只读分析引擎支持的索引有三种类型：Zonemap Index、Bloom Filter Index、Bitmap Index。
通常来说，建立索引对高基数列且低选择率的谓词查询，能大幅减少查询数据量，从而显著优化查询速度。

> ⚠ **注意：**
> 目前仅2.2410.4.0及以后的版本支持索引创建语句的能力，同时创建索引需要用户具备 INDEX 权限，如需使用该能力，请先为账号添加此权限。

Zonemap Index

Zonemap Index 属于系统内置索引，用户无需对其特别关注。它会自动维护每一列的统计信息，会针对每一个数据块记录其最大值、最小值以及是否有 NULL 等信息。
对于等值查询、范围查询、IS NULL 这些场景，可以使用最大值、最小值等信息来判断数据文件和数据块是否包含满足条件的数据。如果不包含则跳过不读对应的文件或数据块。通过这种方式，能够减少不必要的 I/O 操作，从而有效加速查询过程。

Bloom Filter Index

Bloom Filter Index 是基于 Bloom Filter 的一种跳数索引。它的原理是利用 Bloom Filter 跳过等值查询指定条件不满足的数据块，以达到减少 I/O，查询加速的效果。
Bloom Filter 是由 Bloom 在1970年提出的一种多哈希函数映射的快速查找算法。通常应用在一些需要快速判断某个元素是否属于集合，但是并不严格要求100%正确的场景，Bloom Filter 具备以下特点：

- 空间效率高的概率型数据结构，用来检查一个元素是否在一个集合中。
- 对于一个元素检测是否存在的调用，Bloom Filter 会告诉调用者两个结果之一：可能存在或者一定不存在。

适用场景

Bloom Filter Index 能够对等值查询（包括 = 和 IN ）加速，对高基数字段效果比较好。

限制条件

- Bloom Filter Index 对 = 和 IN 之外的查询没有效果，比如 !=，NOT INT，>，< 等。
- Bloom Filter Index 仅支持：最大长度为256的 INT 类型、String 类型、最大长度为256的 Decimal 类型、Time、Date、DateTime 字段类型。
- 不支持对表达式创建索引，也不支持多列联合索引。
- 单主键列或者多字段联合主键中的第一列不支持创建 Bloom Filter Index。

使用索引

在执行 SQL 时，如果 where 条件中的等值谓词或者 IN 谓词中的字段创建了 Bloom Filter Index，那么查询时将自动应用索引进行查询加速。

相关命令

- 索引创建语句

```
CREATE INDEX IF NOT EXISTS 索引名称 USING BLOOM_FILTER ON 表名(列名);
```

- 索引删除语句

```
DROP INDEX 索引名称 ON 表名;
```

- 索引查询语句

```
SHOW CREATE TABLE 表名;
```

Bitmap Index

Bitmap Index 是用位图表示的索引，对列的每个键值建立一个位图。相对于其它索引，Bitmap Index 的优点是占用的存储空间非常小，创建和使用非常快，缺点是修改操作锁粒度大，不适合频繁更新的场景。

适用场景

- 适合建在值重复度高的列上，建议在100到100000之间，例如：职业、地级市等。重复度过高则对比其他类型的索引没有明显优势，重复度过低，则空间效率和性能会大大降低。
- 特定类型的查询例如 count、or、and 等逻辑操作只需要进行位运算。例如：通过多个条件组合查询，`select count(*) from table where city = '南京市' and job = '医生' and Type = 'iphone' and gender ='男'` 。类似这种场景，如果在每个查询条件列上都建立了 Bitmap Index，则数据库可以进行高效的 bit 运算，能够精确定位到需要的数据，减少磁盘 I/O，并且筛选出的结果集越小，Bitmap Index 的优势越明显。
- 适用于即席查询、多维分析等分析场景。如果有一张表有100列，用户会使用其中的20个列作为查询条件（任意使用这20个列上的 N 的列），在这些列上创建20个 Bitmap Index，那么所有的查询都可以应用到索引。

不适用场景

- 值重复度低的列，例如：身份证号码、手机号码等。
- 重复度过高的列，例如：性别，可以建立 Bitmap Index，但不建议单独作为查询条件使用，建议与其他条件共同过滤。
- 经常需要更新修改的列。

限制条件

- Bitmap Index 支持 =、!=、>、<、>=、<=、in、is null、is not null 等表达式，但多个谓词间只能是 and 连接。
- Bitmap Index 仅支持：最大长度为256的 INT 类型、String 类型、最大长度为256的 Decimal 类型、Time、Date、DateTime 字段类型。
- 不支持对表达式创建索引，也不支持多列联合索引。
- 单主键列或者多字段联合主键中的第一列不支持创建 Bitmap Index。

相关命令

- 索引创建语句

```
CREATE INDEX IF NOT EXISTS 索引名称 USING BITMAP ON 表名(列名);
```

- 索引删除语句

```
DROP INDEX 索引名称 ON 表名;
```

- 索引查询语句

```
SHOW CREATE TABLE 表名;
```

系统变量

最近更新时间：2025-02-13 14:23:13

本文详细列举了在 SQL HINT 中可以使用 set_var 进行指定的变量列表。

Runtime Filter 相关

系统变量	含义	参数类型	默认值	取值范围	作用域	支持 SET_VAR Hint
libra_enable_runtime_filter	是否打开 Runtime Filter。	BOOL	ON	ON：打开 Runtime Filter OFF：关闭 Runtime Filter	Global & Session	是
libra_runtime_filter_type	设置可以分配的 Runtime Filter 类型。	VARCHAR	MIN_MAX，BLOOM_FILTER，IN_FILTER	BLOOM_FILTER：构建 JOIN BUILD 端 JOIN KEY 的 Bloom Filter 来进行 Probe 端的数据过滤。 MIN_MAX：构建 JOIN BUILD 端 JOIN KEY 的最大最小值来进行 Probe 端的数据过滤。 IN：构建 JOIN BUILD 端 JOIN KEY 的值列表来进行 Probe 端的数据过滤。 空字符串：表示关闭 Runtime Filter 功能。	Global & Session	是
libra_enable_cost_based_runtime_filter	开启/关闭基于代价的 Runtime Filter。	BOOL	ON	ON： 开启基于代价的 Runtime Filter 分配。 OFF：关闭基于代价的 Runtime Filter 分配。	Global & Session	是
libra_max_in_runtime_filter_ndv	在基于代价的 Runtime Filter 中，生成 IN 类型的 Runtime Filter 时，BUILD 端最大的 NDV（Number of Distinct Values）值。	INT	1024	0 – MaxValue	Global & Session	是
runtime_filter_wait_time_ms	表示在 RF USE 侧首次等待 RF Ready 的最大时间，0表示无限等待直到超时或中断 Query 执行。	INT	100	[0，MaxValue)	Global & Session	是

Late materialization 相关

系统变量	含义	参数类型	默认值	取值范围	作用域	支持 SET_VAR Hint
libra_enable_late_materialization	是否打开延时物化。	BOOL	OFF	ON：打开延时物化 OFF：关闭延时物化	Global & Session	是

并行&并发相关

系统变量	含义	参数类型	默认值	取值范围	作用域	支持 SET_VAR Hint
max_threads	查询在执行时的并行度。	INT	节点的 CPU 核数	0 - MaxValue（建议不超过 CPU 核数的两倍）	Global & Session	是

执行引擎相关

系统变量	含义	参数类型	默认值	取值范围	作用域	支持 SET_VAR Hint
max_bytes_before_external_agg_uniq_exact	count distinct 使用计算函数 uniqExactDisk 的时候，hashset 超过这个限制值时开始切外存，默认为0，不做外存计算。	INT	0	[0，MaxValue)	Global & Session	是
one_bucket_max_temp_file_size	count distinct 使用计算函数 uniqExactDisk 的时候，每个 bucket 落盘文件最大值，超过这个值需要切换新的文件写入。	INT	0	[0，MaxValue)	Global & Session	是
count_distinct_implementation	count distinct 计算使用函数的名称，默认使用 uniqExactDisk。	VARCHAR	uniqExactDisk	uniqExactDisk：带落盘功能的精准 CountDistinct 计算方法。 uniq：CountDistinct 近似计算方法。 uniqExact：纯内存的精准 CountDistinct 计算方法。	-	是
join_build_concurrency	指定用于 Join Build 所使用的并行度，默认为0，代表使用 Default 或者指定 MaxThreads 。	INT	0	[0，MaxValue)	Global & Session	是
enable_local_tunnel	支持本机的 Sender 和 Receiver 的通信优化。	BOOL	ON	ON：开启 Local Channel 本机通信优化。 OFF：关闭 Local Channel 本机通信优化。	Global & Session	是
mpp_max_packet_size	Sender 发送的数据包最大大小，0代表未限制。	INT	0	[0，MaxValue)	Global & Session	是
tunnel_buffer_size	Sender 端缓冲发送队列 BUFFER 的数量，默认为1。	INT	1	[0，MaxValue)	Global & Session	是

版权所有：腾讯云计算（北京）有限责任公司

pipeline_executor_use_thread_manager	Pipeline 模型使用动态线程池模型。	BOOL	OFF	ON：开启 Pipeline 模型使用动态线程池。 OFF：关闭 Pipeline 模型使用动态线程池。	Global & Session	是
enable_order_by_push_down	开启/关闭 order by 下推功能。	BOOL	ON	ON：开启 order by 下推功能。 OFF：关闭 order by 下推功能。	Global & Session	是
max_block_size	设置 Pipeline 执行引擎中所传递数据 Block 的行数。	INT	65409	[1，MaxValue)	Global & Session	是
shuffle_after_join	设置 join 后是否将数据打散到多个 Pipeline。	BOOL	ON	ON：开启打散功能。 OFF：关闭打散功能。	Global & Session	是

系统表
SLOW_LOG

最近更新时间：2025-05-14 14:39:13

功能

SLOW_LOG 是记录慢 SQL 的系统表。在只读分析引擎中 SQL 执行时间超过300ms的 SQL 将被记录到此系统表中。默认保留7天的慢 SQL 执行信息。

查询示例

```
select * from libra_system.slow_log;
```

字段说明

字段名称	类型	是否可以为 NULL	描述
start_time	text	NO	SQL 开始执行时间。
tso	bigint(20) unsigned	NO	SQL 的唯一标志。
sql	text	NO	SQL 文本信息（最长4KB，超过将截断）。
duration	bigint(20) unsigned	NO	SQL 的执行时间（单位 ms）。
details	text	NO	SQL 执行详细信息。
conn_id	bigint(20) unsigned	NO	客户端连接的 Session ID。
user	text	NO	执行用户名。
db	text	NO	涉及数据库。
digest	text	NO	保留列名（暂时未使用）。
succ	tinyint(3) unsigned	NO	SQL 是否执行成功。
host	text	NO	SQL 执行节点信息。

版权所有：腾讯云计算（北京）有限责任公司

性能白皮书
TPC-H 测试方法

最近更新时间：2025-05-21 17:47:22

TDSQL-C MySQL 版的只读分析引擎主要针对复杂的 SQL 和数据分析而设计。对于此类场景的性能评估手段在业界常用 TPC-H 的基准测试集来进行。因此，本文主要基于 TPC-H 基准来测试只读分析引擎的性能情况，以供用户参考。

前提条件

- 准备好实例形态为预置资源的集群，请参见 创建集群 。
- 集群状态为运行中，并创建只读分析引擎实例。
- 准备一个数据库账号，用于数据查询，请参见 创建账号 。
- 为集群中的只读分析引擎实例设置一个合理的实例规格。

> **说明：**
> 测试结果与实例计算规格的大小成正比。在数据分析场景中，想要性能好，则需要更多的资源支撑，可根据实际的情况选用对应的实例规格。

构建测试表

TPC-H 测试集中包含有八张数据表与一个视图，请根据以下创建语句，将表创建在 TDSQL-C MySQL 版集群中。

```
create database tpch;
use tpch;

drop table if exists customer;
create table `customer` (
  `c_custkey` bigint(20) not null,
  `c_name` varchar(25) not null,
  `c_address` varchar(40) not null,
  `c_nationkey` bigint(20) not null,
  `c_phone` char(15) not null,
  `c_acctbal` decimal(15,2) not null,
  `c_mktsegment` char(10) not null,
  `c_comment` varchar(117) not null,
  primary key (`c_custkey`)
) engine=innodb default charset=utf8mb4 collate=utf8mb4_bin;

drop table if exists lineitem;
create table `lineitem` (
  `l_orderkey` bigint(20) not null,
  `l_partkey` bigint(20) not null,
  `l_suppkey` bigint(20) not null,
  `l_linenumber` bigint(20) not null,
  `l_quantity` decimal(15,2) not null,
  `l_extendedprice` decimal(15,2) not null,
  `l_discount` decimal(15,2) not null,
  `l_tax` decimal(15,2) not null,
  `l_returnflag` char(1) not null,
  `l_linestatus` char(1) not null,
  `l_shipdate` date not null,
  `l_commitdate` date not null,
  `l_receiptdate` date not null,
  `l_shipinstruct` char(25) not null,
  `l_shipmode` char(10) not null,
  `l_comment` varchar(44) not null,
  primary key (`l_orderkey`,`l_linenumber`)
) engine=innodb default charset=utf8mb4 collate=utf8mb4_bin;
```

版权所有：腾讯云计算（北京）有限责任公司

```
drop table if exists nation;
create table `nation` (
  `n_nationkey` bigint(20) not null,
  `n_name` char(25) not null,
  `n_regionkey` bigint(20) not null,
  `n_comment` varchar(152) default null,
  primary key (`n_nationkey`)
) engine=innodb default charset=utf8mb4 collate=utf8mb4_bin;

drop table if exists orders;
create table `orders` (
  `o_orderkey` bigint(20) not null,
  `o_custkey` bigint(20) not null,
  `o_orderstatus` char(1) not null,
  `o_totalprice` decimal(15,2) not null,
  `o_orderdate` date not null,
  `o_orderpriority` char(15) not null,
  `o_clerk` char(15) not null,
  `o_shippriority` bigint(20) not null,
  `o_comment` varchar(79) not null,
  primary key (`o_orderkey`)
) engine=innodb default charset=utf8mb4 collate=utf8mb4_bin;

drop table if exists part;
create table `part` (
  `p_partkey` bigint(20) not null,
  `p_name` varchar(55) not null,
  `p_mfgr` char(25) not null,
  `p_brand` char(10) not null,
  `p_type` varchar(25) not null,
  `p_size` bigint(20) not null,
  `p_container` char(10) not null,
  `p_retailprice` decimal(15,2) not null,
  `p_comment` varchar(23) not null,
  primary key (`p_partkey`)
) engine=innodb default charset=utf8mb4 collate=utf8mb4_bin;

drop table if exists partsupp;
create table `partsupp` (
  `ps_partkey` bigint(20) not null,
  `ps_suppkey` bigint(20) not null,
  `ps_availqty` bigint(20) not null,
  `ps_supplycost` decimal(15,2) not null,
  `ps_comment` varchar(199) not null,
  primary key (`ps_partkey`,`ps_suppkey`)
) engine=innodb default charset=utf8mb4 collate=utf8mb4_bin;

drop table if exists region;
create table `region` (
  `r_regionkey` bigint(20) not null,
  `r_name` char(25) not null,
  `r_comment` varchar(152) default null,
  primary key (`r_regionkey`)
) engine=innodb default charset=utf8mb4 collate=utf8mb4_bin;

drop table if exists supplier;
create table `supplier` (
  `s_suppkey` bigint(20) not null,
  `s_name` char(25) not null,
  `s_address` varchar(40) not null,
```

```
  `s_nationkey` bigint(20) not null,
  `s_phone` char(15) not null,
  `s_acctbal` decimal(15,2) not null,
  `s_comment` varchar(101) not null,
  primary key (`s_suppkey`)
) engine=innodb default charset=utf8mb4 collate=utf8mb4_bin;

##  视图
create view revenue0 (supplier_no, total_revenue) as
    select l_suppkey, sum(l_extendedprice * (1 - l_discount))
        from
            lineitem
        where
            l_shipdate >= date '1995-02-01'
            and l_shipdate < date '1995-02-01' + interval '3' month
        group by
           l_suppkey;
```

生成测试数据

您可以通过访问 TPC 官方网站 进行注册并下载 TPC-H 官方测试工具。下载完成后，请将这些测试文件上传至您的服务器，并随后执行数据生成工具的编译过程。

```
unzip TPC-H_Tools_v3.0.0.zip
cd TPC-H_Tools_v3.0.0/dbgen
make
```

