



# 云数据库 GaussDB(for MySQL)

## 用户指南

天翼云科技有限公司

---

# 目 录

---

<b>1 产品介绍</b> .....	<b>8</b>
1.1 什么是云数据库 GaussDB(for MySQL).....	8
1.2 产品架构.....	8
1.3 常用概念.....	9
1.4 产品优势和价值.....	10
1.5 实例说明.....	10
1.5.1 数据库实例规格.....	10
1.5.2 数据库实例状态.....	11
1.5.3 数据库引擎和版本.....	11
1.6 权限管理.....	12
1.7 使用限制.....	15
1.8 GaussDB(for MySQL)与其他服务的关系.....	16
<b>2 快速入门</b> .....	<b>18</b>
2.1 创建实例.....	18
2.2 连接实例.....	21
2.2.1 实例连接方式简介.....	21
2.2.2 通过内网连接 GaussDB(for MySQL)实例.....	23
2.2.2.1 内网连接实例流程.....	23
2.2.2.2 设置安全组规则.....	24
2.2.2.3 通过内网连接 GaussDB(for MySQL)实例.....	26
2.2.3 通过公网连接 GaussDB(for MySQL)实例.....	32
2.2.3.1 公网连接实例流程.....	32
2.2.3.2 绑定弹性公网 IP.....	32
2.2.3.3 设置安全组规则.....	33
2.2.3.4 通过公网连接 GaussDB(for MySQL)实例.....	35
<b>3 用户指南</b> .....	<b>41</b>
3.1 计费管理.....	41
3.1.1 续费.....	41
3.1.2 包周期实例转按需.....	42
3.1.3 按需实例转包周期.....	43

---

3.1.4 退订包周期实例.....	44
3.2 数据迁移.....	45
3.2.1 使用 mysqldump 迁移 GaussDB(for MySQL)数据.....	45
3.3 实例生命周期管理.....	49
3.3.1 修改实例名称.....	49
3.3.2 修改实例备注.....	49
3.3.3 删除实例.....	50
3.3.4 重启实例.....	51
3.3.5 重启节点.....	52
3.3.6 修改节点名称.....	53
3.3.7 导出实例.....	53
3.3.8 回收站.....	54
3.4 变更实例.....	54
3.4.1 变更实例的 CPU 和内存规格.....	54
3.4.2 磁盘容量变更（包年包月）.....	56
3.4.3 设置可维护时间段.....	57
3.4.4 自定义列表项.....	58
3.4.5 升级内核小版本.....	59
3.5 只读节点.....	60
3.5.1 只读节点简介.....	60
3.5.2 创建只读节点.....	60
3.5.3 管理只读节点.....	61
3.5.4 只读节点升主节点.....	61
3.5.5 删除只读节点.....	62
3.6 数据安全性.....	63
3.6.1 重置管理员密码.....	63
3.6.2 修改实例内网安全组.....	65
3.6.3 数据库安全设置.....	65
3.7 数据备份.....	66
3.7.1 备份类型.....	66
3.7.2 设置自动备份策略.....	67
3.7.3 创建手动备份.....	68
3.7.4 导出备份信息.....	69
3.7.5 删除手动备份.....	70
3.8 数据恢复.....	71
3.8.1 恢复方案概览.....	71
3.8.2 将数据库实例恢复到指定时间点.....	71
3.8.3 将备份恢复至实例.....	72
3.9 连接管理.....	74

3.9.1 绑定和解绑弹性公网 IP .....	74
3.9.2 修改数据库端口 .....	76
3.10 参数模板管理 .....	77
3.10.1 创建参数模板 .....	77
3.10.2 编辑参数模板 .....	78
3.10.3 导出参数 .....	79
3.10.4 比较参数模板 .....	80
3.10.5 查看参数修改历史 .....	81
3.10.6 复制参数模板 .....	82
3.10.7 重置参数模板 .....	83
3.10.8 应用参数模板 .....	83
3.10.9 查看参数模板应用记录 .....	84
3.10.10 修改参数模板描述 .....	85
3.10.11 删除参数模板 .....	85
3.11 监控指标与告警 .....	86
3.11.1 支持的监控指标 .....	86
3.11.2 查看监控指标 .....	93
3.11.2.1 查看实例监控指标 .....	93
3.11.3 设置告警规则 .....	94
3.11.3.1 创建实例告警规则 .....	94
3.11.4 设置秒级监控 .....	96
3.12 日志管理 .....	98
3.12.1 查看日志的方法 .....	98
3.12.2 查看错误日志 .....	98
3.12.3 查看慢日志 .....	99
3.12.4 设置全量 SQL .....	100
3.13 任务中心 .....	101
3.13.1 查看任务 .....	101
3.13.2 删除任务 .....	102
3.14 标签 .....	102
<b>4 常见问题 .....</b>	<b>105</b>
4.1 产品咨询 .....	105
4.1.1 使用 GaussDB(for MySQL)要注意些什么 .....	105
4.1.2 为何使用了 GaussDB(for MySQL)后网站登录较慢 .....	105
4.1.3 GaussDB(for MySQL)如何自动进行故障切换 .....	106
4.1.4 GaussDB(for MySQL)是否支持计算存储分离 .....	106
4.2 数据库连接 .....	106
4.2.1 外部服务器能否访问 GaussDB(for MySQL)数据库 .....	106
4.2.2 GaussDB(for MySQL)数据库连接数满的排查思路 .....	106

---

4.2.3 GaussDB(for MySQL)数据库实例支持的最大数据连接数是多少 .....	107
4.2.4 ECS 无法连接到 GaussDB(for MySQL)实例的原因 .....	108
4.2.5 如何通过 JDBC 连接 MySQL 数据库 .....	108
4.2.6 如何创建和连接 ECS .....	114
4.2.7 客户端问题导致连接失败 .....	115
4.2.8 绑定公网 IP 后无法 ping 通的解决方案 .....	115
4.2.9 测试连通性失败，如何排查 .....	116
4.2.10 GaussDB(for MySQL)跨地域内网能访问吗 .....	117
4.2.11 GaussDB(for MySQL)实例连接数过多，存在什么隐患 .....	117
4.2.12 ECS 和 GaussDB(for MySQL)部署在同一区域的不同的 VPC 内，网络不通怎么办 .....	117
4.2.13 如何查看当前时间所有连接数据库的 IP .....	117
4.3 安装客户端 .....	118
4.3.1 如何安装 MySQL 客户端 .....	118
4.4 数据库迁移 .....	119
4.4.1 GaussDB(for MySQL)支持导入哪些数据库引擎的数据 .....	119
4.5 数据库权限 .....	119
4.5.1 GaussDB(for MySQL)提供 root 帐号或 super 权限吗 .....	119
4.6 数据库性能 .....	120
4.6.1 GaussDB(for MySQL) CPU 使用率高的解决方法 .....	120
4.6.2 联合索引设置不当导致慢 SQL 的解决办法 .....	120
4.6.3 长事务产生大量临时表导致内存超限的解决办法 .....	122
4.6.4 持锁长事务导致后续业务报等锁超时的解决办法 .....	123
4.7 数据库基本使用 .....	124
4.7.1 主备机分别执行 MATCH AGAINST 语句后结果不一致的原因 .....	124
4.7.2 使用 LOAD DATA 导入本地数据 .....	124
4.8 备份与恢复 .....	126
4.8.1 GaussDB(for MySQL)能够保存多长时间的备份 .....	126
4.8.2 如何清理云数据库 GaussDB(for MySQL)的备份空间 .....	126
4.8.3 如何将 GaussDB(for MySQL)数据库备份到弹性云服务器上 .....	127
4.8.4 如何查看 GaussDB(for MySQL)备份空间使用情况 .....	127
4.8.5 为什么自动备份会失败 .....	128
4.8.6 GaussDB(for MySQL)的备份是如何收费的 .....	128
4.9 数据库参数修改 .....	128
4.9.1 如何修改时区 .....	128
4.9.2 GaussDB(for MySQL)密码过期策略 .....	129
4.9.3 如何修改 GaussDB(for MySQL)数据库字符集 .....	130
4.9.4 使用 utf8mb4 字符集存储 emoji 表情到 GaussDB(for MySQL)实例 .....	132
4.9.5 GaussDB(for MySQL)如何设置表名大小写敏感 .....	133
4.9.6 是否支持使用 SQL 命令修改全局参数 .....	133

---

4.10 网络安全.....	134
4.10.1 GaussDB(for MySQL)有哪些安全保障措施.....	134
4.10.2 如何防止任意源连接数据库.....	134
4.10.3 访问 GaussDB(for MySQL)实例应该如何配置安全组.....	134
4.10.4 将根证书导入 Windows/Linux 操作系统.....	135
4.10.5 如何管理 GaussDB(for MySQL)安全性.....	136
4.11 日志管理.....	137
4.11.1 如何查看 GaussDB(for MySQL)执行过的所有 SQL 日志.....	137
4.11.2 如何查看 GaussDB(for MySQL)慢 SQL? .....	137
4.11.3 GaussDB(for MySQL)服务如何开启并查看 binlog 文件? .....	138
4.11.4 如何修改 binlog 保留时长.....	139
4.11.5 如何查看 GaussDB(for MySQL)数据库的死锁日志 .....	139
4.12 版本升级.....	140
4.12.1 如何查看云数据库 GaussDB(for MySQL)实例的版本号.....	140
4.12.2 GaussDB(for MySQL)是否支持版本升级.....	141
<b>5 故障排除.....</b>	<b>142</b>
5.1 备份恢复.....	142
5.1.1 mysqldump 导出数据报错权限不足.....	142
5.1.2 使用 mysqlbinlog 工具获取 binlog.....	142
5.1.3 canal 解析 binlog 报错.....	143
5.1.4 使用 mysqldump 导出大表的注意事项.....	144
5.1.5 mysqldump 的 6 大使用场景的导出命令.....	144
5.1.6 增加表字段后出现运行卡顿现象 .....	145
5.1.7 怎么解决执行 mysqldump 出现 SET @@SESSION.SQL_LOG_BIN 等 SQL 的问题.....	146
5.1.8 canal 工具报错权限不足.....	147
5.2 连接类.....	147
5.2.1 root 账号的 ssl_type 修改为 ANY 后无法登录.....	147
5.2.2 SSL 使用与介绍.....	148
5.2.3 对各个 IP 地址的解释说明.....	149
5.2.4 客户端 TLS 版本 MySQL 不一致导致 SSL 连接失败.....	150
5.2.5 GaussDB(for MySQL)建立连接慢导致客户端超时 .....	151
5.2.6 连接数据库报错 Access denied.....	152
5.2.7 mariadb-connector SSL 方式连接数据库失败.....	154
5.2.8 使用 root 帐号连接数据库失败.....	154
5.2.9 客户端连接实例后会自动断开.....	155
5.2.10 istio-citadel 证书机制导致每隔 45 天出现断连.....	157
5.3 SQL 类.....	157
5.3.1 建表时 timestamp 字段默认值无效.....	157
5.3.2 索引长度限制导致修改 varchar 长度失败.....	158

5.3.3 delete 大表数据后，再查询同一张表时出现慢 SQL.....	159
5.3.4 更新 emoji 表情数据报错 Error 1366.....	160
5.3.5 存储过程和相关表字符集不一致导致执行缓慢.....	161
5.3.6 报错 ERROR [1412]的解决方法.....	161
5.3.7 存在外键的表无法删除.....	162
5.3.8 GROUP_CONCAT 结果不符合预期.....	163
5.3.9 创建二级索引报错 Too many keys specified.....	164
5.3.10 distinct 与 group by 优化.....	164
5.3.11 为什么有时候用浮点数做等值比较查不到数据.....	166
5.3.12 表空间膨胀问题.....	167
5.3.13 MySQL 创建用户提示服务器错误(ERROR 1396).....	168
5.3.14 执行 alter table xxx discard/import tablespace 报错.....	170
5.3.15 数据库报错 Native error 1461 的解决方案.....	170
5.3.16 创建表失败报错 Row size too large 的解决方案.....	170
5.4 参数类.....	171
5.4.1 客户端修改全局参数失败.....	171
5.4.2 客户端超时参数设置导致连接超时退出.....	172
5.4.3 修改全局变量成功但未生效.....	172
5.4.4 GaussDB(for MySQL) timeout 相关参数简介.....	173
5.5 性能资源类.....	174
5.5.1 冷热数据问题导致 SQL 执行速度慢.....	174
5.5.2 复杂查询造成磁盘满.....	175
5.5.3 业务死锁导致响应变慢.....	176
5.5.4 GaussDB(for MySQL)实例 CPU 升高定位思路.....	177
5.5.5 大并发慢查询导致 CPU 资源耗尽问题.....	179
5.6 基本使用类.....	181
5.6.1 查看 GaussDB(for MySQL)的存储容量.....	181
5.6.2 修改库名和修改表名.....	182
5.6.3 字符集和字符序的默认选择方式.....	183
5.6.4 自增字段值跳变的原因.....	184
5.6.5 表的自增 AUTO_INCREMENT 初值与步长.....	190
5.6.6 修改表的自增 AUTO_INCREMENT 值.....	192
5.6.7 自增主键达到上限，无法插入数据.....	194
5.6.8 自增字段取值.....	195
5.6.9 自增属性 AUTO_INCREMENT 为何未在表结构中显示.....	199
5.6.10 空用户的危害.....	200
5.6.11 慢日志显示 SQL 语句扫描行数为 0.....	202

# 1 产品介绍

## 1.1 什么是云数据库 GaussDB(for MySQL)

云数据库 GaussDB(for MySQL)是最新一代企业级高扩展海量存储分布式数据库，完全兼容 MySQL。采用计算存储分离架构，128TB 的海量存储，故障秒级切换，既拥有商业数据库的高可用和性能，又具备开源低成本效益。

GaussDB(for MySQL)支持的版本请参见[数据库引擎和版本](#)。

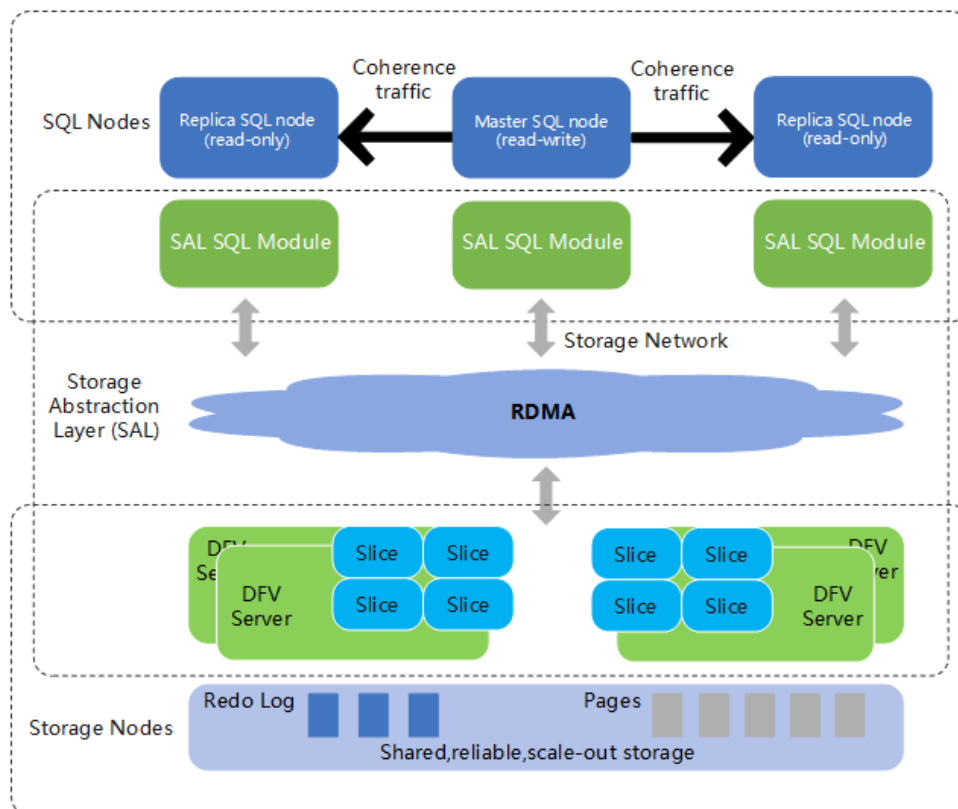
## 1.2 产品架构

云数据库 GaussDB(for MySQL)整体架构自下向上分为三层。

1. 存储层：基于 DFV 存储，提供分布式、强一致和高性能的存储能力，此层来保障数据的可靠性以及横向扩展能力，保证数据的可靠性不低于 99.999999999%。DFV (Data Functions Virtualization)是提供的一套通过存储和计算分离的方式，构建以数据为中心的全栈数据服务架构的解决方案。
2. 存储抽象层 (Storage Abstraction Layer)：将原始数据库基于表文件的操作抽象为对应分布式存储，向下对接 DFV，向上提供高效调度的数据库存储语义，是数据库高性能的核心。
3. SQL 解析层：与 MySQL 8.0 开源版 100%兼容，客户业务从 MySQL 生态可以平滑迁移，从其他数据库迁移也能使用 MySQL 生态的语法、工具，降低开发、学习成本。基于原生 MySQL，在 100%兼容的前提下进行大量内核优化，以及开源加固，开源生态，商用能力。



图1-1 架构图



## 1.3 常用概念

了解以下概念，有助于您更好地选购和使用 GaussDB(for MySQL)：

- **区域 (Region)：**从地理位置和网络时延维度划分，同一个 Region 内共享弹性计算、块存储、对象存储、VPC 网络、弹性公网 IP、镜像等公共服务。Region 分为通用 Region 和专属 Region，通用 Region 指面向公共租户提供通用云服务的 Region；专属 Region 指只承载同一类业务或只面向特定租户提供业务服务的专用 Region。一般情况下，GaussDB(for MySQL)实例应该和弹性云服务器实例位于同一地域，以实现最高的访问性能。
- **可用区 (Availability Zone, 简称 AZ)：**一个 AZ 是一个或多个物理数据中心的集合，有独立的风火水电，AZ 内逻辑上再将计算、网络、存储等资源划分成多个实例，可用区是指在某个地域内拥有独立电力和网络的物理区域。  
可用区之间内网互通，不同可用区之间物理隔离。每个可用区都不受其他可用区故障的影响，并提供低价、低延迟的网络连接，以连接到同一地区其他可用区。通过使用独立可用区内的 GaussDB(for MySQL)，可以保护您的应用程序不受单一位置故障的影响。同一 Region 的不同 AZ 之间没有实质性区别。
- **规格：**每个节点的资源配置，比如 16 核 64GB。

## 1.4 产品优势和价值

GaussDB(for MySQL)为计算存储分离、云化架构的企业级云数据库。

### 超高性能

相比开源 MySQL，性能提升 7 倍，可达到百万级 QPS。

### 高扩展性

- 横向扩展：支持分钟级添加只读节点，最大支持 15 只读，解决性能扩展问题。
- 纵向扩展：支持规格升级，应对不确定的业务增长。
- 存储扩展：根据数据容量自动弹性伸缩，无须提前规划容量，最大支持 128TB，解决海量数据问题。

### 高可靠性

支持跨 AZ 部署、异地容灾，金融级别可靠性。

跨 AZ 部署，数据三副本，安全性有保障。

### 高兼容性

100% 兼容 MySQL，应用上云无须改造。

### 非中间件式架构

业务性能正常情况下，无需搭载分布式数据库中间件分库分表。

## 1.5 实例说明

### 1.5.1 数据库实例规格

表1-1 X86 性能规格

规格	vCPU (个)	内存 (GB)
独享版	2	16
	4	32
	8	64
	16	128
	32	256
	64	512

数据库实例规格请以实际环境为准。

#### 说明

可通过调整 `max_connections` 参数值来修改最大连接数。

## 1.5.2 数据库实例状态

### 数据库实例状态

数据库实例状态是数据库实例的运行情况。用户可以使用管理控制台查看数据库实例状态。

表1-2 状态及说明

状态	说明
正常	数据库实例正常和可用。
异常	数据库实例不可用。
创建中	正在创建数据库实例。
创建失败	数据库实例创建失败。
重启中	正在重启数据库实例。
实例名称修改中	正在修改数据库实例名称。
端口修改中	正在修改数据库实例的数据库端口。
规格变更中	正在变更数据库实例的 CPU 和内存规格。
添加只读中	正在进行数据库实例添加只读节点。
删除只读中	正在进行数据库实例删除只读节点。
只读升主中	只读节点正在切换为主节点。
备份中	正在备份数据库实例。
证书配置变更中	正在进行数据库实例证书配置变更。
已删除	数据库实例已被删除，对于已经删除的实例，将不会在实例列表中显示。

## 1.5.3 数据库引擎和版本

GaussDB(for MySQL)目前支持的数据库引擎和版本如表 1-3 所示。

表1-3 数据库引擎和版本

数据库引擎	兼容的数据库版本
-------	----------

数据库引擎	兼容的数据库版本
GaussDB(for MySQL)	MySQL 8.0

## 1.6 权限管理

如果帐号已经能满足您的要求，不需要创建独立的 IAM 用户进行权限管理，您可以跳过本章节，不影响您使用 GaussDB(for MySQL)的其他功能。

如果您需要对购买的 GaussDB(for MySQL)资源，为企业中的员工设置不同的访问权限，为达到不同员工之间的权限隔离，您可以使用统一身份认证服务（Identity and Access Management，简称 IAM）进行精细的权限管理。该服务提供用户身份认证、权限分配、访问控制等功能，可以帮助您安全的控制资源的访问。

通过 IAM，您可以在帐号中给员工创建 IAM 用户，并授权控制员工对资源的访问范围。例如您的员工中有负责软件开发的人员，您希望开发人员拥有 GaussDB(for MySQL)的使用权限，但是不希望拥有删除 GaussDB(for MySQL)等高危操作的权限，那么您可以使用 IAM 为开发人员创建用户，通过授予仅能使用 GaussDB(for MySQL)，但是不允许删除 GaussDB(for MySQL)的权限，控制开发人员对 GaussDB(for MySQL)资源的使用范围。

IAM 是提供权限管理的基础服务，无需付费即可使用，您只需要为您帐号中的资源进行付费。

### GaussDB(for MySQL)权限

默认情况下，管理员创建的 IAM 用户没有任何权限，需要将其加入用户组，并给用户组授予策略或角色，才能使得用户组中的用户获得对应的权限，这一过程称为授权。授权后，用户就可以基于被授予的权限对云服务进行操作。

GaussDB(for MySQL)部署时通过物理区域划分，为项目级服务。授权时，“作用范围”需要选择“区域级项目”，然后在指定区域对应的项目中设置相关权限，并且该权限仅对此项目生效；如果在“所有项目”中设置权限，则该权限在所有区域项目中都生效。访问 GaussDB(for MySQL)时，需要先切换至授权区域。

- 策略：IAM 最新提供的一种细粒度授权的能力，可以精确到具体服务的操作、资源以及请求条件等。基于策略的授权是一种更加灵活的授权方式，能够满足企业对权限最小化的安全管控要求。例如：针对云数据库 GaussDB(for MySQL)，管理员能够控制 IAM 用户仅能对某一类数据库资源进行指定的管理操作。

如表 1-4 所示，包括了云数据库 GaussDB(for MySQL)的所有系统权限。

表1-4 GaussDB(for MySQL)系统权限

策略名称	描述	类别
GaussDB FullAccess	云数据库 GaussDB(for MySQL)服务的所有执行权	系统策略

策略名称	描述	类别
	限。	
GaussDB ReadOnlyAccess	云数据库 GaussDB(for MySQL)服务的只读访问权限。	系统策略

表 1-5 列出了云数据库 GaussDB(for MySQL)常用操作与系统权限的授权关系，您可以参照该表选择合适的系统权限。

表1-5 常用操作与系统权限的关系

操作	GaussDB FullAccess	GaussDB ReadOnlyAccess
创建 GaussDB(for MySQL)实例	支持	不支持
删除 GaussDB(for MySQL)实例	支持	不支持
查询 GaussDB(for MySQL)实例列表	支持	支持

表1-6 常用操作与对应授权项

操作名称	授权项	备注
修改参数模板	gaussdb:param:modify	-
变更数据库实例的规格	gaussdb:instance:modifySpec	-
创建数据库实例	gaussdb:instance:create	界面选择 VPC、子网、安全组需要配置： vpc:vpcs:list vpc:vpcs:get vpc:subnets:get vpc:securityGroups:get 创建加密实例需要在项目上配置 KMS Administrator 权限。 创建包周期实例需要配置 CBC 权限： bss:renewal:view bss:renewal:update bss:balance:view

操作名称	授权项	备注
		bss:order:view
创建手动备份	gaussdb:backup:create	-
查询备份列表	gaussdb:backup:list	-
查询错误日志	gaussdb:log:list	-
重启实例	gaussdb:instance:restart	-
查询数据库实例列表	gaussdb:instance:list	-
创建参数模板	gaussdb:param:create	-
删除参数模板	gaussdb:param:delete	-
修改备份策略	gaussdb:instance:modifyBackupPolicy	-
查看参数模板	gaussdb:param:list	-
删除实例	gaussdb:instance:delete	-
删除手动备份	gaussdb:backup:delete	-
查询项目标签	gaussdb:tag:list	-
应用参数模板	gaussdb:param:apply	-
批量添加删除项目标签	gaussdb:instance:dealTag	-
变更配额	gaussdb:quota:modify	-
升级数据库实例版本	gaussdb:instance:upgrade	-
只读升主	gaussdb:instance:switchover	-
修改数据库端口	gaussdb:instance:modifyPort	-
修改实例安全组	gaussdb:instance:modifySecurityGroup	-
修改读写内网地址	gaussdb:instance:modifyIp	界面选择 ip 需要配置: vpc:vpcs:listvpc:vpcs:get
开启、关闭 SSL	gaussdb:instance:modifySSL	-
修改实例名称	gaussdb:instance:rename	-
添加只读节点	gaussdb:instance:addNodes	-
删除只读节点	gaussdb:instance:deleteNodes	-
修改存储空间	gaussdb:instance:modifyStorage	-

操作名称	授权项	备注
	rageSize	
修改数据库实例密码	gaussdb:instance:modifyPassword	-
绑定公网 IP	gaussdb:instance:bindPublicIp	界面列出公网 IP 需要配置： vpc:publicIps:getvpc:publicIps:list
解绑公网 IP	gaussdb:instance:unbindPublicIp	-
修改监控策略	gaussdb:instance:modifyMonitorPolicy	-
修改节点倒换优先级	gaussdb:instance:modifySwitchoverPriority	-
修改可维护时间窗	gaussdb:instance:modifyMaintenanceWindow	-
节点隔离	gaussdb:instance:isolateNodes	-
创建数据库用户	gaussdb:user:create	-
删除数据库用户	gaussdb:user:delete	-
修改数据库用户密码	gaussdb:user:modify	-
查询数据库用户	gaussdb:user:list	-
数据库用户授权	gaussdb:user:grantPrivilege	-
回收数据库用户权限	gaussdb:user:revokePrivilege	-
创建数据库	gaussdb:database:create	-
删除数据库	gaussdb:database:delete	-
查询数据库列表	gaussdb:database:list	-

