云数据库 MySQL

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
- 使用限制
- 操作总览
- 维护管理实例
  - 查看实例健康状态
  - 设置实例维护时间
  - 为实例指定项目
  - 调整数据库实例规格
  - 修改实例名称
  - 修改端口号
  - 迁移可用区
  - 主备切换
  - 销毁实例
- 升级实例
  - 升级 MySQL 5.5 和 MySQL 5.6 的数据库版本
  - 升级 MySQL 5.7 到 MySQL 8.0
  - 升级内核小版本
  - 双节点升级三节点
  - 一键升级云盘版
- CPU 弹性扩容
  - 概览
  - 简介
  - 设置自动扩容
  - 设置自定义扩容
  - 开启事件告警
  - 查询扩容历史
  - 性能表现
  - 计费说明
  - 常见问题
- 只读/灾备实例
  - 创建只读实例
  - 管理只读实例 RO 组
  - 管理只读实例延迟复制
  - 管理灾备实例
- 数据库代理
  - 概览
  - 简介
    - 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
    - 开通数据库代理
    - 设置数据库代理连接地址
    - 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
    - 重新均衡负载
    - 关闭数据库代理

接入模式
实例故障转发
事务拆分功能
防闪断功能
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
连接池功能
连接池概述
开启和关闭连接池功能
自适应负载均衡功能
其他功能
Hint 语法使用
数据库管理（DMC）
DMC 简介
DMC 管理
账号管理
创建账号
设置密码复杂度
重置密码
修改账号权限
修改授权访问的主机地址
修改连接数
删除账号
修改备注
参数配置
设置实例参数
使用参数模板
对比不同模板参数
参数模板性能对比
参数配置建议
智能参数调优
系统参数模板变更记录
备份与回档
备份数据库
设置备份落冷
跨地域备份
备份加密
备份保留设置
设置备份下载规则
查看备份空间
设置本地 binlog 保留
使用备份文件恢复数据库
使用物理备份恢复数据库
使用逻辑备份恢复数据库
下载备份
回档数据库
删除备份
克隆实例
数据迁移
使用 DTS 服务迁移
导入 SQL 文件
离线迁移数据

网络与安全
访问管理
访问管理概述
授权策略语法
可授权的资源类型
控制台示例
管理云数据库安全组
切换网络
开启外网连接地址
通过 CLB 开启外网服务
开启透明数据加密
设置 SSL 加密
监控与告警
监控功能
告警策略（腾讯云可观测平台）
告警通知（DBbrain）
设置事件告警
操作日志
操作日志
日志投递
只读分析引擎
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
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
标签
标签概述
编辑标签

操作指南
使用限制

最近更新时间：2025-06-03 09:40:42

数据量限制

由于资源的有限性，为了隔离用户间性能影响，云数据库 MySQL 对各种类型的 MySQL 实例做了数据量限制。下面从技术角度介绍 MySQL 在大数据量下的单个实例以及单个表使用的影响：
大数据量实例：云数据库默认存储引擎是 InnoDB，当实例中的数据及索引页均能被 InnoDB 的 cache、buffer 所缓存住时，MySQL 实例能够支持很大的并发访问。如果实例数据量过大，会导致 cache、buffer 频繁的数据换出换进，MySQL 瓶颈很快转到 IO 上，访问吞吐量直趋下降（例如，某云数据库实例本可以支持8000次每秒访问，当数据量为 cache、buffer 大小的两倍时，仅能支持700次每秒左右的访问）。
大数据量表：单表数据量过大后，MySQL 对单表资源的管理成本（数据、索引等）变更，将直接影响表的处理效率。例如，当一个业务表（InnoDB）数据量达到了10GB后，更新操作延迟明显增大，直接影响了业务的响应时间，后来不得不通过迁移分表来缓解这个问题。

> **说明：**
> 单个实例的表数量超过30万后，可能会造成备份、监控、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30万。

连接数限制

MySQL 的连接数上限为 MySQL 的系统变量 max_connections，当 MySQL 实例连接数量超过 max_connections 时，新的连接将无法建立。
云数据库默认连接数可在 MySQL 控制台 单击**实例 ID**，进入**数据库管理** > **参数设置**页查看。用户可以根据需要自行调整 max_connections 的值。但连接数越多，消耗系统资源也越多。如果连接数超过实际系统的负载承受能力范围，必然影响系统服务质量。
关于 max_connections 请参见 MySQL 官方文档 。

连接云数据库的 MySQL 客户端的限制

建议使用 CVM 系统自带的 MySQL 客户端和 lib 库，连接云数据库实例。

关于慢查询的说明

- 对于使用 Linux 云服务器的开发者，可以通过云数据库导出工具获取慢查询日志，请参见 下载备份文件和日志 。
- 对于使用 Windows 云服务器的开发者，暂时不能直接获取慢查询日志。如果有需要，请 提交工单 联系我们获取慢查询日志文件。

云数据库的 binlog 保存时间说明

云数据库 MySQL binlog 日志文件可以保留7天 – 1830天，默认为7天（单击**实例 ID** 进入**备份恢复** > **自动备份设置**里可配置保留时间）。
binlog 保留时间过长或增长太快，都会导致备份空间变大，一旦备份空间超出系统的赠送空间，将会产生额外的备份空间费用。

字符集说明

> **说明：**
> 从官方 MySQL 8.0.29版本开始，原有 UTF8 编码格式变为 UTF8MB4。

云数据库 MySQL 5.7版本默认字符集编码格式是 UTF8。
云数据库 MySQL 8.0版本默认字符集编码格式分如下两种情形：
- 情形一：官方 MySQL 8.0.29以下版本实例的默认字符集编码格式为 UTF8。
- 情形二：官方 MySQL 8.0.29及以上版本实例的默认字符集编码格式为 UTF8MB4。详细变更说明请参见 MySQL 官网文档 。

虽然云数据库支持默认字符集编码的设置，但建议您在创建表时，显式的指定表的编码，并在连接建立时指定连接的编码。这样您的应用将会有更好的移植性。
关于 MySQL 字符集的相关资源请参见 MySQL 官方文档 。
您可以通过 SQL 语言或 MySQL 控制台修改字符集。

通过 SQL 语言修改字符集

1. 通过 SQL 语言执行如下语句可修改云数据库实例的默认字符集编码：

```
SET @@global.character_set_client = utf8;
```

```
SET @@global.character_set_results = utf8;
SET @@global.character_set_connection = utf8;
SET @@global.character_set_server = utf8;
```

执行语句后，其中 @@global.character_set_server 等10分钟左右将自动同步到本机文件进行持久化（另外3个变量不会同步到本机文件），迁移或重启将保持设置后的值。

2. 执行如下语句可修改当前连接的字符集编码：

```
SET @@session.character_set_client = utf8;
SET @@session.character_set_results = utf8;
SET @@session.character_set_connection = utf8;
```

或者

```
SET names utf8;
```

3. 对于 PHP 程序，可通过如下函数设置当前连接的字符集编码：

```
bool mysqli::set_charset(string charset);
```

或者

```
bool mysqli_set_charset(mysqli link, string charset);
```

4. 对于 Java 程序，可通过如下方式设置当前连接的字符集编码：

```
jdbc:mysql://localhost:3306/dbname?useUnicode=true&characterEncoding=UTF-8
```

通过 MySQL 控制台修改字符集

1. 登录 MySQL 控制台，在实例列表，单击**实例 ID**，进入实例详情页。
2. 在实例信息里找到字符集/排序规则，单击修改图标即可修改字符集。

3. 在弹出的对话框，选择字符集，单击**确定**。

操作限制

1. 请不要修改 MySQL 实例已有账号的信息和权限，这个操作可能会令部分集群服务失效。
2. 创建库和表时建议统一使用 InnoDB 引擎，这个选择能使实例在支持高访问的能力上有更好的表现。
3. 请不要修改、停止 master-slave 关系，这个操作可能会令热备失效。

数据库表名的限制

不支持以中文作为表名，请在创建表时注意。中文表名可能会导致回档、升级等流程失败。

数据库账户权限

云数据库 MySQL 不对用户提供实例 super user 权限，用户可以修改的参数以实际控制台为准，可以通过 MySQL 控制台 找到目标实例，单击**实例 ID**，进入**数据库管理** > **参数设置**页，查看可修改的参数范围。

网络选择

推荐您使用私有网络，在私有网络内，用户可以自由定义网段划分、IP 地址和路由策略。与基础网络相比，私有网络更适合有网络自定义配置需求的场景。私有网络和基础网络的对比请参见 管理网络 。

操作总览

最近更新时间：2025-06-05 15:34:12

本文介绍云数据库 MySQL 控制台的实例列表页和管理页内的相关功能操作，指导您轻松管理 MySQL 数据库。

实例列表页

在 云数据库 MySQL 控制台 的实例列表页可查看实例相关信息，以及管理实例。

功能	介绍
登录	在操作列单击**登录**，将使用新一代数据库工具登录数据库，操作请参见 DMC 简介 。
管理	在操作列单击**管理**，可进入实例管理页。
升级版本	• 双节点、三节点实例：在操作列单击**更多** > **升级版本**，可对数据库版本进行升级，操作请参见 升级数据库引擎版本 。 • 单节点、云盘版实例：暂不支持升级数据库引擎版本。
购买相同配置	勾选目标实例后，在操作列单击**更多** > **购买相同配置**，可快速在购买页创建和所选实例相同配置的新实例。
调整配置	在操作列单击**更多** > **调整配置**，可对数据库实例进行配置调整（扩缩容），支持实例升级与降级，操作请参见 调整数据库实例规格 。云盘版实例还可通过调整配置快速添加和删除只读节点，操作请参见 调整实例配置 。
按量转包年包月	在操作列单击**更多** > **按量转包年包月**，可修改按量计费实例的计费模式，操作请参见 按量转包年包月 。
编辑标签	在操作列单击**更多** > **编辑标签**，可对实例资源的标签进行编辑管理，操作请参见 编辑标签 。
销毁/退货	在操作列单击**更多** > **销毁/退货**，可对实例进行自助退还，操作请参见 销毁实例 。
安全组	在操作列单击**更多** > **安全组**，可为实例配置安全组或修改该实例的安全组，操作请参见 管理云数据库安全组 。
新建	在实例列表上方，选择**新建**，可跳转至购买页创建实例，操作请参见 创建 MySQL 实例 。
续费	勾选目标实例后，在实例列表上方，单击**续费**，可为实例进行手动续费，操作请参见 续费说明 。
重启	勾选目标实例后，在实例列表上方，单击**重启**对实例进行重启操作，支持批量重启（勾选多个实例）。 • 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。
回档	勾选目标实例后，在实例列表上方，选择**更多操作** > **回档**，借助备份可以将数据库回档至某个指定的时间，操作请参见 回档数据库 。
参数设置	勾选目标实例后，在实例列表上方，选择**更多操作** > **参数设置**，可修改该实例的参数值，支持批量设置（勾选多个实例），操作请参见 设置实例参数 。
设置为自动续费	勾选目标实例后，在实例列表上方，选择**更多操作** > **设置为自动续费**，可将非自动续费的实例设置为每月到期后自动续费，支持批量设置（勾选多个实例），操作请参见 续费说明 。
取消自动续费	勾选目标实例后，在实例列表上方，选择**更多操作** > **取消自动续费**，可将已设置为自动续费的实例设取消自动续费，支持批量取消（勾选多个实例）。
一键升级	勾选需要的目标实例后，在实例列表上方，选择**一键升级**，可将云数据库 MySQL 双节点/三节点实例迁移至 TDSQL-C MySQL 版，操作请参见 执行一键升级 。

一键诊断	在实例列表上方，选择**一键诊断**，可跳转至 数据库智能管家 DBbrain 控制台，可对账号下实例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 MySQL 诊断优化 。
对比监控	在实例列表上方，勾选2个 – 6个目标实例后，选择**对比监控**，可在控制台右侧弹窗下查看和对比多个实例监控情况。

实例管理页

登录 云数据库 MySQL 控制台，在实例列表，单击**实例 ID** 或**操作**列的**管理**，可以进入实例管理页面。您可以切换查看实例详情、实例监控、数据库管理、安全组、备份恢复、操作日志、只读实例、数据库代理、数据安全、连接检查。

功能页	功能项	介绍
实例详情	查看实例健康状态	帮助用户及时发现实例的异常情况，优化系统的性能和安全配置，操作请参见 查看实例健康状态 。
	设置实例维护时间	系统会不定期在维护时间内对实例进行维护，操作请参见 设置实例维护时间 。
	为实例指定项目	将实例分配至不同的项目进行管理，操作请参见 为实例指定项目 。
	调整数据库实例规格	通过控制台快捷调整实例规格，灵活的扩缩容。 • 单节点、双节点、三节点实例操作请参见 调整数据库实例规格 。 • 云盘版实例操作请参见 调整实例配置 。
	修改端口号	支持修改实例内网地址的端口号。 • 单节点、双节点、三节点实例操作请参见 修改端口号 。 • 云盘版实例操作请参见 读写地址管理 和 只读地址管理 。
	迁移可用区	将云数据库 MySQL 实例迁移至同一地域内的其它可用区，操作请参见 迁移可用区 。 双节点经济型实例不支持迁移可用区。
	主备切换	切换实例的主库和备库。 • 双节点、三节点实例操作请参见 主备切换 。 • 云盘版实例操作请参见 主备切换 。
	CPU 弹性扩容	自动或自定义扩展当前实例的 CPU 上限，操作请参见 CPU 弹性扩容 。仅通用型双节点、通用型三节点以及单节点本地盘（只读实例）架构的实例支持此功能。单节点（云盘）、双节点经济型以及云盘版架构的实例不支持此功能。
	管理灾备实例	双节点、三节点实例支持创建灾备实例，操作请参见 管理灾备实例 ，单节点（云盘）、双节点经济型、云盘版实例不支持创建灾备实例。
	切换网络	修改实例的内网网络。 • 单节点、双节点、三节点实例操作请参见 切换网络 。 • 云盘版实例操作请参见 读写地址管理 和 只读地址管理 。
	开启外网连接地址	操作请参见 开启外网连接地址 。
	修改数据复制方式	根据自身业务特点，设置实例的数据复制方式，以提高云数据库的可用性。操作请参见 修改数据复制方式 。
实例监控		您可以查看当前数据库运行的众多核心指标的监控，分为访问、负载、查询缓存、表、InnoDB、MyISAM 等维度的监控，详细介绍请参见 监控功能 和 告警功能 。云盘版实例支持在实例监控页面，切换节点来查看监控情

		况，操作请参见 查看实例监控 。
数据库管理	数据库列表	DMC 控制台可以新建库表、库管理、数据导入/导出、支持 SQL 窗口等，操作请参见 DMC 管理 。
	参数设置	通过控制台查看和修改部分参数，并可以在控制台查询参数修改记录。 • 设置实例参数 。 • 使用参数模板 。 • 对比不同模板参数 。 • 单节点、双节点、三节点实例4核及以上操作参见 智能参数调优 ，云盘版实例暂不支持智能参数调优。
	账号管理	可以对系统默认的 root 账号进行管理，也可以创建和管理新的账号。 • 创建账号 。 • 设置密码复杂度 。 • 重置密码 。 • 修改账号权限 。 • 修改授权访问的主机地址 。 • 删除账号 。
安全组		您可以对数据库进行安全组的配置，操作请参见 管理云数据库安全组 。
备份恢复	备份数据库	可以使用自动备份或手动备份的方式来备份数据库。操作请参见 备份数据库 。
	设置备份落冷	支持将备份文件进行落冷处理，以降低备份的存储成本。 • 双节点、三节点实例操作请参见 设置备份落冷 。 • 单节点、云盘版实例暂不支持设置备份落冷。
	跨地域备份	支持将备份文件放到另一个地域存储。 • 双节点、三节点、云盘版实例操作请参见 跨地域备份 。 • 单节点实例暂不支持跨地域备份。
	备份加密	支持对物理备份以及日志备份（binlog）进行加密。 • 双节点、三节点实例操作请参见 备份加密 。 • 单节点、云盘版实例暂不支持备份加密。
	设置备份下载规则	对备份的下载限制进行设置。 • 双节点、三节点实例操作请参见 设置备份下载规则 。 • 单节点、云盘版实例暂不支持设置备份下载规则。
	查看备份空间	查看实例备份空间及免费额度，操作请参见 查看备份空间 。
	设置本地 binlog 保留	设置实例的 binlog 保留周期，操作请参见 设置本地 binlog 保留 ，单节点不支持设置本地 binlog 保留。
	使用物理备份恢复数据库	双节点、三节点实例支持用 XtraBackup 工具，将 MySQL 物理备份文件恢复至其他主机上的自建数据库，操作请参见 使用物理备份恢复数据库 。
	使用逻辑备份恢复数据库	双节点、三节点实例支持使用逻辑备份文件进行手动还原数据，操作请参见 使用逻辑备份恢复数据库 。
	下载备份	双节点、三节点实例的逻辑备份和物理备份支持下载，操作请参见 下载备份 ，单节点、云盘版实例的快照备份不支持下载。
	回档数据库	数据损毁可自行通过回档修复，操作请参见 回档数据库 。
	删除备份	仅双节点、三节点实例的手动备份文件支持删除，操作请参见 删除备份 。
	克隆实例	可帮助用户快速回档实例至新购的 MySQL 实例，操作请参见 克隆实例 。
操作日志		• 云盘版、双节点、三节点实例可以查看慢日志、错误日志、回档日志、可进行日志投递，操作请参见 操作日志 和 日志投递 。 • 单节点实例可以查看慢日志、错误日志，操作请参见 操作日志 。
只读实例	创建只读实例	创建一个或多个只读实例，以支持用户的读写分离和一主多从应用场景。

		• 双节点、三节点实例操作请参见 创建只读实例 。 • 云盘版实例可以创建1 – 5个只读节点，操作请参见 增加实例节点 。 • 云盘版实例还可以创建1 – 15个独立只读实例，操作请参见 创建只读实例 。
	管理只读实例 RO 组	双节点、三节点实例支持管理只读实例 RO 组，操作请参见 管理只读实例 RO 组 。
	管理只读实例延迟复制	双节点、三节点实例支持管理只读实例延迟复制，操作请参见 管理只读实例延迟复制 。
数据库代理		用于代理应用服务端访问数据库时的请求，提供自动读写分离、连接池、连接保持等高级功能，单节点、双节点经济型实例不支持数据库代理，双节点、三节点、云盘版实例操作请参见 数据库代理概览 。
数据安全	开启透明数据加密	双节点、三节点实例支持开启透明数据加密，操作请参见 开启透明数据加密 。双节点经济型不支持开启透明数据加密。
	设置 SSL 加密	双节点、三节点实例支持设置 SSL 加密，操作请参见 设置 SSL 加密 。
连接检查		可以检测云数据库可能存在的连接访问问题，并根据提供的解决方法处理访问问题，以确保您的云数据库能够正常访问，操作请参见 一键连接检查工具 。

维护管理实例
查看实例健康状态

最近更新时间：2025-06-03 09:40:42

云数据库 MySQL 实例详情页增加了实例健康状态、异常告警以及实例在可用性、性能与安全方面的配置展示，帮助用户及时发现实例的异常情况，优化系统的性能和安全配置，提高实例的安全性和稳定性。本文介绍在实例详情页查看实例的健康状态及相关操作。

前提条件

已 创建云数据库 MySQL 实例 。

查看实例健康状态

1. 登录 MySQL 控制台 。
2. 在实例列表，单击**实例 ID** 或**操作**列的**管理**，进入实例详情页面。
3. 在详情页右侧，可查看实例的健康状态，包括实例健康状态评分、异常告警项、功能配置部署情况。

相关操作

页面	操作项
实例详情页	一键连接检查
	一键诊断
	SQL 限流
	查看告警
	查看配置

一、一键连接检查

如果用户无法通过内外网正常访问 MySQL 实例，一键连接检查工具可协助用户轻松排查内外网的连接问题，在内网地址后，单击**一键连接检查**，可跳转至连接检查页面，后续操作可参见 一键连接检查工具 。

二、一键诊断和 SQL 限流

在实例健康状态下，会显示实例当前的健康状态评分，健康状态评分为100分，用于快速判断实例是否健康运行，支持手动刷新。

单击评分后的**一键诊断**，可以跳转至数据库智能管家 DBbrain 控制台的异常诊断页，异常诊断能为用户提供实例实时的性能监控、健康巡检、故障诊断和优化，让用户既可以直观地感知数据库实例实时的运行状况，也可以定位实时出现的性能异常，并根据优化建议进行系统优化。

如何在异常诊断页查看监控信息、诊断信息等，请参见 异常诊断 。

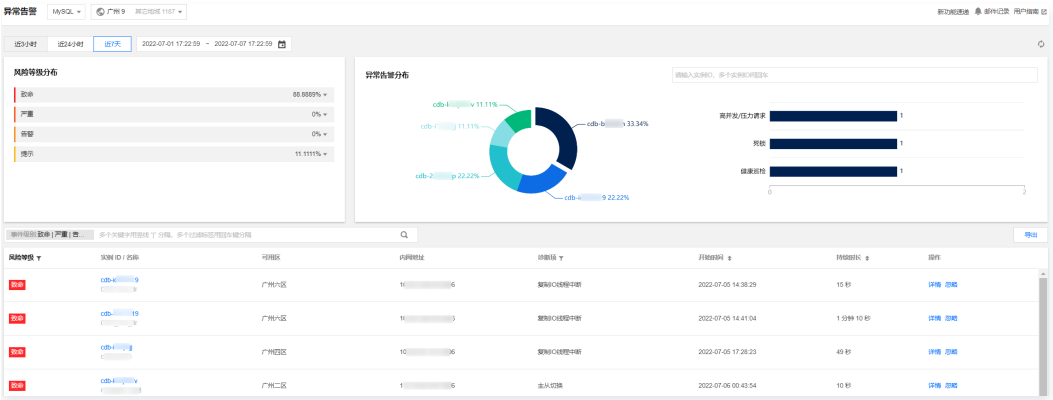
您可以通过创建 SQL 限流任务，自主设置 SQL 类型、最大并发数、限流时间、SQL 关键词，来限制数据库的请求访问量和 SQL 并发量，进而达到服务的可用性，不同的任务之间不会发生冲突，请参见 SQL 限流 。

三、查看告警

在实例详情页，异常告警下会显示实例近3小时内的异常告警项，用户可以根据异常告警的提示获取到实例某个阶段不同风险等级的风险项明细，便于用户针对性进行优化和调整，异常告警项由数据库智能管家 DBbrain 异常诊断功能统计。

单击**查看告警**，弹出右侧边框，可快捷显示当前实例不同级别的异常告警项。

然后单击右上角的**告警详情**，可跳转至数据库智能管家 DBbrain 控制台的异常告警页，可详细了解触发异常告警实例的基本信息、风险等级、诊断项以及持续时长等。详细了解异常告警页的信息以及操作，请参见 异常告警 。



四、查看配置

云数据库 MySQL 从可用性、性能、安全三个方面，为用户提供数据库功能的配置详情，用户可根据业务情况，选择对应类别，参考配置。

单击**查看配置**，弹出右侧边框，可显示数据库功能的配置详情，详细说明如下：

- 可用性（根据可用性级别展示配置项说明）

单点故障

配置项	描述	配置级别	操作
空间	预测剩余空间是否满足未来30天增长需求。	–	空间分析
防闪断	在数据库实例进行计划内有损切换、转移时，提供无损的应用连续性，避免连接中断。	增强型配置	• 防闪断功能 • 故障演练
多可用区部署	云数据库 MySQL 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	–	• 高可用性（多可用区） • 故障演练
多可用区部署（数据库代理）	云数据库 MySQL 数据库代理多可用区部署为代理提供多可用区故障转移支持。	–	开通数据库代理
三节点	三节点采用一主两备架构，支持强同步复制方式，提供金融级的可靠性和高可用性	增强型配置	三节点（原金融版）
多 RO 实例 RO 组	单个只读实例存在单点风险，建议至少单个 RO 组内存在2个只读实例，保证可用性。	强要求配置	创建只读实例
多可用区 RO 实例	单个 RO 组内 RO 实例仅能部署在一个可用区，可创建多个 RO 组实现跨可用区部署。	–	管理只读实例 RO 组

灾备实例	帮助用户以较低的成本提升跨地域连续服务的能力，同时提升数据的可靠性。	–	管理灾备实例
跨地域备份	将备份文件同步到另一个地域存储，帮助用户提升监管与容灾恢复能力。	–	跨地域备份

可用区级故障

配置项	描述	配置级别	操作
空间	预测剩余空间是否满足未来30天增长需求。		空间分析
防闪断	在数据库实例进行计划内有损切换、转移时，提供无损的应用连续性，避免连接中断。	增强型配置	• 防闪断功能 • 故障演练
多可用区部署	云数据库 MySQL 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。	强要求配置	• 高可用性（多可用区） • 故障演练
多可用区部署（数据库代理）	云数据库 MySQL 数据库代理多可用区部署为代理提供多可用区故障转移支持。	强要求配置	开通数据库代理
三节点	三节点采用一主两备架构，支持强同步复制方式，提供金融级的可靠性和高可用性。	增强型配置	三节点（原金融版）
多 RO 实例 RO 组	单个只读实例存在单点风险，建议至少单个 RO 组内存在2个只读实例，保证可用性。	强要求配置	创建只读实例
多可用区 RO 实例	单个 RO 组内 RO 实例仅能部署在一个可用区，可创建多个 RO 组实现跨可用区部署。	强要求配置	管理只读实例 RO 组
灾备实例	帮助用户以较低的成本提升跨地域连续服务的能力，同时提升数据的可靠性。	–	管理灾备实例
跨地域备份	将备份文件同步到另一个地域存储，帮助用户提升监管与容灾恢复能力。	–	跨地域备份

地域级故障

配置项	描述	配置级别	操作
空间	预测剩余空间是否满足未来30天增长需求。	–	空间分析
防闪断	在数据库实例进行计划内有损切换、转移时，提供无损的应用连续性，避免连接中断。	增强型配置	• 防闪断功能 • 故障演练
多可用区部署	云数据库 MySQL 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。	强要求配置	• 高可用性（多可用区） • 故障演练
多可用区部署（数据库代理）	云数据库 MySQL 数据库代理多可用区部署为代理提供多可用区故障转移支持。	强要求配置	开通数据库代理
三节点	三节点采用一主两备架构，支持强同步复制方式，提供金融级的可靠性和高可用性。	增强型配置	三节点（原金融版）

多 RO 实例 RO 组	单个只读实例存在单点风险，建议至少单个 RO 组内存在2个只读实例，保证可用性。	强要求配置	创建只读实例
多可用区 RO 实例	单个 RO 组内 RO 实例仅能部署在一个可用区，可创建多个 RO 组实现跨可用区部署。	强要求配置	管理只读实例 RO 组
灾备实例	帮助用户以较低的成本提升跨地域连续服务的能力，同时提升数据的可靠性。	强要求配置	管理灾备实例
跨地域备份	将备份文件同步到另一个地域存储，帮助用户提升监管与容灾恢复能力。	增强型配置	跨地域备份

全场景高可用性架构

配置项	描述	配置级别	操作
空间	预测剩余空间是否满足未来30天增长需求。	–	空间分析
防闪断	在数据库实例进行计划内有损切换、转移时，提供无损的应用连续性，避免连接中断。	强要求配置	• 防闪断功能 • 故障演练
多可用区部署	云数据库 MySQL 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。	强要求配置	高可用性（多可用区） 故障演练
多可用区部署（数据库代理）	云数据库 MySQL 数据库代理多可用区部署为代理提供多可用区故障转移支持。	强要求配置	开通数据库代理
三节点	三节点采用一主两备架构，支持强同步复制方式，提供金融级的可靠性和高可用性。	强要求配置	三节点（原金融版）
多 RO 实例 RO 组	单个只读实例存在单点风险，建议至少单个 RO 组内存在2个只读实例，保证可用性。	强要求配置	创建只读实例
多可用区 RO 实例	单个 RO 组内 RO 实例仅能部署在一个可用区，可创建多个 RO 组实现跨可用区部署。	强要求配置	管理只读实例 RO 组
灾备实例	帮助用户以较低的成本提升跨地域连续服务的能力，同时提升数据的可靠性。	强要求配置	管理灾备实例
跨地域备份	将备份文件同步到另一个地域存储，帮助用户提升监管与容灾恢复能力。	强要求配置	跨地域备份

- **性能**

配置项	描述	操作
并行查询	通过并行查询能力，调动多核计算资源，缩短大查询响应时间。	简介

异步删除大表	DROP TABLE 会将原数据库文件 (.ibd) 后台分批次清理，降低 DROP 操作带来的性能影响。	异步删除大表
秒级加列	秒级加列功能是通过只修改数据字典的方法来实现大表快速加列，减少对系统的影响。	秒级加列
秒级更改列	秒级更改列功能是通过只修改数据字典的方法来实现大表快速更改列，减少对系统的影响。	秒级更改列
只读实例	创建只读实例，以支持读写分离和一主多从应用场景，提高数据库的读负载能力。	创建只读实例
CPU 弹性扩容	开启自动 CPU 弹性扩容后，可有效保障线上业务稳定性。	CPU 弹性扩容
自动读写分离	通过数据库代理配置访问读写分离策略，实现读取能力的弹性扩展，分担数据库压力。	设置数据库代理读写属性

- **安全**

配置项	描述	操作
安全组	安全组是一种虚拟防火墙，控制实例级别的出入流量，是重要的网络安全隔离手段。	管理云数据库安全组
SSL	SSL 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。	设置 SSL 加密
密码复杂度	设置密码复杂度，提升数据库访问密码的强度，保障数据库的安全性。	设置密码复杂度
TDE 透明加密	TDE 的加解密操作对用户透明，实时进行加密和解密，满足静态数据加密的合规性要求。	开启透明数据加密
延迟只读	可通过设置延迟复制以及在延迟期间选择启动恢复到指定时间，来实现高效回档数据和故障快速回溯。	管理只读实例延迟复制
备份加密	开启备份加密后，新增物理备份文件会自动加密存储。	备份加密
备份下载限制	备份文件默认可以通过公网或内网进行下载，可以通过下载配置进行相应的限制。	设置备份下载规则
数据库审计	记录对数据库的访问及 SQL 语句执行情况，提高数据安全等级。	开通审计服务

版权所有：腾讯云计算（北京）有限责任公司

设置实例维护时间

最近更新时间：2024-10-18 14:14:22

操作场景

维护时间对于云数据库 MySQL 而言非常重要，为保证您的云数据库 MySQL 实例的稳定性，后台系统会不定期在维护时间内对实例进行维护操作。建议您对业务实例设置自己可接受的维护时间，一般设置在业务低峰期，将对业务的影响降到最低。
另外，建议实例规格调整、实例版本升级、实例内核升级等涉及数据搬迁的操作也放置在维护时间内（目前主实例、只读实例与灾备实例都支持设置维护时间）。以数据库实例规格升级为例，实例规格升级若涉及到数据搬迁，那么在升级完成时会发生秒级数据库连接闪断。在发起升级时选择**切换时间为维护时间内**，实例规格切换将会在实例升级完成后的下一个**维护时间**内发起。需要注意的是，选择切换时间为**维护时间内**时，数据库规格升级完成时不会立即切换，会保持同步直到实例的**维护时间**内发起切换，因此可能会延长整个实例升级所需时间。

> **说明：**
> - 云数据库 MySQL 在进行维护前，会向腾讯云账户内设置的联系人发送短信和邮件，请注意查收。
> - 云数据库 MySQL 会在数据库实例设置的维护时间周期内，进行数据一致性对比（仅做数据对比，不会有其他运维操作，不涉及 database 层面和 table 层面），以确保数据库主从数据的一致性，降低实例切换后数据异常的风险。数据一致性对比期间会有一定的性能损耗，请将维护时间周期设置在业务低峰期。如果当前数据库运行负载较高，在维护时间周期内不会发起数据一致性对比任务。
> - 实例切换时会发生秒级数据库连接闪断，请确保业务具备重连机制。

操作步骤

设置维护时间

1. 登录 MySQL 控制台，在实例列表，单击实例 ID 或**操作**列的**管理**，进入实例详情页面。
2. 在详情页的**维护窗口**下，单击任意编辑图标。

3. 在弹出的对话框，选择您所需的维护周期和维护时间，单击**确定**。

立即切换

若某任务选择在维护时间内切换，但因特殊情况需在未到维护时间内做切换，可单击**操作**列的**立即切换**。

说明：

- 立即切换适用于实例规格升级、版本升级、内核升级等涉及数据搬迁的操作。
- 版本升级操作下，若实例关联多个实例，切换操作会按照灾备实例、只读实例、主实例的顺序依次进行。

为实例指定项目

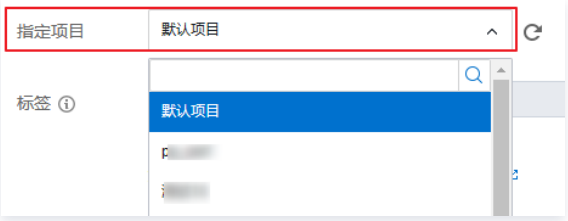
最近更新时间：2024-10-18 14:14:22

云数据库 MySQL 支持将实例分配至不同的项目进行管理，相关配置注意事项如下：

- 只读实例、灾备实例属于主实例的关联实例，需要与主实例处于同一项目中。
- 数据库实例在项目间进行分配和移动，不会影响实例对外提供的服务。
- 用户须在新购实例时为实例指定所属的项目，缺省为**默认项目**。
- 已指定项目的实例可通过**分配至项目**功能重新指定到其他项目。

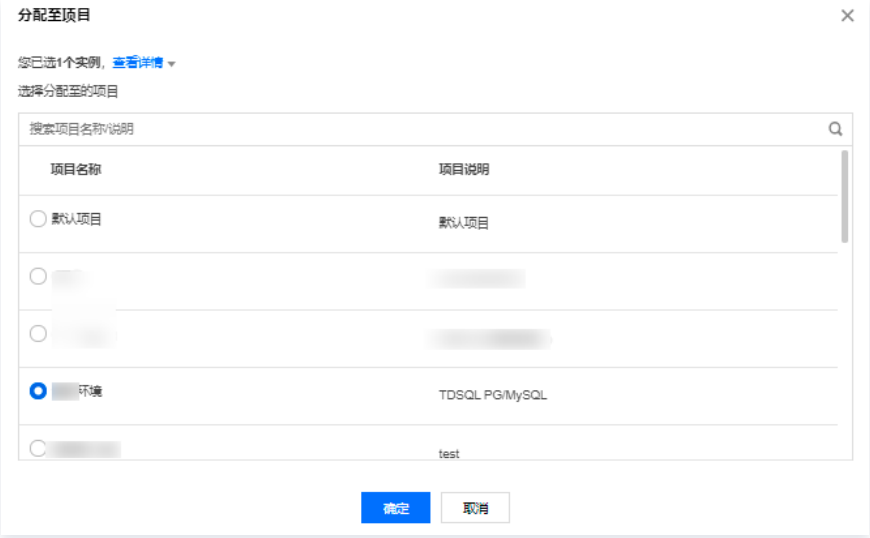
购买页分配项目

1. 登录 MySQL 购买页 。
2. 购买实例时，在指定项目后，可直接为新购实例分配项目。



控制台修改所属项目

1. 登录 MySQL 控制台 ，在实例列表，单击**实例 ID** 或**操作**列的**管理**，进入实例详情页面。
2. 在**实例信息** > **所属项目**后单击**转至其他项目**，在弹出对话框，选择对应项目后，单击**确定**。



> **说明：**
> 支持批量分配项目操作，多选实例后，在实例列表上方选择**更多操作** > **分配至项目**即可统一设置。

调整数据库实例规格

最近更新时间：2025-06-05 15:34:12

云数据库 MySQL 支持通过控制台快捷调整实例规格，提供灵活的扩缩容操作。您可根据业务所处的实际情况（业务初期、业务快速发展期、业务高峰期、业务低谷期等）灵活的调整 MySQL 实例的规格，从而更好地满足资源充分利用和成本实时优化等需求。
调整费用变更请参见 调整实例费用说明 。
云盘版实例在调整配置操作中，还可以增删实例的只读节点，变更节点可用区等，请参见 调整实例配置 。

实例硬盘超限说明

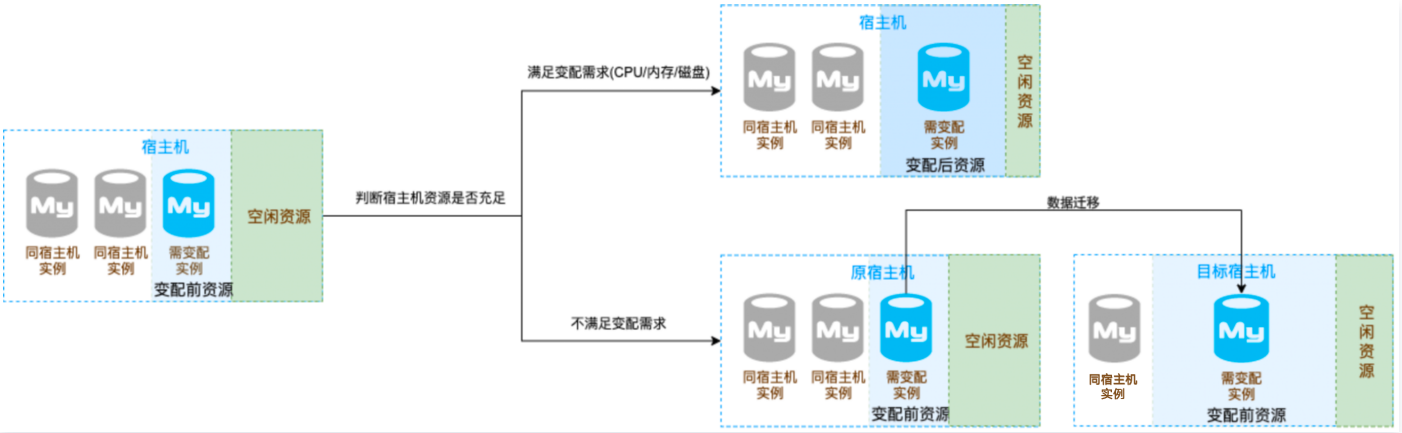
- 为保障您业务正常进行，当硬盘空间快要满时，请及时升级数据库实例规格或者购买硬盘空间。

> **说明：**
> 硬盘空间可在 MySQL 控制台 实例详情页查看，您也可通过 配置告警 及时接收硬盘告警消息。

- 实例存储数据量超过实例当前存储空间时，实例会被锁住，仅能读取数据不能写入，需扩容或在控制台删除部分数据库表解除只读，如果您的实例是单节点架构，则只能通过扩容来解除只读。
- 为避免数据库重复触发锁定状态，仅当实例剩余空间大于20%或大于50GB时，实例会解除锁定状态，恢复正常读写功能。

变配介绍

用户在控制台变更实例配置后，系统会通过数据迁移的方式来完成变更，默认情况下为普通变配，如果本地（即实例所在的宿主机）剩余资源充足，也可选择极速变配。变配流程如下：



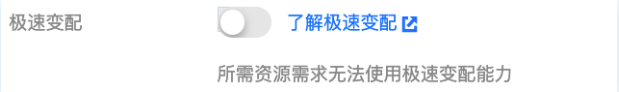
- **普通变配**：通过将当前宿主机的实例数据迁移至新宿主机的实例来完成变配，期间需要迁移数据、对比校验数据，数据量级较大时，变配整体耗时较长，且调整配置完毕时可能会涉及实例切换。
- **极速变配**：整体变配耗时较短，是否涉及主从切换以实际变配页面提示为准，目前分为如下几种情形。

磁盘类型	变配项	变配影响
本地盘	仅实例计算规格变更或者组合变配（规格升降级 + 磁盘扩缩容）	当本地资源充足，且涉及的参数变更中的参数不涉及重启时，当前变更不涉及主从切换和重启。变配页面会提示：“当前变更不会进行数据迁移和实例重启”。
		当本地资源充足，但涉及的参数变更中的参数涉及重启时，当前变更涉及主从切换和重启。变配页面会提示：“实例在调整配置期间实例访问不受影响；准备完成后会进行切换，会有秒级别的闪断，请确保业务具备重连机制”。
		当本地资源不满足变配需求时，当前变更涉及主从切换和重启。变配页面会提示：“实例在调整配置期间实例访问不受影响；准备完成后会进行切换，会有秒级别的闪断，请确保业务具备重连机制”。
	磁盘扩缩容	仅变更磁盘容量时，不涉及主从切换和重启。变配页面会提示：“当前变更不会进行数据迁移和实例重启”。

> **注意：**
> - 实例变配是否能触发极速变配与宿主机的资源情况密切相关，请您以实际调整配置页面中，极速变配开关是否被激活为准。

- 若极速变配开关被激活，则代表当前宿主机资源充足，可支持极速变配。

- 若极速变配开关未被激活，则宿主机资源不足，无法进行极速变配。



- 本地剩余资源充足，满足极速变配条件时，默认使用极速变配能力，如不需使用，则在调整配置页面进行关闭。（即在调整配置页面可选择开启或关闭“极速变配”按钮）。

注意事项

- RO 实例在开启独立 VIP 访问时，RO 实例不支持极速变配。
- RO 实例所在 RO 组存活实例数量小于等于延迟剔除最小实例数量时，RO 实例不支持极速变配。
- RO 实例所在 RO 组只有一个 RO 实例时，该 RO 实例不支持极速变配。
- 升级小版本时，是否涉及重启，请以变配页面的提示为准。

调整配置规则

- 云数据库 MySQL 实例及其所关联的只读实例和灾备实例处于正常状态下（运行中），并且当前没有任务执行时才能够发起调整配置操作。
- 调整配置过程中，不允许取消本次调整配置操作。
- 调整配置前后实例的名称、访问 IP、访问端口均不发生变化。
- 调整配置过程中，请尽量避免修改 MySQL 的全局参数、用户密码等操作。
- 调整配置过程中，可能会涉及到数据的搬迁，期间 MySQL 实例可正常访问，业务不受影响。
- 调整配置完毕时可能会涉及实例切换（即秒级 MySQL 数据库连接断开），建议程序有自动重连功能，并且选择实例可维护时间内做切换。请参见 设置实例维护时间 。
- MySQL 单节点 – 基础型，在调整配置过程中，存在大约15分钟实例不可用的情况，建议在业务低峰期操作。

实例规格和存储表

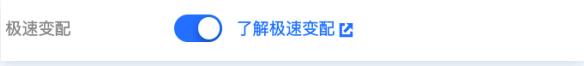
有关实例规格和存储表，请参见 数据库实例规格 。

> **说明：**
> 不同地域的实例规格对应的存储空间上限可能不同，请以实际购买页为准。

通过控制台调整实例配置

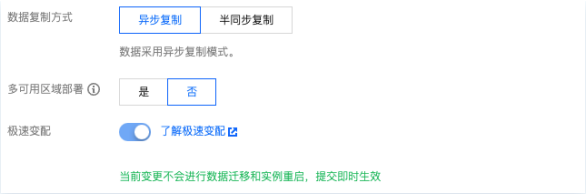
1. 登录 MySQL 控制台，在实例列表，选择需要调整的实例，在**操作**列选择**更多** > **调整配置**。
2. 在弹出的对话框，选择调整后的配置，单击**提交**。

> **说明：**
> - 本地剩余资源充足时，会触发极速变配能力，即本页面的**极速变配**按钮会生效，供用户选择开启或关闭。



> - 在您仅有硬盘（存储空间）调整场景时不涉及重启，此时开启极速变配系统会给出执行时间的选择。

> - 部分极速变配场景不涉及重启，此时提交的变配会在调整配置完毕时即时生效。如下图：



- 主实例触发迁移变配时，会显示数据校验敏感度设置项，可根据实际情况，以高、标准、低的方式进行数据校验速度的设置。
  - 高：当数据库负载压力升高时暂停数据一致性对比任务，避免数据一致性对比任务带来额外性能影响，会延长数据一致性对比任务时长。
  - 标准：当数据库负载压力较大时暂停数据一致性对比任务，避免数据一致性对比任务带来较明显性能影响。
  - 低：当数据库负载压力较大时依然执行数据一致性对比任务，缩短数据一致性对比任务时长。
- 若开启了 CPU 弹性扩容功能 的实例要进行调整配置操作，则需注意，调整配置完成后，CPU 弹性扩容功能将会失效，请及时观察调整配置后的 CPU 是否满足业务需求，如需再次使用 CPU 弹性扩容功能，请重新开启。

通过 API 调整实例配置

用户可以使用 UpgradeDBInstance 接口调整实例配置，请参见 调整云数据库实例的配置 。

热点问题

变配期间，是否会出现主备延迟问题？

主实例变配过程中会有数据对比，有可能产生主备延迟。

调整数据库实例规格，对实例有什么影响吗？

- MySQL 在调整配置过程中，可能进行数据迁移，期间实例访问不受影响；迁移完成后会进行切换，会有秒级别闪断，请确保业务具备重连机制。
- MySQL 单节点 – 基础型，在调整配置过程中，存在大约15分钟实例不可用的情况，建议在业务低峰期操作。

为什么我的数据库无法降配？

可能是您的存储已经超过了硬盘最大容量，需要先清理下数据，让剩余空闲空间大于20%或者大于50G，才能进行降配操作。

控制台调整配置后，为什么一直显示在“升级待切换”状态中？

有可能是您在 控制台 调整配置时，选择的**切换时间**为**维护时间内切换**，导致无法立即切换。
如您需立即切换，可在实例列表的**操作**列单击**立即切换**，切换会有秒级别闪断，请确保业务具备重连机制。

调整实例规格，升级需要多长时间？

数据库的升级时间和您当前数据库中的数据量的大小以及数据复制速度有关，具体时间无法进行预估。
升级过程中不会影响您当前实例的正常使用，在升级完成切换 VIP 时会有秒级的瞬断。

调整实例进度，在哪里可以查看？

您可在控制台的 任务列表 查看任务进度。

硬盘空间即将用尽，怎么办？

用户硬盘空间使用达到85%以上，建议考虑删除无用数据或者扩容硬盘空间，通过 控制台（实例列表右侧**操作**列选择**更多** > **调整配置**）可以进行硬盘空间扩容。

如何确认我的实例扩内存（扩容磁盘）支持极速变配？

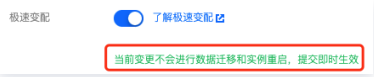
您可在变配页面，通过查看极速变配开关状态判断是否可以使用极速变配能力，不需要使用时，可以手动关闭。不满足极速变配需求时，无法打开该能力。

扩容内存或者磁盘是否会影响实例的小版本？

当实例小版本不是最新的情况下，会升级实例内核小版本至最新，升级小版本时极速变配能力会重启数据库。

极速变配是否会对实例重启？

在特定情况下，会重启实例，是否涉及重启，会在变配页面下方有相关提示，如下图：



> **说明：**
> 其中，在满足实例小版本是最新以及触发极速变配能力的情况下，仅变配磁盘不会重启。

在控制台升级时，如何判断是否为极速变配？

您可通过变配页面的**极速变配**按钮是否打开来判断。

API 接口变更配置时，如何判断是否为极速变配？

API 接口目前仅支持迁移方式进行变配升级，无法提供极速变配的指定，后续会提供 API 的能力。

数据库变配时，是否会调整数据库参数？

会调整 innodb_buffer_pool_size 参数以适应配置的变化。

数据库变配时，触发极速变配后，参数会调整吗？

与普通变配相同，参数会跟随实例规格的变化而调整。

极速变配对比普通变配，有什么区别？

极速变配减少了数据迁移的时间。

修改实例名称

最近更新时间：2025-02-25 16:21:52

实例名称主要用于对云数据库 MySQL 实例进行区分和管理，您可以通过控制台对实例名称进行修改。
本文为您介绍如何通过控制台修改实例名称。

实例 ID/名称说明

- 主实例、只读实例、灾备实例以及只读分析引擎均支持修改实例名称。
- 实例名称仅支持数字、英文大小写字母、中文以及特殊字符-_./()[]（）+=:：@且长度不能超过60。

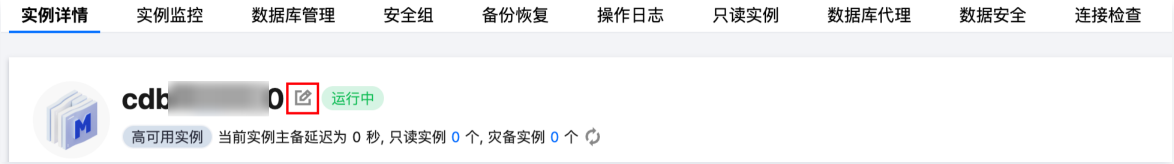
在实例列表中修改实例名称

1. 登录 MySQL 控制台 。
2. 在实例列表，找到目标实例，单击实例名称后的编辑图标。

3. 输入新的实例名称，单击**确定**。

在实例详情页修改实例名称

1. 登录 MySQL 控制台 。
2. 在实例列表，找到目标实例，单击**实例 ID** 或**操作**列的**管理**，进入实例管理页面。
3. 在实例管理页面里的实例详情页下，单击实例名称后的编辑图标。



4. 输入新的实例名称，单击**确定**。

相关 API

API	描述
修改云数据库实例名	接口（ModifyDBInstanceName）用于修改云数据库实例的名称。

修改端口号

最近更新时间：2025-04-30 10:00:43

云数据库 MySQL 支持修改实例内网地址的端口号，本文为您介绍通过控制台进行修改的步骤。

> **说明：**
> - 云数据库 MySQL 仅支持修改实例内网地址的端口号，外网地址端口号不支持修改。
> - 云盘版实例的只读节点默认开启内网地址，支持对读写节点、只读节点的网络和端口进行修改，操作请参见 读写地址管理 和 只读地址管理 。

前提条件

已 创建 MySQL 实例 。

> **说明：**
> 如果需要修改只读实例的内网地址端口号，需先确保只读实例已开启内网地址。
>
>
>

修改主实例/只读实例内网地址端口号

1. 登录 MySQL 控制台 ，在实例列表，单击**实例 ID** 或操作列的**管理**，进入实例详情页。
2. 在内网地址后单击编辑图标。

3. 在弹窗下，修改内网端口号，单击**确定**。

修改只读组的内网地址端口号

1. 登录 MySQL 控制台 。
2. 在实例列表找到需要修改只读组内网地址的主实例，单击其**实例 ID** 或操作列的**管理**，进入实例详情页。
3. 进入只读实例页，在 RO 组的内网地址后单击编辑图标。

4. 在弹窗下，修改内网端口号，单击**确定**。

迁移可用区

最近更新时间：2025-06-03 17:59:22

您可以将云数据库 MySQL 实例迁移至同一地域内的其它可用区。迁移可用区后，实例的所有属性、配置和连接地址都不会改变。迁移所需时间跟实例的数据量有关。
例如在如下场景中，您可以选择迁移可用区：

- 假设您正在尝试修改实例的实例类型，但我们无法在当前可用区中启动新实例类型的实例。在这种情况下，您可以将实例迁移到能够启动该实例类型的可用区。
- 当前可用区已无资源进行扩容的情况下，您也可以将实例迁移至同地域内其他资源充足的可用区，以满足业务需要。

前提条件

- 实例状态为运行中，且实例为双节点、三节点或云盘版架构，其中双节点经济型不支持迁移可用区。
- 实例所在的地域需要有多个可用区，才支持迁移可用区功能。

费用说明

本功能免费。即使将实例从单可用区迁移至多个可用区，也不收取费用。

功能说明

- 迁移可用区不会造成虚拟 IP（VIP）的变更。
- 迁移可用区主实例与 RO 没有解耦，迁移后的主实例，依然能够同跨地域 RO 进行同步。
- RO 实例可以选择可用区。
- 如果目标实例在 DTS 过程中，在云平台有任务锁，无法进行跨区迁移。
- 如果有正在执行的 DTS 任务，可用区迁移后，需要重启相应的 DTS 任务。
- 如果主实例在导出 dumper 过程中经历跨区迁移切换，DTS 导出会失败。

使用影响

切换时实例可用区会受到短暂影响，请确保应用具有自动重连机制。

使用限制

- 迁移可用区时系统会检查实例磁盘是否超用，若磁盘超用则无法进行可用区迁移，建议扩容磁盘后重试，若磁盘空间已超过当前实例规格的最大存储上限，建议升级实例规格配置后重试，详细规格与磁盘上限和相关操作请参见 调整数据库实例规格 。
- RO 实例不支持跨地域迁移。
- 迁移可用区暂不支持开启数据库代理的实例，请关闭数据库代理后进行跨可用区迁移。
- 迁移切换时不能通过 RO 组访问（剔除）。
- 双节点、三节点、云盘版架构下迁移可用区，主备可用区的选择受地域及区域剩余资源限制，在控制台迁移时选择目标可用区，备可用区选项会自动更新。
- 与主实例在同一可用区的 RO 实例默认随着主实例迁移至目标可用区，与主实例不在同一可用区的 RO 实例暂不支持迁移可用区操作。

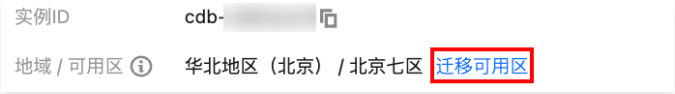
迁移类型

迁移类型	场景	支持类型
从一个可用区迁移至另一个可用区	实例所在可用区出现满负载或者其它影响实例性能的情况。	主实例、RO 实例、灾备实例
从一个可用区迁移至多个可用区	提高实例的容灾能力，实现跨机房容灾。主备实例分别位于不同的可用区。相对于单可用区实例，多可用区实例可以承受更高级别的灾难。例如，单可用区实例可以承受服务器和机架级别的故障，而多可用区实例可以承受机房级别的故障。	主实例、RO 实例、灾备实例
从多个可用区迁移至一个可用区	为了满足特定功能的要求。	主实例、RO 实例、灾备实例

操作步骤

1. 登录 MySQL 控制台 ，在实例列表，单击**实例 ID** 或操作列的**管理**，进入实例详情页面。

2. 在**实例详情页**的**地域/可用区**后单击**迁移可用区**，或在**可用性信息** > **部署方式**后单击**变更可用区**。



3. 在弹出的对话框，调整相关配置，确认无误后，单击**提交**。

- **目标可用区**：下拉框可进行主可用区变更，选择**多可用区部署**为**是**，可修改备可用区。
- **数据校验延迟阈值**：（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。该阈值设置较小时，有可能会导致迁移时间过长。

- **切换时间**：可选择维护时间内或迁移完成时进行切换，详细请参见 设置实例维护时间 。
- **同可用区 RO 实例迁移**：（主实例在同可用区有只读实例时此设置才会出现），选择同可用区 RO 是否跟随主实例迁移至目标可用区。

> **说明：**
> 实例在调整配置过程中，可能会进行数据迁移，期间实例访问不受影响；迁移完成后会进行切换，会有秒级别的闪断，请确保业务具备重连机制。

主备切换

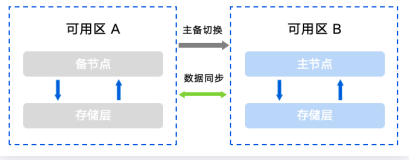
最近更新时间：2025-04-30 10:00:43

云数据库 MySQL 支持切换实例的主库和备库。在实例出现故障时，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，以上为实例遇到特殊情况例如故障时系统进行的自动切换，除此之外，您还可以通过控制台进行手动切换。

> **说明：**
> 云盘版实例支持只读节点切换为读写节点，详细操作说明请参见 读写节点切换 。

背景

在企业级应用中，数据库通常是关键的业务系统，一旦数据库出现故障或停机，将会对业务产生严重影响。因此，为了保证系统的可用性和数据的完整性，需要采用高可用性方案，如主备复制，主备切换是主备复制方案中的一种重要技术手段，可以在主数据库出现故障时快速切换到备用数据库，避免业务中断和数据丢失。



前提条件

- 实例为双节点、三节点架构。
- 实例状态为运行中，并无其他任务执行中。

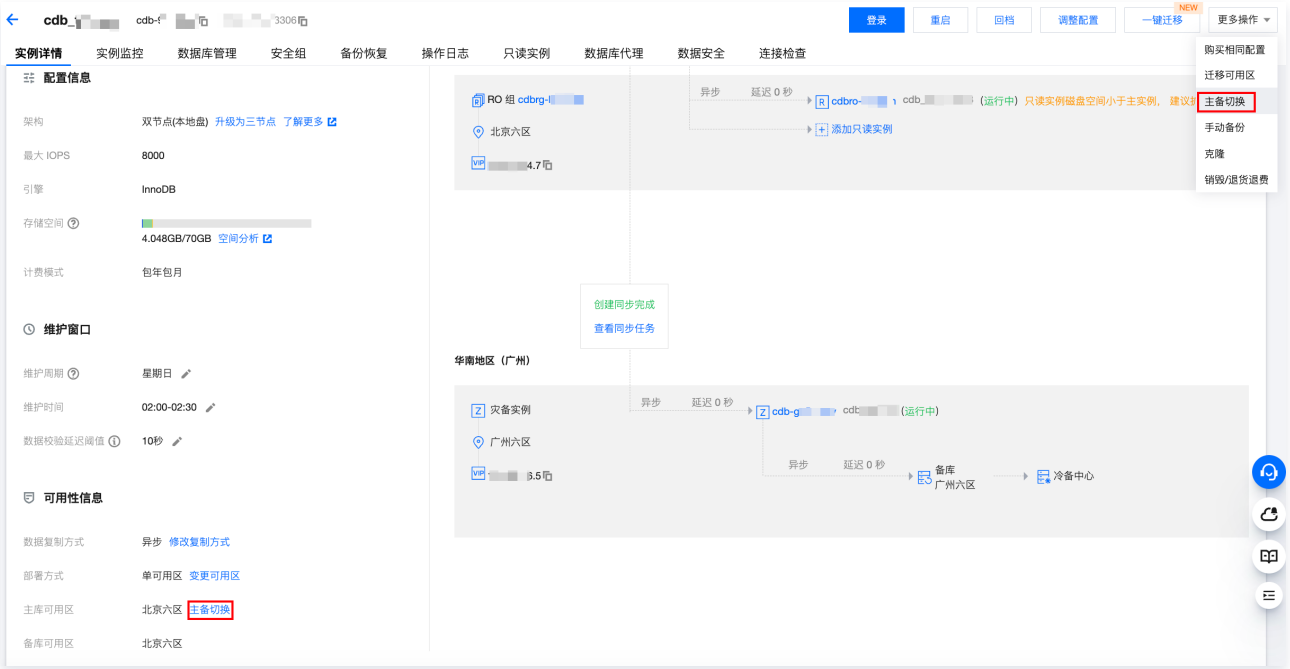
注意事项

- 主备库切换过程中，会有秒级别连接闪断。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。
- 主备库切换后，实例连接地址保持不变，原来的主实例会作为备实例，应用自动连接到新的主实例（即原备实例）。

操作步骤

> **说明：**
> 主备切换后不支持自动回切，但可根据实际情况进行手动回切（即主备切换后，再进行一次主备切换）。

1. 登录 MySQL 控制台，在实例列表，单击**实例 ID** 或操作列的**管理**，进入实例详情页面。
2. 在**实例详情页**的**可用性信息**下单击**主备切换**，或在**实例详情页**右上角，单击**更多操作** > **主备切换**。



3. 在弹窗下选择切换时间，勾选“**主备实例切换时，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**。
   - **维护时间内**：主备可用区数据同步完成后，会在下一个维护时间内进行主备切换，期间实例状态为“**主备待切换**”，且无法进行调整配置、销毁实例等操作。设置实例维护时间相关操作请参见 设置实例维护时间 。
   - **立即切换**：主备可用区数据同步完成后会立即切换。

热点问题

实例的主备库切换成功后，实例的地址是否变更？

切换后实例地址保持不变，原来的主实例会作为备实例，应用自动连接到新的主实例（即原备实例）。

手动切换后，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。

销毁实例

最近更新时间：2025-06-03 10:28:42

操作场景

根据业务需求，您可以在控制台自助退还按量计费和包年包月实例。

- 包年包月实例退还后，实例被移入云数据库回收站保留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。
- 按量计费实例退还后，实例被移入云数据库回收站保留24小时，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。