> **注意：**
> 如果服务器中未安装 gcc 或 make 等编译工具，请使用服务器的安装镜像进行相关编译组件的安装。必要的编译工具包括：gcc-c++，gcc，make。

编译完成后，即可生成 dbgen 工具，执行如下命令。

```
scale=100
chunk=10
for i in `seq 1 $chunk`
do
./dbgen -s $scale -C $chunk -S $i -f
done
```

> **说明：**
> 上述命令中的 scale 代表生成的数据大小，如填写100，则代表生成100G数据，chunk 指生成的数据文件被拆分为多少份。这两个参数可根据实际情况进行调整。在测试大数据集场景，建议 chunk 设置更大一些，这样子文件会被拆分得更小，导入的时候可以并发执行导入。

导入数据

下面介绍如何将 TPC-H 测试数据分别导入 TDSQL-C MySQL 版中。下表列出了 TPC-H 100G 场景测试数据集中的表数据条数作为参考。其他数据规格下的数据条数与 100G 成比例增加或减少。如 10G 数据则数据条目也是 100G 的 1/10。

表名	数据条数
customer	15000000
lineitem	600037902
nation	25
orders	150000000

版权所有：腾讯云计算（北京）有限责任公司

part	20000000
partsupp	80000000
region	5
supplier	1000000

通过 TPC-H 的 dbgen 工具生成的表文件存放于工具所在的目录，文件名将以 tbl 后缀结尾。若在生成数据时，进行了分割，则会以数字结尾。
如“partsupp.tbl.1”的格式。

```
ls *.tbl
customer.tbl  lineitem.tbl  nation.tbl  orders.tbl  partsupp.tbl  part.tbl  region.tbl  supplier.tbl
```

对于 TDSQL-C MySQL 版集群，我们可以通过 load data 工具将数据导入。首先进入存放生成测试表数据的目录中，然后执行如下命令将数据导入到读写实例中：

> **说明：**
> 下列命令为进入到存放 TPC-H 数据的目录中，将文件逐一导入到数据库中。可根据实际情况进行调整脚本。

```
cd /data/tpchdata
HOST=172.16.0.22
PORT=3306
USER=root
Password=xxxxx
DATABASE=tpch
ls *.tbl* | while read filename
do
    tablename=`echo $filename | awk -F'.tbl' '{print $1}'`
    mysql -u${USER} -h${HOST} -P${PORT} -p${PASSWORD} ${DATABASE} -e "LOAD DATA LOCAL INFILE '${filename}'
INTO TABLE ${tablename} FIELDS TERMINATED BY '|' LINES TERMINATED BY '\n';"
done
```

完成数据导入到读写实例后，可通过SQL 命令查看导入的数据是否准确齐全。可在控制台中，直接设置 tpch 库整库加载到只读分析引擎中。
同时，您也可以通过执行命令行登录到只读分析引擎中执行 SQL 或者通过控制台查询数据加载状态。

```
mysql -uroot -p'' -h10.1.1.3 -P2000 -c
select * from information_schema.libra_table_status;
```

当表的 REPLICATION_STEP 字段状态为 Change Propagation 时，则代表当前数据已经加载完成。
在只读分析引擎中数据完成加载后，即可对表进行统计信息收集。

```
ANALYZE TABLE customer;
ANALYZE TABLE lineitem;
ANALYZE TABLE nation;
ANALYZE TABLE orders;
ANALYZE TABLE part;
ANALYZE TABLE partsupp;
ANALYZE TABLE region;
ANALYZE TABLE supplier;
```

收集完统计信息，则可登录到只读分析引擎中执行 TPC-H 的测试 SQL。TPC-H 的测试 SQL 一共有 22 条。详细的 SQL 文本参考如下。您可以复制 SQL 并访问到只读分析引擎中进行粘贴执行：

> **注意：**

请注意，使用 MySQL 客户端登录到只读分析引擎中执行 SQL 时，需要在登录参数中加入“-c”，这样可以使得在 SQL 中的hint 正常透传到数据库，以获取更好的性能体验。如mysql -uroot -p" -h10.1.1.3 -P2000 -c 。

```
# Q1
select l_returnflag, l_linestatus, sum(l_quantity) as sum_qty, sum(l_extendedprice) as sum_base_price,
sum(l_extendedprice * (1 - l_discount)) as sum_disc_price, sum(l_extendedprice * (1 - l_discount) * (1 +
l_tax)) as sum_charge, avg(l_quantity) as avg_qty, avg(l_extendedprice) as avg_price, avg(l_discount) as
avg_disc, count(*) as count_order from lineitem where l_shipdate <= date_sub('1998-12-01', interval 108
day) group by l_returnflag, l_linestatus order by l_returnflag, l_linestatus;
# Q2
select /*+  SET_VAR(libra_enable_runtime_filter=1) SET_VAR(libra_enable_cost_based_runtime_filter=0)
PX_JOIN_FILTER_ID(0[BLOOM_FILTER],2[BLOOM_FILTER]) */ s_acctbal, s_name, n_name, p_partkey, p_mfgr,
s_address, s_phone, s_comment from part, supplier, partsupp, nation, region where p_partkey = ps_partkey
and s_suppkey = ps_suppkey and p_size = 30 and p_type like '%STEEL'and s_nationkey = n_nationkey and
n_regionkey = r_regionkey and r_name = 'ASIA'and ps_supplycost = (select min(ps_supplycost) from partsupp,
supplier, nation, region where p_partkey = ps_partkey and s_suppkey = ps_suppkey and s_nationkey =
n_nationkey and n_regionkey = r_regionkey and r_name = 'ASIA') order by s_acctbal desc, n_name, s_name,
p_partkey limit 100;
# Q3
select  l_orderkey, sum(l_extendedprice * (1 - l_discount)) as revenue, o_orderdate, o_shippriority from
customer, orders, lineitem where c_mktsegment = 'AUTOMOBILE' and c_custkey = o_custkey and l_orderkey =
o_orderkey and o_orderdate < '1995-03-13' and l_shipdate > '1995-03-13' group by l_orderkey, o_orderdate,
o_shippriority order by revenue desc, o_orderdate limit 10;
# Q4
select o_orderpriority, count(*) as order_count from orders where o_orderdate >= '1995-01-01' and
o_orderdate < date_add('1995-01-01', interval '3' month) and exists ( select * from lineitem where
l_orderkey = o_orderkey and l_commitdate < l_receiptdate ) group by o_orderpriority order by
o_orderpriority;
# Q5
select /*+  SET_VAR(libra_enable_runtime_filter=1) SET_VAR(libra_enable_cost_based_runtime_filter=0)
PX_JOIN_FILTER_ID(3[BLOOM_FILTER],4[BLOOM_FILTER]) */ n_name, sum(l_extendedprice * (1 - l_discount)) as
revenue from customer, orders, lineitem, supplier, nation, region where c_custkey = o_custkey and
l_orderkey = o_orderkey and l_suppkey = s_suppkey and c_nationkey = s_nationkey and s_nationkey =
n_nationkey and n_regionkey = r_regionkey and r_name = 'MIDDLE EAST' and o_orderdate >= '1994-01-01' and
o_orderdate < date_add('1994-01-01', interval '1' year) group by n_name order by revenue desc;
# Q6
select  sum(l_extendedprice * l_discount) as revenue from lineitem where l_shipdate >= '1994-01-01' and
l_shipdate < date_add('1994-01-01', interval '1' year) and l_discount between 0.06 - 0.01 and 0.06 + 0.01
and l_quantity < 24;
# Q7
select /*+   SET_VAR(libra_enable_runtime_filter=1) SET_VAR(libra_enable_cost_based_runtime_filter=0)
PX_JOIN_FILTER_ID(1[BLOOM_FILTER], 3[BLOOM_FILTER]) */  supp_nation, cust_nation, l_year, sum(volume) as
revenue from ( select /*+  leading((n1,supplier),lineitem,((n2,customer),orders)) */ n1.n_name as
supp_nation, n2.n_name as cust_nation, extract(year from l_shipdate) as l_year, l_extendedprice * (1 -
l_discount) as volume from supplier, lineitem, orders, customer, nation n1, nation n2 where s_suppkey =
l_suppkey and o_orderkey = l_orderkey and c_custkey = o_custkey and s_nationkey = n1.n_nationkey and
c_nationkey = n2.n_nationkey and ( (n1.n_name = 'JAPAN' and n2.n_name = 'INDIA') or (n1.n_name = 'INDIA'
and n2.n_name = 'JAPAN') ) and l_shipdate between '1995-01-01' and '1996-12-31' ) as shipping group by
supp_nation, cust_nation, l_year order by supp_nation, cust_nation, l_year;
# Q8
select /*+ SET_VAR(libra_enable_runtime_filter=1) SET_VAR(libra_enable_cost_based_runtime_filter=0)
PX_JOIN_FILTER_ID(1[BLOOM_FILTER], 2[BLOOM_FILTER],5[BLOOM_FILTER]) */ o_year, sum(case when nation =
'INDIA' then volume else 0 end) / sum(volume) as mkt_share from ( select /*+
leading(region,n1,customer,orders,(part,lineitem),(supplier,n2)) */ extract(year from o_orderdate) as
o_year, l_extendedprice * (1 - l_discount) as volume, n2.n_name as nation from part, supplier, lineitem,
orders, customer, nation n1, nation n2, region where p_partkey = l_partkey and s_suppkey = l_suppkey and
l_orderkey = o_orderkey and o_custkey = c_custkey and c_nationkey = n1.n_nationkey and n1.n_regionkey =
r_regionkey and r_name = 'ASIA' and s_nationkey = n2.n_nationkey and o_orderdate between '1995-01-01' and
'1996-12-31' and p_type = 'SMALL PLATED COPPER' ) as all_nations group by o_year order by o_year;
# Q9
```

```
select /*+ SET_VAR(libra_enable_runtime_filter=1) SET_VAR(libra_enable_cost_based_runtime_filter=0)
PX_JOIN_FILTER_ID(4[BLOOM_FILTER]) JOIN_FILTER_NDV_AMP_RATIO(4:0.1)  */  nation, o_year, sum(amount) as
sum_profit from ( select /*+ hash_join_probe(orders) */n_name as nation, extract(year from o_orderdate) as
o_year, l_extendedprice * (1 - l_discount) - ps_supplycost * l_quantity as amount from part, supplier,
lineitem, partsupp, orders, nation where s_suppkey = l_suppkey and ps_suppkey = l_suppkey and ps_partkey =
l_partkey and p_partkey = l_partkey and o_orderkey = l_orderkey and s_nationkey = n_nationkey and p_name
like '%dim%' ) as profit group by nation, o_year order by nation, o_year desc;
# Q10
select /*+  HASH_JOIN_PROBE((nation,customer)), leading(nation, customer,(orders, lineitem))  */
c_custkey, c_name, sum(l_extendedprice * (1 - l_discount)) as revenue, c_acctbal, n_name, c_address,
c_phone, c_comment from customer, orders, lineitem, nation where c_custkey = o_custkey and l_orderkey =
o_orderkey and o_orderdate >= '1993-08-01' and o_orderdate < date_add('1993-08-01', interval '3' month)
and l_returnflag = 'R' and c_nationkey = n_nationkey group by c_custkey, c_name, c_acctbal, c_phone,
n_name, c_address, c_comment order by revenue desc limit 20;
# Q11
select ps_partkey, sum(ps_supplycost * ps_availqty) as value from partsupp, supplier, nation where
ps_suppkey = s_suppkey and s_nationkey = n_nationkey and n_name = 'MOZAMBIQUE' group by ps_partkey having
sum(ps_supplycost * ps_availqty) > ( select sum(ps_supplycost * ps_availqty) * 0.0001000000 from partsupp,
supplier, nation where ps_suppkey = s_suppkey and s_nationkey = n_nationkey and n_name = 'MOZAMBIQUE' )
order by value desc;
# Q12
select /*+  SET_VAR(libra_enable_runtime_filter=1) SET_VAR(libra_enable_cost_based_runtime_filter=0)
PX_JOIN_FILTER_ID(0[BLOOM_FILTER]) */ l_shipmode, sum(case when o_orderpriority = '1-URGENT' or
o_orderpriority = '2-HIGH' then 1 else 0 end) as high_line_count, sum(case when o_orderpriority <> '1-
URGENT' and o_orderpriority <> '2-HIGH' then 1 else 0 end) as low_line_count from orders, lineitem where
o_orderkey = l_orderkey and l_shipmode in ('RAIL', 'FOB') and l_commitdate < l_receiptdate and l_shipdate
< l_commitdate and l_receiptdate >= '1997-01-01' and l_receiptdate < date_add('1997-01-01', interval '1'
year) group by l_shipmode order by l_shipmode;
# Q13
select  c_count, count(*) as custdist from (select c_custkey, count(o_orderkey) as c_count from customer
left outer join orders on c_custkey = o_custkey and o_comment not like '%pending%deposits%'group by
c_custkey ) c_orders group by c_count order by custdist desc, c_count desc;
# Q14
select  100.00 * sum(case when p_type like 'PROMO%'then l_extendedprice * (1 - l_discount) else 0 end) /
sum(l_extendedprice * (1 - l_discount)) as promo_revenue from lineitem, part where l_partkey = p_partkey
and l_shipdate >= '1996-12-01'and l_shipdate < date_add('1996-12-01', interval '1' month);
# Q15
select  s_suppkey, s_name, s_address, s_phone, total_revenue from supplier, revenue0 where s_suppkey =
supplier_no and total_revenue = (select max(total_revenue) from revenue0 ) order by s_suppkey;
# Q16
select  p_brand, p_type, p_size, count(distinct ps_suppkey) as supplier_cnt from partsupp, part where
p_partkey = ps_partkey and p_brand <> 'Brand#34'and p_type not like 'LARGE BRUSHED%'and p_size in (48, 19,
12, 4, 41, 7, 21, 39) and ps_suppkey not in (select s_suppkey from supplier where s_comment like
'%Customer%Complaints%') group by p_brand, p_type, p_size order by supplier_cnt desc, p_brand, p_type,
p_size;
# Q17
select /*+ SET_VAR(libra_enable_runtime_filter=1) SET_VAR(libra_enable_cost_based_runtime_filter=0)
PX_JOIN_FILTER_ID(0[BLOOM_FILTER],1[BLOOM_FILTER]) */ sum(l_extendedprice) / 7.0 as avg_yearly from
lineitem, part where p_partkey = l_partkey and p_brand = 'Brand#44'and p_container = 'WRAP PKG'and
l_quantity < (select 0.2 * avg(l_quantity) from lineitem where l_partkey = p_partkey );
# Q18
select /*+  PX_JOIN_FILTER_ID(0,1,2)  leading(customer, (orders, lineitem@sel_2))
hash_join_probe(customer) JOIN_FILTER_NDV_AMP_RATIO(0:0.00001)  JOIN_FILTER_NDV_AMP_RATIO(2:0.00001)
JOIN_FILTER_NDV_AMP_RATIO(1:0.0001) */ c_name, c_custkey, o_orderkey, o_orderdate, o_totalprice,
sum(l_quantity) from customer, orders, lineitem where o_orderkey in (select l_orderkey from lineitem group
by l_orderkey having sum(l_quantity) > 314 ) and c_custkey = o_custkey and o_orderkey = l_orderkey group
by c_name, c_custkey, o_orderkey, o_orderdate, o_totalprice order by o_totalprice desc, o_orderdate limit
100;
# Q19
select /*+  SET_VAR(libra_enable_runtime_filter=1) SET_VAR(libra_enable_cost_based_runtime_filter=0)
PX_JOIN_FILTER_ID(0[BLOOM_FILTER]) */ sum(l_extendedprice* (1 - l_discount)) as revenue from lineitem,
```

```
part where (p_partkey = l_partkey and p_brand = 'Brand#52'and p_container in ('SM CASE', 'SM BOX', 'SM
PACK', 'SM PKG') and l_quantity >= 4 and l_quantity <= 4 + 10 and p_size between 1 and 5 and l_shipmode in
('AIR', 'AIR REG') and l_shipinstruct = 'DELIVER IN PERSON') or (p_partkey = l_partkey and p_brand =
'Brand#11'and p_container in ('MED BAG', 'MED BOX', 'MED PKG', 'MED PACK') and l_quantity >= 18 and
l_quantity <= 18 + 10 and p_size between 1 and 10 and l_shipmode in ('AIR', 'AIR REG') and l_shipinstruct
= 'DELIVER IN PERSON') or (p_partkey = l_partkey and p_brand = 'Brand#51'and p_container in ('LG CASE',
'LG BOX', 'LG PACK', 'LG PKG') and l_quantity >= 29 and l_quantity <= 29 + 10 and p_size between 1 and 15
and l_shipmode in ('AIR', 'AIR REG') and l_shipinstruct = 'DELIVER IN PERSON');
# Q20
select  s_name, s_address from supplier, nation where s_suppkey in (select ps_suppkey from partsupp where
ps_partkey in (select p_partkey from part where p_name like 'green%') and ps_availqty > (select 0.5 *
sum(l_quantity) from lineitem where l_partkey = ps_partkey and l_suppkey = ps_suppkey and l_shipdate >=
'1993-01-01'and l_shipdate < date_add('1993-01-01', interval '1' year) ) ) and s_nationkey = n_nationkey
and n_name = 'ALGERIA'order by s_name;
# Q21
select /*+  SET_VAR(libra_enable_runtime_filter=1) SET_VAR(libra_enable_cost_based_runtime_filter=0)
JOIN_FILTER_NDV_AMP_RATIO(0:0.5) JOIN_FILTER_NDV_AMP_RATIO(1:0.5) PX_JOIN_FILTER_ID(3[BLOOM_FILTER],1,0)
*/ s_name, count(*) as numwait from supplier, lineitem l1, orders, nation where s_suppkey = l1.l_suppkey
and o_orderkey = l1.l_orderkey and o_orderstatus = 'F' and l1.l_receiptdate > l1.l_commitdate and exists (
select * from lineitem l2 where l2.l_orderkey = l1.l_orderkey and l2.l_suppkey <> l1.l_suppkey ) and not
exists ( select * from lineitem l3 where l3.l_orderkey = l1.l_orderkey and l3.l_suppkey <> l1.l_suppkey
and l3.l_receiptdate > l3.l_commitdate ) and s_nationkey = n_nationkey and n_name = 'EGYPT' group by
s_name order by numwait desc, s_name limit 100;
# Q22
select  cntrycode, count(*) as numcust, sum(c_acctbal) as totacctbal from (select substring(c_phone from 1
for 2) as cntrycode, c_acctbal from customer where substring(c_phone from 1 for 2) in ('20', '40', '22',
'30', '39', '42', '21') and c_acctbal > (select avg(c_acctbal) from customer where c_acctbal > 0.00 and
substring(c_phone from 1 for 2) in ('20', '40', '22', '30', '39', '42', '21') ) and not exists (select *
from orders where o_custkey = c_custkey ) ) as custsale group by cntrycode order by cntrycode;
```