## 1.7 使用限制

GaussDB(for MySQL)使用上有一些固定限制，用来提高实例的稳定性和安全性，具体详见[表 1-7](#)。

表1-7 功能使用限制

功能	使用限制
GaussDB(for MySQL)访问	<ul style="list-style-type: none"> <li>如果 GaussDB(for MySQL)数据库实例没开通公网访问，则该实例必须与弹性云服务器在同一个虚拟私有云内才能访问。</li> <li>弹性云服务器必须处于目标 GaussDB(for MySQL)数据库实例所属安全组允许访问的范围内。 如果 GaussDB(for MySQL)数据库实例与弹性云服务器处于不同的安全组，系统默认不能访问。需要在 GaussDB(for MySQL)数据库的安全组添加一条“入”的访问规则。“入”规则开放 TCP 协议，使用数据库实例的默认端口。</li> <li>GaussDB(for MySQL)数据库实例的默认端口： 主备版默认端口为 3306，需用户手动修改端口号后，ECS 或外网才能访问其他端口。具体操作请参见<a href="#">修改数据库端口</a>。</li> </ul>
数据库的 root 权限	创建实例页面只提供管理员 root 用户权限（仅限主备版）。
修改数据库参数设置	大部分数据库参数可以通过控制台进行修改。
数据迁移	使用 mysqldump 迁移 GaussDB(for MySQL)数据。
重启 GaussDB(for MySQL)实例	无法通过命令行重启，必须通过 GaussDB(for MySQL)的管理控制台重启实例。
查看 GaussDB(for MySQL)备份	GaussDB(for MySQL)数据库实例在对象存储服务上的备份文件，对用户不可见。

## 1.8 GaussDB(for MySQL)与其他服务的关系

GaussDB(for MySQL)与其他服务之间的关系，具体表 1-8 所示。

表1-8 与其他服务的关系

相关服务	交互功能
弹性云服务器 (ECS)	GaussDB(for MySQL)配合弹性云服务器 (Elastic Cloud Server, 简称 ECS) 一起使用，通过内网连接 GaussDB(for MySQL)可以有效的降低应用响应时间、节省公网流量费用。
虚拟私有云 (VPC)	对您的 GaussDB(for MySQL)数据库实例进行网络隔离和访问控制。
对象存储服务 (OBS)	存储您的 GaussDB(for MySQL)数据库实例的自动和手动备份数据。



相关服务	交互功能
云监控服务 (Cloud Eye)	云监控服务是一个开放性的监控平台，帮助用户实时监测 GaussDB(for MySQL)资源的动态。云监控服务提供多种告警方式以保证及时预警，为您的服务正常运行保驾护航。
云审计服务 (CTS)	云审计服务 (Cloud Trace Service, 简称 CTS)，为用户提供云服务资源的操作记录。
数据复制服务 (DRS)	使用数据复制服务，实现数据库平滑迁移上云。
分布式数据库中间件 (DDM)	对于云数据库 GaussDB(for MySQL)，使用分布式数据库中间件服务 (Distributed Database Middleware, 简称 DDM)，后端对接多个数据库实例，实现分布式数据库的透明访问。

# 2 快速入门

## 2.1 创建实例

### 操作场景

本文将介绍在 GaussDB(for MySQL)数据库服务的管理控制台创建实例的过程。

### 操作步骤



- 步骤 1 登录管理控制台。
- 步骤 2 单击管理控制台左上角的 ，选择区域和项目。
- 步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。
- 步骤 4 在“实例管理”页面，单击“创建数据库实例”。
- 步骤 5 在“服务选型”页面，填写并选择实例相关信息后，单击“立即创建”。

表2-1 基本信息

参数	描述
区域	实例所在区域，也可在实例管理页面左上角切换。  须知 不同区域内的产品内网不互通，且创建后不能更换，请谨慎选择。
实例名称	实例名称长度在 4 个到 64 个字符之间，必须以字母开头，可以包含字母、数字、中划线或下划线，不能包含其他特殊字符。
数据库引擎	GaussDB(for MySQL)。
兼容的数据库版本	MySQL 8.0。
可用区类型	可用区指在同一区域下，电力、网络隔离的物理区域，可用区之间内网互通，不同可用区之间物理隔离。

参数	描述
时区	由于世界各国与地区经度不同，地方时也有所不同，因此会划分为不同的时区。时区可在创建实例时选择，后期不可修改。

表2-2 规格与存储

参数	描述
性能规格	不同性能规格对应不同连接数和最大 IOPS。 创建成功后可进行规格变更，请参见 <a href="#">变更实例的 CPU 和内存规格</a> 。
节点数量	每个实例默认只有 1 个主节点，其余节点为只读节点，“包年/包月”和“按需计费”实例单次创建最多可批量申请 9 个只读节点。 实例创建成功后，您可根据业务需要，添加只读节点，具体请参见 <a href="#">创建只读节点</a> 。
存储设置	您的存储空间会有必要的文件系统开销，这些开销包括索引节点和保留块，以及数据库运行必需的空间。创建时无需选择容量。

表2-3 网络

参数	描述
虚拟私有云	GaussDB(for MySQL)数据库实例所在的虚拟专用网络，可以对不同业务进行网络隔离。您需要创建或选择所需的虚拟私有云。 如果没有可选的虚拟私有云，GaussDB(for MySQL)数据库服务默认为您分配资源。
内网安全组	内网安全组限制实例的安全访问规则，加强 GaussDB(for MySQL)数据库服务与其他服务间的安全访问。请确保所选取的内网安全组允许客户端访问数据库实例。 如果不创建内网安全组或没有可选的内网安全组，GaussDB(for MySQL)数据库服务默认为您分配内网安全组资源。

表2-4 数据库配置

参数	描述
管理员账户名	数据库的登录名默认为 root。
管理员密码	所设置的密码，长度为 8~32 个字符，至少包含以下字符中的 3 种：大写字母、小写字母、数字、特殊字符~!@#%^*_=-+?的组合。请您输入高强度密码并定期修改，以提高安全性，防止出现

参数	描述
	密码被暴力破解等安全风险。 请妥善保管您的密码，因为系统将无法获取您的密码信息。 实例创建成功后，如需重置密码，请参见 <a href="#">重置管理员密码</a> 。
确认密码	必须和管理员密码相同。

表2-5 参数模板

参数	描述
参数模板	数据库参数就像是数据库引擎配置值的容器，参数模板中的参数可应用于一个或多个相同类型的数据库实例。实例创建成功后，参数模板可进行修改。  <b>须知</b> 创建数据库实例时，为确保数据库实例正常创建，自定义参数模板中相关规格参数如下不会下发，而是采用系统默认的推荐值。 “innodb_buffer_pool_size” “innodb_log_buffer_size” “max_connections” “innodb_buffer_pool_instances” “innodb_page_cleaners” “innodb_parallel_read_threads” “innodb_read_io_threads” “innodb_write_io_threads” “threadpool_size” 具体请参见 <a href="#">编辑参数模板</a> 。
企业项目	该参数针对企业用户使用，如需使用该功能，请联系客服申请开通。  企业项目是一种云资源管理方式，企业项目管理服务提供统一的云资源按项目管理，以及项目内的资源管理、成员管理。  请在下拉框中选择所在的企业项目，其中，default 为默认项目。更多企业项目相关的信息，请参见《企业管理用户指南》。

表2-6 标签

参数	描述
标签	可选配置，对关系型数据库的标识。使用标签可以方便识别和管理您拥有的关系型数据库服务资源。每个实例最多支持 10 个标签配额。  实例创建成功后，您可以单击实例名称，在标签页签下查看对应

参数	描述
	标签。关于标签的详细操作，请参见 <a href="#">标签</a> 。

表2-7 购买周期（包年/包月）

参数	描述
购买时长	选择所需的时长，系统会自动计算对应的配置费用，时间越长，折扣越大。

**步骤 6** 对于按需计费的实例，进行规格确认。

- 如果需要重新选择实例规格，单击“上一步”，返回上个页面修改实例信息。
- 如果规格确认无误，单击“提交”，完成创建实例的申请。

跳过[步骤 7](#)，直接执行[步骤 8](#)。

**步骤 7** 对于包年/包月模式的实例，进行订单确认。

- 如果需要重新选择实例规格，单击“上一步”，回到上个页面修改实例信息。
- 如果订单确认无误，单击“去支付”，进入“付款”页面。

对于“包年/包月”模式的实例，选择付费方式，付款成功后，才会创建。

**步骤 8** GaussDB(for MySQL)数据库实例创建成功后，用户可以在“实例管理”页面对其进行查看和管理。

- 创建实例过程中，状态显示为“创建中”。创建完成的实例状态为“正常”，此时，实例才可以正常使用。
- 创建实例时，系统默认开启自动备份策略。实例创建成功后，备份策略不允许关闭，并且系统会自动创建一个全量备份。
- 数据库端口默认为 3306，实例创建成功后可修改。  
具体请参见[修改数据库端口](#)。

----结束

## 2.2 连接实例

### 2.2.1 实例连接方式简介

表2-8 连接方式

连接方式	连接地址	使用场景	说明

连接方式	连接地址	使用场景	说明
内网连接	读写内网地址	系统默认提供内网 IP 地址。当应用部署在弹性云服务器上，且该弹性云服务器与 GaussDB(for MySQL)实例处于同一区域，同一 VPC 时，建议单独使用内网 IP 连接弹性云服务器与 GaussDB(for MySQL)数据库实例。	<ul style="list-style-type: none"><li>• 安全性高，可实现 GaussDB(for MySQL)的较好性能。</li><li>• 推荐使用内网连接。</li></ul>
公网连接	弹性公网 IP	不能通过内网 IP 地址访问 GaussDB(for MySQL)实例时，使用公网访问，建议单独绑定弹性公网 IP 连接弹性云服务器（或公网主机）与 GaussDB(for MySQL)数据库实例。	<ul style="list-style-type: none"><li>• 降低安全性。</li><li>• 为了获得更快的传输速率和更高的安全性，建议您将应用迁移到与您的 GaussDB(for MySQL)实例在同一 VPC 内，使用内网连接。</li></ul>

### 📖 说明

- VPC：虚拟私有云（Virtual Private Cloud，简称 VPC）。
- ECS：弹性云服务器（Elastic Cloud Server，简称 ECS）。
- 若弹性云服务器和 GaussDB(for MySQL)数据库实例处于同一个 VPC 内，则无需申请外网地址。
- 首次使用 GaussDB(for MySQL)，请先了解[使用限制](#)。

## 操作步骤

### 通过内网连接 GaussDB(for MySQL)实例

- 步骤 1 [购买实例](#)。根据业务需求，确认 GaussDB(for MySQL)实例的规格、网络配置、数据库帐户配置信息等。
- 步骤 2 [设置安全组规则](#)。
- ECS 与 GaussDB(for MySQL)实例在相同安全组时，默认 ECS 与 GaussDB(for MySQL)实例互通，无需设置安全组规则，执行[通过内网连接 GaussDB\(for MySQL\)实例](#)。

### 须知

ECS 和 GaussDB(for MySQL)实例必须处于同一 VPC。

- ECS 与 GaussDB(for MySQL)实例在不同安全组时，需要为 GaussDB(for MySQL)和 ECS 分别设置安全组规则。

- 设置 GaussDB(for MySQL)安全组规则：为 GaussDB(for MySQL)所在安全组配置相应的入方向规则。
- 设置 ECS 安全组规则：安全组默认规则为出方向上数据报文全部放行，此时，无需对 ECS 配置安全组规则。当在 ECS 所在安全组为非默认安全组且出方向规则非全放通时，需要为 ECS 所在安全组配置相应的出方向规则。

步骤 3 通过内网连接 GaussDB(for MySQL)实例。提供两种连接方式通过 MySQL 客户端连接实例：普通连接和 SSL 连接。其中，SSL 连接实现了数据加密功能，具有更高的安全性。

----结束

### 通过公网连接 GaussDB(for MySQL)实例

步骤 1 购买实例。根据业务需求，确认 GaussDB(for MySQL)实例的规格、网络配置、数据库帐户配置信息等。

步骤 2 绑定弹性公网 IP。弹性公网 IP 提供独立的公网 IP 资源，包括公网 IP 地址和公网出口带宽服务。可以在虚拟私有云管理控制台申请弹性公网 IP 并将该弹性公网 IP 绑定至 GaussDB(for MySQL) 实例。

步骤 3 设置安全组规则。从安全组外访问安全组内的 GaussDB(for MySQL)实例时，需要为 GaussDB(for MySQL) 实例所在安全组配置相应的入方向规则。

步骤 4 通过公网连接 GaussDB(for MySQL)实例。提供两种连接方式通过 MySQL 客户端连接实例：普通连接和 SSL 连接。其中，SSL 连接实现了数据加密功能，具有更高的安全性。

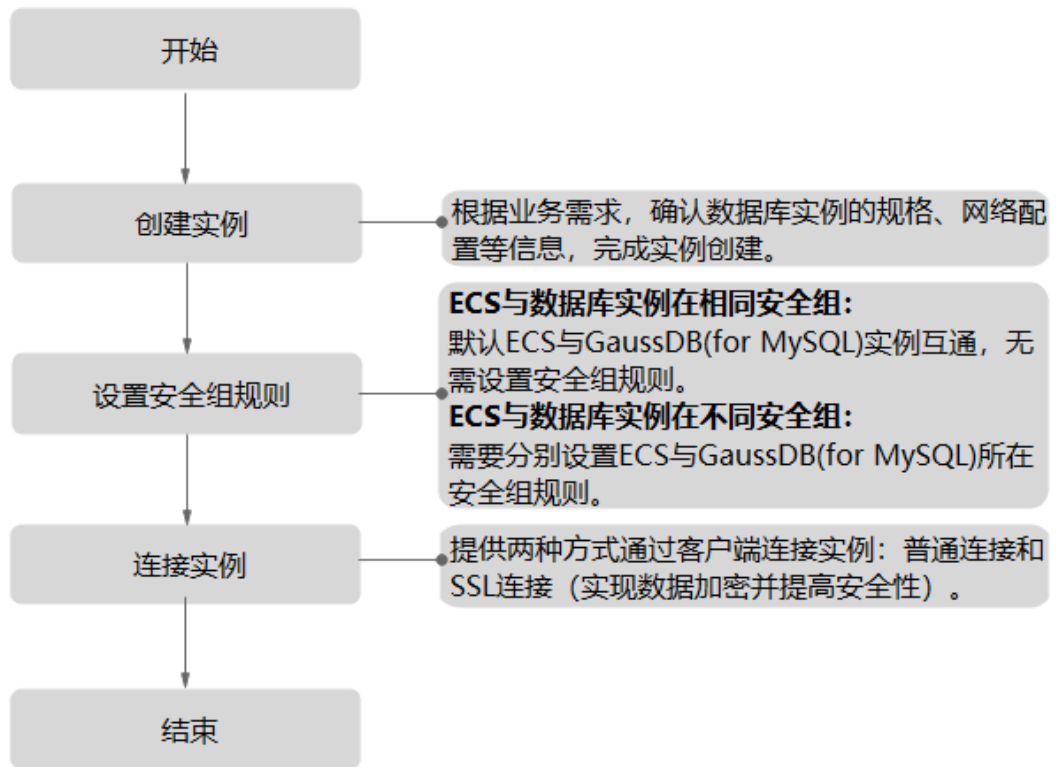
----结束

## 2.2.2 通过内网连接 GaussDB(for MySQL)实例

### 2.2.2.1 内网连接实例流程

通过内网连接 GaussDB(for MySQL)实例的使用流程介绍如图 2-1 所示。

图2-1 通过内网连接实例



### 2.2.2.2 设置安全组规则

#### 操作场景

安全组是一个逻辑上的分组，为同一个虚拟私有云内具有相同安全保护需求，并相互信任的弹性云服务器和 GaussDB(for MySQL)数据库实例提供访问策略。为了保障数据库的安全性和稳定性，在使用 GaussDB(for MySQL)数据库实例之前，您需要设置安全组，开通需访问数据库的 IP 地址和端口。

内网连接 GaussDB(for MySQL)实例时，设置安全组分为以下两种情况：

- ECS 与 GaussDB(for MySQL)实例在相同安全组时，默认 ECS 与 GaussDB(for MySQL)实例互通，无需设置安全组规则，执行[通过内网连接 GaussDB\(for MySQL\)实例](#)。
- ECS 与 GaussDB(for MySQL)实例在不同安全组时，需要为 GaussDB(for MySQL)和 ECS 分别设置安全组规则。
  - 设置 GaussDB(for MySQL)安全组规则：为 GaussDB(for MySQL)所在安全组配置相应的入方向规则。
  - 设置 ECS 安全组规则：安全组默认规则为出方向上数据报文全部放行，此时，无需对 ECS 配置安全组规则。当在 ECS 所在安全组为非默认安全组且出方向规则非全放通时，需要为 ECS 所在安全组配置相应的出方向规则。



## 注意事项

因为安全组的默认规则是在出方向上的数据报文全部放行，同一个安全组内的弹性云服务器和 GaussDB(for MySQL)数据库实例可互相访问。安全组创建后，您可以在安全组中定义各种访问规则，当 GaussDB(for MySQL)数据库实例加入该安全组后，即受到这些访问规则的保护。

- 默认情况下，一个租户可以创建 500 条安全组规则。
- 为一个安全组设置过多的安全组规则会增加首包延时，因此，建议一个安全组内的安全组规则不超过 50 条。
- 当需要从安全组外访问安全组内的 GaussDB(for MySQL)数据库实例时，需要为安全组添加相应的入方向规则。

### 说明

为了保证数据及实例安全，请合理使用权限。建议使用最小权限访问，并及时修改数据库默认端口号（3306），同时将可访问 IP 地址设置为远程主机地址或远程主机所在的最小子网地址，限制远程主机的访问范围。

源地址默认的 IP 地址 0.0.0.0/0 是指允许所有 IP 地址访问安全组内的 GaussDB(for MySQL)数据库实例。

## 操作步骤

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面，选择目标实例，单击实例名称，进入实例的“基本信息”页面。

步骤 5 设置安全组规则。

在“网络信息”模块的“内网安全组”处，单击安全组名称，进入安全组页面。

步骤 6 在“入方向规则”子页签下单击“添加规则”，在“添加入方向规则”弹出框中填写安全组信息，单击“确定”。

单击  可以依次增加多条入方向规则。

表2-9 入方向参数说明

参数	说明	取值样例
协议端口	安全组规则作用的协议。 <ul style="list-style-type: none"> <li>• 目前支持“全部方通”、“全部 TCP”“自定义 TCP”、“全部 UDP”、“自定义 UDP”“ICMP”和“GRE”等协议。</li> <li>• 选择“全部放通”，表示全部</li> </ul>	自定义 TCP

参数	说明	取值样例
	协议端口。	
	端口：允许远端地址访问弹性云服务器指定端口。	<b>通过内网连接实例时</b> ，输入已购买弹性云服务器的目标实例的端口。 <ul style="list-style-type: none"> <li>• 单个端口：例如 22</li> <li>• 连续端口：例如 22-30</li> <li>• 全部端口：为空或 1-65535</li> </ul>
源地址	源地址：可以是 IP 地址、安全组。 xxx.xxx.xxx.xxx/32（IPv4 地址） xxx.xxx.xxx.0/24（子网） 0.0.0.0/0（任意地址）	0.0.0.0/0
描述	安全组规则的描述信息，非必填项。 描述信息内容不能超过 255 个字符，且不能包含“<”和“>”。	-
操作	支持复制或删除一条安全组规则。只有一条安全组规则时不能删除。	-

----结束

### 2.2.2.3 通过内网连接 GaussDB(for MySQL)实例

GaussDB(for MySQL)实例提供 Linux 操作系统和 Windows 操作系统连接实例的方法。

- Linux 操作系统下使用 MySQL 客户端连接实例，并且提供 [SSL 连接（推荐）](#) 和 [非 SSL 连接](#) 两种连接方式。其中，SSL 连接实现了数据加密功能，具有更高的安全性。
- Windows 操作系统下使用 [MySQL-Front 客户端连接实例](#)。

#### 前提条件

1. 创建并登录弹性云服务器。
  - 创建并登录弹性云服务器，请参见《弹性云服务器用户指南》中“创建弹性云服务器”和“登录弹性云服务器”。
  - 通过弹性云服务器连接 GaussDB(for MySQL)数据库实例，需要具备以下条件。
    - i. 该弹性云服务器与目标实例必须处于同一 VPC 内。
    - ii. 该弹性云服务器必须处于目标实例所属安全组允许访问的范围内。

- 1) 如果目标实例和 ECS 所属安全组为**默认安全组**，则无需设置安全组规则。
- 2) 如果目标实例和 ECS 所属安全组为**非默认安全组**，请查看安全组规则是否允许该弹性云服务器访问。具体操作请参考[设置安全组规则](#)。

如果安全组规则允许弹性云服务器访问，即可连接实例。

如果安全组规则不允许弹性云服务器访问，则需添加安全组规则。该弹性云服务器必须处于目标实例所属安全组允许访问的范围内。

## 2. 在准备的弹性云服务器上安装客户端。

- Linux 操作系统中，您需要在弹性云服务器上安装 MySQL 客户端。建议您下载的 MySQL 客户端版本高于已创建的 GaussDB(for MySQL)实例中数据库版本。


如何获取相应安装包及完成安装，请参见[如何安装 MySQL 客户端](#)。

- Windows 操作系统中，您可以使用任何通用的数据库客户端连接到 GaussDB(for MySQL)实例且连接方法类似。

本章节以 MySQL-Front 为例，描述 Windows 操作系统连接实例具体操作步骤，详情请参见[使用 MySQL-Front 连接实例](#)。

## SSL 连接（推荐）


步骤 1 登录管理控制台。


步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面，单击实例名称进入“基本信息”页面。

步骤 5 在“实例信息”模块的“SSL”处，查看 SSL 开关状态。

- 开关打开，执行[步骤 6](#)。
- 开关关闭，单击 ，在弹出框中，单击“是”，开启 SSL 加密。然后执行[步骤 6](#)。

步骤 6 单击“SSL”处的 ，下载根证书或捆绑包。

步骤 7 登录 Linux 系统的弹性云服务器。

步骤 8 在弹性云服务器上安装 MySQL 客户端。详情请参见[如何安装 MySQL 客户端](#)。

步骤 9 将根证书导入弹性云服务器。

请参见[将根证书导入 Windows/Linux 操作系统](#)。

步骤 10 执行如下命令，连接 GaussDB(for MySQL)数据库实例。

```
mysql -h <hostIP> -P <port> -u <userName> -p --ssl-ca=<cafile>
```

表2-10 参数说明

参数	说明
<hostIP>	读写内网地址。 在目标实例的“基本信息”页面，“网络信息”模块查看“读写内网地址”信息。
<port>	数据库端口，默认 3306。 在目标实例的“基本信息”页面，“网络信息”模块的“数据库端口”。
<userName>	用户名，即 GaussDB(for MySQL)数据库管理员帐号，默认为 root。
<cafile>	相应的 SSL 证书文件，该文件需放在执行该命令的路径下。

示例：

使用 root 用户通过 SSL 的方式连接数据库实例，示例如下：

```
mysql -h 172.xx.xx.xx -P 3306 -u root -p --ssl-ca=ca.pem
```


出现如下提示时，输入数据库帐号对应的密码：


```
Enter password:
```

----结束


## 非 SSL 连接

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面，单击实例名称进入“基本信息”页面，

步骤 5 在“实例信息”模块的“SSL”处，关闭 SSL 开关。确保 SSL 处于关闭状态 。

步骤 6 登录 Linux 系统的弹性云服务器。

步骤 7 在弹性云服务器上安装 MySQL 客户端。详情请参见[如何安装 MySQL 客户端](#)。

步骤 8 执行如下命令，连接 GaussDB(for MySQL)数据库实例。

```
mysql -h <hostIP> -P <port> -u <userName> -p
```

表2-11 参数说明

参数	说明
----	----

参数	说明
<hostIP>	读写内网地址。 在目标实例的“基本信息”页面，“网络信息”模块查看“读写内网地址”信息。
<port>	数据库端口，默认 3306。 在目标实例的“基本信息”页面，“网络信息”模块的“数据库端口”。
<userName>	用户名，即 GaussDB(for MySQL)数据库管理员帐号，默认为 root。

示例：

使用 root 用户连接数据库实例，示例如下：

```
mysql -h 172.xx.xx.xx -P 3306 -u root -p
```

出现如下提示时，输入数据库帐号对应的密码：

```
Enter password:
```

#### 📖 说明

若连接失败，请确保各项[前提条件](#)正确配置后，重新尝试连接。

----结束

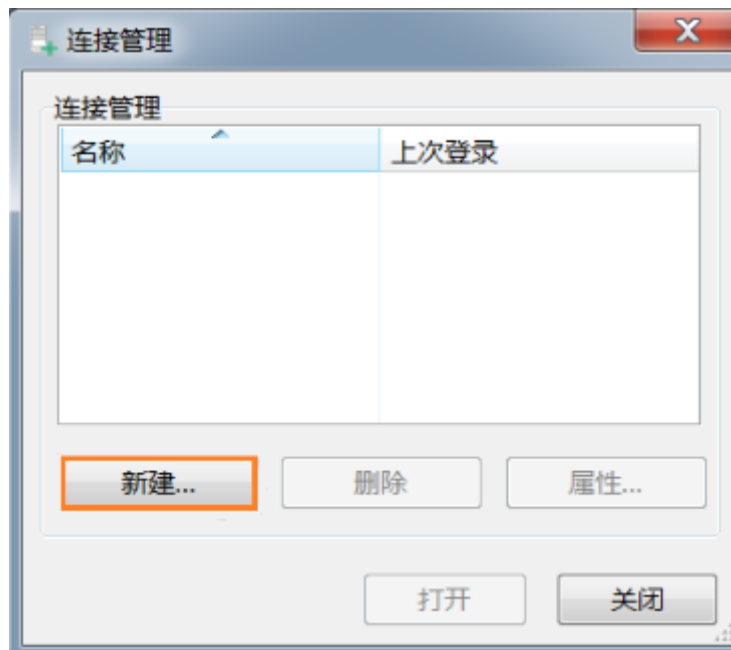
## MySQL-Front 连接

步骤 1 在弹性云服务器上安装 MySQL-Front 客户端。

步骤 2 启动 MySQL-Front 客户端。

步骤 3 在连接管理对话框中，单击“新建”。

图2-2 连接管理



步骤 4 输入需要连接的 GaussDB(for MySQL)数据库实例信息，然后单击“确定”。

图2-3 添加信息

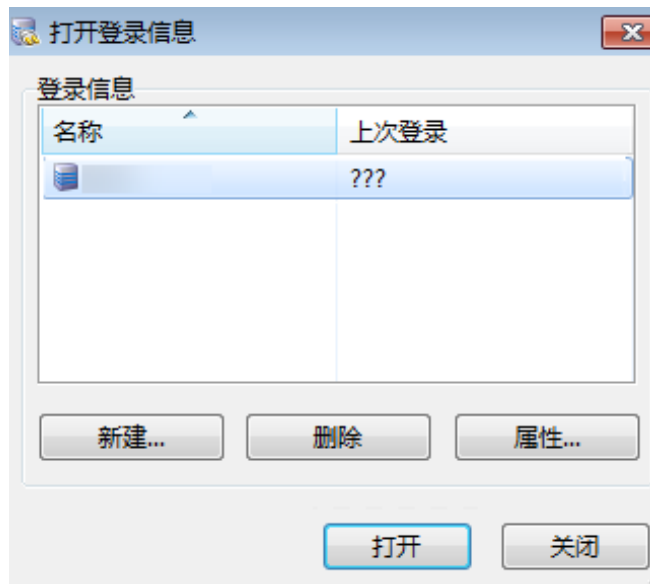


表2-12 参数说明

参数	说明
名称	连接数据库的任务名称。若不填写，系统默认与 Host 一致。
主机	目标实例的读写内网地址。查看目标实例的连接地址信息的步骤如下： 1. 登录 GaussDB(for MySQL)管理控制台。 2. 选择目标实例所在区域。 3. 单击目标实例名称，进入“基本信息”页面。 4. 在“网络信息”模块，可查看“读写内网地址”信息。
端口	输入 GaussDB(for MySQL)实例的内网端口。
用户	需要访问 GaussDB(for MySQL)实例的帐号名称，默认 root。
密码	要访问 GaussDB(for MySQL)数据库实例的帐号所对应的密码。