自助退还后，实例的状态一旦变为**隔离中**时，就不再产生与该实例相关的费用。

注意事项

- 实例销毁后数据将无法找回，备份文件会同步销毁，无法在云上进行数据恢复，请提前做好备份文件的转存，建议进行 备份保留设置 。
- 实例销毁后 IP 资源同时释放，如果该实例有相关的只读或灾备实例：
  - 只读实例将同时被销毁。
  - 灾备实例将会断开同步连接，自动升级为主实例。
- 实例销毁后，退款处理：
  - 5天无理由自助退还的金额将退还至腾讯云账户。
  - 普通自助退还的金额将按购买支付使用的现金和赠送金支付比例退还至您的腾讯云账户。
  - 推广奖励渠道订单退款将收取订单实际现金支付金额的25%作为退款手续费。 推广奖励渠道订单暂不支持自助退款，请 提交工单 发起退款申请。

操作步骤

1. 登录 MySQL 控制台 ，在实例列表，选择所需实例，在**操作**列选择**更多** > **销毁/退还**或**销毁实例**。
2. 在弹出的对话框，完成**是否保留备份文件**设置，单击**确定销毁**。

是否保留备份文件	说明
不保留	选择不保留，表示实例销毁后，备份会同步销毁，不进行保存。
保留	选择保留，表示实例销毁后，会根据选择的范围保存备份，选择范围的选项如下： • 保留全部：表示保留该实例在当前地域产生的全部备份，可选择是否保留 Binlog 文件。 • 保留最近一个：表示保留该实例在当前地域产生的最新的1个备份，默认不可选择保留 Binlog 文件。 • 保留时间段内的：表示保留该实例在当前地域产生的，并且从所选时间之后产生的备份，可选择是否保留 Binlog 文件。例如：选择保留2025年04月28日之后产生的备份。

3. 操作完成后，实例会进入回收站，实例状态为“隔离中”。

> **说明：**
> 若选择保留备份文件，当实例从回收站下线后，保留的备份文件可在 数据库备份 > 备份列表 > 已销毁实例备份 中进行查询和下载。

概览
备份列表
下载配置
数据备份列表
日志备份列表
已销毁实例备份
计费
已销毁实例备份不占用赠送空间，会按照标准备份进行计费，请您及时查阅相关说明
实例下线时间
全部
昨天
近 7 天
近 30 天
选择时间
选择时间
实例 ID
请输入实例 ID
实例 ID / 名称
备份开始 / 结束时间
文件数
备份大小
实例下线时间
备份地域
操作
2025-04-29 12:16:33 ~ 2025-04-29 12:16:33
共 1 个
单击可显示下载
6.31 MB
2025-04-29 14:18:33
北京
删除

升级实例
升级 MySQL 5.5 和 MySQL 5.6 的数据库版本

最近更新时间：2025-05-15 14:29:32

操作场景

本文档指导您在需要升级云数据库 MySQL 引擎版本时，通过操作控制台进行版本升级。
云数据库 MySQL 支持以下版本的数据库引擎升级：

- MySQL5.5到 MySQL5.6
- MySQL5.6到 MySQL5.7
- MySQL5.7到 MySQL8.0

本文为您介绍通过控制台将实例数据库版本从 MySQL5.5版本升级至 MySQL5.6版本，以及 MySQL5.6版本升级至 MySQL5.7版本。

> **说明：**
> - 不支持降级数据库引擎版本。
> - 不支持跨主要版本升级，例如将 MySQL5.5实例升级到 MySQL5.7，您必须先将实例升级到 MySQL5.6。
> - 单节点云盘版的实例不支持升级引擎版本。

版本升级规则

- 不支持 `create table … as select …` 语法。
- 云数据库 MySQL5.6、5.7主从同步基于 GTID 实现，默认仅支持 InnoDB 引擎。
- 从 MySQL5.5版本升级到 MySQL5.6，若最近一次全量备份为逻辑冷备，升级过程中会将 MyISAM 引擎的表转换为 InnoDB，**建议您升级前先完成 MyISAM 到 InnoDB 的转换**。
- 升级期间，云数据库 MySQL 会清空 slow_log 表。如果要保留日志信息，请在升级主要版本之前保存日志内容。
- **若待版本升级的实例有关联其他实例（主实例、只读实例等），为保证数据复制一致性，会连同一起进行版本升级**。
- 云数据库 MySQL 版本升级会涉及到数据搬迁，通常需要较长时间，请您耐心等待，期间您的业务不受影响，可以访问。
- 版本升级完毕时会涉及实例切换（即秒级 MySQL 数据库连接断开），建议程序有自动重连功能，并且建议选择实例可维护时间内做切换。有关维护时间请参考 设置实例维护时间 。
- 单个实例的表数量超过30万后，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30万。

操作步骤

1. 登录 MySQL 控制台 ，在实例列表，选择需要升级的实例，在**操作**列选择**更多** > **升级版本**。

   > **说明：**
   > MySQL8.0无法升级到更高版本。

2. 在弹出的对话框，选择需要的数据库版本，单击**升级**。
   数据库版本升级涉及到数据搬迁，所以在升级完成时会发生秒级的 MySQL 数据库连接闪断。可在发起升级时选择切换时间为**维护时间内**，会在实例升级完成后的下一个**维护时间内**发起切换。

   > **注意：**
   > 选择切换时间为**维护时间内**时，数据库规格升级完成时不会立即切换，会保持同步直到实例的**维护时间**内发起切换，因此可能会延长整个实例升级所需时间。

热点问题

升级前，云数据库 MySQL 是否会自动进行数据备份？

云数据库每天实时双机热备，基于数据备份 + 日志备份（binlog）可支持7日内 – 1830日内无损恢复。

云数据库可以从 MySQL5.7降级到 MySQL5.6吗？

不支持降级。如您需要 MySQL5.6版本，可以先销毁/退货 MySQL5.7实例，再重新购买 MySQL5.6实例，给您造成的不便，敬请谅解。

升级期间，是否会出现主备延迟问题？

主实例升级会有数据对比，有可能产生主备延迟。

版本升级完毕时会涉及实例切换，会影响云数据库 MySQL 实例的正常使用吗？

实例的正常升级流程不会影响您的使用，但实例连接会有秒级闪断，建议程序有自动重连功能，并选择实例可维护时间内做切换。

云数据库 MySQL 版本升级需要多长时间？在哪里可以查看升级进度？

数据库的升级时间和您当前数据库中的数据量的大小以及数据复制等有关，因此无法给出具体时间。
云数据库 MySQL 版本升级会涉及到数据搬迁，通常需要较长时间，请您耐心等待，期间您的业务不受影响，可以访问。

为什么实例一直是“升级待切换”的状态？

有可能是您切换时间选择的是**维护时间内切换**，实例在升级完成后，将在维护时间内进行切换操作。
如您需立即切换，可在实例列表的**操作**列单击**立即切换**，切换会有秒级别闪断，请确保业务具备重连机制。

升级 MySQL 5.7 到 MySQL 8.0

最近更新时间：2025-06-03 17:59:22

云数据库 MySQL 支持通过控制台直接升级数据库版本，本文为您介绍通过控制台将实例数据库版本从 MySQL 5.7 升级至 MySQL 8.0。

操作场景

MySQL 8.0在性能和功能上有较大的提升和增强，在高并发、大数据量场景下的性能表现更为突出，能够提供更强大的系统性能和稳定性。
云数据库 MySQL 支持以下版本的数据库引擎升级：

- MySQL 5.5到 MySQL 5.6
- MySQL 5.6到 MySQL 5.7
- MySQL 5.7到 MySQL 8.0

> **说明：**
> - 单节点云盘版的实例不支持升级引擎版本。
> - 升级过程中对现有业务的访问无影响。

MySQL 5.7升级8.0注意事项

功能限制

限制类别	限制详情
实例限制	仅支持双节点、三节点本地盘实例进行数据库版本 MySQL 5.7 升级 MySQL 8.0 操作。
	• 若主实例有多个只读实例，只读实例会跟随一起升级。 • 若主实例存在灾备实例，需要先释放灾备实例，升级完成后再重新创建灾备实例。
	升级前，请确保实例状态为运行中，且没有其他正在执行或待执行的任务，如存在，请等待任务结束后再发起数据库版本升级任务。
	升级前会对实例进行兼容性检查，检查结果有效时长为24小时，超过24小时后需要重新检查，检查期间如增加只读实例或灾备实例也需要重新执行检查任务。
升级限制	不支持跨大版本升级。例如 MySQL 5.6 无法直接升级至 MySQL 8.0，需要先升级到 MySQL 5.7，再升级到 MySQL 8.0。
	升级后不支持降级。 **说明：** 升级完成后，旧版本的备份集无法用于恢复新版本实例，如需进行克隆或回档操作，请选择实例升级后生成的备份集或时间点。旧版本的备份集支持下载后用于本地恢复操作。
数据库限制	数据库实例升级 MySQL 8.0 后，参数 sql_mode 中不支持的设置会重置为 MySQL 8.0 的默认值。
	若 RocksDB 存储引擎中使用了分区表，则不支持升级至 MySQL 8.0，需要修改为 InnoDB 引擎或删除分区表才能进行升级。
	数据库实例中若存在 MySQL 8.0 预留关键字的表、视图、存储过程、触发器等会导致升级失败。
	数据库中的存储过程、触发器、视图或函数中若包含 Changes in MySQL 8.0 ，则会导致升级失败。
	如果您使用了 MyISAM 或 Memory 引擎，则不支持升级，请先转换为 InnoDB 引擎。 **说明：** 如果您的数据库已是 InnoDB 引擎，但部分表仍是其他引擎，请使用 ALTER TABLE <表名> engine=InnoDB; 命令将表也转换为 InnoDB 引擎后，再进行升级。

升级前准备

- 请务必仔细对比升级前后版本的差异，建议新建一个 MySQL 8.0 版本实例进行语法测试，避免应用使用的低版本语法或特性在升级高版本后不支持。具体版本差异请参见 MySQL 8.0和 MySQL 5.7版本功能差异列表 。
- 建议您克隆原实例，先使用克隆实例进行升级测试，确认各项功能正常后再升级原实例。
- 升级前请检查最近一周是否有全量数据备份的成功记录，必要时请进行一次全量数据备份。
- 在数据库升级过程中，云数据库 MySQL 服务会出现闪断，请您尽量在业务低峰期执行升级操作，或确保您的应用有自动重连机制。
- 升级前请确保预留充足的磁盘可用空间。
- 升级前请查看 保留关键字 ，您在创建自定义函数的过程中应避免使用这些保留关键字。
- 为确保 MySQL 在新版本中的稳定性和性能，完成数据库版本升级后，云数据库 MySQL 不再支持查看和修改在 MySQL 8.0 中被废弃的 MySQL 5.7 的参数，详情可参考 Server and Status Variables and Options Added，Deprecated，or Removed in MySQL 8.0 。在数据库版本升级前，建议您备份相关参数的修改记录，以便后续操作和审计。

操作步骤

1. 登录 MySQL 控制台 ，在实例列表，选择需要升级的实例，在操作列选择**更多** > **升级版本**。

> **说明：**
> 您也可以单击**实例 ID** 进入实例详情页，在数据库版本后单击**升级版本**。
>
> 实例配置　通用型-1核2000MB内存，100GB存储空间　调整配置
> 数据库版本　MySQL5.7 20240331　升级版本　升级内核小版本

2. 在数据库版本升级窗口，完成如下配置，阅读并**勾选**数据库版本升级规则，单击**升级检查**。

参数	说明
数据校验延迟阈值	输入数据同步过程中的数据校验延迟阈值，可输入为1–10的整数，**延迟阈值设置越小，可能导致升级完成时间越长**。
切换时间	• 维护时间内：选择维护时间内，会在实例升级完成后的下一个维护时间内发起切换，设置维护时间请参见 设置实例维护时间 。 • 升级完成时：选择升级完成时，会在实例数据搬迁完成时立即进行版本切换。
数据库版本	默认选中 MySQL 8.0。

default_collation_for_utf8mb4	指定字符集的默认排序规则，具体可选字符集排序规则请以实际页面选项为准。

3. 升级检查过程中，系统会拉起检查任务，检查待升级的实例下是否满足**无灾备实例**、**参数配置符合规范要求**，检查完成且符合升级条件后，单击**升级**。
   3.1 检查实例下是否有**灾备实例**，若主实例存在灾备实例，则检查不通过，需要先释放灾备实例，升级完成后再重新创建灾备实例。
   3.2 检查参数配置是否符合规范要求，由系统自动检查判定，如不符合规范要求，可根据检查框内的错误信息对应调整。

> **说明：**
> 升级检查任务执行成功后，请在24小时内执行升级，超时需重新检查。

4. 执行升级后，返回实例列表，在左侧导航栏单击**任务列表**，可以查看该实例数据库版本的升级任务进度和详情。
5. 在任务列表找到刚才数据库版本升级的任务，可查询任务执行进度，单击其操作列的**任务详情**，可显示更多升级相关的信息。
6. 当实例状态由**配置变更中**变为**运行中**时，即完成数据库版本的升级。

> **说明：**
> 若升级的切换时间选择**维护时间内**，则实例升级完成后还未到下一个维护时间时，实例状态会显示**待切换**，如您需立即切换，可在实例列表的**操作**列单击**立即切换**，切换会有秒级别闪断，请确保业务具备重连机制。

附录：MySQL 8.0 对比 MySQL 5.7 的优势

- MySQL 8.0引入了300+项新特性，包括不可见索引、降序索引、函数索引、公用表表达式、窗口函数、快速加列、隐式主键等，这些功能可以使开发更加灵活和高效。
- MySQL 8.0对性能进行了改进，引入了直方图、HASH JOIN、并行查询、JSON 列等性能增强的功能，提升了数据库的查询性能和响应时间，使数据库能够更快地处理大规模数据和复杂的查询。
- MySQL 8.0支持 caching_sha2_password 认证插件，提高了安全性。
- MySQL 8.0 Optimizer 优化器功能增强。
- MySQL 8.0复制性能增强。
- MySQL 8.0支持创建多值索引、派生条件下推优化。
- MySQL 8.0支持读取 MySQL 授权表。
- MySQL 8.0支持资源分配控制。

热点问题

升级前，云数据库 MySQL 是否会自动进行数据备份？

云数据库 MySQL 每天实时双机热备，基于数据备份 + 日志备份（binlog）可支持7日内-1830日内无损恢复。

云数据库可以从 MySQL 8.0 降级到 MySQL 5.7 吗？

不支持降级。如您需要 MySQL 5.7 版本，可以先销毁/退货 MySQL 8.0 实例，再重新购买 MySQL 5.7 实例，给您造成的不便，敬请谅解。

升级期间，是否会出现主备延迟问题？

主实例升级会有数据对比，有可能产生主备延迟。

版本升级完毕时会涉及实例切换，会影响云数据库 MySQL 实例的正常使用吗？

实例的正常升级流程不会影响您的使用，但实例连接会有秒级闪断，建议程序有自动重连功能，并选择实例可维护时间内做切换。

云数据库 MySQL 版本升级需要多长时间？在哪里可以查看升级进度？

数据库的升级时间和您当前数据库中的数据量的大小以及数据复制等有关，因此无法给出具体升级所需时间。
云数据库 MySQL 版本升级会涉及到数据搬迁，通常需要较长时间，请您耐心等待，期间您的业务不受影响，可以访问。
在任务列表可以查看任务执行进度和任务详情，您可关注任务执行进度了解升级情况。

为什么实例一直是“升级待切换”的状态？

有可能是您切换时间选择的是**维护时间内切换**，实例在升级完成后，将在维护时间内进行切换操作。
如您需立即切换，可在实例列表的**操作**列单击**立即切换**，切换会有秒级别闪断，请确保业务具备重连机制。

升级内核小版本

最近更新时间：2025-06-10 17:00:32

云数据库 MySQL 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，或者您当前实例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，云数据库 MySQL 会在您实例的维护时间内进行升级和修复。
> - 当云数据库 MySQL 发生触发实例迁移的操作（如升降配实例规格、扩缩容磁盘容量、数据库版本升级等）时，系统不会将您的实例升级至最新内核小版本，如有需要，您可手动进行升级。当主实例拥有 RO 实例时，系统会自动判断主从同步兼容性，主实例迁移时不会使用比 RO 实例更新的小版本。

升级规则

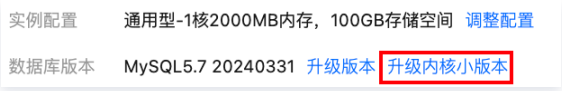
- 若待升级内核版本的实例有关联其他实例（主实例、只读实例），为保证数据复制一致性，关联实例会一起进行内核小版本升级。
- 云数据库 MySQL 版本升级会涉及到数据迁移，通常迁移速度与数据量大小有关系，请您耐心等待，期间您的业务不受影响，可正常访问。

注意事项

- 版本升级完成时会涉及实例切换（即秒级 MySQL 数据库连接断开），建议程序有自动重连功能，并且建议选择实例可维护时间内做切换，请参考 设置实例维护时间 。
- 单个实例的表数量超过30万后，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30万。
- 内核小版本升级后无法降级。

操作步骤

1. 登录 MySQL 控制台，在实例列表，找到目标实例，单击**实例 ID** 或**操作**列的**管理**，进入实例详情页。
2. 在实例详情页的**数据库版本**后，单击**升级内核小版本**。



3. 在弹出的对话框中，选择相关配置后，单击**升级**。
   数据校验延迟阈值：（主实例升级时此设置才会出现），阈值为1秒 – 10秒的整数。数据一致性校验过程中可能产生延迟，需设置数据延迟阈值，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。该阈值设置较小时，有可能会导致迁移时间过长。

   > **注意：**
   > 数据库内核小版本升级会涉及到数据迁移，所以在升级完成时会发生秒级的 MySQL 数据库连接闪断。建议您选择切换时间为**维护时间内**，实例会在升级完成后的下一个维护时间内发起切换。

升级内核小版本
当前版本
目标版本
内核小版本之间的差异，可参考文档
数据校验延迟阈值
10
秒
请输入 1 ~ 10 的整数（延迟阈值设置越小，可能导致升级完成时间越长）
切换时间
维护时间内
升级完成时
切换时间说明
维护时间03:00-04:00（实例详情页可修改维护时间）
在升级过程中，可能会进行数据迁移，期间实例访问不受影响；迁移完成后会进行切换，会有秒级别闪断，请确保业务具备重连机制
同时升级相关实例，查看详情
升级
取消

双节点升级三节点

最近更新时间：2025-05-16 17:30:02

操作场景

本文为您介绍通过 MySQL 控制台升级 双节点 实例至 三节点 实例的操作。

> **说明：**
> - 仅支持 MySQL 双节点实例升级至三节点实例。
> - 双节点经济型暂不支持升级三节点。
> - 升级过程不会影响实例的正常使用。
> - 您也可以直接通过 购买页 直接购买三节点实例。

操作步骤

1. 登录 MySQL 控制台，在实例列表，单击**实例 ID** 或**操作**列的**管理**，进入实例详情页。
2. 在实例详情页**配置信息**的**架构**处，单击**升级为三节点**。

3. 在弹出对话框，选择数据复制方式和可用区，确认无误后，单击**提交**。

- 数据复制方式：各复制方式介绍请参见 数据库实例复制 。
- 多可用区域部署：多可用区部署可保护数据库，以防数据库实例发生故障或可用区中断：
  - 三节点时，备库可以选择与主库相同或不同可用区，推荐部署方案：一个备库与主库同一可用区部署，另一个备库与主库跨可用区部署。
  - 目前仅部分主可用区支持选择不同备可用区，支持选择不同备可用区的主可用区可在 购买页 的主备可用区处查看。

> **说明：**
> 上图为包年包月实例升级三节点时的界面图，按量计费实例升级三节点时的界面如下。

4. 支付完成后，返回实例列表，实例状态为**配置变更中**，待状态变为**运行中**，即可正常使用。

热点问题

如何查看实例的架构？

您可登录 MySQL 控制台，在实例列表的**配置**处查看架构信息，也可单击**实例 ID** 或**操作**列的**管理**，进入实例详情页，在**配置信息**的**架构**处查看架构信息。

如何查看实例的主备库可用区？

您可登录 MySQL 控制台，在实例列表，单击**实例 ID** 或**操作**列的**管理**，进入实例详情页，在下方的可用性信息处查看可用区信息。

一键升级云盘版

最近更新时间：2025-05-15 14:29:32

本文为您介绍通过控制台一键升级功能，将云数据库 MySQL 双节点、三节点实例升级为云盘版实例。

前提条件

- 实例架构为云数据库 MySQL 双节点、三节点。
- 实例状态为运行中。

注意事项

- 升级至云盘版后不支持回退，请谨慎操作。
- 升级至云盘版后，原实例备份文件将不会保留，如需保留原实例备份文件，请提前下载进行保存。

操作步骤

1. 登录 MySQL 控制台 。
2. 在实例列表找到需要升级的实例，在其**计费模式**列单击**一键升级**，或勾选目标实例后单击实例列表上方的**一键升级**。

3. 在弹窗下选择**升级至云数据库 MySQL 云盘版**，单击**立即升级**。
4. 在弹出的界面右侧，单击**更改配置**。
5. 在弹窗下设置实例的相关配置信息后，单击**确定**。

参数	说明
可用区	选择读写节点和只读节点的可用区。
硬盘类型	默认为云盘。
实例规格	选择实例的类型、CPU、内存。
硬盘	选择硬盘类型和存储空间。存储空间范围：30 – 32000GB，数字以10为单位。存储空间至少为数据文件1.2倍且不能低于原实例磁盘规格。
数据保护空间	设置数据保护空间，范围1 – 10GB。为保障数据库实例的可恢复能力，云盘版实例需要设置一定的空间用于数据库实例的数据保护，该部分空间无法存放数据，属于系统保护空间，防止数据写满后导致实例异常。

6. 由于云盘版的告警指标升级，您需要 新建告警策略 ，操作可参考 告警策略 。
7. 更改完配置以及选择完告警策略后，在跳转的界面下方，单击**升级检查**。
8. 升级检查的检查项如下表所示，如有未通过的检查项，请根据提示进行调整后重新进入升级检查，升级检查通过后可进行下一步。

检查项	检查实例类型	说明
审计	主实例 只读实例	检查原实例是否开启数据库审计，若开启了，则不可迁移，您可在备份数据库审计日志后关闭数据库审计，再重新升级云盘版。关闭数据库审计请参见 关闭审计服务 。
备份	主实例	检查原实例是否开启定期保留策略、备份落冷、Binlog 落冷，若有开启，则不可迁移，您需要关闭后重试。相关介绍和操作请参见 备份数据库 和 设置备份落冷 。
备份加密	主实例	检查原实例是否开启备份加密，若开启了，则不可迁移，您需要关闭后重试。相关介绍和操作请参见 备份加密 。
灾备实例	主实例	检查原实例下是否有灾备实例，若存在，则不可迁移，您需要先释放灾备实例后再重试。释放实例操作请参见 销毁实例 。

弹性扩容	主实例 只读实例	检查原实例是否开启了 CPU 弹性扩容，若开启了，则不可迁移，您需要关闭后重试。相关介绍和操作请参见 CPU 弹性扩容 。
实例状态	主实例 只读实例	检查原实例运行状态是否正常，若有其他任务正在进行，请在其他任务结束后，待实例状态为运行中时再进行重试，其他异常状态，可根据处理建议进行调整。
单节点	主实例 只读实例	检查原实例架构，仅双节点、三节点实例支持一键升级为云盘版，若原实例为单节点架构，则不可迁移，请选择支持迁移的实例架构进行操作。
SSL	主实例 只读实例	检查原实例是否开启 SSL 加密，若开启了，则不可迁移，您可关闭 SSL 加密后重试，相关操作请参见 设置 SSL 加密 。
版本号检查	主实例 只读实例	检查原实例的内核小版本，需要满足如下情形才可迁移，升级内核小版本操作请参见 升级内核小版本 。 • MySQL5.7 20230601及以上版本。 • MySQL8.0 20230630及以上版本。 **说明：** 若原实例内核小版本为最新发布版本，但云盘版实例暂未支持该内核小版本时，不可迁移。
TDE 加密	主实例	检查原实例是否开启 TDE 加密，若开启了，则不可迁移，请更换未开启 TDE 加密的实例进行升级。

9. 选择升级切换时间，支持选择在维护时间内或者升级完成时进行切换，设置维护时间请参见 设置维护时间 。
10. 阅读并勾选提示：**实例在升级过程中，会进行数据迁移，期间实例访问不受影响；升级准备完成后会进行切换，会有秒级别的闪断，请确保业务具备重连机制。**
11. 单击**开始升级**，升级进度可在 任务列表 进行查询。
12. 若切换时间您选择的是升级完成时，则在数据一致性校验完成后原实例就会立即升级为云盘版，实例状态为运行中即完成升级。
13. 若切换时间您选择的是维护时间内，则此次升级会在您设置的实例维护时间内进行升级，若您希望立即完成升级，可以在实例列表，在目标实例的操作列单击**立即切换**，并在弹窗下阅读注意事项后，再次单击**切换**，等到实例状态更新为运行中即完成升级。

说明：
- 实例切换时，实例会切换至云盘版架构。
- 切换至云盘版架构后会生成新的只读 IP 地址。
- 切换至云盘版架构后，读写节点的 IP 地址将和升级前实例保持一致。

CPU 弹性扩容
概览

最近更新时间：2025-06-05 15:34:12

本章节为用户介绍云数据库 MySQL 的 CPU 弹性扩容相关功能。

章节概览

CPU 弹性扩容		说明	文档指引
功能简介		介绍 CPU 弹性扩容功能的背景、功能实现流程、说明事项，操作范围。	简介
操作项	自动扩容	介绍 CPU 弹性扩容功能扩容方式为自动扩容时的功能开启、修改和关闭的操作方法。	设置自动扩容
	自定义扩容	介绍 CPU 弹性扩容功能扩容方式为自定义扩容时的功能开启、修改、关闭的操作方法。	设置自定义扩容
	事件告警	介绍 CPU 弹性扩容功能相关的事件告警指标和设置事件告警的方法。	开启事件告警
	查询扩容历史	介绍如何查询实例的 CPU 扩容历史。	查询扩容历史
性能测试		通过性能测试工具 SysBench，对比开启 CPU 弹性扩容能力的性能表现。	性能表现
计费		介绍 CPU 弹性扩容功能的计费方式和计费单价。	计费说明
常见问题		陆续更新使用 CPU 弹性扩容功能遇到的常见问题及解决方法。	常见问题

版权所有：腾讯云计算（北京）有限责任公司

简介

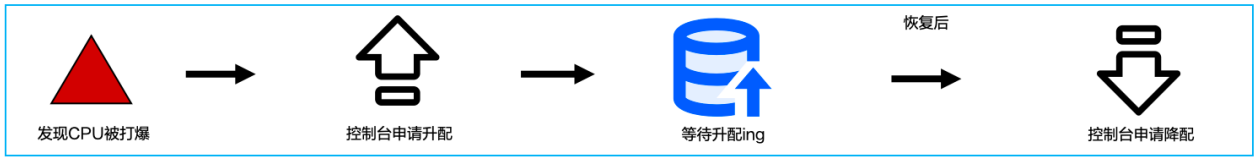
最近更新时间：2025-06-05 15:34:12

云数据库 MySQL 支持设置实例 CPU 的自动扩容以及自定义扩容，能更好的缓解突发请求带来的性能压力，适配业务高峰流量，保障线上业务稳定性，并且支持在流量下降时进行 CPU 回缩，避免资源浪费。本文介绍 CPU 弹性扩容的功能简介。

背景

在较多业务场景中，业务高峰期或突发流量洪峰会导致服务器负载面临极大的压力，应用程序的 CPU 使用率会急剧增加，如果不能及时扩容 CPU 资源，可能会导致系统性能下降、请求响应时间延长等问题，严重影响用户体验和满意度。除了突发流量的场景，当出现数据库缓存命中率低、索引失效、数据库死锁、数据库查询语句复杂、大量并发更新操作等场景且硬件配置不足时，也会造成 CPU 被打爆的情况。

- **传统解决方案**

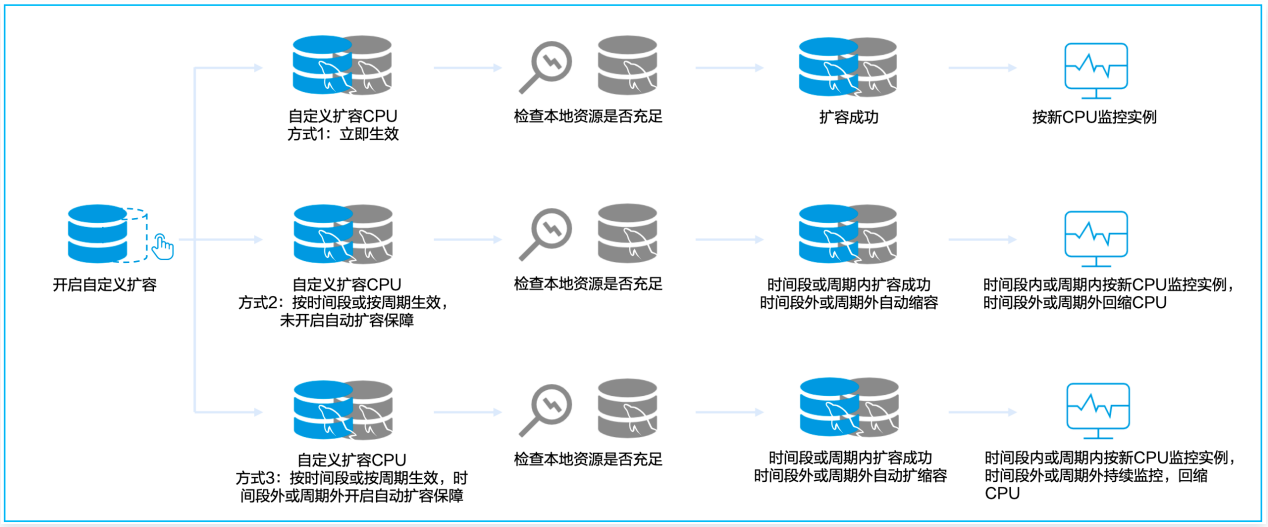


当发现 CPU 被打爆时，由用户通过控制台手动申请升配操作，然后等待升配完成，升配完成的时间受数据量影响，等待时长不确定，且升配过程中可能会出现闪断，会影响业务的正常运行，从升配完成到业务流量恢复平稳后，还需要人工通过控制台申请降配。综上，传统解决方案存在需要反应迅速、预期时间不定、有闪断风险、后续工作较多的不足。

- **CPU 弹性扩容**

场景1：日常开启，面对不确定/未知流量

场景2：有明确的活动时间，存在流量高峰



CPU 弹性扩容功能基于云环境优势，可实现动态分配 CPU 资源（当数据库访问量增加或 CPU 资源占用率上升时，可以自动添加更多的 CPU 资源，并在高峰期结束后自动缩减）。用户可在控制台上选择是否开启 CPU 弹性扩容功能，根据业务的需求和业务量动态地配置数据库的 CPU 资源，从而完成弹性扩展，应对高峰压力，确保数据库实例的高性能、高可用性和高稳定性。

功能类型

CPU 弹性扩容功能类型分为自动扩容和自定义扩容。其中，开启自动扩容时，需要对扩容规则和缩容规则进行设置；开启自定义扩容时，需要先选择扩容核数，再选择扩容时间。在选择扩容时间时，支持选择立即生效，则 CPU 会立即扩容，扩容后不会自动缩容，需要手动关闭，也支持选择按时间段或按周期进行扩容，则在所选时间段或周期内 CPU 会扩容，在所选时间段外或周期外会自动缩容，您也可以开启时间段外或周期外的自动扩容，这使得您可以更精细的设置实例 CPU 的弹性策略，更加符合实际业务波动。

自动扩容

- **自动扩容说明**

  开启 **CPU 弹性扩容** > **自动扩容**后，当观测窗口内数据库实例的 CPU 平均使用率达到设定的阈值时，在原始计算规格的基础上，系统会将 CPU 核数提升2倍。例如原始计算规格的 CPU 核数为4核，则会提升至8核，同时，实例的 IOPS 会同步上升，CPU 核数每提升1核，IOPS 增加1000。若用户的腾讯云账户余额不足或主机中没有足够的 CPU 资源（概率较低），则扩容不会进行，随即发送扩容失败事件。

  > **说明：**
  > 仅支持在原始计算规格的基础上将 CPU 核数提升2倍，无法在提升后的 CPU 核数上继续提升。例如提升后的 CPU 核数为8核，此时无法再提升至16核。

- **自动缩容说明**

  开启 **CPU 弹性扩容** > **自动扩容**后，在设定的缩容规则观测窗口期间内，CPU 使用率低于设定的阈值时，系统会自动将 CPU 核数以及 IOPS 回缩至原始计算规格。

  > **说明：**
  > 开启 **CPU 弹性扩容** > **自动扩容**功能后，系统会以最新的配置参数为准，对数据库实例进行监控，当数据库实例满足自动回缩条件时，系统会对数据库实例进行自动回缩。

  要了解开启自动扩容、关闭自动扩容、扩缩容规则设置的操作，请参见 设置自动扩容 。

自定义扩容

- **自定义扩容说明**

  开启 **CPU 弹性扩容** > **自定义扩容**时，若扩容时间选择为立即生效，则实例立即按照扩容后的 CPU 核数运行。若扩容时间选择为按时间段或按周期，则实例会在所选时间段内或周期内按照扩容后的 CPU 核数运行，在所选时间段外或周期外会自动缩容。若扩容时间选择为按时间段或按周期，同时开启了自动扩容保障，则在所选时间段外或周期外，实例会根据设置的扩容和缩容规则进行自动扩缩容。

  > **说明：**
  > CPU 弹性扩容功能中，**自动扩容保障即表示自动扩容**，开启自动扩容保障可以避免时间段或周期外的 CPU 突发流量。

- **手动关闭说明**

  在开启 **CPU 弹性扩容** > **自定义扩容**后，若扩容时间选择为立即生效，则当业务无需更多 CPU 资源时，用户需手动关闭 CPU 弹性扩容，否则收费会持续进行。
  要了解开启自定义扩容、关闭自定义扩容的操作，请参见 设置自定义扩容 。

自动扩容与自定义扩容核数说明

- 自动扩容的 CPU 扩容核数默认为实例原始 CPU 核数的2倍，不支持修改。例如实例原始 CPU 为4核，则扩容核数为4核，扩容后该实例的 CPU 核数为8。
- 自定义扩容的 CPU 扩容核数支持1核单位，最高支持为实例原始 CPU 核数的2倍。例如实例原始 CPU 为4核，则扩容核数支持设置为5、6、7、8核。
- 无论采用自动扩容还是自定义扩容，实例原始的 CPU 核数均不能超过32核。例如实例原始规格为48核488000MB，则不支持开启 CPU 弹性扩容功能。

计费说明

CPU 弹性扩容功能的详细计费说明，请参见 计费说明 。

事件告警说明

针对自动扩容和自定义扩容，支持设置相关的事件告警并配置告警通知。以下为 CPU 弹性扩容相关的事件指标及说明。

扩容类型	事件指标	说明
自动扩容	CPUExpansion	CPU 自动扩容成功
	CPUExpansionFailed	CPU 自动扩容失败
	CPUConstraction	CPU 自动回缩成功

自定义扩容	CPUExpansion	CPU 自定义扩容成功
	CPUExpansionFailed	CPU 自定义扩容失败
	CPUConstraction	CPU 自定义扩容关闭

自动扩容失败以及自定义扩容失败的原因有两种：用户的腾讯云账户余额不足或者主机 CPU 资源不足（极小概率）。若账户余额不足导致扩容失败，用户可为账户充值后重试，其他原因可 提交工单 反馈。
要了解事件告警的设置步骤，请参见 开启事件告警 。

功能影响说明

- 双节点、三节点实例 CPU 弹性扩容功能会同时在主节点和备节点上生效，即如果 CPU 弹性扩容功能生效后双节点、三节点实例发生了 HA 切换，那么切换后的节点也将为扩容后的 CPU 规格，自动缩容或关闭自定义扩容后，会自动回缩至原始 CPU 规格。
- 只读实例和灾备实例需单独开启功能，不同时扩容，即主实例 CPU 弹性扩容功能触发生效后，该主实例挂载的只读实例、灾备实例不会同时随之扩容 CPU，均需要单独开启 CPU 弹性扩容功能。
- 若开启了 CPU 弹性扩容功能的实例要进行 调整配置 操作，则需注意，调整配置完成后，CPU 弹性扩容功能将会关闭，请及时观察调整配置后的 CPU 是否满足业务需求，如需再次使用 CPU 弹性扩容功能，请重新开启。
- 若开启了 CPU 弹性扩容功能的实例进行迁移操作，包括用户主动发起迁移、流程中同步迁移（例如主实例升级迁移连带只读实例升级迁移场景），则需注意，迁移完成后，CPU 弹性扩容功能将会关闭，请及时观察调整配置后的 CPU 是否满足业务需求，如需再次使用 CPU 弹性扩容功能，请重新开启。

相关操作

- 设置自动扩容
- 设置自定义扩容
- 开启事件告警

设置自动扩容

最近更新时间：2025-06-05 15:34:12

本文介绍设置自动扩容的相关操作。

前提条件

- 实例架构为云数据库 MySQL 通用型双节点、通用型三节点或单节点本地盘（只读实例）。
- 使用 CPU 弹性扩容功能前实例的计算规格需小于等于32核。
- 腾讯云账户内需要有足够的余额支撑扩容。
- 底层机器可支持扩容。若底层机器已无法支持扩容，则设置自动扩容操作会失败。

计费说明

自动扩容计费相关说明请参见 计费说明 。

开启自动扩容

> **说明：**
> 自动扩容开启后，存在极低概率无法弹起，后台会多次重试，直至弹起。

1. 登录 MySQL 控制台，在实例列表，单击**实例 ID** 或操作列的**管理**，进入实例详情页。
2. 在**实例详情** > **弹性扩容**后，单击**开启**。

3. 在 **CPU 弹性扩容**窗口下，完成如下配置，确认扩容费用，单击**立即扩容**。

参数	说明
扩容方式	支持自动扩容和自定义扩容，这里选择自动扩容。 • 自动扩容：根据性能快速弹性适配高峰流量，在流量下降时进行规格回缩。 • 自定义扩容：自定义时间段进行扩容，并支持时间段外设置自动扩容保障，即自动扩容。
扩容核数	默认和当前实例规格的 CPU 核数一致，例如当前实例规格为6核24000MB，则扩容核数为6核，成功触发扩容后，实例 CPU 为12核。
增加 IOPS	默认每扩容1核，IOPS 增加1000。
扩容规则	设置触发扩容的观测周期和阈值。 • 观测周期：支持选项为15秒、30秒、45秒、1分钟、3分钟、5分钟、10分钟、15分钟、30分钟。

	• 阈值：支持选项为40%、50%、60%、70%、80%、90%。
缩容规则	设置触发缩容的观测周期和阈值。 • 观测周期：支持选项为5分钟。 • 阈值：支持选项为10%、20%、30%。

4. 成功开启自动扩容后，界面如下图所示。

修改 CPU 弹性扩容策略

开启自动扩容后，仅支持修改自动扩容能力下的扩容和缩容规则，您可在**实例详情** > **弹性扩容**后单击**修改**，进行重新配置。
开启自动扩容后，不支持通过**修改**操作直接调整扩容方式（由自动扩容改为自定义扩容），如需调整扩容方式，请先关闭自动扩容，然后在重新开启 CPU 弹性扩容时调整扩容方式。

关闭自动扩容

> **说明：**
> 关闭自动扩容后，已扩容的 CPU 会立即回缩到实例扩容前的 CPU 核数，回缩后不再根据扩容阈值进行扩容。

1. 登录 MySQL 控制台，在实例列表，单击**实例 ID** 或操作列的**管理**，进入实例详情页。
2. 在**实例详情** > **弹性扩容**后单击**关闭**。
3. 在**结束扩容 CPU** 的弹窗中单击**确定**。

热点问题

开启 CPU 弹性扩容的自动扩容能力后，如何判断自动扩容是否生效？

当实例的 CPU 平均利用率达到设置的自动弹性扩容阈值时，该实例会自动扩容。您可在实例详情页的实例配置后查询是否生效，若 CPU 核数后额外显示（实际核数），则表示生效。如下图所示，4（8）核表示当前已经是自动扩容到了8核。

相关 API

API	描述
开启 CPU 弹性扩容	接口（StartCpuExpand）用于开启实例的 CPU 弹性扩容功能。
查询实例的 CPU 弹性扩容策略信息	接口（DescribeCPUExpandStrategyInfo）用于查询实例的 CPU 弹性扩容信息。
查询扩容历史	接口（DescribeCpuExpandHistory）用于查询扩容历史。
关闭 CPU 弹性扩容	接口（StopCpuExpand）用于关闭实例的 CPU 弹性扩容功能。

设置自定义扩容

最近更新时间：2025-06-05 15:34:12

本文介绍设置自定义扩容的相关操作。

前提条件

- 实例架构为云数据库 MySQL 通用型双节点、通用型三节点或单节点本地盘（只读实例）。
- 使用 CPU 弹性扩容功能前实例的计算规格需小于等于32核。
- 腾讯云账户内需要有足够的余额支撑扩容。

计费说明

自动扩容计费相关说明请参见 计费说明 。

开启自定义扩容

> **说明：**
> 自定义扩容开启后，按时间段、按周期、开启自动扩容保障（即时间段外自动扩容）存在极低概率无法弹起，后台会多次重试，直至弹起。

1. 登录 MySQL 控制台 ，在实例列表，单击**实例 ID** 或操作列的**管理**，进入实例详情页。
2. 在**实例详情** > **弹性扩容**后，单击**开启**。

3. 在 **CPU 弹性扩容**窗口下，设置扩容方式、扩容核数、扩容时间，单击**立即扩容**。

扩容方式：支持自动扩容和自定义扩容，这里选择自定义扩容。
- 自动扩容：根据性能快速弹性适配高峰流量，在流量下降时进行规格回缩。
- 自定义扩容：自定义时间段进行扩容，并支持时间段外设置自动扩容保障，即自动扩容。

扩容核数：设置额外增加的 CPU 核数，支持1核单位扩容，最高扩容核数为当前规格 CPU 的1倍核数，如8核16G最大扩容核数为8。
增加 IOPS：CPU 核数每提升1核，IOPS 同步增加1000。
扩容时间：支持**立即生效**、**按时间段**、**按周期**三种选择，请根据实际需求选择其中一种进行设置。

立即生效

扩容时间：选择立即生效，表示功能开启成功后立即触发扩容，扩容后不能自动回缩，需手动关闭扩容，关闭后可立即回缩至原始 CPU 核数。

按时间段

扩容时间：选择按时间段，表示功能开启成功后按照所选时间生效触发扩容，若未开启自动扩容保障，则在所选时间段外自动缩容。

- **开始时间**：默认为开启功能的当天，可自定义选择开始日期和时间，时间粒度为30分钟。
- **结束时间**：自定义选择结束日期和时间，时间粒度为30分钟。若实例未开启自动续费，则最长可选结束时间为实例的到期时间。

自动扩容保障：默认关闭，可根据需要选择是否开启自动扩容保障。开启后，在扩容所选时间段外，可通过扩缩容规则实现自动扩缩容。

- **扩容核数**：默认和当前实例规格的 CPU 核数一致。
- **增加 IOPS**：默认每扩容1核，IOPS 增加1000。
- **扩容规则**：设置触发扩容的观测周期和阈值。
  - **观测周期**：支持选项为15秒、30秒、45秒、1分钟、3分钟、5分钟、10分钟、15分钟、30分钟。
  - **阈值**：支持选项为40%、50%、60%、70%、80%、90%。
- **缩容规则**：设置触发缩容的观测周期和阈值。
  - **观测周期**：支持选项为5分钟。
  - **阈值**：支持选项为10%、20%、30%。

按周期

扩容时间：选择按周期，表示功能开启成功后按照所选周期生效触发扩容，若未开启自动扩容保障，则在所选周期外自动缩容。

- **周期选择**：默认勾选周一至周日，可自定义选择周期，至少需选择一天。
- **起止时间**：默认为08:00 – 00:00（次日），可自定义选择起止时间，时间粒度为30分钟。

自动扩容保障：默认关闭，可根据需要选择是否开启自动扩容保障。开启后，在扩容所选周期外，可通过扩缩容规则实现自动扩缩容。

- **扩容核数**：默认和当前实例规格的 CPU 核数一致。
- **增加 IOPS**：默认每扩容1核，IOPS 增加1000。
- **扩容规则**：设置触发扩容的观测周期和阈值。
  - **观测周期**：支持选项为15秒、30秒、45秒、1分钟、3分钟、5分钟、10分钟、15分钟、30分钟。
  - **阈值**：支持选项为40%、50%、60%、70%、80%、90%。

- **缩容规则**：设置触发缩容的观测周期和阈值。
  - **观测周期**：支持选项为5分钟。
  - **阈值**：支持选项为10%、20%、30%。

4. 当实例状态由“配置弹性扩容策略中”变为“运行中”，即开启成功。

修改 CPU 弹性扩容策略

开启自定义扩容后，仅支持对实例当前的 CPU 弹性扩容策略进行调整，如需更换成其他策略，请先关闭 CPU 弹性扩容，然后在重新开启 CPU 弹性扩容时更换成其他策略。例如当前的 CPU 弹性扩容策略为自定义扩容2核，扩容时间为立即生效，则仅支持通过**修改**操作调整扩容的核数，不支持通过修改操作更换扩容方式或更换扩容时间。

如您当前的变更控制台允许操作，您可在**实例详情** > **弹性扩容**后单击**修改**，进行重新配置。

关闭自定义扩容

> **说明：**
> 关闭自定义扩容后，当前实例将无法获得弹性支持。

1. 登录 MySQL 控制台，在实例列表，单击**实例 ID** 或操作列的**管理**，进入实例详情页。
2. 在**实例详情** > **弹性扩容**后单击**关闭**。
3. 在**结束扩容 CPU** 的弹窗中单击**确定**。

相关 API

API	描述
开启 CPU 弹性扩容	接口（StartCpuExpand）用于开启实例的 CPU 弹性扩容功能。
查询实例的 CPU 弹性扩容策略信息	接口（DescribeCPUExpandStrategyInfo）用于查询实例的 CPU 弹性扩容信息。
查询扩容历史	接口（DescribeCpuExpandHistory）用于查询扩容历史。
关闭 CPU 弹性扩容	接口（StopCpuExpand）用于关闭实例的 CPU 弹性扩容功能。

开启事件告警

最近更新时间：2025-05-21 14:17:22

云数据库 MySQL 支持事件告警功能，当用户设置了关于 CPU 弹性扩容相关的事件告警并配置告警通知，如果检测到：触发扩容、扩容失败、回缩、关闭扩容等，事件告警将通过多种可选的方式向用户发送告警通知。本文介绍如何设置 CPU 弹性扩容相关的事件告警。

操作场景

用户可以创建事件告警用于在云数据库 MySQL 开启 CPU 弹性扩容功能后，相关状态改变（CPU 性能扩容、CPU 性能扩容失败、CPU 性能回缩）时触发警报并发送相关消息。

操作步骤

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**。
2. 在**告警策略**页，监控类型选择为**云产品监控**，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。这里选择 **云数据库** > **MySQL** > **主机监控/备机监控**。
- **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**，分别添加 CPU 性能扩容、CPU 性能扩容失败、CPU 性能回缩告警事件。

- **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，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 CPU 性能扩容功能相关告警的实例），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。

查询扩容历史

最近更新时间：2025-06-05 15:34:12

云数据库 MySQL 支持查询实例的 CPU 扩容历史，便于您了解实例的 CPU 扩容状态、核数等扩容信息。本文介绍查询扩容历史操作。

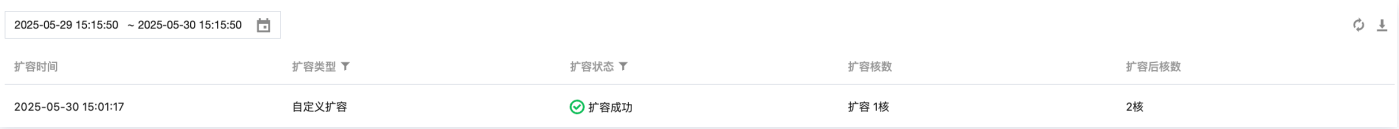
前提条件

实例开启过 CPU 弹性扩容功能，并有扩容记录产生。

扩容历史字段说明

字段	说明
扩容时间	记录扩缩容动作发生的时间。
扩容类型	记录扩容的类型：自动扩容、自定义扩容。支持按扩容类型对列表进行筛选。
扩容状态	记录扩容的状态：扩容成功、扩容失败、缩容成功。支持按扩容状态对列表进行筛选。
扩容核数	记录扩容或缩容的核数。
扩容后核数	记录扩缩容后实例的核数。

列表工具说明



- 支持刷新列表。
- 支持下载列表显示记录到本地的操作，下载文件格式为“.csv”。

操作步骤

1. 登录 MySQL 控制台，在实例列表，单击**实例 ID** 或操作列的**管理**，进入实例详情页。
2. 在**实例详情** > **弹性扩容**上方，单击**扩容历史**。

3. 跳转至**操作日志** > **扩容历史**，选择需要查询的时间范围（最长时间范围为一个月，可精确至秒），完成后即可在列表里查询对应扩容历史。

相关 API

API	描述
查询扩容历史	接口（DescribeCpuExpandHistory）用于查询扩容历史。

性能表现

最近更新时间：2025-06-10 17:00:32

本文介绍通过性能测试工具 SysBench，对比开启 CPU 弹性扩容能力的性能表现。

SysBench 工具介绍

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。

测试环境

> **说明：**
> 以下环境为示例所使用的环境。

云数据库 MySQL 性能测试所使用的环境：

- 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 和云数据库 MySQL 实例网络类型均为私有网络（VPC）且在同一子网下

测试的云数据库 MySQL 实例信息如下：

三组实例：北京七区，通用型2核4G，双节点云数据库 MySQL。

- A 组：NO-CPU-Expand，不开启 CPU 弹性扩容功能。
- B 组：AUTO-CPU-Expand，开启 CPU 弹性自动扩容功能。
- C 组：MANUAL-CPU-Expand，开启 CPU 弹性手动扩容功能。

参数设置：

控制台设置参数 max_prepared_stmt_count 为最高值，以应对高并发。

测试流程

步骤1：

在数据库内创建 sbtest 数据库，开启 B 组 CPU 弹性自动扩容，开启 C 组 CPU 弹性手动扩容。

步骤2：

Sysbench 准备数据，单表10w数据量，10张表。

步骤3：

Sysbench 读写测试，单表10w数据量，10张表。

步骤4：

Sysbench 清除数据，关闭 C 组 CPU 弹性手动扩容。

测试说明

测试会受到时段、时长、机器等多种因素影响，建议多次测试对比结果。

执行命令

根据以上步骤，分别进行测试。

> **说明：**
> 请将以下命令中的 xxxx 替换为云数据库 MySQL 测试实例所需的实际信息，具体参数说明如下：
> - host：测试实例的内网地址。
> - port：端口号。
> - user：用户名。
> - password：上述用户名对应的密码。
> - time：时间，单位为秒。
> - threads：并发数。

- table_size：每个测试表中要插入的记录数。
- tables：测试表的数量。
- events：测试中要执行的事件总数。
- threads：测试中使用的线程数。

1. 准备数据，单表10w数据量，10张表。

```
sysbench --db-driver=mysql --mysql-host=xxxx --mysql-port=xxxx --mysql-user=xxxx --mysql-
password='xxxx' --mysql-db=sbtest --table_size=100000 --tables=10 --events=0 --time=xxxx --threads=xxxx
oltp_read_write prepare
```

2. 运行高并发，单表10w数据量，10张表。

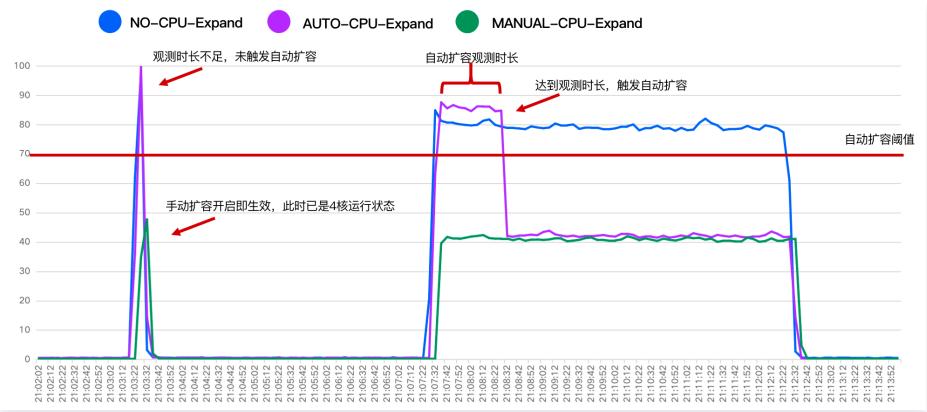
```
sysbench --db-driver=mysql --mysql-host=xxxx --mysql-port=xxxx --mysql-user=xxxx --mysql-
password='xxxx' --mysql-db=sbtest --table_size=100000 --tables=10 --events=0 --time=xxxx  --
threads=xxxx --percentile=95 --report-interval=1 oltp_read_write run
```

3. 清理数据。

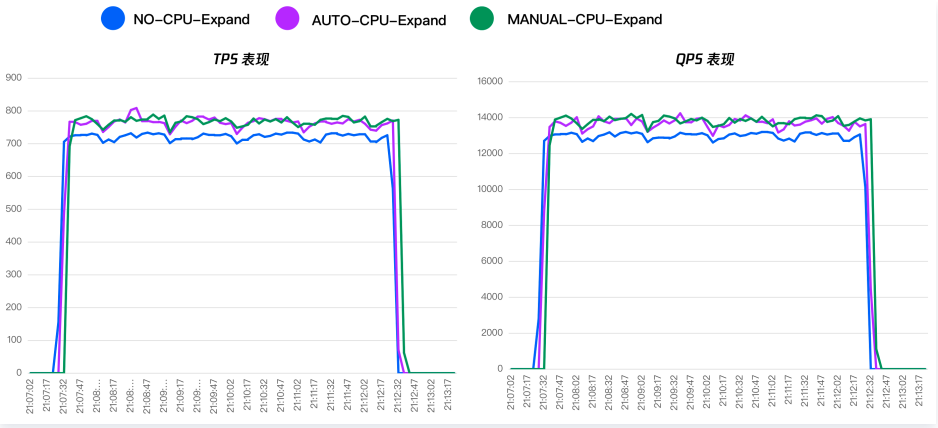
```
sysbench --db-driver=mysql --mysql-host=xxxx --mysql-port=xxxx --mysql-user=xxxx --mysql-
password='xxxx' --mysql-db=sbtest --table_size=100000 --tables=10 --events=0 --time=xxxx --threads=xxxx
--percentile=95  oltp_read_write cleanup
```

测试结果

说明：
如下图所示，在三组 CPU 弹性扩容功能的对比实验结果中，我们可以看到：
- 第一阶段：21:03开启数据注入，NO-CPU-Expand 和 AUTO-CPU-Expand 的 CPU 利用率达到70%以上，但时长未超过1min。因此，AUTO-CPU-Expand 组未触发自动扩容功能，而 MANUAL-CPU-Expand 一开启即扩容，已经是4核运行状态，CPU 利用率不到50%；
- 第二阶段：21:07开启高并发测试，NO-CPU-Expand 的 CPU 利用率始终在70%以上，AUTO-CPU-Expand 的 CPU 利用率在70%以上，超过1min时触发了自动扩容，秒级扩容成功，利用率降低至50%以下，MANUAL-CPU-Expand 的 CPU 利用率始终保持在50%以下。



说明：
除了 CPU 利用率有显著变化外，开启 CPU 弹性扩容功能的实例性能表现同样明显，如下图所示，开启弹性扩容功能的 TPS、QPS 均高于未开启的实例。注意：测试的性能表现与测试时段、时长、机器等因素有一定的关系，建议多次测试对比结果。


NO-CPU-Expand
AUTO-CPU-Expand
MANUAL-CPU-Expand
TPS 表现
QPS 表现

计费说明

最近更新时间：2025-06-05 15:34:12

本文介绍 CPU 弹性扩容计费相关说明。

自动扩容计费说明

- 开启 **CPU 弹性扩容** > **自动扩容**后，成功触发弹性扩容时按扩容的 CPU 核数进行收费，未扩容时不额外收费。
- 当扩容核数为1时，为避免出现持续间歇高峰的影响，在自动扩容时存在最短保护时间，为10min，即自动扩容超过10min，则按实际扩容时间计算费用；自动扩容小于等于10min，则按照10min计算费用。

计费公式

CPU 弹性扩容功能中扩容方式为自动扩容时，采用按量计费，按分钟计费，每小时扣费一次。
计费公式 =（单核费用 x 增加的 CPU 核数）x 扩容时长（分钟）/60。

自动扩容计费示例

示例1

广州地域下某双节点实例的 CPU 规格为4核，触发性能自动扩容后增加至8核，扩容时间为1小时，广州地域的单价为0.54元/核/小时，则计费为：0.54（单价）x 4（额外增加的 CPU 核数）x 1（扩容时长）= 2.16元。

示例2

广州地域下某双节点实例的 CPU 规格为2核，触发性能自动扩容后增加至4核，扩容时间为30分钟，广州地域的单价为0.54元/核/小时，则计费为：0.54（单价）x 2（额外增加的 CPU 核数）x 30/60（扩容时长）= 0.54元。

自定义扩容计费说明

计费公式

CPU 弹性扩容功能中扩容方式为自定义扩容时，采用按量计费，按分钟计费，每小时扣费一次。
计费公式 =（单核费用 x 增加的 CPU 核数）x 扩容时长（分钟）/60。

扩容时间的选择及计费说明

- **当扩容时间为立即扩容时**，开启功能后立即生效并触发扩容，同步开始收费，直至您关闭扩容。关闭扩容操作请参见 设置自定义扩容 。计费示例请参见 示例1 。
- **当扩容时间为按时间段或按周期，未开启自动扩容保障时**，开启功能后按照所选时间段或所选周期生效触发扩容，并同步开始收费，所选时间段外或所选周期外不计费。计费示例请参见 示例2 和 示例3 。
- **当扩容时间为按时间段或按周期，并且开启自动扩容保障时**，开启功能后在所选时间段内或所选周期内生效触发扩容，并同步开始收费，在所选时间段外或所选周期外，成功触发弹性扩容时按扩容的 CPU 核数进行收费，未扩容时不额外收费。计费示例请参见 示例4 和示例5 。

自定义扩容计费示例

示例1

北京地域下某双节点实例的 CPU 规格为2核，在上午10点整该实例开启 CPU 弹性扩容，扩容方式为自定义扩容，扩容时间为立即扩容，额外增加2核 CPU 资源，下午17点整关闭 CPU 弹性扩容，扩容时间为7小时，北京地域的单价为0.54元/核/小时，则该实例关于自定义扩容的计费为：0.54（单价）x 2（额外增加的 CPU 核数）x 7（扩容时长）= 7.56元。

示例2

北京地域下某双节点实例的 CPU 规格为2核，该实例开启了 CPU 弹性扩容，扩容核数为2，扩容方式为自定义扩容，扩容时间为按时间段，开始时间为2025年5月1日00:00，结束时间为2025年5月15日00:00，未开启自动扩容保障，直至5月31日未修改 CPU 弹性扩容的策略。则该实例在2025年5月关于自定义扩容的计费为：0.54（单价）x 2（额外增加的 CPU 核数）x 360（扩容时长）= 388.8元。

示例3

北京地域下某双节点实例的 CPU 规格为2核，该实例于2025年4月30日开启了 CPU 弹性扩容，扩容核数为2，扩容方式为自定义扩容，扩容时间为按周期，周期选择为周一至周五，起止时间为00:00 – 00:00，未开启自动扩容保障，直至5月31日未修改 CPU 弹性扩容的策略。则该实例在2025年5月关于自定义扩容的计费为：0.54（单价）x 2（额外增加的 CPU 核数）x 528（扩容时长）= 570.24元。

示例4

北京地域下某双节点实例的 CPU 规格为2核，该实例开启了 CPU 弹性扩容，扩容核数为2，扩容方式为自定义扩容，扩容时间为按时间段，开始时间为2025年5月1日00:00，结束时间为2025年5月15日00:00，开启了自动扩容保障，直至5月31日未修改 CPU 弹性扩容的策略。假设从5月16日至5月31日该实例触发性能自动扩容的时间为10小时，则该实例在2025年5月关于自定义扩容的计费为：{0.54（单价）x 2（额外增加的 CPU 核数）x 360（扩容时长）} + {0.54（单价）x 2（额外增加的 CPU 核数）x 10} = 581.04元。

示例5

北京地域下某双节点实例的 CPU 规格为2核，该实例于2025年4月30日开启了 CPU 弹性扩容，扩容核数为2，扩容方式为自定义扩容，扩容时间为按周期，周期选择为周一至周五，起止时间为00:00 – 00:00，开启了自动扩容保障，直至5月31日未修改 CPU 弹性扩容的策略。假设该实例在5月1日至5月31日的所有周六和周日总的触发性能自动扩容的时间为20小时，则该实例在2025年5月关于自定义扩容的计费为：{0.54（单价）x 2（额外增加的 CPU 核数）x 528（扩容时长）} + {0.54（单价）x 2（额外增加的 CPU 核数）x 20（扩容时长）} = 591.84。

单核计费单价

单节点本地盘（只读实例）

地域	单价（元/核/小时）
成都、重庆	0.205
广州、上海、北京、南京	0.27
中国香港、东京、首尔、曼谷	0.315
法兰克福、圣保罗	0.25
新加坡、雅加达、硅谷、弗吉尼亚	0.415
深圳金融、上海金融、北京金融	0.405

通用型双节点实例

地域	单价（元/核/小时）
成都、重庆	0.41
广州、上海、北京、南京	0.54
中国香港、东京、首尔、曼谷	0.63
法兰克福、圣保罗	0.5
新加坡、雅加达、硅谷、弗吉尼亚	0.83
深圳金融、上海金融、北京金融	0.81

通用型三节点实例

地域	单价（元/核/小时）
成都、重庆	0.615
广州、上海、北京、南京	0.81
中国香港、东京、首尔、曼谷	0.945
法兰克福、圣保罗	0.75
新加坡、雅加达、硅谷、弗吉尼亚	1.245
深圳金融、上海金融、北京金融	1.215

版权所有：腾讯云计算（北京）有限责任公司

常见问题

最近更新时间：2025-06-05 15:34:12

本文介绍 CPU 弹性扩容相关常见问题。

为什么控制台上没有看到 CPU 弹性扩容功能?