版权所有：腾讯云计算（北京）有限责任公司

TPC-H 性能结果

最近更新时间：2024-08-22 15:30:51

本文档介绍 TDSQL-C MySQL 版的“只读分析引擎”测试 TPC-H 的性能情况。

> **注意：**
> 只读分析引擎实例的性能情况根据不同的实例规格存在差异。本文档的测试结果仅做参考。测试方法可参考“TPC-H 测试方法”

TPC-H

实例规格	磁盘大小	测试数据集
通用型 32C 128GB	200GB	100GB

性能结果：

查询 SQL	执行时间(s)
Q1	3.66
Q2	0.39
Q3	2.37
Q4	1.53
Q5	2.91
Q6	0.74
Q7	1.99
Q8	1.6
Q9	7.8
Q10	2.81
Q11	0.42
Q12	1.79
Q13	3.32
Q14	1.12
Q15	1.2
Q16	1.18
Q17	1.42
Q18	3.62
Q19	1.74
Q20	1.64
Q21	5.04
Q22	0.51
总计	48.8

版权所有：腾讯云计算（北京）有限责任公司

数据库安全和加密
安全组
简介

最近更新时间：2024-10-12 17:49:11

本文为您介绍安全组相关概念和说明。

什么是安全组

安全组 是一种虚拟防火墙，具备有状态的数据包过滤功能，用于设置云服务器、负载均衡、云数据库等实例的网络访问控制，控制实例级别的出入流量，是重要的网络安全隔离手段。
您可以通过配置安全组规则，允许或禁止安全组内的实例的出流量和入流量。

安全组规则与限制

- 详细安全组规则请参见 安全组概述 。
- 安全组的使用限制及配额，请参见 使用限制总览 中的安全组相关限制章节。

安全组说明

- TDSQL-C MySQL 版安全组目前仅支持私有网络 VPC 内网访问和外网访问的网络控制，暂不支持对基础网络的网络控制。
- 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。
- TDSQL-C MySQL 版安全组支持读写实例、只读实例、数据库代理。

TDSQL-C MySQL 版安全组相关操作

- 创建和管理云数据库安全组 。
- 添加或修改安全组规则 。
- 为云数据库配置安全组 。

版权所有：腾讯云计算（北京）有限责任公司

管理云数据库安全组
创建和管理云数据库安全组

最近更新时间：2024-10-12 18:28:21

本文介绍为 TDSQL-C MySQL 版集群创建安全组及创建后对安全组的管理。

创建安全组

1. 登录 私有网络控制台 。
2. 在左侧导航选择**安全组**，在上方选择**地域**，单击**新建**。



3. 在弹出的对话框，完成如下配置，确认后单击**确定**。



表1:

参数	说明
模板	根据安全组中的数据库实例需要部署的服务，选择合适的模板，简化安全组规则配置。如下表2所示。
名称	自定义设置安全组名称。
所属项目	默认选择默认项目，可指定为其他项目，便于后期查询和管理。
备注	自定义，简短地描述安全组，便于后期管理。

高级选项	可在高级选项中为安全组配置标签，默认无标签。可按需进行添加，标签详情请参见 标签产品文档 。
显示模板规则	若选择了模板，可单击显示模板规则展示规则详情。

表2：

模板	说明	场景
放通全部端口	默认放通全部端口到公网和内网，具有一定安全风险。	-
放通22，80，443，3389端口和ICMP协议	默认放通（22，80，443，3389）TCP 协议端口和 ICMP 协议，内网全放通。	安全组中的实例需要部署 Web 服务。
自定义	安全组创建成功后，按需自行添加安全组规则。具体操作请参见下文 添加安全组规则 。	-

管理安全组

创建安全组后，您可对安全组进行管理操作，详细操作请参见以下相关步骤。

- 查看安全组
- 移出安全组
- 克隆安全组
- 删除安全组

修改或添加安全组规则

最近更新时间：2024-10-22 17:33:41

安全组用于管理是否放行来自公网或者内网的访问请求。为安全起见，安全组入站方向大多采取拒绝访问策略，您可以根据需要修改或添加安全组规则。本文为您介绍通过控制台修改或添加安全组规则。

前提条件

- 已创建一个安全组。具体操作请参见 创建云数据库安全组 。
- 您已经知道云服务器实例需要允许或禁止哪些公网或内网的访问。更多安全组规则设置的相关应用案例，请参见 安全组应用案例 。

修改安全组规则

1. 登录 安全组页 ，找到创建的目标安全组，单击其**操作**列的**修改规则**。
2. 在安全组规则页面，根据需要修改安全组规则所属的方向（入站/出站），单击入站/出站规则页签。
3. 找到需要修改的安全组规则，单击操作列的编辑，即可对已有规则进行修改。

添加安全组规则

1. 登录 安全组页 ，找到创建的目标安全组，单击其**操作**列的**修改规则**。
2. 在安全组规则页面，选择**入站规则** > **添加规则**。
3. 在弹出的对话框中，设置规则。



- **类型**：默认选择**自定义**，您也可以选择其他系统规则模板，例如 **Windows 登录**模板、**Linux 登录**模板、**Ping 模板**、**HTTP(80) 模板**、**HTTPS(443)**模板、**MySQL(3306)**模板和 **SQLServer(1433)**模板。
- **来源**：流量的源（入站规则）或目标（出站规则），请指定以下选项之一：

来源	说明
IP 地址或 CIDR 段	用 CIDR 表示法（IPV4；如203.0.113.0、203.0.113.0/24或者0.0.0.0/0，其中0.0.0.0/0代表匹配所有 IPv4 地址。IPV6：如FF05::B5、FF05:B5::/60、::/0或者0::0/0，其中::/0或者0::0/0代表匹配所有 IPv6 地址）。
参数模板-IP 地址	引用 参数模板 中的 IP 地址对象。
参数模板-IP 地址组	引用 参数模板 中的 IP 地址组对象。
安全组	引用安全组 ID，您可以引用以下安全组的 ID： • 当前安全组：当前安全组表示与云数据库关联的安全组 ID。 • 其他安全组：其他安全组表示同一区域中同一项目下的另一个安全组 ID。 **说明：** • 引用安全组 ID 法作为高阶功能，您可选择使用。所引用安全组的规则不会被添加到当前安全组。 • 在配置安全组规则时，如果在来源/目标中输入安全组 ID ，表示仅将此安全组 ID 所绑定的云数据库实例、弹性网卡的内网 IP 地址作为来源/目标，不包括外网 IP 地址。

说明：

> IP 地址后面的/**数字**为子网掩码，其中**数字**表示子网掩码中网络位的长度。例如，192.168.0.0/24表示一个网段，子网掩码 **/24** 表示192.168.0.0的前24位为网络位，后8位为主机位，即在192.168.0.0/24子网内，可分配的主机 IP 范围为：192.168.0.0 - 192.168.0.255。

- **协议端口**：填写协议类型和端口范围，协议类型支持 TCP、UDP、ICMP、ICMPv6 及 GRE。您也可以引用 参数模板 中的协议端口或协议端口组。

> **注意：**
> - 连接 TDSQL-C MySQL 版，须放通 TDSQL-C MySQL 版实例端口。您可登录 TDSQL-C MySQL 版控制台 单击**集群 ID** 进入详情页，在连接信息处查看端口。
> - TDSQL-C MySQL 版控制台安全组页面设置的安全组规则，对内网地址和外网地址（若开启后），均统一生效。

协议端口支持格式如下：
  - 单个端口，如 `TCP:80` 。
  - 多个离散端口，如 `TCP:80,443` 。
  - 连续端口，如 `TCP:3306-20000` 。
  - 所有端口，如 `TCP:ALL` 。
- **策略**：默认选择**允许**。
  - 允许：放行该端口相应的访问请求。
  - 拒绝：直接丢弃数据包，不返回任何回应信息。
- **备注**：自定义，简短地描述规则，便于后期管理。

4. 单击**完成**。
5. 在安全组规则页面，单击**出站规则**，并参考 步骤2 - 步骤4，完成安全组出站规则的添加。

为云数据库匹配安全组

最近更新时间：2025-01-15 10:06:43

安全组是腾讯云提供的防火墙，可以对云数据库进行入流量控制。您可以在购买集群时绑定安全组，购买集群后也可以在控制台绑定安全组。
本文介绍通过控制台为 TDSQL-C MySQL 版配置安全组。

注意：

- 目前 TDSQL-C MySQL 版安全组仅支持私有网络云数据库配置。
- TDSQL-C MySQL 版支持为**读写实例、只读实例、只读分析引擎、数据库代理**配置不同安全组，互相不影响。

配置安全组

购买集群时配置安全组

1. 登录 购买页 。
2. 完成数据库配置和基础信息配置，在高级配置下配置安全组，完成余下购买设置，单击**立即购买**即可 。



购买集群后在控制台配置安全组

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在上方选择地域，根据实际控制台视图模式，对应操作。

页签视图

在左侧集群列表，单击**目标集群**，进入集群管理页。

列表视图

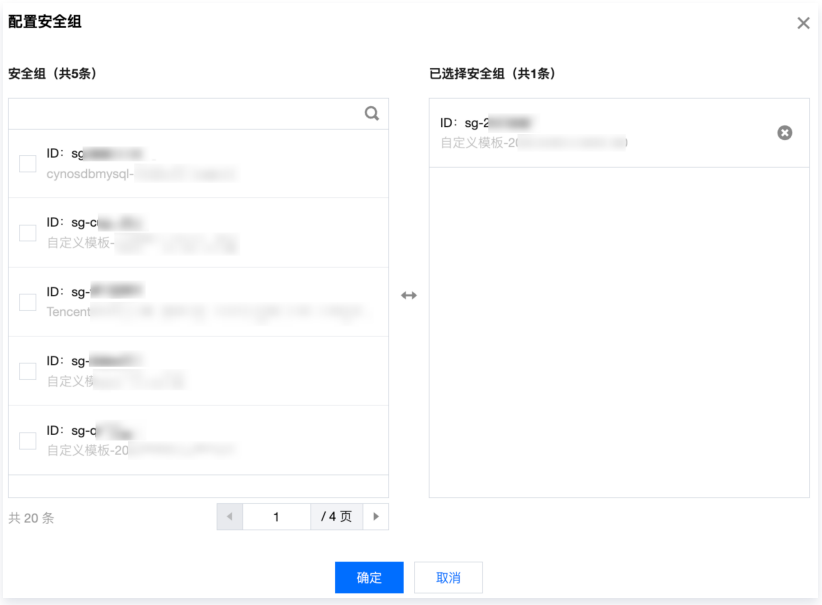
在左侧集群列表找到目标集群，单击**集群 ID** 或**操作**列的**管理**，进入集群管理页。



3. 在集群管理页面，选择**安全组**页，选择需要配置安全组的选项（读写实例、只读实例、只读分析引擎、数据库代理），单击**配置安全组**。



4. 在配置安全组窗口下，选择安全组规则（也可以通过检索安全组 ID 找到您需要的安全组），单击**确定**。



> **说明：**
> - 您可配置多个安全组规则，最多不超过5个。
> - 不同的访问地址可配置不同的安全组，安全组拦截仅会对通过当前地址访问的源进行控制。

调整安全组优先级

当 TDSQL-C MySQL 版实例绑定多个安全组时，多个安全组将按照优先级顺序（如1、2）依次匹配执行，您可以调整安全组的优先级。

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在上方选择地域，在集群列表找到需要的集群，单击**集群 ID** 或**操作**列的**管理**，进入集群管理页面。
3. 在集群管理页面，选择**安全组**页，选择需要配置安全组的读写实例、数据库代理或者只读实例，单击**已加入安全组**下面的**编辑**。
4. 在安全组操作列，通过上调/下调/删除图标调整安全组的优先级，调整后单击**保存**。



访问与授权
访问管理概述

最近更新时间：2024-10-12 11:28:32

存在问题

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

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

解决方案

您可以通过子账号实现不同的人管理不同的服务来规避以上的问题。默认情况下，子账号没有使用云服务的权利或者相关资源的权限。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权限。

访问管理（Cloud Access Management，CAM）可以帮助您安全、便捷地管理对腾讯云服务和资源的访问。您可以使用 CAM 创建子用户、用户组和角色，并通过策略控制其访问范围。CAM 支持用户和角色 SSO 能力，您可以根据具体管理场景针对性设置企业内用户和腾讯云的互通能力。您最初创建的腾讯云主账号，拥有整个账号全部腾讯云服务和资源的完全访问权限，建议您保护好主账号的凭证信息，日常使用子用户或角色进行访问，并开启多因素校验和定时轮换密钥。

当您使用 CAM 的时候，可以将策略与一个用户或一组用户关联起来，策略能够授权或者拒绝用户使用指定资源完成指定任务。有关 CAM 策略的更多相关基本信息，请参见 策略语法，更多使用信息，请参见 策略。

若您不需要对子账户进行云数据库相关资源的访问管理，您可以跳过此章节。跳过这些部分不会影响您对文档中其余部分的理解和使用。

快速入门

CAM 策略必须授权使用一个或多个 TDSQL-C MySQL 版操作，或者必须拒绝使用一个或多个 TDSQL-C MySQL 版操作，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。

> **说明**
>
> - 建议用户使用 CAM 策略来管理 TDSQL-C MySQL 版资源和授权 TDSQL-C MySQL 版操作，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。
> - TDSQL-C MySQL 版暂时不支持相关生效条件设置。

相关内容	链接
了解策略基本结构	策略语法
在策略中定义操作	TDSQL-C MySQL 版的操作
在策略中定义资源	TDSQL-C MySQL 版的资源路径
资源级权限	TDSQL-C MySQL 版支持的资源级权限

授权策略语法

最近更新时间：2024-05-13 15:21:51

CAM 策略语法

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

- **版本 version**：必填项，目前仅允许值为"2.0"。
- **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。
  - **影响 effect**：必填项，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。
  - **操作 action**：必填项，用来描述允许或拒绝的操作。操作可以是 API 或者功能集（一组特定的 API，以 permid 前缀描述）。
  - **资源 resource**：必填项，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。
  - **生效条件 condition**：必填项，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。