步骤 5 在打开登录信息窗口，选中图 2-4 创建的连接，单击“打开”，如下图所示。若连接信息无误，即会成功连接实例。

图2-4 打开登录信息



#### 说明

若连接失败，请确保各项前提条件正确配置后，重新尝试连接。

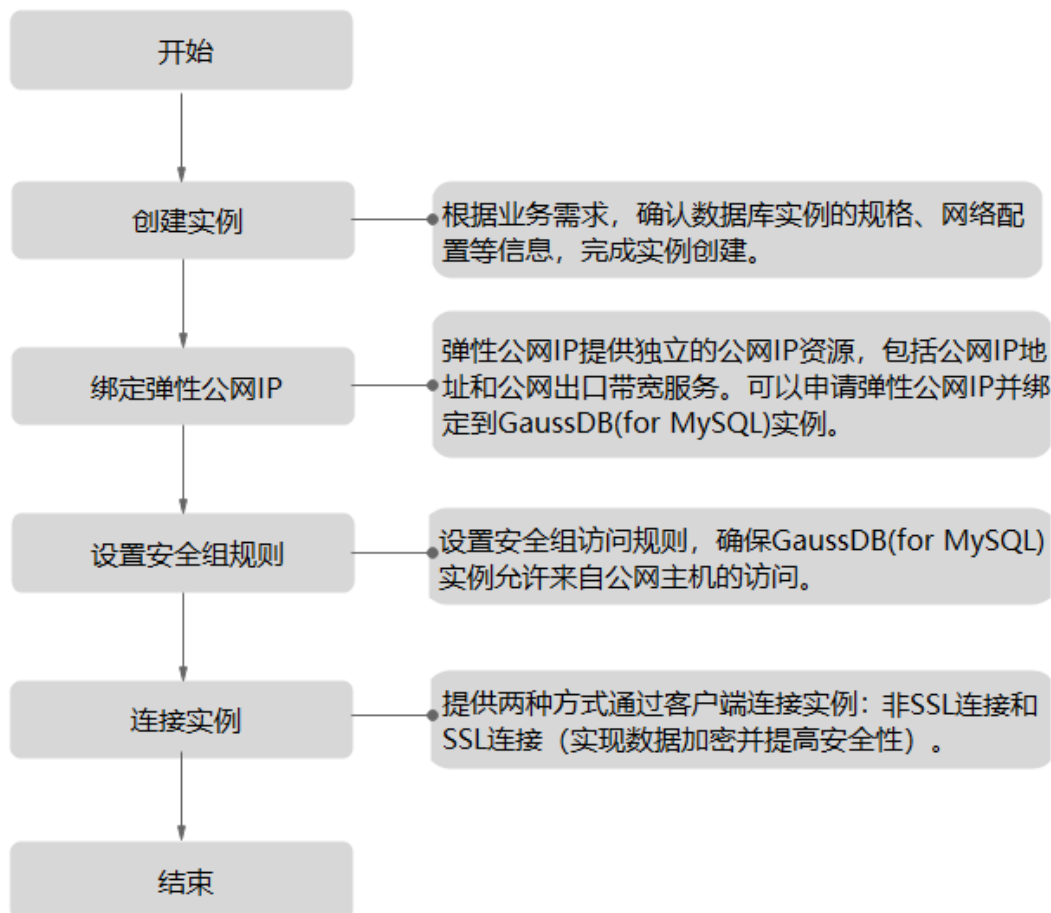
----结束

## 2.2.3 通过公网连接 GaussDB(for MySQL)实例

### 2.2.3.1 公网连接实例流程

通过公网连接 MySQL 实例的使用流程介绍如图 2-5 所示。

图2-5 通过公网连接实例



### 2.2.3.2 绑定弹性公网 IP

#### 操作场景


GaussDB(for MySQL)实例创建成功后，支持用户绑定弹性公网 IP，通过公共网络来访问数据库实例，绑定后也可根据需要解绑。

#### 绑定弹性公网 IP

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。



- 步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。
- 步骤 4 在“实例管理”页面，选择目标实例，单击实例名称，进入实例的“基本信息”页面。
- 步骤 5 在“网络信息”模块，单击“读写公网地址”后面的“绑定”。
- 步骤 6 在弹出框的弹性公网 IP 地址列表中，选择目标弹性公网 IP，单击“确定”，提交绑定任务。

如果没有可用的弹性公网 IP，您可单击“查看弹性公网 IP”，跳转到网络控制台创建弹性公网 IP，创建完成后请返回实例的基本信息页面绑定弹性公网 IP。

#### 须知

您需要设置安全组，开通需访问数据库的 IP 地址和端口，才可以访问数据库实例。请参见[设置安全组规则](#)。

- 步骤 7 在“网络信息”模块“读写公网地址”处查看绑定成功的弹性公网 IP。

----结束

### 2.2.3.3 设置安全组规则

#### 操作场景

安全组是一个逻辑上的分组，为同一个虚拟私有云内具有相同安全保护需求，并相互信任的弹性云服务器和 GaussDB(for MySQL)实例提供访问策略。

为了保障数据库的安全性和稳定性，在使用 GaussDB(for MySQL)实例之前，您需要设置安全组，开通需访问数据库的 IP 地址和端口。

通过弹性公网 IP 连接 GaussDB(for MySQL)实例时，需要为 GaussDB(for MySQL)所在安全组配置相应的入方向规则。

#### 注意事项

因为安全组的默认规则是在出方向上的数据报文全部放行，同一个安全组内的弹性云服务器和 GaussDB(for MySQL)数据库实例可互相访问。安全组创建后，您可以在安全组中定义各种访问规则，当 GaussDB(for MySQL)数据库实例加入该安全组后，即受到这些访问规则的保护。

- 默认情况下，一个租户可以创建 500 条安全组规则。
- 为一个安全组设置过多的安全组规则会增加首包延时，因此，建议一个安全组内的安全组规则不超过 50 条。
- 当需要从安全组外访问安全组内的 GaussDB(for MySQL)数据库实例时，需要为安全组添加相应的入方向规则。


### 说明


为了保证数据及实例安全，请合理使用权限。建议使用最小权限访问，并及时修改数据库默认端口号（3306），同时将可访问 IP 地址设置为远程主机地址或远程主机所在的最小子网地址，限制远程主机的访问范围。

源地址默认的 IP 地址 0.0.0.0/0 是指允许所有 IP 地址访问安全组内的 GaussDB(for MySQL)数据库实例。

## 操作步骤

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面，选择目标实例，单击实例名称，进入实例的“基本信息”页面。

步骤 5 设置安全组规则。

在“网络信息”模块的“内网安全组”处，单击安全组名称，进入安全组页面。

步骤 6 在“入方向规则”子页签下单击“添加规则”，在“添加入方向规则”弹出框中填写安全组信息，单击“确定”。

单击“+”可以依次增加多条入方向规则。

表2-13 入方向参数说明

参数	说明	取值样例
协议端口	安全组规则作用的协议。 <ul style="list-style-type: none"> <li>目前支持“全部方通”、“全部 TCP”“自定义 TCP”、“全部 UDP”、“自定义 UDP”“ICMP”和“GRE”等协议。</li> <li>选择“全部放通”，表示全部协议端口。</li> </ul>	自定义 TCP
	端口：允许远端地址访问弹性云服务器指定端口。	<b>通过内网连接实例时</b> ，输入已购买弹性云服务器的目标实例的端口。 <ul style="list-style-type: none"> <li>单个端口：例如 22</li> <li>连续端口：例如 22-30</li> <li>全部端口：为空或 1-65535</li> </ul>
地址	源地址：可以是 IP 地址、安全组。 xxx.xxx.xxx.xxx/32（IPv4 地址）	0.0.0.0/0

参数	说明	取值样例
	xxx.xxx.xxx.0/24（子网） 0.0.0.0/0（任意地址）	
描述	安全组规则的描述信息，非必填项。 描述信息内容不能超过 255 个字符，且不能包含“<”和“>”。	-
操作	支持复制或删除一条安全组规则。只有一条安全组规则时不能删除。	-

----结束

### 2.2.3.4 通过公网连接 GaussDB(for MySQL)实例

GaussDB(for MySQL)实例提供 Linux 操作系统和 Windows 操作系统连接实例的方法。

- Linux 操作系统下使用 MySQL 客户端连接实例，并且提供 [SSL 连接（推荐）](#) 和 [非 SSL 连接](#) 两种连接方式。其中，SSL 连接实现了数据加密功能，具有更高的安全性。
- Windows 操作系统下使用 [MySQL-Front 客户端](#) 连接实例。

### 操作场景

通过公网连接 GaussDB(for MySQL)实例的场景如下：

- VPC 中的 GaussDB(for MySQL)实例被不同 VPC 中的弹性云服务器访问。
- VPC 中的 GaussDB(for MySQL)实例被公共网络中的服务器访问。

### 前提条件


1. 创建并登录弹性云服务器。
2. 绑定弹性公网 IP 并设置安全组规则。
  - a. 对目标实例绑定弹性公网 IP。  
关于如何绑定弹性公网 IP，请参见[绑定弹性公网 IP 弹性 IP](#)。
  - b. 获取本地设备的 IP 地址。
  - c. 设置安全组规则。  
将 2.b 中获取的 IP 地址及目标实例的端口加入安全组允许访问的范围中。  
关于如何设置安全组规则，请参见[设置安全组规则](#)章节。
  - d. 使用 ping 命令连通 2.a 中绑定的弹性公网 IP 地址，确保本地设备可以访问该弹性公网 IP 地址。
3. 在准备的弹性云服务器上安装客户端。

- Linux 操作系统中，您需要在弹性云服务器上安装 MySQL 客户端。建议您下载的 MySQL 客户端版本高于已创建的 GaussDB(for MySQL)实例中数据库版本。  
如何获取相应安装包及完成安装，请参见[如何安装 MySQL 客户端](#)。
- Windows 操作系统中，您可以使用任何通用的数据库客户端连接到 GaussDB(for MySQL)实例且连接方法类似。  
本章节以 MySQL-Front 为例，描述 Windows 操作系统连接实例具体操作步骤，详情请参见[使用 MySQL-Front 连接实例](#)。

## SSL 连接（推荐）


步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面，单击实例名称进入“基本信息”页面，

步骤 5 在“实例信息”模块的“SSL”处，查看 SSL 开关状态。

- 开关打开，执行[步骤 6](#)。
- 开关关闭，单击 ，在弹出框中，单击“是”，开启 SSL 加密。然后执行[步骤 6](#)。

步骤 6 单击“SSL”处的 ，下载根证书或捆绑包。

步骤 7 登录 Linux 系统的弹性云服务器。

步骤 8 在弹性云服务器上安装 MySQL 客户端，详情请参见[如何安装 MySQL 客户端](#)。

步骤 9 将根证书上传至需连接 GaussDB(for MySQL)实例的弹性云服务器，并导入根证书。

请参见[将根证书导入 Windows/Linux 操作系统](#)。

步骤 10 执行如下命令，连接 GaussDB(for MySQL)数据库实例。

```
mysql -h <hostIP> -P <port> -u <userName> -p --ssl-ca=<cafile>
```

表2-14 参数说明

参数	说明
<hostIP>	主机 IP，在目标实例的“基本信息”页面，“网络信息”模块的“读写公网地址”。
<port>	数据库端口，在目标实例的“基本信息”页面，“网络信息”模块的“数据库端口”。
<userName>	用户名，即 GaussDB(for MySQL)数据库管理员帐号，默认为 root。

参数	说明
<cafile>	相应的 SSL 证书文件，该文件需放在执行该命令的路径下。

示例：

使用 root 用户 SSL 连接数据库实例，示例如下：

**mysql -h 10.xx.xx.xx -P 3306 -u root -p --ssl-ca=ca.pem**

出现如下提示时，输入数据库帐号对应的密码：

```
Enter password:
```


#### 说明

若连接失败，请确保各项前提条件正确配置后，重新尝试连接。

----结束

## 非 SSL 连接

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面，单击实例名称进入“基本信息”页面，

步骤 5 在“实例信息”模块的“SSL”处，关闭 SSL 开关。确保 SSL 处于关闭状态 。

步骤 6 登录 Linux 系统的弹性云服务器。

步骤 7 在弹性云服务器上安装 MySQL 客户端，详情请参见[如何安装 MySQL 客户端](#)。

步骤 8 连接 GaussDB(for MySQL)数据库实例，执行如下命令。

**mysql -h <hostIP> -P <port> -u <userName> -p**

表2-15 参数说明

参数	说明
<hostIP>	主机 IP，在目标实例的“基本信息”页面，“网络信息”模块的“读写公网地址”。
<port>	数据库端口，在目标实例的“基本信息”页面，“网络信息”模块的“数据库端口”。
<userName>	用户名，即 GaussDB(for MySQL)数据库管理员帐号，默认为 root。

示例：

使用 root 用户连接数据库实例。

```
mysql-h 10.xx.xx.xx -P 3306 -u root -p
```

出现如下提示时，输入数据库帐号对应的密码：

```
Enter password:
```

#### 📖 说明

若连接失败，请确保各项前提条件正确配置后，重新尝试连接。

----结束

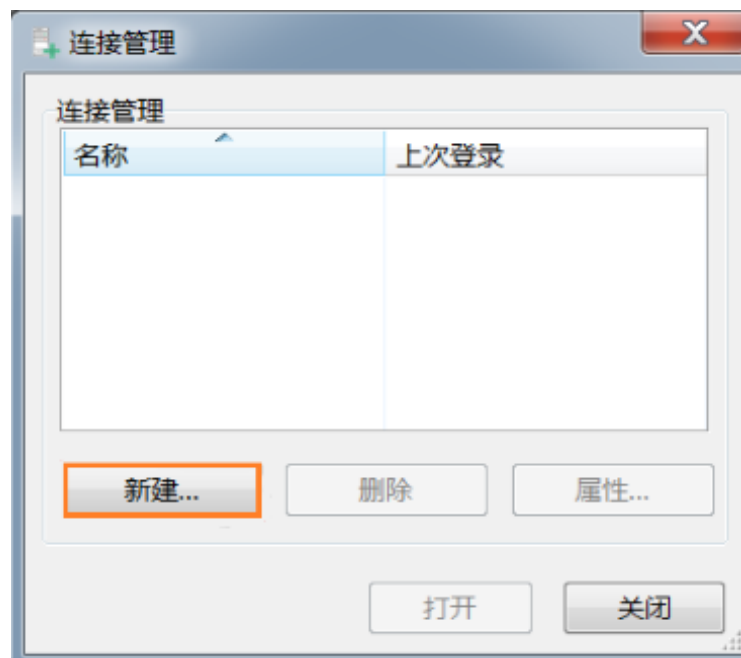
## MySQL-Front 连接

步骤 1 在弹性云服务器上安装 MySQL-Front 客户端。

步骤 2 启动 MySQL-Front 客户端。

步骤 3 在连接管理对话框中，单击“新建”。

图2-6 连接管理



步骤 4 输入需要连接的 GaussDB(for MySQL)数据库实例信息，然后单击“确定”。

图2-7 添加信息



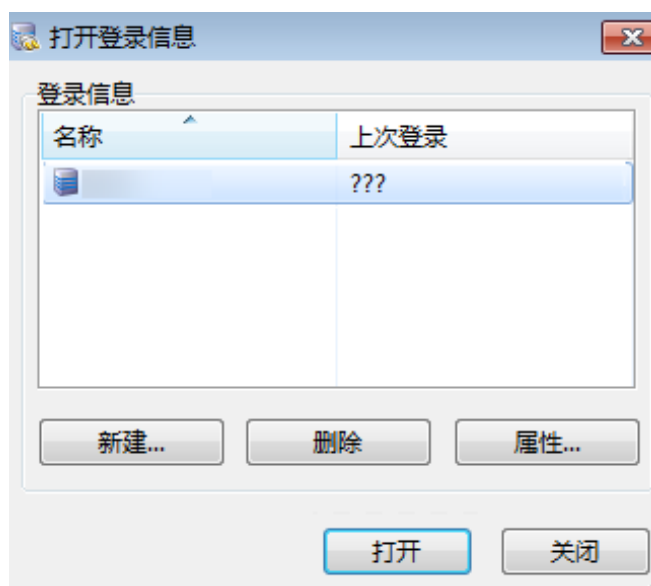
表2-16 参数说明

参数	说明
名称	连接数据库的任务名称。若不填写，系统默认与 Host 一致。
主机	目标实例的弹性公网 IP，在目标实例的“基本信息”页面，“网络信息”模块的“读写公网地址”。
端口	输入 GaussDB(for MySQL)实例的内网端口。
用户	需要访问 GaussDB(for MySQL)实例的帐号名称。默认 root。
密码	要访问 GaussDB(for MySQL)数据库实例的帐号所对应的密码。

**步骤 5** 在打开登录信息窗口，选中创建的连接，单击“打开”，如下图所示。

若连接信息无误，即会成功连接实例。

图2-8 打开登录信息



#### 说明

若连接失败，请确保各项前提条件正确配置后，重新尝试连接。

----结束



# 3 用户指南

## 3.1 计费管理

### 3.1.1 续费

#### 操作场景


您可根据业务需要，对单个“包年/包月”实例进行续费，以保证业务不因欠费而受影响。

#### 约束条件

- 仅 GaussDB(for MySQL) “包年/包月” 的实例支持续费。
- 运行状态为“正常”或“异常”的实例才可进行续费。

#### 操作步骤

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面，选择目标实例，单击“操作”列的“续费”。

您也可以单击目标实例名称，进入“基本信息”页面，在图 3-1 “计费信息”模块的“计费模式”处，单击“续费”。

图3-1 续费

计费信息	
计费模式	包年/包月 <b>续费</b>
订单	
创建时间	2020/02/17 21:41:28 GMT+08:00
到期时间	2020/03/17 23:59:59 GMT+08:00

步骤 5 进入续费页面，对实例进行续费。

----结束

## 3.1.2 包周期实例转按需

### 操作场景

云数据库 GaussDB(for MySQL)服务支持单个包周期（包年/包月）实例转为按需实例，方便用户灵活使用该资费的实例。

#### 须知

实例的按需计费方式需要等包周期到期后才会生效，且自动续费功能会同步失效。

### 单个包周期实例转按需

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。


步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面，选择目标包周期实例，单击“操作”列的“更多 > 转按需”，进入“包周期转按需”页面。

您也可以单击目标实例名称，进入实例的“基本信息”页面，在“计费信息”模块的“计费模式”处，单击“转按需”，进入“包周期转按需”页面。

步骤 5 进入计费模式变更页面，对实例进行计费模式的变更。

步骤 6 包周期转按需创建成功后，用户可以在“实例管理”页面对其进行查看和管理。

在实例列表的右上角，单击刷新列表，可查看到包周期转按需完成后，实例状态显示为“正常”。“计费模式”显示为“按需”。

----结束

### 3.1.3 按需实例转包周期

#### 操作场景


GaussDB(for MySQL)支持单个按需实例转为包周期（包年/包月）实例。由于按需资源费用较高，需要长期使用资源的按需用户可以选择将按需资源转为包周期，继续使用这些资源的同时，享受包周期的优惠资费。执行转包周期后可在订单状态中查询操作是否生效。


#### 约束条件

- 不支持按需实例中的某个节点（主节点和只读节点）单独转包周期。
- 运行状态为冻结、创建失败、规格变更中、扩容中、创建只读中的实例不支持按需实例转包周期。

#### 单个按需实例转包周期

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的, 选择区域和项目。

步骤 3 在页面左上角单击, 选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面，选择目标实例，在操作列下的“更多”中选择“转包周期”，选择存储空间，单击“下一步”进入“按需转包周期”页面。

您也可以单击目标实例名称，进入“基本信息”页面，单击“转包周期”，选择存储空间，单击“下一步”进入“按需转包周期”页面。


您申请的存储空间会有必要的系统开销，这些开销包括索引节点和保留块，以及数据运行的必须空间。存储空间的数值必须为 10 的整数倍。

步骤 5 选择续费规格，以月为单位，最小包周期时长为一个月。

- 如果订单确认无误，单击“去支付”，进入“支付”页面。
- 如果暂未付款，系统将保留您的订单，稍后可在“费用 > 我的订单”中支付或取消订单。并且实例状态为“转包周期中，变更订单未支付，去支付”。

步骤 6 选择支付方式，单击“确认付款”。

步骤 7 按需转包周期创建成功后，用户可以在“实例管理”页面对其进行查看和管理。

在实例列表的右上角，单击刷新列表，可查看到按需转包周期完成后，实例状态显示为“正常”。“计费模式”显示为“包年/包月”。

----结束

### 3.1.4 退订包周期实例

#### 操作场景

对于“包年/包月”模式的数据库实例，您需要退订订单，才可删除数据库资源。退订实例订单时，会同步删除该实例中所有资源，包括对应的只读节点。

#### 方法一

您可在“实例管理”页面的实例列表中，退订包周期实例。

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面，选择目标实例，单击“操作”列的“更多 > 退订”，弹出框中单击“是”。

步骤 5 在“退订资源”页面，确认待退订实例实例信息，并选择退订原因，单击“退订”。

步骤 6 在弹出框中确认是否退订该资源，单击“退订”，提交退订申请。

#### 须知

1. 提交退订后，资源和数据将会被删除并无法找回。
2. 如需保留数据，请务必确认完成数据备份后再提交退订。

步骤 7 查看退订结果。数据库实例订单退订成功后，实例将会被删除。

----结束

#### 方法二

您可前往“费用中心”，退订包周期实例。

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 单击页面右上角的“费用中心”，进入费用中心页面。

步骤 5 在左侧导航栏中选择“订单管理 > 退订与退换货”，进入“退订管理”页面。

步骤 6 在“退订管理”页面，勾选需要退订的实例订单，单击“操作”列的“退订资源”。

- 您可通过产品类型筛选出帐号下所有的 GaussDB(for MySQL)订单。
- 您还可以在订单列表左上角，通过实例名称、订单号或实例 ID 搜索资源。

图3-2 搜索资源



步骤 7 在“退订使用中的资源”页面，确认待退订实例信息，并选择退订原因，单击“退订”。

资源退订相关信息，请参考。

步骤 8 在弹出框中确认是否退订该资源，单击“退订”，提交退订申请。

#### 须知

1. 提交退订后，资源和数据将会被删除并无法找回。
2. 请务必确认完成数据备份后再提交退订。

步骤 9 查看退订结果。数据库实例订单退订成功后，实例将会被删除。

----结束

## 3.2 数据迁移

### 3.2.1 使用 mysqldump 迁移 GaussDB(for MySQL)数据

#### 须知

对于 mysqlpump 工具，由于在并行备份场景有 coredump 问题，不建议您使用，推荐您使用 mysqldump 工具迁移。

#### 迁移准备

GaussDB(for MySQL)支持开启公网访问功能，通过弹性公网 IP 进行访问。您也可通过弹性云服务器的内网访问 GaussDB(for MySQL)。

1. 准备弹性云服务器或可通过公网访问 GaussDB(for MySQL)。

- 通过弹性云服务器连接 GaussDB(for MySQL)数据库实例，需要创建一台弹性云服务器。  
创建并登录弹性云服务器，请参见《弹性云服务器用户指南》中“创建弹性云服务器”和“登录弹性云服务器”。
  - 通过公网地址连接 GaussDB(for MySQL)数据库实例，需具备以下条件。
    - i. 先对 GaussDB(for MySQL)数据库实例绑定公网地址。
    - ii. 保证本地设备可以访问 GaussDB(for MySQL)数据库实例绑定的公网地址。
2. 在准备的弹性云服务器或可访问 GaussDB(for MySQL)数据库的设备上，安装 MySQL 客户端。  
如何获取相应安装包及完成安装，请参见[如何安装 MySQL 客户端](#)。

#### 说明

该弹性云服务器或可访问 GaussDB(for MySQL)数据库的设备需要安装和 GaussDB(for MySQL)数据库服务端相同版本或更高版本的数据库客户端，MySQL 数据库或客户端会自带 mysqldump 和 mysql 工具。

## 导出数据

要将源数据库迁移到 GaussDB(for MySQL)数据库，需要先对其进行导出。

#### 须知

- 相应导出工具需要与数据库引擎版本匹配。
- 数据库迁移为离线迁移，您需要停止使用源数据库的应用程序。

**步骤 1** 登录已准备的弹性云服务器，或可访问 GaussDB(for MySQL)数据库的设备。

**步骤 2** 使用 mysqldump 将元数据导出至 SQL 文件。

#### 须知

MySQL 数据库是 GaussDB(for MySQL)数据库服务管理所必须的数据库，导出元数据时，禁止指定 `--all-database` 参数，否则会造成数据库故障。

```
mysqldump --databases<DB_NAME> --single-transaction --order-by-primary --hex-lob
--no-data --routines --events --set-gtid-purged=OFF -u <DB_USER> -p -h
<DB_ADDRESS> -P<DB_PORT>|sed -e 's/DEFINER[ ]*=[ ]*[^\]*\*/' -e
's/DEFINER[ ]*=.*/FUNCTION/FUNCTION/' -e
's/DEFINER[ ]*=.*/PROCEDURE/PROCEDURE/' -e
's/DEFINER[ ]*=.*/TRIGGER/TRIGGER/' -e 's/DEFINER[ ]*=.*/EVENT/EVENT/' >
<BACKUP_FILE>
```

- DB\_NAME 为要迁移的数据库名称。
- DB\_USER 为数据库用户。
- DB\_ADDRESS 为数据库地址。

- DB\_PORT 为数据库端口。
- BACKUP\_FILE 为导出生成的文件名称。

根据命令提示输入数据库密码。

示例如下：

```
mysqldump --databases gaussdb --single-transaction --order-by-primary --hex-blob --no-data --routines --events --set-gtid-purged=OFF -u root -p -h 192.xx.xx.xx -P 3306 |sed -e 's/DEFINER[ ]*=[ ]*[^\]*\^*/' -e 's/DEFINER[ ]*=.*/FUNCTION/FUNCTION/' -e 's/DEFINER[ ]*=.*/PROCEDURE/PROCEDURE/' -e 's/DEFINER[ ]*=.*/TRIGGER/TRIGGER/' -e 's/DEFINER[ ]*=.*/EVENT/EVENT/' > dump-defs.sql
```

**Enter password:**

命令执行完会生成“dump-defs.sql”文件。

步骤 3 使用 mysqldump 将数据导出至 SQL 文件。

#### 须知

MySQL 数据库是 GaussDB(for MySQL)数据库服务管理所必须的数据库，导出元数据时，禁止指定 `--all-database` 参数，否则会造成数据库故障。

```
mysqldump --databases <DB_NAME> --single-transaction --hex-blob --set-gtid-purged=OFF --no-create-info --skip-triggers -u <DB_USER> -p -h <DB_ADDRESS> -P <DB_PORT> -r <BACKUP_FILE>
```