云数据库 MySQL 通用型双节点、三节点版或单节点本地盘（只读实例）才支持 CPU 弹性扩容功能。用户可检查实例是否为单节点（云盘）实例、独享型实例、云盘版（云盘）实例，以上几种类型的实例暂不支持 CPU 弹性扩容功能。

压测时，开启扩容功能后，监控显示 CPU 利用率为何一直为100%?

CPU 利用率会根据实际 CPU 核数计算，当业务压力能够使用完所有 CPU 性能时，应该关注数据库系统的 TPS、QPS、响应时间等性能相关指标。

为什么 CPU 弹性扩容功能没有开启成功?

CPU 弹性扩容功能开启失败有以下两种可能原因：

- 用户的腾讯云账户余额不足。
- 主机 CPU 资源不足（极小概率）。

请保证腾讯云账户余额充足或 提交工单 反馈。

为什么 CPU 扩容、缩容时没有事件通知?

用户可以根据以下情况进行排查和处理。

- 检查是否开启了 CPU 弹性扩容的相关事件告警，如未开启，可参见 开启事件告警 。
- 检查通知人列表，可能未添加相关用户，可参见 新建消息接收人 添加相关用户。

开启 CPU 弹性扩容后，发生了 HA 切换怎么办?

双节点、三节点实例 CPU 弹性扩容功能会同时在主节点和备节点上生效，如果 CPU 弹性扩容功能生效后双节点、三节点实例发生了 HA 切换，那么切换后的节点也将为扩容后的 CPU 规格，自动缩容或关闭自定义扩容后，会自动回缩至原始 CPU 规格。注意：只读实例和灾备实例需单独开启功能，不同时扩容，即主实例 CPU 弹性扩容功能触发生效后，该主实例挂载的只读实例、灾备实例不会同时随之扩容 CPU，均需要单独开启 CPU 弹性扩容功能。

如何查询 CPU 弹性扩容相关任务进度和详情?

支持通过任务列表，查询 CPU 弹性扩容相关任务进度和详情，操作步骤如下：

1. 登录 云数据库 MySQL 控制台 。
2. 在左侧选择任务列表，选择地域，单击**任务类型**，选择**配置弹性扩容策略**，单击**确定**。
3. 在筛选后的任务列表，找到目标任务，即可查询任务进度，单击其操作列的**任务详情**，可查看详细的任务信息。

为什么开启 CPU 弹性扩容会一直收费?

如果开启 CPU 弹性扩容时，扩容方式选择自定义扩容，扩容时间选择立即生效，则在手动关闭前，是会一直扩容的。开启自定义扩容选择立即生效后请注意根据业务情况，操作手动关闭。
建议设置自动扩容，设置后，会根据用户配置的扩缩容规则自动弹性扩缩容 CPU 规格，并且只在成功触发扩容时，对扩容的核数进行计费，未扩容时不计费。

开启 CPU 弹性扩容的自动扩容后，一小时内，触发扩容后持续时间才6min就自动缩容了，为什么收取10min的费用?

当扩容核数为1时，为避免出现持续间歇高峰的影响，在自动扩容时存在最短保护时间，为10min，即自动扩容超过10min，则按实际扩容时间计算费用；自动扩容小于等于10min，则是按照10min计算费用。

开启 CPU 弹性扩容功能后，如何判断扩容是否生效?

您可在实例详情页的实例配置后查询是否生效，若 CPU 核数后额外显示（实际核数），则表示生效。如下图所示，4（8）核表示当前已经是自动扩容到了8核。

不同扩容方式以及不同扩容时间，其扩容生效的策略是怎样的？

- 扩容方式为自动扩容时，当实例的 CPU 平均利用率达到设置的自动弹性扩容阈值时，该实例会自动扩容，当实例的 CPU 平均利用率达到设置的自动弹性缩容阈值时，该实例会自动缩容。
- 扩容方式为自定义扩容，扩容时间为立即生效时，该实例会立即扩容，在未手动关闭功能前，该实例一直会按扩容后的 CPU 核数运行。
- 扩容方式为自定义扩容，扩容时间为按时间段或按周期，未开启自动扩容保障时，该实例在所选时间段或周期内会进行扩容，在所选时间段或周期外会进行缩容。
- 扩容方式为自定义扩容，扩容时间为按时间段或按周期，开启自动扩容保障时，该实例在所选时间段或周期内会进行扩容，在所选时间段或周期外会根据开启自动扩容保障时的扩缩容规则，进行自动扩缩容。

为什么开启了弹性扩容，但一段时间后自动关闭了？

实例进行升级、迁移等操作后，因涉及底层机器资源的变化，会自动关闭弹性功能。如需再次使用 CPU 弹性扩容功能，请重新开启。

只读/灾备实例
创建只读实例

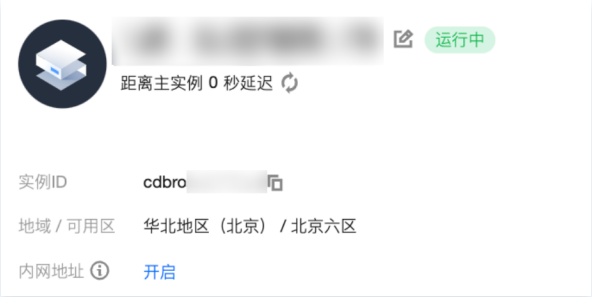
最近更新时间：2025-04-30 10:00:43

操作场景

云数据库 MySQL 支持用户创建一个或多个只读实例，以支持用户的读写分离和一主多从应用场景，可显著提高用户数据库的读负载能力。
在创建只读实例时，有两种不同的引擎可以选择，分别为 InnoDB 引擎和 LibraDB 引擎。InnoDB 引擎可以应对传统情况下主实例无法承载的更多并发读请求。LibraDB 引擎则通过列式存储与并行引擎处理在业务中的复杂 SQL，如多表 Join、大数据的 Group by 、子查询等。
目前已支持数据库代理，创建只读实例后，可以购买数据库代理并开启读写分离功能，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。

说明：

- 只读实例费用请参见 产品定价 。
- 只读实例支持在实例详情页开启独立内网地址，并且支持内网 IP 和端口自定义修改。



- 云盘版实例部署架构为一个读写节点和最多5个只读节点，若您购买的是云盘版实例，增删只读节点的操作请参见 增加实例节点 和 删除实例节点 。云盘版也支持新增独立只读实例，数量为1 – 15个，您可参考本文方法进行创建，需注意独立只读实例不支持切换为读写节点。

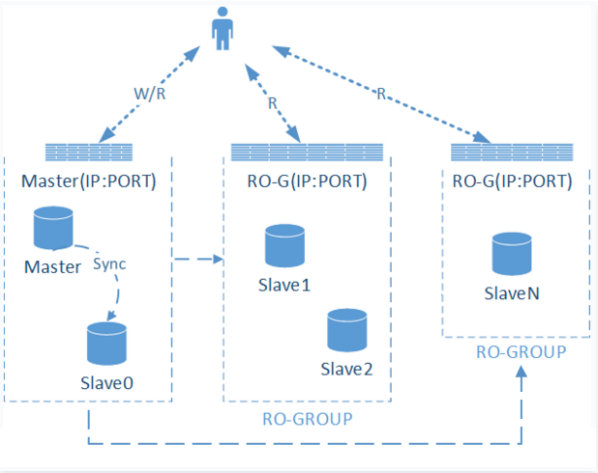
基本概念

- RO 组：带有负载均衡功能的只读实例组。若 RO 组内有多个只读实例，可将用户读请求量均匀分配到组内的每个只读实例上。且 RO 组会对外提供 IP、PORT 供访问数据库。
- 只读实例：可以支持读请求的单节点（无从机）实例。双节点/三节点架构的只读实例无法单独存在，每个只读实例都属于某个 RO 组；云盘版实例的只读节点和独立只读实例不归属于某个 RO 组。

基础架构

只读实例采用 MySQL binlog 主从同步功能，将主实例（源数据库）的更改同步到所有只读实例中。只读实例均采用单节点（不含备机）架构，若只读实例宕机，我们将不断的尝试恢复，若需要更高的可用性，您可以选择 RO 组。

注意：
RO 组中只有一个只读实例的情况会存在单点风险，且此 RO 组不会计入云数据库 MySQL 服务的整体可用性计算中，单个只读实例不提供可用性 SLA 保证，建议 RO 组中至少选购2个只读实例，来保证 RO 组的可用性。



功能限制

- 同一个 RO 组内的实例类型需保持一致，不能既有通用型实例，又有独享型实例。
- 单节点云盘版实例不支持创建只读实例。
- 仅支持**1GB内存、50GB硬盘及以上规格，且 MySQL 5.6 及以上版本、InnoDB 引擎的双节点和三节点主实例/灾备实例、云盘版主实例**购买只读实例，若实例低于此规格，请先升级实例规格。
- 只读实例最低规格要求为1GB内存、50GB硬盘，且必须大于或等于主实例已购存储规格。
- 一个双节点、三节点主实例/灾备实例最多可以创建5个只读实例，一个云盘版实例最多可以创建15个独立只读实例。
- 不支持备份以及回档功能。
- 不支持数据迁移至只读实例。
- 不支持创建/删除数据库，不支持 phpMyAdmin（PMA）。
- 不支持创建/删除账号，不支持为账号授权以及修改账号及密码。

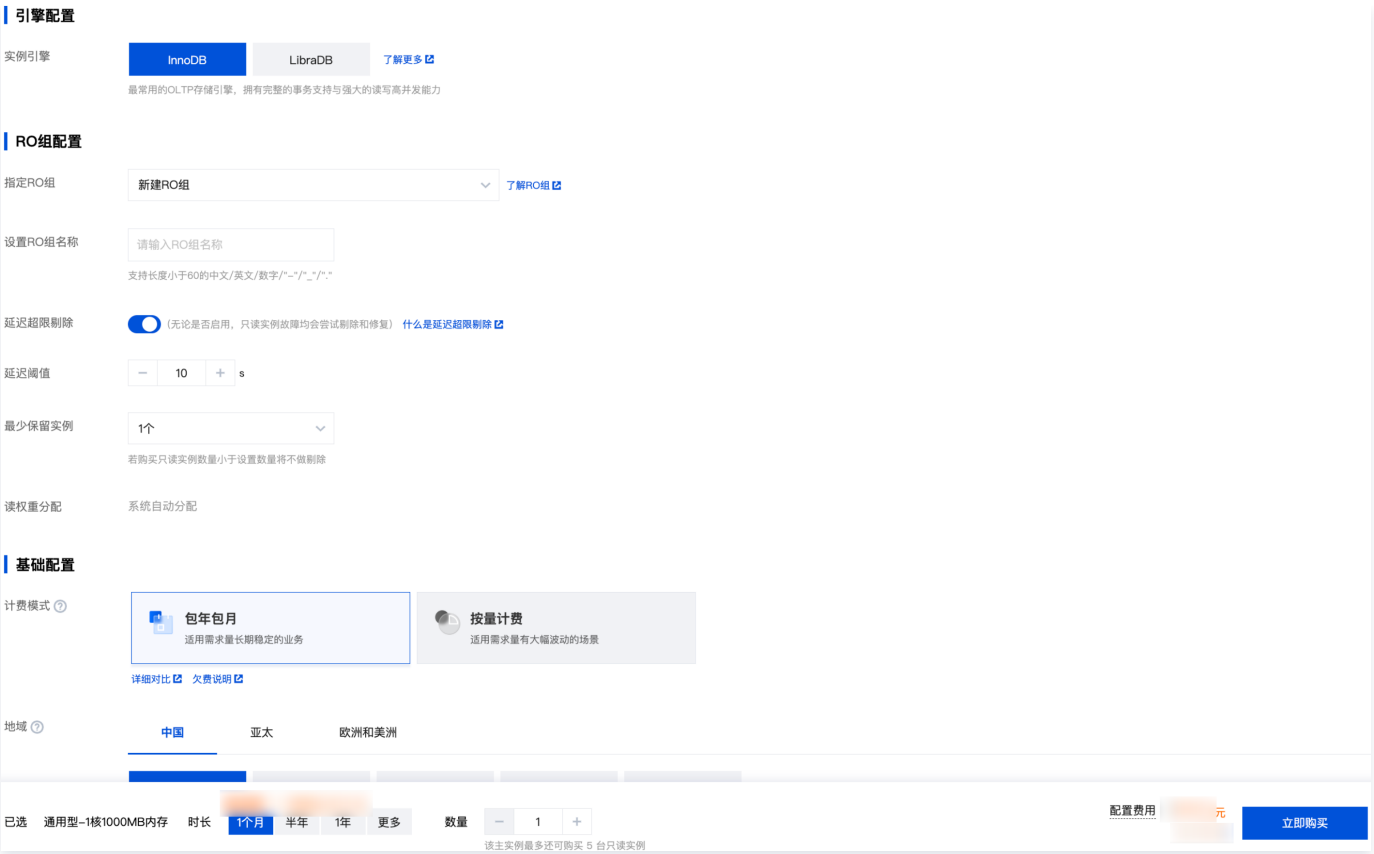
注意事项

- 只读实例无需维护账号与数据库，均从主实例同步。
- 若 MySQL 版本为 5.6 但未开启 GTID，可通过控制台开启 GTID，再添加只读实例。
- 开启 GTID 过程耗时较长，且实例将会有几秒钟的闪断，建议在业务低谷期操作，并在访问数据库的程序中添加重连机制。
- 由于数据同步有延迟，多个只读实例之间可能存在少量数据不一致的情况。各只读实例与主实例之间的同步时延可在控制台查看。
- 只读实例的实例规格可以与主实例不一致，方便用户根据负载情况升级。建议同一 RO 组内的只读实例的实例规格保持一致。

操作步骤

1. 登录 MySQL 控制台，在实例列表，单击**实例 ID** 或**操作**列的**管理**，进入实例详情页。
2. 在实例详情页，单击**实例架构图**中的**添加只读实例**（云盘版为新增独立只读实例），或在只读实例页，单击**新建**，进入购买页。
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**。

> **说明：**
> - 如需统一只读实例与主实例的到期时间，可在 续费管理控制台 设置统一到期日，具体操作步骤请参见 统一到期日 。
> - 云盘版实例新增的独立只读实例，无指定 RO 组相关的操作。



参数	说明
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 InnoDB 引擎。如需了解 LibraDB 引擎的创建和说明，请参见 开启只读分析引擎 。
指定 RO 组	支持系统自动分配、新建 RO 组、已有 RO 组。 系统自动分配：若一次购买多个实例，将为每个实例分配一个独立的 RO 组，权重分配方式默认为系统自动分配。 新建 RO 组：新建一个 RO 组，若一次购买多个实例，将都分配至这个 RO 组，权重分配方式默认为系统自动分配。 已有 RO 组：指定一个已有 RO 组，若一次购买多个实例，将都分配至这个 RO 组。权重分配方式与 RO 组设置相同：如果 RO 组设置为系统自动分配，则根据购买规格自动添加 RO 组；如果为自定义分配，则默认权重为零。由于同一 RO 组内网地址相同，因此若是 VPC 网络将共享同一个安全组设置。若选择已有 RO 组，则在购买时无法再自定义安全组。
设置 RO 组名称	选择新建 RO 组时需设置新 RO 组的名称，支持长度小于60的中文/英文/数字/"-"/"_"/"."
延迟超限剔除	选择是否启用剔除策略。若启用，则需设置延迟阈值和最小保留实例数，被剔除的实例权重自动设置为0。 若只读实例延迟超过阈值被剔除，会向用户发出告警（配置只读实例剔除告警和接收对象请参见 告警功能 ），且实例状态为停服同步中、权重为0，当只读实例延迟时间小于阈值时会重新加入到 RO 组。同时，不管实例是否启用延时超限剔除功能，当只读实例故障被剔除后，待实例修复也会重新加入到 RO 组。
计费模式	支持包年包月和按量计费。
地域	选择只读实例的地域，支持选择与主实例相同或不同的地域，具体可选地域请以实际页面显示为准。
可用区	新建 RO 组时，支持选择与主实例同可用区或跨可用区。不同可用区之间没有实质性区别，选择创建跨可用区 RO 组将提升数据容灾能力，但存在几毫秒的网络延迟。
实例规格	选择只读实例的实例规格，请根据业务实际情况进行规格的选择。
硬盘	选择硬盘空间，硬盘规格上下限跟所选实例规格有关，请以实际购买页面为准。
网络	默认 VPC 与云数据库 MySQL 主实例保持一致，子网为指定可用区的可选子网。同时也支持与主实例不在同一 VPC 网络内。
安全组	可为只读实例选择和对应云数据库 MySQL 实例不同的安全组，以控制实例访问策略。
标签	为只读实例添加标签。

告警策略	为只读实例选择已有告警策略或 新建告警策略 。
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。

4. 购买完成后，返回实例列表，待实例状态变为**运行中**，即可正常使用。

热点问题

只读实例剔除规则是什么？

启用延迟超限剔除后，RO 组会根据延迟阈值和最少保留实例数来判断剔除实例，被剔除的实例权重自动设置为0。若只读实例延迟超过阈值被剔除会向用户发出告警，且实例状态为停服同步中、权重为0，当只读实例延迟时间小于阈值时会重新加入到 RO 组。

- 延迟阈值：为只读实例设置延迟超限阈值，超过阈值的只读实例会被剔除 RO 组。
- 最少保留实例数：组内需要保证的实例下限数，若现有只读实例数小于等于此下限且延迟时间超过阈值，现有只读实例均不被剔除。

若只读实例销毁/退货，对主实例有什么影响？

只读实例销毁/退货，对主实例没有影响。

创建只读实例时，为什么无法选择某个可用区？

无法选择某个可用区，表示该可用区暂无资源，您可以根据实际购买页面选择其他可用区，不影响您使用只读实例。

创建只读实例时，是否可以选择和主实例分布在不同的可用区？

可以。创建只读实例，选择新建 RO 组时，支持选择和主实例不同的可用区，但如果创建只读实例时选择已有 RO 组，则该只读实例的可用区只能与所选的已有 RO 组所在可用区保持一致，不一定和主实例分布在同一可用区。

版权所有：腾讯云计算（北京）有限责任公司

管理只读实例 RO 组

最近更新时间：2025-06-18 10:21:52

操作场景

云数据库 MySQL 支持用户创建一个或多个只读实例组成只读实例 RO 组，适用于读写分离和一主多从应用场景，可显著提高用户数据库的读负载能力。
RO 组是只读实例的集合，一个 RO 组共享一个地址，同时还可以设置其中只读实例的权重进行流量负载均衡、设置延迟剔除等。用户可根据需要部署 RO 组，并将相应的读请求按一定规则发送到只读实例。配置同一个 RO 组内的多个只读实例可起到容灾的作用。
云数据库 MySQL 支持两种类型的 RO 组，分别为：普通 RO 组和分析 RO 组。

- **普通 RO 组**：正常的 InnoDB 引擎的只读实例所使用的 RO 组支持负载均衡、延迟剔除、最小实例保留数等功能。
- **分析 RO 组**：LibraDB 引擎的只读分析引擎实例所使用的 RO 组，仅支持负载均衡的能力。

> **注意：**
> - 分析 RO 组只能管理只读分析引擎，普通 RO 组只能管理只读实例。
> - 仅双节点、三节点架构的主实例或灾备实例支持创建只读实例 RO 组。
> - 如果设置了延迟阈值，只读实例重启或重建后会保持剔除状态，直至延迟恢复到设置的延迟阈值内该只读实例才会重新加入到 RO 组中。

前提条件

创建只读实例之前需要先创建云数据库主实例，请参见 创建 MySQL 实例 。

创建只读实例 RO 组

1. 登录 MySQL 控制台 ，在实例列表，单击**实例 ID** 或**操作**列的**管理**，进入实例管理页。
2. 选择只读实例页，单击**新建**，进入购买页。

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**。
- **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 InnoDB 引擎。
- **指定 RO 组**：选择**新建 RO 组**，若一次购买多个实例，将都分配至这个 RO 组。权重分配方式默认为系统自动分配。
- **设置 RO 组名称**：RO 组名称不要求唯一。支持长度小于60的中文、英文、数字、 `-` 、 `_` 、 `.` 。
- **延迟超限剔除**：指在实例的主从复制过程中，当从库无法及时获取主库的更新内容，并且延迟的时间超过预设的时间阈值时，会自动断开与主库的连接并将从库剔除出复制链路，以保障复制链路的可用性和性能。设置是否启动剔除策略。
  若只读实例延迟超过阈值被剔除会向用户发出告警（配置只读实例剔除告警和接收对象请参见 告警功能 ），且实例状态为停服同步中、权重为0，当只读实例延迟时间小于阈值时会重新加入到 RO 组。同时，不管实例是否启用延时超限剔除功能，当只读实例故障被剔除后，待实例修复也会重新加入到 RO 组。
- **延迟阈值**：为只读实例设置延迟超限阈值，超过阈值的只读实例会被剔除 RO 组。
- **最少保留实例数**：组内需要保证的实例下限数，若现有只读实例数小于等于此下限且延迟时间超过阈值，现有只读实例均不被剔除。
- **读权重分配**：系统自动分配。
- **计费模式**：支持包年包月和按量计费两种计费模式。
- **地域**：默认和主实例一致，也支持选择其他地域。
- **数据库版本**：默认和主实例版本一致。
- **引擎**：默认和主实例一致。
- **架构**：为单节点。单节点架构，性价比高，但单个只读实例存在单点风险，建议对可用性有要求的业务 RO 组内至少选购2个只读实例，保证可用性。
- **数据复制方式**：异步复制。
- **可用区**：新建 RO 组时，支持选择与主实例同可用区或跨可用区。不同可用区之间没有实质性区别，选择创建跨可用区 RO 组将提升数据容灾能力，但存在几毫秒的网络延迟。
- 其他配置详细介绍可参见 创建 MySQL 实例 。
4. 返回实例列表，创建的实例状态为**发货中**，待状态切换成**运行中**，则表示只读实例创建成功。

配置只读实例 RO 组

在只读实例 RO 组的配置界面，可以配置只读实例 RO 组的名称、实例延迟复制、延迟时间、实例延迟超限剔除、延时阈值、最少保留实例、读权重等基本信息。

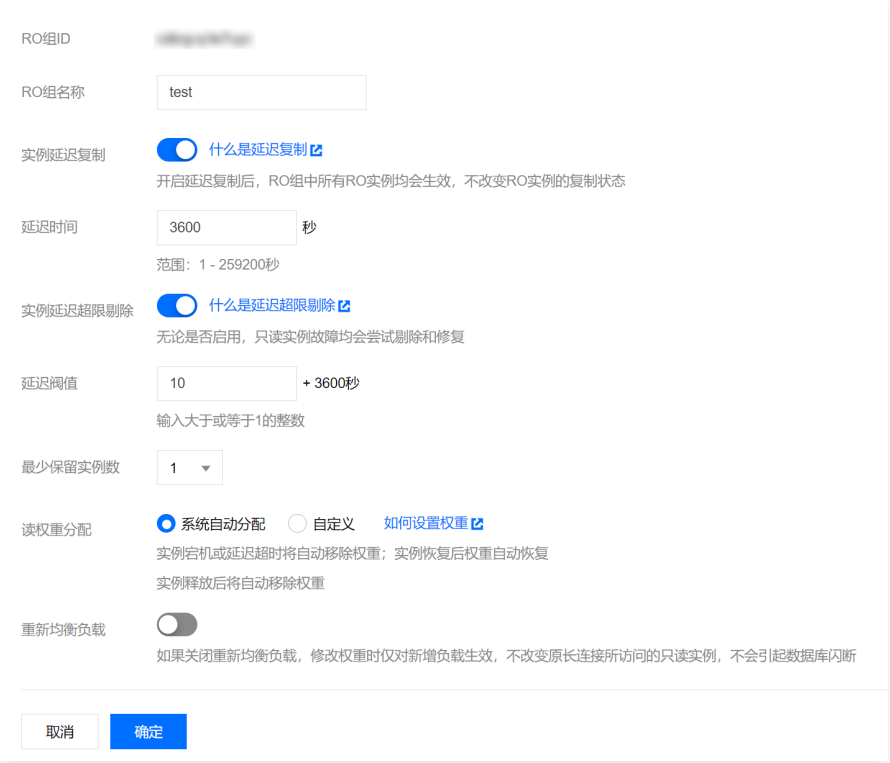
> **说明：**
> - RO 组内只读实例可使用不同规格，读流量权重可设置。
> - 同一 RO 组内只读实例可以支持不同到期时间和计费方式。
> - 同一个 RO 组内的实例类型需保持一致，不能既有通用型实例，又有独享型实例。
> - 开启延迟复制后，RO 组中所有 RO 实例均会生效，不改变 RO 实例的复制状态。
> - 开启延迟复制后，延迟时间选项才会出现。

1. 登录 MySQL 控制台，在实例列表找到目标主实例或灾备实例，单击**实例 ID**，进入实例管理页。
2. 在实例管理页面，选择**只读实例**页，在 RO 组列单击**配置**，进入 RO 组配置页。

3. 在 RO 组配置页，配置 RO 组信息，单击**确定**。
   - **实例延迟复制**：可通过设置延迟复制以及在延迟期间选择启动恢复到指定时间或 GTID（全局事务标识），来实现高效回档数据和故障快速回溯。
   - **延迟时间**：只读实例与主实例间的延迟复制时间，可设置范围：1 – 259200秒。
   - **实例延迟超限剔除**：是否启用剔除策略。被剔除的实例权重自动设置为0。若只读实例延迟超过阈值被剔除会向用户发出告警，配置只读实例剔除告警和接收对象请参见 告警功能 。
   - **延迟阈值**：为只读实例设置延迟超限阈值，超过阈值的只读实例会被剔除 RO 组。
   - **最少保留实例数**：组内需要保证的实例下限数，若现有只读实例数小于等于此下限且延迟时间超过阈值，现有只读实例均不被剔除。
   - **读权重分配**：RO 组支持系统自动分配权重和自定义权重两种权重设置方式。权重输入范围为0 – 100，且必须是整数。系统自动设置双节点、三节点 MySQL 实例的读权重值列表：

实例内存规格（MB）	1000	2000	4000	8000	12000	16000	24000	32000	48000	64000	96000	128000	244000	488000
权重	1	1	2	2	4	4	8	8	10	12	14	16	26	50

- **重新均衡负载**：
  - 关闭重新均衡负载时，修改权重时仅对新增负载生效，不改变原长连接所访问的只读实例，不会引起数据库闪断。
  - 开启重新均衡负载时，数据库会有秒级闪断来断开所有连接，新增连接将按照设置的权重均衡负载。



销毁删除只读实例 RO 组

> **说明：**
> - RO 组不提供手动删除功能。
> - RO 组随着组内最后一个只读实例被彻底销毁而自动删除。
> - 不支持保留空 RO 组。

1. 登录 MySQL 控制台，在实例列表，单击**主实例 ID**，进入实例管理页。
2. 在实例管理页面，选择**只读实例**页，在右侧**操作**列，单击**销毁实例**或**销毁/退还**。
3. 在弹出的对话框，核对销毁信息后单击**确定销毁**，阅读并同意销毁规则后单击**确定**。

热点问题

创建只读实例时，为什么无法选择某个可用区？

无法选择某个可用区，表示该可用区暂无资源，您可以根据实际购买页面选择其他可用区，不影响您使用只读实例。

创建只读实例时，是否可以选择和主实例分布在不同的可用区？

可以。创建只读实例，选择新建 RO 组时，支持选择和主实例不同的可用区，但如果创建只读实例时选择已有 RO 组，则该只读实例的可用区只能与所选的已有 RO 组所在可用区保持一致，不一定和主实例分布在同一可用区。

在已有 RO 组下新建只读实例时失败，提示 `InvalidParameter.RoGroupError.RoCdbTypeError` 是什么原因？

只读实例的实例类型选择错误，同一个 RO 组内的实例类型需保持一致，不能既有通用型实例，又有独享型实例，您可查看对应 RO 组下已有只读实例的实例类型，新建时实例类型与其保持一致。

管理只读实例延迟复制

最近更新时间：2025-06-03 17:59:22

本文为您介绍如何通过 MySQL 控制台设置只读实例延迟复制和启动/停止复制。您可通过设置延迟复制（即只读实例与主实例间的延迟）以及在延迟期间选择启动恢复到指定时间或 GTID（全局事务标识），来实现高效回档数据和故障快速回溯。

- 延迟复制：支持用户通过只读实例 RO 组配置或只读实例管理页开启和设置只读实例与主实例间的延迟复制时间。
- 启动/停止复制：支持用户手动启动或停止只读实例与主实例间的数据同步。

延迟复制说明

- 只读实例开启延时复制后，只读实例会被剔除出只读组且权重设置为0，同时触发只读实例被剔除的告警。此时使用只读组 VIP 地址访问时，流量不会转发到被剔除的只读实例上，仅能使用只读实例的 VIP 地址进行访问。
- 若只读组开启了延迟剔除功能，当只读实例关闭延时复制后，仅只读实例的延迟时间小于只读组的延迟阈值时，才会恢复只读实例的权重，权重恢复的同时，会触发只读实例剔除恢复的告警。
- 启动数据复制至指定位置期间，不能同时进行重启、调整配置、版本升级、升级内核小版本操作。

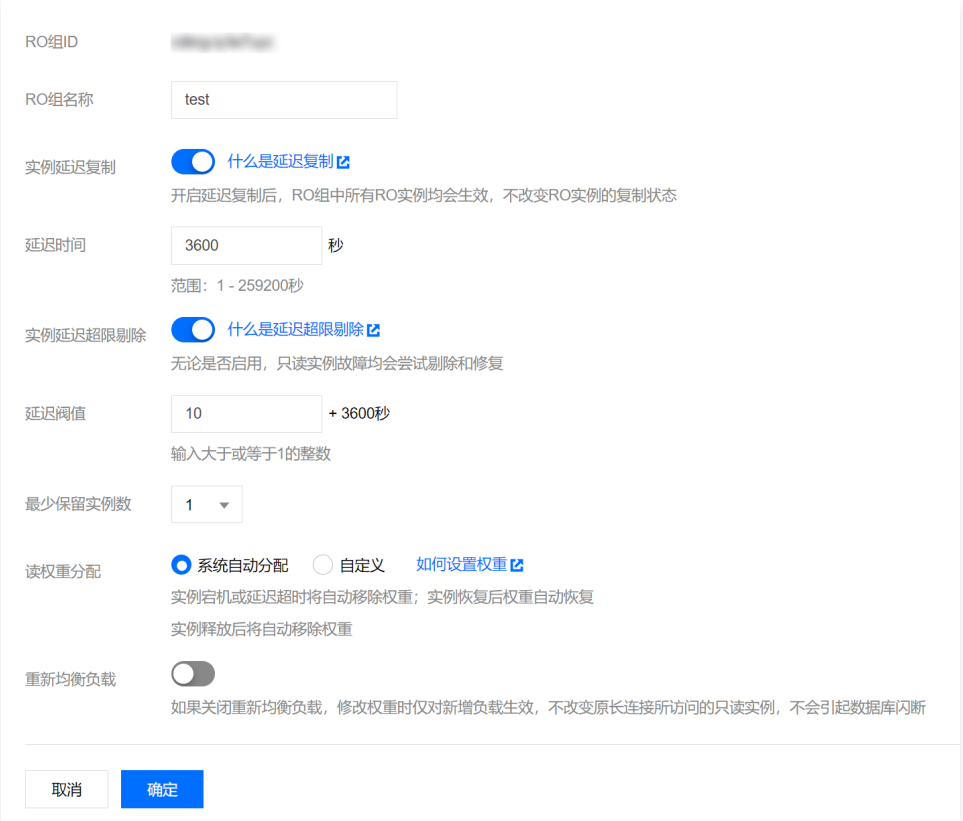
开启延迟复制

> **说明：**
> 只读实例“延迟复制”默认为“不开启”，开启后会显示延迟复制时间。

通过只读实例 RO 组配置开启

1. 登录 MySQL 控制台，在实例列表，单击主**实例 ID**，进入实例管理页。
2. 在实例管理页面，选择**只读实例**页，在 RO 组列单击**配置**，进入 RO 组配置页。

3. 在 RO 组配置页，配置 RO 组信息里开启**实例延迟复制**，**设置延迟时间**，单击**确定**。
   - **实例延迟复制**：可通过设置延迟复制以及在延迟期间选择启动恢复到指定时间或 GTID（全局事务标识），来实现高效回档数据和故障快速回溯。
   - **延迟时间**：只读实例与主实例间的延迟复制时间，可设置范围：1秒 – 259200秒。
   - **实例延迟超限剔除**：指在实例的主从复制过程中，当从库无法及时获取主库的更新内容，并且延迟的时间超过预设的时间阈值时，会自动断开与主库的连接并将从库剔除出复制链路，以保障复制链路的可用性和性能。设置是否启动剔除策略。被剔除的实例权重会自动设置为0。若只读实例延迟超过阈值被剔除会向用户发出告警，配置只读实例剔除告警和接收对象请参见 告警功能 。
   - **延迟阈值**：为只读实例设置延迟超限阈值，超过阈值的只读实例会被剔除 RO 组。
   - **最少保留实例数**：组内需要保证的实例下限数，若现有只读实例数小于等于此下限且延迟时间超过阈值，现有只读实例均不被剔除。
   - **读权重分配**：RO 组支持系统自动分配权重和自定义权重两种权重设置方式。权重输入范围为0 – 100，且必须是整数。
   - **重新均衡负载**：
     - 关闭重新均衡负载时，修改权重时仅对新增负载生效，不改变原长连接所访问的只读实例，不会引起数据库闪断。
     - 开启重新均衡负载时，数据库会有秒级闪断来断开所有连接，新增连接将按照设置的权重均衡负载。



通过只读实例管理页开启

1. 登录 MySQL 控制台，在实例列表，单击只读实例 ID 或操作列的管理，进入只读实例详情页。
2. 在只读实例详情页的部署信息 > 延迟复制处，单击开启。

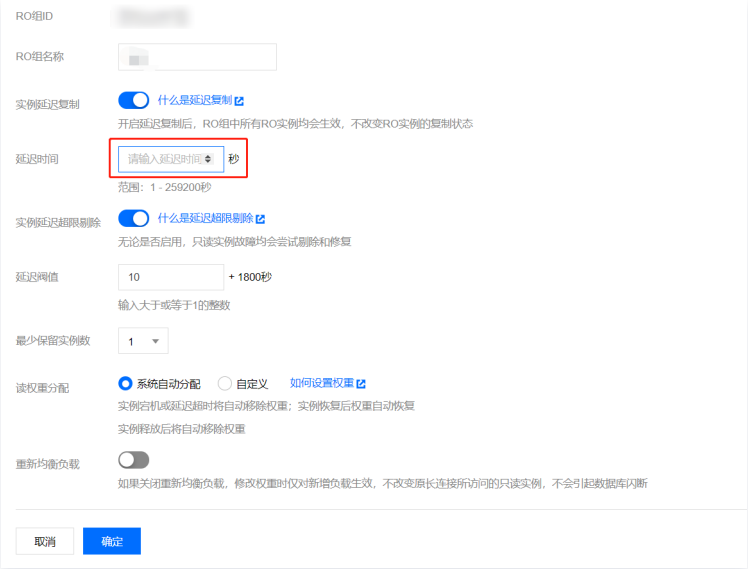
3. 在弹出的对话框，设置延迟时间后，单击确定。

> **说明：**
> 延迟时间范围值为：1秒 – 259200秒。
> 修改 RO 实例延迟时间会同步修改所属 RO 组中其他 RO 实例。

修改延迟复制

通过只读实例 RO 组配置修改

1. 登录 MySQL 控制台，在实例列表，单击**主实例 ID**，进入实例管理页。
2. 在实例管理页面，选择**只读实例**页，在 RO 组列单击**配置**，进入 RO 组配置页。
3. 在 RO 组配置页，修改**延迟时间**，单击**确定**。



通过只读实例管理页修改

1. 登录 MySQL 控制台，在实例列表，单击**只读实例 ID**，进入只读实例详情页。
2. 在只读实例详情页的**部署信息** > **延迟复制**处，单击**修改**。

3. 在弹出的对话框，设置延迟时间后，单击**确定**。

关闭延迟复制

通过只读实例 RO 组配置关闭

1. 登录 MySQL 控制台，在实例列表，单击**主实例 ID**，进入实例管理页。
2. 在实例管理页面，选择**只读实例**页，在 RO 组列单击**配置**，进入 RO 组配置页。
3. 在 RO 组配置页，单击关闭**延迟复制**，单击**确定**。

通过只读实例管理页关闭

1. 登录 MySQL 控制台，在实例列表，单击**只读实例 ID**，进入只读实例详情页。
2. 在只读实例详情页的**部署信息** > **延迟复制**处，单击**关闭**。

3. 在弹出的对话框，确认无误后，单击**确定**。

> **说明：**
> 关闭延迟复制，则延迟复制时间将为0秒，即只读实例与主实例之间将恢复数据的实时同步。

启动数据复制

> **说明：**
> 只读实例**复制状态**默认为**正常**，若用户设置了延迟复制，且在延迟复制的时间段内误删除数据，可通过误操作时的位置与 GTID，将只读实例复制到二进制日志文件位置或 GTID 之前，进而实现快速恢复数据的功能。

1. 登录 MySQL 控制台，在实例列表，单击**只读实例 ID**，进入只读实例详情页。
2. 在只读实例详情页下方的**部署信息** > **复制状态**处，单击**启动**。

3. 在弹出的对话框，单击**确定**。

> **说明：**
> 启动复制后，只读实例恢复与主实例的数据同步。

4. 也可在**部署信息** > **复制操作**选择**复制到指定位置**，支持启动到具体时间点和对应 GTID，恢复到具体时间点或对应 GTID 后，只读实例后续将会停止复制直至切换为正常启动方式，才会继续进行复制。
   - 时间：可选时间范围为停止复制时间到主库当前时间的时间段。
   - GTID：可选范围为只读实例未完成应用的二进制日志之后的所有日志。启动方式选择 GTID 时，可精确到具体事务之前停止复制。
     实例 server_uuid 长度是规定的且均为36位，GTID 格式须为 `server_uuid:transaction_id` 。

> **注意**
> - 若输入的二进制日志位置已在只读实例应用或者大于主实例的位点，会导致启动复制失败。
> - 启动复制时，若有二进制日志存在断点，会导致启动复制失败。
> - 避免因延迟只读实例停止复制过程，导致只读实例磁盘空间超用，只读实例磁盘容量小于5GB时，暂停只读实例的 IO 线程。

5. 复制到指定位置过程中，可单击**复制状态**后的**数据回放中**查询任务详情，支持刷新任务详情。
6. 复制完成后，在**复制状态**后单击**启动**，只读实例即可继续进行复制。

停止数据复制

> **说明：**
> - 仅开启了延迟复制功能，才允许操作停止复制功能，反之则**停止**按钮为不可用状态。
> - 停止复制后同时停止 IO/SQL 线程。

1. 登录 MySQL 控制台，在实例列表，单击**只读实例 ID**，进入只读实例详情页。
2. 在只读实例详情页下方的**部署信息** > **复制状态**处，单击**停止**。

3. 在弹出的对话框，确认无误后，单击**确定**。

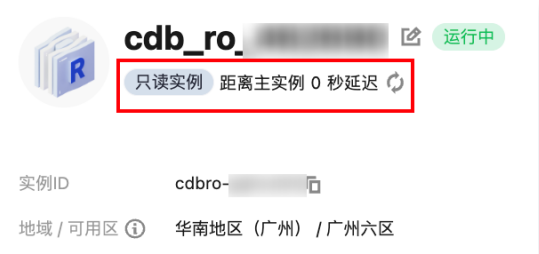
热点问题

如何获取 GTID?

建议您通过执行 flush log 命令获取 binlog 文件，来定位误操作时的位置和 GTID。

如何查看延迟时间?

您可通过 控制台 的实例详情页，查看只读实例与主实例的延迟时间。



如何查看启动复制到指定位置的任务信息?

您可通过 控制台 的任务列表页，查看任务的进度和详情等。

管理灾备实例

最近更新时间：2025-05-21 14:17:22

操作场景

针对业务连续服务和数据可靠性有强需求或是监管需要的场景，云数据库 MySQL 可提供跨地域灾备实例，帮助用户以较低的成本提升业务连续服务的能力，同时提升数据的可靠性。

> **说明：**
> 灾备实例费用与主实例相同，请参见 产品定价 。

功能特点

- 提供独立的数据库连接地址，灾备实例可提供读访问能力，用于就近接入、数据分析等场景，设备冗余成本低。
- 使用主备高可用架构，避免了数据库的单点风险。
- 灾备实例通过内网专线同步，具有较低的同步时延和更高的稳定性，同步链路质量远优于公网网络。
- 目前推广期专线流量费用免费，商业化收费时间将另行通知。

工作原理

- 腾讯云数据库用做灾备数据库的场景下，灾备实例是主实例数据库的复制备份。
- 当主实例发生变化，记录修改的 Log 日志信息会被复制到灾备实例，然后利用日志重放实现数据同步。
- 如果主实例发生故障，可在数秒内激活灾备实例，恢复完整读写功能。

功能限制

- 单节点、云盘版实例不支持创建灾备实例。
- 一个主实例最多支持创建4个灾备实例。
- 仅支持1GB内存、50GB硬盘及以上规格，且 MySQL 5.6及以上版本（MySQL 5.6版本需 提交工单 申请使用此功能）、InnoDB 引擎的双节点/三节点架构的主实例购买灾备实例，若主实例低于此规格，请先升级主实例规格。
- 灾备实例最低规格要求为1GB内存、50GB硬盘，且必须大于或等于主实例已使用存储规格的1.1倍。
- 灾备实例暂不支持：回档、SQL 操作、更改字符集、账号管理、数据导入、回档日志功能。

创建灾备实例

1. 登录 MySQL 控制台，在实例列表，单击**实例 ID** 或**操作**列的**管理**，进入详情页。
2. 在实例详情页的实例架构图中单击**添加灾备实例**，进入灾备实例购买页。

> **说明：**
> GTID（Global Transaction ID）是一种全局事务标识符，用于在复制环境中监测和管理事务，为主实例添加灾备实例，需要主实例已开启 GTID，当前创建实例后，GTID 为默认开启。

3. 在购买页中，设置灾备实例的**计费模式**、**地域**等基本信息。

> **说明：**
> - 创建时长受数据量的影响，期间主实例的控制台操作会被锁定，请妥善安排。

> - 同步策略为立即同步，即创建完灾备实例后会立即同步数据。
> - 暂只支持整个实例的数据同步，请确保磁盘空间充足。
> - 请确保主实例状态为运行中并且没有任何运行相关变更任务执行，如升降配、重启等，否则同步任务有可能失败。

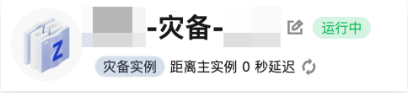
4. 确认无误后，单击**立即购买**，待灾备实例发货。
5. 返回实例列表，待实例状态变为**运行中**，即可进行后续操作。

管理灾备实例

- **查看灾备实例**
  灾备实例可在其所在地域查看，可在实例列表筛选出该地域全部灾备实例。

- **查看从属关系**
  单击每个灾备实例或主实例右方的图标，即可查看从属关系。

- **查看同步延迟**
  可在灾备实例的实例详情页上方查看到主实例和灾备实例之间的同步延迟。



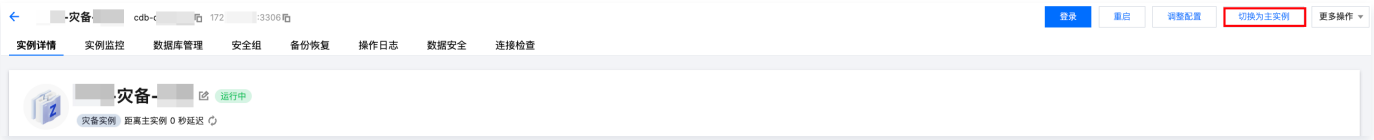
- **灾备实例功能**
  灾备实例提供实例详情、实例监控、数据库管理、安全组、备份恢复、操作日志、数据安全、连接检查等功能。

灾备实例切换为主实例

当您需要切换灾备实例为主实例时，可在控制台主动切换灾备实例为主实例。

> **注意：**
> - 将灾备实例升级为主实例后，灾备实例将成为新的主实例接管运行，新主实例的访问地址会变更，需在业务端重新设置新的访问地址，同时原来的主实例不再承担现有业务。
> - 灾备实例切换为主实例后，将断开与原主实例的同步连接，同步连接断开后不可重连，请谨慎操作。

1. 在 实例列表，选择需要升级为主实例的灾备实例，单击**实例 ID**，进入实例管理页面。
2. 在实例管理页面，单击右上角的**切换为主实例**，即可将灾备实例升级为主实例。切换后将断开与原主实例的同步连接，恢复实例数据库数据写入能力和完整的 MySQL 功能。



数据库代理
概览

最近更新时间：2024-09-13 10:22:01

本章节为用户介绍云数据库 MySQL 的数据库代理相关功能。

章节概览

数据库代理		说明	文档指引
功能简介	数据库代理概述	介绍数据库代理的功能使用场景、代理特性等。	数据库代理概述
	注意事项	介绍数据库代理的注意事项以及各能力支持的版本。	注意事项
内核特性	内核小版本更新说明	介绍数据库代理内核小版本的查看方式及更新记录。	内核小版本更新说明
	升级数据库代理内核小版本	部分能力需要数据库代理高内核版本，不满足的情况，可以通过此文档获取如何升级内核版本的操作方法。	升级数据库代理内核小版本
管理数据库代理	开通数据库代理	介绍开通数据库代理的前提条件、注意事项以及操作步骤。	开通数据库代理
	设置数据库代理连接地址	介绍开通数据库代理后，修改、新增或删除数据库代理连接地址的操作。	设置数据库代理连接地址
	查看和修改访问策略	介绍开通数据库代理后，查看和修改数据库代理连接地址的访问策略。	查看和修改访问策略
	调整数据库代理配置	介绍开通数据库代理后，对代理配置（节点规格/个数，可用区）进行调整的操作。	调整数据库代理配置
	切换数据库代理网络	介绍开通数据库代理后，更改数据库代理连接地址网络的操作。	切换数据库代理网络
	查看数据库代理监控	介绍开通数据库代理后，查看数据库代理各代理节点的连接、请求，或CPU、内存等情况。	查看数据库代理监控
	重新均衡负载	介绍开通数据库代理后，各代理节点的访问出现负载不均衡情况时，手动负载均衡的操作。	重新均衡负载
	关闭数据库代理	介绍关闭数据库代理的操作。	关闭数据库代理
	接入模式	介绍数据库代理的接入模式功能。	接入模式
事务拆分功能		介绍数据库代理的事务拆分功能。	事务拆分功能
防闪断功能		介绍数据库代理的防闪断功能。	防闪断功能
自动读写分离功能	自动读写分离介绍	介绍数据库代理的读写分离功能。	自动读写分离介绍
	设置数据库代理读写属性	介绍开通数据库代理后，对数据库代理连接地址的读写属性进行设置的操作。	设置数据库代理读写属性
连接池功能	连接池概述	介绍数据库代理的连接池功能。	连接池概述
	开启和关闭连接池功能	介绍开启数据库代理后，开启和关闭数据库代理连接池功能的操作。	开启和关闭连接池功能
自适应负载均衡功能		介绍数据库代理的自适应负载均衡功能。	自适应负载均衡功能
其他功能	Hint 语法使用	介绍如何在数据库代理上使用 Hint 语法。	Hint 语法使用

版权所有：腾讯云计算（北京）有限责任公司

简介
数据库代理概述

最近更新时间：2024-10-18 14:14:22

本文介绍云数据库 MySQL 新版数据库代理。
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，具有高可用、高性能、可运维等特点。
云数据库 MySQL 数据库代理还提供自动读写分离、事务拆分、连接池等高级功能，支持挂载跨可用区只读实例、开通多个数据库代理访问地址。

费用说明

数据库代理费用相关请参见 商业化计费说明 。

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。
- 事务内有大量请求，导致主实例负载过高的场景。
- 不同业务需要通过访问地址进行负载分配。
- 跨可用区需要就近访问，降低访问延迟。

读写属性说明

每个数据库代理访问地址都可以分别设置各自的读写属性。

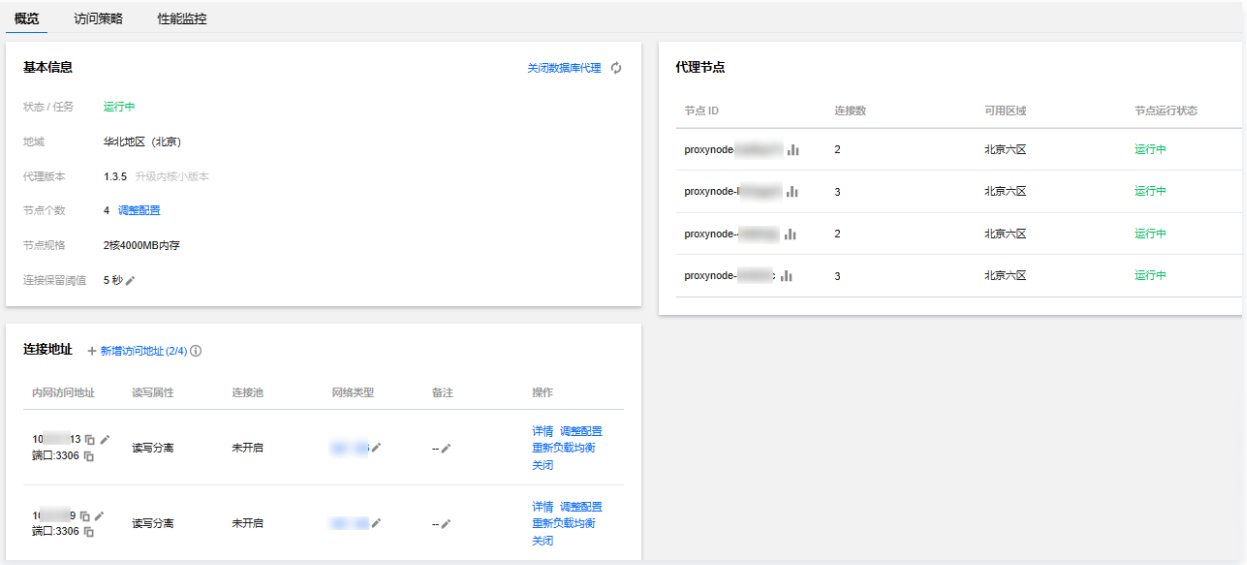
- **读写**：用来支持读写分离功能实现业务线性扩展。
  该属性下至少包含一个主实例和一个只读实例，写请求都只会发往主实例。支持如事务拆分、连接池等读写分离功能，支持只读实例延迟剔除、最小保留实例数、故障转移等策略。
- **只读**：用来支持只读的业务，如报表。
  该属性下至少包含一个只读实例，主实例不会参与路由，支持事务拆分、连接池等功能，支持只读实例延迟剔除、最小保留实例数等策略。

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。
- **高可用**
  可以通过跨可用区部署的方式来提升数据库代理的可用性。
- **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**
  通过配置数据库代理地址的读写属性，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。
  例如，应用程序中只需配置一个代理连接地址（数据库代理读写属性已配置为读写分离时），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。

- **支持连接池**
  有效解决连接数过多或短连接业务频繁建立新连接导致实例负载过高的问题。
- **支持事务拆分**
  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。

数据库代理功能界面



注意事项

最近更新时间：2025-01-21 15:47:02

本文介绍云数据库 MySQL 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，最多支持选择三个可用区，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。
- 可创建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，则支持创建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，代理会连接到主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（主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，可通过重新均衡负载打散连接。
- 数据库代理各能力支持版本：
  - 跨可用区能力请确保代理版本为1.3.1或更高版本。
  - 连接保留阈值请确保代理版本为1.2.1或更高版本。
  - 连接池能力请确保代理版本为1.3.12或更高版本。
  - 事务拆分能力请确保代理版本为1.3.1或更高版本。
  - 新增连接地址能力请确保代理版本为1.3.1或更高版本。
  - 管理只读分析引擎请确保代理版本为1.3.15或更高版本。

> **说明：**
> 升级数据库代理版本操作请参见 升级数据库代理内核小版本 。

数据库代理内核特性
内核小版本更新说明

最近更新时间：2024-10-18 14:14:22

本文介绍云数据库 MySQL 数据库代理的内核版本更新说明。

查看数据库代理版本

您可在实例的 数据库代理 页的**概览** > **代理版本**处查看。

版本更新说明

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 。
- 如不满足云数据库 MySQL 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：2025-05-21 14:17:22

数据库代理会不定期更新内核小版本，部分新能力需要较高代理版本才支持，您可以在控制台手动进行升级。本文为您介绍通过控制台升级数据库代理内核小版本的操作方法。

前提条件

已 开通数据库代理 。

注意事项

- 升级数据库代理内核小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，或确保您的应用有自动重连机制。
- 如出现无法升级情况，可能是数据库内核小版本低于20211031，请先 升级数据库内核版本 ，再升级数据库代理内核版本。
- 数据库代理内核版本1.0、1.1、1.2为较老版本，已经不再维护和更新，可能存在一些已知的问题和漏洞，不建议在生产环境中使用。建议您升级为 1.3.1及以上版本。

> **注意：**
> 数据库代理内核版本1.0和1.1版本里权重为0的 RO 实例不会建连，1.2以上会建连，建议您在升级版本后检查读写分离设置，不参与读写分离的实例需要在控制台上设置不启用，而不是权重设为0。

数据库代理内核版本定义说明

- 尝鲜版：数据库代理的最新版本，通常包含一些新的特性和功能，但也可能存在一些未知 bug。尝鲜版通常是为了让用户体验最新功能而推出的。
- 稳定版：数据库代理的经过测试和验证的版本，通常在生产环境中使用。稳定版相对于尝鲜版来说，更加稳定和可靠，但可能不包含最新的功能和特性。
- 过旧版：数据库代理的较老版本，可能已经不再维护和更新。过旧版可能存在一些已知的问题和漏洞，不建议在生产环境中使用。

操作步骤

1. 登录 MySQL 控制台 ，在上方选择地域，然后单击目标**实例 ID**，进入实例管理页。
2. 在实例管理页，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，检查需升级的目标版本，选择升级切换时间，确认无误后，单击**确定**。
   切换时间：
   - 维护时间内：在维护时间内进行升级，可通过实例详情页修改维护时间信息。
   - 升级完成时：升级操作确认后会立即升级。

> **注意：**
> - 版本升级后不支持回退至低版本。
> - 在升级过程中，会有秒级别闪断，请确保业务具有重连机制。
> - 升级内核小版本是默认所有节点同时升级，若部分节点状态异常，则无法升级。

管理数据库代理
开通数据库代理

最近更新时间：2025-05-15 14:29:32

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、事务拆分、连接池、连接保持等高级功能，具有高可用、高性能、可运维、简单易用等特点。
本文为您介绍通过云数据库 MySQL 控制台为实例开启数据库代理。

前提条件

- 实例状态为运行中。
- 实例为双节点、三节点或云盘版架构。

注意事项

- 双节点经济型的实例暂不支持数据库代理。
- 升级主实例内核小版本会同时升级所关联的只读实例和灾备实例的内核小版本。
- 若实例的内核小版本过低，不支持开启数据库代理，请参见 升级内核小版本 操作升级后再开启数据库代理。

支持地域及版本

- 开启数据库代理时代理支持选择的地域及可用区如下：

> **说明：**
> - 上海（除一区），表示为实例开启数据库代理时，除了上海一区，其他上海的可用区均支持选择。
> - 成都，表示为实例开启数据库代理时，成都地域下的所有可用区均支持选择。

  - 北京（除一、二、四、八区）、上海（除一区）、广州（除一区）、上海金融（除一、二区）、北京金融、成都、重庆、南京、中国香港。
  - 东京、曼谷（除一区）、弗吉尼亚、硅谷、首尔、新加坡（除一区）、雅加达（除一、三区）、圣保罗、法兰克福。
- 数据库代理目前支持的版本：
  - 双节点、三节点 MySQL 5.7（内核小版本须20211030及以上）。
  - 双节点、三节点 MySQL 8.0（内核小版本须20211202及以上）。
  - 云盘版 MySQL 5.7、MySQL 8.0。

操作步骤

1. 登录 MySQL 控制台，在实例列表，选择需要开启代理的主实例，单击**实例 ID** 或**操作**列的**管理**，进入实例管理页面。
2. 在实例管理页面，选择**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**下一步：网络、安全组设置**。

参数	说明
代理版本	选择数据库代理的版本，各代理版本的更新说明请参见 数据库代理版本更新说明 。
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	• 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，最多支持选择三个可用区。 • 选择节点个数，推荐的代理节点个数计算方法与代理规格相关。 ○ 代理规格选择2核4000MB内存，则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。 ○ 代理规格选择4核8000MB内存，则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1/16（向上取整）。例如主实例为8核 CPU，只读实例为16核 CPU，则推荐代理数量 = (8 + 16) / 16 ≈ 2。 ○ 代理规格选择8核16000MB内存，则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1/32（向上取整）。例如主实例为64核 CPU，只读实例为128核 CPU，则推荐代理数量 = (64 + 128) / 32 ≈ 6。 **说明：** • 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，建议选择更高的代理规格。

4. 完成网络、安全组设置，单击**立即购买**。

参数	说明
选择网络	选择数据库代理的网络，仅支持私有网络 VPC。
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。 **说明：** 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 MySQL 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。