操作

在 CAM 策略语句中，您可以从支持 CAM 的任何服务中指定任意的 API 操作。对于 TDSQL-C MySQL 版，请使用以 cynosdb: 为前缀的 API。例如 cynosdb:DescribeClusters 或 cynosdb:ResetAccountPassword。
如果您要在单个语句中指定多个操作的时候，请使用逗号将它们隔开，如下所示：

```
"action":["cynosdb:action1","cynosdb: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":["cynosdb:Describe*"]
```

如果您要指定 TDSQL-C MySQL 版中所有操作，请使用 * 通配符，如下所示：

```
"action": ["cynosdb:*"]
```

资源路径

每个 CAM 策略语句都有适用于自己的资源。
资源路径的一般形式如下：

```
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 cynosdb。
- **region**：地域信息，如 bj。
- **account**：资源拥有者的主账号信息，如 uin/12xxx8。

版权所有：腾讯云计算（北京）有限责任公司

- **resource**：各产品的具体资源详情，如 instance/clusterId 或者 instance/*。

例如，您可以使用特定集群（cynosdbmysql-123abc）在语句中指定它，如下所示：

```
"resource":[ "qcs::cynosdb:bj:uin/12xxx8:instance/cynosdbmysql-123abc"]
```

您还可以使用 * 通配符指定属于特定账户的所有集群，如下所示：

```
"resource":[ "qcs::cynosdb:bj:uin/12xxx8: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 TDSQL-C MySQL 版能够使用的资源和对应的资源描述方法。其中，$为前缀的单词均为代称，region 指地域，account 指账户 ID。

资源	授权策略中的资源描述方法
集群	`qcs::cynosdb:$region:$account:instance/$clusterId`
VPC	`qcs::vpc:$region:$account:vpc/$vpcId`
安全组	`qcs::cvm:$region:$account:sg/$sgId`

可授权的资源类型

最近更新时间：2024-10-12 16:58:45

资源级权限指的是能够指定用户对哪些资源具有执行操作的能力。TDSQL-C MySQL 版部分支持资源级权限，即表示针对支持资源级权限的 TDSQL-C MySQL 版操作，您可以控制何时允许用户执行操作或是允许用户使用特定资源。访问管理（Cloud Access Management，CAM）中可授权的资源类型如下：

资源类型	授权策略中的资源描述方法
集群相关	qcs::cynosdb:$region::instance/* qcs::cynosdb:$region:$account:instanceId/$clusterId

下表将介绍当前支持资源级权限的 TDSQL-C MySQL 版 API 操作，以及每个操作支持的资源。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 *。

集群相关

API 操作	资源路径
DescribeBackupConfig	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
DescribeBackupList	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
DescribeRollbackTimeRange	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
DescribeRollbackTimeValidity	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
ModifyBackupConfig	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
ActivateCluster	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
DescribeClusterDetail	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
IsolateCluster	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
ModifyClusterName	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
ModifyClusterProject	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
OfflineCluster	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
DeleteAccounts	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId

DescribeAccounts	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
ModifyAccountDescription	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
ResetAccountPassword	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
DescribeClusterInstanceGrps	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
ActivateInstance	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
DescribeInstanceDetail	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
IsolateInstance	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
UpgradeInstance	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
ModifyInstanceName	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
OfflineInstance	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
DescribeClusterAddr	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
DescribeClusterNetService	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
DescribeClusterParams	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
DescribeClusterServerInfo	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
DescribeErrorLogs	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
DescribeMaintainPeriod	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
DescribeSlowLogs	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
ModifyClusterParam	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
ModifyMaintainPeriodConfig	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
DescribeDBSecurityGroups	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId

版权所有：腾讯云计算（北京）有限责任公司

ModifyDBInstanceSecurityGroups	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
CloseWan	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
OpenWan	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
DescribeClusters	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
DescribeInstances	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId
DescribeIsolatedInstances	qcs::cynosdb:$region:$account:instanceId/* qcs::cynosdb:$region:$account:instanceId/$clusterId

版权所有：腾讯云计算（北京）有限责任公司

加密
设置 SSL 加密

最近更新时间：2025-05-06 17:23:12

SSL 加密概述

SSL（Secure Sockets Layer）认证是客户端到云数据库服务器端的认证，对用户和服务器进行认证。开通 SSL 加密，可获取 CA 证书，将 CA 证书上传在服务端。在客户端访问数据库时，将激活 SSL 协议，在客户端和数据库服务端之间建立一条 SSL 安全通道，实现数据信息加密传输，防止数据在传输过程中被截取、篡改、窃听，保证双方传递信息的安全性。
SSL 协议要求建立在可靠的传输层协议（TCP）之上，其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。

> **说明：**
> 开启 SSL 加密后，通过 SSL 首次连接已暂停的 Serverless 实例时会出现报错，如遇报错，请稍后重试即可。

背景

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 协议是为解决这三大风险而设计的，理论上可达到：

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

TDSQL-C MySQL 版支持通过开启 SSL 加密来增强链路安全性，并支持下载和安装 SSL CA 证书到需要的应用服务。

> **注意：**
> SSL 加密不保护数据本身，是确保来往于数据库和服务器之间的流量安全，在传输层对网络连接进行加密，能够提升通信数据的安全性和完整性，但会同时增加网络连接响应时间。

前提条件

- 实例版本为 MySQL5.7/8.0。
- 实例形态为预置资源或 Serverless。

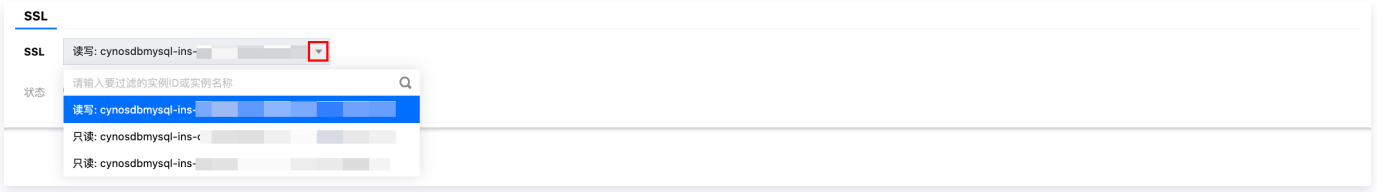
支持版本

TDSQL-C MySQL 版使用 OpenSSL 来实现安全连接。TDSQL-C MySQL 版支持传输层安全性协议（TLS）版本1.0、1.1、1.2和1.3。TLS 支持取决于 MySQL 版本。下表显示了支持 TLS 的 MySQL 版本。

MySQL 版本	TLS1.0	TLS1.1	TLS1.2	TLS1.3
MySQL 5.7	不支持	不支持	支持	支持
MySQL 8.0	支持	支持	支持	支持

开启 SSL 加密

1. 登录 TDSQL-C MySQL 版控制台，在集群列表，单击**集群 ID**，进入集群管理页面。
2. 在集群管理页面选择**数据安全**页，在 **SSL** 下选择需要开启的读写实例或只读实例。



3. 此功能状态默认为未打开，将开关调为开启，然后单击**确定**，开启 SSL 加密。



> **注意：**
> 开启 SSL 过程中，会重启您的数据库实例以加载 SSL 证书，请确保业务具备重连机制。

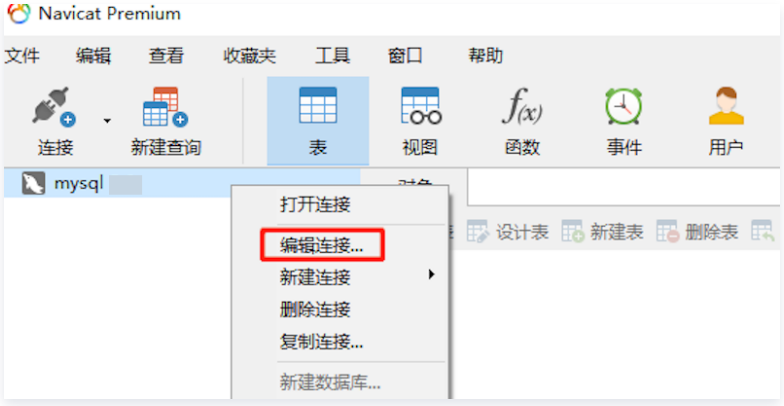
4. 单击**下载**，下载 SSL CA 证书，证书有效期为20年。
   下载的文件为压缩包（TencentDB-SSL-CA.zip），包含如下三个文件：
   - p7b 文件：用于 Windows 系统中导入 CA 证书。
   - jks 文件：Java 中的 truststore 和 keystore 存储文件，密码统一为 tencentdb，用于 Java 程序中导入 CA 证书链。
   - pem 文件：用于其他系统或应用中导入 CA 证书。

配置 SSL CA 证书

开启 SSL 加密后，使用客户端连接云数据库时需要配置 SSL CA 证书。以下以 Navicat 为例，为您介绍 SSL CA 证书安装方法。其它应用或者客户端请参见对应产品的使用说明。

> **说明：**
> 每开启或关闭一次 SSL 加密，其证书就会新生成。

1. 打开 Navicat。
2. 在对应数据库上单击鼠标右键，选择**编辑连接**。

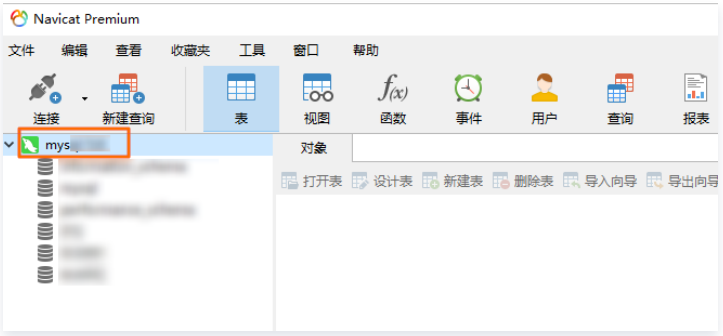


3. 选择 SSL 页签，选择.pem 格式 CA 证书的路径。完成下图设置后单击**确定**。



> **说明：**
> 如果出现 connection is being used 报错，可能由于之前的会话未断开，请关闭 Navicat 后重试。

4. 双击对应数据库，测试能否正常连接。



关闭 SSL 加密

1. 登录 TDSQL-C MySQL 版控制台，在集群列表，单击**集群 ID**，进入集群管理页面。
2. 在集群管理页面选择**数据安全**页，在 **SSL** 下选择需要关闭的读写实例或只读实例。
3. 将**状态**后的开关调为关闭，在弹出的提示框中单击**确定**。



> **说明：**
> 关闭 SSL 过程中，会重启您的数据库实例以卸载 SSL 证书，请确保业务具备重连机制。

使用 MySQL 命令行客户端连接开启 SSL 加密的实例

如果您使用的数据库版本不同，则 MySQL 客户端的连接命令参数有所不同，您可通过如下命令，先查询所使用的数据库版本，再参见后续步骤连接实例。

```
SELECT VERSION();

查询结果示例：
+--------------+
| VERSION()    |
+--------------+
| 8.0.30-txsql |
+--------------+
1 row in set (0.00 sec)
```

1. 通过 TDSQL-C MySQL 版控制台下载 SSL CA 证书，操作请参见 开启 SSL 加密 。
2. 使用 MySQL 命令行客户端，通过命令连接开启 SSL 加密的实例。
   - 客户端数据库版本为 MySQL 5.7/8.0时的命令，通过如下命令连接实例。

     ```
     mysql -h <IP 地址> --ssl-ca=<ca证书路径> --ssl-mode=REQUIRED -P <端口号> -u <用户名> -p
     ```

   - 如果要使用其他的 SSL 模式，例如 VERIFY_CA 或 VERIFY_IDENTITY，则需要通过如下命令连接实例。

     ```
     mysql -h <IP 地址> --ssl-ca=<ca证书路径> --ssl-mode=VERIFY_CA -P <端口号> -u <用户名> -p
     ```

     > **说明：**
     > --ssl-mode 参数表示 SSL 模式，通常情况下，推荐使用 REQUIRED 和 VERIFY_CA 模式，表示要求 MySQL 客户端使用 SSL/TLS 协议连接 MySQL 服务器，并要求验证 MySQL 服务器的 SSL/TLS 证书；而 VERIFY_IDENTITY 模式除了要求验证 MySQL 服务器的 SSL/TLS 证书，还要求客户端使用的主机名与服务器证书中的标识相匹配，否则 MySQL 客户端会拒绝连接 MySQL 服务器。

3. 根据系统提示输入对应用户名的密码。

```
[root@VM-32-12-tencentos ~]# mysql -h [redacted]9 --ssl-ca=ca.pem --ssl-mode=VERIFY_CA -P 3306 -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is [redacted]33
Server version: 8.0.30-txsql 20221220

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
```

常用程序连接开启 SSL 的实例的代码示例

- PHP

```
$conn = mysqli_init();
mysqli_ssl_set($conn,NULL,NULL, "<下载的证书路径>", NULL, NULL);
mysqli_real_connect($conn, '<数据库访问地址>', '<数据库访问用户名>', '<数据库访问密码>', '<指定访问数据库>', <访问端口>, MYSQLI_CLIENT_SSL);
if (mysqli_connect_errno($conn)) {
die('Failed to connect to MySQL: '.mysqli_connect_error());
}
```

- PHP (Using PDO)

```
$options = array(
  PDO::MYSQL_ATTR_SSL_CA => '<下载的证书路径>'
);
```

```
$db = new PDO('mysql:host=<数据库访问地址>;port=<访问端口>;dbname='<指定访问数据库>', '<数据库访问用户名>', '<数据库访问密码>', $options);
```

- Java (MySQL Connector for Java)

```
# generate truststore and keystore in code

String importCert = " -import "+
  " -alias mysqlServerCACert "+
  " -file " + ssl_ca +
  " -keystore truststore "+
  " -trustcacerts " +
  " -storepass password -noprompt ";
String genKey = " -genkey -keyalg rsa " +
  " -alias mysqlClientCertificate -keystore keystore " +
  " -storepass password123 -keypass password " +
  " -dname CN=MS ";
sun.security.tools.keytool.Main.main(importCert.trim().split("\\s+"));
sun.security.tools.keytool.Main.main(genKey.trim().split("\\s+"));

# use the generated keystore and truststore

System.setProperty("javax.net.ssl.keyStore","<下载的证书路径>");
System.setProperty("javax.net.ssl.keyStorePassword","tencentdb");
System.setProperty("javax.net.ssl.trustStore","<下载的证书路径>");
System.setProperty("javax.net.ssl.trustStorePassword","tencentdb");

url = String.format("jdbc:mysql://%s/%s?serverTimezone=UTC&useSSL=true", '<数据库访问地址>', '<指定访问数据库>');
properties.setProperty("user", '<数据库访问用户名>');
properties.setProperty("password", '<数据库访问密码>');
conn = DriverManager.getConnection(url, properties);
```

- .NET (MySqlConnector)

```
var builder = new MySqlConnectionStringBuilder
{
  Server = "<数据库访问地址>",
  UserID = "<数据库访问用户名>",
  Password = "<数据库访问密码>",
  Database = "<指定访问数据库>",
  SslMode = MySqlSslMode.VerifyCA,
  SslCa = "<下载的证书>",
};
using (var connection = new MySqlConnection(builder.ConnectionString))
{
  connection.Open();
}
```

- Python (MySQLConnector Python)

```
try:
  conn = mysql.connector.connect(user='<数据库访问用户名>',
                                 password='<数据库访问密码>',
                                 database='<指定访问数据库>',
                                 host='<数据库访问地址>',
                                 ssl_ca='<下载的证书路径>')
except mysql.connector.Error as err:
```

```
  print(err)
```

- Python (PyMySQL)

```
conn = pymysql.connect(user='<数据库访问用户名>',
                       password='<数据库访问密码>',
                       database='<指定访问数据库>',
                       host='<数据库访问地址>',
                       ssl={'ca': '<下载的证书路径>'})
```

- Django (PyMySQL)

```
DATABASES = {
  'default': {
      'ENGINE': 'django.db.backends.mysql',
      'NAME': '<指定访问数据库>',
      'USER': '<数据库访问用户名>',
      'PASSWORD': '<数据库访问密码>',
      'HOST': '<数据库访问地址>',
      'PORT': '<访问端口>',
      'OPTIONS': {
          'ssl': {'ca': '<下载的证书路径>'}
      }
  }
}
```