以上命令的参数说明如步骤 2 所示。

根据命令提示输入数据库密码。

示例如下：

```
mysqldump --databases gaussdb --single-transaction --hex-blob --set-gtid-purged=OFF --no-create-info --skip-triggers -u root -p -h 192.xx.xx.xx -P 3306 -r dump-data.sql
```

命令执行完会生成“dump-data.sql”文件。

----结束

## 导入数据

通过弹性云服务器或可访问 GaussDB(for MySQL)数据库的设备，用相应客户端连接 GaussDB(for MySQL)数据库实例，将导出的 SQL 文件导入到 GaussDB(for MySQL)数据库。

#### 须知

如果源数据库中包含触发器、存储过程、函数或事件调用，则需确保导入前设置目标数据库参数 `log_bin_trust_function_creators=ON`

步骤 1 导入元数据到 GaussDB(for MySQL)数据库。

```
mysql -f -h <DB_ADDRESS> -P <DB_PORT> -u root -p < <BACKUP_DIR>/dump-  
defs.sql
```

- DB\_ADDRESS 为 GaussDB(for MySQL)数据库实例的 IP 地址。
- DB\_PORT 为当前数据库实例的端口。
- BACKUP\_DIR 为 “dump-defs.sql” 所在目录。

示例如下：

```
mysql -f -h 172.xx.xx.xx -P 3306 -u root -p < dump-defs.sql
```

**Enter password:**

步骤 2 导入数据到 GaussDB(for MySQL)数据库。

```
mysql -f -h <DB_ADDRESS> -P <DB_PORT> -u root -p < <BACKUP_DIR>/dump-  
data.sql
```

- DB\_ADDRESS 为 GaussDB(for MySQL)数据库实例的 IP 地址。
- DB\_PORT 为当前数据库实例的端口。
- BACKUP\_DIR 为 “dump-data.sql” 所在目录。

示例如下：

```
mysql -f -h 172.xx.xx.198 -P 3306 -u root -p < dump-data.sql
```

**Enter password:**

步骤 3 使用 MySQL 工具连接 GaussDB(for MySQL)数据库实例，查看迁移结果。

```
mysql> show databases;
```

示例中，名为 my\_db 的数据库已经被导入了：

```
mysql> show databases;  
+-----+  
| Database          |  
+-----+  
| information_schema |  
| my_db             |  
| mysql             |  
| performance_schema |  
+-----+  
4 rows in set (0.00 sec)
```

----结束



## 3.3 实例生命周期管理

### 3.3.1 修改实例名称

#### 操作场景

GaussDB(for MySQL)支持修改实例名称，以方便用户识别。

#### 操作步骤

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。


步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面，单击目标实例名称后的 ，编辑实例名称。



- 单击“确认”，提交修改。
- 单击“取消”，取消修改。

#### 说明

实例名称长度为 4~64 个字符，必须以字母开头，可包含大写字母、小写字母、数字、中划线或下划线，不能包含其他特殊字符。

您也可以单击目标实例名称，进入实例的“基本信息”页面，在“实例信息”模块的实例名称处，单击 ，修改实例名称。

有关数据库实例的状态，请参见[数据库实例状态](#)中的相关内容。

- 单击 ，提交修改。
- 单击 ，取消修改。

步骤 5 在当前页面，查看修改结果。修改实例名称所需时间通常在 1 分钟以内。

----结束

### 3.3.2 修改实例备注


#### 操作场景

GaussDB(for MySQL)服务实例名称支持添加备注，以方便用户备注分类。

#### 操作步骤


步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤3 在页面左上角单击，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤4 在“实例管理”页面，单击目标实例备注后的，编辑实例备注。

- 单击“确认”，提交修改。
- 单击“取消”，取消修改。

您也可以单击目标实例名称，进入实例的“基本信息”页面，在“实例信息”模块“实例备注”处，单击，修改实例备注。

- 单击，提交修改。
- 单击，取消修改。

#### 说明

实例备注长度最大 64 个字符，只能包含中文、字母、数字、中划线、下划线或英文句号。

步骤5 在实例的“基本信息”页面，查看修改结果。

您也可以在“实例管理”页面，查看修改结果。

----结束

## 3.3.3 删除实例

### 操作场景

您可根据业务需要，在 GaussDB(for MySQL)数据库“实例管理”页面手动删除来释放资源。

#### 须知

- 执行操作中的实例不能手动删除，只有在实例操作完成后，才可被删除。
- 实例删除后将不再产生费用，自动备份会被同步删除，保留的手动备份会继续收取费用。
- 删除实例时，会同步删除其对应的只读节点，请谨慎操作。
- 当数据库实例被删除时，数据库实例的自动备份将被同步删除，手动备份不会被删除。

### 操作步骤

步骤1 登录管理控制台。

步骤2 单击管理控制台左上角的，选择区域和项目。

步骤3 在页面左上角单击，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面的实例列表中，选择需要删除的实例，在“操作”列，选择“更多 > 删除实例”。

步骤 5 在“删除实例”弹框，单击“是”，稍后刷新“实例管理”页面，查看删除结果。

----结束

## 3.3.4 重启实例

### 操作场景

通常是出于维护目的，您可能需要重启数据库实例。例如：对于某些运行参数修改，需要重启实例使之生效。

### 使用限制


- 如果数据库实例处于“异常”状态，可能会重启失败。
- 为了缩短重启时间，建议您在重启过程中尽可能减少数据库活动，以减少中转事务的回滚活动。

#### 须知

- 重启数据库实例会重新启动数据库引擎服务，将导致短暂中断，在此期间，数据库实例状态将显示为“重启中”。
- 重启过程中，实例将不可用。重启后实例会自动释放内存中 buffer pool 的数据，请注意对业务的热点数据进行预热，避免业务高峰期出现阻塞。

### 操作步骤

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面，选择指定的实例，单击“更多 > 重启实例”。

您也可以在“实例管理”页面，单击目标实例实例名称，在实例的“基本信息”页面右上方，单击“重启实例”。

重启实例时，如果该实例下有只读节点，那么对应的只读节点也会被同步重启。

步骤 5 在“重启实例”弹窗框，选择重启时间，目前支持“立即重启”和“可维护时间段重启”。

1. 选择“立即重启”，直接单击“是”重启实例。
2. 选择“可维护时间段重启”，可以单击“修改”对重启时间段进行修改，然后单击“是”，实例将会在可维护时间段重启实例。

修改可维护时间段，不影响原有可维护时间段内定时任务的执行时间。  
可维护时间段内预约的定时重启任务将于下一个时间窗执行。

----结束

## 3.3.5 重启节点

### 操作场景

您可以通过管理控制台重启数据库节点，用于解决数据库连接问题。

### 使用限制


- “异常”状态的数据库实例节点允许重启。
- 重启数据库实例节点所需的时间，取决于您的特定数据库引擎的崩溃恢复过程。为了缩短重启时间，建议您在重启过程中尽可能减少数据库活动，以减少中转事务的回滚活动。


#### 须知

- 重启数据库实例节点会导致短暂中断，在此期间，数据库实例节点状态将显示为“节点重启中”。
- 重启过程中，该节点将不可用。建议您在业务低谷期执行重启节点操作并确保应用具备重连机制。
- 如果该实例存在参数修改，则需要重启实例新参数才会生效。重启实例后才能重启节点。重启前，请做好业务安排，谨慎操作。

### 操作步骤

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面的实例列表中，选择对应实例，单击实例名称进入“基本信息”页面。

步骤 5 在“基本信息”页面的“节点信息”模块，选择目标节点，在“操作”列单击“重启”

步骤 6 在弹框中选择重启时间，目前支持“立即重启”和“可维护时间段重启”。

1. 选择“立即重启”，直接单击“是”重启节点。
2. 选择“可维护时间段重启”，可以单击“修改”对重启时间段进行修改，然后单击“是”，实例将会在可维护时间段重启节点。

修改可维护时间段，不影响原有可维护时间段内定时任务的执行时间。

可维护时间段内预约的定时重启任务将于下一个时间窗执行。

----结束


### 3.3.6 修改节点名称

#### 操作场景

GaussDB(for MySQL)支持修改节点名称，以方便用户识别不同的业务类型。

#### 操作步骤

步骤 1 登录管理控制台。


步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面的实例列表中，选择对应实例，单击实例名称进入“基本信息”页面。

步骤 5 在“基本信息”页面的“节点信息”模块，选择一个或多个目标节点，单击“修改节点名称”。

- 单击“确定”，提交修改。
- 单击“取消”，取消修改。

您也可以单击节点名称后的 ，编辑节点名称，单击“确认”，即可修改节点名称。

#### 说明

- 节点名称长度在 4 个到 128 个字符之间，必须以字母开头，区分大小写，可以包含字母、数字、中划线或下划线，不能包含其他特殊字符。
- 节点名称不允许重名。

步骤 6 在当前页面，查看修改结果。

----结束

### 3.3.7 导出实例

#### 操作场景


您可以导出实例列表，用于查看并分析实例信息。

#### 导出所有实例

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤3 在页面左上角单击，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤4 在“实例管理”页面，单击实例列表右上角，在导出弹框勾选所需导出信息，单击“导出”。

步骤5 导出任务执行完成后，您可在本地查看到一个“.csv”文件。

----结束

### 3.3.8 回收站

云数据库 GaussDB(for MySQL)支持将删除的实例加入回收站管理。您可以在回收站中重建实例恢复数据。

#### 重建实例

在回收站保留期限内的实例可以通过重建实例恢复数据。

步骤1 登录管理控制台。

步骤2 单击管理控制台左上角的，选择区域和项目。

步骤3 在页面左上角单击，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤4 在“回收站”页面，在实例列表中找到需要恢复的目标实例，单击操作列的“重建”。

步骤5 在“重建新实例”页面，选填配置后，提交重建任务，具体可参考[恢复到新实例操作](#)。

----结束

## 3.4 变更实例

### 3.4.1 变更实例的 CPU 和内存规格

#### 操作场景

可以根据业务需要对包年/包月和按需实例的规格进行变更，规格指实例的 CPU/内存。当实例的状态由“规格变更中”变为“正常”，则说明变更成功。

#### 约束限制


- 当实例进行 CPU/内存规格变更时，该实例不可被删除。
- 您只能对整个实例进行规格变更，无法对实例中的单个节点进行操作。
- 仅按需实例支持选择可维护时间段进行规格变更，且该任务不可被取消。

**须知**

- 变更规格后会主备倒换，请选择业务低峰期，避免业务异常中断。
- 变更规格后主节点与只读节点的读内网地址会发生变化，请及时在应用程序中修改您的连接地址以免影响业务。

**操作步骤**

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 您可以通过两种途径进行规格变更。

- 在“实例管理”页面，选择目标实例，单击“操作”列的“更多 > 规格变更”，进入“规格变更”页面。
- 单击目标实例名称，进入“基本信息”页面，在“实例信息”模块的“性能规格”处，单击“规格变更”，进入“规格变更”页面。

步骤 5 在“规格变更”页面，选择所需修改到的性能规格，您可以根据自己的需求缩小或扩大规格，并设置“切换时间”，单击“下一步”。

切换时间支持如下两种方式：

- 立即变更：系统会立即进行规格变更。
- 可维护时间段：在当前可维护时间段实现数据库资源变更，预约变更时间。

步骤 6 在规格确认页面，确认性能规格。

- 如需重新选择，单击“上一步”，修改性能规格。
- 按需计费模式的实例，单击“提交”，提交变更。  
由规格变更产生的费用，您可在“费用中心 > 费用账单”中查看费用详情。
- 包年/包月模式的实例
  - 缩小规格：单击“提交订单”，提交变更。  
由缩小规格产生的退款，系统会自动退还至客户帐户，您可通过单击“费用中心”进入“费用中心”页面，在左侧导航栏“订单管理”下的“我的订单”查看费用详情。
  - 扩大规格：单击“提交订单”，跳转至支付页面，支付成功后，才可进行规格变更。

步骤 7 查看变更结果。

在实例管理页面，可以看到实例状态为“规格变更中”。稍后在对应的“基本信息”页面，查看实例规格，检查修改是否成功。此过程需要 5~15 分钟。

**须知**

GaussDB(for MySQL) 8.0 实例规格变更成功后，系统将根据新内存大小，调整如下参数的值：“innodb\_buffer\_pool\_size”、“innodb\_log\_buffer\_size”、“max\_connections”、“innodb\_buffer\_pool\_instances” “innodb\_page\_cleaners”、“innodb\_parallel\_read\_threads”、“innodb\_read\_io\_threads”、“innodb\_write\_io\_threads”、“threadpool\_size”。

----结束

## 3.4.2 磁盘容量变更（包年包月）

### 操作场景

包年/包月 GaussDB(for MySQL)实例在使用过程中，随着业务数据的变化，最初申请的存储容量可能会和实际使用量存在偏差，GaussDB(for MySQL)实例提供变更存储空间的服务，以满足您的业务需求。

存储空间如果超过当前容量，超出的部分将按需计费，推荐扩容磁盘享受包周期优惠政策。扩容期间，服务不中断，备份业务不受影响。

### 约束条件

- 仅包年/包月实例支持磁盘容量变更，按需实例不支持此操作。
- GaussDB(for MySQL)数据库实例支持无限次容量变更。
- 容量变更过程中，该实例不可重启和删除。

### 操作步骤

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面，选择目标实例，单击“操作”列的“更多 > 容量变更”，进入“容量变更”页面。

您还可以通过单击目标实例名称，进入“基本信息”页面。在“存储/备份空间”模块，单击“容量变更”，进入“容量变更”页面。



图3-3 容量变更入口



步骤5 在“容量变更”页面，选择空间大小，单击“下一步”。

扩容存储空间，每次扩容步长为 10GB，实例所选容量大小必须为 10 的整数倍，最大磁盘容量为 128000GB。

步骤6 进行规格确认。

- 如需重新选择，单击“上一步”，回到上个页面，修改存储空间大小。
- 核对无误后，单击“提交订单”，进入付款页面，选择支付方式，完成支付。

步骤7 查看容量变更结果。

在实例管理页面，稍后单击实例名称，在“基本信息”页面，查看存储空间大小，检查磁盘容量变更是否成功。

----结束

### 3.4.3 设置可维护时间段

#### 操作场景

默认可维护时间段为 02:00~06:00，您可以根据业务需求，设置可维护时间段。建议将可维护时间段设置在业务低峰期，避免业务在维护过程中异常中断。

#### 操作步骤

步骤1 登录管理控制台。

步骤2 单击管理控制台左上角的 ，选择区域和项目。


- 步骤3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。
- 步骤4 在“实例管理”页面，单击目标实例名称，进入“基本信息”页面，在“实例信息”模块的“可维护时间段”处，单击“修改”。
- 步骤5 在“修改可维护时间段”弹框中，选择一个可维护时间段，单击“确定”。

图3-4 修改可维护时间段



#### 说明

修改可维护时间段，不影响原有可维护时间段内定时任务的执行时间。



----结束


## 3.4.4 自定义列表项

### 操作场景

您可以根据自身业务需要，自定义设置实例信息列表项。

### 操作步骤

- 步骤1 登录管理控制台。
- 步骤2 单击管理控制台左上角的 ，选择区域和项目。
- 步骤3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面，单击，选择自定义列表项。

- 系统默认展示以下列表项：实例名称/ID、实例类型、实例备注、数据库引擎、运行状态、计费模式、读写内网地址、操作。

默认实例信息列表项不可更改。

- 您还可以选择其他列表项，主要包括：创建时间、数据库端口和存储空间类型等。

----结束

## 3.4.5 升级内核小版本

### 操作场景


GaussDB(for MySQL)支持手动升级内核小版本，内核小版本的升级涉及性能提升、功能优化或问题修复等。

### 注意事项

- 升级数据库内核小版本会重启 GaussDB(for MySQL)实例，服务可能会出现闪断，请您尽量在业务低峰期执行该操作，或确保您的应用有自动重连机制。
- 如果实例有大量表分区（100w+），重启实例时间可能会达到 2 小时以上。
- 如果主节点和只读节点在同一个 AZ，升级内核小版本会触发一次主备倒换；如果在不同 AZ，则会触发两次主备倒换。主备倒换指主节点与只读节点进行切换。
- 升级实例小版本时，如有只读节点，也会同步升级只读节点的小版本，升级完成会重启实例，请您选择合适的时间升级（不支持单独升级只读实例的小版本）。升级内核小版本后，实例会升级到最新的内核小版本，升级成功，无法降级。
- 小版本升级过程中禁止 event 的 ddl 操作，如 create event、drop event 和 alter event。

### 操作步骤

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的，选择区域和项目。

步骤 3 在页面左上角单击，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面，选择指定的目标实例，单击实例名称。

步骤 5 在“基本信息”页面，“实例信息”模块的“兼容的数据库版本”处，单击“补丁升级”。

您也可以在“实例管理”页面的“数据库引擎”列看到内核小版本升级提示，单击“补丁升级”。

步骤 6 在弹出框中，选择升级方式，单击“确定”。

----结束

## 3.5 只读节点

### 3.5.1 只读节点简介

#### 产品简介

GaussDB(for MySQL)实例支持只读节点。

在对数据库有少量写请求，但有大量读请求的应用场景下，单个实例可能无法抵抗读取压力，甚至对主业务产生影响。为了实现读取能力的弹性扩展，分担数据库压力，您可以在某个区域中创建一个或多个只读节点，利用只读节点满足大量的数据库读取需求，以此增加应用的吞吐量。您需要在应用程序中分别配置主实例和每个只读节点的连接地址，才能实现将写请求发往主节点，而将读请求发往只读节点。

#### 功能特点

- 规格与实例一致。
- 不需要维护帐号与数据库，全部通过主实例同步。
- 提供系统性能监控。

#### 功能限制

- 单个“包年/包月”和“按需计费”实例最多可以创建 15 个只读节点。
- 实例恢复：不支持通过任意时间点创建临时节点，不支持通过备份集覆盖节点。
- 数据迁移：不支持将数据迁移至只读节点。
- 数据库管理：不支持创建和删除数据库。
- 帐号管理：只读节点不提供创建帐号权限，如需增加只读实例帐号，请在主实例上操作。

### 3.5.2 创建只读节点

#### 操作场景

只读节点用于增强实例主节点的读能力，减轻主节点负载。

GaussDB(for MySQL)的“包年/包月”和“按需计费”实例创建成功后，您可根据业务需要，创建只读节点。


#### 操作步骤

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面，选择指定的实例，单击操作列的“更多 > 创建只读”，进入“创建只读”页面。

您也可在实例的“基本信息”页面，单击拓扑图中的，创建只读节点。

**步骤 5** 在“创建只读”页面，选择“故障倒换优先级”和“购买数量”，单击“立即创建”。

故障倒换优先级的取值范围为 1~16，数字越小，优先级越大，即故障倒换时，主节点会优先倒换到优先级高的备节点上，优先级相同的备节点选为主节点的概率相同。

----结束

### 3.5.3 管理只读节点

只读节点创建成功后，您可以根据业务需要对其进行管理，比如：[只读节点升主节点](#)、[删除只读节点](#)。

#### 管理只读

**步骤 1** 登录管理控制台。

**步骤 2** 单击管理控制台左上角的，选择区域和项目。

**步骤 3** 在页面左上角单击，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

**步骤 4** 单击只读节点所在的实例名称，进入实例的“基本信息”页面。

**步骤 5** 在实例的“基本信息”页面的“节点信息”模块，您可以看到该实例下所有只读节点，并对其进行管理。

#### 说明

主节点和只读节点的读内网地址不支持修改。


----结束

### 3.5.4 只读节点升主节点

GaussDB(for MySQL)是一个多节点的实例，其中一个节点是主节点（Master），其他节点为只读节点。除了因系统故障[自动切换](#)外，对于用于高可用演练，或者需指定某个节点为主节点的场景，您也可以[手动切换](#)，指定一个只读节点为新的主节点。

#### 手动切换

**步骤 1** 登录管理控制台。

**步骤 2** 单击管理控制台左上角的，选择区域和项目。

**步骤 3** 在页面左上角单击，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

**步骤 4** 在“实例管理”页面的实例列表中，选择对应实例，单击实例名称进入“基本信息”页面。

步骤5 在“基本信息”页面的“节点信息”模块，选择目标只读节点，在“操作”列单击“只读升主”。

步骤6 在弹出框中单击“是”。

- 切换时可能会出现 30 秒左右的闪断，请确保应用具备重连机制。
- 切换过程中节点运行状态为“只读升主中”，此过程大概需要几秒或几分钟。
- 切换完成后，节点运行状态变为“正常”，您可查看到原先的只读节点和主节点的角色已经互换。

#### 须知

- 只读升主操作可能会造成几秒或几分钟的服务闪断。
- 只读升主操作会导致读内网地址发生切换，为避免对您的业务产生影响，建议使用网络信息中的“读写内网地址”连接实例。

----结束

## 自动切换

GaussDB(for MySQL)采用双活（Active-Active）的高可用实例架构，可读写的主节点和只读节点之间自动进行故障倒换（Failover），系统自动选取新的主节点。

GaussDB(for MySQL)每个节点都有一个故障倒换优先级，决定了故障倒换时被选取为主节点的概率高低。

- 故障倒换优先级的取值范围为 1~16，数字越小，优先级越高，即故障倒换时，主节点会优先倒换到优先级高的只读节点上。
- 当多个节点的优先级相同时，这些节点具有相同的概率被选取为主节点。

GaussDB(for MySQL)按以下步骤自动选取主节点：

1. 系统找出当前可以被选取的所有只读节点。
2. 选择优先级最高的一个或多个只读节点。
3. 如果由于网络原因、复制状态异常等，第一个节点切换失败，则会尝试切换下一个，直至成功。

## 3.5.5 删除只读节点

### 操作场景

对于“按需计费”模式的只读节点，您可根据业务需要，在 GaussDB(for MySQL)数据库“基本信息”页面手动删除来释放资源。

**须知**


只读节点删除后，不可恢复，请谨慎操作。

## 约束条件

- 当实例中的只读节点个数 $\geq 2$ 个时，才可选择删除只读节点，即实例中至少需要保留1个只读节点。
- 执行操作中的实例不能手动删除只读节点。

## 操作步骤

步骤1 登录管理控制台。

步骤2 单击管理控制台左上角的，选择区域和项目。

步骤3 在页面左上角单击，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤4 在“实例管理”页面的实例列表中，选择对应实例，单击实例名称进入“基本信息”页面。

步骤5 在“基本信息”页面的“节点信息”模块，选择目标只读节点，在“操作”列单击“删除”。

步骤6 在弹出框中单击“是”，稍后刷新“实例管理”页面，查看删除结果。

----结束

## 3.6 数据安全

### 3.6.1 重置管理员密码

#### 操作场景

GaussDB(for MySQL)仅支持通过主实例重置管理员密码。

在使用 GaussDB(for MySQL)过程中，如果忘记数据库帐号密码，可以重新设置密码。

以下情况不可重置密码。

- 数据库端口变更中。
- 主实例状态为创建中、恢复中、重启中、端口修改中、规格变更中、只读升主中、异常。


#### 注意事项


- 当您修改数据库实例的密码时，如果该实例中存在只读节点，登录只读节点的密码则会被同步修改。
- 重置密码生效时间取决于该主实例当前执行的业务数据量。

- 请定期（如三个月或六个月）修改用户密码，以提高系统安全性，防止出现密码被暴力破解等安全风险。
- 密码重置发生在备份操作之后，执行恢复操作时，密码会恢复为旧密码。

## 方法一

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面，选择指定的实例，选择“更多 > 重置密码”。

步骤 5 在“重置密码”弹框，输入新管理员密码及确认密码。

### 须知

请妥善保管您的密码，因为系统将无法获取您的密码信息。

所设置的密码，最小长度为 8 个字符，最大长度为 32 个字符，至少包含大写字母、小写字母、数字、特殊字符中的 3 种，其中，可输入~!@#%^\*\_=-+?特殊字符。请您输入高强度密码并定期修改，以提高安全性，防止出现密码被暴力破解等安全风险。

- 单击“确定”，提交重置。
- 单击“取消”，取消本次重置。

----结束

## 方式二

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面，选择指定的实例，单击实例名称，进入实例“基本信息”页面。

步骤 5 在“实例信息”模块的“管理员帐户名”处，单击“重置密码”，输入新管理员密码及确认密码。

### 须知

请妥善保管您的密码，因为系统将无法获取您的密码信息。



所设置的密码，最小长度为 8 个字符，最大长度为 32 个字符，至少包含大写字母、小写字母、数字、特殊字符中的 3 种，其中，可输入~!@#%^\*-\_+=?特殊字符。请您输入高强度密码并定期修改，以提高安全性，防止出现密码被暴力破解等安全风险。

- 单击“确定”，提交重置。
- 单击“取消”，取消本次重置。

----结束

## 3.6.2 修改实例内网安全组

### 操作场景

云数据库 GaussDB(for MySQL)服务支持修改实例的内网安全组。


### 操作步骤



步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面，选择指定的实例，单击实例名称。

步骤 5 在“基本信息”页签“网络信息”模块的“内网安全组”处，单击 ，选择对应的内网安全组。

- 单击 ，提交修改。
- 单击 ，取消修改。

步骤 6 稍后单击“基本信息”页面右上角的 ，查看修改结果。此过程需 1~3 分钟。

----结束

## 3.6.3 数据库安全设置

### 帐户密码等级设置

GaussDB(for MySQL)对在客户端新创建的数据库用户，设置了密码安全策略：

- 口令长度至少 8 个字符。
- 口令至少包含大写字母、小写字母、数字、特殊字符三种字符的组合，其中允许输入~!@#%^\*-\_+=?,()&\$特殊字符。

创建实例时，为用户提供了密码复杂度校验，由于 root 用户可以修改密码复杂度，安全起见，建议修改后的密码复杂度不低于 GaussDB(for MySQL)数据库的初始化设置。

## 帐户说明

您在创建 GaussDB(for MySQL)实例时，系统会自动为实例创建如下系统帐户（用户不可使用），用于给数据库实例提供完善的后台运维管理服务。

### 须知

删除、重命名、修改这些帐户的密码和权限信息，会导致实例运行异常，请谨慎操作。

- **rdsAdmin**: 管理帐户，拥有最高的 superuser 权限，用于查询和修改实例信息、故障排查、迁移、恢复等操作。
- **rdsRepl**: 复制帐户，用于只读节点在主实例上同步数据。
- **rdsBackup**: 备份帐户，用于后台的备份。
- **rdsMetric**: 指标监控帐户，用于 watchdog 采集数据库状态数据。
- **rdsProxy**: 数据库代理帐户，该帐户在开通读写分离时才会自动创建，用于通过读写分离地址连接数据库时鉴权使用。

## 3.7 数据备份

### 3.7.1 备份类型

GaussDB(for MySQL)支持数据库实例的备份和恢复，以保证数据可靠性。

- GaussDB(for MySQL)支持多种备份类型，默认开启自动备份。详细内容请参考[设置自动备份策略](#)。
- GaussDB(for MySQL)会定期进行全量备份和增量备份，用户可恢复数据到任意时间节点。详细内容请参考[将数据库实例恢复到指定时间点](#)。
- 备份操作是实例级的，备份范围包含数据库实例中所有数据。