5. 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，在连接地址下对数据库代理访问地址、网络类型和备注进行修改，也可在连接地址操作项查看连接配置详情、调整配置、重新均衡负载操作。

> **说明：**
> - 您可通过查看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**重新均衡负载**打散连接。
> - 重新均衡负载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。

设置数据库代理连接地址

最近更新时间：2025-06-19 11:16:02

本文为您介绍通过云数据库 MySQL 控制台设置数据库代理连接地址。
数据库代理连接地址独立于原有的数据库连接地址，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。
云数据库 MySQL 在开通数据库代理后默认提供一个数据库代理连接地址，您也可以额外新增数据库代理连接地址、修改数据库代理连接地址或删除数据库代理连接地址。

前提条件

已 开通数据库代理 。

修改数据库代理连接地址

1. 登录 MySQL 控制台 ，在实例列表，选择已开启代理的主实例，单击**实例 ID** 或**操作**列的**管理**，进入实例管理页面。
2. 在实例管理页面，选择**数据库代理**页，在**连接地址**的对应**内网访问地址**后单击 图标。

3. 在弹出的对话框，修改代理地址后，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，建议在低峰期修改，请确保业务具备重连机制。

新增数据库代理连接地址

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

1. 登录 MySQL 控制台 ，在实例列表，选择已开启代理的主实例，单击**实例 ID** 或**操作**列的**管理**，进入实例管理页面。
2. 在实例管理页面，选择**数据库代理**页，在**连接地址**后单击**新增访问地址**。

3. 在新建连接窗口，完成如下配置，单击**确定**。
   步骤1：配置网络

参数	说明
网络	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。 **说明：** 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 MySQL 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。
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，详情请参见 接入模式 。
只读实例延迟剔除	设置只读实例延迟剔除策略，此项开启，可设置延迟剔除阈值和只读实例最小保留数。无论此项是否启用，只读实例故障时均会尝试剔除和恢复。 延迟剔除阈值：输入大于1或等于1的整数，单位为秒。 只读实例最小保留数：数量与主实例拥有只读实例数相关，设置为0时，当只读节点全部被剔除后，所有访问会转发到主实例上，直至只读实例重新加入。
自适应负载均衡	选择是否开启自适应负载均衡功能，开启后可以根据后端的负载情况，将请求路由到负载更低的实例上，以达到数据库实例负载均衡，详情请参见 自适应负载均衡功能 。
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，目前支持会话级别连接池，事务级别连接池。
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。
读权重分配	支持选择系统自动分配或自定义，如开通数据库代理时配置了多个可用区，则支持对通过不同可用区下的代理节点访问数据库的权重进行分别配置。 **说明：** 此处的权重是面向读请求（非事务）权重的分配策略。
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。

删除数据库代理连接地址

> **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，但最后一个地址不支持删除。

1. 登录 MySQL 控制台 ，在实例列表，选择已开启代理的主实例，单击**实例 ID** 或**操作**列的**管理**，进入实例管理页面。
2. 在实例管理页面，选择**数据库代理**页，在**连接地址**下找到目标地址，单击**关闭**。

3. 在弹出的对话框单击**确定**。

查看和修改访问策略

最近更新时间：2025-01-21 15:47:02

云数据库 MySQL 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 MySQL 控制台，在上方选择地域，然后单击**目标实例 ID**，进入实例管理页。
2. 在实例管理页，选择**数据库代理** > **访问策略**。

修改访问策略

1. 登录 MySQL 控制台，在上方选择地域，然后单击**目标实例 ID**，进入实例管理页。
2. 在实例管理页，选择**数据库代理** > **访问策略**，找到目标访问策略，单击**设置**。

> **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**。

3. 在跳转的窗口下，修改具体策略的配置，单击**确定**。

注意：
只读分析引擎不支持“事务拆分”、“连接池”、“只读实例延迟剔除”、“读写分离”、“权重设置”等数据库代理特性。仅支持通过 Hint 语法使用 访问只读分析引擎。

参数	说明
读写属性	修改此代理访问地址的读写属性，支持选择读写分离或只读。若读写属性选择为读写分离，则读权重分配时，主实例/读写节点默认启用且不支持关闭。
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，详情请参见 接入模式 。
只读实例延迟剔除	设置只读实例延迟剔除策略，此项开启，可设置延迟剔除阈值和只读实例最小保留数。无论此项是否启用，只读实例故障时均会尝试剔除和恢复。 延迟剔除阈值：输入大于1或等于1的整数，单位为秒。 只读实例最小保留数：数量与主实例拥有只读实例数相关，设置为0时，当只读节点全部被剔除后，所有访问会转发到主实例上，直至只读实例重新加入。
自适应负载均衡	选择是否开启自适应负载均衡功能，开启后可以根据后端的负载情况，将请求路由到负载更低的实例上，以达到数据库实例负载均衡，详情请参见 自适应负载均衡功能 。
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。此项开启后，可选择支持的连接池类型，目前支持会话级别连接池，事务级别连接池。
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。
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，若开通数据库代理时配置了多个可用区，则支持对不同可用区下的代理节点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。 • 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。
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。

调整数据库代理配置

最近更新时间：2024-10-18 14:14:22

开通数据库代理后，您可对数据库代理的配置进行调整，可通过控制台对数据库代理的代理规格、可用区以及节点个数进行修改。本文为您介绍调整数据库代理配置的操作。

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，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。
- 在该地域资源充足情况下，最多支持选择三个可用区，最少保留一个可用区，保留的可用区为第一行所选可用区。

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	负载均衡方式	切换时间	影响
升级或降级	不变更	不变更	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	-	不会导致连接闪断。
		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	-	不会导致连接闪断
		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	

版权所有：腾讯云计算（北京）有限责任公司

			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 MySQL 控制台，在上方选择地域，然后单击目标**实例 ID**，进入实例管理页。
2. 在实例管理页，选择**数据库代理**页。
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**，即完成配置调整。

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**。
- 若重新均衡负载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均衡负载。

切换数据库代理网络

最近更新时间：2024-09-13 10:22:01

本文为您介绍通过云数据库 MySQL 控制台切换数据库代理的网络。

前提条件

已 开通数据库代理 。

注意事项

- 切换网络会导致该实例数据库代理 IP 变化，如果切换网络时没有手动设置旧 IP 回收时间，则旧 IP 在默认保留24小时后会失效，请及时修改客户端程序上的对应 IP。
- 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 MySQL 实例所在地域下的 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。
- 添加多个数据库代理连接访问地址，可分别为每个访问地址的网络进行设置和修改。

操作步骤

1. 登录 MySQL 控制台，在上方选择地域，然后单击目标实例 ID，进入实例管理页。
2. 在实例管理页，选择**数据库代理**页。
3. 在数据库代理页的**概览** > **连接地址** > **网络类型**下，单击 。

4. 在弹出的对话框，选择新网络，单击**确定**。

- 设置旧 IP 地址回收时间，可设置范围为0 – 168小时。
- 选择系统自动分配 IP 地址或手动指定 IP 地址。

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：2024-09-13 10:22:01

本文为您介绍如何通过云数据库 MySQL 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 。

支持的监控指标

指标项中文名	单位	说明
当前连接数	个	当前节点访问连接个数
请求数	次/秒	访问节点的请求数
读请求数	次/秒	读操作请求数
写请求数	次/秒	写操作请求数
CPU 利用率	%	CPU 的使用情况
内存利用率	%	内存利用情况
内存占用	MB	已使用内存情况

操作步骤

方法一：

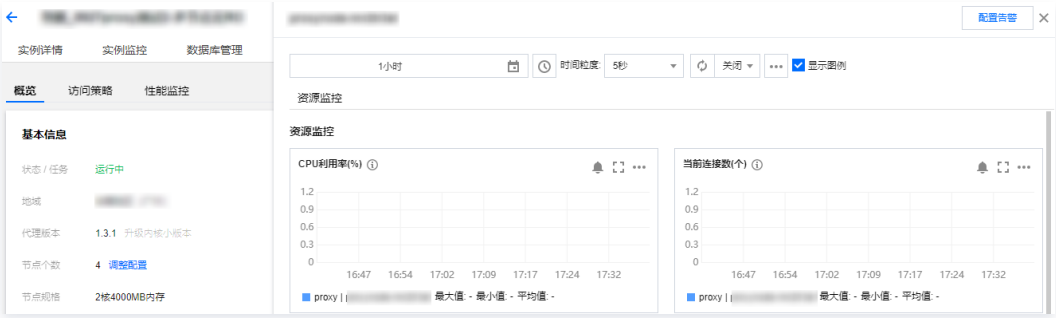
1. 登录 MySQL 控制台，在实例列表，选择已开启代理的主实例，单击**实例 ID** 或**操作**列的**管理**，进入实例管理页面。
2. 在实例管理页面，选择**数据库代理** > **性能监控**页，默认可以查看该实例下所有代理节点的监控，在某一指标的图例中单击对应代理节点，可只显示所选代理节点的指标情况。

> **说明：**
> 粒度为5秒的监控，时间跨度4小时内的监控默认切换为粒度5秒。

方法二：

1. 登录 MySQL 控制台，在实例列表，选择已开启代理的主实例，单击**实例 ID** 或**操作**列的**管理**，进入实例管理页面。
2. 在实例管理页面，选择**数据库代理** > **概览**，在**代理节点**列，单击目标节点 ID 后的 图标，可直接跳转查看该节点的性能监控情况。

跳转后界面如下：


实例详情
实例监控
数据库管理
概览
访问策略
性能监控
基本信息
状态 / 任务
运行中
地域
代理版本
1.3.1
节点个数
4
调整配置
节点规格
2核4000MB内存
配置告警
1小时
时间粒度
5秒
关闭
显示图例
资源监控
资源监控
CPU利用率(%)
当前连接数(个)
1.2
0.9
0.6
0.3
0
16:47
16:54
17:02
17:09
17:17
17:24
17:32
proxy
最大值: - 最小值: - 平均值: -

重新均衡负载

最近更新时间：2024-10-18 14:14:22

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，可通过重新均衡负载打散连接。本文介绍通过控制台手动操作重新均衡负载。

> **说明：**
> 重新均衡负载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。

前提条件

已 开通数据库代理 。

操作步骤

1. 登录 MySQL 控制台，在上方选择地域，然后单击目标**实例 ID**，进入实例管理页。
2. 在实例管理页，选择**数据库代理** > **概览**，在连接地址下找到目标访问地址，在其**操作**列单击**重新均衡负载**。

3. 在弹出的对话框中单击**确定**。

关闭数据库代理

最近更新时间：2025-05-21 14:17:22

本文为您介绍通过云数据库 MySQL 控制台关闭数据库代理。

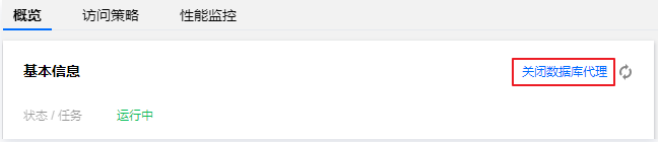
> **说明：**
> 关闭数据库代理前，请确认您的客户端没有通过数据库代理地址访问数据库，关闭数据库后，产生的数据会被删除，不可恢复。

前提条件

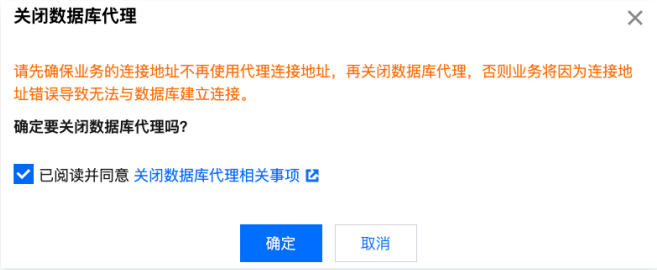
已 开通数据库代理 。

操作步骤

1. 登录 MySQL 控制台 ，在实例列表，选择需要关闭代理的实例，单击**实例 ID** 或**操作**列的**管理**，进入实例管理页面。
2. 在实例管理页面，单击**数据库代理** > **概览** > **基本信息**里的**关闭数据库代理**。



3. 在弹出的对话框，确认无误后，单击**确定**。



接入模式

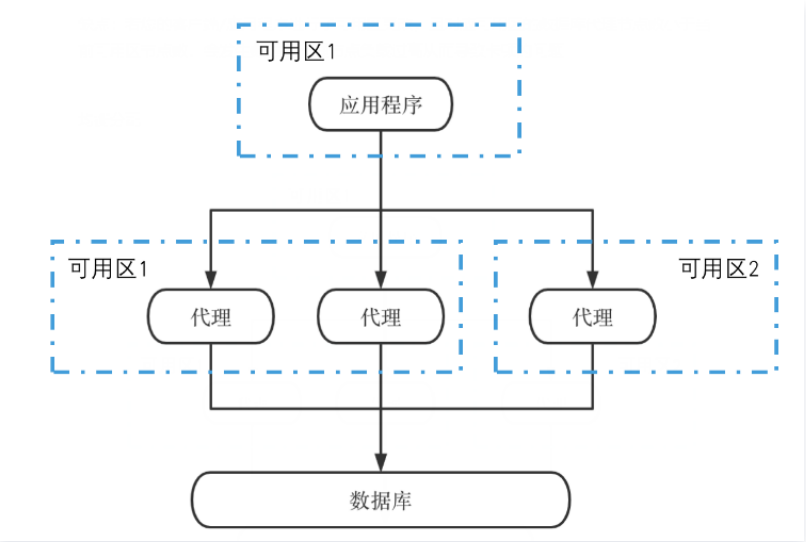
最近更新时间：2025-04-30 10:00:43

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，以下分别为您介绍云数据库 MySQL 数据库代理的两种接入模式的规则和优缺点。

前提条件

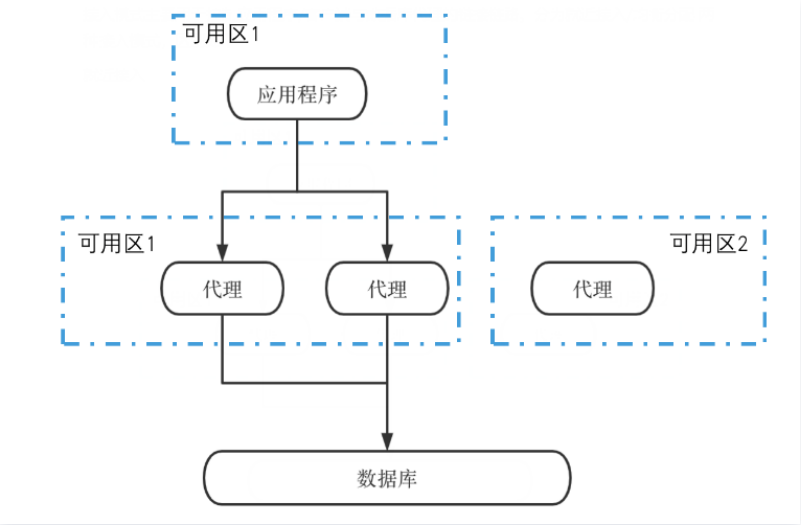
- 已 开通数据库代理 。
- 数据库架构为双节点、云盘版。

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：2025-05-21 14:17:22

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。云数据库 MySQL 的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**主实例**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**主实例**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 MySQL 控制台，在实例列表，选择已开启代理的主实例，单击**实例 ID** 或**操作**列的**管理**，进入实例管理页面。
2. 在实例管理页面，选择**数据库代理**页，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，即可查看不同类型的实例故障转发策略。

事务拆分功能

最近更新时间：2024-10-30 21:37:12

云数据库 MySQL 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。

背景信息

默认情况下，云数据库 MySQL 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。
事务拆分功能默认关闭，您可通过对数据库代理访问地址调整配置，开启事务拆分能力。

前提条件

已 开通数据库代理 。

操作步骤

1. 登录 MySQL 控制台 ，在上方选择地域，然后单击目标实例 ID，进入实例管理页。
2. 在实例管理页，选择**数据库代理** > **访问策略**，找到目标访问策略，单击**设置**。

> **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**。

3. 在调整配置窗口，将事务拆分后的按钮设置为打开，单击**确定**。

防闪断功能

最近更新时间：2025-05-16 17:30:02

本文介绍云数据库 MySQL 数据库代理的防闪断能力。

支持实例架构

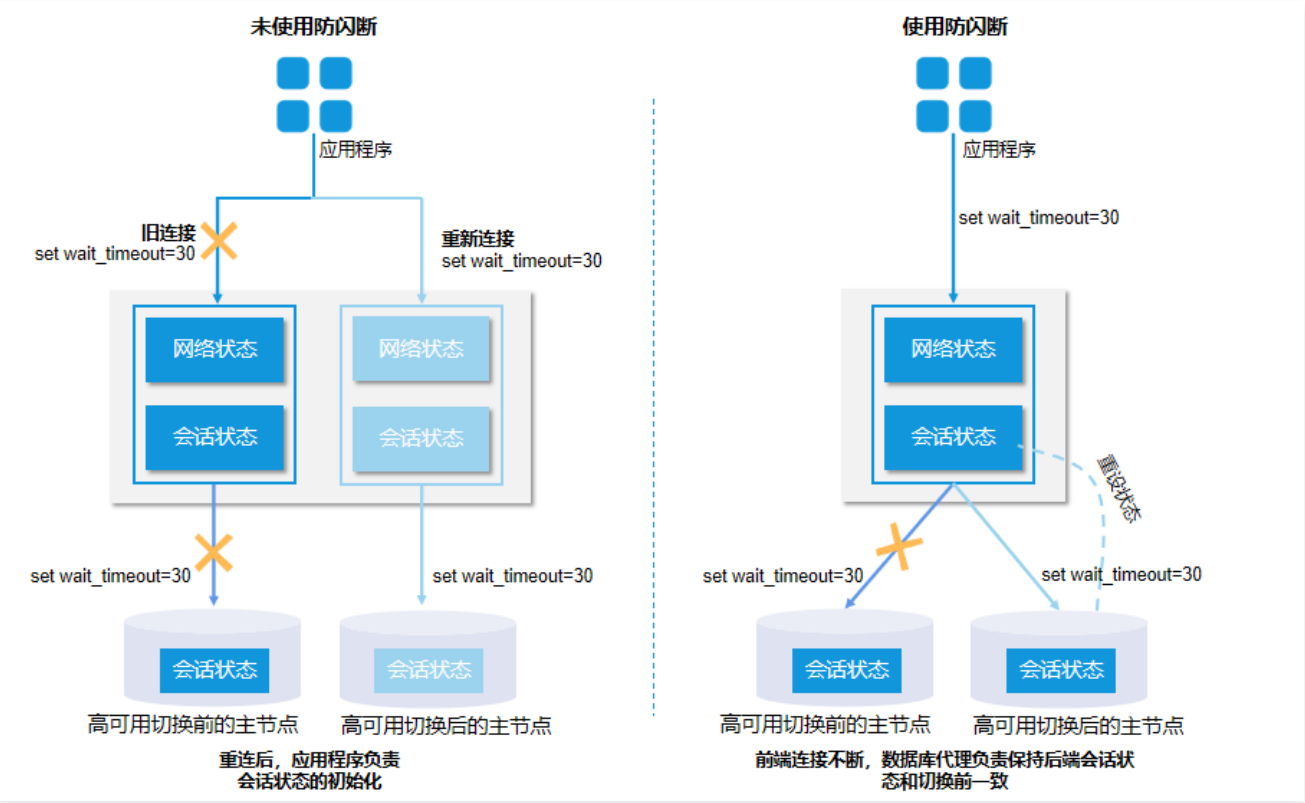
MySQL 双节点（高可用版）、MySQL 三节点。

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。云数据库 MySQL 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 MySQL 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

以下为您介绍云数据库 MySQL 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 和云数据库 MySQL 实例网络类型均为私有网络（VPC）且在同一子网。

测试的云数据库 MySQL 实例信息如下。

- 存储类型：本地 SSD 盘。
- 实例类型：通用型。
- 参数模板：高性能模板。

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，测试高可用 MySQL 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，高可用 MySQL 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
主备切换	100%
内核小版本升级	100%
调整实例规格	100%

自动读写分离功能
自动读写分离介绍

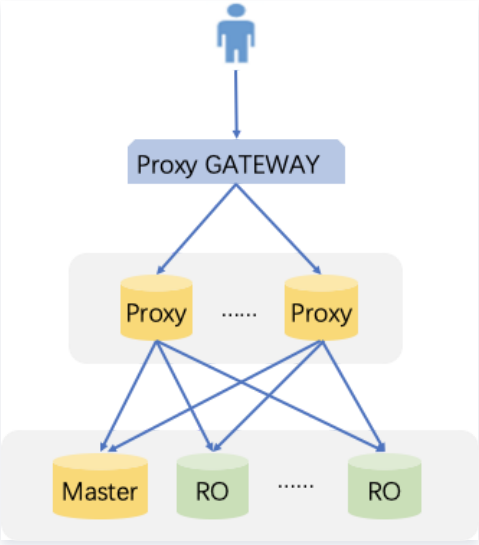
最近更新时间：2024-10-18 14:14:22

本文主要介绍云数据库 MySQL 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。
- 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。

功能说明

云数据库 MySQL 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。

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：2024-09-13 10:22:01

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设置数据库代理的读写属性。

前提条件

已 开通数据库代理 。

操作步骤

1. 登录 MySQL 控制台 ，在上方选择地域，然后单击目标实例 ID，进入实例管理页。
2. 在实例管理页，选择**数据库代理** > **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**。

3. 在调整配置页面，选择对此访问连接的读写属性，完成读权重分配，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。

调整配置
读写属性
读写分离
只读
只读实例延迟剔除
什么是延迟超限剔除
无论是否启用，只读实例故障均会尝试剔除和恢复
连接池状态
什么是连接池
连接池类型
会话级别连接池
事务拆分
读权重分配
系统自动分配
自定义
北京六区
实例 ID / 名称
类型
启用
权重
状态
可用区
主实例
2（自动分配
运行中
北京六区
只读实例
2（自动分配）
运行中
北京六区
故障转移
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
购买新的非延迟只读实例，会自动添加到数据库代理中
确定
取消

连接池功能
连接池概述

最近更新时间：2024-10-18 14:14:22

云数据库 MySQL 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别连接池、事务级别连接池，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。

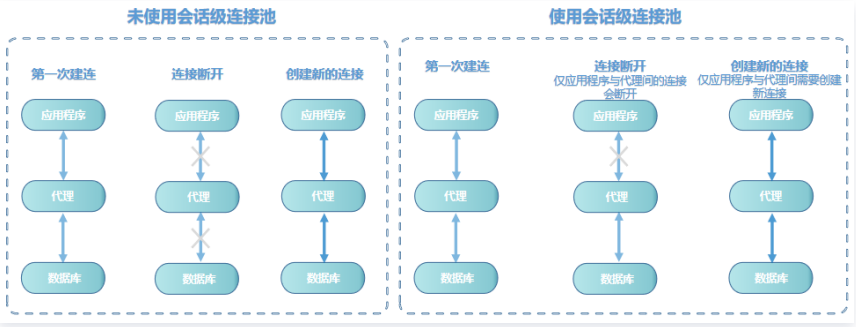
前提条件

- 已 开通数据库代理 。
- 数据库内核小版本为 MySQL 5.7 20211030及以上。
- 数据库内核小版本为 MySQL 8.0 20230630及以上。

> **注意：**
> 数据库代理连接池能力在较低 MySQL 内核小版本中存在 change user 时产生死锁的问题，该问题已在 MySQL 8.0 20230630 内核小版本中修复，若您的主实例内核小版本未达到 MySQL 8.0 20230630，则存在一定的风险，建议您关闭数据库代理连接池能力；若您要继续使用数据库代理连接池能力，建议升级主实例内核小版本至 MySQL 8.0 20230630 或以上版本，升级操作请参见 升级数据库内核小版本 。

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5s，开启连接池功能后支持手动设置，参见 开启连接池功能 ）
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速率来减少 MySQL 主线程的开销，能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。

开启和关闭连接池功能

最近更新时间：2024-10-18 14:14:22

本文为您介绍开启和关闭连接池功能。

前提条件

- 已 开通数据库代理 。
- 数据库内核小版本为 MySQL 5.7 20211030及以上。
- 数据库内核小版本为 MySQL 8.0 20230630及以上。

> **注意：**
> 数据库代理连接池能力在较低 MySQL 内核小版本中存在 change user 时产生死锁的问题，该问题已在 MySQL 8.0 20230630 内核小版本中修复，若您的主实例内核小版本未达到 MySQL 8.0 20230630，则存在一定的风险，建议您关闭数据库代理连接池能力；若您要继续使用数据库代理连接池能力，建议升级主实例内核小版本至 MySQL 8.0 20230630 或以上版本，升级操作请参见 升级数据库内核小版本 。

开启连接池功能

1. 登录 MySQL 控制台，在上方选择地域，然后单击目标**实例 ID**，进入实例管理页。
2. 在实例管理页，选择**数据库代理** > **访问策略**，找到目标访问策略，单击**设置**。

> **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**。

3. 在**调整配置**窗口下，打开连接池状态后的按钮，打开后选择开启会话级别连接池或事务级别连接池（需数据库代理内核版本1.4.1及以上），单击**确定**。

4. 开启连接池功能后，连接保留阈值默认为5s，支持在基本信息下对连接保留阈值进行修改，修改范围为：1 – 300。

关闭连接池功能

1. 登录 MySQL 控制台，在上方选择地域，然后单击目标实例 ID，进入实例管理页。
2. 在实例管理页，选择**数据库代理** > **访问策略**，找到目标访问策略，单击**设置**。

> **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**。

3. 在**调整配置**窗口下，关闭连接池状态后的按钮，单击**确定**。

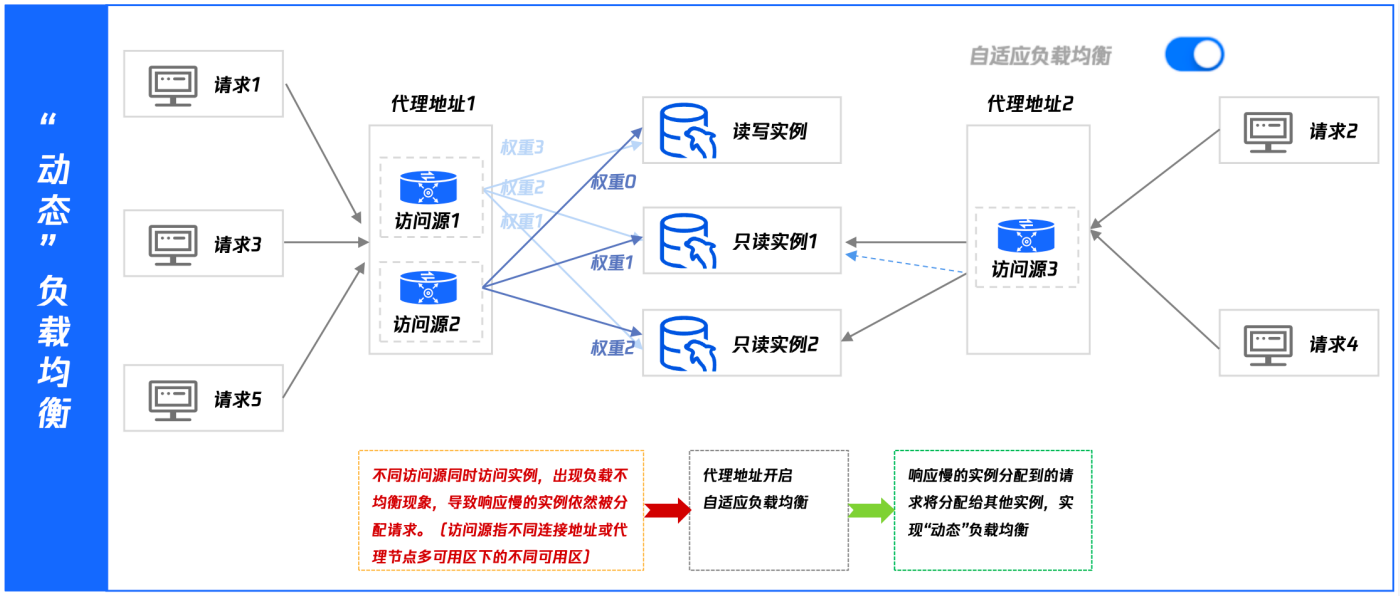
自适应负载均衡功能

最近更新时间：2024-08-08 16:38:51

云数据库 MySQL 数据库代理提供自适应负载均衡功能，可以根据后端的负载情况，将请求路由到负载更低的实例上，以达到数据库实例负载均衡，保证数据库实例响应速度的效果。

背景信息

在数据库的使用场景中，当使用了多个只读实例时，通常需要使用数据库代理进行负载均衡，使多个只读实例可以共同分担请求，以减轻主实例的负载压力。当前云数据库 MySQL 数据库代理已支持通过对各个只读实例分配读权重来实现负载均衡，这是一种“静态”的负载均衡能力，它能确保各个只读实例按照设置的读权重分配到对应的请求数。
如果有不同的业务从不同的数据库代理连接地址同时访问实例时，可能会出现负载不均衡的现象，而这种现象通过“静态”的负载均衡能力可能无法解决。
例如，用户具备1个主实例和2个只读实例（RO1、RO2），其中 RO2 被两个数据库代理连接地址使用，当 RO2 负载升高时，由于这种“静态”的负载均衡能力会使得被分配到的请求数是固定的，所以即使 RO2 的响应变慢，也会被继续分配请求，这将会拖慢整体的响应速度。因此，云数据库 MySQL 数据库代理提供自适应负载均衡功能，当发现 RO2 的请求变慢，负载过高时，数据库代理就会减少分配到 RO2 的请求，并将这些请求分配给其他只读实例，从而实现“动态”的负载均衡，确保数据库实例的整体响应速度。



前提条件

- 实例状态为运行中，且实例为双节点或三节点架构。
- 已 开通数据库代理 。
- 数据库代理内核小版本1.3.8及以上，如不满足请参见 升级数据库代理内核小版本 。
- 数据库内核版本要求如下，如不满足请参见 升级数据库内核小版本 。
  - MySQL 5.7内核小版本须20211030及以上。
  - MySQL 8.0内核小版本须20211202及以上。

操作场景

如果您的实例整体负载较高，出现某个只读实例被打满，而其他实例未被打满的情况，这时开启自适应负载均衡的代理会优先把请求发到仍能响应请求的实例上，这样可以充分利用所有实例的资源。请注意，如果是不合理的慢 SQL 导致的实例被打满，那么自适应负载均衡可能会让所有实例都出现高负载的情况，这种情况下仍需要解决慢 SQL 的问题。
如果您的全部实例规格一致，且所有实例配置的权重一致，需要均分流量时，可以直接选择开启自适应负载均衡，能够适应几乎所有场景的需求。
如下情况，不建议您开启自适应负载均衡功能：

1. 仅有一个数据库代理连接地址，建议使用“静态”的负载均衡能力，不建议开启自适应负载均衡功能。
2. **开启自适应负载均衡功能后，低负载情况下请求会不按设置的读权重进行路由**，如果您希望严格按照设置的读权重进行路由，则不建议开启自适应负载均衡功能。

3. 对于慢 SQL 引起单个只读实例打满的情况，无法通过负载均衡来缓解，此时不建议开启自适应负载均衡功能，建议优先通过优化 SQL，或者增加实例规格来缓解此问题。

注意事项

- 更改负载均衡策略只对更改配置后的新连接生效。如果有存量连接，可能需要执行 重新负载均衡 操作使旧连接生效。
- “动态”的负载均衡只能对经过数据库代理的流量生效，如果有不通过数据库代理而直连数据库的流量，可能会导致负载均衡策略效果不稳定。

开启/关闭自适应负载均衡

> **说明：**
> 以下步骤只针对开启/关闭自适应负载均衡操作，如需详细了解调整配置下的具体设置和说明，包括调整读权重的规则和影响说明，请参见 修改访问策略 。

1. 登录 MySQL 控制台 ，在上方选择地域，然后单击目标实例 ID，进入实例管理页。
2. 在实例管理页，选择**数据库代理** > **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**。

3. 在调整配置页面，根据实际需要，开启或关闭**自适应负载均衡**后的按钮，单击**确定**。

其他功能
Hint 语法使用

最近更新时间：2025-06-20 15:36:52

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，例如，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 MySQL 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 MySQL 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，数据库代理内核小版本大于等于 1.1.3 的版本，支持 prepare。

目前支持四种 Hint：

- 指定到主实例执行：

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

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

-- 强制在主实例上执行更新
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，您可以在实例详情页获取目标实例的实例 ID，如下图所示。

数据库管理（DMC）
DMC 简介

最近更新时间：2024-10-18 14:14:22

概述

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。
目前支持云数据库 MySQL 、云数据库 MariaDB、云原生数据库 TDSQL-C 版、云数据库 Redis、TDSQL MySQL 版。

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。

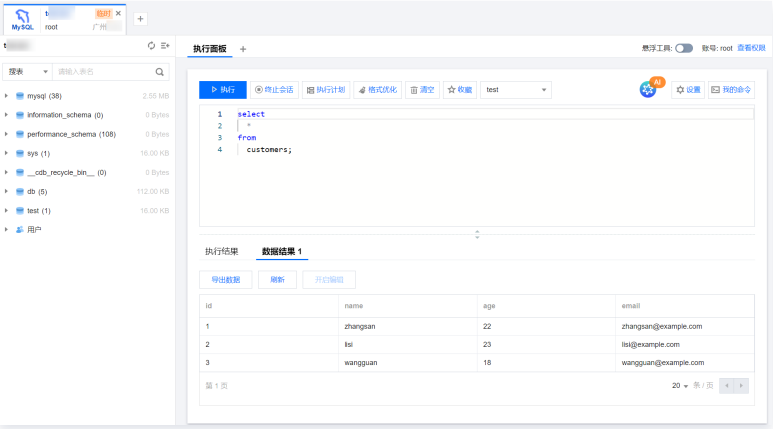
版权所有：腾讯云计算（北京）有限责任公司

DMC 管理

最近更新时间：2025-05-21 14:17:22

通过 DMC 控制台可以轻松实现数据库、表管理、执行 SQL 操作、数据导入导出等。

SQL 窗口展示



相关操作

功能	操作指引
SQL 窗口介绍	用户可通过界面查询和管理数据库数据，参见 SQL 窗口介绍 。
SQL 操作	提供界面化 SQL 编译器，参见 SQL 操作 。
库管理	新建、编辑和删除数据库，参见 库管理 。
表管理	新建表、管理表，参见 表管理 。
可编程对象管理	支持执行、编辑、删除可编程对象，参见 可编程对象管理 。
查看实时会话和 Kill 会话	支持查看数据库实时会话和 Kill 会话，参见 查看实时会话和 Kill 会话 。
数据导入或导出	SQL 窗口支持导入导出数据，参见 数据导入或导出 。

版权所有：腾讯云计算（北京）有限责任公司

账号管理
创建账号

最近更新时间：2025-05-23 09:25:21

操作场景

除系统默认创建的 root 账号外，您可根据业务的需要，通过云数据库 MySQL 控制台创建其他的业务账号。

> **说明：**
> - 仅单节点（云盘）实例、双节点/三节点本地盘实例、云盘版实例支持创建和管理账号，只读实例、灾备实例和只读分析引擎不支持创建及管理账号。
> - 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。

操作步骤

1. 登录 MySQL 控制台，在实例列表，单击**实例 ID** 或**操作**列的**管理**，进入实例管理页面。
2. 在实例管理页面，选择**数据库管理** > **账号管理**页，单击**创建账号**。

3. 在弹出的创建账号对话框，输入账号名、主机、密码、连接数限制和备注等信息，单击**确定**即可完成创建。
   - **账号名：**
     - MySQL 5.5、5.6 版本账号名需要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 `_` 。
     - MySQL 5.7、8.0 版本账号名需要1 – 32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）；多个主机地址以分隔符分隔，分隔符支持换行符、空格和 `;` `,` `|` 。
     - 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。
   - **密码：**密码最小长度需要8 – 64个字符，推荐12个字符以上，至少包含英文、数字和符号 _+-&=!@#$%^*() 中的2种，支持设置密码复杂度以提升数据库安全性，请参见 设置密码复杂度 。
   - **连接数限制：**该账号连接数限制，必须小于等于10240，若不填写则不进行额外限制（均会受到最大连接数限制）。root 账号默认受最大连接数10240的限制。
   - **备注：**为账号输入备注信息，最多可输入255个字符。
4. 数据库账号创建成功后，可在当前实例的数据库账号列表中，对其进行管理。

相关 API

API 名称	描述
CreateAccounts	创建云数据库的账号

设置密码复杂度

最近更新时间：2025-04-30 10:00:43

云数据库 MySQL 支持设置密码复杂度，提升数据库访问密码的强度，保障数据库的安全性。

前提条件

- 数据库版本为：
  - MySQL 5.6，小版本20201231及以上。
  - MySQL 5.7，小版本20201231及以上。
  - MySQL 8.0，小版本20201230及以上。
- 实例架构为双节点、三节点、云盘版。

注意事项

通过 MySQL 控制台创建账号设置密码或重置账号密码时，密码复杂度设置策略无法突破以下初始账号密码限制：

- 长度在8 – 64个字符以内。
- 特殊字符为 `_+-&=!@#$%^*()` 。

开启密码复杂度

> **说明：**
> 开启密码复杂度功能后，新创建账号设置密码或重置账号密码时按照新密码复杂度策略执行密码设置。

在购买页创建实例时开启

1. 登录 MySQL 购买页 。
2. 根据需要配置各项参数，在**密码复杂度**参数项后，选择**开启**。

3. 选择开启后，完成以下设置。

参数	说明
小写和大写的最小字符数	设置范围为1 – 16个字符，默认值为1
数字字符的最小字符数	设置范围为1 – 16个字符，默认值为1
特殊字符的最小字符数	设置范围为1 – 16个字符，默认值为1
密码最小字符数	设置范围为8 – 64个字符，默认值为8，且最小值须大于以上三个参数的最小字符数之和

在控制台对存量实例开启

1. 登录 MySQL 控制台 ，在实例列表，单击实例 ID 或**操作**列的**管理**，进入实例管理页面。
2. 在实例管理页面，选择**数据库管理** > **账号管理**页，单击**密码复杂度**（默认关闭）。

3. 在密码复杂度弹窗下选择开启，完成以下参数设置，单击**确定**。

参数	说明
小写和大写的最小字符数	设置范围为1 – 16个字符，默认值为1
数字字符的最小字符数	设置范围为1 – 16个字符，默认值为1
特殊字符的最小字符数	设置范围为1 – 16个字符，默认值为1
密码最小字符数	设置范围为8 – 64个字符，默认值为8，且最小值须大于以上三个参数的最小字符数之和

关闭密码复杂度

1. 登录 MySQL 控制台，在实例列表，单击实例 ID 或**操作**列的**管理**，进入实例管理页面。
2. 在实例管理页面，选择**数据库管理** > **账号管理**页，单击**密码复杂度**。

3. 在密码复杂度弹窗下选择**关闭**，单击**确定**。

相关文档

- 创建账号
- 重置密码

重置密码

最近更新时间：2024-10-18 14:14:22

操作场景

云数据库 MySQL 支持重置实例密码，如您在使用云数据库 MySQL 过程中，忘记数据库账号密码或需修改密码，可通过控制台重新设置密码。

> **说明：**
> - 云数据库 MySQL 的重置密码功能已纳入 CAM 权限管理，建议对重置密码接口或云数据库 MySQL 实例敏感资源权限收紧，只授权给应该授权的人员。
> - 为了数据安全，建议您定期更换密码，最长间隔不超过3个月。

操作步骤

1. 登录 MySQL 控制台，在实例列表，单击实例 ID 或**操作**列的**管理**，进入实例管理页面。
2. 在实例管理页面，选择**数据库管理** > **账号管理**页，找到需要重置密码的账号，单击**重置密码**或选择**更多** > **重置密码**。

3. 在重置密码对话框，输入新密码和确认密码，单击**确定**。

> **说明：**
> 数据库密码规格需要8 – 64个字符，至少包含英文、数字和符号 _+-&=!@#$%^*() 中的2种。

相关 API

API 名称	描述
ModifyAccountPassword	修改云数据库账号的密码

修改账号权限

最近更新时间：2024-10-18 14:14:22

操作场景

您可以在云数据库 MySQL 控制台授权管理已有的数据库账号，可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。

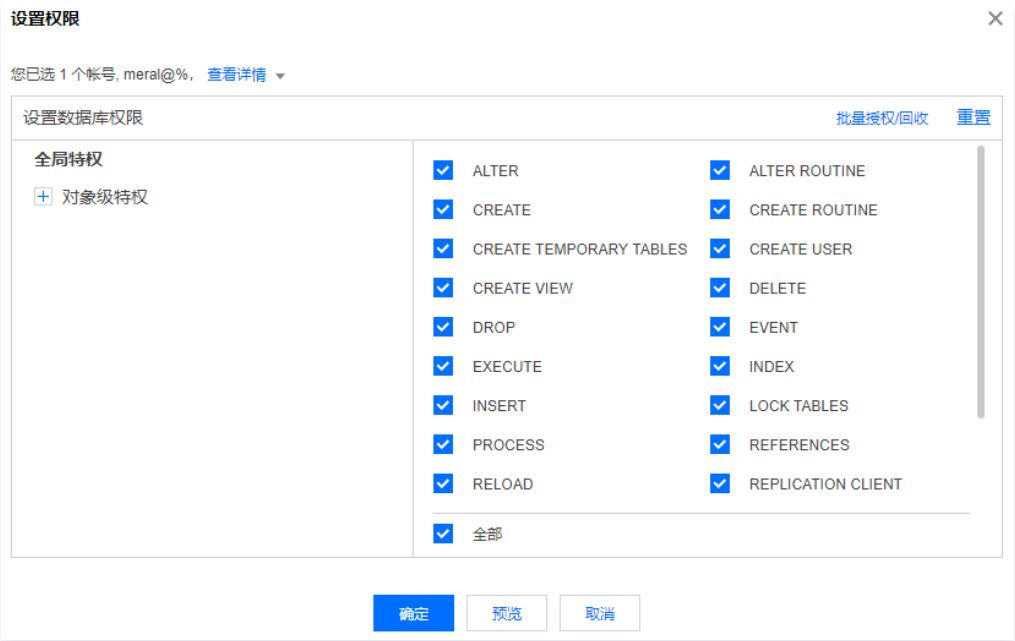
账号授权说明

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。
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 MySQL 控制台，在实例列表，单击**实例 ID** 或**操作**列的**管理**，进入实例管理页面。
2. 在实例管理页面，选择**数据库管理** > **账号管理**页，找到需要修改权限的账号，单击**修改权限**。

3. 在弹出的设置权限对话框，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。



相关 API

API 名称	描述
ModifyAccountPrivileges	修改云数据库实例账号的权限

修改授权访问的主机地址

最近更新时间：2025-04-01 15:18:52

操作场景

您可以通过云数据库 MySQL 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。

前提条件

已创建数据库账号，请参见 创建账号 。

主机地址优先级说明

数据库账号所授权的主机地址（即 IP 网段）存在优先级别，主机地址写入越精确，优先级越高。例如：在同一实例下，账号 A 授权的主机地址为192.167.1.1，另一个同名账号 A 授权的主机地址为192.167.1.%，则主机地址为192.167.1.%的账号将无法登录。温馨提示，在实际应用中，不建议配置多个同名账号并配置不同主机。

操作步骤

1. 登录 MySQL 控制台，在实例列表，单击**实例 ID** 或**操作**列的**管理**，进入实例管理页面。
2. 在实例管理页面，选择**数据库管理** > **账号管理**页，找到需要修改主机的账号，选择**更多** > **修改主机**。

> **说明：**
> root 账号不支持修改主机地址。

3. 在弹出的修改主机对话框，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。

修改连接数

最近更新时间：2025-04-01 15:18:52

云数据库 MySQL 支持修改账号的连接数（每个账号到数据库的最大连接数），以防止单个账号耗尽所有到云数据库 MySQL 的连接。
本文为您介绍如何通过控制台修改数据库账号的连接数。

前提条件

已创建数据库账号，请参见 创建账号 。

操作步骤

1. 登录 MySQL 控制台，在实例列表，单击**实例 ID** 或**操作**列的**管理**，进入实例管理页面。
2. 在实例管理页面，选择**数据库管理** > **账号管理**页，找到需要修改连接数的账号，选择**更多** > **修改连接数**。

> **说明：**
> root 账号不支持修改连接数，受最大连接数10240的限制。

3. 在弹窗下输入新连接数限制，单击**确定**。

> **说明：**
> 连接数可输入的范围应小于等于10240。

相关 API

API	描述
修改账号最大可用连接数	接口（ModifyAccountMaxUserConnections）用于修改云数据库账号最大可用连接数。

删除账号

最近更新时间：2024-10-18 14:14:22

操作场景

若您需要对创建的数据库账号进行停用，可以通过云数据库 MySQL 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

1. 登录 MySQL 控制台，在实例列表，单击**实例 ID** 或**操作**列的**管理**，进入实例管理页面。
2. 在实例管理页面，选择**数据库管理** > **账号管理**页，找到需要删除的账号，选择**更多** > **删除账号**。

3. 在弹出的删除账号对话框，单击**确定**。

相关 API

API 名称	描述
DeleteAccounts	删除云数据库的账号

修改备注

最近更新时间：2025-02-25 16:21:52

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。

前提条件

已创建数据库账号，请参见 创建账号 。

操作步骤

1. 登录 MySQL 控制台 ，在实例列表，单击**实例 ID** 或**操作**列的**管理**，进入实例管理页面。
2. 在实例管理页面，选择**数据库管理** > **账号管理**页，找到需要修改备注的账号，选择**更多** > **修改备注**。

> **说明：**
> root 账号不支持修改备注。

3. 在弹窗下，输入新的备注信息，单击**确定**。

> **说明：**
> 备注信息最多可输入255个字符。

相关 API

API	描述
修改云数据库实例账号的备注信息	接口（ModifyAccountDescription）用于修改云数据库账号的备注信息。

参数配置
设置实例参数

最近更新时间：2025-03-19 15:51:22

您可以通过 MySQL 控制台 查看和修改部分参数，并可以在控制台查询参数修改记录。

> **说明：**
> 主实例和只读实例均支持通过控制台修改参数，操作方法一致，可参见下文进行操作。

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用户可以修改的参数。
- 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。
- 如果希望恢复为默认公式，清空输入的参数内容并应用。

通过参数列表修改参数

批量修改参数

1. 登录 MySQL 控制台，在实例列表，单击**实例 ID** 或**操作**列的**管理**，进入实例管理页面。
2. 选择**数据库管理** > **参数设置**页，单击**批量修改参数**。

   主实例界面如下：

   只读实例界面如下：

3. 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**确认修改**。

4. 在弹出的对话框，选择参数任务的**执行方式**，单击**确定**。

> **说明：**
> - 若选择**立即执行**，所选实例的参数变更任务会立即执行并生效。
> - 若选择**维护时间内**，所选实例的参数变更任务会在实例的 维护时间 内执行并生效。

修改单个参数

1. 登录 MySQL 控制台，在实例列表，单击**实例 ID**，进入实例管理页面。
2. 选择**数据库管理** > **参数设置**页，选择目标参数所在行，在**参数运行值**列，单击 修改参数值。

3. 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。

4. 在弹出的对话框，选择参数任务的**执行方式**，单击**确定**。

> **说明：**
> - 若选择**立即执行**，所选实例的参数变更任务会立即执行并生效。
> - 若选择**维护时间内**，所选实例的参数变更任务会在实例的 维护时间 内执行并生效。

通过导入参数模板修改参数

方式一：通过参数设置页面导入

1. 登录 MySQL 控制台，在实例列表，单击**实例 ID**，进入实例管理页面。
2. 选择**数据库管理** > **参数设置**页，单击**自定义模板**（若之前没有设置过常用的自定义模板，可通过 MySQL 控制台，左侧导航栏的参数模板进入找到自定义模板，单击创建模板即可预先设置参数模板，之后即可通过第2步从自定义模板导入）。

3. 在弹出的对话框，选择参数模板，单击**导入并覆盖原有参数**。

4. 确认参数后，单击**确认修改**。

5. 在弹出的对话框，选择参数任务的**执行方式**，单击**确定**。

> **说明：**
> - 若选择**立即执行**，所选实例的参数变更任务会立即执行并生效。
> - 若选择**维护时间内**，所选实例的参数变更任务会在实例的 维护时间 内执行并生效。

方式二：通过导入参数配置文件修改参数

1. 登录 MySQL 控制台，在实例列表，单击**实例 ID**，进入实例管理页面。
2. 选择**数据库管理** >**参数设置**页，单击**导入参数**。

3. 单击**选择文件**找到需要的参数文件，然后单击**导入并覆盖原有参数**。

4. 确认参数后，单击**确认修改**。
5. 在弹出的对话框，选择参数任务的执行方式，单击**确定**。

> **说明：**
> - 若选择**立即执行**，所选实例的参数变更任务会立即执行并生效。
> - 若选择**维护时间内**，所选实例的参数变更任务会在实例的 维护时间 内执行并生效。

方式三：通过参数模板页面导入

请参见 应用参数模板于实例 。

恢复为默认模板

1. 登录 MySQL 控制台 ，在实例列表，单击**实例 ID**，进入实例管理页面。
2. 选择**数据库管理** > **参数设置**，单击**默认模板**，可选择**高稳定性模板**或**高性能模板**，之后单击**导入并覆盖原有参数**。

3. 单击**确认修改**，跳转至参数修改确认窗口。

4. 在弹出的对话框，选择参数任务的执行方式，阅读和勾选**重启规则**，单击**确定**即可。

> **说明：**
> - 若选择**立即执行**，所选实例的参数变更任务会立即执行并生效。

- 若选择**维护时间内**，所选实例的参数变更任务会在实例的 维护时间 内执行并生效。

参数公式

实例的参数设置支持使用公式，将与实例规格相关的参数设置为公式，当实例规格发生变更时，此处设置的参数值会动态变化，对于变更后的规格仍然适用，使实例始终保持业务运行所需的最佳状态。
参考参数 innodb_buffer_pool_size 的设置：{DBinitMemory*786432}，当实例规格中 DBinitMemory 变更时，此处的参数配置无需修改，innodb_buffer_pool_size 的值将会自动变更。

表达式语法的相关支持详见下表。

支持类别	支持说明	样例
变量	DBinitMemory：实例规格的内存大小，按MB单位取值，整数型。例如，实例规格的内存大小为1000MB，则 DBinitMemory 的值为1024。 DBInitCpu：实例规格的 CPU 核数，整数型。 例如，实例规格为8核，则 DBInitCpu 的值为8。	{DBinitMemory * 786432} 即：内存大小（DBinitMemory）* 百分比（系统默认为75%）* 1024 * 1024（单位换算）
运算符	公式语法：使用 `{}` 包裹。 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。不支持小数，例如系统支持 {MIN(DBInitMemory/4+500,1000000)}，不支持 {MIN(DBInitMemory*0.25+500,1000000)}。 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。不支持小数运算。	–
函数	函数 MAX()，返回整数型或者参数公式列表中最大的值。 函数 MIN()，返回整数型或者参数公式列表中最小的值。	{MAX(DBInitCpu/2,4)}

支持参数公式的参数

说明：
云数据库 MySQL 不断优化参数设置，以下仅列举部分支持参数公式的参数，您可在控制台参数模板下了解更多参数公式。

参数名称	参数描述	默认公式
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)}
table_open_cache_instances	指的是 MySQL 缓存 table 句柄的分区的个数。	{MIN(DBInitMemory/1000,16)}
table_open_cache	表描述符缓存大小，可减少文件打开/关闭次数。	{MAX(DBInitMemory*512/1000,2048)}
table_definition_cache	打开的表缓存实例的数量。	{MAX(DBInitMemory*512/1000,2048)}
max_connections	最大连接数。	{MIN(DBInitMemory/4+500,100000)}
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)}
innodb_write_io_threads	InnoDB 中用于写操作的 I/O 线程数。	{MAX(DBInitCpu/2,4)}

innodb_read_io_threads	InnoDB 中用于读操作的 I/O 线程数。	{MAX(DBInitCpu/2,4)}
innodb_buffer_pool_instances	InnoDB 缓冲池划分的区域数。	{MIN(DBInitMemory/2000,16)}
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。	{DBInitMemory*786432}

导出参数配置文件

1. 登录 MySQL 控制台，在实例列表，单击实例 ID，进入实例管理页面。
2. 选择**数据库管理** > **参数设置**页，单击**导出参数**。

导出参数配置为参数模板

1. 登录 MySQL 控制台，在实例列表，单击**实例 ID**，进入实例管理页面。
2. 选择**数据库管理** > **参数设置**页，单击**另存为模板**，可将现有参数配置存储为参数模板。

自定义时间修改参数

执行参数修改的最后一步时，在弹出的对话框，可自定义参数的修改时间。

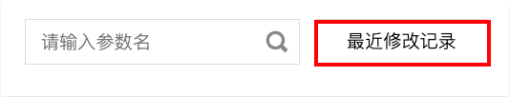
> **说明：**
> 选择**维护时间内**，所选实例的参数变更任务会在实例的 维护时间 内执行并生效。

取消参数修改任务

选择**维护时间内**的修改参数任务提交后，如需取消修改参数，可在任务执行前（即任务状态为**等待执行**），在左侧导航 任务列表 页，单击**操作**列的**撤销**，取消参数修改任务。

查看参数修改记录

1. 登录 MySQL 控制台，在实例列表，单击**实例 ID**，进入实例管理页面。
2. 选择**数据库管理** > **参数设置**页，单击右侧的**最近修改记录**。



3. 在最近参数修改记录页，可查看近期参数修改记录。

后续操作

- 您可以使用数据库参数模板来批量管理数据库的参数配置，请参见 使用参数模板 。
- 相关重要参数的配置建议，请参见 参数配置建议 。

使用参数模板

最近更新时间：2025-06-26 17:48:32

云数据库 MySQL 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，实现自定义参数设置。
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 云数据库 MySQL 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，包含高性能参数模板以及高稳定性模板。
- 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 MySQL 配置文件 my.cnf 导入生成模板。
- 支持从参数设置保存为模板。
- 单个或多个实例进行参数设置时，支持从模板导入。
- 支持两个参数模板之间进行对比。

> **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。

系统默认模板说明

云数据库 MySQL 针对不同系统对于数据库性能、稳定性上要求不同，分别提供了注重性能或稳定性的默认模板。

高稳定性参数模板

高稳定性模板注重数据安全以及运行稳定性，对比高性能参数模板，运行速度较慢。

高性能参数模板

高性能参数模板注重运行的性能，对比高稳定性参数模板，运行速度较快。

> **注意：**
> 目前高性能参数模板仅支持云数据库 MySQL 双节点和三节点的5.6、5.7、8.0版本。

涉及参数如下：

> **说明：**
> 参数模板中部分参数支持设置参数公式，关于参数公式的说明以及支持参数公式的参数，请参见 参数公式 。

差异参数名	默认模板(已废弃)	高性能参数模板	高稳定性模板
innodb_read_io_threads	12	{MAX(DBInitCpu/2,4)}	{MAX(DBInitCpu/2,4)}
innodb_write_io_threads	12	{MAX(DBInitCpu/2,4)}	{MAX(DBInitCpu/2,4)}
max_connections	10000	{MIN(DBInitMemory/4+500,100000)}	{MIN(DBInitMemory/4+500,100000)}
table_definition_cache	768	{MAX(DBInitMemory*512/1000,2048)}	{MAX(DBInitMemory*512/1000,2048)}
table_open_cache	2000	{MAX(DBInitMemory*512/1000,2048)}	{MAX(DBInitMemory*512/1000,2048)}
table_open_cache_instances	16	{MIN(DBInitMemory/1000,16)}	{MIN(DBInitMemory/1000,16)}
innodb_disable_sort_file_cache	OFF	OFF	ON
innodb_log_compressed_pag	ON	OFF	ON

es			
innodb_print_all_deadlocks	OFF	OFF	ON
sync_binlog	0	1000	1
thread_handling	one-thread-per-connection	pool-of-threads	one-thread-per-connection
innodb_flush_redo_using_fdatasync	TRUE	TRUE	TRUE
innodb_fast_ahi_cleanup_for_drop_table	ON	ON	ON
innodb_adaptive_hash_index	FALSE	TRUE	FALSE
innodb_table_drop_mode	ASYNC_DROP	ASYNC_DROP	ASYNC_DROP
innodb_flush_log_at_trx_commit	2	2	1

核心参数说明：

参数	取值	说明
innodb_flush_log_at_trx_commit	1	事务提交时，把事务日志从缓存区写到日志文件中，并且立刻写入到磁盘上。
	2	事务提交时，把事务日志从缓存区写到日志文件中，但不一定立刻写入到磁盘上。日志文件会每秒写入到磁盘，如果写入前系统崩溃，就会导致最后1秒的日志丢失。
innodb_flush_redo_using_fdatasync	true	使用 fdatasync 刷 redo log file。 fdatasync 相比 fsync 不刷元数据部分，对性能略有提升。
	false	使用 fsync 刷 redo log file。相比 fdatasync，对性能略有下降。
sync_binlog	1	事务提交后，将二进制日志文件写入磁盘并立即刷新，相当于同步写入磁盘，不经过系统缓存。
	1000	每写入1000次系统缓存就执行一次写入磁盘并刷新的操作，会有数据丢失的风险。
thread_handling	one-thread-per-connection	每个连接由一条后台线程处理。当有大量的会话存在时，会导致大量的资源争用，大量的线程调度和缓存失效也会导致性能下降。
	pool-of-threads	不同于以前的每个连接由一条后台线程处理，所有连接由一个线程池共同处理，高并发场景下性能更好。
innodb_flush_redo_using_fdatasync	FALSE	redo 日志文件采用 fsync 系统调用来落盘。
	TRUE	redo 日志文件采用 fdatasync 系统调用来落盘。在一些 redo 落盘场景可以避免总是刷文件元数据到存储设备上，来减少开销。
innodb_table_drop_mode	SYNC_DROP	InnoDB 将在 DROP TABLE 结束之前完成删除表。
	ASYNC_DROP	InnoDB 将重命名 ibd 文件到 innodb_async_drop_tmp_dir 并将其放入后台异步删除。
table_definition_cache	400 – 524288	可以存储在定义缓存中的表定义数。 如果使用大量表，可以创建大表定义缓存以加快表的打开速度。 与普通表缓存不同，表定义缓存占用更少的空间并且不使用文件描述符。
table_open_cache	1 – 524288	所有线程的打开表数。 增加这个值会增加 mysqld 需要的文件描述符的数量。

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。

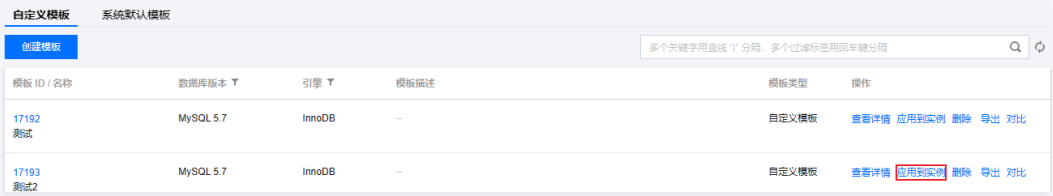
1. 登录 MySQL 控制台，在左侧导航选择**参数模板**页面，单击**创建模板**。

2. 在弹出的对话框，配置如下参数，单击**创建并设置参数**。
- **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。
- **数据库版本**：选择需要的数据库版本。
- **原始模板类型**：选择需要从哪一类默认模板作为创建模板。
- **模板描述**：输入参数模板的简要说明。

3. 创建完成后，您可以在模板详情页修改、导入、导出参数等。

应用参数模板于实例

1. 登录 MySQL 控制台，在左侧导航选择**参数模板**页面。
2. 在参数模板列表中，找到需要应用的模板，单击**应用到实例**。



3. 在弹出页面，选择参数的执行方式和实例，检查参数修改无误后，单击**提交**。
- **执行方式**：默认为**立即执行**；若选择**维护时间内**，所选实例的参数变更任务会在实例的 维护时间 内执行并生效。
- **MySQL 实例**：选择对应地域下，需要应用参数模板的实例。
- **参数对比**：可查看所选实例参数修改前、后的配置。