- Node.js

```
var fs = require('fs');
var mysql = require('mysql');
const serverCa = [fs.readFileSync("<下载的证书路径>", "utf8")];
var conn=mysql.createConnection({
  host:"<数据库访问地址>",
  user:"<数据库访问用户名>",
  password:"<数据库访问密码>",
  database:"<指定访问数据库>",
  port:<访问端口>,
  ssl: {
      rejectUnauthorized: true,
      ca: serverCa
  }
});
conn.connect(function(err) {
if (err) throw err;
});
```

- Golang

```
rootCertPool := x509.NewCertPool()
pem, _ := ioutil.ReadFile("<下载的证书路径>")
if ok := rootCertPool.AppendCertsFromPEM(pem); !ok {
  log.Fatal("Failed to append PEM.")
}
mysql.RegisterTLSConfig("custom", &tls.Config{RootCAs: rootCertPool})
var connectionString string
connectionString = fmt.Sprintf("%s:%s@tcp(%s:<访问端口>)/%s?allowNativePasswords=true&tls=custom","<数据库访问用户名>" , "<数据库访问密码>", "<数据库访问地址>", '<指定访问数据库>')
```

```
db, _ := sql.Open("mysql", connectionString)
```

- Ruby

```
client = Mysql2::Client.new(
      :host     => '<数据库访问地址>',
      :username => '<数据库访问用户名>',
      :password => '<数据库访问密码>',
      :database => '<指定访问数据库>',
      :sslca => '<下载的证书路径>'
  )
```

透明数据加密

最近更新时间：2024-12-18 14:49:12

本文为您介绍 TDSQL-C MySQL 版的透明数据加密功能。

功能概述

透明数据加密（Transparent Data Encryption，以下简称 TDE）功能，指数据的加解密操作对用户透明。

使用场景

使用 TDE 功能，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，从磁盘读取内存时进行解密，可满足静态数据加密的合规性要求。

前提条件

- 实例版本大于等于 MySQL5.7 2.1.12，或者大于等于 MySQL8.0 3.1.15.005。
- 实例状态为运行中。

注意事项

- 使用 TDE 功能需已开启 密钥管理服务 KMS，如未开通，可在开通 TDE 的过程中根据指引开启 KMS 服务。
- 使用 TDE 功能需已授予 KMS 密钥权限，如未授权，可在开通 TDE 的过程中根据指引进行授权。
- 操作账号需具有 QcloudAccessForMySQLRole 的权限，如无权限，可在开通 TDE 的过程中根据指引进行授权。

计费说明

加密使用的密钥由 密钥管理服务 KMS 产生和管理，TDSQL-C MySQL 版的数据加密功能不会额外收费，但密钥管理服务 KMS 有可能产生额外费用，详情请参考 计费概述。

使用限制

- 仅支持对称加密。
- 操作账号开通 TDE 后若撤销授权关系，实例重启会造成数据库不可用。
- TDE 只能开通，无法关闭。即开通 TDE 后，实例上证书密钥无法删除。
- KMS 主密钥删除后，用户实例将不可用。
- 开启 TDE 功能后，如果账号处于欠费状态，则无法从 KMS 获取密钥，可能导致迁移、升级等任务无法正常进行。

操作步骤

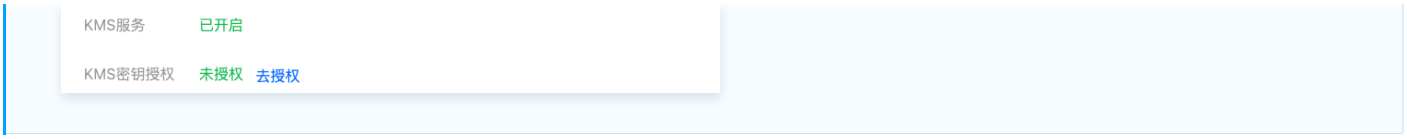
开启数据加密

1. 登录 TDSQL-C MySQL 版控制台，在集群列表，单击集群 ID，进入集群管理页面。
2. 在集群管理页面选择**数据安全**页。
3. 选择**数据加密**，单击**未开通**前面的按钮，打开数据加密功能。



4. 在弹出的对话框中，选择密钥方式后，单击**确定**。

> **说明：**
> 如果您在此之前未开启过 KMS 服务，未进行过 KMS 密钥授权，您需要根据对话框中的提示，在 KMS 服务后单击**去开启**，然后在 KMS 密钥授权后单击**去授权**，完成对应开启或授权操作以解锁数据加密功能的后续设置。





选择密钥

- 使用腾讯云自动生成的密钥：由系统自动生成密钥。
- 使用已有自定义密钥：可选择您自定义创建的密钥。
  - 地域：若使用已有自定义密钥，需要选择已有密钥的所在地域。
  - 密钥：若使用已有自定义密钥，您可通过密钥名称选择已有的密钥，如无已创建的密钥，可单击 前往创建，到密钥管理服务 KMS 控制台进行创建后，再重新选择。

5. 待集群运行状态由“运行中 异步任务：数据加密”变为“运行中”时，即成功开启数据加密，如需了解任务进度，您可前往 任务列表 进行查询。

加密数据表

开启数据加密功能后，用户需对数据库的表执行如下 DDL 操作才能进行数据加密。

- 如果您要在创建表时对表进行加密，请执行如下命令：

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

- 如果您要对已创建的表进行加密，请执行如下命令：

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

解密数据表

开启数据加密功能后，如需对已加密的表进行数据解密，请进行如下 DDL 操作。

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

监控与告警
简介

最近更新时间：2024-10-12 17:07:31

监控与告警概述

监控与告警是用于维护数据库，并为故障解决方案提供可靠性、可用性和性能帮助的重要组成部分。您可以通过监控了解 TDSQL-C MySQL 版数据库的资源使用率、性能及运行状况，通过告警及时掌控和处理数据库出现的突发情况，从而增加您系统的稳定性，提升运维效率，减少运维成本。
TDSQL-C MySQL 版控制台为您提供可视化监控告警页面，支持丰富的性能监控项与便捷的监控功能，帮助您了解数据库运行状况和性能，并可根据您设置的告警规则，通过消息推送的方式帮助您第一时间了解业务异常。让您无需额外开发，即可全面掌控数据库的资源使用及运行情况。

相关概念

- **指标**
  指标是监控的核心概念，表示一个向监控发送的按时间排序的数据点集合，您可以按时间序列来检索关于这些数据点的统计数据。指标作为监控的变量，指标数据代表该变量随时间变化的值。例如，CPU 利用率、内存使用率等。
- **单位**
  单位是指标原始数据的度量单位，如 CPU 利用率单位是%。
- **时间粒度**
  时间粒度是监控统计数据的间隔时间长度，每个时间戳数据代表根据指定粒度对收集的所有数据进行聚合的结果。
- **告警**
  告警管理是腾讯云监控告警服务中的一项功能，对云资源的异常情况进行告警，并提供告警信息查看、告警自定义阈值和告警订阅能力。

支持监控类型

支持监控类型	说明
实例	支持监控读写实例、只读实例
数据库代理	支持监控数据库代理节点，详细请参见 数据库代理监控

监控粒度

TDSQL-C MySQL 版监控时间粒度根据时间长度自适应调整，策略如下：

查询监控时间段	查询起始时间点距当前查询时间点的周期	默认时间粒度	可选择时间粒度
(0h, 1h]	(0d, 1d]	5s	5s/1min/5min
(0h, 1h]	(1d, 15d]	1min	1min/5min
(0h, 1h]	(15d, 31d]	5min	5min
(1h, 24h]	(0d, 15d]	1min	1min/5min/1h
(1h, 24h]	(15d, 31d]	5min	5min/1h
(24h, 7d]	(0d, 31d]	5min	5min/1h/1d
(7d，31d]	(0d, 31d]	1h	1h/1d

> **说明**
> 目前 TDSQL-C MySQL 版最长支持查看31天内的监控数据。

监控指标

TDSQL-C MySQL 版提供了丰富的监控指标，帮助您更全面的了解数据库的资源使用率、性能及运行状况，具体监控项与监控指标请参见 支持的监控指标 。

监控
支持的监控指标

最近更新时间：2025-02-13 16:46:02

TDSQL-C MySQL 版从实例维度以及数据库代理节点维度提供监控指标。

说明：
- 以下监控指标数据可通过控制台、云 API 查看，查看方法请参见 查看监控 。
- 并行查询相关监控指标，为 TDSQL-C MySQL 版8.0内核版本3.1.8及以上版本支持的指标。
- 列存索引相关监控指标，为 TDSQL-C MySQL 版8.0内核版本3.1.14支持的指标。
- 只读分析引擎相关监控指标，请参考 查看监控 。
- 通过 API 调用获取监控数据时，调用参数项使用的监控指标名称请以下表中的“可调用指标名称”列为准。

实例维度监控指标

类别		监控指标名称	可调用指标名称	单位	数据聚合方式
资源监控		CPU 使用率	CpuUseRate	%	MAX
		内存使用率	MemoryUseRate	%	MAX
		内存使用量	MemoryUse	MB	MAX
		存储使用率	StorageUseRate	%	MAX
		存储使用量	StorageUse	GB	MAX
		数据表空间使用量	DataVolumeUsage	GB	MAX
		临时表空间使用量	TmpVolumeUsage	GB	MAX
		undo 表空间使用量	UndoVolumeUsage	GB	MAX
		CCU	Ccu	个	MAX
		每秒发送客户端总流量	BytesSent	MB/秒	MAX
		每秒接收客户端总流量	BytesReceived	MB/秒	MAX
		读请求 IOPS	Readiops	个/秒	MAX
		写请求 IOPS	Writeiops	个/秒	MAX
		总 IOPS	Iops	个/秒	MAX
		读 I/O 吞吐量	ReadBandwidth	MB/秒	MAX
		写 I/O 吞吐量	WriteBandwidth	MB/秒	MAX
		总 I/O 吞吐	IoBandwidth	MB/秒	MAX
		日志占用的存储空间	LogCapacity	GB	MAX
		读写实例写入状态	ReadOnly	0 - 正常 1 - 异常	MAX
引擎监控	连接	每秒执行操作数	Qps	个/秒	MAX
		每秒执行事务数	Tps	个/秒	MAX
		连接数利用率	ConnectionUseRate	%	MAX

		最大连接数	MaxConnections	个	MAX
		当前打开连接数	ThreadsConnected	个	MAX
		已创建的线程数	ThreadsCreated	个	SUM
		运行的线程数	ThreadsRunning	个	MAX
	访问	慢查询数	SlowQueries	个	SUM
		全表扫描数	SelectScan	个	SUM
		查询数	ComSelect	个	SUM
		更新数	ComUpdate	个	SUM
		删除数	ComDelete	个	SUM
		插入数	ComInsert	个	SUM
		覆盖数	ComReplace	个	SUM
		总请求数	Queries	个	SUM
		提交数	ComCommit	个	SUM
		回滚数	ComRollback	个	SUM
		全表扫描复合查询次数	SelectFullJoin	个	SUM
		范围扫描复合查询次数	SelectFullRangeJoin	个	SUM
		排序合并通过次数	SortMergePasses	个	SUM
		Qcache 命中率	QcacheHitRate	%	MIN
		Qcache 使用率	QcacheUseRate	%	MIN
		多表更新数	ComUpdateMulti	个	SUM
		多表删除数	ComDeleteMulti	个	SUM
	表	临时表的数量	CreatedTmpTables	个	SUM
		等待表锁次数	TableLocksWaited	个	SUM
		已经打开的表数	OpenedTables	个	MAX
		立即释放的表锁数	TableLocksImmediate	个	SUM
		表打开缓存命中数	TableOpenCacheHits	个	SUM
		表打开缓存未命中数	TableOpenCacheMisses	个	SUM
	InnoDB	InnoDB 引擎缓存命中率	InnodbCacheHitRate	%	MIN
		InnoDB 引擎缓存使用率	InnodbCacheUseRate	%	MIN
		读磁盘数量	InnodbOsFileReads	个	MAX
		写磁盘数量	InnodbOsFileWrites	个	MAX
		InnoDB_fsyncs 数	InnodbOsFsyncs	个	MAX
		当前 InnoDB 打开表的数量	InnodbNumOpenFiles	个	MAX
		InnoDB 读取量	InnodbDataRead	Byte	SUM
		InnoDB 总读取量	InnodbDataReads	个	SUM

版权所有：腾讯云计算（北京）有限责任公司

	InnoDB 总写入量	InnodbDataWrites	个	SUM
	InnoDB 写入量	InnodbDataWritten	Byte	SUM
	InnoDB 行删除量	InnodbRowsDeleted	个	SUM
	InnoDB 行插入量	InnodbRowsInserted	个	SUM
	InnoDB 行更新量	InnodbRowsUpdated	个	SUM
	InnoDB 行读取量	InnodbRowsRead	个	SUM
	InnoDB 平均获取行锁时间	InnodbRowLockTimeAvg	毫秒	MAX
	InnoDB 等待行锁次数	InnodbRowLockWaits	个	SUM
	InnoDB 脏页数	InnodbBufferPoolPagesDirty	个	MAX
	InnoDB 挂起写入数	InnodbDataPendingWrites	个	MAX
	InnoDB 挂起读取数	InnodbDataPendingReads	个	MAX
	InnoDB 日志等待写入次数	InnodbLogWaits	个	SUM
	InnoDB 日志物理写入次数	InnodbLogWrites	个	SUM
	InnoDB 日志物理写请求次数	InnodbLogWriteRequests	个	SUM
Tmp	临时表数量	CreatedTmpDiskTables	个	SUM
	临时文件数量	CreatedTmpFiles	个	SUM
Handler	读下一行请求数	HandlerReadRndNext	个	SUM
	内部回滚数	HandlerRollback	个	SUM
	内部提交数	HandlerCommit	个	SUM
Buffer	InnoDB 空页数	InnodbBufferPoolPagesFree	个	MAX
	InnoDB 总页数	InnodbBufferPoolPagesTotal	个	MAX
	InnoDB 逻辑读	InnodbBufferPoolReadRequests	个	SUM
	InnoDB 物理读	InnodbBufferPoolReads	个	SUM
	InnoDB 缓冲池写入次数	InnodbBufferPoolWriteRequest	个	SUM
并行查询	当前并行查询线程数	TxsqlParallelThreadsCurrentlyUsed	个	MAX
	并行查询错误数	TxsqlParallelStmtError	个	SUM
	已执行并行查询数	TxsqlParallelStmtExecuted	个	SUM
	回滚串行查询数	TxsqlParallelStmtFallback	个	SUM
列存索引	列存索引执行语句数	InnodbCsiSelectPushdown	个	SUM
	列存索引回退语句数	InnodbCsiSelectFallback	个	SUM
	列存索引同步延时	InnodbCsiLagSeconds	秒	MAX
	列索引存储使用量	InnodbCsiDiskUsage	GB	MAX
	创建 CSI 失败数	InnodbCsiCreateFailed	个	SUM
其他	打开文件总数	OpenFiles	个	MAX

版权所有：腾讯云计算（北京）有限责任公司

		查询缓存命中次数	QcacheHits	个	SUM
		查找缓存次数	QcacheSearchTimes	个	SUM
		查询缓存总次数	QcacheTotalTimes	个	SUM
		查询缓存剩余空间	QcacheFreeMemory	MB	MAX
		缓存结果集成功次数	QcacheInserts	个	SUM
		缓存结果集失败次数	QcacheNotCached	个	SUM
		缓存的查询数量	QcacheQueriesInCache	个	SUM
		淘汰的查询数量	QcacheLowmemPrunes	个	SUM
部署监控		复制状态	ReplicationStatus	0-Yes，1-No	有1取1
		复制延迟	ReplicationDelay	毫秒	MAX
		复制落后的 lsn 距离	ReplicationDelayDistance	Bytes	MAX