### 备份的作用

当数据库或表被恶意或误删除，只能依赖于实例的备份保障数据安全。

### 全量备份

全量备份表示对当前状态下的数据库实例中的所有数据进行一次完整的备份，用户可在任意时刻使用全量备份恢复创建备份时的完整数据。

### 增量备份

GaussDB(for MySQL)数据库系统自动每 5 分钟对上一次自动备份或增量备份后更新的数据进行备份。

## 自动备份

GaussDB(for MySQL)会在数据库实例的备份时段中创建数据库实例的自动备份。系统根据您指定的备份保留期保存数据库实例的自动备份。如果需要，您可以将数据恢复到备份保留期中的任意时间点。

## 手动备份

您还可以创建手动备份对数据库进行备份，手动备份是由用户启动的数据库实例的全量备份，会一直保存，直到用户手动删除。

### 3.7.2 设置自动备份策略

#### 操作场景

创建 GaussDB(for MySQL)数据库实例时，系统默认开启自动备份策略，暂不支持关闭。实例创建成功后，您可根据业务需要设置自动备份策略。GaussDB(for MySQL)按照用户设置的自动备份策略对数据库进行备份。


GaussDB(for MySQL)的备份操作是实例级的，而不是数据库级的。当数据库故障或数据损坏时，可以通过备份恢复数据库，从而保证数据可靠性。由于开启备份会损耗数据库读写性能，建议您选择业务低峰时间段启动自动备份。

开启自动备份策略后，会自动触发一次全量备份。之后仍然会按照策略中的备份时间段和备份周期进行全量备份。实例在执行备份时，会将数据从实例上拷贝并上传到 OBS 备份空间，按照策略中的保留天数进行存放，备份时长和实例的数据量有关。

在进行全量备份的同时系统每 5 分钟会自动生成增量备份，用户不需要设置。生成的增量备份可以用来将库表数据恢复到指定时间点。

#### 查看或修改自动备份策略

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面，选择指定的实例，单击实例名称。

步骤 5 在左侧导航栏，单击“备份恢复”，单击“修改备份策略”。您可以查看到已设置的备份策略，如需修改备份策略，请调整以下参数的值。

- 保留天数：保留天数是指自动备份可保留的时间，默认为 7 天，可选择范围为 1~732 天。对于系统中最近一个全量备份文件，如果在全量备份未超过保留天数前系统会一直保留，直至新的全量备份超过保留天数后才会删除。
  - 增加保留天数，可提升数据可靠性，请根据需要设置。
  - 减少保留天数，会针对已有的备份文件生效，即超出备份保留天数的已有备份文件（包括全量备份和增量备份）会被自动删除，但手动备份不会自动删除，请您谨慎选择。

- 备份时间段：默认为 24 小时中，间隔一小时的随机的一个时间段，例如 01:00~02:00，12:00~13:00 等。

#### 📖 说明

建议根据业务情况，选择业务低峰时段。备份时间段以 UTC 时区保存。如果碰到夏令时/冬令时切换，备份时间段会因时区变化而改变。备份时间段内如果有其他任务长时间运行，可能会导致自动全备重试多次依然发起不了，最终跳过本次全备任务。

- 备份周期：默认全选，可修改，且至少选择一周中的 1 天。

若实例的备份策略被开启，则会立即触发一个全量的自动备份。

步骤 6 单击“确定”，确认修改。

----结束

## 3.7.3 创建手动备份

### 操作场景


GaussDB(for MySQL)支持对运行正常的主节点创建手动备份，用户可以通过手动备份恢复数据，从而保证数据的可靠性。

当数据库实例被删除时，GaussDB(for MySQL)数据库实例的自动备份将被同步删除，手动备份不会被删除。

当删除帐号时，自动备份和手动备份的数据将被同步删除。

### 方式一

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面，选择指定的实例，在操作列选择“更多 > 创建备份”。

步骤 5 在创建备份弹出框中，命名该备份，并添加描述，单击“确定”，提交备份创建，单击“取消”，取消创建。


- 备份名称的长度在 4~64 个字符之间，必须以字母开头，区分大小写，可以包含字母、数字、中划线或者下划线，不能包含其他特殊字符。
- 备份描述不能超过 256 个字符，且不能包含回车和!<"=>&特殊字符。
- 创建手动备份所需时间，由实例的数据量大小决定。
- 实例在执行备份时，会将数据从实例上拷贝并上传到 OBS 备份空间，备份时长和实例的数据量有关。


步骤 6 手动备份创建成功后，用户可在“备份恢复管理”页面，对其进行查看并管理。

----结束

## 方式二

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面，选择指定的实例，单击实例名称。

步骤 5 在左侧导航栏中选择“备份恢复”，单击“创建备份”，命名该备份，并添加描述，单击“确定”，提交备份创建，单击“取消”，取消创建。

- 备份名称的长度在 4~64 个字符之间，必须以字母开头，区分大小写，可以包含字母、数字、中划线或者下划线，不能包含其他特殊字符。
- 备份描述不能超过 256 个字符，且不能包含回车和!<"=>&特殊字符。
- 创建手动备份所需时间，由实例的数据量大小决定。
- 实例在执行备份时，会将数据从实例上拷贝并上传到 OBS 备份空间，备份时长和实例的数据量有关。

步骤 6 手动备份创建成功后，用户可在“备份恢复管理”页面，对其进行查看并管理。

----结束

## 3.7.4 导出备份信息

### 操作场景

GaussDB(for MySQL)服务支持导出备份，用户可以通过导出备份功能将备份信息（实例名称，备份 ID、备份名称、备份类型、备份开始时间，备份结束时间，备份状态，备份大小、描述）导出到 Excel 中，方便用户查看并分析备份信息。

### 约束限制

GaussDB(for MySQL)不支持下载自动备份和手动备份文件。


### 操作步骤

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在左侧导航栏，单击“备份恢复管理”，在“备份恢复管理”页面，勾选需要导出的

备份，在页面右上角单击 ，导出备份信息。

您也可以在“实例管理”页面，单击实例名称，进入“基本信息”页面，在左侧导航栏，单击，在“全量备份”页签下，勾选需要导出的备份，单击“导出”，导出备份信息。

- 目前只可导出当前页面的备份，不可跨页面导出。
- 导出的备份信息列表为 Excel 汇总表格，您可以对其进行分析，以满足业务需求。

步骤 5 查看导出的 GaussDB(for MySQL)备份。

----结束

## 3.7.5 删除手动备份

### 操作场景


GaussDB(for MySQL)支持对手动备份进行删除，从而释放相关存储空间。

#### 须知

手动备份删除后，不可恢复，请您谨慎操作。

### 操作步骤

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在左侧导航栏，单击“备份恢复管理”，在“备份恢复管理”页面，选择目标备份，单击操作列中的“删除”。

您也可以在“实例管理”页面，单击实例名称，进入“基本信息”页面，在左侧导航栏，单击“备份恢复”，选择目标备份，单击操作列中的“删除”。

如下备份不可被删除：

- 自动备份。
- 恢复中和创建中的备份。

步骤 5 单击“是”，删除手动备份。

----结束

## 3.8 数据恢复

### 3.8.1 恢复方案概览

云数据库 GaussDB(for MySQL)提供了多种方式恢复实例的数据，用以满足不同的使用场景，如数据故障或损坏，实例被误删除。

表3-1 恢复方案

恢复类型	恢复场景
将数据库实例恢复到指定时间点	适用于实例级数据恢复场景，将数据还原至某个指定的时间点。支持恢复到新实例、当前实例和已有实例。
将备份恢复至实例	支持将已有的自动备份和手动备份恢复到新实例、当前实例和已有实例。

### 3.8.2 将数据库实例恢复到指定时间点



#### 操作场景

GaussDB(for MySQL)支持恢复实例数据到指定时间点。

#### 须知

请勿在 GaussDB(for MySQL)实例的生命周期内执行“reset master”命令，以免造成恢复到指定时间点功能异常。

#### 操作步骤

- 步骤 1 登录管理控制台。
- 步骤 2 单击管理控制台左上角的 ，选择区域和项目。
- 步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。
- 步骤 4 在“实例管理”页面，选择指定的实例，单击实例名称。
- 步骤 5 在左侧导航栏中选择“备份恢复”页签，单击“恢复到指定时间点”。
- 步骤 6 选择需要恢复的日期和时间区间，选择或输入该恢复时间区间内的一个要恢复到的时间点，选择恢复方式为“新实例”或“当前实例”。
  1. 选择恢复到“新实例”，单击“确定”，跳转到“恢复到新实例”的服务页面，设置相关参数，单击“立即创建”。
    - 区域、数据库引擎、兼容的数据库版本，与原实例相同，不可修改。

- 数据库端口为默认值 3306。
  - 其他参数默认，用户可设置，请参见[创建实例](#)。
2. 选择恢复到“当前实例”，单击“下一步”，核对任务信息，无误后单击“确定”。

恢复到当前实例会将当前实例上的数据全部覆盖，并且恢复过程中数据库不可用，请在业务低峰期进行恢复。

步骤 7 查看恢复结果。

- 恢复到新实例，GaussDB(for MySQL)会为用户重新创建一个和备份创建时的时间点数据相同的实例。可看到实例由“创建中”变为“正常”，说明恢复成功。恢复成功的新实例是一个独立的实例，与原有实例没有关联。如需使用只读节点，请重新在该实例上进行创建。  
新实例恢复成功后，系统会自动执行一次全量备份。
- 恢复到当前实例，恢复到的目标实例运行状态由“恢复中”变为“正常”，说明恢复成功。

----结束


### 3.8.3 将备份恢复至实例

#### 操作场景

GaussDB(for MySQL)支持使用已有的自动备份和手动备份，将实例数据恢复到备份被创建时的状态。该操作恢复的为整个实例的数据。

#### 操作步骤

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的，选择区域和项目。

步骤 3 在页面左上角单击，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 选择需要恢复的备份

您可以在左侧导航栏单击“备份恢复管理”，选择需要恢复的备份，单击操作列的“恢复”。

或者在“实例管理”页面，选择指定的实例，单击实例名称。在左侧导航栏单击“备份恢复”，选择需要恢复的备份，单击操作列的“恢复”。

步骤 5 选择恢复方式，当前支持将备份恢复到“新实例”、“当前实例”和“已有实例”。

1. 选择恢复到“新实例”，单击“确定”，跳转到“恢复到新实例”的服务页面，设置相关参数，单击“立即创建”。
  - 区域、数据库引擎、兼容的数据库版本，与原实例相同，不可修改。
  - 数据库端口为默认值 3306。
  - 其他参数默认，用户可设置，请参见[创建实例](#)。



2. 选择恢复到“当前实例”，单击“下一步”，核对任务信息，无误后单击“确定”。
  - 恢复到当前实例会将当前实例上的数据全部覆盖，并且恢复过程中数据库不可用，请在业务低峰期进行恢复。

图3-5 恢复到当前实例



3. 选择恢复到“已有实例”，勾选需要恢复到的目标实例，单击“下一步”，核对任务信息，无误后单击“确定”。
  - 恢复到目标实例会导致数据、root 密码被覆盖，恢复过程中数据库不可用。目标实例引擎类型、版本号、表名大小写敏感要求与原实例相同。
  - 如果恢复到已有实例后，已有实例的原密码无法连接到数据库，可通过重置密码解决。

图3-6 恢复到已有实例



步骤 6 查看恢复结果。

- 恢复到新实例，GaussDB(for MySQL)会为用户重新创建一个和备份被创建时的时间点数据相同的实例。可看到实例由“创建中”变为“正常”，说明恢复成功。恢复成功的新实例是一个独立的实例，与原有实例没有关联。如需使用只读节点，请重新在该实例上进行创建。

新实例恢复成功后，系统会自动执行一次全量备份。

----结束

## 3.9 连接管理

### 3.9.1 绑定和解绑弹性公网 IP

#### 操作场景

GaussDB(for MySQL)数据库实例创建成功后（默认未绑定“读写公网地址”），您可根据业务需要，绑定“读写公网地址”。

GaussDB(for MySQL)服务支持用户绑定弹性公网 IP，用于在公共网络访问数据库实例，绑定后也可根据需要解绑。

**须知**

- 为保证数据库可正常访问，请确保数据库使用的安全组开通了相关端口的访问权限，假设数据库的访问端口是 3306，那么需确保安全组开通了 3306 端口的访问。
- 绑定弹性公网 IP 时无可选的 EIP，原因可能是该 EIP 已被其他应用绑定，GaussDB(for MySQL)绑定的 EIP 需要是未绑定状态才可以绑定，请重新购买弹性公网 IP。

## 前提条件

用户需要在 VPC 申请一个“读写公网地址”。


对于已绑定“读写公网地址”的实例，需解绑后，才可重新绑定其他“读写公网地址”。

## 注意事项

- 公网访问会降低实例的安全性，请谨慎选择。为了获得更快的传输速率和更高的安全级别，建议您将应用迁移到与您的 GaussDB(for MySQL)在同一区域的弹性云服务器上。

## 绑定弹性公网 IP

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面，选择目标实例，单击实例名称，进入实例的“基本信息”页面。

步骤 5 在网络信息模块，单击“读写公网地址”后面的“绑定”。

步骤 6 在弹出框的弹性公网 IP 地址列表中，选择目标弹性公网 IP，单击“确定”，提交绑定任务。

如果没有可用的弹性公网 IP，您可单击“查看弹性公网 IP”，跳转到网络控制台创建弹性公网 IP，创建完成后请返回实例的基本信息页面绑定弹性公网 IP。

**须知**

您需要设置安全组，开通需访问数据库的 IP 地址和端口，才可以访问数据库实例。

步骤 7 在“网络信息”模块“读写公网地址”处查看绑定成功的弹性公网 IP。

如需解绑，请参见[解绑弹性公网 IP](#)。

----结束

## 解绑弹性公网 IP




- 步骤 1 对于已绑定弹性公网 IP 的实例，在“实例管理”页面，选择目标实例，单击实例名称，进入实例“基本信息”页面。
  - 步骤 2 在网络信息模块，单击“读写公网地址”后面的“解绑”。
  - 步骤 3 在“解绑”弹出框中单击“是”，解绑弹性公网 IP。如需重新绑定，请参见[绑定弹性公网 IP](#)。
- 结束

## 3.9.2 修改数据库端口



### 操作场景

GaussDB(for MySQL)支持修改实例的数据库端口。修改实例的数据库端口，会同时修改主节点和所有只读节点的端口。

### 操作步骤

- 步骤 1 登录管理控制台。
- 步骤 2 单击管理控制台左上角的，选择区域和项目。
- 步骤 3 在页面左上角单击，选择“数据库 > 云数据库 GaussDB(for MySQL)”。
- 步骤 4 在“实例管理”页面，选择指定的实例，单击实例名称。
- 步骤 5 在“基本信息”，在“网络信息”模块“数据库端口”处，单击，修改数据库端口。

GaussDB(for MySQL)数据库端口设置范围为 1025~65534，其中 5342、5343、5344、5345、12017、20000、20201、20202 和 33062 被系统占用不可设置。

- 单击，提交修改。
    - 在弹出框中，单击“确定”，提交修改。
      - i. 修改主节点的数据库端口，对应的实例下所有节点均会被修改且重启。
      - ii. 此过程需要 1~5 分钟左右。
  - 单击，取消修改。
- 步骤 6 在实例的“基本信息”页面，查看修改结果。

----结束

## 3.10 参数模板管理

### 3.10.1 创建参数模板

您可以使用数据库参数模板中的参数来管理数据库引擎配置。数据库参数模板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。

#### 须知

并非所有数据库引擎参数都可在用户创建的数据库参数模板中进行更改。

如果您想使用您自己的数据库参数模板，只需创建一个新的数据库参数模板，创建实例的时候选择该参数模板，如果是在创建实例后有这个需求，可以重新应用该参数模板，请参见[应用参数模板](#)。

若您已成功创建数据库参数模板，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，可通过复制参数模板来实现，请参见[复制参数模板](#)。

以下是您在使用数据库参数模板中的参数时应了解的几个要点：

- 当您在实例的“参数修改”页面，修改参数并保存后，动态参数将立即生效，静态参数需要手动重启实例后生效。此操作仅应用于当前实例，不会对其他实例造成影响。
- 当您在“参数模板管理”页面，更改参数并保存数据库参数模板时，需执行应用后，才对数据库实例生效。动态参数将立即生效，静态参数需要手动重启实例后生效。
- 在数据库参数模板内设置参数不恰当可能会造成性能降低和系统不稳定。修改数据库参数时应始终保持谨慎，且修改数据库参数模板前要备份数据。将参数模板更改应用于生产数据库实例前，您应当在测试数据库实例上试用这些参数模板设置更改。


#### 说明

每个用户最多可以创建 100 个参数模板。

GaussDB(for MySQL)引擎共享参数模板配额。

## 操作步骤

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“参数模板管理”页面，单击“创建参数模板”。

步骤 5 命名并添加对该参数模板的描述，单击“确定”，创建参数模板。

- 选择该数据库引擎参数模板所需应用的参数模板类型。

- 参数模板名称长度在 1~64 个字符之间，区分大小写，可包含字母、数字、中划线、下划线或句点，不能包含其他特殊字符。
- 参数模板的描述长度不能超过 256 个字符，且不能包含回车和>!<"&'=特殊字符。

----结束

## 3.10.2 编辑参数模板

为确保 GaussDB(for MySQL)服务发挥出最优性能，用户可根据业务需求对用户创建的参数模板里边的参数进行调整，并将该模板应用到目标实例。

您可以修改用户创建的数据库参数模板中的参数值，但不能更改默认数据库参数模板中的参数值。

以下是您在使用数据库参数模板中的参数时应了解的几个要点：


- 当您在实例的“参数修改”页面，修改参数并保存后，动态参数将立即生效，静态参数需要手动重启实例后生效。此操作仅应用于当前实例，不会对其他实例造成影响。
- 当您在“参数模板管理”页面，更改参数并保存数据库参数模板时，需执行应用后，才对数据库实例生效。动态参数将立即生效，静态参数需要手动重启实例后生效。

### 说明

系统提供的默认参数模板不允许修改，只可单击参数模板名进行查看。当用户参数设置不合理导致数据库无法启动时，可参考默认参数模板重新配置。

## 批量修改参数

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“参数模板管理”页面的“自定义”页签，选择目标参数模板，单击参数模板名称。

步骤 5 根据需要修改相关参数值。

可进行的操作如下：

### 须知

当某个自定义参数模板里的 `time_zone` 参数被修改后，在创建新实例并应用该参数模板时 `time_zone` 参数不会生效。如果需要修改，请参见 [修改当前实例的参数模板](#) 重新修改实例的 `time_zone` 参数。

- 单击“保存”，在弹出框中单击“是”，保存修改。
- 单击“取消”，放弃本次设置。


- 单击“预览”，可对比参数修改前和修改后的值。

步骤6 参数修改完成后，您可在“参数模板管理”页面单击目标参数模板名称，然后在左侧导航栏中，单击“参数修改历史”查看参数的修改详情。

----结束

## 修改当前实例的参数模板

步骤1 登录管理控制台。

步骤2 单击管理控制台左上角的，选择区域和项目。

步骤3 在页面左上角单击，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤4 在“实例管理”页面，选择指定的实例，单击实例名称，进入实例的基本信息页面。

步骤5 在左侧导航栏中选择“参数修改”，在“参数”页签修改相应参数。

可进行的操作如下：

### 须知

参数模板修改后，根据参数列表中“是否需要重启”提示，进行相应操作：

- 是：在实例列表中，查看“运行状态”，如果显示参数模板变更，等待重启，则需重启实例使之生效。
- 否：无需重启，参数修改立即生效。

- 单击“保存”，在弹出框中单击“是”，保存修改。
- 单击“取消”，放弃本次设置。
- 单击“预览”，可对比参数修改前和修改后的值。

参数修改完成后，您可在“参数修改”页面，选择“参数修改历史”页签查看参数的修改详情。

----结束


## 3.10.3 导出参数


### 操作场景

您可以将该实例对应的参数模板信息（参数名称，值，描述）导出到 CSV 中，方便查看并分析。

### 操作步骤

步骤1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面，选择指定的实例，单击实例名称，进入实例的基本信息页面。

步骤 5 在左侧导航栏中选择“参数修改”，在“参数”页签单击“导出”。在弹出框中，填写文件名称，单击“确定”。此操作将该实例对应的参数模板信息（参数名称，值，描述）导出到 CSV 表中，方便用户查看并分析。

#### 说明

文件名称长度在 4~81 个字符之间，必须以字母开头，可以包含字母、数字、中划线或下划线，不能包含其他特殊字符。

----结束

## 3.10.4 比较参数模板

### 操作场景

您可以比较同数据库类型的实例参数模板，以了解当前实例参数的差异项。

您也可以比较同数据库类型的默认参数模板，以了解当前参数模板的配置情况。

### 比较当前实例参数模板

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面，单击实例名称，进入实例的“基本信息”页签。

步骤 5 在左侧导航栏中选择“参数修改”，在“参数”子页签中单击“比较参数”。

步骤 6 在弹出框中选择当前实例同数据库类型的参数模板，单击“确定”，比较两个参数的差异项。

- 有差异项，则会显示差异参数的如下信息：参数名称、当前实例参数模板的参数值和被比较参数模板的参数值。
- 无差异项，则不显示。


----结束

### 比较目标参数模板

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。



- 步骤3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。
- 步骤4 在“参数模板管理”页面的“自定义”页签，选择一个用户创建的参数模板，单击“比较”。
- 步骤5 选择同一数据库引擎的不同参数模板，单击“确定”，比较两个参数模板之间的配置参数差异项。
- 有差异项，则会显示差异参数模板的如下信息：参数名称、参数值。
  - 无差异项，则不显示。
- 结束

### 3.10.5 查看参数修改历史



#### 操作场景

您可以查看当前实例所使用参数模板以及自定义参数模板的修改历史，以满足业务需要。

#### 说明

用户创建的新参数模板，在未进行参数修改前，无修改历史。

#### 查看当前实例的参数修改历史


- 步骤1 登录管理控制台。
- 步骤2 单击管理控制台左上角的 ，选择区域和项目。
- 步骤3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。
- 步骤4 在“实例管理”页面，选择指定的实例，单击实例名称，进入实例的基本信息页面。
- 步骤5 在左侧导航栏，单击“参数修改”，单击“参数修改历史”。

您可查看参数对应的参数名称、修改前参数值、修改后参数值、修改状态、修改时间、是否应用以及应用时间。

如修改后参数模板未应用，请根据业务需要，参考[应用参数模板](#)，将其应用到对应实例。

----结束

#### 查看目标参数模板的参数修改历史

- 步骤1 登录管理控制台。
- 步骤2 单击管理控制台左上角的 ，选择区域和项目。
- 步骤3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“参数模板管理”页面的“自定义”页签，单击目标参数模板名称。

步骤 5 单击“参数修改历史”。

您可查看参数对应的参数名称、修改前参数值、修改后参数值、修改状态和修改时间。

----结束

## 3.10.6 复制参数模板

### 操作场景


您可以复制您创建的自定义数据库参数模板。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。

复制数据库参数模板之后，您应至少等待 5 分钟，再创建使用该数据库参数模板作为默认参数模板的第一个数据库实例。

您无法复制默认参数模板。不过，您可以创建基于默认参数模板的新参数模板。

### 复制当前实例的参数模板

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面，选择指定的实例，单击实例名称，进入实例的基本信息页面。

步骤 5 在左侧导航栏中选择“参数修改”，在“参数”页签单击“复制”，将该实例对应参数列表导出并生成一个参数模板，供您后期使用。

步骤 6 在弹出框中，填写新参数模板名称和描述，单击“确定”。

- 参数模板名称长度在 1~64 个字符之间，区分大小写，可包含字母、数字、中划线、下划线或句点，不能包含其他特殊字符。
- 参数模板的描述长度不能超过 256 个字符，且不能包含回车和 >!<"&'=特殊字符。

创建完成后，会在“参数模板管理”的“自定义”页签下，生成一个新的参数模板，您可在参数模板列表中对其进行管理。

----结束

### 复制目标参数模板

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤3 在页面左上角单击，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤4 在“参数模板管理”页面的“自定义”页签，选择需要复制的参数模板，单击“复制”。

步骤5 在弹出框中，填写新参数模板名称和描述，单击“确定”。

- 参数模板名称长度在 1~64 个字符之间，区分大小写，可包含字母、数字、中划线、下划线或句点，不能包含其他特殊字符。
- 参数模板的描述长度不能超过 256 个字符，且不能包含回车和>!<"&'=特殊字符。

创建完成后，会生成一个新的参数模板，您可在参数模板列表中对其进行管理。

----结束

### 3.10.7 重置参数模板

#### 操作场景

您可根据自己的业务需求，重置自己创建的参数模板对应的所有参数，使其恢复到默认值。

#### 操作步骤

步骤1 登录管理控制台。

步骤2 单击管理控制台左上角的，选择区域和项目。

步骤3 在页面左上角单击，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤4 在“参数模板管理”页面的“自定义”页签，选择需要设置的参数模板，单击“更多 > 重置”。

步骤5 单击“是”，重置所有参数为其默认值。

#### 说明

对于某些参数模板重置后，您需在实例列表中，选择对应的实例，单击实例名称，在“基本信息”页签中查看参数模板状态，如果显示“等待重启”，则需重启关联的实例，并应用后，使之生效。

----结束

### 3.10.8 应用参数模板

#### 操作场景

参数模板编辑修改后，不会立即应用到实例，您可以根据业务需要应用到实例中。

- 参数“innodb\_buffer\_pool\_size”跟内存强相关，不同规格的实例有不同的区间范围，如果应用参数模板时，该参数超过了实例本身的区间大小，则会取实例区间范围的最大值。
- 参数模板只能应用于相同版本的实例中。
- 在金融版实例上创建的只读实例为普通实例，但沿用了金融版实例的参数模板，所以该只读实例也不能应用其他版本的参数模板。

## 操作步骤

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“参数模板管理”页面，根据参数模板类型不同进行如下操作。

- 若需要将默认参数模板应用到实例，在“系统默认”页签的目标参数模板单击“应用”。
- 若需要将用户自己创建的参数模板应用到实例，在“自定义”页签的目标参数模板单击“更多>应用”。

一个参数模板可被应用到一个或多个实例。

步骤 5 在弹出框中，选择或输入所需应用的实例，单击“确定”。

参数模板应用成功后，您可 [查看参数模板应用记录](#)。

----结束

## 3.10.9 查看参数模板应用记录

### 操作场景

参数模板编辑修改后，您可根据业务需要将其应用到对应实例中，GaussDB(for MySQL)支持查看参数模板所应用到实例的记录。

### 操作步骤

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“参数模板管理”页面：

- 在“系统默认”页签，选择目标参数模板，单击“应用记录”，查看应用记录。
- 在“自定义”页签，选择目标参数模板，单击“更多>应用记录”，查看应用记录。

您可查看参数模板所应用到的实例名称/ID、应用状态、应用时间、失败原因。

----结束

### 3.10.10 修改参数模板描述

#### 操作场景


参数模板创建成功后，用户可根据需要对自己创建的参数模板描述进行修改。


#### 📖 说明


默认参数模板的描述不可修改。

#### 操作步骤

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“参数模板管理”页面的“自定义”页签，选择一个用户创建的参数模板，单击“描述”列 。