> **注意：**
> 应用参数模板到多个实例上时，请确认参数是否适用这些实例，请您谨慎操作。

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 MySQL 控制台，在左侧导航选择**参数模板**页面，单击**模板 ID** 或**操作**列的**查看详情**，进入模板详情页。
2. 在模板详情页，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**保存**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 MySQL 控制台，在左侧导航选择**实例列表**页面，单击**实例 ID** 进入管理页。
2. 选择**数据库管理** > **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**新建并保存**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 MySQL 控制台，在左侧导航选择**参数模板**页面，单击**模板 ID**，进入模板详情页。
2. 在模板详情页，单击**批量修改参数**、或在**参数当前值**列，单击 可修改参数值。

> **注意：**
> 选择**导入参数**，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要与 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，否则会提示导入不成功。

导入参数模板

1. 登录 MySQL 控制台，在左侧导航选择**参数模板**页面，单击**模板 ID**，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**，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要与 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，否则会提示导入不成功。

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

方式一

1. 登录 MySQL 控制台，在左侧导航选择**参数模板**页面。
2. 在参数模板列表，在所需模板的**操作**列，单击**导出**。

方式二

1. 登录 MySQL 控制台，在左侧导航选择**参数模板**页面，单击**模板 ID**，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 MySQL 控制台，在左侧导航选择**参数模板**页面。
2. 在参数模板列表，在所需模板的**操作**列，单击**删除**。
3. 在弹出的对话框，单击**确定**。

后续操作

相关重要参数的配置建议，请参见 参数配置建议 。

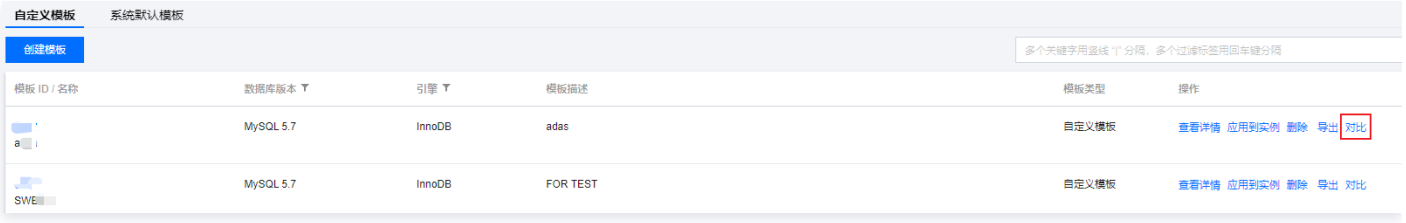
对比不同模板参数

最近更新时间：2024-10-18 14:14:22

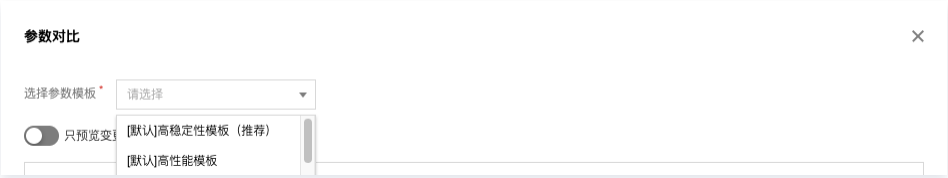
您可以通过 MySQL 控制台对比不同自定义参数模板之间的差异或自定义参数模板与系统默认模板之间的差异，查看参数模板之间的不同。

操作步骤

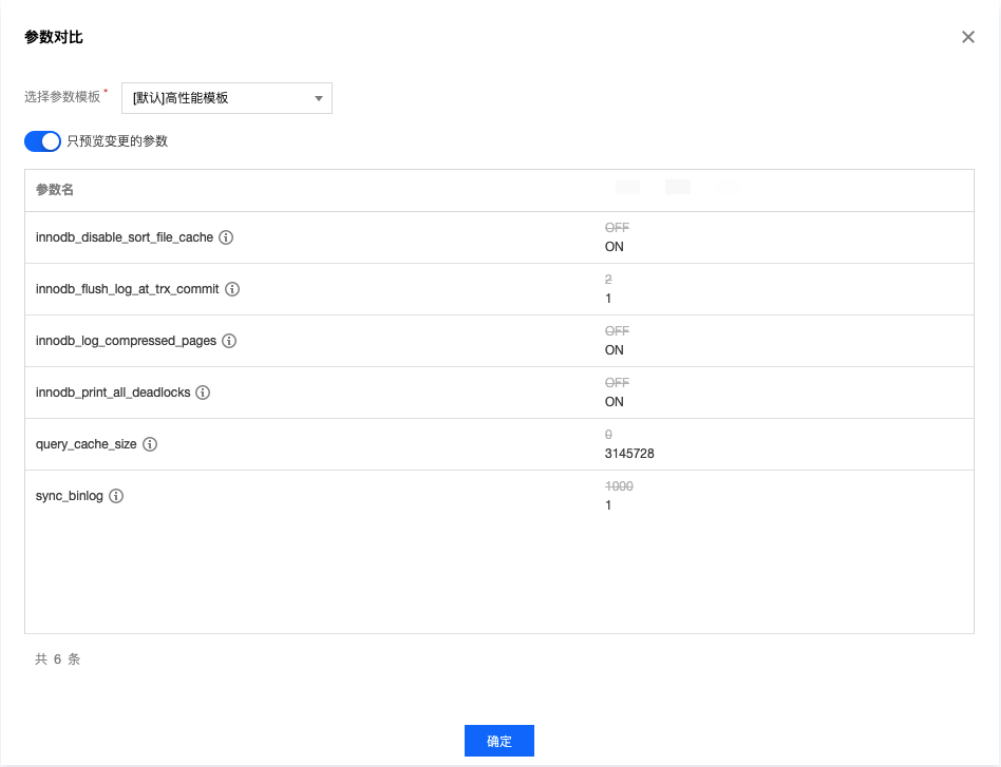
1. 登录 MySQL 控制台，在左侧导航选择**参数模板**页面，选择需要对比的自定义模板，单击**对比**。



2. 在弹出的对话框，通过**选择参数模板**选择需要进行对比的目标模板。



3. 选择后查看参数模板对比结果。



参数模板性能对比

最近更新时间：2025-05-21 14:17:22

测试工具

数据库基准性能测试为 sysbench 1.0.20。

安装工具

本文测试使用的是 Sysbench 1.0.20 版本，安装方法如下：

```
git clone https://github.com/akopytov/sysbench.git
cd sysbench
git checkout 1.0.20
yum install gcc gcc-c++ autoconf automake make libtool bzr mysql-devel git mysql
./autogen.sh
./configure
make -j
make install
```

> **说明：**
> 以上是在压测 CVM（CentOS 系统）上的安装方法，如需安装到其他操作系统，请参见 Sysbench 官方文档。

测试环境

类型	说明
测试实例规格	本次测试选择常用的三种规格进行测试，分别为4核8GB内存、8核32GB内存、16核128GB内存
客户端配置	64核128GB内存
客户端内网带宽	23Gbps
测试数据量	数据库实例内存 * 1.2
测试数据库实例版本	5.6 20210630、5.7 20210630、8.0 20210330

- 客户端规格说明：机器采用了较高配置的客户机器，保证单客户端可以压测出数据库实例的性能，如果客户端配置规格较小，建议采用多个客户并行压测实例来求取数据总和。
- 网络延时说明：测试环境保证客户端机器与数据库实例在同一可用区，测试结果不受网络环境影响。

测试方法

测试数据准备

```
sysbench --db-driver=mysql --mysql-host=xxxx --mysql-port=xxxx --mysql-user=xxxx --mysql-password=xxxx --
mysql-db=sbtest --table_size=xxxx --tables=xxxx --events=0 --time=600 --threads=xxxx --percentile=95 --
report-interval=1 oltp_read_write prepare
```

性能压测命令

```
sysbench --db-driver=mysql --mysql-host=xxxx --mysql-port=xxxx --mysql-user=xxxx --mysql-password=xxxx --
mysql-db=sbtest --table_size=xxxx --tables=xxxx --events=0 --time=600 --threads=xxxx --percentile=95 --
report-interval=1 oltp_read_write run
```

性能压测参数说明：

- `oltp_read_write`，表示调用 /usr/share/sysbench/oltp_read_write.lua 脚本进行 oltp 模式测试。

- `--tables=xxxx`，表示本次用于测试的表数量。
- `--table_size=xxxx`，表示本次测试使用的表行数。
- `--threads=xxxx`，表示本次测试的客户端连接并发数。
- `--report-interval=1`，表示每秒输出一次测试结果。
- `--percentile=95`，表示设定采样比例，默认是95%。
- `--time=600`，表示本次测试的执行时间，600代表600秒。

场景模型

本文用例均使用 sysbench 的 lua 脚本。
针对常用配置类型，对不同的参数模板进行性能测试，测试结果见下文测试结果。

测试结果

5.6 20210630版本

CPU(核)	内存(GB)	threads	测试时长	模板	SysBench QPS	SysBench TPS	avg_lat
4	8	32	10分钟	默认模板(废弃)	34428.69	1721.43	18.59ms
				高性能参数模板	35917.50	1795.87	17.82ms
				高稳定性模板	34834.04	1741.70	18.37ms
8	32	64	10分钟	默认模板(废弃)	61210.19	3060.51	20.91ms
				高性能参数模板	67719.55	3385.98	18.90ms
				高稳定性模板	64910.09	3245.50	19.72ms
16	128	128	10分钟	默认模板(废弃)	106965.44	5348.27	23.93ms
				高性能参数模板	127955.48	6397.77	20.00ms
				高稳定性模板	119509.02	5975.45	21.41ms

5.7 20210630版本

CPU(核)	内存(GB)	threads	测试时长	模板	SysBench QPS	SysBench TPS	avg_lat
4	8	32	10分钟	默认模板(废弃)	34428.69	1721.43	18.59ms
				高性能参数模板	35917.50	1795.87	17.82ms
				高稳定性模板	34834.04	1741.70	18.37ms
8	32	64	10分钟	默认模板(废弃)	61210.19	3060.51	20.91ms
				高性能参数模板	67719.55	3385.98	18.90ms
				高稳定性模板	64910.09	3245.50	19.72ms
16	128	128	10分钟	默认模板(废弃)	106965.44	5348.27	23.93ms
				高性能参数模板	127955.48	6397.77	20.00ms
				高稳定性模板	119509.02	5975.45	21.41ms

8.0 20210330版本

CPU(核)	内存(GB)	threads	测试时长	模板	SysBench QPS	SysBench TPS	avg_lat

4	8	32	10分钟	默认模板(废弃)	32594.79	1629.74	19.63ms
				高性能参数模板	33383.77	1669.19	19.17ms
				高稳定性模板	32071.90	1603.60	19.95ms
8	32	64	10分钟	默认模板(废弃)	65718.22	3285.91	19.47ms
				高性能参数模板	70195.37	3509.77	18.23ms
				高稳定性模板	60704.69	3035.23	21.08ms
16	128	128	10分钟	默认模板(废弃)	132023.66	6601.18	19.38ms
				高性能参数模板	151021.67	7551.08	16.95ms
				高稳定性模板	132391.01	6619.55	19.33ms

版权所有：腾讯云计算（北京）有限责任公司

参数配置建议

最近更新时间：2024-10-18 14:14:22

云数据库 MySQL 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。云数据库 MySQL 提供4种字符集，分别为 LATIN1、UTF8、GBK、UTF8MB4，其中 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。

    > **说明：**
    > sql_mode 参数的调整为全局生效。但需注意，其参数值中 ONLY_FULL_GROUP_BY 配置修改后，如果旧连接是长连接，则还是 session 级别的，因此，only_full_group_by 配置修改后针对新建连接生效，旧连接不一定生效，需断开重连。

  - `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：1

- 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1s。当某个查询执行时间为1s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。
- 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。

智能参数调优

最近更新时间：2025-04-30 10:00:43

本文为您介绍通过 MySQL 控制台 实现智能参数调优。

背景

“深度学习”这个词已经进入了公众的视线。时至今日，相关技术也比较成熟，腾讯云数据库团队也在思考如何借助深度学习的方式来提升数据库的运行效率。首先想到的就是数据库的参数调优。由于业务系统的千差万别，也无法像优化 SQL 一样在细粒度下进行针对性的调优，是令数据库管理者头痛的难题，往往需要借助经验去构筑一套相对较为优异的参数模板。数据库参数调优能力也是专家级数据库管理者的专属技能。
2019-2021年之间，腾讯云数据库团队连续发表2篇论文，分别为《Automatic Database Tuning using Deep Reinforcement Learning》和《An Online Cloud Database Hybrid Tuning System for Personalized Requirements》，并申请了国际专利。现在，基于论文将这一理论研发为一种可用的系统，在真实场景中通过调整数据库参数来提升数据库性能。
为什么需要数据库参数调优服务：

- **参数非常多：**例如 MySQL，有几百个配置项，调优难度大。
- **人力成本高：**需要专职 DBA，依靠专家经验，人时成本高。
- **工具普适性：**现存工具功能有限，耗时久效果一般。
- **云上新需求：**部分用户没有专职运维团队，参数调优很难实现。

前提条件

具备一个状态为运行中的 MySQL 实例。

使用限制

- 云盘版实例暂不支持使用智能参数调优功能。
- 场景智能调优每月有次数限制，每月每个实例可进行3次调优，从每月1日开始重置调优次数。
- AI 智能分析每月有次数限制，每月每个实例可进行1次调优，从每月1日开始重置调优次数（AI 智能分析暂未发布，敬请期待）。
- 实例的 CPU 须为4核及以上，才能使用智能参数调优功能。
- 智能调参任务列表仅会保留最近15次的调参结果。
- 销毁/退还实例或实例到期时，如有智能参数调优任务处于进行中，则该任务会自动中止并删除该任务。
- 一个实例仅支持同时运行一个调优任务，无法发起多个调优任务。
- 智能参数调优功能目前暂时仅支持北京、上海、广州地域，后续将会逐步增加。

操作步骤

已购 MySQL 实例

1. 登录 MySQL 控制台，在上方选择地域，在实例列表，单击实例 ID 或**操作**列的**管理**，进入实例管理页面。
2. 在实例管理页，选择**数据库管理** > **参数设置** > **智能参数调优**。

3. 在智能调优弹窗，选择**场景智能调优**或者**AI 智能分析**的参数调优方式，配置好后单击**开始分析**。

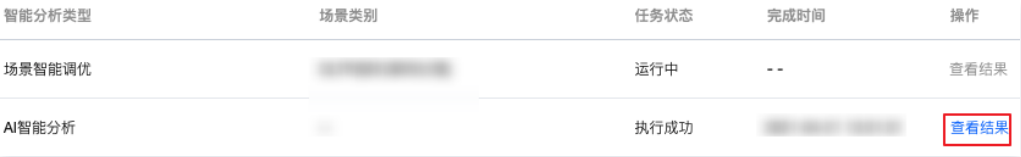
- 如您选择**场景智能调优**方式，之后步骤如下：
  场景智能调优：根据选定的应用场景进行智能分析，效率更高，更具针对性。
  单击**场景选择**下拉键选择业务场景，有三种场景（交易类场景、OLTP 性能测试、压力测试）可供选择。
  选择对应场景后，您可自定义该场景下的业务占比，便于系统更精确地进行分析。配置完成后单击**开始分析**。
  - 交易类场景（TPCC）
    - 定制内容：订单业务(高)、支付业务(高)、订单查询(低)、物流业务(低)、仓储业务(低)
    - 数据读取方式：全缓存(默认)、磁盘读写。
    - 并发度：低、中、高(默认)。
  - OLTP 性能测试（Sysbench）
    - 定制内容：读取业务比例(高)，写入业务比例(默认无)
    - 数据读取方式：全缓存(默认)、磁盘读写。
    - 并发度：低、中、高(默认)。
  - 压力测试（myslap）
    - 并发度：低、中、高(默认)

- 如您选择场景为 **AI 智能分析方式**（此功能暂未发布，敬请期待），之后操作步骤如下：
  AI 智能分析：通过对数据库运行指标进行深度分析，确定数据库业务类型，再通过深度学习算法对不同参数在确定场景下进行性能分析，并给出参数设置建议。
  选择 AI 智能分析 之后，单击**开始分析**。

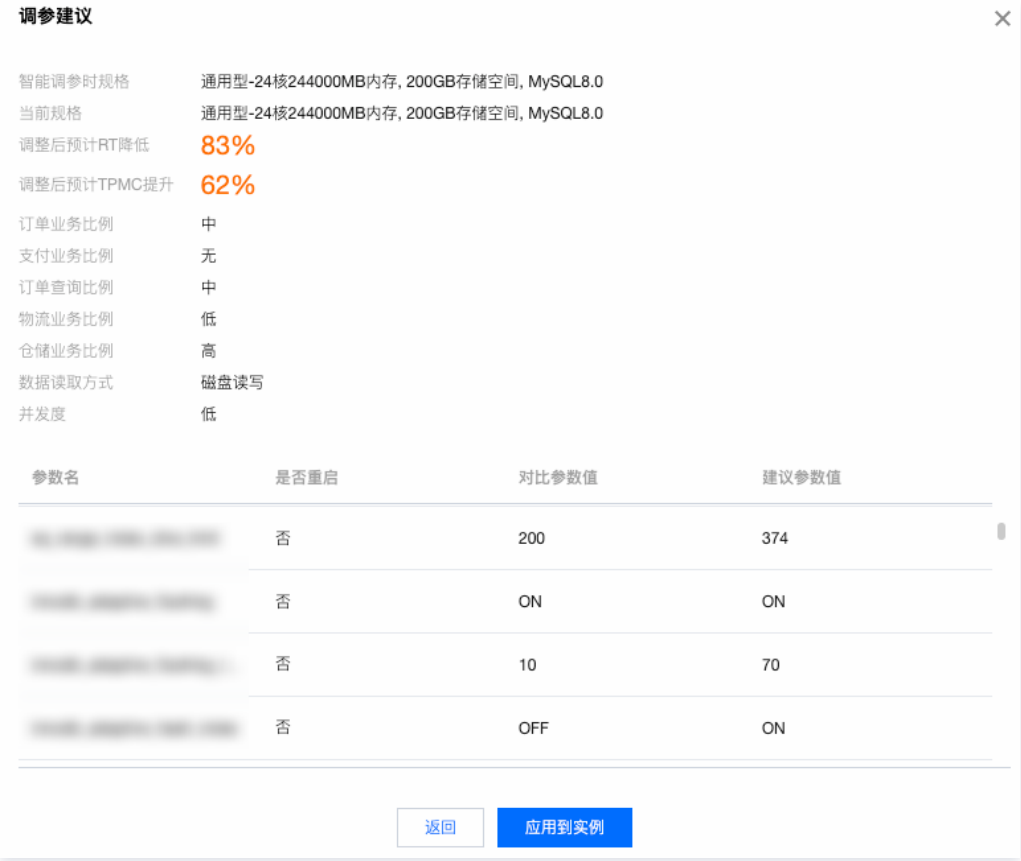
> **注意：**
> - AI 智能分析功能目前正在完善中，暂未发布，敬请期待。
> - AI 智能分析使用了深度学习算法以及大数据分析能力，分析耗时较长，建议在业务低峰期进行。

4. 开始分析后，参数调优任务即进行中，您可在参数设置页面选择**智能参数调优** > **查看任务**了解任务详情。

5. 参数调优任务结束后，在**智能参数调优** > **查看任务**里，单击**操作**列的**查看结果**。



6. 确认参数调优建议后，单击**应用到实例**。



7. 在弹窗下确认参数变更，选择执行方式，阅读并勾选重启规则，单击**确定**。
   执行方式：
   - 立即执行：确定后立刻应用到实例。
   - 维护时间内：在维护时间内将此变更应用到实例，维护时间可在实例详情页修改。

新购 MySQL实例

购买 MySQL 实例时，您可在选定参数模板后，选择是否开启**场景智能调优**。使用场景智能调优后，系统会根据您选择的参数模板以及对应的业务场景进行二次调整，业务场景有三种，包括：交易类场景、OLTP 性能测试、压力测试。
对应修改结果您可以在**数据库管理** > **参数设置** > **场景参数调优** > **查看任务**中查看。

参数模版
[默认]高稳定性模板（推荐）
新建参数模版
场景智能调优
交易类场景
使用场景智能调优后，在应用参数模板后再根据选择的对应业务场景进行二次调整
了解更多

系统参数模板变更记录

最近更新时间：2025-05-21 14:17:22

云数据库 MySQL 提供系统参数模板，用于批量参数设置，系统参数模板的参数可能随版本迭代而优化更新，本文为您介绍系统参数模板中参数的变更记录。

说明：
- 系统参数模板的参数变更，并不会影响已使用过该参数模板的数据库实例。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。
- 使用系统参数模板，请参见 使用参数模板 。

2024年12月

参数名称	MySQL 5.6	MySQL 5.7	MySQL 8.0	变更说明
innodb_stats_persistent_sample_pages	✓	✓	✓	涉及模板：高稳定性、高性能 涉及引擎：InnoDB **变更前** 默认值：20 **变更后** 默认值：200

2024年11月

参数名称	MySQL 5.6	MySQL 5.7	MySQL 8.0	变更说明
explicit_defaults_for_timestamp	✓	✓	✓	涉及模板：高稳定性、高性能 涉及引擎：InnoDB、RocksDB **变更前** 是否重启生效：是 **变更后** 是否重启生效：否
innodb_page_cleaners	–	✓	✓	涉及模板：高稳定性、高性能 涉及引擎：InnoDB **变更前** 默认值：4 **变更后** 默认值：8

2024年09月

参数名称	MySQL 5.7	MySQL 8.0	变更说明
innodb_table_drop_mode	✓	✓	涉及模板：高稳定性 涉及引擎：InnoDB **变更前** 默认值：SYNC_DROP **变更后** 默认值：ASYNC_DROP

2024年03月

参数名称	MySQL 5.6	MySQL 5.7	MySQL 8.0	变更说明
slave_parallel_workers	✓	✓	✓	涉及模板：高性能/高稳定性 涉及引擎：InnoDB/RocksDB **变更前** • 参数范围：[0 | 1 | 2 | 4 | 8 | 16 | 32 | 64]

				• 默认值：0 **变更后** • 参数范围：[0-64] • 默认值：{MIN(DBInitCpu*2,64)}

2024年01月

参数名称	MySQL 5.6	MySQL 5.7	MySQL 8.0	变更说明
thread_pool_oversubscribe	✓	✓	✓	MySQL 5.6、5.7、8.0高性能参数模板里 thread_pool_oversubscribe 参数默认值变更 • 变更前默认值：10 • 变更后默认值：16

2023年12月

参数名称	MySQL 5.6	MySQL 5.7	MySQL 8.0	变更说明
thread_pool_oversubscribe	✓	✓	✓	thread_pool_oversubscribe 参数默认值变更 • 变更前默认值：3 • 变更后默认值：10

2023年08月

参数名称	MySQL 5.7	MySQL 8.0	变更说明
innodb_fast_ahi_cleanup_for_drop_table	✓	✓	innodb_fast_ahi_cleanup_for_drop_table 参数默认值变更 • 变更前默认值：OFF • 变更后默认值：ON

2023年06月

参数名称	MySQL 5.5	MySQL 5.6	MySQL 5.7	MySQL 8.0	变更说明
time_zone	✓	✓	✓	✓	time_zone 参数默认值变更 • 变更前默认值：SYSTEM • 变更后默认值：+8:00

2023年05月

参数名称	MySQL 5.7	MySQL 8.0	变更说明
innodb_async_truncate_work_enabled	✓	–	5.7 版本异步删表功能开关参数变更 由 innodb_async_truncate_work_enabled 变更为 innodb_table_drop_mode。（2023年05月15日）
innodb_parallel_ddl	–	✓	参数下线（2023年05月05日）

2023年03月

参数名称	MySQL 5.7	MySQL 8.0	变更说明
innodb_temp_data_file_path	✓	✓	innodb_temp_data_file_path 参数默认值变更 • 变更前默认值：ibtmp1:12M:autoextend

			• 变更后默认值：ibtmp1:12M:autoextend:max:1048576M

2023年01月

参数名称	MySQL 5.7	MySQL 8.0	变更说明
max_execution_time	✓	✓	max_execution_time 参数支持动态设置 • 是否需要重启：否 • 默认值：0 • 参数可修改值：0 – 4294967295 • 单位：毫秒

2022年08月

参数名称	MySQL 5.7	MySQL 8.0	变更说明
innodb_buffer_pool_size	✓	✓	innodb_buffer_pool_size 参数支持动态设置 • 是否需要重启：否 • 默认值：{DBInitMemory * 786432} • 参数可修改值：{DBInitMemory * 524288} – {DBInitMemory * 943718} • DBinitMemory 为实例规格的内存大小取整数

2022年07月

参数名称	MySQL 5.7	MySQL 8.0	变更说明
innodb_temp_data_file_path	✓	✓	支持 innodb_temp_data_file_path（临时表空间大小）参数修改，参数属性如下： • 是否需要重启：是 • 默认值：ibtmp1:12M:autoextend • 参数可修改值：ibtmp1为12 – 1024MB，勾选 autoextend 后 max 可设置的最大值为2097152MB

2022年03月

参数名称	MySQL 5.6	MySQL 5.7	MySQL 8.0	变更说明
innodb_open_files	✓	✓	✓	参数下线
innodb_stats_sample_pages	–	✓	✓	参数下线
wait_timeout	✓	✓	✓	参数范围值更新为：1 – 31536000
thread_cache_size	✓	✓	✓	参数范围值更新为：1 – 16384

2021年12月

参数名称	MySQL 5.6	MySQL 5.7	MySQL 8.0	变更说明
binlog_row_image	✓	✓	✓	参数默认值统一为 FULL，需注意早期创建的实例此参数默认值为 MINIMAL，支持手动修改。

2020年11月

参数名称	MySQL 8.0	变更说明
innodb_flush_log_at_trx_commit	✓	参数新增
sync_binlog	✓	参数新增
local_infile	✓	参数新增
innodb_log_file_size	✓	参数新增
cdb_recycle_bin_enabled	✓	参数新增
binlog_format	✓	参数范围值更新为：row
innodb_autoinc_lock_mode	✓	参数默认值更新为：2
table_open_cache	✓	参数默认值更新为：2000
slave_pending_jobs_size_max	✓	参数默认值更新为：1073741824
time_zone	✓	参数范围值更新为：[SYSTEM|-12:00|-11:00|-10:00|-09:00|-08:00|-07:00|-06:00|-05:00|-04:00|-03:00|-02:00|-01:00|+00:00|+01:00|+02:00|+03:00|+04:00|+05:00| +05:30|+06:00|+06:30|+07:00|+08:00|+09:00|+10:00|+11:00|+12:00|+13:00]
max_connections	✓	参数范围值更新为：1 – 100000
slave_rows_search_algorithms	✓	参数默认值更新为：TABLE_SCAN,INDEX_SCAN,HASH_SCAN
innodb_open_files	✓	参数默认值更新为：10240
slave_parallel_type	✓	参数范围值更新为：LOGICAL_CLOCK|TABLE|DATABASE

2020年08月

参数名称	MySQL 5.6	MySQL 5.7	变更说明
log_warnings	✓	✓	参数新增
innodb_flush_log_at_trx_commit	✓	✓	参数新增
sync_binlog	✓	✓	参数新增
local_infile	✓	✓	参数新增
innodb_log_file_size	✓	✓	参数新增
binlog_format	✓	✓	参数范围值更新为：row
innodb_autoinc_lock_mode	✓	✓	参数默认值更新为：2
innodb_open_files	✓	✓	参数范围值更新为：1 – 102400
table_open_cache	✓	✓	参数默认值更新为：2000
slave_pending_jobs_size_max	✓	✓	参数默认值更新为：1GB
time_zone	✓	✓	参数范围值更新为： [SYSTEM|-12:00|-11:00|-10:00|-09:00|-08:00|-07:00|-06:00|

			-05:00|-04:00|-03:00|-02:00|-01:00|+00:00|+01:00|+02:00|+03:00|+04:00|+05:00| +05:30|+06:00|+06:30|+07:00|+08:00|+09:00|+10:00|+11:00|+12:00|+13:00]
max_connections	✓	✓	参数范围值更新为：1 - 100000
cdb_more_gtid_feature_supported	-	✓	内核功能全量
cdb_more_gtid_feature_supported	✓	-	参数默认值更新为：OFF
slave_parallel_workers	-	✓	内核功能全量
tls_version	-	✓	参数下线
slave_rows_search_algorithms	✓	✓	参数默认值更新为：TABLE_SCAN,INDEX_SCAN,HASH_SCAN
innodb_open_files	✓	✓	参数默认值更新为：10240

2020年08月

参数名称	MySQL 5.5	变更说明
innodb_autoinc_lock_mode	✓	参数默认值更新为：TABLE_SCAN,INDEX_SCAN,HASH_SCAN
innodb_open_files	✓	参数默认值更新为：10240

备份与回档
备份数据库

最近更新时间：2025-07-02 16:41:31

为防止数据丢失或损坏，您可以使用自动备份或手动备份的方式来备份数据库。

备份简介

备份方式

云数据库 MySQL 单节点（云盘）、双节点（本地盘）、三节点（本地盘）、云盘版（云盘）支持**自动备份**和**手动备份**两种方式来备份数据库。

备份类型

云数据库 MySQL 双节点、三节点支持两种备份类型：

- **物理备份**：物理数据全拷贝（自动备份支持）。
- **逻辑备份**：SQL 语句的备份（手动备份和自动备份均支持）。

> **说明：**
> - 物理备份恢复方式需要用 xbstream 解包，详细操作请参见 使用物理备份恢复数据库 。
> - 单个实例的表数量超过30万后，可能会造成备份失败，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30万。
> - 备份时长与数据库数据文件大小以及表数量成正相关，数据文件越大或表数量越多，备份时长会越长。
> - 由于 Memory 引擎表的数据存储在内存中，因此无法对 Memory 引擎表进行物理备份，为避免丢失数据，建议将 Memory 引擎表转换成 InnoDB 表。
> - 实例存在大量无主键表，可能会造成备份失败，同时会影响实例的高可用性，请及时对无主键的表创建主键或者二级索引。
> - 进行**手动备份** > **逻辑冷备**操作会有全局读锁，建议您通过控制台进行此操作（控制台发起逻辑冷备任务是对备机的备份，不会影响主库）。

物理备份的优势	逻辑备份的劣势
• 备份速度快。 • 支持流式备份和压缩。 • 备份成功率高。 • 恢复简单并且高效。 • 依赖备份进行的耦合操作，例如，增加 RO、增加灾备会变得更快。 • 物理备份完成的平均时长为逻辑备份的八分之一左右。 • 物理备份的导入速度比逻辑备份导入快十倍左右。	• 恢复时需要执行 SQL 和构建索引，恢复时间长。 • 备份速度慢，数据量大的情况下尤为明显。 • 备份过程中会对实例造成压力，可能会加大主从延迟。 • 有可能会丢失浮点数的精度信息。 • 由于各种问题（错误视图等），可能导致备份失败。 • 依赖备份进行的耦合操作，例如，增加 RO、增加灾备会变得缓慢。

云数据库 MySQL 单节点（云盘）、云盘版（云盘）支持快照备份：
快照备份：通过对存储层磁盘创建快照的方式进行备份（自动备份与手动备份均支持）。

> **说明：**
> 单节点（云盘）实例为单个节点，进行备份是在主节点上发起的，如需进行备份，建议在业务低峰期进行操作。

快照备份的优势	快照备份的劣势
• 备份速度快。 • 相对占用体积小。	不支持下载。

备份对象

数据备份	日志备份
• MySQL 双节点、三节点： ○ 自动备份支持全量物理备份。	数据库的 binlog 日志文件备份支持 MySQL 单节点（云盘）、双节点、三节点、云盘版（云盘）：

- 手动备份支持全量物理备份、全量逻辑备份和单库单表逻辑备份。
- 自动备份与手动备份均支持压缩和下载。

- MySQL 单节点（云盘）、云盘版（云盘）：
  - 自动备份支持全量快照备份。
  - 手动备份支持全量快照备份。
  - 自动备份和手动备份均不支持下载。

- 日志文件占用实例备份空间。
- 日志文件支持下载，但不支持压缩。
- 可设置日志文件保留时长。

注意事项

- 云数据库 MySQL 的自动备份于2019年02月26日起，仅支持物理备份。自动备份设置默认方式为物理备份，不再提供逻辑备份。存量自动备份为逻辑备份的实例会陆续被自动切换为物理备份。
  此切换不会影响您业务访问，可能会影响您自动备份的使用习惯。若您需要逻辑备份，可以使用 云数据库 MySQL 控制台 中的手动备份方式或者通过 API 调用 来生成逻辑备份。
- 手动备份默认备份方式为逻辑冷备，若在调整配置前进行的全量备份选择为逻辑冷备，则调整配置的耗时可能较长，若您希望减少调整配置的耗时，建议您在调整配置前进行全量备份时选择为物理冷备。
- 实例备份文件占用备份空间，请合理使用备份空间，超出免费额度的备份空间会进行收费，请参见 备份空间收费说明 。
- 建议选择业务低峰期进行备份。
- 为避免所需备份文件超出保留时间被删除，请及时下载所需的备份文件至本地。
- 备份期间禁止 DDL 操作，避免锁表导致备份失败。
- MySQL 只读实例不支持数据库备份。

自动备份 MySQL 数据

设置自动备份

1. 登录 MySQL 控制台 ，在实例列表，单击**实例 ID** 进入管理页面，选择**备份恢复** > **自动备份设置**。

2. 在弹出的备份设置对话框，选择各备份参数，单击**确定**。参数说明如下：

> **说明：**
> - 回档功能 基于备份周期和备份保留天数内的数据备份 + 日志备份（binlog），缩短自动备份频率和保留天数会影响实例数据的回档时间范围，请您权衡备份配置。
> 例如，设置备份周期为周一、周四，保留天数为7天，则可以回档到7天内（数据备份和日志有效备份实际存储时长）的任意时刻。
> - 自动备份无法手动删除，可设置备份保留时间，到期后会自动删除。
> - 增加数据备份和日志备份保留的天数将可能带来额外的备份空间计费费用。
> - 缩短日志备份保留的天数可能影响实例的数据回档周期。

自动备份设置中，数据备份的设置支持开启定期保留，未开启定期保留的设置叫常规备份设置，以下分别介绍**常规备份设置**和**开启定期备份设置**中的参数说明。

常规备份设置说明

参数	说明
备份开始时间	• 默认时间：系统自动分配的备份开始时间范围（0:00 – 12:00）。 • 自定义：支持自定义选择备份开始的时间范围（如02:00 – 06:00），建议设置为业务低峰期。 **说明：** • 备份开始时间只是备份任务进入调度队列的时间。例如，选择：02:00 – 06:00开启备份，系统会在02:00 – 06:00时间范围内的某一个时间点发起备份任务，具体的发起时间点取决于后端备份策略和备份系统状况。为确保备份任务不对实例运行产生影响，备份任务可能会因为队列导致延迟发起。 • 备份任务消耗时间与数据量成正比，最长不超过24小时。
数据备份保留时间	• MySQL 双节点、三节点、云盘版数据备份文件可以保留7天 – 1830天，默认为7天，到期后备份集自动删除。 • MySQL 单节点（云盘）数据备份文件可以保留7天 – 30天，默认为7天，到期后备份集自动删除。
备份周期	配置规则： • 按每周设置：默认选中星期一至星期日7天，支持自定义选择备份时间，但为了保障您的数据安全，请设置一周至少备份两次。 • 按每月设置：为了保障您的数据安全，一个月中，任意两个相邻的号数之间间隔不能超过2天，例如选择了1号备份，则下次备份日期不能跳过2、3、4号选择5号。 **说明：** 选择按每月设置时，为避免出现连续多日不备份的情况，不能跳过以下日期排列： 27/28/1（日）、28/29/1（日）、29/30/1（日）、28/1/2（日）、29/1/2（日）、30/1/2（日）。
备份落冷（可选）	勾选相应的数据备份落冷策略并指定天数： • 指定标准存储天数：即设置数据备份文件生成多少天后，沉降为标准存储。 • 指定归档存储天数：即设置数据备份文件生成多少天后，沉降为归档存储。如需详细了解有关备份落冷的说明和策略，请参见 设置备份落冷 。其中，归档存储功能目前未开放，敬请期待。 **说明：** 单节点（云盘）、云盘版实例暂不支持设置备份落冷。
日志备份保留时间	• MySQL 双节点、三节点、云盘版日志备份文件可以保留7天 – 3650天，默认为7天，到期后备份集自动删除。 • MySQL 单节点（云盘）日志备份文件可以保留7天 – 30天，默认为7天，到期后备份集自动删除。
备份落冷（可选）	勾选相应的 binlog 备份落冷策略并指定天数： • 指定标准存储天数：即设置 binlog 文件生成多少天后，沉降为标准存储。 • 指定归档存储天数：即设置 binlog 文件生成多少天后，沉降为归档存储。如需详细了解有关备份落冷的说明和策略，请参见 设置备份落冷 。其中，归档存储功能目前未开放，敬请期待。

> **说明：**
> 单节点（云盘）、云盘版实例暂不支持设置备份落冷。

开启定期备份设置说明

> **说明：**
> - 单节点（云盘）、云盘版实例暂不支持定期备份设置功能。
> - 定期备份保留时间只能大于常规备份设置的保留时间。

参数	说明
备份开始时间	• 默认时间：系统自动分配的备份开始时间范围（0:00 – 12:00）。 • 自定义：支持自定义选择备份开始的时间范围（如02:00 – 06:00），建议设置为业务低峰期。 **说明：** • 备份开始时间只是备份任务进入调度队列的时间。例如，选择：02:00 – 06:00开启备份，系统会在02:00 – 06:00时间范围内的某一个时间点发起备份任务，具体的发起时间点取决于后端备份策略和备份系统状况。为确保备份任务不对实例运行产生影响，备份任务可能会因为队列导致延迟发起。 • 备份任务消耗时间与数据量成正比，最长不超过24小时。
数据备份保留时间	MySQL 双节点、三节点数据备份文件可以保留7天 – 1830天，默认为7天，到期后备份集自动删除。

备份周期	配置规则： • 按每周设置：默认选中星期一至星期日7天，支持自定义选择备份时间，但为了保障您的数据安全，请设置一周至少备份两次。 • 按每月设置：为了保障您的数据安全，一个月中，任意两个相邻的号数之间间隔不能超过2天，例如选择了1号备份，则下次备份日期不能跳过2、3、4号选择5号。 **说明：** 选择按每月设置时，为避免出现连续多日不备份的情况，不能跳过以下日期排列： 27/28/1（日）、28/29/1（日）、29/30/1（日）、28/1/2（日）、29/1/2（日）、30/1/2（日）。
定期备份保留时间	数据备份文件可以保留90天 – 3650天，默认为1080天，超过保留期限后备份集自动删除。
定期备份保留策略	支持按每月、每季度或者每年设置备份保留个数。
起始日期	执行定期保留备份开始的时间。
备份落冷（可选）	勾选相应的数据备份落冷策略并指定天数： • 指定标准存储天数：即设置数据备份文件生成多少天后，沉降为标准存储。 • 指定归档存储天数：即设置数据备份文件生成多少天后，沉降为归档存储。如需详细了解有关备份落冷的说明和策略，请参见 设置备份落冷 。其中，归档存储功能目前未开放，敬请期待。
日志备份保留时间	日志备份文件可以保留7天 – 3650天，默认为7天，到期后备份集自动删除。
备份落冷（可选）	勾选相应的 binlog 备份落冷策略并指定天数： • 指定标准存储天数：即设置 binlog 文件生成多少天后，沉降为标准存储。 • 指定归档存储天数：即设置 binlog 文件生成多少天后，沉降为归档存储。如需详细了解有关备份落冷的说明和策略，请参见 设置备份落冷 。其中，归档存储功能目前未开放，敬请期待。

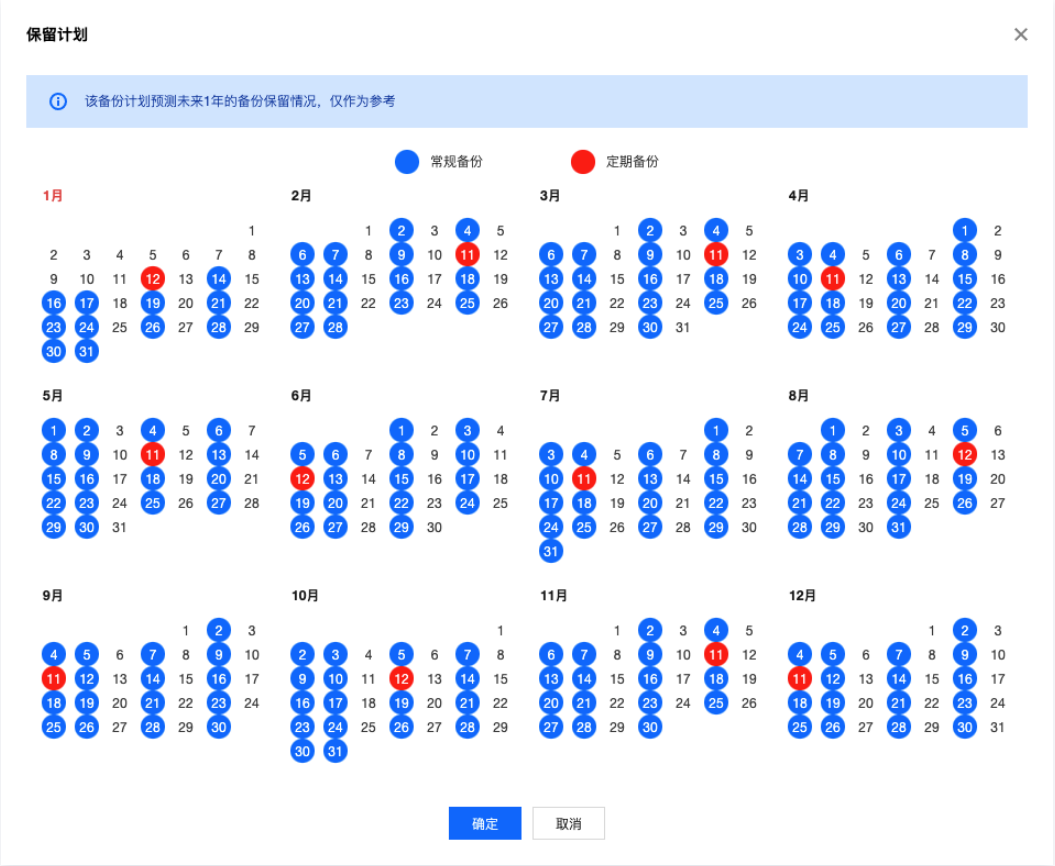
查看保留计划

> **说明：**
> 单节点（云盘）、云盘版实例暂不支持查看保留计划功能。

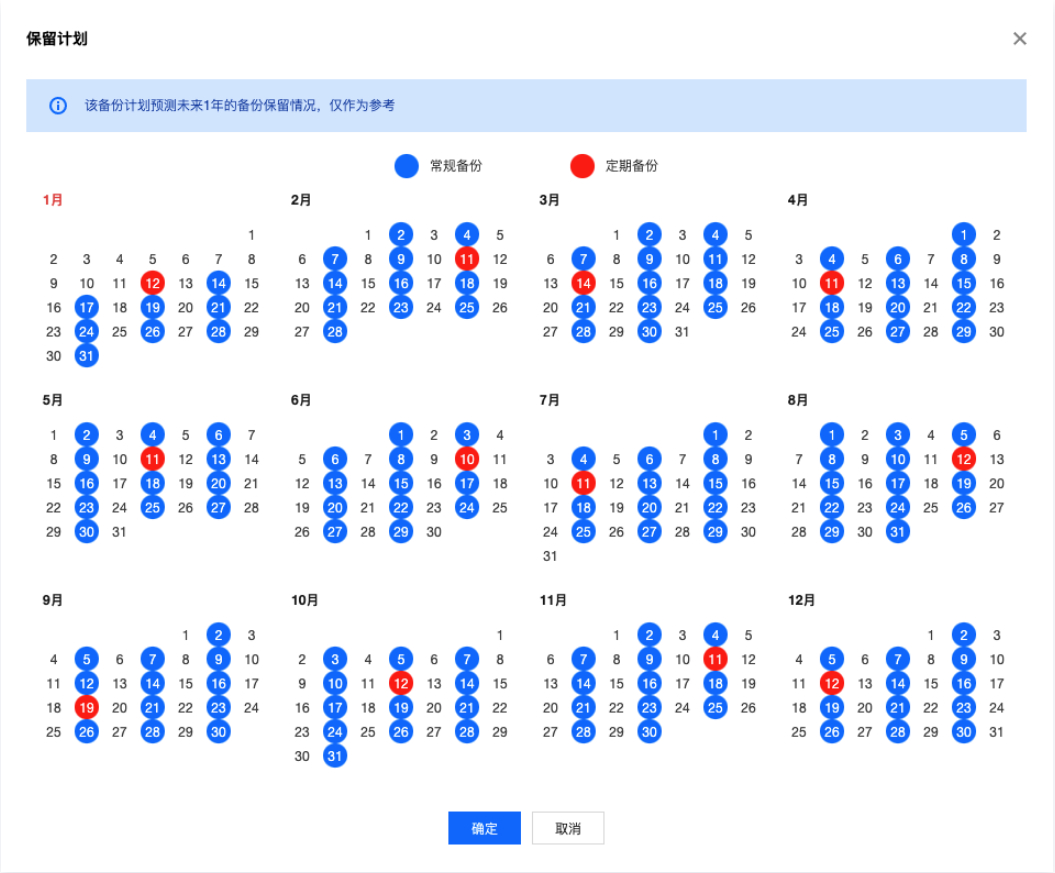
当您在备份设置中选择定期备份保留策略后，可单击**查看保留计划**进行预览。

- 蓝色日期表示常规备份的日期。
- 红色日期表示定期备份的日期。
- 可通过单击常规备份或者定期备份，隐藏对应日期颜色标注，便于预览。
- 备份计划预览暂时为未来1年的备份保留情况，仅作为参考。

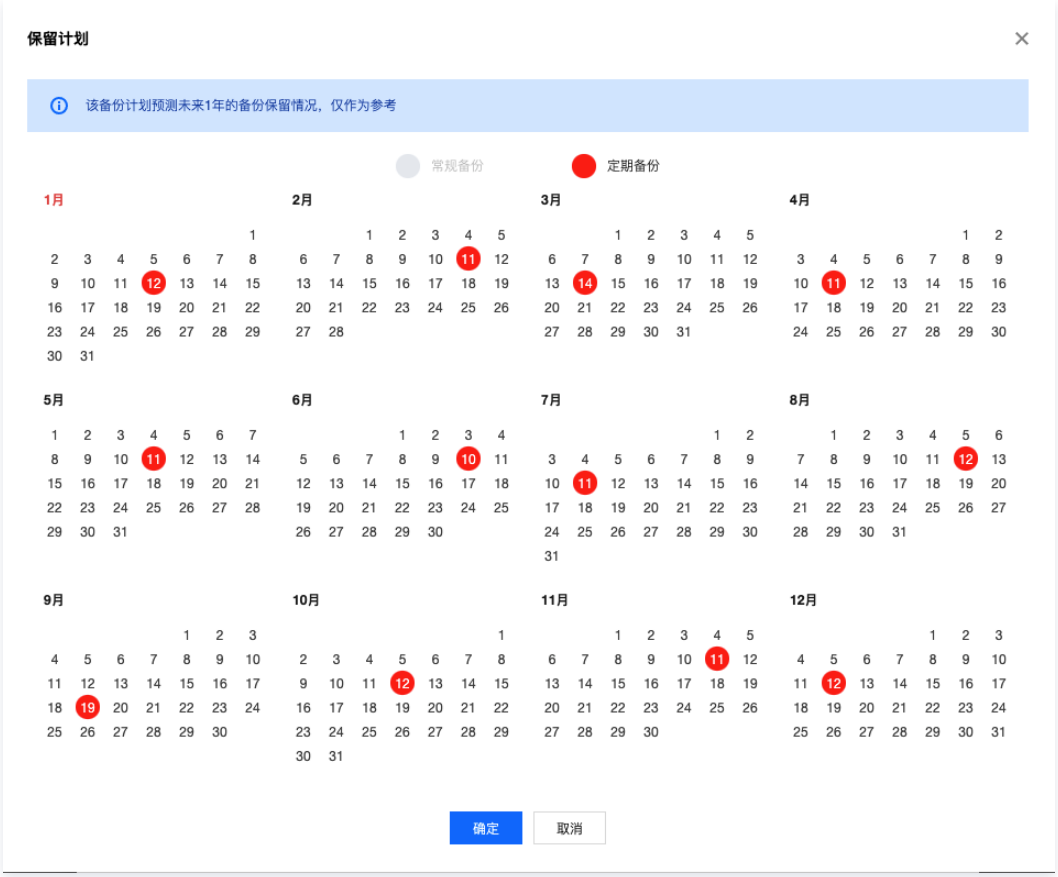
演示一：备份周期为周一、周三、周五、周日。自2022年1月11日起，每月保留1个备份。



演示二：备份周期为周一、周三、周五。自2022年1月11日起，每季度保留3个备份。



演示三：仅显示定期备份。



手动备份 MySQL 数据

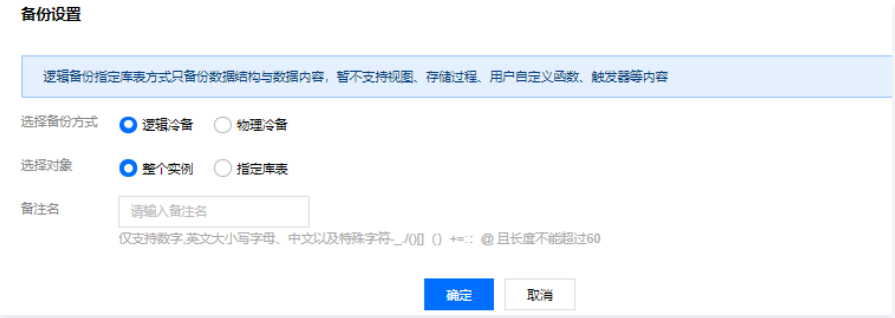
手动备份功能允许用户自助发起备份任务。

说明：

- MySQL 双节点、三节点实例手动备份支持全量物理备份、全量逻辑备份和单库单表逻辑备份。
- MySQL 双节点、三节点实例手动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，没有手动删除的前提下会一直保留至数据库实例下线。
- MySQL 单节点（云盘）实例手动备份支持全量快照备份。
- MySQL 单节点（云盘）实例手动备份不支持删除。
- 实例在执行每天的自动备份任务期间，无法发起手动备份。
- 手动备份会以最高优先级进入备份队列，具体备份开始时间请以备份列表的**任务开始时间**字段为准。

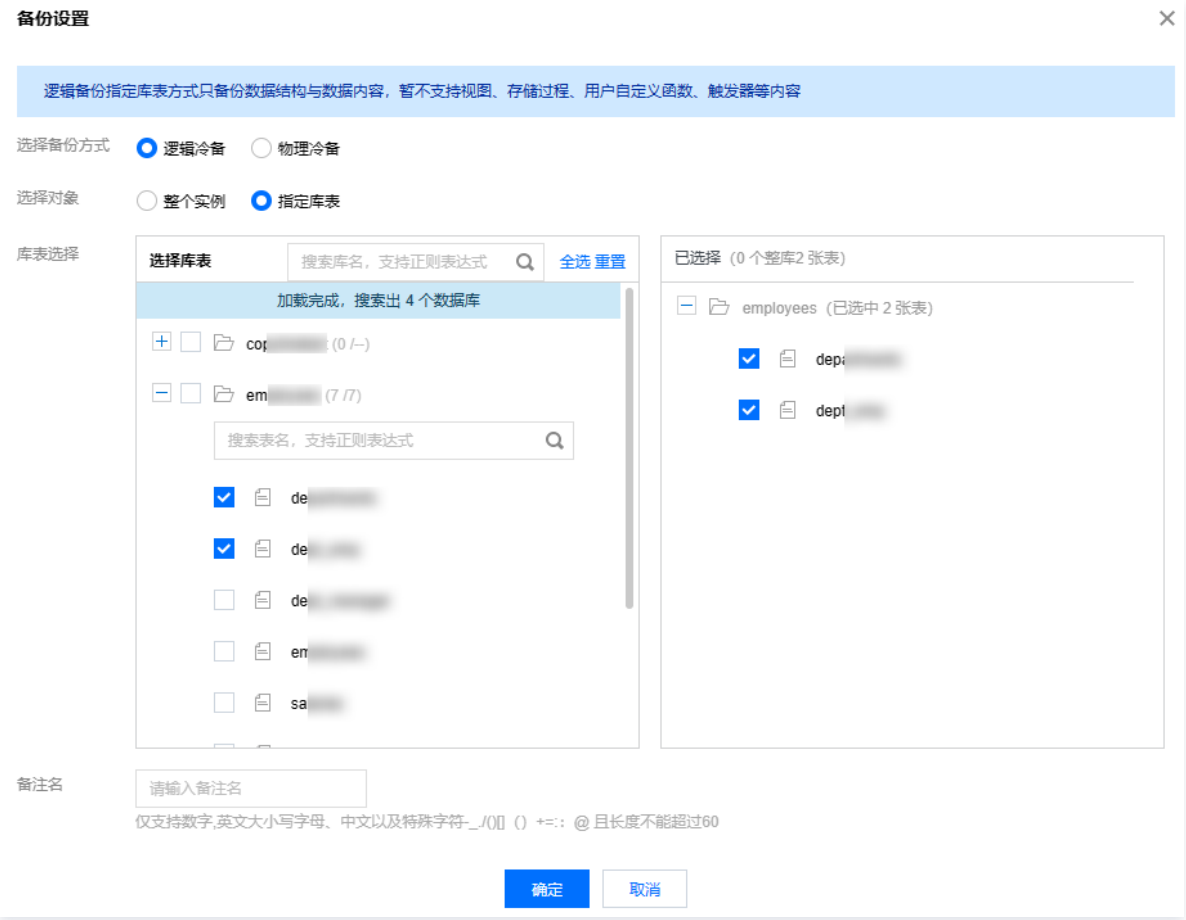
双节点、三节点实例操作步骤

1. 登录 MySQL 控制台，在实例列表，单击**实例 ID** 进入管理页面，选择**备份恢复** > **手动备份**。
2. 在弹出的备份设置对话框，选择备份方式和对象，填写备注名，单击**确定**。



说明：

- 如果是逻辑备份下的单库备份或单表备份，请在左侧**选择库表**里勾选要备份的数据库或数据表，将数据库或表加入右侧列表。若您还没有数据库或数据表，请先创建数据库或表。
- 手动备份默认备份方式为逻辑冷备，若在调整配置前进行的全量备份选择为逻辑冷备，则调整配置的耗时可能较长，若您希望减少调整配置的耗时，建议您在调整配置前进行全量备份时选择为物理冷备。



单节点（云盘）、云盘版（云盘）实例操作步骤

1. 登录 MySQL 控制台，在实例列表，单击**目标实例 ID** 进入管理页面，选择**备份恢复** > **手动备份**。
2. 输入备注名，单击**确定**。

热点问题

1. 超出备份保留时间的备份还可以下载或还原吗?

到期后的备份集会自动删除，无法进行下载还原。

- 建议您根据需求合理设置备份保留时间，或在 MySQL 控制台 下载备份文件至本地（需注意单节点云盘实例备份文件暂不支持下载）。
- 您也可以在控制台通过手动备份实例数据，手动备份会一直保存。

> **说明：**
> 手动备份亦会占用备份空间，请合理使用备份空间，避免造成额外的费用。

2. 备份可以手动删除吗?

- 自动备份无法手动删除，可设置备份保留时间，到期后会自动删除。
- 双节点、三节点实例手动备份可在 MySQL 控制台 的备份列表手动删除，没有手动删除的前提下会一直保留，单节点云盘实例手动备份暂不支持删除。

3. 数据和日志备份是否可以关闭?

不可以关闭。但可以通过 MySQL 控制台 减少备份频率和删除不再使用的手动备份数据来降低备份空间的占用量（单节点云盘实例手动备份暂不支持删除）。

4. 如何减少备份空间开销?