数据库代理节点维度监控指标

要获取数据库代理节点维度监控指标，需先开启数据库代理，详细指标说明请参见 查看数据库代理监控 。

性能监控指标分析实践

您可以通过设置告警策略，对相关性能指标进行监控分析，以下举例介绍几个指标来分析监控数据。

- **CPU 使用率**：CPU 使用率是指数据库服务器中 CPU 在运行过程中的使用率。如果 CPU 负载高，会导致数据库系统响应变慢、查询延时等问题，因此，可以根据实际业务情况对此监控指标设置阈值告警，一般来说，CPU 使用率超过90%，就需要开始关注数据库需要处理的请求量是否过高、查询性能是否优化等问题，并及时进行调整。如果 CPU 消耗的值符合应用程序或数据库的目标（如吞吐量或并发性）并且是预期的，则这个值是合适的。
- **存储使用量**：如果使用的空间始终等于或大于总磁盘空间的85%，则需要调查存储空间的消耗，查看是否可以从实例中删除数据或将数据存档到其他系统以释放空间，如果存储数据量太大，磁盘即将写满，可以通过扩容缓解磁盘空间不足的问题。
- **客户端发送和接收流量**：通过观察监控指标-资源监控-发送或接收客户端流量数据的波动情况，进而检查是业务峰值带来的问题还是数据库的问题，从而进行业务优化或者数据库配置调整。
- **最大连接数**：max_connections 是 TDSQL-C MySQL 版的参数之一，用于指定数据库同时允许的最大连接数。当数据库进程打开一个新的连接时，它会去查询当前已经连接的客户端数量，如果数量已经达到 max_connections 的设定值，则无法建立新的连接。如果您通过监控看到大量连接，并且实例性能和响应时间也有所下降，请考虑约束数据库最大连接数。如果 max_connections 的值设置过高，可能会导致系统出现内存不足、线程崩溃等问题，因此应仔细评估应用程序的需要，进行最佳设置，常用的设置值在200～500之间，也请结合实际应用程序和数据库的连接情况来调整最大连接数。

当性能指标超出设定的告警阈值时，可能需要更改相关参数以优化数据库，帮助提高数据库的性能和可用性。

查看监控

最近更新时间：2025-03-18 17:38:12

TDSQL-C MySQL 版提供丰富的性能监控指标，方便您细致的了解数据库的运行状况和性能。查看监控指标及数据常见的方法有：通过控制台查看、通过创建腾讯云可观测平台 Dashboard 查看以及通过云 API 接口拉取数据查看。推荐您使用控制台查看，可快捷获取目标时间段的监控数据，方便您实时掌握集群的运行状态并通过细粒度的监控数据快速定位运维问题。
本文介绍查看监控数据的方法以及在监控界面的可视化操作。

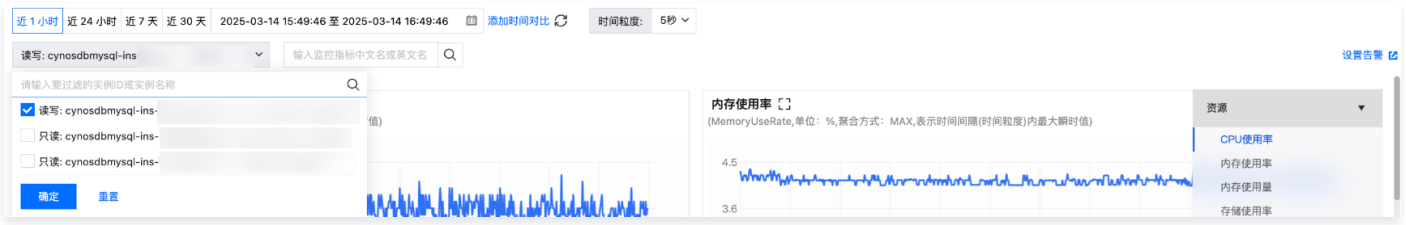
查看监控数据

方案	优势	操作
TDSQL-C MySQL 版控制台查看	便捷、可视化，快速定位问题，符合数据库运维习惯	通过控制台查看监控数据
创建腾讯云可观测平台 Dashboard 查看	支持自定义创建监控指标分组、可视化	创建腾讯云可观测平台 Dashboard 查看监控数据
云 API 拉取数据	灵活拉取监控指标进行分析处理或接入其他平台	使用云 API 拉取监控数据

在监控界面的可视化操作

查看标准监控视图

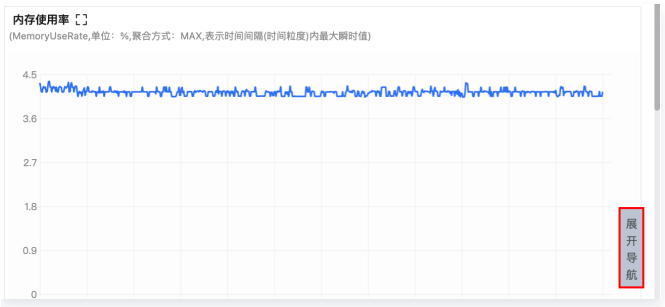
1. 登录 TDSQL-C MySQL 版控制台 。
2. 在上方选择地域，找到目标集群，单击**集群 ID** 或**操作**列的**管理**，进入集群管理页。
3. 在集群管理页选择**监控告警**，单击下拉键选择目标实例（支持单选和多选），单击**确定**。



隐藏和展开导航

您可以隐藏和展开监控指标的导航，展开时，您可以通过单击对应监控指标，快捷查询该指标的监控情况，隐藏时，您可以获得更全的监控视图。

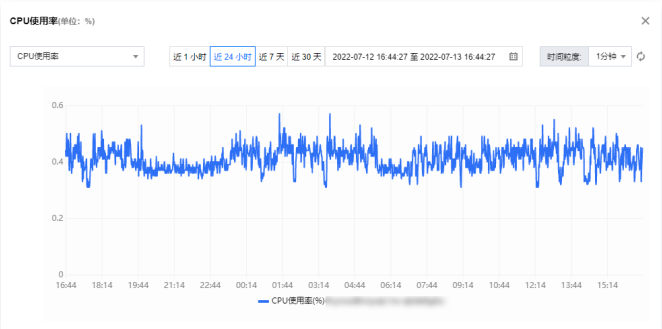




大屏显示图表

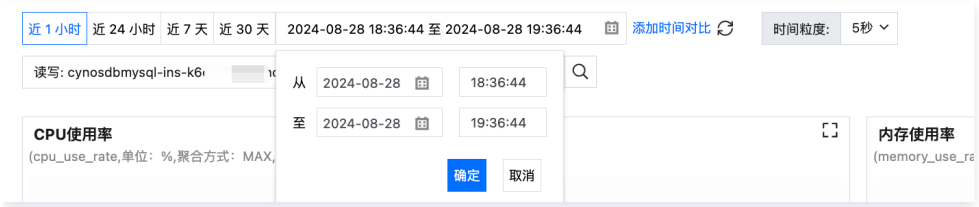
您可以将单个指标进行大屏显示，方便更清晰地预览指标数据。

1. 在监控告警页，单击任意指标右侧的 图标，可全屏显示指标情况。
2. 全大屏显示预览数据后，可筛选指标、针对筛选的指标选择时间范围及时间粒度查看指标情况，单击右上角的 X，关闭大屏显示窗口。



选择监控时间范围

您可以通过选择或自定义时间范围，对该时间段的监控情况进行查询。
在监控告警页，可选择近1小时、近24小时、近7天、近30天或者单击时间框自定义起止时间范围查询监控情况。



添加时间对比

您可以通过添加时间对比，对比多个时间范围的监控数据。

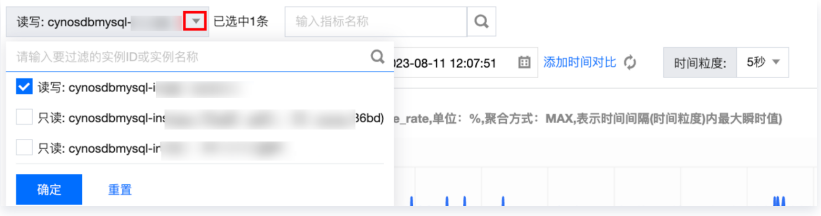
> **说明：**
> 首个时间框内的时间周期决定了新增时间框的时间周期，如：首个时间框内时间周期为3天，则新增的时间框只能选起始时间，结束时间默认为起始时间开始的三天后。

使用限制

- 若选择多个实例查看监控，则不支持时间对比功能。
- 最多支持同时对比3个时间范围的实例监控指标情况。

操作步骤

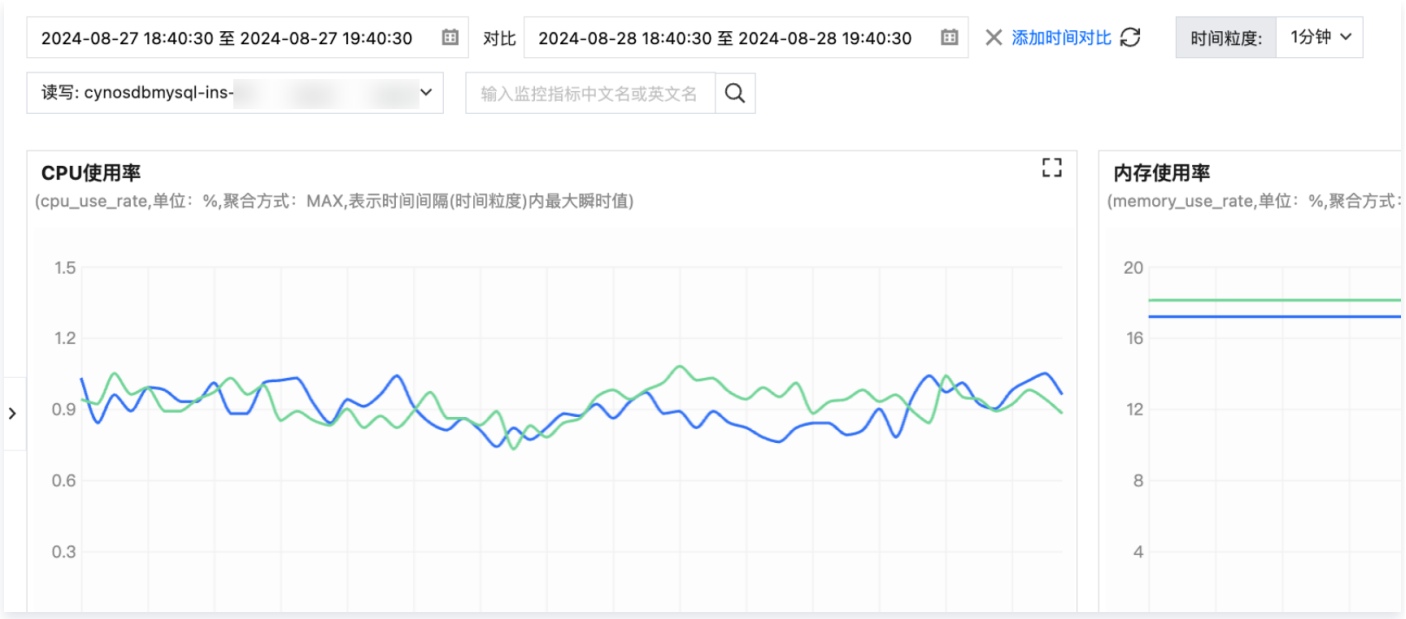
1. 登录 TDSQL-C MySQL 版控制台 。
2. 在上方选择地域，找到目标集群，单击**集群 ID** 或**操作**列的**管理**，进入集群管理页。
3. 在集群管理页，选择**监控告警**页。
4. 在下拉框内选择一个目标实例，单击**确定**。



5. 在监控告警页的自定义时间框后单击**添加时间对比**。



6. 优先在第一个自定义时间框内选择需要查询的时间范围，单击**确定**，然后在第二个时间框内选择需要查询的起始时间，单击**确定**。



说明：
后添加的自定义时间框只需选择起始时间，其时间周期与第一个自定义时间框保持一致，如：第一个自定义时间框时间范围为3天，则后添加的时间范围自动为3天。

7. 选择好对比时间后，即可在下方查询对应实例所选对比时间范围的监控指标详情，支持调整时间粒度进行查看。

设置监控粒度

支持对监控数据设置查询的监控粒度，便于您在所选时间范围内查询不同统计粒度下的监控情况。
在监控告警页，单击**时间粒度**后的下拉键，选择您需要查看的监控粒度。



时间周期和图表时间粒度对照表

查询监控时间段	查询起始时间点距当前查询时间点的周期	默认时间粒度	可选择时间粒度
(0h, 1h]	(0d, 1d]	5s	5s/1min/5min
(0h, 1h]	(1d, 15d]	1min	1min/5min
(0h, 1h]	(15d, 31d]	5min	5min
(1h, 24h]	(0d, 15d]	1min	1min/5min/1h
(1h, 24h]	(15d, 31d]	5min	5min/1h
(24h, 7d]	(0d, 31d]	5min	5min/1h/1d
(7d，31d]	(0d, 31d]	1h	1h/1d

示例：
假设当前时间为2022年7月10日15:00，需要查询监控的起始时间和结束时间为2022年6月29日15:00至2022年6月29日16:00，则查询时间段为1小时，起始时间2022年6月29日15:00距离当前时间2022年7月10日15:00大于1天且小于15天，即默认的时间粒度为1min，可供选择的时间粒度为1min、5min。

说明：

目前 TDSQL-C MySQL 版最长支持查看31天内的监控数据。

控制台查看监控数据

最近更新时间：2024-10-22 17:31:21

本文为您介绍通过控制台查看监控数据。

操作步骤

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在上方选择地域，根据实际控制台视图模式，对应操作。

页签视图

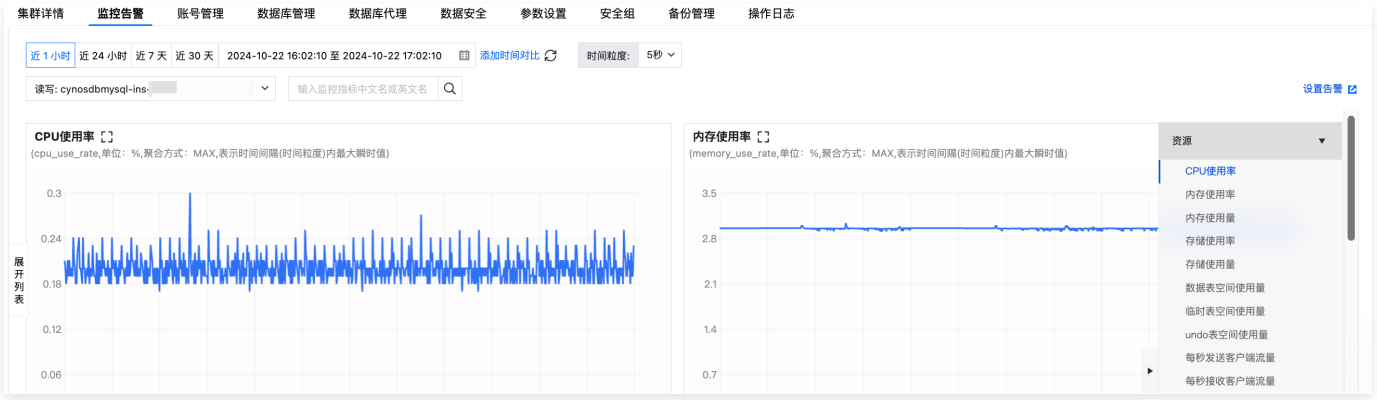
在左侧集群列表，单击**目标集群**，进入集群管理页。

列表视图

在左侧集群列表找到目标集群，单击**集群 ID** 或**操作**列的**管理**，进入集群管理页。



3. 在集群管理页，选择**监控告警**页。
4. 选择目标实例（支持单选和多选），选择需查询的时间范围后，即可对目标实例的各项监控指标进行查阅，支持添加时间对比以及选择时间粒度。



创建 Dashboard 查看监控数据

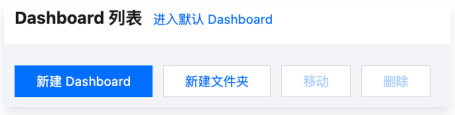
最近更新时间：2025-03-18 17:38:12

本文为您介绍通过创建腾讯云可观测平台 Dashboard 查看监控数据。

操作步骤

> **说明：**
> 如需了解详细创建步骤，请参见 创建 Dashboard 。

1. 登录 腾讯云可观测平台 。
2. 在左侧导航栏选择 **Dashboard** 进入 **Dashboard 列表**页。
3. 在 **Dashboard 列表**下单击**新建 Dashboard**。



4. 在弹出的窗口下选择**新建图表**，或选择**新建图表组**然后在图表组里**新建图表**。



5. 在图表编辑页面，配置如下参数，单击右上侧的**保存**。

参数	说明
选择产品	云产品监控
指标	选择云数据库 > TDSQL-C > MySQL，然后选择目标监控指标
筛选	选择筛选条件，过滤出符合条件的数据在图表上展示。增加多个筛选条件，图表仅展示符合所有条件的数据。筛选对象支持按实例、标签、模板变量进行筛选
group by	选择标签，进行分组聚合展示（选填项）
对比	可选择环比（昨天同时段）、同比（上周同时段）、自定义日期对比
别名	填写图例别名
开启排序功能	开启后可设置排序规则
排序规则	支持选择最大最小值、平均值、求和值，按照升序或降序排序
展示数量	设置指标展示的实例数量
新增指标	如果需要添加监控指标，可以点击新增指标，重复以上的设置。