步骤 5 输入新的描述信息，单击 ，提交修改，单击 ，取消修改。

- 修改成功后，可在参数模板列表的“描述”列查看改后的描述信息。
- 参数模板的描述长度不能超过 256 个字符，且不能包含>!<"&'=特殊字符。

----结束

### 3.10.11 删除参数模板

#### 操作场景

您可删除废弃的参数模板。


#### 须知

- 参数模板删除后，不可恢复，请谨慎操作。
- 默认参数模板不可被删除。

#### 操作步骤

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤3 在页面左上角单击，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤4 在“参数模板管理”页面的“自定义”页签，选择需要删除的参数模板，单击“更多 > 删除”。

步骤5 单击“是”，删除。

----结束

## 3.11 监控指标与告警

### 3.11.1 支持的监控指标

#### 功能说明

通过 Cloud Eye 的资源监控功能可以了解系统的运行情况。本节定义了云数据库 GaussDB(for MySQL)上报云监控的监控指标的命名空间，监控指标列表和监控指标维度。

#### 命名空间

SYS.GAUSSDB

#### 实例支持的监控指标

表3-2 GaussDB(for MySQL)实例支持的监控指标

指标 ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
gaussdb_mysql001_cpu_util	CPU 使用率	该指标用于统计测量对象的 CPU 利用率。	0~100%	GaussDB(for MySQL)实例	1 分钟
gaussdb_mysql002_mem_util	内存使用率	该指标用于统计测量对象的内存利用率。	0~100%	GaussDB(for MySQL)实例	1 分钟
gaussdb_mysql004_bytes_in	网络输入吞吐量	该指标用于统计平均每秒从测量对象的所有网络适配器输入的流量。	$\geq 0$ bytes/s	GaussDB(for MySQL)实例	1 分钟
gaussdb_mysql005_bytes_out	网络输出吞吐量	该指标用于统计平均每秒从测量对象的所有网络	$\geq 0$ bytes/s	GaussDB(for MySQL)实例	1 分钟

指标 ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
t		适配器输出的流量。			
gaussdb_mysql006_conn_count	数据库总连接数	该指标用于统计连接到 GaussDB(for MySQL)服务器的总连接数。	$\geq 0$ counts	GaussDB(for MySQL)实例	1 分钟
gaussdb_mysql007_conn_active_count	当前活跃连接数	该指标用于统计当前活跃的连接数。	$\geq 0$ counts	GaussDB(for MySQL)实例	1 分钟
gaussdb_mysql008_qps	QPS	该指标用于统计 SQL 语句查询次数, 包含 DDL, DML, SHOW 语句, SET 语句和存储过程。	$\geq 0$ Times/s	GaussDB(for MySQL)实例	1 分钟
gaussdb_mysql009_tps	TPS	该指标用于统计平均每秒事务执行次数, 包含提交的和回退的。	$\geq 0$ Times/s	GaussDB(for MySQL)实例	1 分钟
gaussdb_mysql010_innodb_buf_usage	缓冲池利用率	该指标用于统计使用的页与 InnoDB 缓存中数据页总数比例。	0-1	GaussDB(for MySQL)实例	1 分钟
gaussdb_mysql011_innodb_buf_hit	缓冲池命中率	该指标用于统计该段时间读命中与读请求数比例。	0-1	GaussDB(for MySQL)实例	1 分钟
gaussdb_mysql012_innodb_buf_dirty	缓冲池脏块率	该指标用于统计 InnoDB 缓存中脏数据与数据比例。	0-1	GaussDB(for MySQL)实例	1 分钟
gaussdb_mysql013_innodb_reads	InnoDB 读取吞吐量	该指标用于统计 InnoDB 平均每秒读字节数。	$\geq 0$ bytes/s	GaussDB(for MySQL)实例	1 分钟
gaussdb_mysql014	InnoDB	该指标用于统计	$\geq 0$	GaussDB(for MySQL)	1 分钟

指标 ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
_innodb_writes	写入吞吐量	InnoDB 平均每秒写页面数据字节数。 GaussDB(for MySQL)只写入临时表页面。	bytes/s	实例	
gaussdb_mysql017_innodb_log_write_req_count	InnoDB 日志写请求频率	该指标用于统计平均每秒的日志写请求数。	$\geq 0$ counts	GaussDB(for MySQL)实例	1 分钟
gaussdb_mysql019_innodb_log_writes	InnoDB log buffer 写入 log file 的总次数	该指标用于采集 InnoDB 表上的 log buffer 写入 log file 的总次数。	$\geq 0$ counts	GaussDB(for MySQL)实例	1 分钟
gaussdb_mysql020_temp_tbl_count	临时表数量	该指标用于统计 GaussDB(for MySQL)执行语句时在硬盘上自动创建的临时表的数量。	$\geq 0$ counts	GaussDB(for MySQL)实例	1 分钟
gaussdb_mysql028_comdml_del_count	Delete 语句执行频率	该指标用于统计平均每秒 Delete 语句执行次数。	$\geq 0$ counts/s	GaussDB(for MySQL)实例	1 分钟
gaussdb_mysql029_comdml_ins_count	Insert 语句执行频率	该指标用于统计平均每秒 Insert 语句执行次数。	$\geq 0$ counts/s	GaussDB(for MySQL)实例	1 分钟
gaussdb_mysql030_comdml_ins_sel_count	Insert_Select 语句执行频率	该指标用于统计平均每秒 Insert_Select 语句执行次数。	$\geq 0$ counts/s	GaussDB(for MySQL)实例	1 分钟
gaussdb_mysql031_comdml_rep_count	Replace 语句执行频率	该指标用于统计平均每秒 Replace 语句执行次数。	$\geq 0$ counts/s	GaussDB(for MySQL)实例	1 分钟
gaussdb_mysql032_comdml_replace_sel_count	Replace_Selection 语句执行频率	该指标用于统计平均每秒 Replace_Selection 语句执行次数。	$\geq 0$ counts/s	GaussDB(for MySQL)实例	1 分钟



指标 ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
rep_sel_count	行频率	n 语句执行次数。	s		
gaussdb_mysql033_comdml_sel_count	Select 语句执行频率	该指标用于统计平均每秒 Select 语句执行次数。	$\geq 0$ counts/s	GaussDB(for MySQL) 实例	1 分钟
gaussdb_mysql034_comdml_upd_count	Update 语句执行频率	该指标用于统计平均每秒 Update 语句执行次数。	$\geq 0$ counts/s	GaussDB(for MySQL) 实例	1 分钟
gaussdb_mysql035_innodb_delete_row_count	行删除速率	该指标用于统计平均每秒从 InnoDB 表删除的行数。	$\geq 0$ counts/s	GaussDB(for MySQL) 实例	1 分钟
gaussdb_mysql036_innodb_insert_row_count	行插入速率	该指标用于统计平均每秒向 InnoDB 表插入的行数。	$\geq 0$ counts/s	GaussDB(for MySQL) 实例	1 分钟
gaussdb_mysql037_innodb_read_row_count	行读取速率	该指标用于统计平均每秒从 InnoDB 表读取的行数。	$\geq 0$ counts/s	GaussDB(for MySQL) 实例	1 分钟
gaussdb_mysql038_innodb_update_row_count	行更新速率	该指标用于统计平均每秒向 InnoDB 表更新的行数。	$\geq 0$ counts/s	GaussDB(for MySQL) 实例	1 分钟
gaussdb_mysql048_disk_used_size	磁盘使用量	该指标用于统计测量对象的磁盘使用大小。	0GB~128TB	GaussDB(for MySQL) 实例	1 分钟
gaussdb_mysql072_conn_usage	连接数使用率	该指标用于统计当前已用的 GaussDB(for MySQL) 连接数占最大连接数的百分比。	0~100%	GaussDB(for MySQL) 实例	1 分钟
gaussdb_mysql074_slow_query	慢日志个数统计	该指标展示每分钟 GaussDB(for MySQL) 产生慢	$\geq 0$ counts/min	GaussDB(for MySQL) 实例	1 分钟

指标 ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
eries	计	日志的数量。			
gaussdb_mysql077_replication_delay	数据同步延迟	该指标用于采集实例的数据同步延迟时间。	$\geq 0$ s	GaussDB(for MySQL)实例	1 分钟
gaussdb_mysql104_dfv_write_delay	存储写时延	该指标用于统计某段时间写入数据到存储层的平均时延。	$\geq 0$ ms	GaussDB(for MySQL)实例	1 分钟
gaussdb_mysql105_dfv_read_delay	存储读时延	该指标用于统计某段时间从存储层读取数据的平均时延。	$\geq 0$ ms	GaussDB(for MySQL)实例	1 分钟
gaussdb_mysql106_innodb_row_lock_current_waits	InnoDB 行锁数量	该指标用于采集 InnoDB 表上的操作当前正在等待的行锁数量。	$\geq 0$ Locks/s	GaussDB(for MySQL)实例	1 分钟
gaussdb_mysql107_comdml_ins_and_ins_sel_count	Insert 和 Insert_Select 语句执行频率	该指标用于统计平均每秒 Insert 和 Insert_Select 语句的执行次数。	$\geq 0$ counts/s	GaussDB(for MySQL)实例	1 分钟
gaussdb_mysql108_com_commit_count	Commit 语句执行频率	该指标用于统计平均每秒 Commit 语句的执行次数。	$\geq 0$ counts/s	GaussDB(for MySQL)实例	1 分钟
gaussdb_mysql109_com_rollback_count	Rollback 语句执行频率	该指标用于统计平均每秒 Rollback 语句的执行次数。	$\geq 0$ counts/s	GaussDB(for MySQL)实例	1 分钟
gaussdb_mysql110_innodb_bufpool_reads	InnoDB 存储层读请求频率	该指标用于统计平均每秒 InnoDB 从存储层读取数据的请求次数。	$\geq 0$ counts/s	GaussDB(for MySQL)实例	1 分钟
gaussdb_mysql111_innodb_b	InnoDB 读请求	该指标用于统计平均每秒	$\geq 0$ counts/s	GaussDB(for MySQL)实例	1 分钟

指标 ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
ufpool_read_requests	频率	InnoDB 读取数据的请求次数。	s		
gaussdb_mysql14_innodb_bufpool_read_ahead	InnoDB 顺序预读页数	该指标用于采集 InnoDB 表上的顺序预读页数。	$\geq 0$ counts	GaussDB(for MySQL) 实例	1 分钟
gaussdb_mysql15_innodb_bufpool_read_ahead_evicted	InnoDB 顺序预读, 但未访问过的页数	该指标用于采集 InnoDB 表上的顺序预读, 但未访问过的页数。	$\geq 0$ counts	GaussDB(for MySQL) 实例	1 分钟
gaussdb_mysql16_innodb_bufpool_read_ahead_rnd	InnoDB 随机预读页数	该指标用于采集 InnoDB 表上的随机预读页数。	$\geq 0$ counts	GaussDB(for MySQL) 实例	1 分钟
gaussdb_mysql17_innodb_pages_read	InnoDB 读取物理 page 的数量	该指标用于采集 InnoDB 表上的读取物理 page 的数量。	$\geq 0$ counts	GaussDB(for MySQL) 实例	1 分钟
gaussdb_mysql18_innodb_pages_written	InnoDB 写入物理 page 的数量	该指标用于采集 InnoDB 表上的写入物理 page 的数量。	$\geq 0$ counts	GaussDB(for MySQL) 实例	1 分钟
gaussdb_mysql121_innodb_row_lock_time	行锁花费时间	该指标用于统计该段时间内 InnoDB 表上行锁花费时间。	$\geq 0$ ms	GaussDB(for MySQL) 实例	1 分钟
gaussdb_mysql122_innodb_row_lock_waits	行锁等待数	该指标用于统计该段时间内 InnoDB 表上行锁数量。	$\geq 0$ counts/min	GaussDB(for MySQL) 实例	1 分钟
gaussdb_mysql123_sort_range	范围排序数	该指标用于统计该段时间内使用范围扫描完成的排序数。	$\geq 0$ counts/min	GaussDB(for MySQL) 实例	1 分钟

指标 ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
gaussdb_mysql124_sort_rows	行排序数	该指标用于统计该段时间内已排序的行数。	$\geq 0$ counts/ min	GaussDB(for MySQL) 实例	1 分钟
gaussdb_mysql125_sort_scan	扫描表排序数	该指标用于统计该段时间内通过扫描表完成的排序数。	$\geq 0$ counts/ min	GaussDB(for MySQL) 实例	1 分钟
gaussdb_mysql126_table_open_cache_hits	打开表缓存查找的命中数	该指标用于统计该段时间内打开表缓存查找的命中数。	$\geq 0$ counts/ min	GaussDB(for MySQL) 实例	1 分钟
gaussdb_mysql127_table_open_cache_misses	打开表缓存查找的未命中数	该指标用于统计该段时间内打开表缓存查找的未命中数。	$\geq 0$ counts/ min	GaussDB(for MySQL) 实例	1 分钟
gaussdb_mysql128_long_trx_count	未关闭的长事务个数	该指标用于统计未关闭的长事务个数。	$\geq 0$ counts	GaussDB(for MySQL) 实例	150 秒
gaussdb_mysql342_iostat_iops_write	IO 写 IOPS	该指标用于采集磁盘每秒写次数。	$\geq 0$ counts/ s	GaussDB(for MySQL) 实例	1 分钟
gaussdb_mysql344_iostat_iops_read	IO 读 IOPS	该指标用于采集磁盘每秒读次数。	$\geq 0$ counts/ s	GaussDB(for MySQL) 实例	1 分钟
gaussdb_mysql346_iostat_throughput_write	IO 写带宽	该指标用于采集磁盘每秒写带宽。	$\geq 0$ bytes/s	GaussDB(for MySQL) 实例	1 分钟
gaussdb_mysql348_iostat_throughput_read	IO 读带宽	该指标用于采集磁盘每秒读带宽。	$\geq 0$ bytes/s	GaussDB(for MySQL) 实例	1 分钟
gaussdb_mysql371_taurus_binlog_total	Binlog 文件个数	该指标用于统计 GaussDB(for MySQL)Binlog	$\geq 0$	GaussDB(for MySQL) 实例	5 分钟

指标 ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
_file_counts		文件数量。			
gaussdb_mysql378_create_temp_tbl_per_min	临时表每分钟创建数	该指标用于统计 GaussDB(for MySQL) 执行语句时在硬盘上每分钟自动创建的临时表的数量。	$\geq 0$ counts/min	GaussDB(for MySQL) 实例	1 分钟

## 维度

表3-3 监控指标维度

Key	Value
gaussdb_mysql_instance_id	GaussDB(for MySQL)实例 ID
gaussdb_mysql_node_id	GaussDB(for MySQL)实例节点 ID

## 3.11.2 查看监控指标

### 3.11.2.1 查看实例监控指标

#### 操作场景

云监控服务可以对 GaussDB(for MySQL) 的运行状态进行日常监控。您可以通过管理控制台，直观地查看 GaussDB(for MySQL) 的各项监控指标。通过监控数据库运行时的系统资源利用率，您可以识别出什么时间段资源占用率最高，然后到错误日志或慢日志中分析可能存在问题的 SQL 语句，从而优化数据库性能。

#### 前提条件


- GaussDB(for MySQL) 正常运行。  
故障、删除状态的 GaussDB(for MySQL)，无法在云监控中查看其监控指标。当 GaussDB(for MySQL) 再次启动或恢复后，即可正常查看。

#### 说明


- 故障 24 小时的 GaussDB(for MySQL)，云监控将默认该 GaussDB(for MySQL) 不存在，并在监控列表中删除，不再对其进行监控，但告警规则需要用户手动清理。
- GaussDB(for MySQL) 已正常运行一段时间（约 10 分钟）。  
对于新创建的 GaussDB(for MySQL)，需要等待一段时间，才能查看上报的监控数据和监控视图。

## 查看节点监控指标

步骤 1 登录管理控制台。

步骤 2 在页面左上角单击 ，选择“管理与监管 > 云监控服务 CES”，进入“云监控服务”信息页面。

步骤 3 在左侧导航栏选择“云服务监控 > 云数据库 GaussDB(for MySQL)”。

步骤 4 选择目标实例，单击实例名称左侧的 ，选中一个目标节点，单击操作列的“查看监控指标”，查看此节点的监控指标。

您还可以通过如下方式跳转到云监控服务页面查看监控指标：

- 在 GaussDB(for MySQL)的“实例管理”页面，单击目标实例名称进入基本信息页面，在页面右上角，单击“查看监控指标”，跳转到云监控服务页面，查看监控指标。
- 在 GaussDB(for MySQL)的“基本信息”页面底部，节点信息对应的操作列下，单击“查看监控指标”，跳转到云监控服务页面，查看监控指标。

步骤 5 在 Cloud Eye 页面，可以查看实例监控信息。

Cloud Eye 支持的性能指标监控时间窗包括：近 1 小时、近 3 小时、近 12 小时、近 24 小时、近 7 天。

----结束

## 3.11.3 设置告警规则

### 3.11.3.1 创建实例告警规则

#### 操作场景

通过设置 GaussDB(for MySQL)告警规则，用户可自定义监控目标与通知策略，及时了解 GaussDB(for MySQL)运行状况，从而起到预警作用。

设置 GaussDB(for MySQL)的告警规则，包括设置告警规则名称、服务、维度、监控对象、监控指标、告警阈值、监控周期和是否发送通知等参数。

#### 创建实例告警规则

步骤 1 登录管理控制台。

步骤 2 选择“管理与部署 > 云监控服务 CES”，进入“云监控服务”信息页面。

步骤 3 在左侧导航栏选择“云服务监控 > 云数据库 GaussDB(for MySQL)”。

步骤 4 选择目标实例，单击实例名称左侧的 ，选择操作列的“创建告警规则”。

图3-7 创建实例告警规则



步骤5 在“创建告警规则”页面，配置相关参数。

1. 配置告警规则的基本信息。

表3-4 配置规则信息

参数	参数说明
名称	系统会随机产生一个名称，用户也可以进行修改。 取值样例：alarm-b6al
描述	告警规则描述（此参数非必填项）。

2. 配置告警内容参数。

表3-5 配置告警内容参数

参数	参数说明
触发规则	根据需要可选择关联模板、导入已有模板或自定义创建。 <ul style="list-style-type: none"> <li>• 选择“关联模板”后，所关联模板内容修改后，该告警规则中所包含策略也会跟随修改。</li> <li>• 选择“自定义创建”后，可根据实际需求配置告警策略及告警级别。</li> </ul>
模板	选择需要导入的模板。 您可以选择系统预置的默认告警模板，或者选择自定义模板。
告警策略	触发告警规则的告警策略。 告警规则内最多可添加 50 条告警策略，若其中一条告警策略达到条件都会触发告警。

3. 配置告警通知参数。

表3-6 配置告警通知

参数	参数说明
----	------

参数	参数说明
发送通知	配置是否发送邮件、短信、HTTP 和 HTTPS 通知用户。
通知方式	根据需要可选择通知组或主题订阅两种方式。
通知组	需要发送告警通知的通知组。
通知对象	需要发送告警通知的对象，可选择“云帐号联系人”或主题。 <ul style="list-style-type: none"> <li>云帐号联系人：注册帐号时的手机和邮箱。</li> <li>主题：创建主题后，您需要单击主题列表操作栏的添加订阅按钮，添加通知方式。</li> </ul>
生效时间	该告警规则仅在生效时间内发送通知消息。 如生效时间为 08:00-20:00，则该告警规则仅在 08:00-20:00 发送通知消息。
触发条件	可以选择“出现告警”、“恢复正常”两种状态，作为触发告警通知的条件。

#### 4. 配置归属企业项目和标签。

表3-7 配置规则信息

参数	参数说明
归属企业项目信息	告警规则所属的企业项目。只有拥有该企业项目权限的用户才可以查看和管理该告警规则。
标签	使用标签可以方便识别和管理您拥有的数据库服务资源。

步骤6 单击“立即创建”，告警规则创建完成。

----结束

### 3.11.4 设置秒级监控

用于提高监控指标的瞬时精确值，GaussDB(for MySQL)支持秒级监控，包括 1 秒监控和 5 秒监控。


#### 计费说明

GaussDB(for MySQL)为您免费提供监控频率为 60 秒/次的监控服务，开通秒级监控会产生额外费用，且计费方式为按需计费（每小时扣费一次），不足一小时按照实际使用时长收费。



## 开启秒级监控

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面，选择指定的 GaussDB(for MySQL)实例，单击实例名称。

步骤 5 在左侧导航树，单击“高级运维”。

步骤 6 在“高级运维”页面，选择“实时监控”页签，单击“秒级监控”后的。

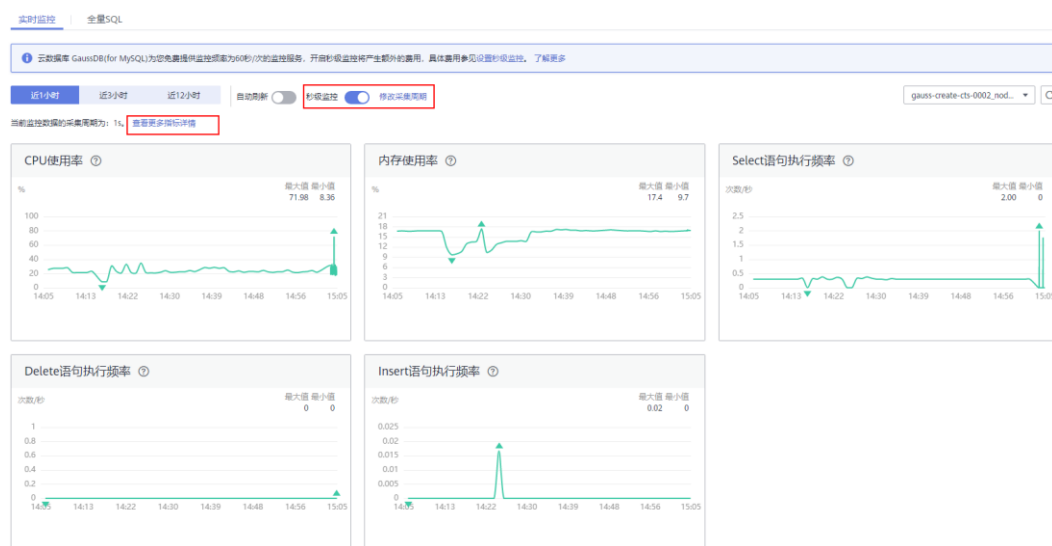
步骤 7 在开启秒级弹框中选择采集周期，单击“确定”，开启秒级监控。

步骤 8 查看秒级监控指标数据。当前秒级监控支持设置的监控指标有：CPU 使用率、内存使用率、Select 语句执行频率、Delete 语句执行频率、Insert 语句执行频率。

可单击“查看更多指标详情”，查看更多指标。

根据业务需要，如需修改采集周期，请单击“修改采集周期”进行修改。

图3-8 秒级监控指标



----结束

## 关闭秒级监控

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面，选择指定的 GaussDB(for MySQL)实例，单击实例名称。

步骤 5 在左侧导航树，单击“高级运维”。


步骤 6 在“高级运维”页面，选择“实时监控”页签，单击“秒级监控”后的。

步骤 7 在关闭秒级监控弹框中选择“是”，关闭秒级监控。

----结束

## 修改采集周期

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面，选择指定的 GaussDB(for MySQL)实例，单击实例名称。

步骤 5 在左侧导航树，单击“高级运维”。

步骤 6 在“高级运维”页面，选择“实时监控”页签，单击“秒级监控”后的“修改采集周期”。

步骤 7 在修改采集周期弹框中选择采集周期，单击“是”，修改采集周期。

----结束

## 3.12 日志管理

### 3.12.1 查看日志的方法

您可以通过云数据库 GaussDB(for MySQL)管理控制台查看数据库日志，相关操作请参见 [查看错误日志](#)和[查看慢日志](#)。

### 3.12.2 查看错误日志

GaussDB(for MySQL)的日志管理功能支持查看数据库级别的日志，包括数据库运行的错误信息，以及运行较慢的 SQL 查询语句，有助于您分析系统中存在的问题。

#### 查看日志明细

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面，选择目标实例，单击实例名称，进入实例的“基本信息”页面。

步骤 5 在左侧导航树，单击“日志管理”。

步骤 6 在“错误日志”页签，选择节点，查看节点对应的错误日志详情。

- 可查看对应节点在不同时间段的错误日志。
- 可查看以下级别的错误日志：
  - ALL
  - INFO
  - WARNING
  - ERROR
  - FATAL
  - NOTE

----结束

### 3.12.3 查看慢日志

#### 操作场景


慢日志用来记录执行时间超过当前慢日志阈值“long\_query\_time”（默认是 10 秒）的语句，您可以通过查询慢日志的日志明细、统计分析情况，查找出执行效率低的语句，进行优化。

GaussDB(for MySQL)支持以下执行语句类型：

- SELECT
- INSERT
- UPDATE
- DELETE
- CREATE
- ALTER
- DROP

#### 查看日志明细

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面，选择目标实例，单击实例名称，进入“基本信息”页面。


步骤 5 在左侧导航树，单击“日志管理”。

步骤 6 选择“慢日志”页签，查看慢 SQL 语句的详细信息。

步骤 7 查看不同节点和不同 SQL 语句类型的慢日志记录。

单击页面右上角的下拉列表，可以分别查看不同节点和不同 SQL 语句类型的慢日志记录。

慢日志功能支持查看的 SQL 语句类型：SELECT、INSERT、UPDATE、DELETE、CREATE、ALTER、DROP。

单击页面右上角，查看某一段时间内的慢日志记录。

----结束

## 慢日志明文显示

### 说明

通过管理控制台开启慢日志明文显示开关后，开启后暂不支持关闭。

默认 SQL 执行语句脱敏状态，开启慢日志明文显示期间抓取的日志中，SQL 执行语句明文显示。


明文显示的日志 30 天后会自动删除，如果实例删除，相关日志也同步删除。

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的，选择区域和项目。

步骤 3 在页面左上角单击，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面，选择目标实例，单击实例名称，进入“基本信息”页面。

步骤 5 在左侧导航栏单击“日志管理”，在“慢日志”页签下，单击“开启慢日志明文显示”右侧开关。

步骤 6 在“开启慢日志明文显示”弹出框中，单击“是”，开启慢日志明文显示功能。

----结束

## 3.12.4 设置全量 SQL

当您开启全量 SQL 功能，系统会将所有的 SQL 文本内容进行存储，以便进行分析。

GaussDB(for MySQL)默认关闭全量 SQL 功能，本章节指导您如何打开、关闭全量 SQL 功能。

### 开启全量 SQL



步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的，选择区域和项目。

步骤 3 在页面左上角单击，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面，选择指定的 GaussDB(for MySQL)实例，单击实例名称，进入实例的“基本信息”页签。

步骤 5 在左侧导航树，单击“高级运维”。

步骤 6 在“高级运维”页面，选择“全量 SQL”页签，单击  设置为 ，并在弹出框中勾选“已阅读并同意本服务对您的数据进行存储”，单击“确定”，完成开启全量 SQL。


#### 说明

保留天数默认为 1 天，到期后自动删除。

----结束



## 关闭全量 SQL

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面，选择指定的 GaussDB(for MySQL)实例，单击实例名称，进入实例的“基本信息”页签。

步骤 5 在左侧导航树，单击“高级运维”。选择“全量 SQL”页签，单击  设置为 ，关闭该实例的全量 SQL。

----结束


## 3.13 任务中心

### 3.13.1 查看任务

您可以通过“任务中心”查看任务执行进度和结果，并进行管理。


#### 操作步骤

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“任务中心”页面，选择目标任务，查看任务信息。

- 通过任务名称/任务 ID、实例名称/ID 确定目标任务，或在右上角的“全部任务类型”的下拉列表中，输入任务名称来确定目标任务。
- 单击页面右侧的 ，查看某一段时间内的任务执行进度和状态，默认时长为一周。  
任务保留时长最多为 30 天。