- 删除不再使用的手动备份数据（手动备份可在 MySQL 控制台 的实例管理页 > 备份恢复页面进行删除，需注意，单节点云盘实例手动备份暂不支持删除）。
- 降低非核心业务的数据自动备份频率（可在控制台调整备份周期和备份保留时间，一周至少备份2次）。

> **说明：**
> 回档功能 基于备份周期和备份保留天数内的数据备份 + 日志备份（binlog），缩短自动备份频率和保留天数会影响实例数据的回档时间范围，请您权衡备份配置。

- 缩短非核心业务的数据备份和日志备份保存时间（备份保留时间为7天已经能满足大多数场景需要）。
- 设置备份落冷，自定义备份落冷策略，转换备份文件存储类型，降低存储成本。

业务场景	备份保留时间
核心业务	建议7天 – 3650天，建议开启定期备份，实现定期将备份进行长期保存
非核心、非数据类业务	建议7天
归档业务	建议数据备份保留时间设置为7天，根据实际业务需求手动备份数据，用完及时删除
测试业务	建议数据备份保留时间设置为7天，根据实际业务需求手动备份数据，用完及时删除

设置备份落冷

最近更新时间：2025-07-07 19:42:31

云数据库 MySQL 支持将备份文件进行落冷处理，以降低备份的存储成本。本文为您介绍备份落冷的相关说明和设置。

> **说明：**
> - 云盘版实例暂不支持备份落冷功能。
> - 落冷备份无法使用赠送的免费空间。
> - 归档存储功能目前未开放，敬请期待。

简介

备份落冷：指将已生成的备份文件（包括数据备份文件和 binlog 备份文件）进行存储类型转换，以降低成本。云数据库 MySQL 会根据您设置的备份落冷规则在超过设置时间后自动进行备份文件存储类型转换。
在**自动备份 > 设置备份保留时间**后，备份文件将按照备份策略进行存储，在设置的保留时间内，您可以通过**备份落冷**将备份文件生成 N 天（支持自定义）后的备份沉降为标准存储，进一步还可以将备份文件生成 X 天（支持自定义）后的备份沉降为归档存储，这个过程称之为备份沉降。
备份文件存储类型定义

- 标准存储：适用于频繁访问和使用的业务场景，可用于回档、克隆，不占用免费空间，支持内网下载，暂不支持外网下载。
- 归档存储：适用于较低频率访问的业务场景，归档存储有最低90天的存储时间要求，若归档存储时间不足90天，按90天计算存储费用，不支持下载。

功能说明

- 支持设置备份文件生成指定天数后，降级为标准存储，指定天数最小值为30天，最大值不能超过设置的备份保留时间。
- 支持设置备份文件生成指定天数后，降级为归档存储，指定天数最小值为90天，最大值为设置的备份保留时间。
- 备份存储类型转换只能由热到冷（常规存储 > 标准存储 > 归档存储），不能反向进行。
- 对于任意时间下的备份落冷设置，落冷策略都将以北京时间（GMT+8）次日的0时为准开始执行。对于设置规则之后的备份，由于生命周期任务是异步执行，此时通常最晚于次日的24时前完成执行。时间是以备份文件转储的修改时间为标准开始计算。
- 备份落冷支持在数据文件备份以及 binlog 文件备份进行单独设置。
- 备份落冷支持配合定期保留使用。
- 标准存储的备份文件仅能通过内网下载，暂不支持通过外网下载。归档存储的备份文件内网和外网暂时均不支持下载。

计费说明

备份落冷后将采用更低的计费价格以降低存储成本，详细计费说明请参见 备份落冷计费说明 。

备份落冷策略说明

1. 当备份保留时间 ≤ 31天时，备份落冷无法设置，如需使用请将备份保留时间设置在31天以上。

2. 当31天 < 备份保留时间 ≤ 180天时，归档存储天数无法设置。

3. 标准存储天数不能大于或等于归档天数。

4. 标准存储天数和归档存储天数都只能小于备份保留时间。
5. 备份落冷规则勾选后生效，不勾选则不落冷至相应的存储中。

设置备份落冷

1. 登录 MySQL 控制台，在实例列表，单击**实例 ID** 进入管理页面，选择**备份恢复** > **自动备份设置**。

2. 在弹出的备份设置对话框，根据需要设置备份保留时间和周期后，或者设置日志备份保留时间后，勾选相应的备份落冷策略并指定天数。

3. 阅读和勾选备份空间收费说明，单击**确定**。

备份落冷设置示例

示例1：常规备份 + 备份落冷设置背景

企业 A 在广州地域有一个云数据库 MySQL 实例，其数据备份保留时间为200天，常规备份周期为每星期一、星期三、星期五、星期日，于2022年12月31日对其设置备份落冷策略：备份文件生成90天后沉降至标准存储，180天后沉降至归档存储。

落冷预期

- 备份落冷策略生效时间：2023年1月1日内生效。
- 存量备份文件：以2023年1月1日为起点，已生成超过90天的存量备份文件将沉降至标准存储，超过180天的存量备份文件将沉降至归档存储。
- 新增备份文件：以2023年1月1日为起点，新生成的备份文件超过90天后将沉降至标准存储，超过180天后将沉降至归档存储。

示例2：定期备份 + 备份落冷设置背景

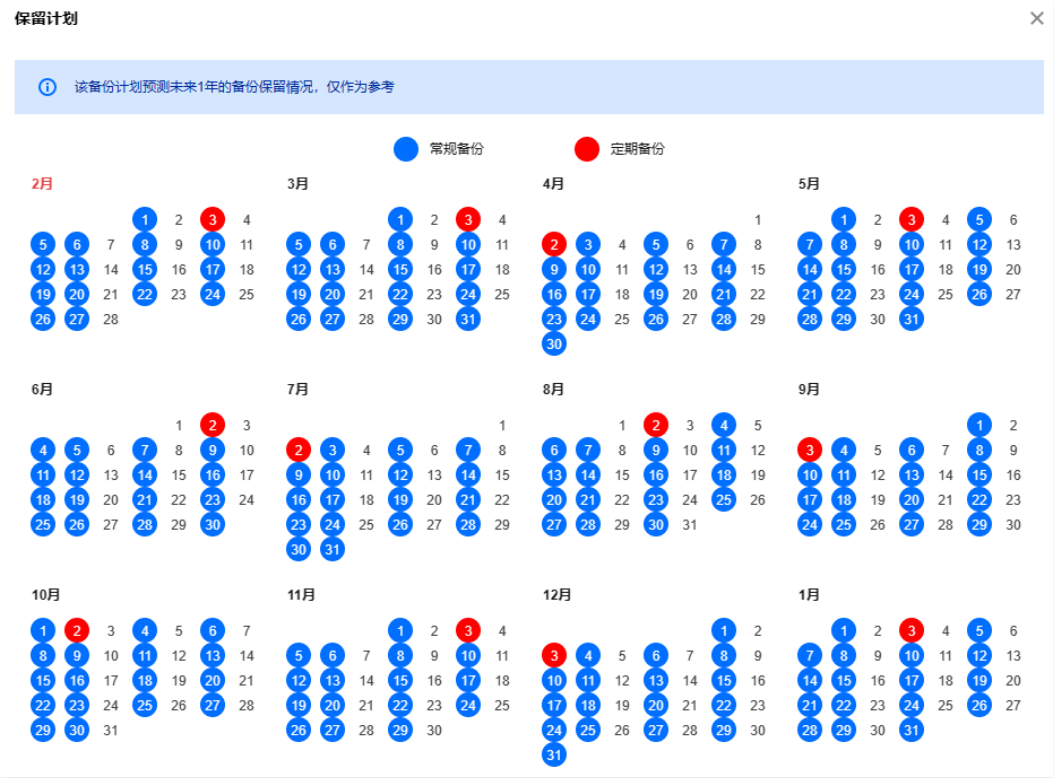
企业 B 在广州地域有一个云数据库 MySQL 实例，其数据备份保留时间为365天，常规备份周期为每星期一、星期三、星期五、星期日，定期备份保留时间为

365天，每月定期保留1个备份文件，于2023年1月31日对其设置备份落冷策略：备份文件生成90天后沉降至标准存储，180天后沉降至归档存储。

落冷预期

- 备份落冷策略生效时间：2023年2月1日内生效。
- 存量备份文件：以2023年2月1日为起点，已生成超过90天的存量备份文件将沉降至标准存储，超过180天的存量备份文件将沉降至归档存储。
- 新增备份文件：以2023年2月1日为起点，新生成的备份文件超过90天后将沉降至标准存储，超过180天后将沉降至归档存储。

下图展示的备份保留计划中，每个新产生的备份文件都将按备份落冷策略设置的时间进行沉降，且备份沉降数据只能由热变冷。



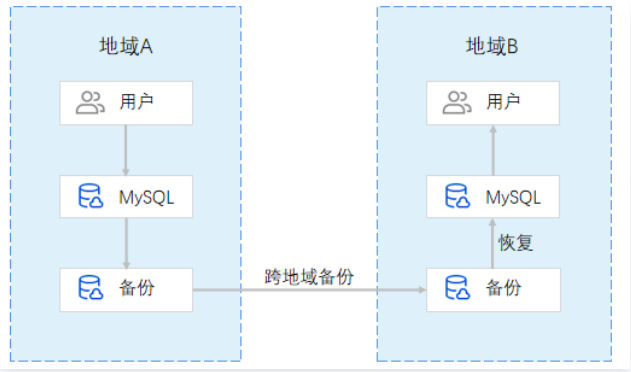
跨地域备份

最近更新时间：2025-06-03 17:59:22

云数据库 MySQL 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。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。
云数据库 MySQL 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。



跨地域备份说明

- 单节点云盘版实例暂不支持跨地域备份功能。
- 跨地域备份可恢复数据到实例所在地域或备份所在地域。
- 开启跨地域备份后，不会影响本地默认备份，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。
- 跨地域备份的保留天数仅影响跨地域存储的备份生命周期。
- 存储在异地的备份以及 binlog 无法使用免费存储空间，跨地域的备份占用会计算到所属主实例所在备份地域。
- 开启或关闭跨地域备份以及修改地域选项不会影响存量备份。
- 开通跨地域备份后，会将最近一次有效备份以及最近一次有效备份时间至当前时间的 binlog 同步至目标地域。
- 再次开启跨地域备份，会导致开启时间点前的异地备份无法使用时间点恢复能力。

计费说明

跨地域备份计费说明请参见 备份空间收费说明 。

跨地域备份和本地默认备份的区别

对比项	跨地域备份	本地默认备份
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（最多指定两个）	备份存储在实例所在地域
备份恢复	可以恢复数据到： • 原实例 • 目的地域的新实例	可以恢复到： • 原实例 • 当前地域的新实例
是否占用免费存储空间	不可占用免费存储空间	可占用免费存储空间

支持地域

目前支持此功能的地域如下，其他城市后续陆续开放。

源地域	支持备份到的目的地域
北京、上海、广州、成都、南京、重庆、中国香港、新加坡	北京、上海、广州、成都、南京、重庆、中国香港、新加坡 **说明：** 支持备份至除源地域自身以外的地域，具体以控制台实际支持地域为准。
雅加达、曼谷、首尔、东京、硅谷、弗吉尼亚、法兰克福、圣保罗	新加坡

开启跨地域备份

1. 登录 MySQL 控制台，在实例列表，单击实例 ID 或**操作**列的**管理**，进入实例管理页面。
2. 在实例管理页面，选择**备份恢复** > **跨地域备份**。

3. 在跨地域备份设置窗口完成配置后，单击**确定**开启跨地域备份。

- 跨地域备份开关：默认为关闭。
- 备份 binlog：当开启跨地域备份时自动打开，开启后可单独关闭备份 binlog。
- 备份地域：备份地域最少选择一个与主实例不同的地域，最多可选择两个。
- 备份地域保留时长：默认为7天，支持设置为3天 – 1830天，到期后备份集会自动删除。

4. 跨地域备份完成后，备份会同步到目标地域，可在源实例备份列表查询。

5. 跨地域备份的文件在**备份地域**项下面会展示所有您选择备份的地域。
   备份列表可查看字段为：

字段	说明
文件名	创建备份时可设置文件名，创建后暂不支持修改
备份时间点	备份开启时间
任务开始时间任务结束时间	执行备份开始和完成的时间
备份大小	本次备份文件的大小
类型	备份文件类型，双节点、三节点实例的跨地域备份类型为物理冷备，云盘版实例的跨地域备份类型为快照备份
备份方式	自动备份
备份方法	全量备份
备份地域	展示备份文件存在的所有地域
状态	展示备份任务状态
操作	下载 克隆

关闭跨地域备份

您可通过控制台关闭跨地域备份功能，关闭后，跨地域备份文件不会立刻删除，备份文件会根据设置的生命周期到期后自动删除。

1. 登录 MySQL 控制台，在实例列表，单击实例 ID 或**操作**列的**管理**，进入实例管理页面。
2. 在实例管理页面，选择**备份恢复** > **跨地域备份**。
3. 在跨地域备份设置窗口选择关闭，单击**确定**。

常见问题

为什么关闭跨地域备份后还会有费用产生？

当使用完免费存储空间时，超过额度的备份空间会计费。在关闭跨地域备份之后，不会再产生新的备份，但旧的备份不会马上删除。已有备份有保留时长，到期前的时间备份文件仍然占用存储空间，所以到期前还会有费用产生。您可以设置备份保留时长，等到期后备份文件会全部自动清除，就不会再收取跨地域备份的费用。

备份加密

最近更新时间：2025-06-03 17:59:22

云数据库 MySQL 支持备份加密功能，支持对物理备份以及日志备份（binlog）进行加密，开启备份加密后，如果需要使用加密的备份，您需要先下载备份，然后下载对应备份文件的加密密钥进行解密。本文为您介绍开启、关闭备份加密功能，下载备份文件密钥。

前提条件

MySQL 实例为双节点、三节点架构。

注意事项

- 开启备份加密后，不会对开启前的备份进行加密，新增备份文件会自动加密存储。
- 备份加密无法修改密钥。
- 开启备份加密后，在控制台上使用备份时不需要您手动解密，后端会自动解密后执行相关操作（例如克隆等），但是如果下载加密备份到本地，您需要再下载对应备份加密文件的密钥，以便后续对此备份进行解密。关于解密备份文件，您可参见 使用物理数据恢复数据库 。

开启备份加密

1. 登录 MySQL 控制台，在实例列表，单击**实例 ID** 或**操作**列的**管理**，进入实例管理页面。
2. 在实例管理页，选择**备份恢复**，单击**备份加密**。

3. 在弹出的对话框选择开启备份加密的类型，单击**确定**。

> **说明：**
> 开启对应的备份加密后，物理备份时间预期会延长20%，日志备份时间预期会延长30%。

4. 开启备份加密后，根据自动备份设置新生成的备份文件，会自动加密。
5. 您也可单击**手动备份**，在手动备份设置页，选择配置后，单击**确定**即可对新增的物理备份进行加密。

参数	说明
选择备份方式	选择物理冷备。
选择对象	默认为这个实例。
备份加密开关	默认为开启状态。
备份名	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。

关闭备份加密

> **说明：**
> 关闭备份加密后，不会对关闭前的备份进行解密，新增备份文件不会加密存储。

1. 登录 MySQL 控制台，在实例列表，单击目标**实例 ID** 或**操作**列的**管理**，进入实例管理页面。
2. 在实例管理页，选择**备份恢复**，单击**备份加密**。

3. 在弹出的对话框，关闭对应的备份加密类型，单击**确定**。

下载备份文件密钥

1. 登录 MySQL 控制台，在实例列表，单击目标**实例 ID** 或**操作**列的**管理**，进入实例管理页面。
2. 在实例管理页面，选择**备份恢复** > **数据备份列表**页，选择需要下载的备份对应的解密密钥，在**操作**列单击**下载密钥**。

3. 下载成功后，文件格式如下图所示。

常见问题

云数据库 MySQL 的加密算法是什么？

云数据库 MySQL 采用的加密算法是 AES-256。

备份保留设置

最近更新时间：2025-04-30 10:57:02

云数据 MySQL 支持在实例销毁前或者在实例销毁时进行备份保留设置，可以避免实例误删除导致备份被立即销毁，可确保实例在销毁后，用户仍然拥有一定时间的该实例的备份。本文为您介绍备份保留设置相关操作及说明。

前提条件

- 实例架构为双节点、三节点。
- 实例下已产生备份文件。

计费说明

详细计费价格请参见 备份空间收费说明 。

操作步骤

备份保留设置

1. 登录 MySQL 控制台 ，在实例列表，单击**实例 ID** 或**操作**列的**管理**，进入实例管理页面。
2. 在实例管理页，选择**备份恢复**，单击**备份保留设置**。

3. 在弹窗中完成如下配置，单击**确定**。

参数	说明
实例销毁后备份文件	设置实例销毁后是否保留备份文件，支持选择不保留或者保留。 • 不保留：选择不保留，表示实例销毁后，备份会同步销毁，不进行保存，无法在云上进行数据恢复。 • 保留：选择保留，表示实例销毁后，会根据选择的范围保存备份，这部分备份涉及计费。
保留策略	实例销毁后备份文件选择保留时，此项显示，保留策略选项如下： • 保留全部：表示保留该实例在当前地域产生的全部备份，可选择是否保留 Binlog 文件。 • 保留最近一个：表示保留该实例在当前地域产生的最新的1个备份，默认不可选择保留 Binlog 文件。 • 保留时间段内的：表示保留该实例在当前地域产生的，并且从所选时间之后产生的备份，可选择是否保留 Binlog 文件。例如：选择保留2025年04月28日之后产生的备份。
Binlog 文件	保留策略选择“保留全部”或“保留时间段内的”时，此项显示，可选择是否保留 Binlog 文件。

已销毁实例备份的查询与下载

实例销毁后备份文件选择为“保留”的实例，在被销毁下线后，已销毁实例的备份可以通过数据库备份页面进行查询和下载。

1. 登录 MySQL 控制台 ，在左侧导航选择**数据库备份**页。
2. 在**数据库备份**页，选择**备份列表** > **已销毁实例备份**，可在列表中查询已保留的备份信息。

- 筛选列表：支持对**实例下线时间**进行筛选以快速获取对应下线时间内的备份列表。
- 过滤实例 ID：支持通过实例 ID，精准过滤出需要查询的实例的已销毁备份。
- 刷新：支持刷新列表。
- 筛选列表字段：支持通过设置按钮 ⚙，选择您想显示的列表详细信息。
- 下载：支持下载列表（10000个以内的数据量）。

3. 如需下载备份文件，可先找到目标备份文件，在字段“文件数”下单击对应文件数量，即可在弹出的侧边栏中查询具体的备份文件和下载操作。

4. 单击下载后，在弹窗中设置**下载条件**，单击**生成下载链接**，生成成功后，即可复制链接通过 wget 命令进行下载。

> **说明：**
> - 下载地址有效期均为12小时，过期后请主动刷新页面重新获取。
> - wget 命令格式：wget -c '<备份文件下载地址>' -O 自定义文件名.xb

设置备份下载规则

最近更新时间：2025-02-25 16:21:53

云数据库 MySQL 实例备份文件默认可以通过公网或内网进行下载，当您需要对下载进行限制时，可以通过下载配置进行相应的调整。

> **说明：**
> 数据库备份下载配置目前支持的地域：广州、上海、北京、深圳、成都、重庆、南京、中国香港、北京金融、上海金融、深圳金融、新加坡、硅谷、法兰克福、首尔、曼谷、东京。

设置备份下载规则

1. 登录 MySQL 控制台，在左侧导航选择**数据库备份**页，在上方选择地域。
2. 选择**下载配置**页，可查看对应地域的备份下载配置信息，单击**编辑**，对下载限制条件进行配置。

> **说明：**
> 外网下载默认开启，开启外网下载时，外网和内网均可以下载备份。

3. 在弹出页面，配置下载详情，单击**确定**。
   - 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：
     - 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 地址条件值使用 `,` 进行分割。
     - IP 条件支持 IP 地址以及 IP 地址段作为条件值。
     - 当 IP、VPC 条件均为空时，内网下载不受限制。

4. 配置完成后，返回**下载配置**页可查看生效的条件。

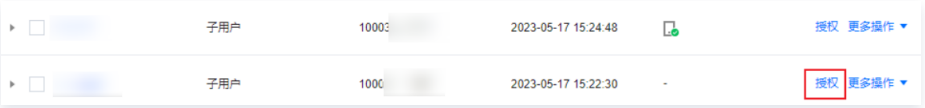
授权子账号使用设置备份下载规则

默认情况下，子账号没有设置 MySQL 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。

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 。

给子账号授权

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. 在弹出的对话框，选择 **QcloudCDBFullAccess云数据库（CDB）全读写访问权限**预设策略，单击**确定**，即可完成子用户授权。

策略语法

MySQL 数据库设置下载备份规则的 CAM 策略描述如下：

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

- **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 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（以 name 前缀描述）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。

API 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。对于数据库审计，请使用以 name/cdb: 为前缀的 API 。如果您要在单个语句中指定多个操作，请使用逗号将它们隔开，如下所示：

```
"action":["name/cdb:action1","name/cdb: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":["name/cdb:Describe*"]
```

资源路径

资源路径的一般形式如下：

```
qcs::service_type::account:resource
```

- service_type：产品简称，此处为 cdb。
- account：资源拥有者的主账号信息，如 uin/326xxx46。
- resource：产品的具体资源详情，每个 MySQL 实例（instanceId）就是一个资源。

示例如下：

```
 "resource": ["qcs::cdb::uin/326xxx46:instanceId/cdb-kfxxh3"]
```

其中，cdb-kfxxh3 是 MySQL 实例资源的 ID，在这里是 CAM 策略语句中的资源 resource。

示例

以下示例仅为展示 CAM 用法，MySQL 数据库设置备份下载规则的完整 API 请参见 API 文档 。

```
{
        "version":"2.0",
        "statement":
        [
           {
              "effect":"allow",
              "action": ["name/cdb: ModifyBackupDownloadRestriction"],
              "resource": ["*"]
             }
        ]
}
```

自定义 MySQL 数据库设置备份下载规则策略

1. 以主账号身份登录 访问管理控制台 ，在策略列表，单击**新建自定义策略**。

2. 在弹出的对话框，选择**按策略生成器创建**。
3. 在选择服务和操作页面，选择各项配置，单击 添加声明后，单击下一步。

- 效果（Effect）：选择允许或拒绝，表示对操作项的许可执行。
- 服务（Service）：选择云数据库 MySQL。
- 操作（Action）：选择 MySQL 数据库设置下载备份规则的 API，请参见 API 文档 。
- 资源（Resource）：请参见 资源描述方式，输入 `*` ，表示可以设置所在地域 MySQL 实例的备份下载规则。

4. 在编辑策略页面，按命名规范，输入“策略名称”（例如 BackupDownloadRestriction）和“描述”后，单击**完成**。

5. 返回策略列表，即可查看刚创建的自定义策略。

查看备份空间

最近更新时间：2025-04-30 10:00:44

本文为您介绍如何通过控制台查看 MySQL 实例备份空间及免费额度。

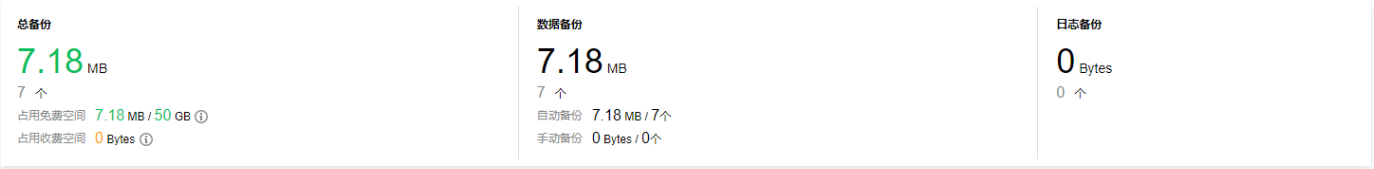
操作场景

- 对于双节点（本地盘）、三节点（本地盘）实例，云数据库 MySQL 实例备份文件所占用的备份空间按地域分配，备份空间相当于该地域中所有 MySQL 数据库备份的存储空间之和，包括自动数据备份、手动数据备份和日志备份。延长备份保留时间或增加手动备份频次会增加数据库备份存储空间。
- 对于单节点（云盘）实例、云盘版（云盘）实例，云数据库 MySQL 实例备份文件所占用的备份空间按实例维度分配，一个单节点（云盘）实例或一个云盘版（云盘）实例的免费备份空间大小为这个实例存储空间的200%。

查看本地盘实例备份空间及免费额度

1. 登录 MySQL 控制台，在左侧导航选择**数据库备份**页。
2. 在上方选择地域，**概览**页可查看对应地域的备份信息：备份总计、备份趋势、备份统计。
   - 备份总计：展示总备份、数据备份、日志备份的空间大小和数量，以及总备份的免费空间占用情况。

> **说明：**
> - 绿色：表示总备份使用量没有超过赠送空间。
> - 橙色：表示总备份使用量已经超过赠送空间，产生收费空间，请参见 备份空间收费说明 。



   - 备份趋势：展示各个总计数据趋势。
   - 备份统计：展示该地域下实例的实例 ID/名称（单击可跳转到实例详情页）、备份空间（ 支持按照容量大小排序 ）、数据和日志备份等信息， 右上角搜索框支持搜索实例名/ID。
3. 在上方选择**备份列表**页，备份列表分为数据备份列表和日志备份列表。在列表单击实例名可以跳转到实例详情页；备份列表支持时间段筛选，及实例名和实例 ID 模糊搜索。

   - **数据备份列表**
     - 支持信息字段过滤：
       类型：全部、逻辑冷备、物理冷备。
       备份方式：全部、自动、手动。
       备份方法：目前仅支持全量。
     - 支持按备份时间点、任务开始时间、任务结束时间、备份大小进行排序。
     - 在操作列单击**详情**可跳转到实例备份恢复页，单击**下载**可进行备份下载，仅手动备份支持**删除**操作。
   - **日志备份列表**
     - 支持按日志数据开始时间、日志数据结束时间字段排序。
     - 在操作列单击**详情**可跳转到实例备份恢复页，单击**下载**可进行日志下载。

查看云盘版实例备份空间及免费额度

1. 登录 MySQL 控制台，在实例列表，单击目标实例 ID 或**操作**列的**管理**，进入实例管理页面。

2. 在实例管理页面，选择**备份恢复**，即可查看该实例对应的实际占用空间和免费空间额度。

热点问题

超出免费额度的备份空间如何收费？如何减少备份空间开销？

请参见 备份空间收费说明 。

设置本地 binlog 保留

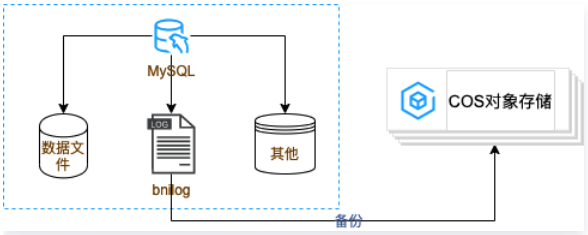
最近更新时间：2025-06-03 17:59:22

本文为您介绍如何通过控制台设置 MySQL 实例的本地 binlog 保留周期。

> **说明：**
> - 单节点本地盘实例（只读实例）不支持设置本地 binlog 保留。
> - 云盘版实例、双节点实例、三节点实例、单节点云盘实例、灾备实例支持设置本地 binlog 保留，保留时长策略分别如下。
> - 云盘版实例、双节点实例、三节点实例的本地 binlog 保留时长（小时）默认为120，范围：6 – 168。
> - 灾备实例的本地 binlog 保留时长（小时）默认为120，范围：120 – 168。
> - 单节点云盘实例的本地 binlog 保留时长（小时）默认为120，范围：0 – 168。
> - 若双节点实例、三节点实例下无灾备实例，则该主实例的本地 binlog 保留时长（小时）范围是：6 – 168；若双节点实例、三节点实例下有灾备实例，或者要为双节点实例、三节点实例添加灾备实例，为避免同步异常，该主实例的本地 binlog 保留时长（小时）不能设置低于120小时，范围是：120 – 168。

binlog 说明

云数据库 MySQL 实例在执行大事务或大量 DML 时会产生较多的 binlog，binlog 写满 256MB 会进行一次切割，切割后的 binlog 文件会被上传至对象存储 COS，然后控制台的日志列表会显示上传到 COS 的 binlog 文件。



操作场景

binlog 在上传至 COS 前，会暂存于实例磁盘，从而导致磁盘使用空间增加。您可以通过控制台设置 Binlog 保留周期来控制占用空间的比例，也可以扩容磁盘，建议磁盘使用率不超过80%，其次建议您及时清理无用数据。

- binlog 是 MySQL 同步数据的基础，为保障数据库的可恢复性、稳定性、高可用能力，云数据库 MySQL 不支持关闭 binlog。
- 生成的 binlog 会先通过系统的 自动备份功能 备份到 COS，备份后的 binlog 会根据设置的本地 binlog 保留策略进行删除，当前正在使用的 binlog 无法被清理，防止异常。因此，清理的过程有一定的延迟，设置后请耐心等待一段时间。

> **说明：**
> 清理过期 binlog 日志规则：
> 每60秒检查一次本地 binlog 日志，如果检查到 binlog 开始时间或占用空间不符合设定的保留规则时，会加入到待删除队列。删除队列中的 binlog 会按照时间排序，从最早的 binlog 开始逐一删除，直至清空待删除队列。

操作步骤

1. 登录 MySQL 控制台，在实例列表页，单击实例 ID，进入实例管理页面。
2. 在实例管理页面，选择**备份恢复**页，单击**本地 Binlog 设置**。
3. 在弹出的对话框，填写需要保留的时长以及空间使用率，确认无误后，单击**确定**，您可参考下面对应实例操作进行设置。

双节点、三节点实例操作

- 保留时长（小时）：开启日志备份后，本地 Binlog 的保留最大时间。默认为120，可修改范围：6 – 168 整数。

> **注意：**
> 若双节点、三节点实例的本地 Binlog 保留时长配置低于120小时，则该主实例将无法正常购买灾备实例。如需挂载灾备实例，该主实例的本地 Binlog 保留时长设置应不低于120小时。

- 空间使用率不超过（%）：默认为30，可修改范围：30 – 50 整数。

  本地 Binlog 空间使用率 = 本地 Binlog 大小 / 实例总可用（购买）空间大小。此为循环使用空间，超出后则从最早的 Binlog 开始清理，直到空间使用率低于该比例且本地 Binlog 不超过保留时长。

单节点云盘实例操作

- 保留时长（小时）：开启日志备份后，本地 Binlog 的保留最大时间。默认为120，可修改范围：0 – 168 整数。
- 空间使用率不超过（%）：默认为30，可修改范围：30 – 50 整数。

  本地 Binlog 空间使用率 = 本地 Binlog 大小 / 实例总可用（购买）空间大小。此为循环使用空间，超出后则从最早的 Binlog 开始清理，直到空间使用率低于该比例且本地 Binlog 不超过保留时长。

灾备实例操作

- 保留时长（小时）：开启日志备份后，本地 Binlog 的保留最大时间。默认为120，可修改范围：120 – 168 整数。
- 空间使用率不超过（%）：默认为30，可修改范围：30 – 50 整数。
  本地 Binlog 空间使用率 = 本地 Binlog 大小 / 实例总可用（购买）空间大小。此为循环使用空间，超出后则从最早的 Binlog 开始清理，直到空间使用率低于该比例且本地 Binlog 不超过保留时长。

云盘版实例操作

- 保留时长（小时）：开启日志备份后，本地 Binlog 的保留最大时间。默认为120，可修改范围：6 – 168 整数。
- 空间使用率不超过（%）：默认为30，可修改范围：30 – 50 整数。
  本地 Binlog 空间使用率 = 本地 Binlog 大小 / 实例总可用（购买）空间大小。此为循环使用空间，超出后则从最早的 Binlog 开始清理，直到空间使用率低于该比例且本地 Binlog 不超过保留时长。

热点问题

本地 binlog 保留设置过小是否会影响数据库恢复?

不影响，产生的 binlog 会通过自动备份功能尽快上传至 COS，尚未上传的 binlog 不会进行清理，本地 binlog 保留设置过小会影响回档速度，请进行适当配置。

本地 binlog 保留默认设置是多少?

本地 binlog 默认保留时长：120小时，空间使用率默认不超过30%，您可根据需要设置本地 binlog 保留时长。

binlog 是否会占用实例磁盘空间?

会，生成的 binlog 会先自动备份到 COS，备份后的 binlog 会根据设置保留策略进行删除，期间 binlog 会暂存至实例磁盘。

为什么我的本地 binlog 保留时长最低只能设置到120小时?

- 请检查您的实例是否是双节点或三节点实例并挂载有灾备实例，如果挂载有灾备实例，为了避免同步异常，该主实例的本地 binlog 保留时长最低只能设置到120小时。
- 请检查您选择修改本地 binlog 保留时长的实例是否是灾备实例，如果是灾备实例，为了保证灾备同步正常，灾备实例修改的范围是：120 – 168。

为什么我的主实例购买不了灾备实例?

若要为主实例创建灾备实例，需同时满足以下条件，您可对应检查。

- 主实例为双节点或三节点实例。
- 主实例的本地 binlog 保留时长大于等于120小时。
- 主实例为1GB内存、50GB硬盘及以上规格，且版本为 MySQL 5.6及以上版本（MySQL 5.6版本需 提交工单 申请使用此功能），引擎为 InnoDB 引擎。

使用备份文件恢复数据库
使用物理备份恢复数据库

最近更新时间：2025-06-10 17:00:32

操作场景

> **说明：**
> 为节约存储空间，云数据库 MySQL 的物理备份和逻辑备份文件，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）进行压缩与打包。

开源软件 Percona Xtrabackup 可以用于对数据库进行备份恢复，本文为您介绍使用 XtraBackup 工具，将 MySQL 物理备份文件恢复至其他主机上的自建数据库。

> **注意：**
> 如果使用透明加密或 Instant DDL 功能，则无法使用物理备份在自建系统上进行恢复。

- XtraBackup 只支持 Linux 平台，不支持 Windows 平台。
- Windows 平台恢复数据请参考 命令行工具迁移数据 。

前提条件

- 下载并安装 XtraBackup 工具。

  > **说明：**
  > 请确保您下载的工具的版本与您数据库的版本一致，例如您的实例数据库版本为 MySQL 8.0.30，则请对应下载8.0.30版本的 Percona XtraBackup 工具。点击 下载地址 可跳转查询版本和下载工具，安装介绍请参见 Percona XtraBackup 指引 。

- 支持的实例版本：MySQL 双节点、三节点。

  > **说明：**
  > 本文以 CentOS 操作系统的云服务器和 MySQL 5.7 版本为例进行演示。

步骤1：
下载备份文件

您可通过控制台下载云数据库 MySQL 的数据备份、日志备份。

> **说明：**
> 默认每个 IP 限制10个链接，每个链接下载速度可达20Mpbs - 30Mpbs。

1. 登录 MySQL 控制台 ，在实例列表，单击实例 ID 或**操作**列的**管理**，进入实例管理页面。
2. 在实例管理页面，选择**备份恢复** > **数据备份列表**页， 选择需要下载的备份，在**操作**列单击**下载**。
3. 在弹出的对话框，下载地址后单击**复制**，获取下载链接。
4. 登录本地自建 MySQL 所在的 Linux 云服务器，运用 wget 命令进行内网高速下载物理备份。

   > **说明：**
   > - 下载地址有效期均为12小时，过期后请主动刷新页面重新获取。
   > - wget 命令格式：wget -c '备份文件下载地址' -O 自定义文件名.xb

   示例如下：

```
wget -c 'https://mysql-database-backup-sh-1218.cos.ap-nanjing.myqcloud.com/12427%2Fmysql%2F0674-ffba-
11e9-b592-70bd%2Fdata%2Fautomatic-delete%2F2019-12-03%2Fautomatic%2Fxtrabackup%2Fbk_61_156758150%2Fcdb-
```

```
293fl9ya_backup_20191203000202.xb?sign=q-sign-algorithm%3Dsha1%26q-ak%3DAKzxfbLJ1%26q-sign-time%3D1575374119%3B1575417319%26q-key-time%3D1575374119%3B1575417319%26q-header-list%3D%26q-url-param-list%3D%26q-signature%3Dba959757&response-content-disposition=attachment%3Bfilename%3D%22yuan177685_backup_20191203000202.xb%22&response-content-type=application%2Foctet-stream' -O /data/test.xb
```

步骤2：下载备份解密密钥（开启了备份加密功能才需操作此步骤）

您可通过控制台下载所需物理备份对应的数据备份解密密钥。

> **说明：**
> 每个数据库备份的解密密钥独立生成，如开启了备份加密功能，下载备份文件时需要连同解密密钥一并下载保留。

1. 登录 MySQL 控制台，在实例列表，单击实例 ID 或**操作**列的**管理**，进入实例管理页面。
2. 在实例管理页面，选择**备份恢复** > **数据备份列表**页， 选择需要下载的备份对应的解密密钥，在**操作**列单击**下载密钥**。

步骤3：恢复数据

3.1 解包备份文件

使用 xbstream 命令将数据备份文件解包到目标目录。

```
xbstream -x --decrypt=AES256 --encrypt-key-file=<备份密钥文件> --parallel=2  -C /data/mysql < /data/test.xb
```

> **说明：**
> - 未开启备份加密时，解包备份文件命令为：`xbstream -x  -C /data/mysql < /data/test.xb` 。
> - 本文目标目录以 `/data/mysql` 为数据文件恢复存储，您可根据实际情况替换为实际路径。
> - `/data/test.xb` 替换为您的备份文件。

解包结果如下图所示：

```
[root@[redacted] mysql]# ll /data/mysql
总用量 2920
-rw-r----- 1 root root     459 4月  20 10:12 backup-my.cnf.qp
drwxr-x--- 2 root root    4096 4月  20 10:12 __cdb_recycle_bin__
-rw-r----- 1 root root     991 4月  20 10:12 ib_buffer_pool.qp
-rw-r----- 1 root root  161172 4月  20 10:12 ibdata1.qp
drwxr-x--- 2 root root    4096 4月  20 10:12 mysql
-rw-r----- 1 root root     222 4月  20 10:12 mysql-bin.000033.qp
-rw-r----- 1 root root     125 4月  20 10:12 mysql-bin.index.qp
-rw-r----- 1 root root 2366987 4月  20 10:12 mysql.ibd.qp
drwxr-x--- 2 root root    4096 4月  20 10:12 performance_schema
drwxr-x--- 2 root root    4096 4月  20 10:12 sys
-rw-r----- 1 root root  199604 4月  20 10:12 undo_001.qp
-rw-r----- 1 root root  198976 4月  20 10:12 undo_002.qp
-rw-r----- 1 root root     158 4月  20 10:12 xtrabackup_binlog_info.qp
-rw-r----- 1 root root      95 4月  20 10:12 xtrabackup_checkpoints
-rw-r----- 1 root root     872 4月  20 10:12 xtrabackup_info.qp
-rw-r----- 1 root root     540 4月  20 10:12 xtrabackup_logfile.qp
-rw-r----- 1 root root     191 4月  20 10:12 xtrabackup_slave_info.qp
-rw-r----- 1 root root     130 4月  20 10:12 xtrabackup_tablespaces.qp
```

如需解包的备份文件是 binlog 备份，命令如下。

```
openssl enc -d -aes256 -k <encrypt_key> -in <download_binlog_name> -out <output_binlog_name>;
```

- <encrypt_key>：替换为实际的备份密钥文件内容。
- <download_binlog_name>：替换为实际加密的 binlog 文件名。
- <output_binlog_name>：替换为实际解密的 binlog 文件名。

3.2 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. 通过如下命令解压出 qpress 二进制文件。

```
tar -xf qpress-11-linux-x64.tar -C /usr/local/bin
source /etc/profile
```

3. 使用如下命令将目标目录下所有以 `.qp` 结尾的文件都解压出来。

```
xtrabackup --decompress --target-dir=/data/mysql
```

> **说明：**
> - `/data/mysql` 为之前存储备份文件的目标目录 ，您可根据实际情况替换为实际路径。
> - Percona Xtrabackup 在2.4.6及以上版本中才支持 `--remove-original` 选项。
> - `xtrabackup` 默认在解压缩时不删除原始的压缩文件，若需解压完删除原始的压缩文件，可在上面的命令中加上 `--remove-original` 参数。

```
[root@VM_17_57_centos data]# xtrabackup --decompress  --target-dir=/data
xtrabackup: recognized server arguments: --datadir=/var/lib/mysql --log_bin=/var/lib/mysql/mysql-bin --server-id=1573
xtrabackup: recognized client arguments: --datadir=/var/lib/mysql --log_bin=/var/lib/mysql/mysql-bin --server-id=1573 --decompress=1 --target-dir=/data
xtrabackup version 2.4.12 based on MySQL server 5.7.19 Linux (x86_64) (revision id: 170eb8c)
181204 10:39:13 [01] decompressing ./test/fc1.ibd.qp
181204 10:39:13 [01] decompressing ./test/fc2.ibd.qp
181204 10:39:13 [01] decompressing ./test/sbtest4.frm.qp
181204 10:39:13 [01] decompressing ./test/sbtest3.ibd.qp
181204 10:39:14 [01] decompressing ./test/fc1.frm.qp
181204 10:39:14 [01] decompressing ./test/sbtest5.frm.qp
181204 10:39:14 [01] decompressing ./test/sbtest7.frm.qp
181204 10:39:14 [01] decompressing ./test/sbtest2.frm.qp
181204 10:39:14 [01] decompressing ./test/sbtest1.ibd.qp
181204 10:39:14 [01] decompressing ./test/sbtest4.ibd.qp
181204 10:39:14 [01] decompressing ./test/sbtest9.ibd.qp
181204 10:39:14 [01] decompressing ./test/sbtest5.ibd.qp
181204 10:39:14 [01] decompressing ./test/fc2.frm.qp
181204 10:39:14 [01] decompressing ./test/sbtest10.ibd.qp
181204 10:39:14 [01] decompressing ./test/sbtest7.ibd.qp
181204 10:39:14 [01] decompressing ./test/mytest.ibd.qp
181204 10:39:14 [01] decompressing ./test/sbtest2.ibd.qp
181204 10:39:14 [01] decompressing ./test/sbtest6.frm.qp
181204 10:39:14 [01] decompressing ./test/sbtest1.frm.qp
181204 10:39:14 [01] decompressing ./test/sbtest10.frm.qp
181204 10:39:14 [01] decompressing ./test/sbtest3.frm.qp
181204 10:39:14 [01] decompressing ./test/sbtest8.ibd.qp
181204 10:39:14 [01] decompressing ./test/sbtest6.ibd.qp
181204 10:39:14 [01] decompressing ./test/sbtest8.frm.qp
181204 10:39:14 [01] decompressing ./test/mytest.frm.qp
181204 10:39:14 [01] decompressing ./test/sbtest9.frm.qp
181204 10:39:14 [01] decompressing ./xtrabackup_logfile.qp
181204 10:39:14 [01] decompressing ./ibdata1.qp
181204 10:39:14 [01] decompressing ./undo001.qp
181204 10:39:14 [01] decompressing ./mysql/general_log.frm.qp
181204 10:39:14 [01] decompressing ./mysql/procs_priv.MYI.qp
181204 10:39:14 [01] decompressing ./mysql/ndb_binlog_index.MYI.qp
181204 10:39:14 [01] decompressing ./mysql/slow_log.frm.qp
181204 10:39:14 [01] decompressing ./mysql/general_log.CSV.qp
181204 10:39:14 [01] decompressing ./mysql/time_zone_leap_second.MYD.qp
```

3.3 Prepare 备份文件

备份解压出来之后，执行如下命令进行 apply log 操作。

```
xtrabackup --prepare  --target-dir=/data/mysql
```

执行后若结果中包含如下输出，则表示 prepare 成功。

```
InnoDB: Starting shutdown...
InnoDB: Shutdown completed; log sequence number 922626089
181204 10:47:24 completed OK!
```

3.4 修改配置文件

1. 执行如下命令打开 `backup-my.cnf` 文件。

```
vi /data/mysql/backup-my.cnf
```

> **说明：**
> 本文以目标目录 `/data/mysql` 为例，您可以根据实际情况将其替换成实际路径。

2. 由于存在的版本问题，请将解压文件 `backup-my.cnf` 中如下参数进行注释。
   - innodb_checksum_algorithm
   - innodb_log_checksum_algorithm
   - innodb_fast_checksum
   - innodb_page_size
   - innodb_log_block_size
   - redo_log_version
   - server_uuid
   - master_key_id

```
# This MySQL options file was generated by innobackupex.

# The MySQL server
[mysqld]
innodb_data_file_path=ibdata1:12M:autoextend
innodb_log_files_in_group=2
innodb_log_file_size=536870912
innodb_undo_directory=.
innodb_undo_tablespaces=0
server_id=0
#innodb_checksum_algorithm=innodb
#innodb_log_checksum_algorithm=innodb
#innodb_fast_checksum=false
#innodb_page_size=16384
#innodb_log_block_size=512
#redo_log_version=0
```

3.5 修改文件属性

修改文件属性，并检查文件所属为 mysql 用户。

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

```
总用量 34884
-rw-r----- 1 mysql mysql      424 11月  9 18:15 backup-my.cnf
-rw-r----- 1 mysql mysql 12582912 11月  9 18:12 ibdata1
-rw-r----- 1 mysql mysql 12582912 11月  9 16:30 ibtmp1
-rw-r----- 1 mysql mysql 10485760 11月  9 16:30 undo001
drwxr-x--- 2 mysql mysql     4096 11月  9 16:30 mysql
drwxr-x--- 2 mysql mysql     4096 11月  9 16:30 test
drwxr-x--- 2 mysql mysql     4096 11月  9 16:30 performance_schema
drwxr-x--- 2 mysql mysql     4096 11月  9 16:30 sbtest
```

步骤4：启动 mysqld 进程并登录验证

1. 启动 mysqld 进程。

```
mysqld_safe --defaults-file=/data/mysql/backup-my.cnf --user=mysql --datadir=/data/mysql &
```

2. 客户端登录 mysql 验证。

```
mysql  -uroot
```

```
[[redacted]154822]$ mysql -S /tmp/mysql_3307.sock
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.6.23 Source distribution

Copyright (c) 2000, 2011, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| test               |
+--------------------+
2 rows in set (0.00 sec)

mysql>
```

备份相关问题

请参见 备份常见问题 和 备份失败原因 。

使用逻辑备份恢复数据库

最近更新时间：2024-10-18 14:14:22

操作场景

> **说明：**
> 为节约存储空间，云数据库 MySQL 的物理备份和逻辑备份文件，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）进行压缩与打包。

云数据库 MySQL 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，并下载获取整个实例/部分库表的逻辑备份文件，本文为您介绍如何使用逻辑备份文件进行手动还原。

- 本文介绍的恢复方式仅适用于 Linux 平台，暂不支持 Windows 平台。
- Windows 平台恢复数据请参考 命令行工具迁移数据 。
- 支持的实例版本：MySQL 双节点、三节点。

操作步骤

步骤1：下载备份文件

1. 登录 MySQL 控制台 ，在实例列表，单击**实例 ID** 或**操作**列的**管理**，进入实例管理页面。
2. 在实例管理页面，选择**备份恢复** > **数据备份列表**页， 选择需要下载的备份，在**操作**列单击**下载**。
3. 在弹出的对话框，推荐您复制下载地址，并 登录到云数据库所在 VPC 下的 CVM（Linux 系统）中，运用 wget 命令进行内网高速下载，更高效。

> **说明：**
> - 您也可以选择**本地下载**直接下载，但耗时较多。
> - wget 命令格式：wget -c '备份文件下载地址' -O 自定义文件名.xb

示例如下：

```
wget -c 'https://mysql-database-backup-bj-118.cos.ap-beijing.myqcloud.com/12427%2Fmysql%2F42d-11ea-b887-6c0b82b%2Fdata%2Fautomatic-delete%2F2019-11-28%2Fautomatic%2Fxtrabackup%2Fbk_204_10385%2Fcdb-1pe7bexs_backup_20191128044644.xb?sign=q-sign-algorithm%3Dsha1%26q-ak%3D1%26q-sign-time%3D1574269%3B1575417469%26q-key-time%3D1575374269%3B1517469%26q-header-list%3D%26q-url-param-list%3D%26q-signature%3Dfb8fad13c4ed&response-content-disposition=attachment%3Bfilename%3D%2141731_backup_20191128044644.xb%22&response-content-type=application%2Foctet-stream' -O test0.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
```

> **说明：**
> `test0.xb` 替换为您的备份文件。

解包结果如下图所示：

```
ubuntu@VM-15-106-ubuntu:~$ xbstream -x < test0.xb
ubuntu@VM-15-106-ubuntu:~$ ll
total 1090720
drwxr-xr-x 9 ubuntu ubuntu      4096 Dec  2 18:30 ./
drwxr-xr-x 3 root   root        4096 Oct 26  2016 ../
-rw-rw-r-- 1 ubuntu ubuntu    702901 Nov 23 08:37 1254408587%2Fmysql%2F7f129d66-ac5f-11e9-b8bc-6c0b
kuoSr
-rw-rw-r-- 1 ubuntu ubuntu       419 Nov 25 21:40 backup-my.cnf
-rw-rw---- 1 ubuntu ubuntu       396 Nov 25 20:06 backup-my.cnf.qp
-rw-r--r-- 1 ubuntu ubuntu     23576 Dec  2 18:30 .bash_history
-rw-r--r-- 1 ubuntu ubuntu       220 Oct 26  2016 .bash_logout
-rw-r--r-- 1 ubuntu ubuntu      3773 Mar 16  2018 .bashrc
drwx------ 2 ubuntu ubuntu      4096 Nov 28  2017 .cache/
-rw-rw---- 1 ubuntu ubuntu    408724 Dec  2 18:30 cdb-jp0zua5k_backup_20191202182218.sql.qp
-rw-rw-r-- 1 ubuntu ubuntu  12582912 Nov 25 21:42 ibdata1
```

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 cdb-jp0zua5k_backup_20191202182218.sql.qp .
```

> **说明：**
> 请根据解压时间，找到 `.sql.qp` 后缀的备份文件，并将 `cdb-jp0zua5k_backup_20191202182218` 替换为该文件名。

解压结果如下图所示：

```
ubuntu@VM-15-106-ubuntu:~$ qpress -d cdb-jp0zua5k_backup_20191202182218.sql.qp .
ubuntu@VM-15-106-ubuntu:~$ ll
total 1091792
drwxr-xr-x 9 ubuntu ubuntu      4096 Dec  2 18:32 ./
drwxr-xr-x 3 root   root        4096 Oct 26  2016 ../
-rw-rw-r-- 1 ubuntu ubuntu    702901 Nov 23 08:37 1254408587%2Fmysql%2F7f129d66-ac5f-11e9-b
kuoSr
-rw-rw-r-- 1 ubuntu ubuntu       419 Nov 25 21:40 backup-my.cnf
-rw-rw---- 1 ubuntu ubuntu       396 Nov 25 20:06 backup-my.cnf.qp
-rw-r--r-- 1 ubuntu ubuntu     23657 Dec  2 18:32 .bash_history
-rw-r--r-- 1 ubuntu ubuntu       220 Oct 26  2016 .bash_logout
-rw-r--r-- 1 ubuntu ubuntu      3773 Mar 16  2018 .bashrc
drwx------ 2 ubuntu ubuntu      4096 Nov 28  2017 .cache/
-rw-rw-r-- 1 ubuntu ubuntu   1097176 Dec  2 18:32 cdb-jp0zua5k_backup_20191202182218.sql
-rw-rw---- 1 ubuntu ubuntu    408724 Dec  2 18:30 cdb-jp0zua5k_backup_20191202182218.sql.qp
-rw-rw-r-- 1 ubuntu ubuntu  12582912 Nov 25 21:42 ibdata1
-rw-rw---- 1 ubuntu ubuntu    179769 Nov 25 20:06 ibdata1.qp
```

步骤4：导入备份至目标数据库

> **注意：**

在导入时如果存在权限问题（提示 SET GLOBAL 为禁止操作）报错，可注释 SQL 文件中的：
`SET GLOBAL INNODB_STATS_AUTO_RECALC=ON */;`，或者直接删除 SET GLOBAL 那条语句。

```
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!50606 SET @OLD_INNODB_STATS_AUTO_RECALC=@@INNODB_STATS_AUTO_RECALC */;
/*!50606 SET GLOBAL INNODB_STATS_AUTO_RECALC=ON */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
```

考虑到云数据库 MySQL 的稳定性和安全性，云数据库 MySQL 限制了 super、shutdown、file 权限，有时在云数据库 MySQL 上执行 SET 语句时，会出现如下的报错：
`#1227-Access denied;you need(at least one of)the SUPER privilege (s) for this operation`

执行如下命令导入 sql 文件至目标数据库：