6. 保存成功后，即可查看所创建指标的监控数据。

版权所有：腾讯云计算（北京）有限责任公司

云 API 拉取监控数据

最近更新时间：2024-10-14 09:57:11

本文为您介绍通过云 API 接口拉取 TDSQL-C MySQL 版对应指标的监控数据，详细操作和更多示例请参见 拉取指标监控数据 。

接口描述

传入 TDSQL-C MySQL 版的命名空间、对象维度描述和监控指标即可获得相应的监控数据。
接口调用频率限制为：20次/秒，1200次/分钟。单请求最多可支持批量拉取10个实例的监控数据，单请求的数据点数限制为1440个。
若您需要调用的指标、对象较多，可能存在因限频出现拉取失败的情况，建议尽量将请求按时间维度均摊。
默认接口请求频率限制：20次/秒。

输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 公共请求参数 。

参数名称	必选	类型	描述
Action	是	String	公共参数，本接口取值：GetMonitorData
Version	是	String	公共参数，本接口取值：2018-07-24
Region	是	String	公共参数，详见产品支持的 地域列表
Namespace	是	String	命名空间，如 QCE/cynosdb_mysql。详细命名空间说明请参阅 监控指标
MetricName	是	String	指标名称，如 CPUUsagerate（CPU 利用率），仅支持单指标拉取
Instances.N	是	Array of Instance	实例对象的维度组合，格式为 key-value 键值对形式的集合。不同类型的实例字段完全不同，TDSQL-C MySQL 版的维度组合请参见 监控指标
Period	否	Integer	监控统计周期，如60。默认为取值为300，单位为s
StartTime	否	Timestamp ISO8601	起始时间，如2021-07-15T19:51:23+08:00
EndTime	否	Timestamp ISO8601	结束时间，如2021-07-15T20:51:23+08:00，默认为当前时间。EndTime 不能小于 StartTime

输出参数

参数名称	类型	描述
Period	Integer	统计周期
MetricName	String	指标名
DataPoints	Array of DataPoint	数据点数组
StartTime	Timestamp ISO8601	开始时间
EndTime	Timestamp ISO8601	结束时间
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId

示例

拉取某个 TDSQL-C MySQL 版实例某段时间内统计周期为5分钟的 CPU 利用率监控数据。
输入示例

```
POST / HTTP/1.1
Host: monitor.tencentcloudapi.com
```

```
Content-Type: application/json
X-TC-Action: GetMonitorData
<公共请求参数>

{
    "Namespace": "QCE/cynosdb_mysql",
    "MetricName": "CPuUsageRate",
    "Period": 3600,
    "Instances": [
        {
            "Dimensions": [
                {
                    "Name": "InstanceId",
                    "Value": "cynosdbmysql-ins-edpn3t6b"
                }
            ]
        }
    ],
    "StartTime": "2022-07-15T10:00:00",
    "EndTime": "2022-07-15T15:00:00"
}
```

输出示例

```
{
  "Response": {
    "DataPoints": [
      {
        "Dimensions": [
          {
            "Name": "InstanceId",
            "Value": "cynosdbmysql-ins-edpn3t6b"
          }
        ],
        "Timestamps": [
          1657850400,
          1657854000,
          1657857600,
          1657861200,
          1657864800
        ],
        "Values": [
          0.26,
          0.24,
          0.23,
          0.26,
          0.24
        ]
      }
    ],
        "EndTime": "2022-07-15 15:00:00",
        "MetricName": "CPuUsageRate",
        "Period": 3600,
    "RequestId": "71c72744-bec5-49d0-b42c-433609ab4166"
         "StartTime": "2022-07-15 10:00:00"
  }
}
```

告警
监控指标告警
设置告警流程简介

最近更新时间：2024-10-12 11:28:31

TDSQL-C MySQL 版支持针对监控指标设置告警，通过实时监控性能指标来了解数据库的资源使用及运行情况；通过设置告警通知来及时获取告警信息；通过查看告警历史来回溯告警历史详情，帮助改善数据库运行状态。

设置告警

按照以下步骤，您可以获取告警通知并实现异常问题定位，从而帮助您及时掌控和处理数据库出现的突发情况，增加您系统的稳定性，提升运维效率，减少运维成本。

步骤	说明	操作
步骤1：设置告警策略	用于在 TDSQL-C MySQL 版集群状态改变时触发警报并发送相关消息	设置告警策略
步骤2：设置告警通知	用于添加异常告警的接收对象、设置告警通知的时间周期以及通知方式	设置告警通知
步骤3：查看告警历史	用于回溯没及时查看的告警通知和查看近半年的告警历史记录	查看告警历史

版权所有：腾讯云计算（北京）有限责任公司

设置告警策略

最近更新时间：2024-10-22 17:27:22

本文为您介绍通过控制台创建告警策略和创建后管理告警策略的方法。

操作场景

您可以创建告警用于在 TDSQL-C MySQL 版状态改变时触发警报并发送相关消息。创建的告警会根据每隔一段时间监控的指标相对于给定阈值的情况判断，是否需要触发相关通知。
状态改变触发告警后，您可以及时进行相应的预防或补救措施。因此，合理地创建告警能帮助您提高 TDSQL-C MySQL 版的健壮性和可靠性。有关告警的更多信息，请参见腾讯云可观测平台的 告警配置 。
当您需要针对 TDSQL-C MySQL 版的某个状态发送告警时，需要先创建告警策略。告警策略包括名称、类型和告警触发条件三个必要组成部分。每个告警策略是一系列告警触发条件的集合，告警触发条件是“或”关系，即一个条件满足，就会发送告警。告警将发送至告警策略关联的所有人，用户接收到告警后可以及时查看并采取相应措施。创建告警策略后，支持管理告警策略（修改告警策略名称或备注、修改告警触发条件、编辑告警对象等）

操作步骤

设置告警策略

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在上方选择地域，根据实际控制台视图模式，对应操作。

页签视图

在左侧集群列表，单击**目标集群**，进入集群管理页。

列表视图

在左侧集群列表找到目标集群，单击**集群 ID** 或**操作**列的**管理**，进入集群管理页。



3. 在集群管理页选择**监控告警**，单击右侧的**设置告警**。



4. 在跳转页面选择**告警配置** > **告警策略**，单击**新建策略**。
5. 在新建策略页中，完成以下参数设置，单击**完成**。

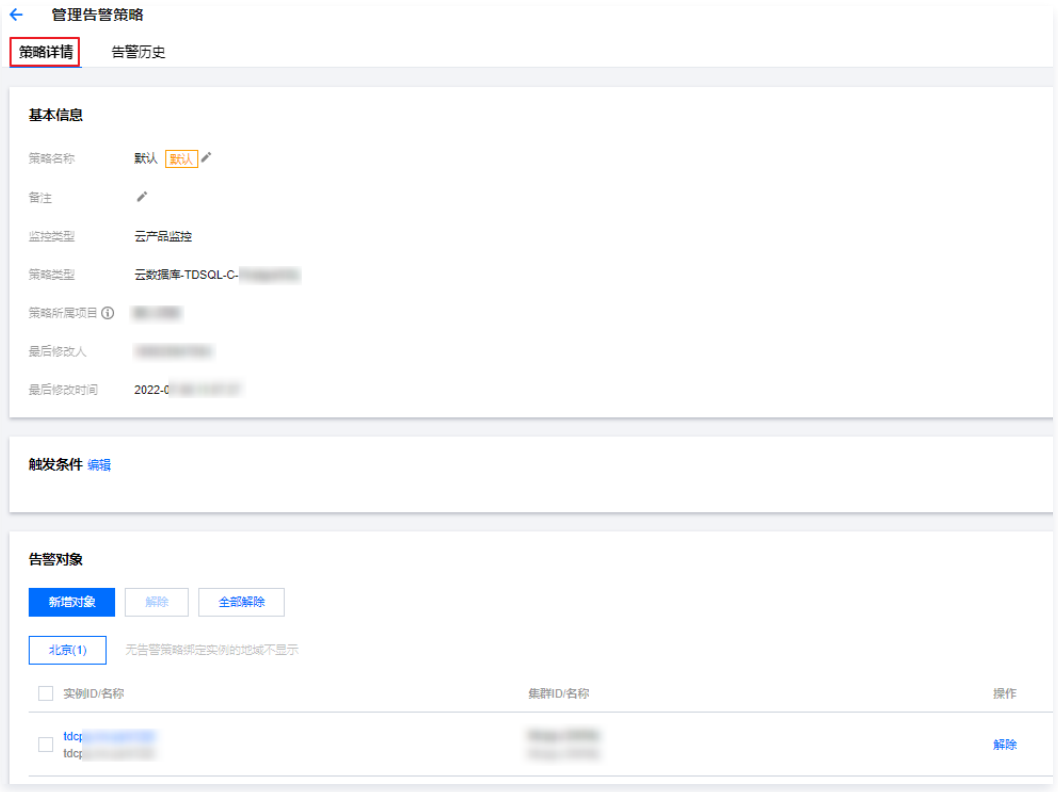


参数	说明
策略名称	填写策略名称，最多输入60个字符。
备注	给此策略添加备注信息，最多输入100个字符。
监控类型	选择云产品监控。
策略类型	选择云数据库 > TDSQL-C > MySQL。
告警对象	通过选择对象所在的地域或搜索对象的实例 ID 找到需要关联的实例。
触发条件	告警触发条件是指标、比较关系、阈值、统计粒度和持续 N 个监控数据点组成的一个有语义的条件。 您可以自定义设置指标告警和事件告警触发条件，根据业务需求配置告警指标、统计粒度、告警阈值、告警分级和告警频率，也可以直接使用触发条件模板和预置触发条件，请参见 配置告警触发条件 。
配置告警通知	支持选择系统预设通知模板和用户自定义通知模板，每个告警策略最多只能绑定三个通知模板，详情请参考 通知模板 。

管理告警策略

1. 在 告警策略列表页，单击目标告警策略名称，进入管理告警策略页。

2. 在管理告警策略页选择**策略详情**。



3. 根据需要选择修改项进行重新设置，如修改策略名称、修改备注、修改触发条件、解除或新增告警对象等。

设置告警通知

最近更新时间：2024-10-22 17:21:12

设置完告警策略后需要添加告警接收对象，帮助您获取告警信息，此操作可在设置告警通知下进行添加。
在配置告警策略页面 > 配置告警通知项，支持选择通知模板或者新建告警通知模板，便于多个策略一键复用模板，帮助减少重复配置用户通知。本文将指导您通过控制台设置告警通知模板。
如需详细了解相关限制和说明，请参见 新建通知模板 。

设置告警通知模板

1. 登录 腾讯云可观测平台 > 通知模板 。
2. 单击**新建通知模板**，在新建通知模板页面完成如下信息配置，单击**完成**。



配置类型	配置项	说明
基本信息	模板名称	自定义模板名称。
	通知类型	选择要接收的通知类型，支持告警触发和告警恢复两种类型，支持多选。 • 告警触发：告警触发时发送通知 • 告警恢复：告警恢复时发送通知

	通知语言	选择要接收的通知语言类型，支持中文和英文
	所属标签	选择对应标签键和标签值，方便对绑定标签下的实例进行管理； 支持选择多个标签键和标签值。如需修改请前往标签控制台修改。
通知操作	用户通知	• 接收对象：可选用户、用户组和值班表，如需创建用户组请参见 新建用户组，如需创建值班表请参见 配置值班表。 • 通知周期：定义接收告警日期 • 通知时段：定义一天中接收告警时段 • 接收渠道：支持邮件、短信、微信、企业微信和电话五种告警渠道。支持添加多种用户通知方式，为不同用户维度设置不同的告警接收渠道和通知时段。 • 电话告警配置： ○ 轮询拨打：对所有接收人逐一轮询拨打告警电话，支持设置轮询次数、轮询顺序、轮询间隔、触达通知。 轮询次数：在无有效触达时，对所有接收人逐一轮询拨打的最多次数。 轮询顺序：电话告警按照接收人顺序轮询拨打，上下拖动接收人调整拨打顺序。 轮询间隔：电话告警按照接收人顺序轮询拨打的时间间隔，可设置单次轮询内间隔和两次轮询间隔时间。 ○ 同时拨打：支持多个电话同时拨打，多人同时接到电话告警，及时处理故障。 ○ 触达通知：电话成功接收或轮询结束发送信息给所有接收人。短信需计算额度。
	接口回调	• 填写公网可访问到的 URL 作为回调接口地址，最多可填写6个告警回调地址。腾讯云可观测平台将及时把告警信息推送到该地址（例如：企业微信、钉钉、slack、飞书等），当 HTTP 返回200为验证成功。 接口回调还支持基于 BasicAuth 的用户安全验证，保证用户数据安全。告警接口回调字段说明请参见 配置回调接口。 • 对于企微、飞书、钉钉和 slack 群机器人，填写群成员 userid 后，告警消息将会提醒对应群成员查看，群成员提醒配置请参见 告警接收 相关文档。
	投递日志服务	当告警触发时，系统会将告警触发/恢复通知数据推送至 腾讯云日志服务 CLS 的日志主题中。可对告警通知进行检索分析。 操作步骤： 1. 在投递日志服务中，勾选**启用**。 2. 单击弹框中的**同意授权**（新用户第一次启用才需进行授权），授权腾讯云可观测平台将告警信息推送到日志服务中。 3. 选择需要投递的日志集所在地域、日志集和日志主题。

注意：

- 回调地址保存后自动验证一次您的 URL，验证超时时间为5s；当用户创建的告警策略被触发或被恢复均会通过接口回调推送告警消息，此告警消息最多推送三次， 每次请求的超时时间为5s。
- 当用户创建的告警策略被触发或恢复时，均会通过接口回调推送告警消息。接口回调也支持重复告警。
- 告警回调 API 出方向 IP 为动态随机分配，无法将具体的 IP 信息提供给您，但 IP 端口固定为80端口，建议您根据80端口在安全组上配置加全放通策略。

配置告警通知

1. 登录 腾讯云可观测平台 > 告警管理 > 告警配置。
2. 单击**新建策略**，在告警策略配置窗口里的**配置告警通知**选择告警通知模板（最多可选择3个通知模板）。
3. 完成配置告警策略和告警通知后，单击**完成**。



4. 配置完成后，如有异常触发了告警，系统会根据您选择的告知渠道（邮件、短信、微信、企业微信、电话）发送给接收对象。

查看告警历史

最近更新时间：2024-10-22 17:21:12

您可以通过控制台回溯触发的告警历史或查看近半年的告警历史详情，通过告警可快速定位到具体问题，并做进一步排查，通过对告警历史情况的分析，帮助您掌控和改善数据库运行状态。
本文为您介绍通过控制台查看告警历史。如需详细了解相关功能说明，请参见 查看告警历史 。

操作步骤

1. 登录 腾讯云可观测平台 ，选择**告警管理 > 告警配置**。
2. 在告警策略列表，找到需要查看告警历史的对应告警策略，在其操作列单击**告警历史**。
3. （可选）单击左上角的**时间筛选**，筛选需要查看告警历史的时间范围。支持今天、昨天、近7天、近30天快速筛选，您也可以自定义时间范围。最多可查看近半年的告警历史。
4. （可选）您可以在搜索框中输入告警对象的信息搜索对应的历史记录，例如：策略名称、告警内容、告警对象等。
5. （可选）还可以单击**筛选**，根据策略类型、用户、告警对象、所属实例组、所属项目搜索出符合条件的告警历史记录。



事件告警
事件告警简介

最近更新时间：2024-10-12 10:19:25

简介

腾讯云事件总线（EventBridge）是一款安全，稳定，高效的无服务器事件管理平台，作为流数据和事件的自动收集、处理、分发管道，通过可视化的配置，通过设置事件规则可实现对 TDSQL-C MySQL 版异常事件的监控，帮助用户获取应用异常状态以及时优化和运维。

产品功能

- 事件收集：提供标准事件投递接口，完成云产品事件、SaaS 服务、自定义应用事件等不同事件源的规范化接入。
- 事件管理：提供事件的可管理特性，通过格式匹配、内容筛选、格式转换、追踪、归档、重放等能力，为客户在事件驱动（EDA）架构下提供更多支持。
- 事件投递：支持多种类型投递目标接入，具有高可扩展能力，可基于实际业务场景提供不同解决方案。