- 系统支持查看以下状态的任务：
  - 执行中
  - 完成
  - 失败

----结束

### 3.13.2 删除任务


对于不再需要展示的任务，您可以通过“任务中心”进行任务记录的删除。删除任务仅删除记录，不会删除数据库实例或者停止正在执行中的任务。

#### 须知

删除任务将无法恢复，请谨慎操作。

#### 操作步骤

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“任务中心”页面，选择目标任务，单击操作列的“删除”，在弹出框中单击“确定”，删除任务。

GaussDB(for MySQL)支持删除以下状态的任务：

- 完成
- 失败

----结束

## 3.14 标签


#### 操作场景

标签管理服务（Tag Management Service，TMS）用于用户在云平台，通过统一的标签管理各种资源。TMS 服务与各服务共同实现标签管理能力，TMS 提供全局标签管理能力，各服务维护自身标签管理。

- 建议您先在 TMS 系统中设置预定义标签。
- 标签由“键”和“值”组成，每个标签中的一个“键”只能对应一个“值”。
- 每个实例最多支持 10 个标签配额。

## 添加标签

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面，选择指定的实例，单击实例名称，进入实例的“基本信息”页签。

步骤 5 在左侧导航栏，单击“标签”，单击“添加标签”，在“添加标签”弹出框中，输入标签的键和值，单击“确定”。


- 标签的键不能为空且必须唯一，长度为 1~36 个字符，只能包含英文字母、中文、数字、中划线和下划线。
- 标签的值可以为空字符串，长度为 0~43 个字符，只能包含英文字母、中文、数字、中划线、下划线和英文句点。

步骤 6 添加成功后，您可在当前实例的所有关联的标签集合中，查询并管理自己的标签。

----结束

## 编辑标签

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面，选择指定的实例，单击实例名称。

步骤 5 在左侧导航栏，单击“标签”，选择需要编辑的标签，单击“编辑”，在“编辑标签”弹出框中修改标签值，单击“确定”。


- 编辑标签时，不能修改标签的键，只能修改标签的值。
- 标签的值可以为空字符串，长度为 0~43 个字符，只能包含英文字母、中文、数字、中划线、下划线和英文句点。

步骤 6 编辑成功后，您可在当前实例的所有关联的标签集合中，查询并管理自己的标签。

----结束

## 删除标签

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面，选择指定的实例，单击实例名称。

步骤 5 在左侧导航栏，单击“标签”，选择需要删除的标签，单击“删除”，在“删除标签”弹出框中单击“是”。

步骤 6 删除成功后，该标签将不再显示在实例的所有关联的标签集合中。

----结束



# 4 常见问题

## 4.1 产品咨询

### 4.1.1 使用 GaussDB(for MySQL)要注意些什么

1. 实例的操作系统，对用户都不可见，这意味着，只允许用户使用应用程序访问数据库对应的 IP 地址和端口。
2. 对象存储服务（Object Storage Service，简称 OBS）上的备份文件以及 GaussDB(for MySQL)服务使用的弹性云服务器（Elastic Cloud Server，简称 ECS），都对用户不可见，它们只对 GaussDB(for MySQL)服务的后台管理系统可见。
3. 查看实例列表时请确保与实例选择的区域一致。
4. 申请 GaussDB(for MySQL)后，您还需要做什么。

申请 GaussDB(for MySQL)实例后，您不需要进行数据库的基础运维（比如高可用、安全补丁等），但是您还需要重点关注以下事情：

- a. GaussDB(for MySQL)实例的 CPU、内存等资源是否足够，如果资源不足需及时变更规格。
- b. GaussDB(for MySQL)实例的数据存储空间是否足够，如资源不足需及时扩容。（超出时会自动扩容，但超出部分是按需收费，价格比自行扩容贵）。
- c. GaussDB(for MySQL)实例是否存在性能问题，是否有大量的慢 SQL，SQL 语句是否需要优化，是否有多余的索引或者缺失的索引等。

### 4.1.2 为何使用了 GaussDB(for MySQL)后网站登录较慢

推荐您做如下两个处理：

- 通过 GaussDB(for MySQL)服务的管理控制台查看 GaussDB(for MySQL)实例的性能情况。
- 与应用程序有很大关系，使用命令查看当前数据库连接状态，比较本地数据库和 GaussDB(for MySQL)的差异。

### 4.1.3 GaussDB(for MySQL)如何自动进行故障切换

创建 GaussDB(for MySQL)实例时，除主节点外，默认创建了一个只读节点。当主节点故障时，系统会自动切换到只读节点，只读节点提升为主节点，原来故障的主节点也会自动恢复为只读节点。

### 4.1.4 GaussDB(for MySQL)是否支持计算存储分离

GaussDB(for MySQL)使用了计算和存储分离的设计思想，在高可用、备份恢复和升级扩展等方面，给您带来了全方位提升的体验。

## 4.2 数据库连接

### 4.2.1 外部服务器能否访问 GaussDB(for MySQL)数据库

#### 开通公网访问的实例

对于开通公网访问功能的 GaussDB(for MySQL)实例，可以通过外网进行访问。

具体请参见：

[通过公网连接 GaussDB\(for MySQL\)实例](#)

#### 未开通公网访问的实例

- 在虚拟私有云中开通虚拟专用网络（Virtual Private Network，简称 VPN），通过虚拟专用网络连接 GaussDB(for MySQL)。
- 将 GaussDB(for MySQL)与弹性云服务器创建在同一个虚拟专用网络下，通过弹性云服务器来访问 GaussDB(for MySQL)。

具体请参见：

[通过内网连接 GaussDB\(for MySQL\)实例](#)

### 4.2.2 GaussDB(for MySQL)数据库连接数满的排查思路

数据库连接数表示应用程序可以同时连接到数据库的数量，与您的应用程序或者网站能够支持的最大用户数没有关系。

数据库连接数过多，可能会导致业务侧无法正常连接，也会导致实例全量备份和增量备份失败，影响业务的正常使用。

#### 排查思路

1. 请及时排查业务侧连接是否有效，优化实例连接，释放不必要的连接。
2. 规格偏小，请对数据库进行规格扩容。
3. 云监控服务目前可以监控数据库 CPU、内存、磁盘、连接数等指标，并且设置告警策略，出现告警时可以提前识别风险。具体请参考《云监控服务用户指南》。

## 解决方法

1. 通过内网连接数据库实例。用内网连接，不会出现因为带宽等原因的拥塞。  
具体请参见：  
[通过公网连接 GaussDB\(for MySQL\)实例](#)
2. 通过控制台设置参数 `innodb_adaptive_hash_index=off`，关闭自适应 hash 索引，减少锁等待。  
参数修改具体请参见[编辑参数模板](#)。
3. 优化慢查询。

### 4.2.3 GaussDB(for MySQL)数据库实例支持的最大数据连接数是多少

GaussDB(for MySQL)服务对此未做限制，取决于数据库引擎参数的默认值和取值范围，例如 GaussDB(for MySQL)引擎的 `max_connections` 和 `max_user_connections` 参数，用户可在参数模板自定义。

#### 修改最大连接数

连接数支持在线修改，具体方法请参见[编辑参数模板](#)。

支持命令方式修改最大连接数。

1. 执行以下命令，查看最大连接数。  
**show global variables like 'max\_connections';**
2. 在 `my.cnf` 文件中修改 `mysqld` 下参数 `max_connections` 的值。  
**[mysqld]**  
**max\_connections = 1000**

#### 关于 max\_connections

`max_connections`：允许同时连接的客户端总数。如果设定值为 `default`，表示该参数和数据库实例的内存（单位：GB）相关，计算公式如下：

`max_connections` 上限估计数值 = 节点可用内存 / 单个连接预估占用内存。

- 节点可用内存=总内存 - Buffer Pool 占用内存 - 1GB（`mysqld` 进程/操作系统/监控程序等）。
- 单个连接预估占用内存（`single_thread_memory`）= `thread_stack`（256KB）+ `binlog_cache_size`（32KB）+ `join_buffer_size`（256KB）+ `sort_buffer_size`（256KB）+ `read_buffer_size`（128KB）+ `read_rnd_buffer_size`（256KB）= 大约 1MB。

不同内存规格配置的默认 `max_connections` 值如下表所示。

表4-1 不同内存规格配置的默认 `max_connections` 值

内存(GB)	连接数
512	100000

内存(GB)	连接数
384	80000
256	60000
128	30000
64	18000
32	10000
16	5000
8	2500
4	1500
2	800

#### 4.2.4 ECS 无法连接到 GaussDB(for MySQL)实例的原因

遇到该问题，参考以下步骤排查解决。

**步骤 1** 先确认弹性云服务器和 GaussDB(for MySQL)实例是否在同一个虚拟私有云下。

- 如果在，执行**步骤 2**。
- 如果不在，需要重新创建弹性云服务器实例，使之和 GaussDB(for MySQL)实例在同一个虚拟私有云下。

**步骤 2** 查看弹性云服务器实例是否添加安全组。

- 如果有，检查安全组的配置规则是否满足要求。  
请参见“[创建实例](#)”中的“安全组”的描述，然后执行**步骤 3**。
- 如果没有，从弹性云服务器的实例详情页面，进入虚拟私有云页面，选择“安全组”，添加安全组。

**步骤 3** 在弹性云服务器上，测试是否可以正常连接到 GaussDB(for MySQL)实例地址的端口。

GaussDB(for MySQL)主备版实例的默认端口为 3306。

```
telnet <连接地址> {端口号}
```

- 如果可以通信，说明网络是正常的。
- 如果端口不通，请联系售后技术支持协助排查。

----结束

#### 4.2.5 如何通过 JDBC 连接 MySQL 数据库

通过 JDBC 连接实例的方式有无需下载 SSL 证书连接和用户下载 SSL 证书连接两种，其中使用 SSL 证书连接通过了加密功能，具有更高的安全性。GaussDB(for MySQL)新

实例默认开启 SSL 数据加密，SSL 连接实现了数据加密功能，但同时会增加网络连接响应时间和 CPU 消耗，请评估对业务的性能影响，根据需要进行设置。

## 前提条件

用户需要具备以下技能：

- 熟悉计算机基础知识。
- 了解 java 编程语言。
- 了解 JDBC 基础知识。

## 使用 SSL 证书连接

使用 SSL 证书连接实例，即通过证书校验并连接数据库。


### 说明

以下提供的方式不适用于数据库用户的 ssl\_type 为 x509 的认证方式。

执行命令查看当前用户的 ssl\_type 值：

```
select ssl_type from mysql.user where user = 'xxx';
```

步骤 1 下载 CA 证书或捆绑包。

1. 在“实例管理”页面，单击实例名称进入“基本信息”页面。
2. 在“数据库信息”区域，单击“SSL”开关右侧的.

步骤 2 使用 keytool 工具通过 CA 证书生成 truststore 文件。

```
<keytool 工具的安裝路径> ./keytool.exe -importcert -alias <MySQLCACert> --file
<ca.pem> -keystore <truststore_file> -storepass <password>
```

表4-2 变量说明

变量	说明
<keytool 工具的安裝路径>	请替换为 JDK 或 JRE 安装路径的 bin 目录，例如 C:\Program Files (x86)\Java\jdk-11.0.7\bin。
<MySQLCACert>	请设置 truststore 文件的名称。建议设置为具有业务意义的名称，便于后续识别。
<ca.pem>	请替换为步骤 1 中下载解压后 CA 证书的名称，例如 ca.pem。
<truststore_file>	请设置 truststore 文件的存放路径。
<password>	请设置 truststore 文件的密码。

代码示例（使用 JDK 安装路径下的 keytool 工具生成 truststore 文件）：

```
Owner: CN=MySQL_Server_8.0.22_Auto_Generated_CA_Certificate
Issuer: CN=MySQL_Server_8.0.22_Auto_Generated_CA_Certificate
Serial number: 1
Valid from: Thu Feb 16 11:42:43 EST 2017 until: Sun Feb 14 11:42:43 EST 2027
```

```

Certificate fingerprints:
    MD5: 18:87:97:37:EA:CB:0B:5A:24:AB:27:76:45:A4:78:C1
    SHA1: 2B:0D:D9:69:2C:99:BF:1E:2A:25:4E:8D:2D:38:B8:70:66:47:FA:ED

SHA256:C3:29:67:1B:E5:37:06:F7:A9:93:DF:C7:B3:27:5E:09:C7:FD:EE:2D:18:86:F4:9C:40:D
8:26:CB:DA:95: A0:24
    Signature algorithm name: SHA256withRSA Subject Public Key Algorithm: 2048-bit
RSA key
    Version: 1
    Trust this certificate? [no]: y
    Certificate was added to keystore
    
```

步骤 3 通过 JDBC 连接 MySQL 数据库，代码中的 JDBC 链接格式如下：

```

jdbc:mysql://<instance_ip>:<instance_port>/<database_name>?
requireSSL=<value1>&useSSL=<value2>&verifyServerCertificate=<value3>&trustCertificateKeyStoreUrl=file:
<truststore_file>&trustCertificateKeyStorePassword=<password>
    
```

表4-3 参数说明

参数	说明
<instance_ip>	请替换为实例的 IP 地址。 说明 <ul style="list-style-type: none"> <li>• 如果通过弹性云服务器连接，“instance_ip”是实例的“读写内网地址”。您可以在该实例“基本信息”页面的“网络信息”区域查看。</li> <li>• 如果通过公网连接，“instance_ip”为该实例已绑定的“弹性公网 IP”，即读写公网地址。您可以在该实例“基本信息”页面的“网络信息”区域查看。</li> </ul>
<instance_port>	请替换为实例的数据库端口，默认为 3306。 说明 您可以在该实例“基本信息”页面的“网络信息”区域查看。
<database_name>	请替换为连接实例使用的数据库名，默认为 mysql。
<value1>	requireSSL 的值，用于设置服务端是否支持 SSL 连接。取值如下： <ul style="list-style-type: none"> <li>• true：支持。</li> <li>• false：不支持。</li> </ul> 说明 requireSSL 与其他连接参数、sslMode 之间的关系请参考表 4-4。
<value2>	useSSL 的值，用于设置客户端是否使用 SSL 连接服务端。取值如下： <ul style="list-style-type: none"> <li>• true：使用。</li> <li>• false：不使用。</li> </ul> 说明 useSSL 与其他连接参数、sslMode 之间的关系请参考表 4-4。
<value3>	verifyServerCertificate 的值，客户端是否校验服务端的证书。取

参数	说明
	值如下： <ul style="list-style-type: none"> <li>• true: 校验。</li> <li>• false: 不校验。</li> </ul> 说明 verifyServerCertificate 与其他连接参数、sslMode 之间的关系请参考表 4-4。
<truststore_file>	请替换为步骤 2 中为 truststore 文件设置的存储路径。
<password>	请替换为步骤 2 中为 truststore 文件设置的密码。

表4-4 连接参数与 SSLMode 的关系说明

useSSL	requireSSL	verifyServerCertificate	sslMode
false	不涉及	不涉及	DISABLED
true	false	false	PREFERRED
true	true	false	REQUIRED
true	不涉及	true	VERIFY_CA

代码示例（连接 MySQL 数据库的 java 代码）：

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.sql.SQLException;

public class JDBCtest {
    static final String USER = "xxx";
    static final String PASS = "xxx";

    public static void main(String[] args) {
        Connection conn = null;
        Statement stmt = null;

        String url = "jdbc:mysql://<instance_ip>:<instance_port>/<database_name>?
requireSSL=true&useSSL=true&verifyServerCertificate=true&trustCertificateKeyStoreUr
l=file:
<truststore_file>&trustCertificateKeyStorePassword=<password>";

        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            conn = DriverManager.getConnection(url, USER, PASS);
        }
    }
}
```

```

stmt = conn.createStatement();
String sql = "show status like 'ssl%'";
ResultSet rs = stmt.executeQuery(sql);

int columns = rs.getMetaData().getColumnCount();
for (int i = 1; i <= columns; i++) {
    System.out.print(rs.getMetaData().getColumnName(i));
    System.out.print("\t");
}

while (rs.next()) {
    System.out.println();
    for (int i = 1; i <= columns; i++) {
        System.out.print(rs.getObject(i));
        System.out.print("\t");
    }
}
rs.close();
stmt.close();
conn.close();
} catch (SQLException se) {
    se.printStackTrace();
} catch (Exception e) {
    e.printStackTrace();
} finally {
    // release resource ....
}
}
}

```

----结束

## 无证书连接

### 📖 说明

该方式不对服务端进行证书校验，用户无需下载 SSL 证书。

步骤 1 通过 JDBC 连接 GaussDB(for MySQL)数据库实例，代码中的 JDBC 链接格式如下：

```
jdbc:mysql://<instance_ip>:<instance_port>/<database_name>?useSSL=false
```

表4-5 变量说明

变量	说明
<instance_ip>	<p>请替换为实例的 IP 地址。</p> <p>说明</p> <ul style="list-style-type: none"> <li>• 如果通过弹性云服务器连接，“instance_ip”是实例的“读写内网地址”。您可以在该实例“基本信息”页面的“网络信息”区域查看。</li> <li>• 如果通过公网连接，“instance_ip”为该实例已绑定的“弹性公网 IP”，即读写公网地址。您可以在该实例“基本信息”页面的“网络信息”区域查看。</li> </ul>
<instance_port>	请替换为实例的数据库端口，默认为 3306。



变量	说明
	说明 您可以在该实例“基本信息”页面的“网络信息”区域查看。
<database_name>	请替换为连接实例使用的数据库名，默认为mysql。

代码示例（连接 MySQL 数据库的 java 代码）：

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;

public class MyConnTest {
    final public static void main(String[] args) {
        Connection conn = null;
        // set sslmode here.
        // no ssl certificate, so do not specify path.
        String url = "jdbc:mysql://192.168.0.225:3306/my_db_test?useSSL=false";
        try {
            Class.forName("com.mysql.jdbc.Driver");
            conn = DriverManager.getConnection(url, "root", "password");
            System.out.println("Database connected");

            Statement stmt = conn.createStatement();
            ResultSet rs = stmt.executeQuery("SELECT * FROM mytable WHERE
columnfoo = 500");
            while (rs.next()) {
                System.out.println(rs.getString(1));
            }
            rs.close();
            stmt.close();
            conn.close();
        } catch (Exception e) {
            e.printStackTrace();
            System.out.println("Test failed");
        } finally {
            // release resource ....
        }
    }
}
```

----结束

## 相关问题

- 问题描述

当您使用 8.0 及以上版本的 JDK 通过 SSL 证书模式连接 MySQL 时，报如下类似错误：

```
javax.net.ssl.SSLHandshakeException: No appropriate protocol (protocol is
disabled or
cipher suites are inappropriate)
```

```
    at sun.security.ssl.HandshakeContext.<init> (HandshakeContext.java:171)
~[na:1.8.0_292]
    at
sun.security.ssl.ClientHandshakeContext.<init> (ClientHandshakeContext.java:98)
~
[na:1.8.0_292]
    at sun.security.ssl.TransportContext.kickstart (TransportContext.java:220) ~
[na:1.8.0_292]
    at sun.security.ssl.SSLSocketImpl.startHandshake (SSLSocketImpl.java:428) ~
[na:1.8.0_292]
    at
com.mysql.cj.protocol.ExportControlled.performTlsHandshake (ExportControlled.java:316) ~
[mysql-connector-java-8.0.17.jar:8.0.17]
    at
com.mysql.cj.protocol.StandardSocketFactory.performTlsHandshake (StandardSocketFactory.java:188) ~[mysql-connector-java8.0.17.jar:8.0.17]
    at
com.mysql.cj.protocol.a.NativeSocketConnection.performTlsHandshake (NativeSocketConnection.java:99) ~[mysql-connector-java8.0.17.jar:8.0.17]
    at
com.mysql.cj.protocol.a.NativeProtocol.negotiateSSLConnection (NativeProtocol.java:331) ~
[mysql-connector-java8.0.17.jar:8.0.17]
... 68 common frames omitted
```

- 解决方法

您可以在[步骤 3](#)中的代码链路中，根据客户端使用的 Jar 包指定对应参数值进行连接。示例如下：

- mysql-connector-java-5.1.xx.jar

```
jdbc:mysql://<instance_ip>:<instance_port>/<database_name>?
requireSSL=true&useSSL=true&verifyServerCertificate=true&trustCertificateKeyStoreUrl=file:
<truststore_file>&trustCertificateKeyStorePassword=<password>&
enabledTLSProtocols=TLSv1.2
```

- mysql-connector-java-8.0.xx.jar

```
jdbc:mysql://<instance_ip>:<instance_port>/<database_name>?
requireSSL=true&useSSL=true&verifyServerCertificate=true&trustCertificateKeyStoreUrl=file:
<truststore_file>&trustCertificateKeyStorePassword=<password>&
tlsVersions =TLSv1.2
```

## 4.2.6 如何创建和连接 ECS

1. 创建弹性云服务器，请参见《弹性云服务器用户指南》。

- 该弹性云服务器用于连接 GaussDB(for MySQL)实例，需要与目标实例处于同一虚拟私有云内。
- 正确配置安全组，使得弹性云服务器可以通过“连接地址”访问 GaussDB(for MySQL)实例。

2. 连接弹性云服务器，请参见《弹性云服务器用户指南》中“登录弹性云服务器”的内容。

## 4.2.7 客户端问题导致连接失败

客户端问题导致连接 GaussDB(for MySQL)失败，可以从以下几个方面检查。

1. 弹性云服务器的安全策略
 

对于 Windows 平台，可检查 Windows 的安全策略是否开放 GaussDB(for MySQL) 端口。对于 Linux 平台，可使用 iptables 检查防火墙及端口的放行情况。
2. 应用配置错误
 

常见的有连接地址写错、端口参数配置错误和 JDBC 等的连接参数配置错误。
3. 用户名或密码错误
 

如果连接数据库时出现类似如下错误，请检查用户名或密码是否正确。

  - [Warning] Access denied for user 'username'@'yourIp' (using password: NO)
  - [Warning] Access denied for user 'username'@'yourIp' (using password: YES)

### 说明

如问题仍未解决，请联系售后技术支持。

## 4.2.8 绑定公网 IP 后无法 ping 通的解决方案

### 场景排查

1. 检查安全组规则。
2. 检查“网络 ACL”规则。
3. 相同区域主机进行 ping 测试。

### 解决方案



1. 检查安全组规则。
  - a. 登录管理控制台。
  - b. 单击管理控制台左上角的 ，选择区域和项目。
  - c. 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。
  - d. 在“实例管理”页面，选择目标实例，单击实例名称，进入实例的“基本信息”页面。
  - e. 在“网络信息”模块的“内网安全组”处，单击安全组名称，进入安全组页面。
  - f. 检查弹性云服务器网卡对应的安全组是否放通了“入方向”的“ICMP”规则。

表4-6 安全组规则

放行	类型	协议和端口	原地址
----	----	-------	-----

放向	类型	协议和端口	原地址
入方向	IPv4	Any: Any	0.0.0.0/0 0.0.0.0/0 表示所有 IP 地址
入方向	IPv4	ICMP: Any	0.0.0.0/0 0.0.0.0/0 表示所有 IP 地址

2. 检查“网络 ACL”规则。
  - a. 排查“网络 ACL”是否放通。查看“网络 ACL”状态，查看当前是开启状态还是关闭状态。
  - b. 检查“弹性公网 IP”绑定的网卡是否在“网络 ACL”关联的子网下。
  - c. 若“网络 ACL”为“开启”状态，需要添加 ICMP 放通规则进行流量放通。

#### 说明

需要注意“网络 ACL”的默认规则是丢弃所有出入方向的包，若关闭“网络 ACL”后，其默认规则仍然生效。

3. 相同区域主机进行 ping 测试。  
在相同区域的弹性云服务器去 ping 没有 ping 通的弹性公网 IP，如果可以正常 ping 通说明虚拟网络正常，请联系客服获取技术支持。


## 4.2.9 测试连通性失败，如何排查

### 场景排查

1. 排查安全组规则。
2. 排查网络 ACL。
3. 排查弹性云服务器内部网卡信息。
4. 排查不通端口。

### 解决方案

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面，单击目标实例名称，进入“基本信息”页面，在“连接信息”模块查看 GaussDB(for MySQL)实例的虚拟私有云。

步骤 5 查看加入分布式事务的 GaussDB(for MySQL)或 ECS 是否有相同 VPC。

- 相同，请查看《虚拟私有云用户指南》中的网络连接类常见问题排查。
- 不相同

- 公网访问：需要通过 EIP 建立连接。请参考[绑定弹性公网 IP](#) 为 GaussDB(for MySQL)实例绑定 EIP。
- 内网访问：为两个不同的虚拟私有云建立对等连接，实现内网互通。
- 将 ECS 的虚拟私有云切换为与 GaussDB(for MySQL)相同的虚拟私有云。

----结束

## 4.2.10 GaussDB(for MySQL)跨地域内网能访问吗

跨地域内网默认不能访问，不同区域的云服务之间内网互不相通。您可以通过云连接或者 VPN 打通网络实现内网访问。

- 云连接：对于不同区域的 VPC，不区分是否同一帐号，都可以互连，跨区域连接实现全球云上网络。
- 虚拟专用网络 VPN：基于 Internet 使用加密隧道将不同区域的 VPC 连接起来。具备成本低、配置简单、即开即用等优点。但它的网络质量依赖 Internet。

## 4.2.11 GaussDB(for MySQL)实例连接数过多，存在什么隐患

GaussDB(for MySQL)连接数过多，可能会导致业务侧无法正常连接，也会导致实例全量备份和增量备份失败，影响业务的正常使用。

### 解决方案

1. 请及时排查业务侧连接是否有效，优化实例连接，释放不必要的连接。
2. 云监控服务目前可以监控数据库 CPU、内存、磁盘、连接数等指标，并且设置告警策略，出现告警时可以提前识别风险。

## 4.2.12 ECS 和 GaussDB(for MySQL)部署在同一区域的不同的 VPC 内，网络不通怎么办

同一区域内，不同 VPC 下 GaussDB(for MySQL)内网是不通的。且创建实例后不支持切换 VPC。

### 解决方法：

1. 为两个不同的虚拟私有云建立对等连接，实现内网互通。
2. 将 ECS 的虚拟私有云切换为与 GaussDB(for MySQL)相同的虚拟私有云。

## 4.2.13 如何查看当前时间所有连接数据库的 IP

在数据库上执行以下 SQL 语句，统计 IP 连接数。

```
SELECT substring_index(host, ':', 1) AS host_name, state, count(*) FROM information_schema.processlist GROUP BY state, host_name;
```

## 4.3 安装客户端

### 4.3.1 如何安装 MySQL 客户端

MySQL 官网提供了针对不同操作系统的客户端安装包，以 Red Hat Linux 为例。单击[此处](#)下载 8.0 的最新版本，单击[此处](#)下载其他归档版本。下面将介绍如何获取相应安装包及完成安装。

#### 操作步骤

##### 步骤 1 获取安装包。

在下载页面找到对应版本[链接](#)，以 `mysql-community-client-8.0.21-1.el6.x86_64` 为例，打开页面后，即可下载安装包。

图4-1 下载

#### MySQL Community Downloads

Login Now or Sign Up for a free account.