```
mysql -uroot -P3306 -h127.0.0.1 -p < cdb-jp0zua5k_backup_20191202182218.sql
```

说明：
- 本文以导入本地3306端口的 MySQL 为例，您可根据实际情况替换。
- `cdb-jp0zua5k_backup_20191202182218.sql` 替换为通过 qpress 实际解压出的 sql 文件。

下载备份

最近更新时间：2025-07-07 19:42:31

云数据库 MySQL 控制台提供备份的文件列表，并可以下载备份文件，通过下载的备份文件可将数据库恢复到其他数据库（如自建数据库）。
本文为您介绍如何通过控制台下载备份。

> **注意：**
> - 不同地域的下载规则不同，开通跨地域备份功能后，您选择备份的每个地域的下载规则都需进行配置。
> - 单节点（云盘）实例、云盘版实例手动备份产生的数据备份为快照备份方式，暂不支持下载。
> - 在 设置备份落冷 功能中，标准存储的备份文件仅能通过内网下载，暂不支持通过外网下载。归档存储的备份文件内网和外网暂时均不支持下载。

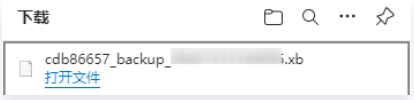
支持下载/解压/删除的文件类型

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	手动	✓	✓	✓
	物理备份	手动	✓	✓	✓
		自动	✓	✓	×
	快照备份	手动	×	×	×
日志备份	物理备份	自动	✓	✓	×

下载数据备份文件

1. 登录 MySQL 控制台 。
2. 在上方选择地域，找到目标实例，单击**实例 ID** 或**操作**列的**管理**，进入实例管理页。
3. 在实例管理页，选择**备份恢复**页，单击**数据备份列表**。

4. 在备份列表的操作列，单击**下载**，进入下载页，根据是否开启外网下载，有不同的操作。
   - 开启外网下载时：选择需要地域下的备份文件**复制下载链接**运用 wget 命令进行高速下载或进行**本地下载**。
   - 关闭外网下载时：选择需要地域下的备份文件，单击**生成下载链接**，然后复制下载链接运用 wget 命令进行高速下载。
5. 下载文件展示。



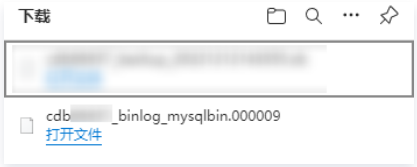
下载日志备份文件

1. 登录 MySQL 控制台 。
2. 在上方选择地域，找到目标实例，单击**实例 ID** 或**操作**列的**管理**，进入实例管理页。
3. 在实例管理页，选择**备份恢复**页，单击**日志备份列表**。

4. 在备份列表的操作列，单击**下载**，进入下载页，根据是否开启外网下载，有不同的操作。

- 开启外网下载时：选择需要地域下的备份文件**复制下载链接**运用 wget 命令进行高速下载或进行**本地下载**。
- 关闭外网下载时：选择需要地域下的备份文件，单击**生成下载链接**，然后复制下载链接运用 wget 命令进行高速下载。

5. 下载文件展示。



解压备份文件

- 解压物理备份文件请参考 使用物理备份恢复数据库 。
- 解压逻辑备份文件请参考 使用逻辑备份恢复数据库 。

回档数据库

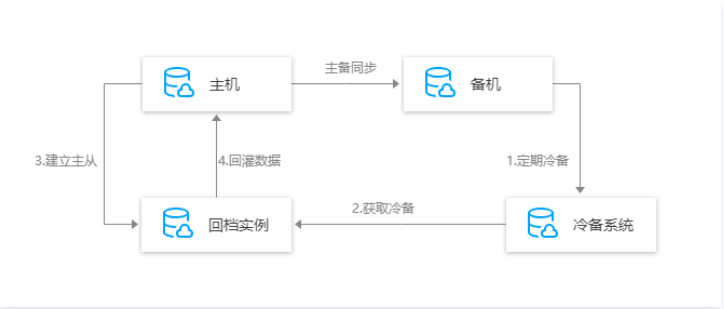
最近更新时间：2025-07-08 17:22:01

操作场景

云数据库 MySQL 不会改动用户的任何数据，因用户个人原因造成的数据损毁可自行回档修复。使用回档功能可对腾讯云平台中的数据库或表进行回档操作，回档是基于数据备份 + 日志备份（binlog），可进行实时数据回档。
云数据库 MySQL 回档通过定期镜像和实时流水重建，将云数据库或表回档到指定时间，且可以保证所有数据的时间切片一致，期间原有数据库或表的访问不受影响，回档操作会产生新的数据库或表至原实例中。回档完后，用户可以看到原来的数据库或表，以及新建的数据库或表。

功能原理

回档基于最近一次 冷备 + 对应的 `binlog` 回档到指定时间点。



1. 冷备系统每天会从 MySQL 备机导出数据到冷备系统。
2. 回档时，首先从回档系统申请一台回档实例，然后从冷备系统导出冷备数据导入临时实例（根据回档方式导入不同数据）。
3. 回档实例和 MySQL 主实例建立主备关系，并设置需要回档的时间和数据库表。
4. 将回档后的数据库表拷贝到 MySQL 主实例。

功能限制

- 单节点云盘版实例不支持回档。
- 只支持主实例回档，不支持只读实例和灾备实例回档。
- 目前只支持指定库表回档，回档后的库表会回写到源实例（需要重命名），不支持整实例回档。
- MySQL 5.6、5.7、8.0版本支持回档，MySQL 5.5版本不支持回档，建议 MySQL 5.5版本尽快升级至更高版本，升级请参见 升级数据库引擎版本 。
- 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。

注意事项

- 回档功能与自动备份设置的备份周期时间和保留天数相关联，提供基于保留天数下且备份周期时间内数据备份 + 日志备份（binlog）的回档，备份周期设置请参见 自动备份 MySQL 数据 。为保证数据安全，自动备份设置的备份周期请设置为一周至少备份两次 MySQL。
- 包年包月实例未到期但账号欠费后，备份相关服务会降级，禁止回档数据库，若需进行回档操作，请充值至账号余额为正。
- 如果需要回档的库表不存在或被误删，需要先登录数据库并创建库表，再使用控制台回档。
- 如果回档之前的冷备份没有该表，灾备会失败。

操作步骤

1. 登录 MySQL 控制台，在实例列表，选择一个或多个需要回档的实例，选择**更多操作** > **回档**。

> **说明：**
> - 若只需进行一个实例的回档，也可进入实例管理页，在右上角单击**回档**。
> - 同一个 APPID 下可同时发起最多5个回档任务。

2. 在回档页面，选择回档库表，单击**下一步：设置回档时间和库表名**。

回档方式：
- 普通：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。
- 快速：全量备份+库级别 binlog，如有跨库操作，且关联库未被同时选中，将会导致回档失败。
- 极速：全量备份+表级别 binlog，如有跨表操作，且关联表未被同时选中，将会导致回档失败。

> **说明：**
> - 目前仅支持名称为数字、字母、下划线及其组合的库表回档，库表名为其他特殊字符的暂不支持回档。
> - 在指定库表回档模式下，同一实例同时仅支持500个库或表同时回档。
> - 如果回档执行 binlog 时，涉及其他库表的复合操作，SQL 语句有可能会失败。
> - 如果回档执行 binlog 时，表涉及了外键等约束，SQL 语句有可能会失败。

3. 设置回档后的库表名和回档时间，单击**回档**。

> **说明：**
> - 每个实例只能设置一个回档时间。

- 选择设置批量回档时间，所有库表以批量回档时间为准。
- 选择设置单表回档时间，库表以各自设置的回档时间为准。
- 回档后的库表名仅支持64位以内的英文、数字、小数点（.）、短横线（-）、下划线（_）、$。

4. 提交成功后返回**操作日志** > **回档日志**页，可查看回档进度，单击**查看详情**可实时查看回档日志。
5. 回档完成后，选择**数据库管理** > **数据库列表**页，可在原实例中看到回档后的新库表。

删除备份

最近更新时间：2025-06-03 17:59:22

您可以删除数据库备份，以节约备份空间开销。

> **说明：**
> 自动备份无法手动删除，手动逻辑备份和手动物理备份可在备份列表手动删除，手动快照备份不支持删除。

操作步骤

1. 登录 MySQL 控制台，在实例列表，单击**实例 ID**，进入实例管理页面。
2. 在实例管理页面，选择**备份恢复**页，在备份列表的**操作**列，单击**删除**。

3. 在弹出的对话框，确定要删除备份文件，单击**确定**。

> **注意：**
> 删除后备份文件无法恢复。

克隆实例

最近更新时间：2025-06-03 17:59:22

本文为您介绍如何通过控制台克隆 MySQL 实例，来帮助您快速回档实例至新购的 MySQL 实例。

操作场景

云数据库 MySQL 提供克隆实例功能，支持通过克隆将实例恢复到日志备份保留时间内的任意时间点，并且支持恢复到指定的备份集。克隆会先根据用户所选的时间点并通过备份数据创建一个新实例，新实例经过验证后，用户可再将数据通过 DTS 迁回源实例，或者直接使用克隆的新实例。

克隆方式

- 按时间点克隆：将实例恢复到任意时间点，时间点的选择范围由您设置的日志保留时长决定。
- 按备份集克隆：将实例恢复到任一物理备份的备份集，备份集选择范围由您设置的数据备份保留时长决定。

新实例费用

- 新的克隆实例的计费模式支持在克隆界面选择包年包月或者按量计费，详细计费说明请参见 计费概述 。
- 新的克隆实例在完全克隆成功后，才开始进行计费。

前提条件

- 支持的实例架构：MySQL 单节点、双节点、三节点、云盘版。
- 源实例状态须为**运行中**。
- 若克隆方式选择**按备份集**克隆，源实例必须有已经完成的物理备份或快照备份，备份状态可至 控制台 的备份列表查看。
- 账户内余额须大于0。

功能限制

- 不支持克隆只读实例。
- MySQL 5.6、5.7、8.0版本支持克隆，MySQL 5.5版本不支持克隆，建议 MySQL 5.5版本尽快升级至更高版本，升级请参见 升级数据库引擎版本 。

注意事项

- 新克隆实例规格须大于或等于源实例。
- 新克隆实例硬盘空间大小须大于要克隆的数据大小，否则可能导致克隆失败。
- 新克隆实例的可用区、数据库版本、复制方式、默认数据库参数须与源实例相同。
- 在克隆的过程中，新克隆实例不会在控制台列表展示，克隆成功后才可见。
- 为保障克隆后数据一致性，会自动设置参数 event_scheduler 为 OFF，如需打开，请在克隆完成后在克隆实例的参数列表开启。

操作步骤

1. 登录 MySQL 控制台 ，在实例列表，单击**实例 ID** 或**操作**列的**管理**，进入实例管理页面。
2. 在实例管理页面，选择**备份恢复** > **数据备份列表**页， 在左上角单击**克隆**，或在需要克隆的备份“操作”列单击**克隆**。

3. 进入购买页，选择克隆实例的恢复方式和相应配置，确认无误后，单击**立即购买**。
- 按时间点恢复：可从源数据库实例上恢复出具体到某个时间点的数据库实例，时间点选择范围由您设置的日志备份保留时长所决定。
- 按备份集恢复：可从指定的备份集中恢复出新实例，备份集选择范围由您设置的数据备份保留时长所决定。

> **说明：**
> 备份保留时长可至 控制台 的备份列表查看。

4. 购买成功后，可在**备份恢复** > **克隆列表**页下查看克隆详情。
5. 实例克隆成功后，可在实例列表页查看新购实例的详情。

相关文档

- 数据库单库和单表级别的数据库恢复，请参见 回档数据库 。
- 恢复数据到自建数据库，请参见 使用物理备份恢复数据库 或 使用逻辑备份恢复数据库 。

热点问题

克隆过程中是否会影响源实例的访问？

克隆过程使用原有备份集以及上传至 COS 中的 binlog 进行克隆，不会影响源实例的访问。

数据迁移
使用 DTS 服务迁移

最近更新时间：2023-02-06 17:36:57

数据迁移服务 DTS 支持将本地 IDC、第三方云厂商、腾讯云数据库 MySQL 迁移至 MySQL，同时支持结构迁移、全量数据迁移以及增量数据迁移，可以实现在不停服的情况下，平滑迁移数据到腾讯云数据库 MySQL。

- 不同腾讯云账号下的实例间迁移
- MySQL 迁移至 MySQL
- MySQL 迁移常见问题

导入 SQL 文件

最近更新时间：2024-11-13 15:32:32

操作场景

云数据库 MySQL 支持通过控制台导入 SQL 文件，本功能可以在选择的数据库上执行 SQL 语句，您可以利用此功能创建数据库/表、更改表结构来完成对实例的初始化或者变更。

> **说明：**
> 云数据库 MySQL 仅双节点、三节点实例支持导入 SQL 文件功能。

注意事项

导入 SQL 文件功能，支持直接上传文件和从 COS 下载文件两种方式，如果您需要采用从 COS 下载文件这种方式来导入 SQL 文件到您的云数据库 MySQL，您需要先将文件上传至 COS，才能获取对应的 COS 源文件链接，详细操作请参见 附录1：上传文件到 COS 和获取 COS 源文件链接 。

操作步骤

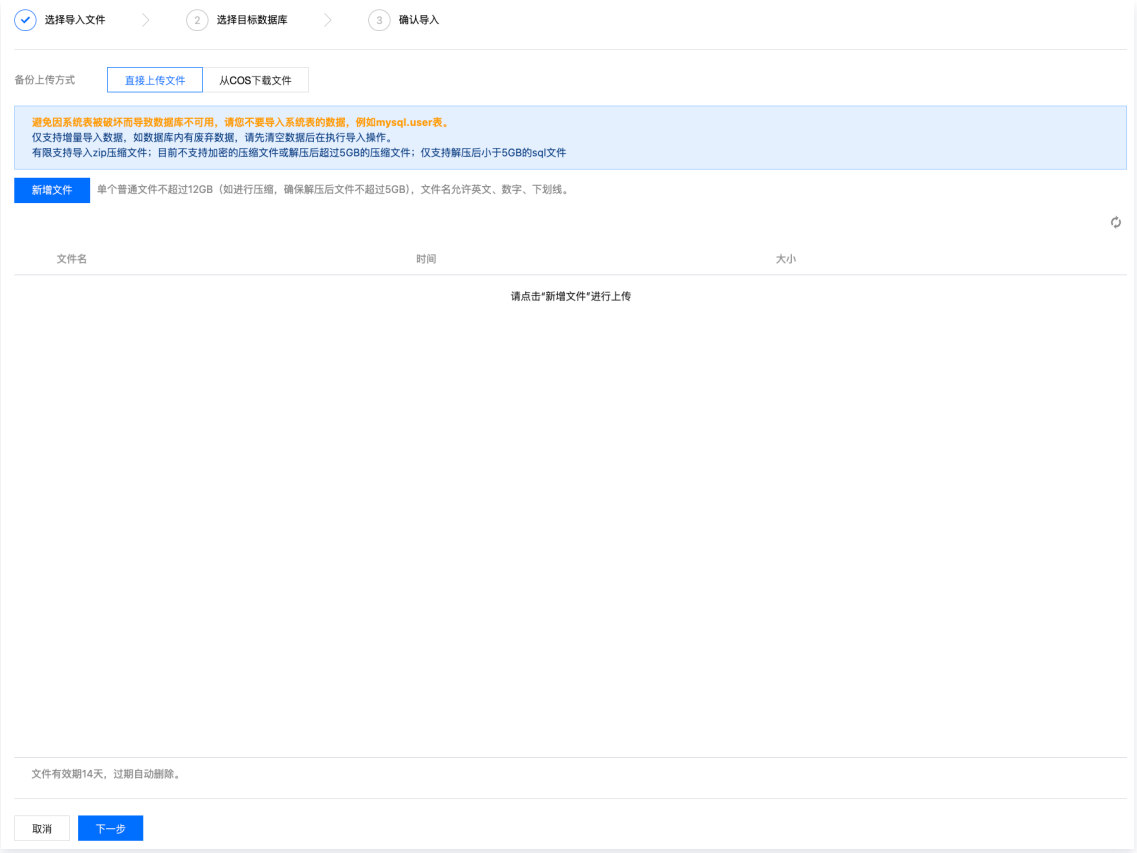
1. 登录 MySQL 控制台 ，在实例列表，单击实例 ID 或**操作**列的**管理**，进入实例管理页面。
2. 在实例管理页面，选择**数据库管理** > **数据库列表**页，单击**数据导入**。

3. 根据需要选择如下任意一种方式进行操作。

方式1：直接上传文件

> **说明：**
> - 避免因系统表被破坏而导致数据库不可用，请勿导入系统表的数据，例如 mysql.user 表。
> - 仅支持增量导入数据，如数据库内有废弃数据，请先清空数据后再执行导入操作。
> - 有限支持导入 zip 压缩文件；目前不支持加密的压缩文件或解压后超过5GB的压缩文件；仅支持解压后小于5GB的 SQL 文件。
> - 单个普通文件不超过12GB（如进行压缩，确保解压后文件不超过5GB），文件名允许英文、数字、下划线。
> - 上传的文件有效期为14天，过期会自动删除。

在弹窗下，单击**新增文件**导入文件，上传完成后，单击**下一步**。



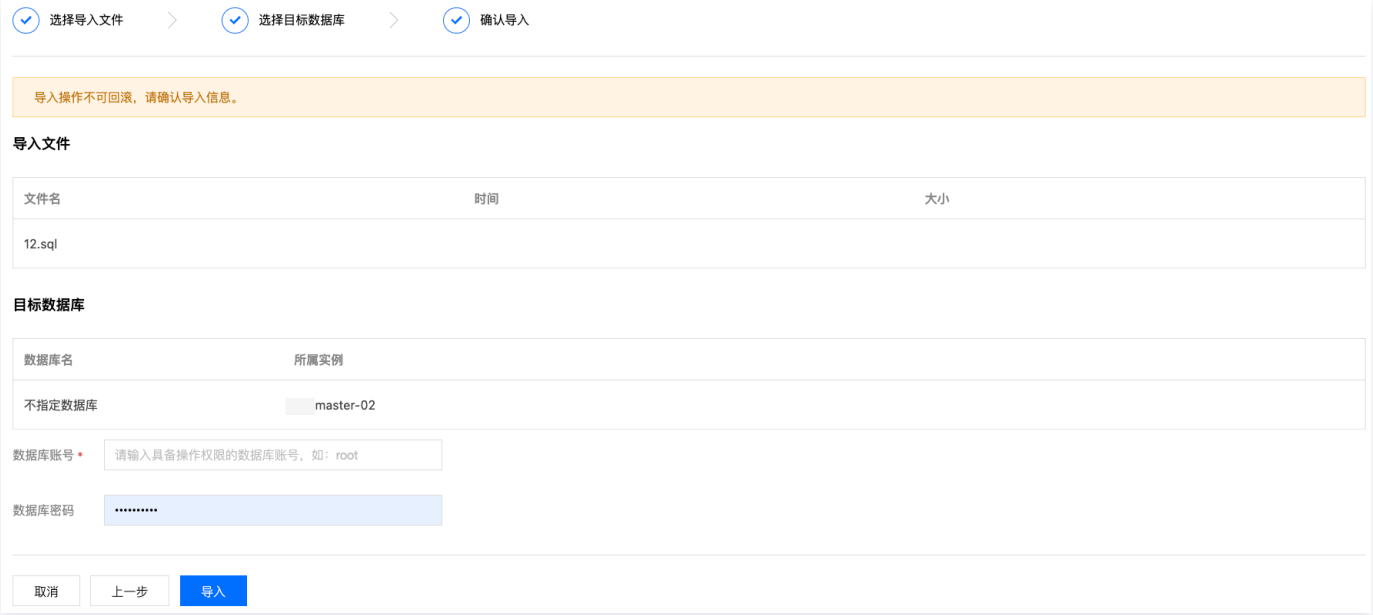
方式2：从 COS 下载文件

在弹窗下，粘贴 COS 源文件链接，单击**下一步**。

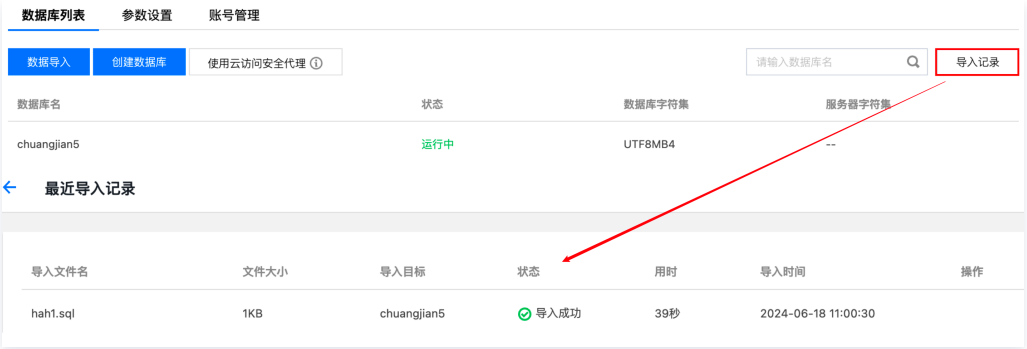
> **说明：**
> 需要先将文件上传至 COS，才能获取对应的 COS 源文件链接，详细操作请参见 附录1：上传文件到 COS 和获取 COS 源文件链接 。

4. 在选择目标数据库页，选择目标数据库（也支持不指定数据库），单击**下一步**。
5. 在确认页，确认导入数据无误后，输入账号密码，单击**导入**。

> **注意：**
> - 导入操作不可回滚，请确认好导入信息。
> - 如忘记账号密码，可参见 重置密码 修改密码。



6. 任务执行后，文件导入的状态（是否导入成功），需要从导入记录中查询。您可在数据库列表右侧单击**导入记录**，跳转至**最近导入记录**页面进行查询。



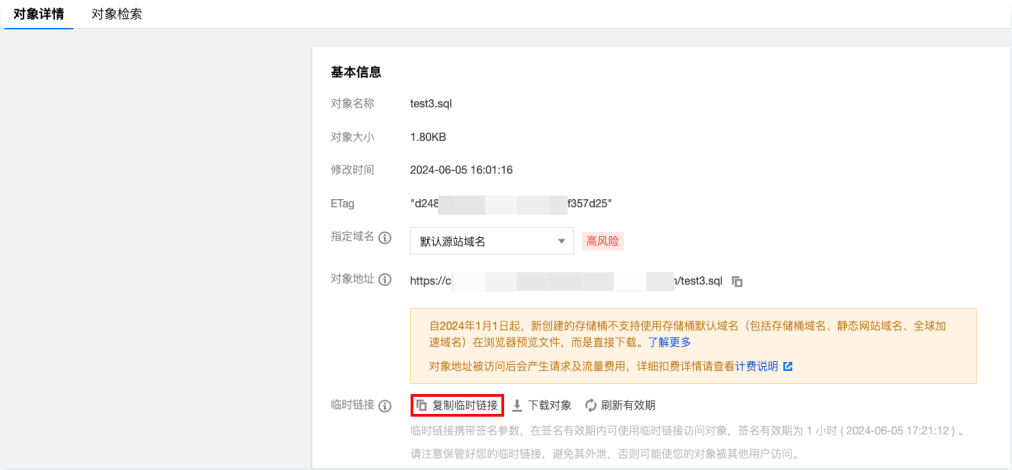
附录1：上传文件到 COS 和获取 COS 源文件链接

1. 登录 COS 控制台。
2. 在左侧导航选择**存储桶列表**页，单击**创建存储桶**。
3. 在弹出的创建对话框，配置对应信息，单击**创建**。

> **说明：**
> 存储桶地域需要和云数据库 MySQL 实例地域相同。

4. 返回存储桶列表，单击存储桶名或**操作**列的**配置管理**。
5. 选择**文件列表**页，单击**上传文件**，可以选择单个或多个文件上传。
6. 文件上传完后，在文件列表找到目标文件，单击其右上角的三个点，然后单击**详情**。

7. 在对象详情页面的基本信息中单击**复制临时链接**，即可获取该文件的 COS 源文件链接。



常见问题

导入 SQL 文件失败，提示“发起导入失败（InternalError.UndefinedError）”

数据导入功能的设置步骤分为：选择导入文件、选择目标数据库、确认导入。其中在确认导入环节，需要输入当前实例对应的账号和密码，若输入错误，则会提示发起导入失败。请在确认导入环节，输入正确的账号和密码，重新导入，若忘记密码，可 重置密码 后重试。

离线迁移数据

最近更新时间：2025-05-23 09:25:22

本文为您介绍如何通过控制台和命令行工具两种方案迁移数据。

控制台迁移数据

通过控制台迁移数据有物理备份和逻辑备份两种方式，详细操作请参见：

- 使用物理备份恢复数据库
- 使用逻辑备份恢复数据库

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数据库 MySQL 版本的 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。
> - 请保证源和目标数据库版本一致，源和目标数据库字符集一致，mysqldump 工具的版本一致。您可以通过参数 --default-character-set 来指定字符集。
> - 若导入数据时失败，可能是权限问题，可以在生成导入文件时，通过参数 --set-gtid-purged=OFF 来生成导入文件。

```
shell > mysqldump <options> <db_name> <tbl_name ...> > <bak_pathname>
```

   - <options>：为导出选项或参数，常用的选项包括：-h，-u，-p 等，常用的参数例如：--default-character-set。
   - <db_name>：为数据库名称，需要替换需要的数据库。
   - <tbl_name>：为表名称，需要替换为需要的数据库下的表名。
   - <bak_pathname>：为导出路径名，替换为实际的生成文件路径。

   更多 mysqldump 导出数据说明，请参见 MySQL 官方手册 。
2. 通过 MySQL 命令行工具导入数据至目标数据库，方式如下：

```
shell > mysql -h <hostname> -P <port> -u <username> -p < <bak_pathname>
```

   - <hostname>：替换为还原数据的目标主机。
   - <port>：替换为目标主机的端口。
   - <username>：替换为目标主机的数据库用户名。
   - <bak_pathname>：替换为备份文件的完整路径。

迁移数据（Windows 系统）

1. 使用 Windows 系统的 mysqldump 工具生成待导入的 SQL 文件，具体请参考 命令行工具迁移数据 中的描述。
2. 进入命令提示符，通过 MySQL 命令行工具导入数据至目标数据库。

3. 登录目标 MySQL 数据库，执行 `show databases;` 命令，可查看到备份的数据库已导入至目标数据库。

```
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| db_blog            |
| la                 |
| lailailai          |
| lalalallalalaa     |
| mysql              |
| performance_schema |
| sakila             |
| test               |
| world              |
+--------------------+
10 rows in set (0.01 sec)
```

迁移数据（Linux 系统）

本文以 Linux 系统的 CVM 为例，从 CVM 访问数据库请参见 访问 MySQL 数据库 。

1. 登录 CVM，使用 MySQL 命令行工具 mysqldump 生成待导入的 SQL 文件。以云数据库上的 db_blog 数据库为例。

```
Send CtrlAltDel        连接成功：如果长时间处于黑屏状态，请按任意键唤醒。如需粘贴命令，请点击这里
[root@VM_74_55_centos home]# mysqldump -h [redacted] -P [redacted] -u root -p db_blog > /home/db_blog.bak
Enter password:
[root@VM_74_55_centos home]# ls /home
db_back  db_blog.bak
[root@VM_74_55_centos home]#
```

2. 通过 MySQL 命令行工具，将数据还原至目标数据库。

3. 登录目标 MySQL 数据库，执行 `show databases;` 命令，可查看到备份的数据库已导入至目标数据库。

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 682944
Server version: 8.0.22-txsql 20220831

Copyright (c) 2000, 2022, 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.

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

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| copylinetest       |
| employees          |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| db_blog            |
| usertestddl        |
+--------------------+
8 rows in set (0.01 sec)

mysql>
```

导入数据文件字符集编码问题

1. 云数据库导入数据文件如果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。

热点问题

使用数据导入功能出现报错：wait err=exit status 255, busi errmsg=import sql file failed, err=ERROR 1227 (42000) at line 24: Access denied; 如何处理?

使用数据导入功能出现42000报错，表示在导入 SQL 文件时遇到了访问被拒绝的错误，导致导入 SQL 文件失败，可能是当前用户没有足够的权限，建议通过如下两种方式进行处理：

- 使用 MySQL 命令行工具 mysqldump 重新生成待导入的 SQL 文件，引用参数：--set-gtid-purged=OFF，然后再重新尝试导入。生成导入 SQL 文件命令请参考 命令行工具迁移数据 。
- 您也可以根据报错提示，手动注释相关行以规避导入的权限问题，示例如下：

```
SET @MYSOLDUMP TEMP LOG BIN = COSESSION.SQL LOG BIN;
SET @@SESSION.SQL LOG BIN = 0;
-- GTID state at the beginning of the backup
SET @@GLOBAL.GTID PURGED=/*!80000 '+'*/
```

网络与安全
访问管理
访问管理概述

最近更新时间：2024-10-30 21:37:12

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。
- 您无法限制其他人的访问权限，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 策略的更多基本信息，请参见 策略语法 。

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

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

任务	链接
了解策略基本结构	策略语法
在策略中定义操作	云数据库的操作
在策略中定义资源	云数据库的资源路径
云数据库支持的资源级权限	云数据库支持的资源级权限
控制台示例	控制台示例

授权策略语法

最近更新时间：2024-10-18 14:14:23

策略语法

CAM 策略：

```
{
        "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（以 cdb: 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。

云数据库的操作

在云数据库策略语句中，您可以从支持云数据库的任何服务中指定任意的 API 操作。对于云数据库，请使用以 cdb: 为前缀的 API 。例如 cdb:CreateDBInstance 或者 cdb:CreateAccounts 。
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":["cdb:action1","cdb: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":["cdb:Describe*"]
```

如果您要指定云数据库中所有操作，请使用 * 通配符，如下所示：

```
"action": ["cdb:*"]
```

云数据库的资源

每个 CAM 策略语句都有适用于自己的资源。
资源的一般形式如下：

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

- **project_id**：描述项目信息，仅为了兼容 CAM 早期逻辑，无需填写。
- **service_type**：产品简称，如 cdb。

- **region**：地域信息，如 ap-guangzhou。
- **account**：资源拥有者的主账号信息，如 uin/65xxx763。
- **resource**：各产品的具体资源详情，如 instanceId/instance_id1 或者 instanceId/*。

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

```
"resource":[ "qcs::cdb:ap-guangzhou:uin/65xxx763:instanceId/cdb-k05xdcta"]
```

您还可以使用 * 通配符指定属于特定账户的所有实例，如下所示：

```
"resource":[ "qcs::cdb:ap-guangzhou:uin/65xxx763:instanceId/*"]
```

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"]
```

下表描述了云数据库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
实例	`qcs::cdb:$region:$account:instanceId/$instanceId`
VPC	`qcs::vpc:$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：2025-05-23 09:25:22

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。云数据库部分支持资源级权限，即表示针对支持资源级权限的云数据库操作，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 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
云数据库实例相关	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`

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

支持资源级授权的 API 列表

API 操作	资源路径
AddTimeWindow	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
AssociateSecurityGroups	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
CloseWanService	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
CreateAccounts	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
CreateBackup	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
CreateDBImportJob	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
DeleteAccounts	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
DeleteBackup	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
DeleteTimeWindow	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
DescribeAccountPrivileges	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
DescribeAccounts	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
DescribeBackupConfig	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
DescribeBackupDatabases	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
DescribeBackupDownloadDbTableCode	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
DescribeBackups	`qcs::cdb:$region:$account:instanceId/*`

	qcs::cdb:$region:$account:instanceId/$instanceId
DescribeBackupTables	qcs::cdb:$region:$account:instanceId/* qcs::cdb:$region:$account:instanceId/$instanceId
DescribeBinlogs	qcs::cdb:$region:$account:instanceId/* qcs::cdb:$region:$account:instanceId/$instanceId
DescribeDatabases	qcs::cdb:$region:$account:instanceId/* qcs::cdb:$region:$account:instanceId/$instanceId
DescribeDBImportRecords	qcs::cdb:$region:$account:instanceId/* qcs::cdb:$region:$account:instanceId/$instanceId
DescribeDBInstanceCharset	qcs::cdb:$region:$account:instanceId/* qcs::cdb:$region:$account:instanceId/$instanceId
DescribeDBInstanceConfig	qcs::cdb:$region:$account:instanceId/* qcs::cdb:$region:$account:instanceId/$instanceId
DescribeDBInstanceGTID	qcs::cdb:$region:$account:instanceId/* qcs::cdb:$region:$account:instanceId/$instanceId
DescribeDBInstanceRebootTime	qcs::cdb:$region:$account:instanceId/* qcs::cdb:$region:$account:instanceId/$instanceId
DescribeDBSwitchRecords	qcs::cdb:$region:$account:instanceId/* qcs::cdb:$region:$account:instanceId/$instanceId
DescribeDBSecurityGroups	qcs::cdb:$region:$account:instanceId/* qcs::cdb:$region:$account:instanceId/$instanceId
DescribeInstanceParamRecords	qcs::cdb:$region:$account:instanceId/* qcs::cdb:$region:$account:instanceId/$instanceId
DescribeInstanceParams	qcs::cdb:$region:$account:instanceId/* qcs::cdb:$region:$account:instanceId/$instanceId
DescribeRoGroups	qcs::cdb:$region:$account:instanceId/* qcs::cdb:$region:$account:instanceId/$instanceId
DescribeRollbackRangeTime	qcs::cdb:$region:$account:instanceId/* qcs::cdb:$region:$account:instanceId/$instanceId
DescribeSlowLogs	qcs::cdb:$region:$account:instanceId/* qcs::cdb:$region:$account:instanceId/$instanceId
DescribeSupportedPrivileges	qcs::cdb:$region:$account:instanceId/* qcs::cdb:$region:$account:instanceId/$instanceId
DescribeTables	qcs::cdb:$region:$account:instanceId/* qcs::cdb:$region:$account:instanceId/$instanceId
DescribeTimeWindow	qcs::cdb:$region:$account:instanceId/* qcs::cdb:$region:$account:instanceId/$instanceId
DescribeDatabasesForInstances	qcs::cdb:$region:$account:instanceId/* qcs::cdb:$region:$account:instanceId/$instanceId
DescribeMonitorData	qcs::cdb:$region:$account:instanceId/* qcs::cdb:$region:$account:instanceId/$instanceId

版权所有：腾讯云计算（北京）有限责任公司

DescribeTableColumns	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
DropDatabaseTables	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
IsolateDBInstance	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
ModifyAccountDescription	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
ModifyAccountPassword	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
ModifyAccountPrivileges	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
ModifyAutoRenewFlag	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
ModifyBackupConfig	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
ModifyBackupInfo	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
ModifyDBInstanceName	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
ModifyDBInstanceProject	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
ModifyDBInstanceSecurityGroups	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
ModifyDBInstanceVipVport	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
ModifyInstanceParam	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
ModifyDBInstanceModes	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
ModifyTimeWindow	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
ModifyProtectMode	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
OfflineDBInstances	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
OpenDBInstanceGTID	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
OpenWanService	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
ReleaseIsolatedDBInstances	`qcs::cdb:$region:$account:instanceId/*`

	`qcs::cdb:$region:$account:instanceId/$instanceId`
RestartDBInstances	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
StartBatchRollback	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
SubmitBatchOperation	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
SwitchDrInstanceToMaster	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
SwitchForUpgrade	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
DisassociateSecurityGroups	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
UpgradeDBInstance	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`
UpgradeDBInstanceEngineVersion	`qcs::cdb:$region:$account:instanceId/*` `qcs::cdb:$region:$account:instanceId/$instanceId`

不支持资源级授权的 API 列表

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 *。

API 操作	API 描述
CreateDBInstance	创建云数据库实例（包年包月）
CreateDBInstanceHour	创建云数据库实例（按量计费）
CreateParamTemplate	创建参数模板
DeleteParamTemplate	删除监控模板监控项
DescribeProjectSecurityGroups	查询项目安全组信息
DescribeDefaultParams	查询默认的可设置参数列表
DescribeParamTemplateInfo	查询参数模板详情
DescribeParamTemplates	查询参数模板列表
DescribeAsyncRequestInfo	查询异步任务的执行结果
DescribeTasks	查询云数据库实例任务列表
DescribeUploadedFiles	查询导入 SQL 文件列表
ModifyParamTemplate	修改参数模板
RenewDBInstance	续费云数据库实例
StopDBImportJob	终止数据导入任务
DescribeRoMinScale	查询只读实例支持的最小规格
DescribeRequestResult	查询任务详情
DescribeRoMinScale	获取只读实例购买或升级的最小规格

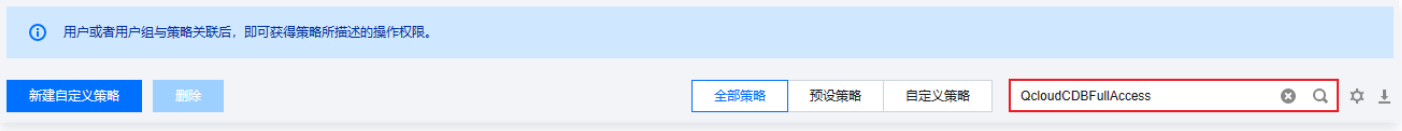
控制台示例

最近更新时间：2024-10-18 14:14:23

您可以使用访问管理 CAM 策略让用户拥有在云数据库控制台中查看和使用特定资源的权限。该部分的示例能够使用户使用控制台的特定部分的策略。

云数据库的全读写策略

如果您想让用户拥有创建和管理云数据库实例的权限，您可以对该用户使用名称为：QcloudCDBFullAccess 的策略。
进入 策略管理 界面，在右上角搜索框搜索 QcloudCDBFullAccess，即可找到该策略。



策略语法如下：

```
{
    "statement": [
        {
            "action": [
                "cdb:*"
            ],
            "effect": "allow",
            "resource": "*"
        },
        {
            "action": [
                "vpc:*"
            ],
            "effect": "allow",
            "resource": "*"
        },
        {
            "action": [
                "cvm:*"
            ],
            "effect": "allow",
            "resource": "qcs::cvm:::sg/*"
        },
        {
            "action": [
                "cos:*"
            ],
            "effect": "allow",
            "resource": "*"
        },
        {
            "action": [
                "monitor:*",
                "cam:ListUsersForGroup",
                "cam:ListGroups",
                "cam:GetGroup"
            ],
            "effect": "allow",
            "resource": "*"
        },
        {
            "action": [
                "kms:CreateKey",
```

```
                "kms:GenerateDataKey",
                "kms:Decrypt",
                "kms:ListKey"
            ],
            "effect": "allow",
            "resource": "*"
        }
    ],
    "version": "2.0"
}
```

以上策略是通过让用户分别对云数据库、私有网络、安全组、对象存储、密钥管理服务和 Monitor 中所有资源进行 CAM 策略授权来达到目的。

云数据库的只读策略

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

> **说明：**
> 建议配置云数据库的只读策略。

您可以进入 策略管理 界面，单击列项**服务类型**在下拉选项中选择**云数据库 MySQL**，就可以在结果中找到该策略。
策略语法如下：

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

云数据库相关资源的只读策略

如果您想要让用户只拥有查询云数据库实例及相关资源（私有网络、安全组、对象存储、Monitor）的权限，但不允许该用户拥有创建、删除和修改等操作的权限，您可以对该用户使用名称为：QcloudCDBReadOnlyAccess 的策略。
您可以进入 策略管理 界面，单击列项**服务类型**在下拉选项中选择**云数据库 MySQL**，就可以在结果中找到该策略。
策略语法如下：

```
{
    "statement": [
        {
            "action": [
                "cdb:Describe*"
            ],
            "effect": "allow",
            "resource": "*"
        },
        {
            "action": [
                "monitor:Describe*",
                "monitor:Get*",
                "cam:ListUsersForGroup",
                "cam:ListGroups",
```

```
                "cam:GetGroup"
            ],
            "effect": "allow",
            "resource": "*"
        }
    ],
    "version": "2.0"
}
```

以上策略是通过让用户分别对如下操作进行 CAM 策略授权来达到目的：

- 云数据库中所有以单词“Describe”开头的操作。
- 私有网络中所有以单词“Describe”开头的操作、所有以单词“Inquiry”开头的操作和所有以单词“Get”开头的操作。
- 安全组中所有以单词“DescribeSecurityGroup”开头的所有操作。
- 对象存储中所有以单词“List”开头的操作、所有以单词“Get”开头的操作、所有以单词“Head”开头的操作和名为“OptionsObject”的操作。
- Monitor 中所有的操作。

授权用户拥有非资源级的 API 接口的操作权限策略

如果您想要让用户只拥有非资源级的 API 接口的操作权限，您可以对该用户使用名称为：QcloudCDBProjectToUser 的策略。
进入 策略管理 界面，单击列项**服务类型**在下拉选项中选择**云数据库 MySQL**，就可以在结果中找到该策略。
策略语法如下：

```
{
    "version": "2.0",
    "statement": [
        {
            "action": [
                "cdb:BalanceRoGroupLoad",
                "cdb:CancelBatchOperation",
                "cdb:CreateBatchJobFiles",
                "cdb:CreateDBInstance",
                "cdb:CreateDBInstanceHour",
                "cdb:CreateMonitorTemplate",
                "cdb:CreateParamTemplate",
                "cdb:DeleteBatchJobFiles",
                "cdb:DeleteMonitorTemplate",
                "cdb:DeleteParamTemplate",
                "cdb:DescribeBatchJobFileContent",
                "cdb:DescribeBatchJobFiles",
                "cdb:DescribeBatchJobInfo",
                "cdb:DescribeProjectSecurityGroups",
                "cdb:DescribeDefaultParams",
                "cdb:DescribeMonitorTemplate",
                "cdb:DescribeParamTemplateInfo",
                "cdb:DescribeParamTemplates",
                "cdb:DescribeRequestResult",
                "cdb:DescribeRoGroupInfo",
                "cdb:DescribeRoMinScale",
                "cdb:DescribeTasks",
                "cdb:DescribeUploadedFiles",
                "cdb:ModifyMonitorTemplate",
                "cdb:ModifyParamTemplate",
                "cdb:ModifyRoGroupInfo",
                "cdb:ModifyRoGroupVipVport",
                "cdb:StopDBImportJob",
                "cdb:UploadSqlFiles"
            ],
            "effect": "allow",
            "resource": "*"
```

```
        }
    ]
}
```

授权用户拥有特定云数据库的操作权限策略

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

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

授权用户拥有批量云数据库的操作权限策略

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

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

授权用户拥有特定地域云数据库的操作权限策略

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

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

自定义策略

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

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

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

管理云数据库安全组

最近更新时间：2025-05-21 14:17:22

操作场景

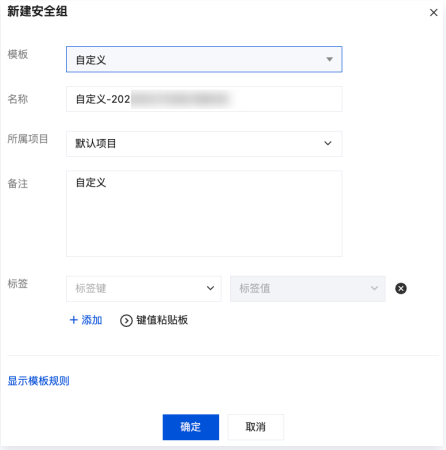
安全组 是一种有状态的包含过滤功能的虚拟防火墙，用于设置单台或多台云数据库的网络访问控制，是腾讯云提供的重要的网络安全隔离手段。安全组是一个逻辑上的分组，您可以将同一地域内具有相同网络安全隔离需求的云数据库实例加到同一个安全组内。云数据库与云服务器等共享安全组列表，安全组内基于规则匹配，具体规则与限制请参见 安全组详细说明 。

> **说明：**
> - 云数据库 MySQL 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。
> - 云数据库 MySQL 安全组支持主实例（包括读写节点）、只读实例（包括只读节点）与灾备实例。

为云数据库配置安全组

步骤一：创建安全组

1. 登录 云服务器控制台 。
2. 在左侧导航选择**安全组**页，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**。



- 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。如下表所示：

模板	说明	说明
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端口和 ICMP 协议，内网全放通。	此模板对云数据库不生效。
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 添加安全组规则 。	–

- 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，便于后期管理。
- 备注：自定义，简短地描述安全组，便于后期管理。
- 标签：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 。

步骤二：添加安全组规则

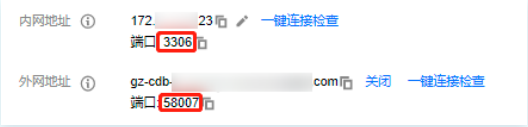
1. 在 安全组页，在需要设置规则的安全组行中，单击**操作**列的**修改规则**。

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，推荐选择 MySQL(3306) 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：

指定的源/目标	说明
单个 IPv4 地址或 IPv4 地址范围	用 CIDR 表示法（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）。
单个 IPv6 地址或 IPv6 地址范围	用 CIDR 表示法（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）。
引用安全组 ID，您可以引用以下安全组的 ID： • 安全组 ID • 其他安全组	当前安全组表示与安全组关联的云服务器。 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。
引用 参数模板 中的 IP 地址对象或 IP 地址组对象	–

- 协议端口：填写协议类型和端口范围，您也可以引用 参数模板 中的协议端口或协议端口组。

> **注意：**
> 连接云数据库 MySQL，须放通 MySQL 实例端口。您可登录 MySQL 控制台，单击**实例 ID** 进入详情页查看端口。
>
> 
>
>
> - MySQL 内网默认端口为3306，同时支持自定义端口，若修改过默认端口号，安全组中需放通 MySQL 新端口信息。
> - MySQL 外网端口由系统自动分配，不支持自定义，外网开启后将受到安全组网络访问策略的控制，配置安全策略时需放通内网访问端口3306。
> - MySQL 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。

- 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**确定**，完成安全组入站规则的添加。

案例

场景：您创建了一台云数据库 MySQL，并希望通过云服务器 CVM 访问云数据库 MySQL。
解决方法：添加安全组规则时，在**类型**中选择 MySQL(3306)，开通3306号协议端口。
您还可以根据实际需求，放通全部 IP 或指定 IP（IP 段），配置可通过 CVM 访问云数据库 MySQL 的 IP 来源。

方向	类型	来源	协议端口	策略
入方向	MySQL(3306)	全部 IP：0.0.0.0/0	TCP:3306	允许

		指定 IP：输入您指定的 IP 或 IP 段		

步骤三：配置安全组

安全组是腾讯云提供的实例级别防火墙，可以对云数据库进行入流量控制。您可以在购买实例时绑定安全组，也可以购买实例后在控制台绑定安全组。

> **注意：**
> 目前云数据库 MySQL 安全组仅支持**私有网络云数据库**配置。

1. 登录 MySQL 控制台，在实例列表，单击**实例 ID**，进入实例管理页面。
2. 在实例管理页面，选择**安全组**页，单击**配置安全组**。

3. 在弹出的对话框，选择需要绑定的安全组，单击**确定**，即可完成安全组绑定云数据库的操作。

导入安全组规则

1. 在 安全组页，选择需要的安全组，单击安全组 ID/名称。
2. 在入/出站规则页签，单击**导入规则**。

3. 在弹出的对话框，选择已编辑好的入站/出站规则模板文件，单击**开始导入**。

> **说明：**
> 如果需要导入规则的安全组下已存在安全组规则，建议您先导出现有规则，否则导入新规则时，将覆盖原有规则。

克隆安全组

1. 在 安全组页，在列表的**操作**列选择**更多** > **克隆**。
2. 在弹出的对话框，选定目标地域、目标项目后，单击**确定**。若新安全组需关联 CVM，请重新进行管理安全组内云服务器。

删除安全组

1. 在 安全组页，选择需要删除的安全组，在**操作**列选择**更多** > **删除**。
2. 在弹出的对话框，单击**确定**。若当前安全组有关联的 CVM 则需要先解除安全组才能进行删除。

切换网络

最近更新时间：2025-04-30 10:00:44

操作场景

腾讯云网络分为 基础网络和私有网络 VPC，给用户带来不同的优质服务。在此基础之上，我们提供如下更灵活的服务，方便您管理网络。

- **网络间切换**
  - 基础网络切换为私有网络：支持单台云数据库主实例的基础网络切换至私有网络。
  - 私有网络 A 切换为私有网络 B：支持单台云数据库主实例的私有网络 A 切换至私有网络 B。
- **设置自定义 IP 端口**
  - 自定义主实例 IP：支持在实例详情页自定义主实例 IP 和端口。
  - 自定义只读实例 IP：支持在实例详情页自定义只读实例 IP 和端口。

> **说明：**
> 云盘版实例具备1个读写节点和最多5个只读节点，其切换网络和修改端口的操作略有不同，详细可参见 读写地址管理 和 只读地址管理 。

注意事项

- 将基础网络切换为私有网络后，仅相同私有网络下客户端可以互通。可以通过 配置 私有网络网段，将私有网络 IP 保持与基础网络 IP 地址不变。
- 旧 IP 地址保留时间默认为24小时，最长保留时间支持168小时，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。
- 基础网络切换私有网络后不可逆，云数据库切换至私有网络后与其他私有网络及基础网络的云服务不互通。
- 若切换的云数据库为主实例，有挂载只读实例或灾备实例，当主实例切换网络后，挂载的只读实例或灾备实例不会随主实例自动切换网络，需手动进行网络切换。

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。
- 选择网络后，子网默认展示为所选实例同一可用区的子网 IP，您也可以自定义选择同地域其他可用区的子网 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

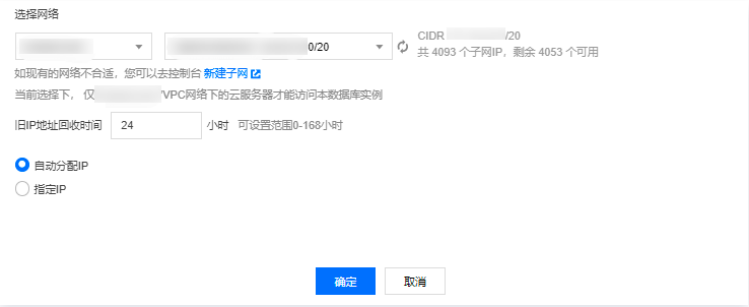
切换网络

1. 登录 MySQL 控制台，在实例列表，单击**实例 ID** 或**操作**列的**管理**，进入实例详情页面。
2. 在实例信息的**所属网络**后，根据网络间切换类型，单击**转 VPC 网络**或**更换网络**。
3. 在弹出的对话框，选择私有网络以及相应子网，单击**确定**。

> **说明：**
> - 若未指定 IP 地址，系统将自动分配。
> - 目的 VPC 只能选择 MySQL 实例所在地域的 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。
> - 建议选择云服务器所在地域的 VPC，否则云服务器无法通过内网访问 MySQL（除非在两个 VPC 之间创建 对等连接 或 云联网 ）。

- **基础网络转私有网络**

- **私有网络转私有网络**



4. 返回实例详情页，可查看到实例的所属网络。

> **说明：**
> - 若旧 IP 地址回收时间设置为0小时，则更换网络后会立即回收旧 IP 地址。
> - 若旧 IP 地址回收时间设置为非0小时，则更换网络后旧 IP 地址会在设置的时间后进行回收，如需提前释放旧 IP 地址，可在**实例详情**页 > **基本信息** > **内网地址**后，单击保留地址，然后对目标地址操作立即释放。
>
>
>
>
>
>

RO 组切换网络

1. 登录 MySQL 控制台，在实例列表，单击**实例 ID** 或**操作**列的**管理**，进入实例详情页面。
2. 在只读实例的 **RO 组信息**中，根据网络间切换类型（基础网络转 VPC / VPC 转 VPC），单击**更换子网**或**转 VPC 网络**。

3. 在弹出的对话框，选择私有网络以及子网，单击**确定**。

> **说明：**
> - 若未指定 IP 地址，系统将自动分配。
> - 目的 VPC 只能选择 MySQL 实例所在地域的 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。
> - 建议选择云服务器所在地域的 VPC，否则云服务器无法通过内网访问 MySQL（除非在两个 VPC 之间创建 对等连接 或 云联网 ）。

4. 系统切换完成后，即可在只读实例页查看 RO 组的所属网络。

> **说明：**
> - 若旧 IP 地址回收时间设置为0小时，则更换网络后会立即回收旧 IP 地址。
> - 若旧 IP 地址回收时间设置为非0小时，则更换网络后旧 IP 地址会在设置的时间后进行回收，如需提前释放旧 IP 地址，可在**只读实例**页找到目标 RO 组，在**内网地址**后单击保留地址，然后对目标地址操作立即释放。

自定义 IP 端口

1. 登录 MySQL 控制台，在实例列表，单击**实例 ID** 或**操作**列的**管理**，进入实例详情页面。

2. 在实例基本信息的内网地址、端口后，单击 。

> **注意：**
> 修改内网地址和端口会影响正在访问的数据库业务。

3. 在弹出的对话框，自定义 IP、端口，确认无误后，单击**确定**。

开启外网连接地址

最近更新时间：2025-06-03 17:59:22

云数据库 MySQL 支持开启外网连接地址，开启后，可方便日常测试和管理，提升用户开发和使用的便利性。本文为您介绍外网地址的相关说明，以及如何通过云数据库 MySQL 控制台，手动开启或关闭外网地址。
开启外网地址后，您可以使用系统分配的域名和端口来实现外网访问 MySQL 实例，相关操作可参见 通过外网地址连接数据库 。

> **说明：**
> - 外网访问出现的故障，不会计入云数据库 MySQL 服务的整体可用性计算。
> - 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。
> - 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。

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**，云数据库 MySQL 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。 计费信息（数据库产品创建实例，此实例用户无需支付费用） 实例计费模式 按量计费 网络计费模式 按流量计费

三、注意事项

- **目前云数据库 MySQL 实例开启外网后为 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。

- 云数据库 MySQL 实例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，您可在对云数据库 MySQL 实例配置安全组时，**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 管理云数据库安全组 。

- 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 告警策略，策略类型选择如下图所示。

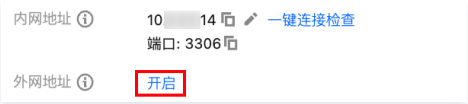
使用限制

目前广州、上海、北京、成都、重庆、南京、中国香港、新加坡、首尔、东京、硅谷、弗吉尼亚、法兰克福地域的主实例，支持开启外网。只读实例支持开启外网的地域，请以控制台为准。若您的实例所在地域不支持开启外网但需外网访问服务，请参见 通过 CLB 开启外网服务 实现外网访问云数据库 MySQL 实例。

开启外网地址

观看视频

1. 登录 MySQL 控制台，在实例列表，单击**实例 ID** 或**操作**列的**管理**，进入实例详情页面。
2. 在实例详情页下的**外网地址**处，单击**开启**。



3. 在弹出的对话框，单击**确定**。

> **说明：**
> 开启成功后，您可以使用系统分配的域名和端口通过外网访问云数据库 MySQL，生效时间大概需要5分钟。

关闭外网地址

> **说明：**
> - 关闭后，您将无法通过外网域名和端口访问云数据库 MySQL，且该域名会被释放，再次申请后无法保证域名保持不变，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。
> - 关闭外网时，支持该外网能力的 CLB（免费）实例将在3 – 5分钟后同步下线。

1. 登录 MySQL 控制台，在实例列表，单击**实例 ID** 或**操作**列的**管理**，进入实例详情页面。
2. 在实例详情页下的**外网地址**后面单击**关闭**。
3. 在弹出的对话框，确认无误后，单击**确定**。
4. 返回实例详情页，**外网地址**处不再显示外网地址，外网功能即关闭。

通过 CLB 开启外网服务

最近更新时间：2025-05-21 14:17:22

云数据库 MySQL 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

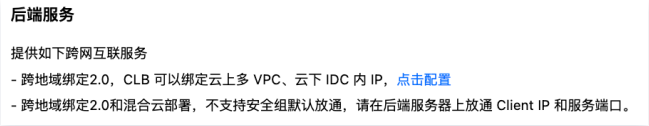
> **说明：**
> 如果在云数据库 MySQL 同地域已经有负载均衡实例，就可以不用购买。

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：**
> - 地域需选择云数据库 MySQL 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 MySQL 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 MySQL 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，然后单击右侧出现的**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 MySQL 实例的内网 IP 地址和端口，单击**确认**完成绑定。

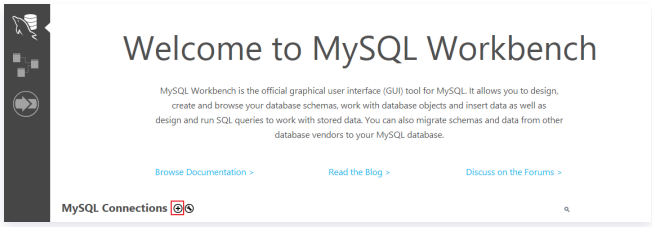
> **注意：**
> 登录的账号必须是标准账号（带宽上移），如无法绑定，请 提交工单 协助处理。

步骤4：配置 MySQL 安全组

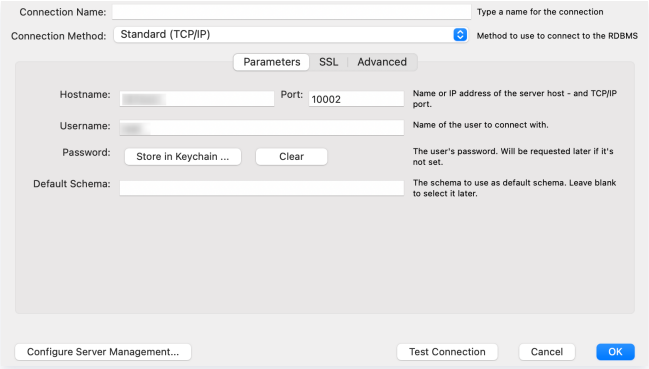
1. 登录 MySQL 控制台，选择地域，在实例列表，单击**实例 ID** 或**操作**列的**管理**，进入实例管理页面。
2. 在实例管理页面，选择**安全组**页，单击**配置安全组**，配置安全组规则为放通全部端口，确认安全组允许外部 IP 访问，详细配置方法请参见 配置安全组 。

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。

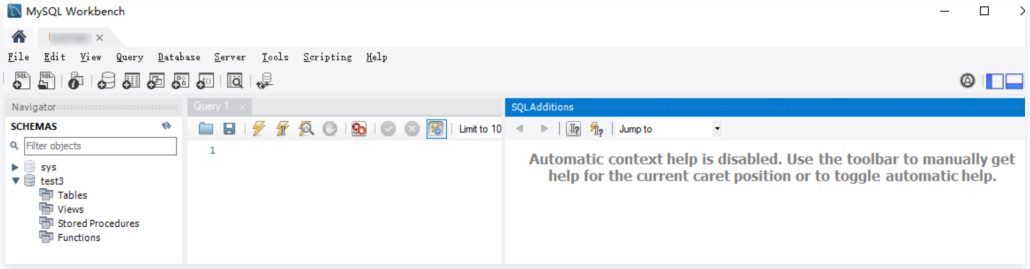


6. 在弹出的窗口下，完成如下配置后，单击 **OK**。



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 MySQL 实例的账号名。在实例管理页 > 数据库管理 > 账号管理下创建的账号。
Password	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 MySQL 实例。
8. 成功连接后的界面如下所示。



开启透明数据加密

最近更新时间：2024-12-11 09:41:22

操作场景

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

密钥管理说明

云数据库 MySQL 不提供加密所需的密钥和证书，透明数据加密功能所使用的密钥由 密钥管理系统 KMS 产生和管理，涉及密钥的相关说明如下。

- 透明数据加密功能不会额外收费，但密钥管理系统会产生额外费用，请参见 计费概述 。
- 密钥管理系统（后付费版）将于2024年12月30日停止服务，即密钥管理系统将不支持按量计费这种后付费模式，仅支持预付费的计费模式。详情请参见 关于密钥管理系统（后付费版）下线的公告 。
  - 对于存量使用密钥管理系统（后付费版）的用户，当账号处于欠费状态时将无法从 KMS 获取密钥，可能导致迁移、升级等任务无法正常进行。
  - 对于新购密钥管理系统（预付费版）的用户，当账号处于欠费状态时，由于密钥管理系统已经预付费了一定周期，在此周期内，不会影响获取 KMS 密钥，迁移、升级等任务不会受此影响。请注意 KMS 密钥续费时间，如果到期后 KMS 密钥没有续费，也会影响透明数据加密功能的使用，管理 KMS 密钥可前往 密钥管理系统控制台 。
- 云数据库 MySQL 实例和密钥管理系统支持的地域情况不同，在创建密钥时，若密钥管理系统上无国内对应的地域，您可选择在广州地域创建，若无海外对应的地域，您可选择在香港地域创建。
- 开启 TDE 加密功能后，若一个账号（UIN）下没有创建过加密表，则在密钥列表可能无法显示对应密钥信息，若一个账号（UIN）下创建过加密表，则可以看到对应密钥信息。创建加密表的操作请参考 常见问题 。

限制条件

- 实例架构须为通用型或独享型的双节点、三节点。
- 实例类型须为 MySQL 5.7、MySQL 8.0。
- 已开通 KMS 服务。如未开通，可在开通数据加密过程中根据引导开通 KMS。
- 已授予 KMS 密钥权限。如未授予，可在开通数据加密过程中根据引导进行授权。
- 操作账号需具有 QcloudAccessForMySQLRole 的权限，如无权限，可在开通数据加密过程中根据引导进行授权。

注意事项

- 撤销授权关系后，重启会造成 MySQL 数据库不可用。
- TDE 加密功能开通后无法关闭。
- 开启 TDE 加密功能后，用户如果要恢复数据到本地，需要先解密数据。
- 开启 TDE 加密功能后，可提高静态数据的安全性，但同时会影响访问加密数据库的读写性能，请结合实际情况选择开启 TDE 加密功能。
- 如果主实例关联只读实例或灾备实例，只需主动开通主实例的 TDE 功能即可，只读实例和灾备实例的数据加密会一并开通。
- 使用 TDE 功能时，请确保 KMS 密钥处于正常使用状态，否则将无法从 KMS 获取密钥，可能导致迁移、升级等任务无法正常进行。
- 开启 TDE 加密功能后，会增加 CPU 资源的消耗，大约会影响5%左右的性能。
- 开启 TDE 加密功能后，经过数据库身份验证的应用和用户可以透明地访问应用数据。
- 开启 TDE 加密功能后，备份压缩效果可能降低。

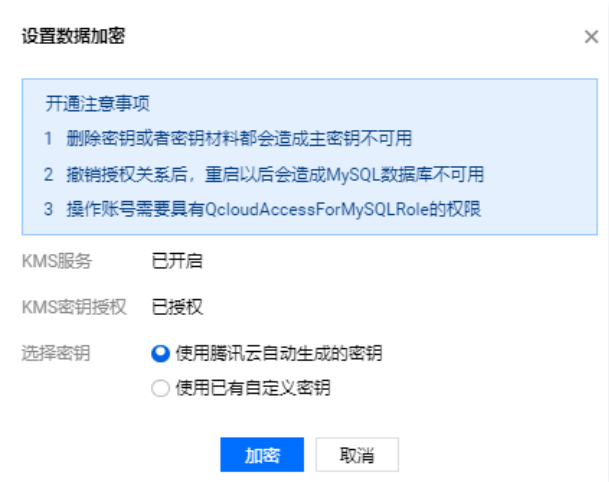
操作步骤

开启 TDE 加密功能

1. 登录 MySQL 控制台，在实例列表，单击实例 ID 或**操作**列的**管理**，进入管理页面。
2. 选择**数据安全**页，单击**加密状态**后的开关。

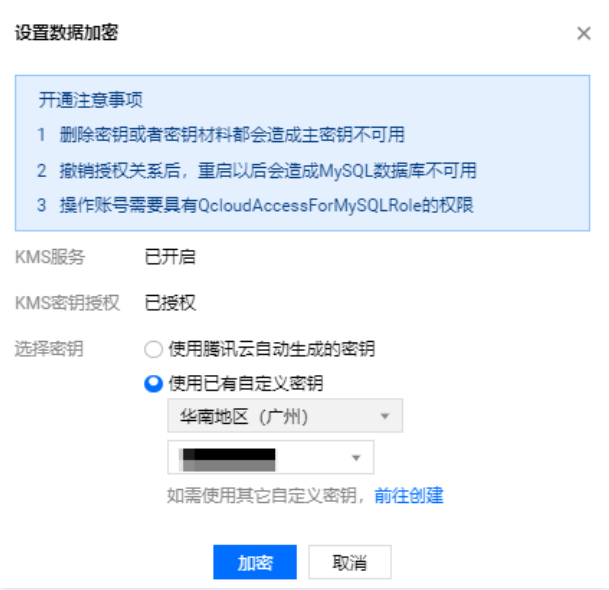
> **注意：**
> - 启用数据加密功能的实例，不支持使用物理备份恢复至其他主机上的自建数据库。
> - 数据加密开通后不可关闭。

3. 在弹出的对话框，开通 KMS 服务和授予 KMS 密钥权限，选择密钥后，单击**加密**。
   - 选择**使用腾讯云自动生成的密钥**时，由腾讯云自动生成密钥。



   - 选择**使用已有自定义密钥**时，可选择自己创建的密钥。

> **说明：**
> 如无自定义密钥，需单击**前往创建**，在密钥管理系统控制台创建密钥，详情请参见 创建密钥 。



加密数据表

开启 TDE 加密功能后，用户需对 MySQL 的表执行如下 DDL 操作才能进行数据加密。

- 如果您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

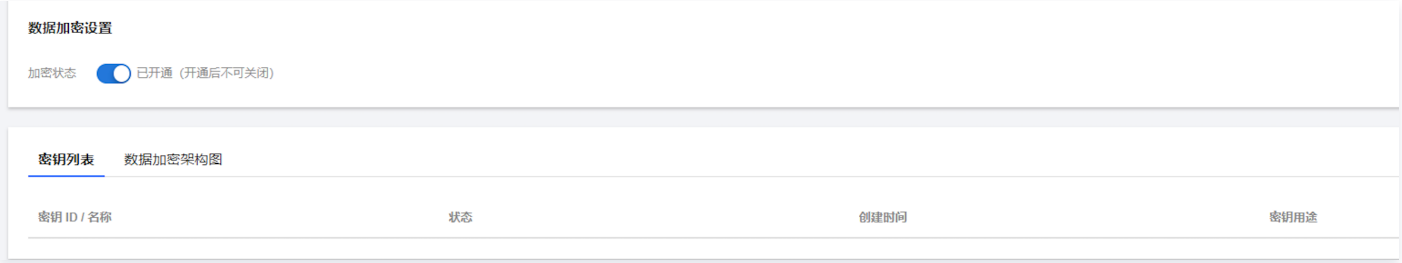
开启 TDE 加密功能后，用户还需要对 MySQL 的表进行如下 DDL 操作才能进行数据解密。
如果您要对加密的表进行解密，请执行如下命令：

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

常见问题

为什么开启 TDE 加密后，密钥列表无密钥信息？

问题现象



开启 TDE 加密后的正常密钥列表

处理建议

1. 首先查询 KMS 加密状态是否正常，账号是否欠费，云数据库 MySQL 实例和密钥管理系统是否存在到期未付费情况。请确保以上几项都正常后重试。
2. 如果您是首次使用 TDE 加密，在您的账号（UIN）下未创建过加密表，则密钥列表会没有密钥信息，请参考以下命令，在您的数据实例中创建加密表后重试。

```
  CREATE TABLE `user_test` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT,
  `userId` int(11) NOT NULL,
  `age` int(11) NOT NULL,
  `name` varchar(64) DEFAULT NULL,
  `ins_date` varchar(10) DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `idx_ins_date` (`ins_date`),
  KEY `idx_userId` (`userId`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 ENCRYPTION='Y';
```

为什么开启 TDE 加密后，密钥无法使用？

请确认您的 KMS 加密状态是否正常，账号是否欠费，云数据库 MySQL 实例和密钥管理系统是否存在到期未付费情况。由于密钥管理系统（后付费版）已无法创建，新购密钥管理系统仅支持购买预付费版，因此，存量使用密钥管理系统（后付费版）的用户，当 KMS 加密状态异常或账号欠费时，密钥将无法使用，请为账号充值后重试。

设置 SSL 加密

最近更新时间：2025-05-21 14:17:22

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。

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。

云数据库 MySQL 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 MySQL5.6/5.7/8.0。
- 实例架构为双节点/三节点。
- 实例引擎为 InnoDB/RocksDB。

支持版本

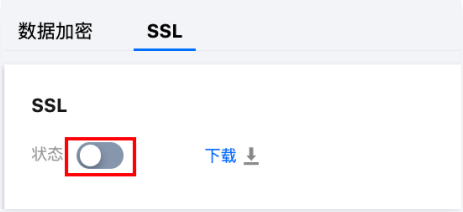
MySQL 使用 OpenSSL 来实现安全连接。云数据库 MySQL 支持传输层安全性协议（TLS）版本1.0、1.1、1.2和1.3。TLS 支持取决于 MySQL 版本。下表显示了支持 TLS 的 MySQL 版本。

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 MySQL 控制台，在实例列表，单击**实例 ID** 或**操作**列的**管理**，进入实例管理页面。
2. 在实例管理页面的**数据安全**页下，选择 **SSL** 页。

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。
- 主实例开启 SSL 窗口如下：



> **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。

- RO 实例开启 SSL 界面如下：

> **注意：**
> 配置 RO 实例 SSL 功能会同步配置所属 RO 组中其他 RO 实例。

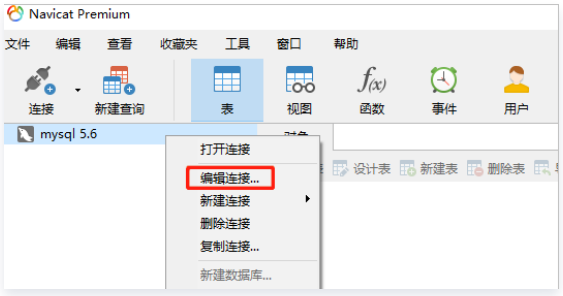
4. 单击**下载**，下载 SSL CA 证书，证书有效期为20年。
   下载的文件为压缩包（TencentDB-CA-Chain.zip），包含如下三个文件：
   - p7b 文件：用于 Windows 系统中导入 CA 证书。
   - jks 文件：Java 中的 truststore 和 keystore 存储文件，密码统一为 tencentdb，用于 Java 程序中导入 CA 证书链。
   - pem 文件：用于其他系统或应用中导入 CA 证书。

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：**
> 云数据库 MySQL 每开启或关闭一次 SSL 加密，其证书就会新生成。

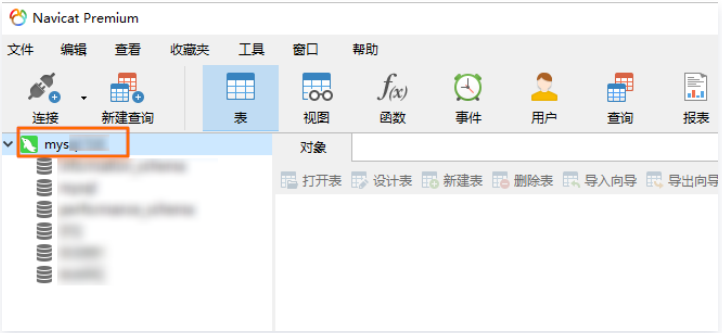
1. 打开 Navicat。
2. 在对应数据库上单击鼠标右键，选择**编辑连接**。



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 MySQL 控制台，在实例列表，单击**实例 ID** 或**操作**列的**管理**，进入实例管理页面。
2. 在实例管理页面的**数据安全**页下，选择 **SSL** 页。
3. 单击已开通前面的开关按钮，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. 通过云数据库 MySQL 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。
   - MariaDB 客户端，通过如下命令连接实例。

     ```
     mysql -h <IP 地址> --ssl-ca=<ca证书> --ssl -P <端口号> -u <用户名> -p
     ```

   - 客户端数据库版本为 MySQL 5.6时，通过如下命令连接实例。

     ```
     mysql -P <端口号> -h <IP 地址>  -u <用户名> -p<密码> --ssl-ca<ca证书>
     ```

   - 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> --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> --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>'
  )
```