基本概念

请参见 事件总线基本概念 。

限制说明

请参见 事件总线限制说明 。

为 TDSQL-C MySQL 版集群设置事件告警

- 了解 支持的事件类型 。
- 设置事件告警 。

支持的事件类型

最近更新时间：2025-06-13 14:56:52

本文为您介绍 TDSQL-C MySQL 版支持上报告警的事件类型。

说明：
- 详细监控事件及处理建议请参见 TDSQL-C MySQL 事件总线解释 。
- 为保证事件名称清晰和方便理解，2024年2月27日起，TDSQL-C MySQL 版事件指标名称“HA”优化为“计划外高可用切换”，“故障规避”优化为“计划内高可用切换”，感谢您的理解和支持。

事件中文名	事件英文名	事件描述
云 API 操作事件（基于操作审计投递）	Cloud API actions (CloudAudit)	用户调用云 API 的操作
计划外高可用切换	HA	实例发生计划外的高可用切换
控制台操作事件（基于操作审计投递）	Console operations (CloudAudit)	用户使用控制台的操作
数据库代理异常	Database agent exception	数据库代理出现异常
小程序操作事件（基于操作审计投递）	Mini program operations (CloudAudit)	用户使用小程序的操作
内存 OOM	OOM	系统内存使用过载
实例只读（存储超限）	Instance read-only due to storage full	存储空间容量超过限制，无法写入，只能读取
版本升级事件	Version upgrade events	集群执行小版本升级
跨可用区延迟	Cross-AZ latency	跨可用区主备延迟过高 **说明：** 对于跨可用区延迟事件，推送告警的条件如下： • 延迟超过2GB，并且持续时间超过2小时会推送告警。 • 延迟超过3600秒，并且持续时间超过2小时会推送告警。
数据库代理挂载节点剔除	The database agent mount node is removed	因故障导致数据库代理挂载节点剔除
RDMA 降级	RdmaDowngrade	实例发生 RDMA 降级
计划内高可用切换	FailureAvoid	发生设备故障注意规避
数据库审计低风险	AuditLowRisk	审计风险等级
数据库审计中风险	AuditMediumRisk	审计风险等级
数据库审计高风险	AuditHighRisk	审计风险等级
集群隔离	ClusterIsolated	集群进入隔离状态
Serverless 实例成功跨机迁移	ServerlessExpandNodeBySwitchSuccess	Serverless 实例跨机迁移执行成功
存储用量超限恢复	StopStorageOver	存储空间恢复未超限状态

版权所有：腾讯云计算（北京）有限责任公司

存储用量超限	ReportStorageOver	存储空间使用超限

设置事件告警

最近更新时间：2024-10-22 17:21:12

TDSQL-C MySQL 版支持事件告警功能，当您设置了事件模式、事件目标及接收对象，如果检测到异常事件，事件告警将通过多种可选的方式向接收对象发送告警通知。本文介绍如何设置事件告警。

步骤1：开通事件总线

腾讯云事件总线通过访问管理（Cloud Access Management，CAM）来实现权限管理。CAM 是腾讯云提供的权限及访问管理服务，主要用于帮助客户安全管理腾讯云账户下的资源的访问权限。用户可以通过 CAM 创建、管理和销毁用户（组），并使用身份管理和策略管理控制其他用户使用腾讯云资源的权限。使用事件总线 EventBridge 前，您需在产品页开通该服务。主账号开通方法及为子账号授权使用此服务，请参见 开通事件总线 。

步骤2：设置事件告警

> **注意：**
> - 对于 TDSQL-C MySQL 版产生的告警、审计等运维事件，将全部投递至**云服务事件集**，该投递为默认投递，不支持更改或编辑。
> - 开启腾讯云事件总线服务后，将为您自动在**广州地域**创建默认云服务事件集，TDSQL-C MySQL 版所产生的告警事件将自动投递至此。

方法1：事件总线配置

1. 登录 腾讯云可观测平台 > 事件集 。
2. 在上方选择地域为**广州**。
3. 单击云服务事件集下的 **default** 事件集。



4. 在 default 事件集详情页单击**管理事件规则**，在跳转页面单击**新建**（这里以新建事件规则为例，如已有事件规则，可直接修改，无需重新创建）。
5. 在新建事件规则页面完成如下配置后单击**下一步**。



参数	说明
规则名称	填写规则名称，只能包含字母、数字、下划线、连字符，以字母开头，以数字或字母结尾，2个 - 60个字符。
规则描述	填写规则描述，只能包含数字、中英文及常用标点符号，不超过200个字符。
标签	自定义是否启用标签，启用后可以对该事件规则添加标签。

数据转换	事件数据转换可以帮助您轻松的对事件内容进行简单的处理。例如，您可以对事件中的字段进行提取解析和映射重组后，再投递到事件目标。
事件示例	提供了事件结构示例，为配置事件匹配规则做参考，您可以在事件示例选择下找到目标模板以作参考，云服务类型选择 TDSQL-C MySQL 版。
事件模式	支持表单模式和自定义事件，这里建议使用表单模式更为便捷。
云服务类型	选择 TDSQL-C MySQL 版。
事件类型	选择告警的事件类型，支持全选，详细支持的事件类型请参见 TDSQL-C MySQL 版事件总线解释 。
匹配对象	选择事件告警的匹配对象，支持选择所有实例或输入目标实例。

6. 在事件目标页完成相关配置，勾选**立即启用事件规则**，单击**完成**。（触发方式有多种类型，您可按需选择，下图以触发方式选择“消息推送”为例，渠道推送将会推送给设定的联系人，需要在接收对象处单独设定接收用户，通知时段请对应账号用户协商确认，接收渠道可选择邮件、站内信、短信、电话）

> **说明：**
> 如需配置多个事件目标，可单击**添加**进行设置。

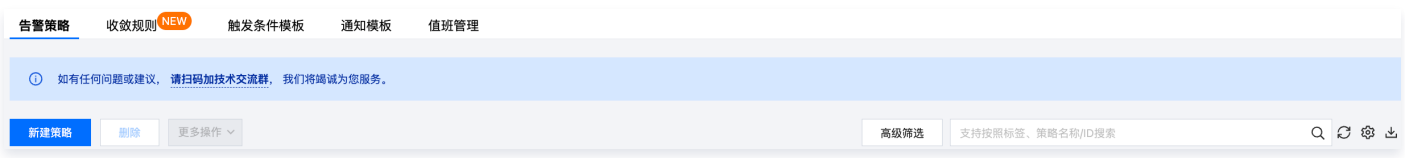


7. 事件消息推送设定完毕后，如发生配置的事件告警信息，系统将会按您设置的接收对象的接收渠道去通知。

方法2：腾讯云可观测平台告警策略配置

创建告警策略

1. 登录 腾讯云可观测平台控制台 ，在左侧导航选择**告警管理** > **告警配置**页。
2. 在告警策略列表页，单击**新建策略**。



3. 在新建策略页中，完成基本信息、告警规则、告警通知的设置。
   - **策略类型**：分为主机监控和备机监控，分别适用于不同的实例类型。
   - **告警对象**：可通过选择对象所在的地域或搜索对象的实例 ID 找到需要关联的对象实例。

- **触发条件**：找到事件告警，单击**添加事件**，根据业务需要添加告警事件，详细支持的事件类型请参见 TDSQL-C MySQL 版事件总线解释 。



- **配置告警通知**：支持选择系统预设通知模板和用户自定义通知模板，每个告警策略最多只能绑定三个通知模板，自定义通知模板请参见 新建通知模板 。
  - 选择系统预设模板



  - 新建模板



4. 确认无误后，单击**完成**。

关联告警对象

创建完告警策略后，您也可以为其关联其他告警对象（需要此告警策略的其他实例），当对象达到告警触发条件时会发送告警。

1. 在 告警策略列表页，单击**告警策略名称**，进入管理告警策略页。
2. 在管理告警策略页的**告警对象**栏，单击**新增对象**。
3. 在弹出的对话框，选择您需要关联的告警对象，单击**确定**，即可关联告警对象。

SQL 基本操作

最近更新时间：2024-05-13 15:21:51

本文为您列举了部分常用 SQL 命令。
如需了解更详细的 SQL 命令信息，包括命令参数和限制条件等，请参见 MySQL 官方指南 。

查询版本

方法一：

```
MySQL [(none)]> SELECT CYNOS_VERSION();
+-------------------------+
| CYNOS_VERSION() |
+-------------------------+
| 5.7.mysql_cynos.1.3.10 |
+-------------------------+
1 row in set (0.00 sec)
```

方法二：

```
MySQL [(none)]> SELECT @@CYNOS_VERSION;
+-------------------------+
| @@CYNOS_VERSION |
+-------------------------+
| 5.7.mysql_cynos.1.3.10 |
+-------------------------+
1 row in set (0.00 sec)
```

方法三：

```
MySQL [(none)]> SHOW VARIABLES LIKE 'CYNOS_VERSION';
+---------------+-------------------------+
| Variable_name | Value |
+---------------+-------------------------+
| cynos_version | 5.7.mysql_cynos.1.3.10 |
+---------------+-------------------------+
1 row in set (0.01 sec)
```

数据库相关

命令	示例
创建数据库并指定字符集	`CREATE DATABASE db01 DEFAULT CHARACTER SET gbk COLLATE gbk_chinese_ci;`
删除数据库	`DROP DATABASE db01;`

账号相关

命令	示例
创建账号	`CREATE USER 'username'@'host' IDENTIFIED BY 'password';`
删除账号	`DROP USER 'username'@'host';`
赋权	`GRANT SELECT ON db01.* TO 'username'@'host';`
查询数据库中的账号	`SELECT user,host FROM mysql.user;` 或

	`SHOW GRANTS FOR 'username'@'hostname';`
权限回收	收回全部权限：`REVOKE ALL PRIVILEGES,GRANT OPTION FROM 'username'@'host';` 收回指定权限：`REVOKE UPDATE ON *.* FROM 'username'@'host';`

使用 SCF 连接 TDSQL-C MySQL 版

最近更新时间：2024-10-14 10:16:01

如果您需要在云函数（SCF）中使用 TDSQL-C MySQL 版，您可使用连接池或云函数团队提供的 SDK 来连接 TDSQL-C MySQL 版。连接池具备自动重连功能，可有效避免因云函数底层或者数据库释放连接，造成连接不可用的情况。
详细步骤请参见 使用 SCF 连接数据库 。

标签
标签简介

最近更新时间：2024-10-12 17:24:25

标签是腾讯云提供的用于标识云上资源的标记，是一个键-值对（Key-Value），详细介绍请参见 标签概述 。
您可以根据各种维度（例如，业务、用途、负责人等）使用标签对 TDSQL-C MySQL 版资源进行分类管理。通过标签可以非常方便地过滤出对应的资源。标签键-值对（Key-Value），在腾讯云没有任何语言的含义，会严格按字符串进行解析匹配，在使用过程中需要注意 使用限制 。
下面通过一个具体的案例来介绍如何使用标签。

示例

案例背景
某公司在腾讯云上拥有5个 TDSQL-C MySQL 版集群，分属电商、游戏、文娱三个部门，服务于营销活动、游戏 、后期制作等业务，三个部门对应的运维负责人为张三、李四、王五。
设置标签
为了方便管理，该公司使用标签来分类管理对应的数据库资源，定义了下述标签键/值。

标签键	标签值
部门	电商、游戏、文娱
业务	营销活动、游戏 、后期制作
运维负责人	张三、李四、王五

将上述标签键/值绑定到 TDSQL-C MySQL 版上，资源与标签键/值的关系如下表所示：

instance-id	部门	业务	运维负责人
ceshi-abc1	电商	营销活动	张三
ceshi-abc2	游戏	游戏	李四
ceshi-abc3	游戏	游戏	李四
ceshi-abc4	文娱	后期制作	王五
ceshi-abc5	文娱	后期制作	王五

相关操作

- 为集群创建标签 。
- 编辑和删除标签 。

版权所有：腾讯云计算（北京）有限责任公司

创建标签和绑定资源

最近更新时间：2024-05-13 15:05:51

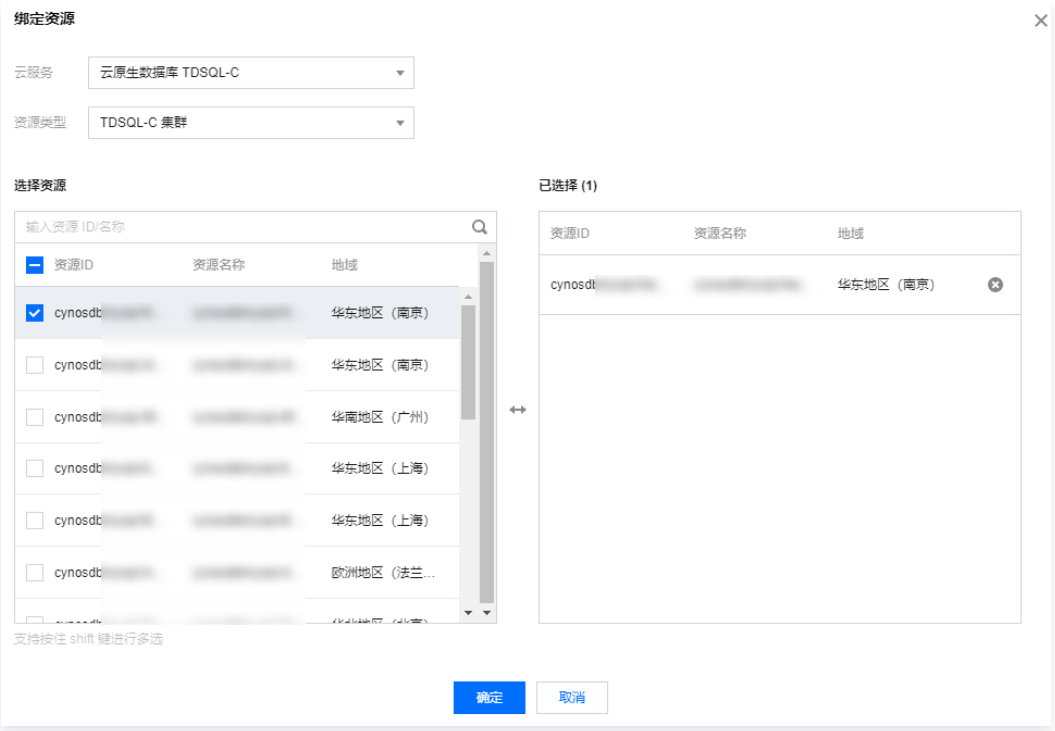
您可以通过创建标签来完成对 TDSQL-C MySQL 版资源的分类和统一管理，本文为您介绍如何创建标签及绑定资源。

创建标签

1. 登录 标签控制台 。
2. 在左侧导航栏中，单击**标签列表**，进入标签列表页面。
3. 单击**新建标签**，在新建标签对话框，新建标签或选择已有标签键为其新增标签值，单击添加标签键可一次性创建多个标签。
4. 单击**确定**完成创建。

绑定资源

1. 在标签列表页面找到新创建的标签键和标签值，单击**绑定资源**。
2. 在绑定资源页面，选择云服务为云原生数据库 TDSQL-C，资源类型为 TDSQL-C 集群，然后勾选需要绑定的资源。



3. 单击**确定**。

编辑和删除标签

最近更新时间：2024-05-13 09:12:12

您可以按照下列操作对集群进行标签的添加、更换、设置和解绑操作。

前提条件

已完成 创建标签和绑定资源 。

操作步骤

在集群列表页面，根据实际使用的视图模式进行操作：

页签视图

1. 登录 TDSQL-C MySQL 版控制台，在左侧集群列表，单击目标集群，进入集群管理页。
2. 在集群管理页，标签后单击 。



3. 在弹窗下，可添加、修改、设置、解绑标签，确认无误后单击**确定**。



列表视图

方法一

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在上方选择地域，单击集群 ID 或**操作**列的**管理**进入集群管理页。
3. 在**集群详情**页的**基本信息** > **标签**后单击 。
4. 在弹窗下，可添加、修改、设置、解绑标签，确认无误后单击**确定**。



方法二

1. 登录 TDSQL-C MySQL 版控制台 。
2. 在上方选择地域，在集群列表，选择目标集群的**操作**列的**更多** > **编辑标签**。



3. 在弹窗下，可添加、修改、设置、解绑标签，确认无误后单击**确定**。

> **说明：**
> 如需对多个集群编辑标签，可在集群列表，勾选需要编辑标签的集群，在列表上方选择**更多操作** > **编辑标签**进入标签编辑弹窗进行编辑。