监控与告警
监控功能

最近更新时间：2025-05-21 14:17:22

为方便用户查看和掌握实例的运行信息，云数据库 MySQL 提供了丰富的性能监控项与便捷的监控功能（自定义视图、时间对比、合并监控项等）。用户可登录云数据库 MySQL 控制台，进入实例管理页的**实例监控**查看。

> **说明：**
> - 您可以通过腾讯云可观测平台 API 中的 拉取指标监控数据、云数据库 MySQL 监控指标 来获取实例的监控指标。
> - 您也可以为监控指标 创建 Dashboard，动态分析指标监控数据。
> - 单个实例的表数量超过30万后，可能会影响数据库监控，请合理规范表的数量，控制单个实例表数量不超过30万。

支持监控的实例类型

云数据库 MySQL 支持主实例（包括读写节点）、只读实例（包括只读节点与只读分析引擎）、灾备实例，以及数据库代理节点的监控，并为每个实例提供独立的监控视图供查询。

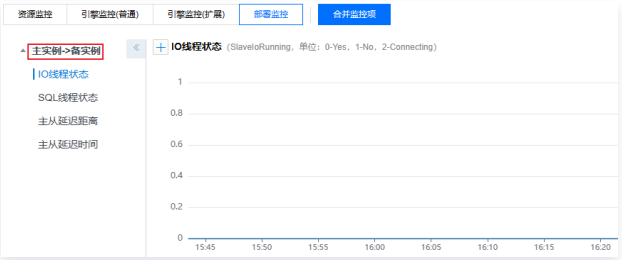
监控分类

云数据库 MySQL 包括资源监控、引擎监控（普通）、引擎监控（扩展）、部署监控4种监控类型，通过查看不同监控类型的指标，可快速、准确的了解实例性能及运行状况。

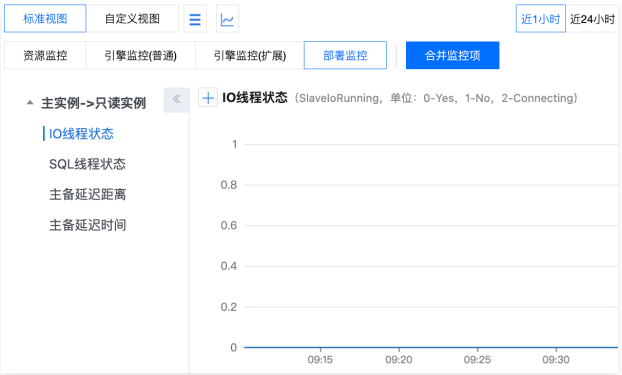
> **说明：**
> 云数据库 MySQL 单节点云盘版实例支持的监控类型包括资源监控和引擎监控（普通），暂不支持引擎监控（扩展）和部署监控。

资源监控：提供 CPU、内存、磁盘及网络相关的监控数据。
引擎监控（普通）：提供连接数、锁信息、热点表、慢查询等相关的监控数据，方便您诊断故障及性能优化。
引擎监控（扩展）：提供更为丰富引擎相关的监控指标，协助您最大限度发现数据库存在或潜在的健康问题。
部署监控：提供主从延迟相关的监控指标。部署监控分为主机和备机：

- 若实例为主实例，则实例部署监控的对象是主实例与其隐藏备机之间的链路，部署监控展示为隐藏备机的 IO、SQL 线程状态，主从延迟距离和主从延迟时间指的是主实例与其隐藏备机之间。



- 若实例为只读实例，则实例部署监控的对象是主实例与只读实例之间的链路，部署监控展示为只读实例的 IO、SQL 线程状态，主从延迟距离和主从延迟时间指的是只读实例与主实例之间。



- 若实例为灾备实例：
  - 灾备实例部署监控的对象是灾备实例与主实例之间的链路，部署监控展示为灾备实例的 IO、SQL 线程状态，主从延迟距离和主从延迟时间指的是灾备实例与主实例之间。
  - 灾备实例部署监控的对象是灾备实例与其隐藏备机之间的链路，部署监控展示为其隐藏备机的 IO、SQL 线程状态，主从延迟距离和主从延迟时间指的是灾备实例与其隐藏备机之间。

监控粒度

自2018年08月11日起，云数据库 MySQL 监控粒度实行自适应策略，暂不支持监控粒度的自定义选择。监控粒度自适应策略如下：

时间跨度	监控粒度	自适应说明	保留时长
(0h, 4h]	5s	时间跨度在4小时内，监控粒度为5秒	1天
(4h, 2d]	1min	时间跨度超过4小时，但在2天内，监控粒度调整为1分钟	15天
(2d, 10d]	5min	时间跨度超过2天，但在10天内，监控粒度调整为5分钟	31天
(10d, 30d]	1h	时间跨度超过10天，但在30天内，监控粒度调整为1小时	62天
(30d，180d]	1天	时间跨度超过30天，但在180天内，监控粒度调整为1天	180天

> **说明：**
> 目前云数据库 MySQL 最长支持查看180天内的监控数据。

监控指标

腾讯云可观测平台从实例维度为云数据库 MySQL 实例提供丰富的监控指标，具体请参见下表。
除了通过监控指标来掌握实例的运行状况，云数据库 MySQL 也支持监控事件和事件告警，通过设置事件规则、事件目标及推送用户，当检测到异常事件，事件告警将通过多种可选的方式向用户发送告警通知，帮助用户更全面的掌控实例的运行信息。

- 详细了解事件告警，请参见 事件总线 。
- 详细了解支持的监控事件，请参见 云数据库 MySQL 事件列表 。
- 只读分析引擎的监控指标，请参见 监控告警 。

> **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	指标说明
CPU 利用率	CpuUseRate	%	允许闲时超用，CPU 利用率可能大于100%。
内存利用率	MemoryUseRate	%	允许闲时超用，内存利用率可能大于100%。
内存占用	MemoryUse	MB	允许闲时超用，实际内存占用可能大于购买规格。
磁盘利用率	VolumeRate	%	• 未加入白名单：磁盘利用率 = 数据使用空间 / 实例购买空间。 • 添加白名单：磁盘利用率 =（数据使用空间 + 日志使用空间）/ 实例购买空间。

			说明： 如果您需要展示磁盘利用率 =（数据使用空间 + 日志使用空间）/ 实例购买空间，请 联系售后 开启。
磁盘使用空间	RealCapacity	MB	磁盘的使用空间。
磁盘占用空间	Capacity	MB	磁盘的占用空间。
IOPS	Iops	次/秒	每秒的输入输出量。
IOPS 利用率	IopsUseRate	%	IOPS 利用率。
内网出流量	BytesSent	Byte/秒	每秒发送的字节数。
内网入流量	BytesReceived	Byte/秒	每秒接受的字节数。
每秒执行操作数	Qps	次/秒	数据库每秒执行的 SQL 数（含 insert、select、update、delete、replace），QPS 指标主要体现云数据库实例的实际处理能力。
每秒执行事务数	Tps	次/秒	数据库每秒传输的事务处理个数。
连接数利用率	ConnectionUseRate	%	当前打开连接数/最大连接数。
最大连接数	MaxConnections	个	最大连接数。
当前打开连接数	ThreadsConnected	个	当前打开的连接的数量。
慢查询数	SlowQueries	次	查询时间超过 long_query_time 秒的查询的个数。
全表扫描数	SelectScan	次/秒	执行全表搜索查询的数量。
查询数	SelectCount	次/秒	每秒查询数。
更新数	ComUpdate	次/秒	每秒更新数。
删除数	ComDelete	次/秒	每秒删除数。
插入数	ComInsert	次/秒	每秒插入数。
覆盖数	ComReplace	次/秒	每秒覆盖数。
总请求数	Queries	次/秒	所有执行的 SQL 语句，包括 set，show 等。
查询使用率	QueryRate	%	每秒执行操作数 QPS/推荐每秒操作数。
临时表数量	CreatedTmpTables	个/秒	内存临时表数量。
等待表锁次数	TableLocksWaited	次/秒	需要等待的表锁数。
缓存命中率	InnodbCacheHitRate	%	InnoDB 缓存命中率。
缓存使用率	InnodbCacheUseRate	%	InnoDB 缓存使用率。
读磁盘数量	InnodbOsFileReads	次/秒	InnoDB 读磁盘数量。
写磁盘数量	InnodbOsFileWrites	次/秒	InnoDB 写磁盘数量。
fsync 数量	InnodbOsFsyncs	次/秒	InnodbFsync 数量。
当前 InnoDB 打开表的数量	InnodbNumOpenFiles	个	当前 InnoDB 打开表的数量。
缓存命中率	KeyCacheHitRate	%	MyISAM 缓存命中率。
缓存使用率	KeyCacheUseRate	%	MyISAM 缓存使用率。

提交数	ComCommit	次/秒	每秒提交次数。
回滚数	ComRollback	次/秒	每秒回滚次数。
已创建的线程数	ThreadsCreated	次/秒	已创建的线程数。
运行的线程数	ThreadsRunning	个	运行的线程数。
磁盘临时表数量	CreatedTmpDiskTables	次/秒	磁盘临时表数量。
临时文件数量	CreatedTmpFiles	次/秒	临时文件数量。
读下一行请求数	HandlerReadRndNext	次/秒	读下一行请求数。
内部回滚数	HandlerRollback	次/秒	每秒事务被回滚的次数。
内部提交数	HandlerCommit	次/秒	内部提交数。
InnoDB 空页数	InnodbBufferPoolPagesFree	个	InnoDB 空页数。
InnoDB 总页数	InnodbBufferPoolPagesTotal	个	InnoDB 总页数。
InnoDB 逻辑读	InnodbBufferPoolReadRequests	次/秒	InnoDB 缓冲池预读页次数。
InnoDB 物理读	InnodbBufferPoolReads	次/秒	InnoDB 磁盘读页次数。
InnoDB 读取量	InnodbDataRead	Byte/秒	InnoDB 读取量。
InnoDB 总读取量	InnodbDataReads	次/秒	InnoDB 总读取量。
InnoDB 总写入量	InnodbDataWrites	次/秒	InnoDB 总写入量。
InnoDB 写入量	InnodbDataWritten	Byte/秒	InnoDB 引擎每秒已经完成写数据的字节数。
InnoDB 行删除量	InnodbRowsDeleted	次/秒	InnoDB 引擎每秒删除的行数。
InnoDB 行插入量	InnodbRowsInserted	次/秒	InnoDB 引擎每秒插入的行数。
InnoDB 行更新量	InnodbRowsUpdated	次/秒	InnoDB 引擎每秒更新的行数。
InnoDB 行读取量	InnodbRowsRead	次/秒	InnoDB 引擎每秒读取的行数。
InnoDB 平均获取行锁时间	InnodbRowLockTimeAvg	毫秒	InnoDB 引擎行锁定的平均时长。
InnoDB 等待行锁次数	InnodbRowLockWaits	次/秒	InnoDB 引擎每秒等待行锁定的次数。
键缓存内未使用的块数量	KeyBlocksUnused	个	MyISAM 引擎未使用键缓存块的个数。
键缓存内使用的块数量	KeyBlocksUsed	个	MyISAM 引擎已使用键缓存块的个数。
键缓存读取数据块次数	KeyReadRequests	次/秒	MyISAM 引擎每秒读取键缓存块的次数。
硬盘读取数据块次数	KeyReads	次/秒	MyISAM 引擎每秒读取硬盘数据块的次数。
数据块写入键缓冲次数	KeyWriteRequests	次/秒	MyISAM 引擎每秒写键缓存块的次数。

版权所有：腾讯云计算（北京）有限责任公司

数据块写入磁盘次数	KeyWrites	次/秒	MyISAM 引擎每秒写硬盘数据块的次数。
已经打开的表数	OpenedTables	个	实例维度已经打开的表数。
立即释放的表锁数	TableLocksImmediate	次/秒	立即释放的表锁数。
打开文件总数	OpenFiles	个	打开文件总数。
日志使用量	LogCapacity	MB	日志使用空间。
IO 线程状态	SlaveIoRunning	状态值： 0 – Yes 1 – No 2 – Connecting	主实例 IO 线程状态。 **说明：** 状态值中，0表示“运行”，1表示“断开”，2表示“建联中”。
SQL 线程状态	SlaveSqlRunning	状态值： 0 – Yes 1 – No	主实例 SQL 线程状态。 **说明：** 状态值中，0表示“运行”，1表示“断开”。
主备延迟距离	MasterSlaveSyncDistance	MB	主备不同步的距离。
主备延迟时间	SecondsBehindMaster	秒	主备差距时间。

热点问题

如何配置主从延迟监控？

配置主从延迟监控分为以下两种场景，您可根据实际场景对应操作。

场景一：为主实例配置主从延迟监控

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**，然后选择**告警配置** > **新建策略**。
2. 在告警策略页，策略类型项选择：**云数据库** > **MySQL** > **备机监控**。

> **说明：**
> 为主实例配置主从延迟监控，策略类型要选择备机监控，监控的是备机到主机的延迟信息。

3. 在配置告警规则下完成对监控项“主从延迟距离”和“主从延迟时间”的触发条件设置，然后根据您的需要完成其他配置项的设置，单击**完成**。
4. 设置完成后，当“主从延迟距离”和“主从延迟时间”监控项满足触发条件时即可触发告警。

场景二：为 RO 和灾备实例配置主从延迟监控

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**，然后选择**告警配置** > **新建策略**。
2. 在告警策略页，策略类型项选择：**云数据库** > **MySQL** > **主机监控**。

> **说明：**
> - 为 RO 实例配置主从延迟监控，策略类型只能选主机监控，监控的是 RO 实例到其主实例的延迟信息。
> - 为灾备实例配置主从延迟监控，策略类型若选择主机监控，则监控的是灾备实例到其主实例的延迟信息，若选择备机监控，则监控的是灾备实例的备机到灾备实例的延迟信息。

3. 在配置告警规则下完成对监控项“主从延迟距离”和“主从延迟时间”的触发条件设置，然后根据您的需要完成其他配置项的设置，单击**完成**。
4. 设置完成后，当“主从延迟距离”和“主从延迟时间”监控项满足触发条件时即可触发告警。

告警策略（腾讯云可观测平台）

最近更新时间：2025-05-21 14:17:22

本文为您介绍通过腾讯云可观测平台控制台，创建告警策略和关联告警对象的操作。

操作场景

您可以创建告警用于在云产品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应用程序的健壮性和可靠性。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 。
当用户需要针对某个产品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。

> **注意：**
> 请确认您已经设置默认告警接收人，否则腾讯云数据库的默认告警策略将无法通知到您。

操作步骤

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. 在告警策略列表页中，单击**新建**。
3. 在新建策略页中，设置策略名称、策略类型、告警对象、触发条件等内容。

- **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。
  - 部署类监控为主机：当监控实例为主实例时，由于其主实例不为任何实例的从实例，因此其主机下复制相关的监控数据无效，此时其 IO、SQL 线程状态为未启动。仅当监控实例为灾备实例和只读实例时，其对应复制相关的监控数据才有效，且 IO、SQL 线程才是启动的状态。
  - 部署类监控为备机：双节点、三节点的主实例与灾备实例默认为主备架构，因此，仅当监控实例为主实例及灾备实例时，其备机下复制相关的监控数据才有效。用于反映主实例、灾备实例与其隐藏备节点的延迟距离及时间，建议关注备机下相关的监控数据，若主实例及灾备实例存在故障时，其监控实例对应隐藏的备节点可快速提升为主实例。
- **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**：指标、比较关系、阈值、统计周期和持续周期组成的一个有语义的条件。例如：指标为磁盘使用率、比较关系为>、阈值为80%、统计周期为5分钟、持续周期为2个周期。表示：每5分钟收集一次磁盘使用率数据，若某台云数据库的磁盘使用率连续两次大于80%则触发告警。

- **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 。

4. 确认无误后，单击**完成**。

关联告警对象

创建完告警策略后，您可以为其关联一些告警对象，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。

热点问题

如何配置主从延迟监控?

配置主从延迟监控分为以下两种场景，您可根据实际场景对应操作。

场景一：为主实例配置主从延迟监控

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，然后选择**告警配置** > **新建策略**。
2. 在告警策略页，策略类型项选择：云数据库 > MySQL > **备机监控**。

> **说明：**
> 为主实例配置主从延迟监控，策略类型要选择备机监控，监控的是备机到主机的延迟信息。

3. 在配置告警规则下完成对监控项“主从延迟距离”和“主从延迟时间”的触发条件设置，然后根据您的需要完成其他配置项的设置，单击**完成**。
4. 设置完成后，当“主从延迟距离”和“主从延迟时间”监控项满足触发条件时即可触发告警。

场景二：为 RO 和灾备实例配置主从延迟监控

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**，然后选择**告警配置** > **新建策略**。
2. 在告警策略页，策略类型项选择：**云数据库** > MySQL > **主机监控**。

> **说明：**
> - 为 RO 实例配置主从延迟监控，策略类型只能选主机监控，监控的是 RO 实例到其主实例的延迟信息。
> - 为灾备实例配置主从延迟监控，策略类型若选择主机监控，则监控的是灾备实例到其主实例的延迟信息，若选择备机监控，则监控的是灾备实例的备机到灾备实例的延迟信息。

3. 在配置告警规则下完成对监控项“主从延迟距离”和“主从延迟时间”的触发条件设置，然后根据您的需要完成其他配置项的设置，单击**完成**。
4. 设置完成后，当“主从延迟距离”和“主从延迟时间”监控项满足触发条件时即可触发告警。

告警通知（DBbrain）

最近更新时间：2024-10-18 14:14:23

本文为您介绍在 MySQL 控制台查看异常告警（DBbrain）消息的操作。
异常告警（DBbrain）通知服务为用户实时推送 MySQL 实例异常告警消息，方便用户及时便捷地了解数据库异常诊断问题。
异常告警消息历史列表中存放了所有被推送过的异常告警消息，便于用户快速查阅历史推送过的异常诊断问题。

查看告警

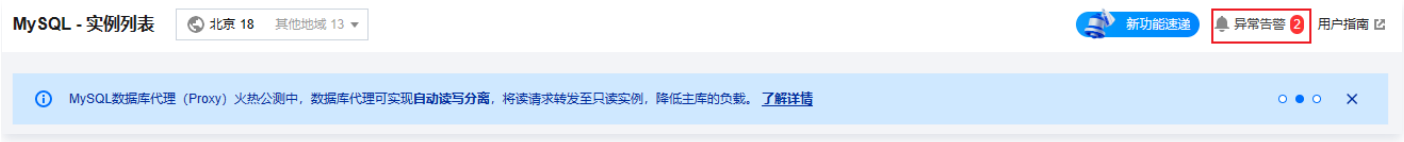
方式一

登录 MySQL 控制台，如果实例在当前时间出现了异常诊断问题，在控制台的右上角会为您实时弹窗推送异常告警消息通知，消息通知包含数据库实例的实例ID/名称、诊断项、开始时间等，方便您快速了解数据库实例的诊断问题。

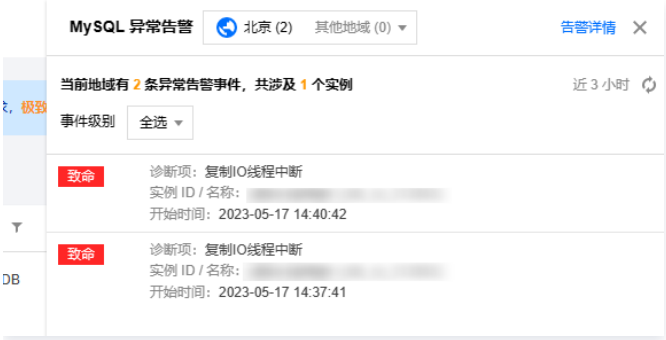
- 单击消息通知中的**查看异常诊断详情**，可以查看该实例具体的诊断详情及优化建议。
- 勾选消息通知中的**今日不再提醒**，则当天您账号下的数据库实例发生异常诊断问题时，将不会弹窗为您推送异常告警消息。

方式二

登录 MySQL控制台，在左侧导航选择**实例列表**、**任务列表**、**参数模板**、**回收站**或**置放群组**页，在右上角单击**异常告警**，可展开异常告警历史消息列表，按钮旁边展示了用户账号下实例所产生的告警问题数量。



在展开的异常告警历史消息列表，可以查看到所有被推送过的异常告警历史消息，支持切换地域查看，单击每条告警消息，可查看各异常告警事件的异常诊断详情。



设置事件告警

最近更新时间：2025-05-21 14:17:22

云数据库 MySQL 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：

- 对于云数据库 MySQL 产生的告警事件，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，云数据库 MySQL 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. 点击云服务事件集下的 **default** 事件集。

4. 点击**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点击**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，云服务类型选择云数据库 MySQL。
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	选择云数据库 MySQL。
事件类型	根据需要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 云数据库 MySQL 事件列表 。
匹配对象	选择事件告警的匹配对象。

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受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 云数据库 MySQL 事件列表 。

   - **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。

操作日志

操作日志

最近更新时间：2025-06-03 17:59:22

操作场景

超过指定时间的 SQL 语句查询称为“慢查询”，对应语句称为“慢查询语句”，数据库管理员（DBA）对慢查询语句进行分析并找到慢查询出现原因的过程称为“慢查询分析”。
您可以在控制台的操作日志页面查看实例的慢日志明细、错误日志明细、回档日志、日志投递、扩容历史，及慢日志下载。也可以在命令行界面（CLI）或者使用云数据库 API 查看、下载数据库日志，请参见 查询慢查询日志 与 查询二进制日志 。

> **说明：**
> 云数据库 MySQL 单节点（云盘）实例、云盘版实例暂不支持慢日志下载、回档、日志投递功能。云盘版实例也不支持查看扩容历史。

MySQL 慢查询相关说明

- long_query_time：慢查询阈值参数，精度可达微秒级别，默认为1s，SQL 语句执行时间超过此数值时，会被记录到慢日志。
  若调整 long_query_time 参数，不会影响原有的慢日志，例如慢日志阈值参数为1s，则上报超过1s的慢日志记录，后续调整为2s，原有上报的日志仍然会显示。
- log_queries_not_using_indexes：是否记录未使用索引的查询，默认 OFF。

操作步骤

> **说明：**
> - 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。
> - 对于云盘版实例，在操作日志中的慢日志明细和错误日志明细，都支持选择实例下具体的节点进行查询。

1. 登录 MySQL 控制台，在实例列表，单击**实例 ID** 或**操作**列的**管理**，进入实例管理页面。
2. 在实例管理页面，选择**操作日志**页，可选择查看实例的慢日志明细、错误日志明细、回档日志、日志投递、扩容历史，及慢日志下载。

功能项	说明
慢日志明细	记录1个月内数据库中执行时间超过1s的 SQL 语句
慢日志下载	提供慢日志的下载
错误日志明细	记录了每次启动和关闭的详细信息，以及运行过程中所有较为严重的警告和错误信息
回档日志	记录回档任务的运行状态和进展
日志投递	采集来源为云数据库 MySQL 实例的慢日志和错误日志数据，并投递至日志服务（Cloud Log Service，CLS）进行分析
扩容历史	记录实例的 CPU 弹性扩容历史。

3. 在慢日志下载页面，单击**操作**列的**下载**，下载慢日志。
4. 在弹出的对话框，推荐您复制下载地址，并 登录到云数据库所在 VPC 下的 CVM（Linux 系统）中，运用 wget 命令进行内网高速下载，更高效。

说明：

- 若日志大小为0KB，则无法下载。
- 您也可以选择**本地下载**直接下载，但耗时较多。
- wget 命令格式：wget -c '日志文件下载地址' -O 自定义文件名.log。

示例如下：

```
wget -c 'http://szx.dl.cdb.tencentyun.com:303/cfdee?appid=1210&time=1591&sign=aIGM%3D' -O test.log
```

日志投递

最近更新时间：2025-04-30 10:00:44

云数据库 MySQL 提供日志投递功能，通过日志投递，可采集来源为云数据库 MySQL 实例的慢日志和错误日志数据，并投递至日志服务（Cloud Log Service，CLS）进行分析，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投递功能。

前提条件

使用该功能前，请确保您已开通 日志服务 CLS 。

使用限制

仅云盘版实例、双节点/三节点本地盘实例、RO 实例支持使用日志投递功能。

慢日志定义

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 cdb-xxx。
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。

错误日志定义

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 cdb-xxx。
content	String	错误 志内容。
level	String	错误 志的级别。 如 warning、note、error 等。 级别 志则为 none，如死锁的错误 志等。

开启慢日志投递

1. 登录 MySQL 控制台，在实例列表，单击**实例 ID** 进入管理页面。
2. 在实例管理页，选择**操作日志** > **日志投递**。
3. 在**慢日志投递**下，单击编辑图标。

4. 在弹窗下选择**开启**，并完成如下配置，单击**确定**。

参数	说明
地域	选择日志投递的地域，支持异地投递。
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。支持选择已有的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，作为慢日志投递的归类。 • 创建日志集：可新建日志集作为慢日志投递的归类，创建日志集的命名格式为 **cloud_自定义_logset**，其中自定义部分仅支持英文字母、数字、下划线，长度不可超过20。
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_自定义_topic**，其中自定义部分仅支持英文字母、数字、下划线，长度不可超过20。 **说明：** 您可对日志主题进行管理，详细请参见 管理日志主题 。
保存时间	选择慢日志投递数据的保存时间，默认为30天，支持选择1天-3600天，日志过期后将自动清除数据。若日志主题选择了已有的日志主题，则保留时间会默认和对应已有的日志主题保存时间一致。
创建索引	默认开启。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。 了解详情 若日志主题选择了已有的日志主题，则索引状态会默认和对应已有的日志主题的索引状态一致。

5. 成功开启慢日志投递后，您可在日志投递下看到慢日志投递状态为开启，单击日志主题名称，可跳转至 日志服务控制台 进行后续分析管理。

关闭慢日志投递

> **说明：**
> 关闭慢日志投递后，已产生的慢日志投递数据，会跟随开启慢日志投递时所选择的保存时间而保存，过期后才会自动清除数据。

1. 登录 MySQL 控制台，在实例列表，单击**实例 ID** 进入管理页面。
2. 在实例管理页，选择**操作日志** > **日志投递**。
3. 在**慢日志投递**下，单击编辑图标。
4. 在弹窗下选择**关闭**，单击**确定**。

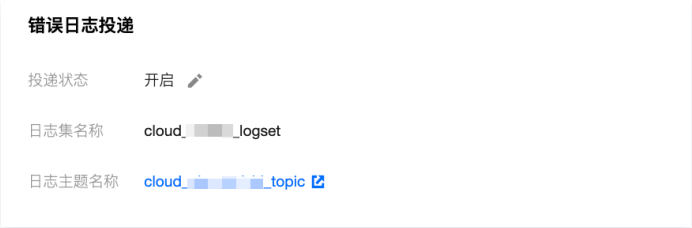
开启错误日志投递

1. 登录 MySQL 控制台，在实例列表，单击**实例 ID** 进入管理页面。
2. 在实例管理页，选择**操作日志** > **日志投递**。
3. 在**错误日志投递**下，单击编辑图标。
4. 在弹窗下选择**开启**，并完成如下配置，单击**确定**。

参数	说明
地域	选择日志投递的地域，支持异地投递。
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，作为错误日志投递的归类。 • 创建日志集：可新建日志集作为错误日志投递的归类，创建日志集的命名格式为 **cloud_自定义_logset**，其中自定义部分仅支持英文字母、数字、下划线，长度不可超过20。
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_自定义_topic**，其中自定义部分仅支持英文字母、数字、下划线，长度不可超过20。

	说明： 您可对日志主题进行管理，详细请参见 管理日志主题 。
保存时间	选择错误日志投递数据的保存时间，默认为30天，支持选择1天-3600天，日志过期后将自动清除数据。若日志主题选择了已有的日志主题，则保留时间会默认和对应已有的日志主题保存时间一致。
创建索引	默认开启。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。了解详情 若日志主题选择了已有的日志主题，则索引状态会默认和对应已有的日志主题的索引状态一致。

5. 成功开启错误日志投递后，您可在日志投递下看到错误日志投递状态为开启，单击日志主题名称，可跳转至 日志服务控制台 进行后续分析管理。



关闭错误日志投递

> **说明：**
> 关闭错误日志投递后，已产生的错误日志投递数据，会跟随开启错误日志投递时所选择的保存时间而保存，过期后才会自动清除数据。

1. 登录 MySQL 控制台，在实例列表，单击**实例 ID** 进入管理页面。
2. 在实例管理页，选择**操作日志** > **日志投递**。
3. 在**错误日志投递**下，单击编辑图标。
4. 在弹窗下选择**关闭**，单击**确定**。

相关文档

- 日志集
- 管理日志主题
- 仪表盘
- 数据加工
- 检索分析

只读分析引擎
简介

最近更新时间：2025-04-23 15:39:22

本文为您介绍云数据库 MySQL 基于 LibraDB 引擎实现的全新功能：只读分析引擎。

> **说明：**
> 只读分析引擎目前免费试用中，正式商业化收费时间为2025年05月19日。产品定价请参见 计费概述 。

背景

云数据库 MySQL 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，云数据库 MySQL 全新支持了只读分析引擎功能，为用户提供高效且实时的数据分析服务。

什么是只读分析引擎

只读分析引擎是云数据库 MySQL 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

只读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只读分析引擎，并且在无需分析加速的时候，可随时关闭只读分析引擎，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，可快速云数据库 MySQL 中的存量数据加载至只读分析引擎中。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。但是对于只读分析引擎而言，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只读分析引擎加速的库表，或者有数据分析价值的库表加载到只读分析引擎中进行数据分析，灵活控制只读分析引擎所占用的磁盘大小。
- **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

只读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。通过只读分析引擎的功能，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，使用只读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，使用只读分析引擎，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只读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。通过借助只读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。

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 16:41:31

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。

可用性差异

- 只读分析引擎暂不支持配置变更。
- 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。
- 如云数据库 MySQL 开启了只读分析引擎，则在“主实例”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_SUBTRACTION	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）。
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 11:52:02

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。若在云数据库 MySQL 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，在云数据库 MySQL 的主实例中执行 `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主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，若云数据库 MySQL 的主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：**
  > 若在云数据库 MySQL 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：**
> 当在云数据库 MySQL 的主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。

函数与运算符说明
位运算支持说明

最近更新时间：2025-01-21 15:47:02

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：2025-05-21 14:17:22

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。

版权所有：腾讯云计算（北京）有限责任公司

比较运算支持说明

最近更新时间：2025-01-21 15:47:02

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：2025-01-21 15:47:02

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 17:36:42

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：2025-01-21 15:47:02

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 17:36:42

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

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。

聚合函数支持说明

最近更新时间：2025-05-23 09:25:22

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 17:23:32

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 17:23:32

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 17:23:32

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 17:57:22

本文为您介绍如何为云数据库 MySQL 创建只读分析引擎实例。

前提条件

- 准备好云数据库 MySQL 实例，请参见 创建 MySQL 实例 。
- 实例状态为运行中。

操作步骤

> **说明：**
> - 目前仅支持在已经创建好的实例下，进行只读分析引擎的创建，不支持在新建实例的时候进行只读分析引擎的创建。
> - 仅支持在 MySQL 5.7、MySQL 8.0 的内核版本中创建只读分析引擎。
> - RocksDB 内核的实例不支持创建只读分析引擎。
> - 仅支持在双节点、三节点架构的实例下创建只读分析引擎。

1. 登录 MySQL 控制台 。
2. 在实例列表找到需要添加只读分析引擎的实例，单击**实例 ID** 或**操作**列的**管理**，进入实例管理页。
3. 选择**只读实例**页，单击**新建**进入只读实例创建页面。

> **说明：**
> 您也可以在实例详情页，单击实例架构图下的**添加只读分析引擎**进入**只读实例**页，然后再单击**新建**。
>
>
>

4. 完成如下配置，单击**立即购买**。

参数	说明
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。
指定分析 RO 组	支持系统自动分配、新建 RO 组、已有 RO 组。 系统自动分配：若一次购买多个实例，将为每个实例分配一个独立的 RO 组，权重分配方式默认为系统自动分配。 新建 RO 组：新建一个 RO 组，若一次购买多个实例，将都分配至这个 RO 组，权重分配方式默认为系统自动分配。 已有 RO 组：指定一个已有 RO 组，若一次购买多个实例，将都分配至这个 RO 组。权重分配方式与 RO 组设置相同：如果 RO 组设置为系统自动分配，则根据购买规格自动添加 RO 组；如果为自定义分配，则默认权重为零。由于同一 RO 组内网地址相同，因此若是 VPC 网络将共享同一个安全组设置。若选择已有 RO 组，则在购买时无法再自定义安全组。 **注意：** 分析 RO 组不支持延迟超限剔除等能力。其他的作用与普通 RO 组一致。
设置分析 RO 组名称	选择新建 RO 组时需设置新 RO 组的名称，支持长度小于60的中文/英文/数字/"-"/"_"/"."
计费模式	在主实例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；若主实例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。

地域	选择只读分析引擎实例所属的地域。目前仅支持云数据库 MySQL 主实例处于同一地域。
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 。
可用区	选择只读分析引擎实例所属的可用区，具体可选的可用区请以实际页面显示为准。
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	选择硬盘空间，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。
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。
数据加载方式	指只读分析引擎加载对应云数据库 MySQL 主实例中数据的方式。默认为无对象。 • 全对象：指将主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。
网络	默认 VPC 与云数据库 MySQL 主实例保持一致，子网为指定可用区的可选子网。同时也支持与主实例不在同一 VPC 网络内。
安全组	可为只读分析引擎实例选择和对应云数据库 MySQL 实例不同的安全组，以控制实例访问策略。
标签	为只读分析引擎实例添加标签。
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。

5. 完成购买后，即可返回实例列表，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：2025-05-21 14:17:22

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 连接实例	通过 DMC 连接实例，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 。
内网连接	使用云服务器 CVM 通过云数据库 MySQL 内网地址连接数据库。
外网连接	使用云服务器 CVM 或第三方应用通过云数据库 MySQL 外网地址连接数据库。

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.xx.xx.12 -P3306 -utest -p -c -A

mysql> /* to server cdbaro-rmoskdfp */ 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以及之后的数据库代理版本中支持使用 `/*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 cdbaro-rmoskdfp */  * 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 语法的同时添加 SQL 执行计划的语法。

```
mysql> select /* to server cdbaro-rmoskdfp */  /*+ 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：2025-06-03 17:59:22

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。本文为您介绍云数据库 MySQL 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 MySQL 控制台 ，在实例列表，单击**实例 ID** 或操作列的**管理**，进入实例详情页面。
2. 在实例架构图下，可查看当前主实例下存在的只读分析引擎实例。

3. 单击目标只读分析引擎实例的**实例 ID**，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

和主实例、只读实例的方法一致。

修改安全组

和主实例、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 管理云数据库安全组 。

修改 IP 和端口

和主实例、只读实例的方法一致，请参考 修改内网地址 。

更改网络

和主实例、只读实例的方法一致，请参考 切换网络 。

开启或关闭外网

和主实例、只读实例的方法一致，请参考 开启外网连接地址 。

> **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

和主实例、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 销毁实例 。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。
按量计费实例删除后，实例被移入回收站保留24小时，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。

调整配置

最近更新时间：2025-04-27 17:57:22

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

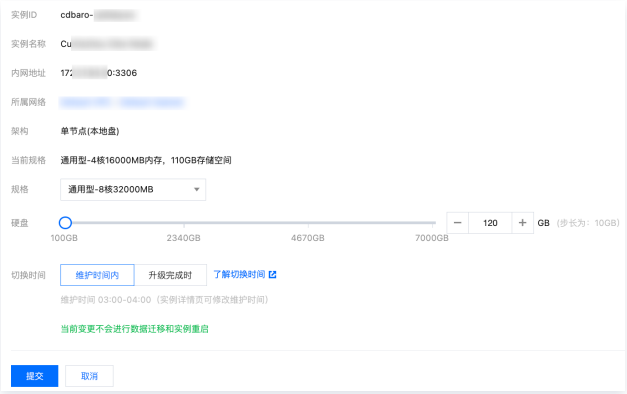
云数据库 MySQL 只读分析引擎实例的计算与存储并未分离，所以当遇到资源扩容、磁盘缩容时可能会出现跨服务器变配的场景。当实例变配时，会自动拉起一个新规格的实例节点，待节点数据同步后，自动将新的节点与旧的节点进行访问路由切换，最后完成变配操作。

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 MySQL 控制台 。
2. 在实例列表，找到需要调整配置的只读分析引擎实例，在其**操作**列选择**更多** > **调整配置**。

3. 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**提交**。



> **说明：**
> 调整配置是否会触发数据迁移和实例重启请以实际调整配置页面提示为准，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，立即触发切换。
> - 选择维护时间内，会在实例的维护时间内进行切换。修改实例维护时间可参见 设置实例维护时间 。

多节点高可用

最近更新时间：2025-04-27 17:57:22

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。

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 计费概述 。

调整节点数量

> **注意：**
> 只读分析引擎实例目前默认采用从云数据库 MySQL 的主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 MySQL 控制台 。
2. 在实例列表，找到目标只读分析引擎实例，在其**操作**列单击**管理**，进入实例详情页面。
3. 在**实例详情** > **实例节点**下，单击**节点变更**。
4. 在跳转页面中，根据需要新增或删除节点，然后单击**提交**。

- 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**提交**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。

5. 完成操作后即可返回实例详情页面，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 17:57:22

通过云数据库 MySQL 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和告警设置。

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 监控功能 。

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	QPS	LibraDBQps	Count/s	max
	总连接数	LibraDBConnections	Count	max/last
	错误请求数	LibraDBErrorQueries	Count	sum
	活跃连接数	LibraDBActiveConnections	Count	max/last
	返回结果集大小	LibraDBResultSize	MBytes	sum
	数据延迟时间	LibraDBLatency	s	max
	每秒同步处理行数	LibraDBSinkRowsPerSec	Count/s	max/avg
	全量加载时每秒写行数	LibraDBDumpRowsPerSec	Count/s	max/avg
	全量加载时每秒写数据大小	LibraDBDumpSizePerSec	MBytes/s	max/avg
	平均响应时间	LibraDBDuration	Ms	avg
分析引擎节点监控	CPU 使用率	LibraDBCpuUseRate	%	last
	内存使用率	LibraDBMemUseRate	%	last
	存储使用率	LibraDBStorageUseRate	%	last
	存储使用量	LibraDBStorageUsed	MBytes	last
	读吞吐	LibraDBReadThroughput	MBytes/s	max
	写吞吐	LibraDBWriteThroughput	MBytes/s	max
	网络入流量	LibraDBBytesReceived	MBytes	max/avg
	网络出流量	LibraDBBytesSent	MBytes	max/avg
	CompactRowset 数量	LibraDBCompactRowsets	Count	last
	收到的 MppTask 数	LibraDBMPPTasks	Count	sum
	正在运行的 MppTask 数	LibraDBRunningMPPTasks	Count	last
	数据文件大小	LibraDBDatafileSize	MBytes	last
	擎缓存使用量	LibraDBCached	MBytes	last
	缓存命中率	LibraDBCacheHitsRate	%	avg
	节点延迟时间	LibraDBNodeLatency	s	max
	节点每秒同步处理行数	LibraDBNodeSinkRowsPerSec	Count/s	max

	节点全量加载时每秒写行数	LibraDBNodeDumpRowsPerSec	Count/s	max
	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云数据库 MySQL 主实例和只读实例的方法一致，详细操作指导可参考 告警策略 。

管理参数

最近更新时间：2025-05-23 09:25:22

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，也不建议新增过多连接，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	此参数控制当 Group By 遇到空结果集时，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 17:49:32

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

- 准备好云数据库 MySQL 实例，请参见 创建 MySQL 实例 。
- 实例状态为运行中。

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主实例拉取数据使用。请不要删除此账号。
> - 在初次加载数据到只读分析引擎时，会从主实例拉取数据，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，分别是全对象、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. 数据加载方式为全对象。

   全对象模式下，云数据库 MySQL 实例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 MySQL 控制台 ，在实例列表，单击**实例 ID** 或**操作**列的**管理**，进入实例详情页面。
2. 在实例架构图下找到只读分析引擎实例，单击**实例 ID** 进入只读分析引擎详情页面。
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。

实例详情
数据库管理
安全组
运行中
只读分析引擎
主实例ID
cdb-
实例ID
cdbaro-
地域 / 可用区
华南地区（广州） /
内网地址
10 6
端口: 3306
外网地址
开启
所属网络
lit
更换网络
实例配置
独享型-4核16000MB内存，100GB存储空间
调整配置
分析引擎版本
1.2404.19.1
创建时间
2025-01-14 14:34:57
计费模式
按量计费
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
标签
编辑
数据对象
修改对象
数据加载限制
请输入对象名
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 17:23:32

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

- 准备好云数据库 MySQL 实例，请参见 创建 MySQL 实例 。
- 实例状态为运行中。
- 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 MySQL 控制台 ，在实例列表，单击**实例 ID** 或操作列的**管理**，进入实例详情页面。
2. 在实例架构图下找到只读分析引擎实例，单击**实例 ID** 进入只读分析引擎详情页面。
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”、“数据同步异常”、“数据同步完成”等，详细请以控制台为准。
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)
```

控制数据加载速度

最近更新时间：2025-01-21 15:47:03

操作场景

在进行数据全量加载的时候，会对云数据库 MySQL 的主实例造成一定的资源开销，特别是在数据量较大的实例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主实例的性能稳定性。可通过如下几种方式去控制在数据加载过程中对云数据库 MySQL 主实例的影响，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

- 准备好云数据库 MySQL 实例，请参见 创建 MySQL 实例 。
- 实例状态为运行中。
- 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 。

取消部分对象加载到只读分析引擎

最近更新时间：2025-01-21 15:47:03

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

- 准备好云数据库 MySQL 实例，请参见 创建 MySQL 实例 。
- 实例状态为运行中。
- 已 加载数据到只读分析引擎 。

操作步骤

方法一

下面为您介绍从只读分析引擎实例取消数据加载的步骤。

1. 登录 MySQL 控制台 ，在实例列表，单击只读分析引擎的**实例 ID** 或操作列的**管理**，进入只读分析引擎的实例详情页面。

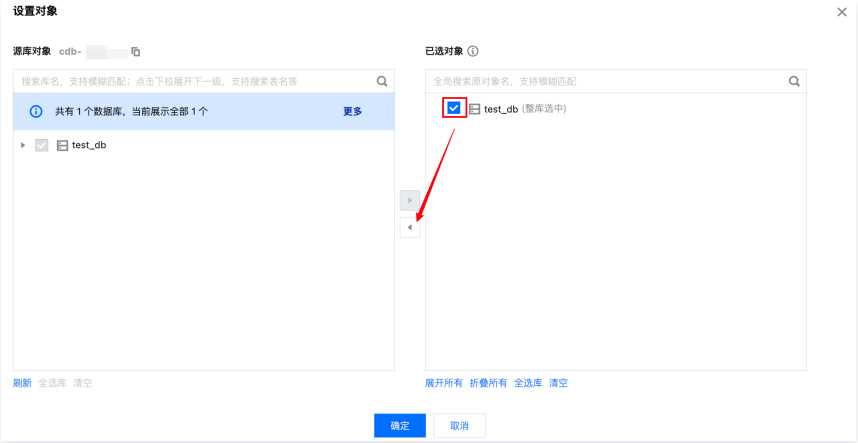
2. 在**数据对象**后单击**修改对象**。

3. 在弹窗中，在右侧的同步对象下取消已经加载到只读分析引擎的对象，单击**确定**。

方法二

下面为您介绍从挂载只读分析引擎实例的主实例详情页面进入，取消数据加载的步骤。

1. 登录 MySQL 控制台 ，在实例列表，单击**实例 ID** 或操作列的**管理**，进入实例详情页面。
2. 在实例架构图下找到只读分析引擎实例，单击**实例 ID** 进入只读分析引擎详情页面。
3. 单击**修改对象**，在弹窗中，在右侧的同步对象下取消已经加载到只读分析引擎的对象，单击**确定**。


设置对象
源库对象 cdb-
搜索库名，支持模糊匹配；点击下拉展开下一级，支持搜索表名等
共有 1 个数据库，当前展示全部 1 个
更多
test_db
已选对象
全局搜索原对象名，支持模糊匹配
test_db (整库选中)
刷新 全选库 清空
展开所有 折叠所有 全选库 清空
确定
取消

执行查询
收集统计信息

最近更新时间：2025-02-11 17:23:32

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。

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。
- **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>;
```

SELECT 语句说明

最近更新时间：2025-07-08 10:44:42

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，请通过云数据库 MySQL 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。

> **注意：**
> WITH ROLLUP 语法只有在 3.2503.x 及后续的版本中才支持。

- 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。

CTE 语法使用说明

最近更新时间：2025-02-11 17:23:32

云数据库 MySQL 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 21:26:42

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)
```

执行 SQL 并查看执行计划

最近更新时间：2025-05-21 14:17:22

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。

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。

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 filter，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：2025-05-21 14:17:22

云数据库 MySQL 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 session 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

- 主实例：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 15:38:42

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 设置实例参数 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：2025-01-21 15:47:03

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                                                   |
+-----------------------------------------------------------------------------------------------+
```

> **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。

```
+-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+
| query plan                                                                                                                                                                          |
+-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+
| ==========================================================                                                                                                                          |
| |ID|OPERATOR                        |NAME|EST.ROWS        |                                                                                                                                |
| ----------------------------------------------------------                                                                                                                          |
| |0 |PROJECTION                      |    |1250000000000.00|                                                                                                                                |
| |1 |└HASH JOIN                      |    |1250000000000.00|                                                                                                                                |
| |2 |  ├HASH JOIN (B)                |    |100000000.00    |                                                                                                                                |
| |3 |  │├TABLE FULL SCAN (B)        |t1  |10000.00        |                                                                                                                                |
| |4 |  │└TABLE FULL SCAN (P)        |t3  |10000.00        |                                                                                                                                |
| |5 |  └HASH JOIN (P)                |    |100000000.00    |                                                                                                                                |
| |6 |    ├TABLE FULL SCAN (B)        |t2  |10000.00        |                                                                                                                                |
| |7 |    └TABLE FULL SCAN (P)        |t4  |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.t1.a, singleton.t2.a) eq(singleton.t3.a, singleton.t2.a) eq(singleton.t3.a, singleton.t4.a)]                                      |
|   2 - NON EQUAL: CARTESIAN ,JOIN TYPE: inner join                                                                                                                                   |
|   3 - STORAGE: LIBRASTORE, BLOCK OFFSET: 1                                                                                                                                          |
|   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                                                                                                                                          |
+-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+
```

> **注意：**
> - 存在多个 LEADING Hint 时会导致 Hint 失效。
> - 优化器无法按照 LEADING 进行表连接时 Hint 会失效。

SET_VAR(NAME="VALUE")

使用说明

SET_VAR(XXXX="YY")，用于在 SQL 执行期间临时修改系统变量，在 SQL 执行完毕后，指定系统变量会自动恢复为原始值，使用方法如下所示。

参考示例

```
SELECT /*+ SET_VAR(max_threads=64) */ * FROM t1
```

此条 SQL 通过 SET_VAR Hint 临时指定了在 SQL 执行期间的最大线程数为 64。

> **注意：**
> 并不是所有参数都支持 SET_VAR Hint，使用前请确认目标参数可以支持 Hint 修改。支持通过 Hint 修改的 参数请参考 系统变量 。

NO_PX_JOIN_FILTER_ID(ID)/PX_JOIN_FILTER_ID(ID)

使用说明

no_px_join_filter_id(ID)/px_join_filter_id(ID)，用于控制优化器关闭或开启 RuntimeFilter。

参考示例

具体使用方式参考 Runtime Filter 使用手册 。

常见 Hint 问题

MYSQL 客户端消除 Hint 导致不生效

MySQL 命令行客户端在 5.7 版本之前默认清除了 Optimizer Hints。如果需要在这些早期版本的客户端中使用 Hint 语法，需要在启动客户端时加上 `--comments` 选项，例如：`mysql -h 127.0.0.1 -P 4000 -u root -c` 。

跨库查询不指定库名导致 Hint 不生效

对于查询中有需要跨库访问的表，需要显示的在 Hint 中指定数据库名称，否则可能会出现 Hint 不生效的问题。例如对于下面这条 SQL：

```
SELECT /*+ SHUFFLE_JOIN(t1) */ * FROM test1.t1, test2.t2 WHERE t1.id = t2.id;
```

由于当前 t1 表在当前 database 中，从而导致 Hint 失效。Warning 信息如下所示。

```
mysql> show warnings;
```

```
+---------+------+-----------------------------------------------------------------------------------------
-------------------------------------------------------------+
| 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:                                                                  |
|   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：

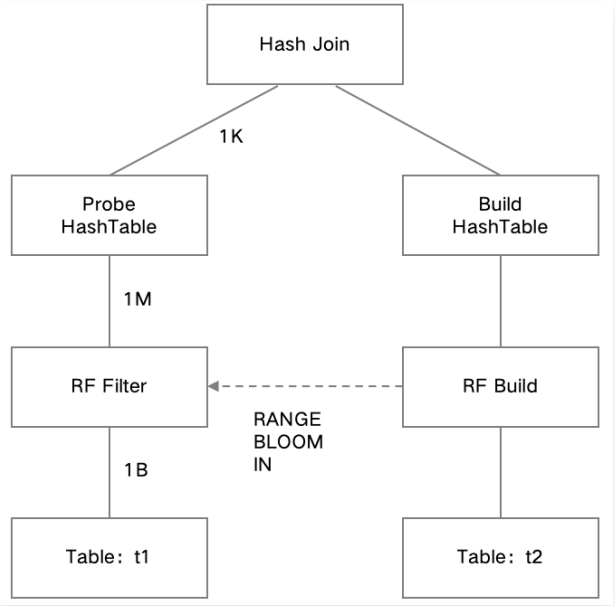
```
MySQL> SELECT * /*+ SET_VAR(max_threads = 64)) */ FROM t;
```

```
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：2025-01-21 15:47:03

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。

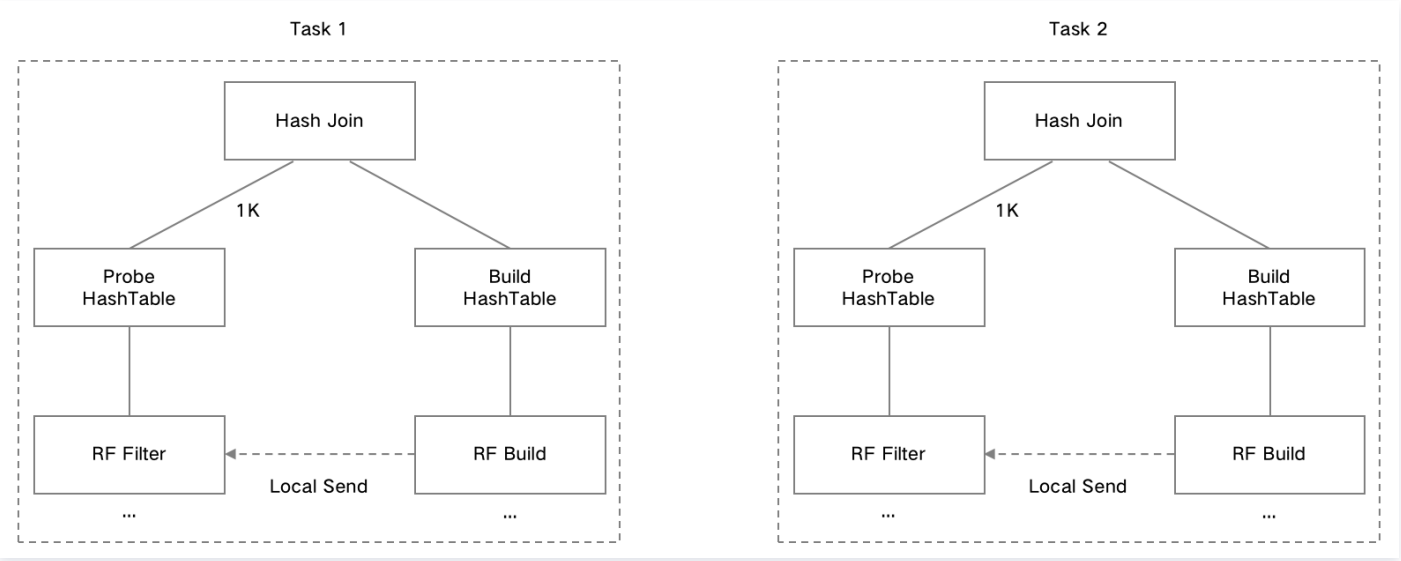


只读分析引擎的 Runtime Filter 由两部分构成，分别是 RF Build 和 RF Filter。RF Build 是在 HASH JOIN 的 Build 端，用来构建 Runtime Filter。RF Filter 则在对应 HASH JOIN 表的 Probe 端的 TableScan 上，用于提前过滤数据以提升性能。

Runtime Filter 类型

Local Runtime Filter

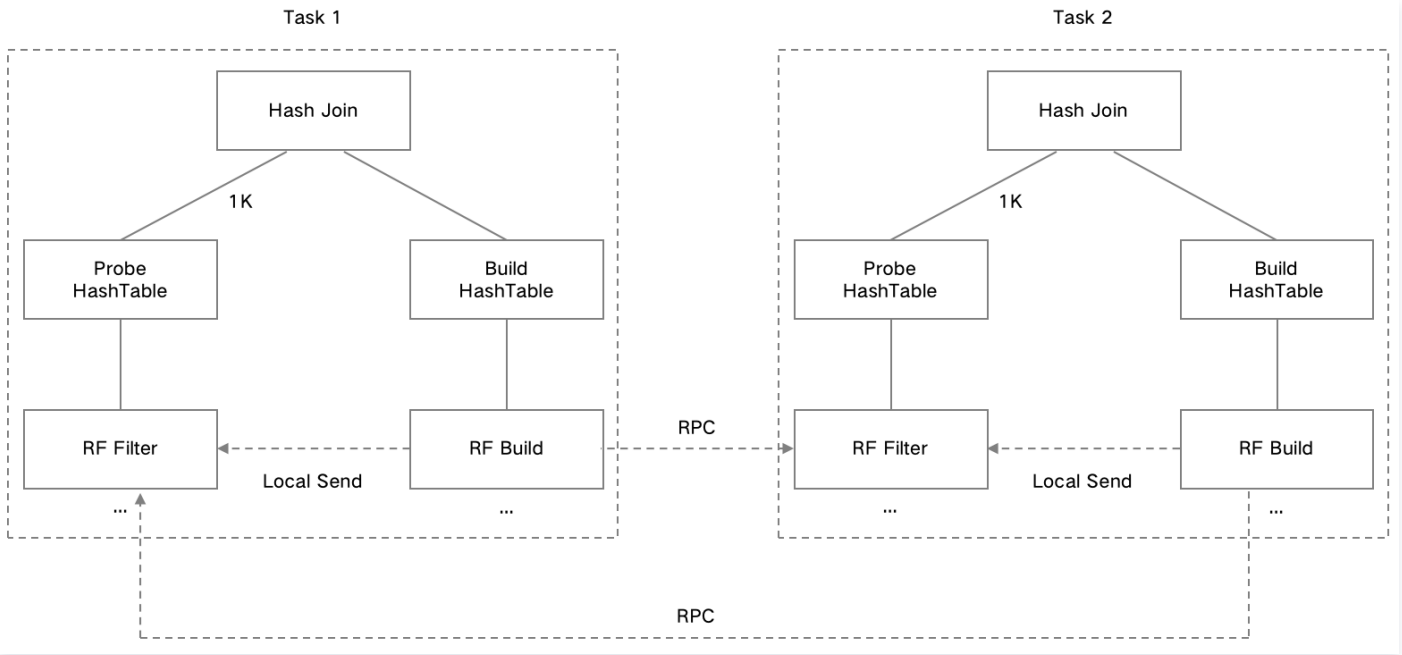
Local Runtime Filter 通常用于数据 JOIN 的时候没有被 Shuffle 的场景。此时，当前构建节点的 Runtime Filter 已经可以满足 Probe 的需要，就无须网络传输，直接将 Runtime Filter 的数据传递给 Probe 侧即可使用。



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。



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;
+-----------------------
```

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：2025-01-21 15:47:03

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 17:23:32

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：2025-05-21 14:17:22

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 IN，>，< 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 14:19:52

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

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 14:41:01

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：2025-05-23 09:25:22

云数据库 MySQL 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

- 准备好云数据库 MySQL 实例，请参见 创建 MySQL 实例，架构请选择双节点或三节点。
- 基于以上 MySQL 实例创建一个只读分析引擎，并为此只读分析引擎设置一个合理的实例规格，请参见 开启只读分析引擎 。
- 准备一个数据库账号，用于数据查询，请参见 创建账号 。

> **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，将表创建在云数据库 MySQL 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;
```

```
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 测试数据分别导入云数据库 MySQL 实例中。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

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
```

对于云数据库 MySQL 实例，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 数据加载 。
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 -P3306 -c
show replication tables;
```

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 -P3306 -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;
```

TPC-H 性能结果

最近更新时间：2025-01-21 15:47:03

本文档介绍云数据库 MySQL 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

标签
标签概述

最近更新时间：2024-10-18 14:14:23

简介

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），具体介绍请参见 标签概述 。
您可以根据各种维度（例如，业务、用途、负责人等）使用标签对云数据库 MySQL 资源进行分类管理。通过标签可以非常方便地筛选过滤出对应的资源。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，在使用过程中只需要注意 使用限制 即可。
下面通过一个具体的案例来介绍标签的使用。

案例背景

某公司在腾讯云上拥有10台云数据库 MySQL，分属电商、游戏、文娱三个部门，服务于营销活动、游戏 A、游戏 B、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云数据库 MySQL 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 A、游戏 B、后期制作
运维负责人	张三、李四、王五

将这些标签键/值绑定到云数据库 MySQL 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
cdb-abcdef1	电商	营销活动	王五
cdb-abcdef2	电商	营销活动	王五
cdb-abcdef3	游戏	游戏 A	张三
cdb-abcdef3	游戏	游戏 B	张三
cdb-abcdef4	游戏	游戏 B	张三
cdb-abcdef5	游戏	游戏 B	李四
cdb-abcdef6	游戏	游戏 B	李四
cdb-abcdef7	游戏	游戏 B	李四
cdb-abcdef8	文娱	后期制作	王五
cdb-abcdef9	文娱	后期制作	王五
cdb-abcdef10	文娱	后期制作	王五

使用标签

创建与删除标签方法请参见 操作指南 。
编辑云数据库 MySQL 标签方法请参见 编辑标签 。

版权所有：腾讯云计算（北京）有限责任公司

编辑标签

最近更新时间：2024-10-18 14:14:23

您可以按照下列操作对资源进行标签的编辑操作。

对单个实例编辑标签

1. 登录 MySQL 控制台，在实例列表，选择**操作**列的**更多** > **编辑标签**。
2. 在弹出的对话框，可添加、修改、删除标签，确认无误后，单击**确定**。

对多个实例编辑标签

1. 登录 MySQL 控制台，在实例列表，勾选需要编辑标签的实例，在列表上方选择**更多操作** > **编辑标签**。
2. 在弹出的对话框，可添加、修改、删除标签，确认无误后，单击**确定**。

版权所有：腾讯云计算（北京）有限责任公司