An Oracle Web Account provides you with the following advantages:

- Fast access to MySQL software downloads
- Download technical White Papers and Presentations
- Post messages in the MySQL Discussion Forums
- Report and track bugs in the MySQL bug system

**Login »**  
using my Oracle Web account

**Sign Up »**  
for an Oracle Web account

MySQL.com is using Oracle SSO for authentication. If you already have an Oracle Web account, click the Login link. Otherwise, you can sign up for a free account by clicking the Sign Up link and following the instructions.

**No thanks, just start my download.**

#### 📖 说明

通过红框所在的[链接](#)，可直接下载安装包。

##### 步骤 2 将安装包上传到弹性云服务器。

#### 📖 说明

创建弹性云服务器时，要选择操作系统，例如 Redhat6.6，并为其绑定 EIP（Elastic IP，弹性公网 IP）。然后使用远程连接工具将安装包上传到弹性云服务器，再用 PuTTY 连接到弹性云服务器。

##### 步骤 3 执行以下命令安装 MySQL 客户端。

```
sudo rpm -ivh mysql-community-client-8.0.21-1.el6.x86_64.rpm
```

### 说明

- 如果安装过程中报 conflicts，可增加 replacefiles 参数重新安装，如下：

```
rpm -ivh --replacefiles mysql-community-client-8.0.21-1.el6.x86_64.rpm
```

- 如果安装过程中提示需要安装依赖包，可增加 nodeps 参数重新安装，如下：

```
rpm -ivh --nodeps mysql-community-client-8.0.21-1.el6.x86_64.rpm
```

----结束

## 4.4 数据库迁移

### 4.4.1 GaussDB(for MySQL)支持导入哪些数据库引擎的数据

- 相同引擎数据库之间数据导入导出，称之为同构型数据库之间数据导入导出。
- 不同引擎数据库之间数据导入导出，称之为异构型数据库之间数据导入导出。例如，从 Oracle 导入数据到 GaussDB(for MySQL)支持的数据库引擎。

异构型数据库之间由于格式不同，不支持直接导入导出。但只要导入导出的格式数据兼容，理论上，导入表数据也是可行的。

异构型数据库之间数据导入导出，一般需要第三方软件，通过数据复制的方式来实现。比如，先使用工具从 Oracle 中，以文本的格式导出表记录，然后利用 Load 语句导入到支持的数据库引擎。

## 4.5 数据库权限

### 4.5.1 GaussDB(for MySQL)提供 root 帐号或 super 权限吗

GaussDB(for MySQL)提供的管理员帐号为 root 帐号，具有去除 super、file、shutdown 和 create tablespace 后的最高权限。

大部分的云数据库服务平台，都未给 root 帐号提供 super 权限。因为一旦用户拥有了 super 权限，就可以执行很多管理性的命令，比如 reset master, set global..., kill, reset slave 等，这类操作很有可能导致 GaussDB(for MySQL)出现不可预知的异常和故障。这一点是云服务平台和本地搭建 MySQL 比较大的区别，GaussDB(for MySQL)提供服务化能力，那就需要保证实例的稳定正常运行。

对于客户要求 super 权限的场景，GaussDB(for MySQL)提供了服务化能力，也可以通过其他手段绕过 super 权限的限制。

举例如下：

举例 1：有些用户喜欢登录数据库执行如下命令来修改参数，这在 GaussDB(for MySQL)是被禁止的，您只能通过 GaussDB(for MySQL)界面中的参数修改功能来实现。

```
set global 参数名=参数值;
```

如果您的脚本中包含 set global 命令导致 super 缺失，请删除 set global 命令，通过 console 的参数修改。

举例 2：有些用户执行如下命令报错，这也是因为没有 super 权限导致的，只需要去除 definer='root' 关键字即可。

```
create definer='root'@'%' trigger(procedure)...
```

如果您缺失 super 权限，可以使用 mysqldump 导入数据，请参考[使用 mysqldump 迁移 MySQL 数据](#)导入和导出数据。

## 4.6 数据库性能

### 4.6.1 GaussDB(for MySQL) CPU 使用率高的解决方法

使用云数据库 GaussDB(for MySQL)时，如果您的 CPU 使用率很高或接近 100%，会导致数据读写处理缓慢、无法获取连接、出现报错等，从而影响业务正常运行。

#### 解决方案

1. 通过查看慢 SQL 日志来确定是否存在运行缓慢的 SQL 查询以及各个查询的性能特征（如果有），从而定位查询运行缓慢的原因。  
查询 MySQL 日志，请参见[查看慢日志](#)。
2. 查看 GaussDB(for MySQL)实例的 CPU 使用率指标，协助定位问题。
3. 创建只读节点专门负责查询。减轻主实例负载，分担数据库压力。
4. 多表关联查询时，关联字段要加上索引。
5. 尽量避免用 select\* 语句进行全表扫描，可以指定字段或者添加 where 条件。

### 4.6.2 联合索引设置不当导致慢 SQL 的解决办法

#### 场景描述

业务侧云数据库 GaussDB(for MySQL)实例上以往执行耗时 8 秒的查询，在 11:00 后耗时超过 30 秒。

#### 原因分析

1. 查看查询变慢对应的时间段中，实例 CPU 监控指标并无飙升情况且使用率一直都比较低，因此排除了 CPU 冲高导致查询变慢的可能。
2. 分析对应时间段该实例的慢日志，该 SQL 执行快时其扫描行数为百万级，当 SQL 执行慢时其扫描行数为千万级，与业务确认该表短期内并无大量数据插入，因此推断执行慢是因为未走索引或选错索引。且通过 EXPLAIN 查看该 SQL 的执行计划确实是全表扫描。



图4-2 慢日志

select query_date, sum(queue) queue, sum(server_user_num) serverUserNum, su...	SELECT	1	6.027128 s	0.000105	125	2119000
select query_date, sum(queue) queue, sum(server_user_num) serverUserNum, su...	SELECT	1	5.479857 s	0.000104	123	2085096
select query_date, sum(queue) queue, sum(server_user_num) serverUserNum, su...	SELECT	1	5.288558 s	0.000105	123	2085096
select query_date, sum(queue) queue, sum(server_user_num) serverUserNum, su...	SELECT	1	33.501792 s	0.000064	140	15961077
select query_date, sum(queue) queue, sum(server_user_num) serverUserNum, su...	SELECT	1	34.342761 s	0.000171	140	15961077
select query_date, sum(queue) queue, sum(server_user_num) serverUserNum, su...	SELECT	1	44.536072 s	0.000167	140	15961077
select query_date, sum(queue) queue, sum(server_user_num) serverUserNum, su...	SELECT	1	46.501796 s	0.000095	140	15961077
select query_date, sum(queue) queue, sum(server_user_num) serverUserNum, su...	SELECT	1	33.050387 s	0.000099	139	15944097
select query_date, sum(queue) queue, sum(server_user_num) serverUserNum, su...	SELECT	1	38.523306 s	0.000101	139	15944097
select query_date, sum(queue) queue, sum(server_user_num) serverUserNum, su...	SELECT	1	40.108127 s	0.000090	139	15944097

- 在实例上对该表执行 **SHOW INDEX FROM** 检查三个字段的基数。

图4-3 查看基数

```

***** 3. row *****
Table: [redacted]
Non_unique: 1
Key_name: idx_query_date_channel_group_id
Seq_in_index: 1
Column_name: query_date
Collation: A
Cardinality: 133994
Sub_part: NULL
Packed: NULL
Null: YES
Index_type: BTREE
Comment:
Index_comment:
***** 4. row *****
Table: [redacted]
Non_unique: 1
Key_name: idx_query_date_channel_group_id
Seq_in_index: 2
Column_name: channel
Collation: A
Cardinality: 405333
Sub_part: NULL
Packed: NULL
Null: YES
Index_type: BTREE
Comment:
Index_comment:
***** 5. row *****
Table: [redacted]
Non_unique: 1
Key_name: idx_query_date_channel_group_id
Seq_in_index: 3
Column_name: group_id
Collation: A
Cardinality: 16213328
Sub_part: NULL
Packed: NULL
Null: YES
Index_type: BTREE
    
```

可知基数最小的字段“query\_date”在联合索引的第一位，基数最大的字段“group\_id”在联合索引最后一位，而且原 SQL 包含对“query\_date”字段的范围

查询，导致当索引走到“query\_date”就会停止匹配，后面两个字段已经无序，无法走索引。

所以该 SQL 本质上只能利用到对“query\_date”这一列的索引，而且还有可能因为基数太小，导致优化器成本估计时选择了全表扫描。

业务重新创建了联合索引将“group\_id”字段放在第一位，“query\_date”字段放在最后一位后，查询耗时符合预期。

## 解决方案

1. 查询变慢首先确认是否由于 CPU 利用率达到性能瓶颈导致执行慢。
2. 库表结构设计不合理，索引缺失或索引设置不恰当会导致慢 SQL。
3. 表数据大批量插入删除等操作可能会导致统计信息未能及时更新，建议定期执行 **ANALYZE TABLE** 防止执行计划走错。

## 4.6.3 长事务产生大量临时表导致内存超限的解决办法

### 场景描述

云数据库 GaussDB(for MySQL)实例在 11:30 到 12:27 分内存使用率持续上升，最终触发内存超限。

图4-4 内存使用率



### 原因分析

1. 查看 processlist.log 日志，查询到有两个慢 SQL 与图 4-4 中内存增长的时间比较匹配。

图4-5 查询慢 SQL

```

2022-03-20 12:27:45
20220320 dbankptdb Query 3811 executing select app_ver,prj_name,login_status,sum(case login_status
when 'true' then 1 else 0 end)as success,sum(case login_status when 'false' then 1 else 0 end)as failac
,count(login_status),sum(case login_status when 'true' then 1 else 0 end)) / count(login_status)
from
where login_status is not null
and (date_format(p_4,'%Y-%m-%d') BETWEEN '2022-03-21' AND '2022-03-28')
group by app_ver
20220320 dbankptdb Query 2079 executing select app_ver,prj_name,login_status,sum(case login_status
when 'true' then 1 else 0 end)as success,sum(case login_status when 'false' then 1 else 0 end)as failac
,count(login_status),sum(case login_status when 'true' then 1 else 0 end)) / count(login_status)
from
where login_status is not null
and (date_format(p_4,'%Y-%m-%d') BETWEEN '2022-03-21' AND '2022-03-28')
group by app_ver
    
```

2. 查看慢查询中的表数据量约 90GB，数据行数约 10 亿行，且通过图 4-5 中的执行时间可以看出，两个 SQL 执行了 40~50 分钟，跟监控中内存增长的时间基本一致，确定是临时表内存不受控导致。



### 解决方案

1. 升级实例规格，将内存利用率维持在合理范围，防止业务突增导致实例 OOM。变更实例规格的详细内容请参考。
2. 根据业务实际情况优化慢查询。

## 4.6.4 持锁长事务导致后续业务报等锁超时的解决办法

### 场景描述

业务侧实例上报错误码 1205，等锁超时提示。

MySQL error code MY-001205 (ER\_LOCK\_WAIT\_TIMEOUT): **Lock wait timeout exceeded; try restarting transaction**

### 原因分析

1. 查看监控指标“行锁花费时间”，监控到行锁等待时间较长，说明该系统出现过锁冲突的现象。

监控指标详细内容请参考[查看监控指标](#)。



2. 登录实例，执行如下 SQL，查看系统当前存在的长事务，以及事务持有的行锁信息。

```
select trx_mysql_thread_id, trx_id, trx_state, trx_started, trx_tables_locked,
trx_rows_locked, trx_isolation_level, trx_query, trx_operation_state from
information_schema.innodb_trx order by trx_started;
```

```
mysql> select trx_mysql_thread_id, trx_id, trx_state, trx_tables_locked, trx_rows_locked, trx_isolation_level, trx_query, trx_operation_state from information_schema.innodb_trx order by trx_started;
```

trx_mysql_thread_id	trx_id	trx_state	trx_started	trx_tables_locked	trx_rows_locked	trx_isolation_level	trx_query	trx_operation_state
416	51965	RUNNING	2022-07-28 19:27:55	1	1	READ COMMITTED	NULL	NULL
415	51967	LOCK WAIT	2022-07-29 00:11:03	1	2	READ COMMITTED		fetching rows

- information\_schema.innodb\_trx 表包含了当前 innodb 内部正在运行的事务信息。

- **trx\_started**: 表示事务的开始时间, 用来判断当前事务是否是长事务, 当前时间减去开始时间就是事务的执行时间。
- **trx\_state** : 表示当前事务的状态, 取值如下:
  - **RUNNING**: 运行。
  - **LOCK WAIT**: 等待锁。

#### 说明

如果事务当前的状态是 **LOCK WAIT**, 即表示事务持有行锁。

- **ROLLING BACK**: 正在回滚。
- **COMMITTING**: 正在提交。

## 解决方案

由于持锁长事务长时间未提交或回滚导致后续操作阻塞, 如果持锁长事务已经阻塞了后续的业务, 需要将长事务 **KILL**, 后续业务侧尽量避免持锁长事务。

## 4.7 数据库基本使用

### 4.7.1 主备机分别执行 MATCH AGAINST 语句后结果不一致的原因

**MATCH (columnName) AGAINST ('keywords')**用于检索 MySQL 的 FullText index, 其返回的结果是 keywords 在每一行的 columnName 列中的相关度。主机采用 **persistent** 方式获取统计信息, 备机采用 **transient** 方式获取统计信息, 获取表的统计信息中的行数字段(stat\_n\_rows)会存在一些误差, 导致同一张表的统计信息的行数字段(stat\_n\_rows)在主备是不一样的, 而 **MATCH ... AGAINST ...**的结果, 也就是相关度的计算中使用了 stat\_n\_rows, 导致主备的 **MATCH ... AGAINST ...**的结果不一致。

### 4.7.2 使用 LOAD DATA 导入本地数据

此章节介绍如何通过 **LOAD DATA** 将本地数据导入 GaussDB(forMySQL)。

#### 语法

```
LOAD DATA LOCAL
  INFILE 'file_name'
  [REPLACE | IGNORE]
  INTO TABLE tbl_name
  [CHARACTER SET charset_name]
  [{FIELDS | COLUMNS}
   [TERMINATED BY 'string']
   [[OPTIONALLY] ENCLOSED BY 'char']
  ]
  [LINES
   [TERMINATED BY 'string']
  ]
  [IGNORE number {LINES | ROWS}]
  [(col_name_or_user_var
   [, col_name_or_user_var] ...)]
```

## 参数

- `file_name` : 要导入的本地文件的路径。
- `REPLACE|IGNORE`: 指定遇到重复记录是替换还是忽略。
- `tbl_name` : 要导入的目标表名。
- `CHARACTER SET charset_name`: 指定文件的编码, 建议和 GaussDB(for MySQL) 实例上的编码一致, 否则可能乱码。
- `FIELDS TERMINATED BY 'string'`: 用来指定每列之间的分割符号, 默认值为 `\t`。
- `[OPTIONALLY] ENCLOSED BY 'char'`: 用来忽略数据源字段中的符号。
- `LINES TERMINATED BY 'string'`: 可以指定行之间的换行符, 默认值为 `\n`。

### 说明

有些 windows 上的文本文件的换行符可能为 `\r\n`, 由于是不可见字符, 请仔细检查。

- `IGNORE number LINES` : 设置导入数据时忽略开始的某几行。
- `(column_name_or_user_var,...)` : 设置导入的列, 如果不设置, 默认按照列的顺序来导入数据。
- 更多参数参考 MySQL 的 [load data infile](#) 官方文档说明。其他参数的先后顺序不能乱, 顺序参考[官方说明](#)。

## 标准示例

前提条件:

- 服务端必须开启 `local_infile` 开关, 在目标实例的基本信息页面, 进入参数修改页面里将此参数修改为 `ON`。
- 客户端必须开启 `local-infile` 开关, 在 `my.cnf` 中配置 `local-infile` 或连接数据库使用 `--local-infile=1` 参数。

```
[mysql]
local-infile
```

1. 将本地文件 `qq.txt` 中的数据导入 `test` 表中, `qq.txt` 文件共有 5 行数据, 列分隔符为 `'`, 行分隔符 `\n`, 内容如下:

```
1,a
2,b
3,c
4,d
5,"e"
```

2. 创建对应表 `test`, sql 如下:

```
CREATE TABLE test (
  `id` int NOT NULL,
  `a` varchar(4) NOT NULL,
  PRIMARY KEY (`id`)
);
```

3. 在客户端中执行 `LOAD DATA` 命令将 `qq.txt` 文件中的数据导入到 `test` 表中, 设置字符集为 `utf8`, 忽略数据源字段中的双引号。

```
mysql> LOAD DATA LOCAL INFILE '/data/qq.txt' IGNORE INTO TABLE test CHARACTER
SET 'utf8' FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"' LINES TERMINATED
```

```
BY '\n';
Query OK, 5 rows affected, 1 warning (0.00 sec)
Records: 5 Deleted: 0 Skipped: 0 Warnings: 1

mysql> select * from test;
+----+----+
| id | a |
+----+----+
| 1 | a |
| 2 | b |
| 3 | c |
| 4 | d |
| 5 | e |
+----+----+
5 rows in set (0.00 sec)
```

### 须知

1. 数据导入阶段会在一定程度上影响 GaussDB(for MySQL)实例性能，请选择在业务低峰时间导入。
2. 建议不要同时发起多个 LOAD DATA 请求。多个 LOAD DATA 同时进行，数据高并发写入，表锁竞争以及系统 IO 抢占会影响总体效率，可能会出现 SQL 事务超时现象，导致 LOAD DATA 全部失败。

## 4.8 备份与恢复

### 4.8.1 GaussDB(for MySQL)能够保存多长时间的备份

云数据库 GaussDB(for MySQL)实例的自动备份有效期根据用户设置的备份天数而定。详情请参见[设置自动备份策略](#)。

手动备份没有时间限制，用户可根据需要进行手动删除。详情请参见[创建手动备份](#)。

备份存储在对象存储服务上，不占用您创建的数据库空间。

### 4.8.2 如何清理云数据库 GaussDB(for MySQL)的备份空间

GaussDB(for MySQL)的备份空间中存放的是自动备份、手动备份文件。

- **清理自动备份（全量备份+增量备份）**

自动备份文件不支持手动删除，可通过[修改备份策略](#)调整备份保留天数，超出备份保留天数的已有备份文件会被自动删除。

- **清理手动备份（全量备份）**

手动备份文件支持手动删除，具体请参见[删除手动备份](#)。

### 4.8.3 如何将 GaussDB(for MySQL)数据库备份到弹性云服务器上

您可以通过导出 SQL 语句的方式将数据库备份到弹性云服务器上。弹性云服务器不限制存放哪些数据，但是数据必须符合国家法律法规。您可以在弹性云服务器上存放数据库备份，但不建议将弹性云服务器作为数据库备份空间使用。

强烈推荐使用云数据库 GaussDB(for MySQL)将备份数据存放到专业的对象存储服务上，以获得更高的数据可靠性和服务保障。

### 4.8.4 如何查看 GaussDB(for MySQL)备份空间使用情况

通过管理控制台的“存储/备份空间”模块，可以详细查看备份空间使用情况。

#### 操作步骤



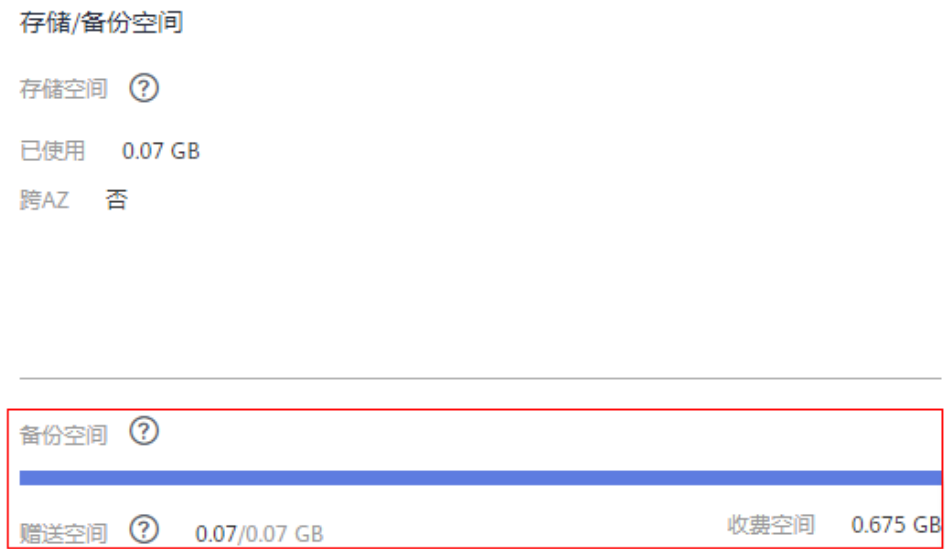
- 步骤 1 登录管理控制台。
- 步骤 2 单击管理控制台左上角的 ，选择区域和项目。
- 步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。
- 步骤 4 在“实例管理”页面，选择指定的实例，单击实例名称。
- 步骤 5 进入“基本信息”页面，在“存储/备份空间”模块查看备份空间使用情况。

图4-6 备份空间



#### 说明

备份空间包含实例数据备份和 binlog 备份。

GaussDB(for MySQL)提供和实例磁盘大小相同的免费存储空间，用于存放您的备份数据。

----结束

## 4.8.5 为什么自动备份会失败

出现自动备份失败可能有以下原因：

1. 网络环境稳定性不足。如网络延时、中断，系统会识别并延时半小时再次执行备份，您也可以及时进行一次手动备份。
2. 系统多任务执行复杂性。如任务等待、中止，系统会识别并延时半小时再次执行备份，您也可以及时进行一次手动备份。
3. 实例状态异常。如实例故障、状态变更中，系统会识别并在状态恢复正常后再次执行备份，您也可以及时进行一次手动备份。
4. 参数修改异常。如修改参数并关联实例后导致实例故障，您可以对比参数修改前后的值是否正确、修改的参数是否存在关联参数需要一并修改、或者尝试恢复默认参数重启实例。
5. 导入数据异常。
6. 如果以上操作仍无法解决问题，请联系技术支持。

## 4.8.6 GaussDB(for MySQL)的备份是如何收费的

云数据库 GaussDB(for MySQL)的备份文件存储在对象存储服务上，不占用用户购买实例的存储空间，GaussDB(for MySQL)提供了和实例磁盘大小相同的免费存储空间，用于存放您的备份数据。

自动备份的生命周期和实例的生命周期相同。如果实例删除，那么自动备份也会同时被删除，如果您有手动备份，手动备份不会自动删除。

举个例子，假如用户购买实例时，存储空间选择了 200GB，那么所有的备份空间合计，只有超过 200GB 才开始计费，前 200GB 是免费的。对于超过免费容量的存储空间，目前仅支持按需计费，用多少收取多少费用。

### 须知

免费的存储空间是在收取了数据盘的存储空间费用后赠送的，数据盘存储空间冻结以后不再收费，因此不再享受备份赠送空间。

用户实例冻结后，将没有免费的存储空间，会导致实例原有自动备份收费。

- 如果选择解冻实例，将恢复免费的存储空间。
- 如果直接删除冻结的实例，原有的自动备份将会同时被删除，备份空间不会继续收费。

## 4.9 数据库参数修改

### 4.9.1 如何修改时区

GaussDB(for MySQL)支持创建实例时选择时区，创建完成后，可以修改时区。



## 操作步骤

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面，选择指定的实例，单击实例名称。

步骤 5 在左侧导航栏，选择“参数修改”。

步骤 6 在搜索框中搜索时区参数，例如 time\_zone。

步骤 7 选择时区，单击“保存”。

步骤 8 在弹框中，单击“是”，完成时区的修改。

----结束

## 4.9.2 GaussDB(for MySQL)密码过期策略

GaussDB(for MySQL)8.0 版本支持通过设置全局变量“default\_password\_lifetime”来控制用户密码的默认过期时间。

参数“default\_password\_lifetime”的值为 N，表示密码 N 天后过期，单位为天。默认值为 0，表示创建的用户密码永不过期。

```
mysql> show variables like 'default_password_lifetime';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| default_password_lifetime | 0 |
+-----+-----+
1 row in set (0.00 sec)
```

### 修改全局自动密码过期策略

- 您可以在云数据库 GaussDB(for MySQL)界面，通过设置参数“default\_password\_lifetime”的值，修改密码过期策略。  
参数修改具体请参见[编辑参数模板](#)。
- 通过命令修改全局变量“default\_password\_lifetime”的值。  
**mysql> set global default\_password\_lifetime=0;**

### 查看当前所有用户的密码过期时间

执行以下命令：

```
mysql> select user,host,password_expired,password_last_changed,password_lifetime
from user;
```

```
mysql> select user,host,password_expired,password_last_changed,password_lifetime from user;
+-----+-----+-----+-----+-----+
| user      | host      | password_expired | password_last_changed | password_lifetime |
+-----+-----+-----+-----+-----+
| mysql.session | localhost | N                | 2020-01-17 15:02:23   | NULL              |
| mysql.sys     | localhost | N                | 2020-01-17 15:02:23   | NULL              |
| rdsAdmin      | localhost | N                | 2020-01-17 15:02:30   | 0                 |
| root          | %         | N                | 2020-03-05 14:23:54   | NULL              |
| rdsRepl       | 192.168.% | N                | 2020-01-17 15:02:45   | 0                 |
| rdsMetric     | 192.168.% | N                | 2020-01-17 15:02:30   | 0                 |
| rdsBackup     | localhost | N                | 2020-01-17 15:02:30   | 0                 |
| u_test01      | %         | N                | 2020-03-05 14:28:10   | 30                |
| u_test02      | %         | N                | 2020-03-05 14:28:38   | NULL              |
| jeffrey       | localhost | N                | 2020-03-05 15:23:17   | NULL              |
+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

## 查看指定用户的密码过期策略

执行以下命令：

```
mysql> show create user jeffrey@'localhost';
```

```
mysql> show create user jeffrey@localhost;
+-----+-----+-----+-----+-----+
| CREATE USER for jeffrey@localhost |
| CREATE USER 'jeffrey'@'localhost' IDENTIFIED WITH 'mysql_native_password' AS '*1369F151658FC902555853119A9CBB0540B007F' REQUIRE NONE PASSWORD EXPIRE DEFAULT ACCOUNT UNLOCK |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

“EXPIRE DEFAULT” 表示遵从全局到期策略。

## 设置指定用户的密码过期策略

- 创建用户的同时设置密码过期策略

```
create user 'script'@'localhost' identified by '*****' password expire interval 90 day;
```

- 创建用户后设置密码过期策略

```
ALTER USER 'script'@'localhost' PASSWORD EXPIRE INTERVAL 90 DAY;
```

- 设置密码永不过期

```
mysql> CREATE USER 'mike'@'%' PASSWORD EXPIRE NEVER;
```

```
mysql> ALTER USER 'mike'@'%' PASSWORD EXPIRE NEVER;
```

- 设置密码遵从全局到期策略

```
mysql> CREATE USER 'mike'@'%' PASSWORD EXPIRE DEFAULT;
```

```
mysql> ALTER USER 'mike'@'%' PASSWORD EXPIRE DEFAULT;
```


## 4.9.3 如何修改 GaussDB(for MySQL)数据库字符集


GaussDB(for MySQL)的“utf8”只支持每个字符最多三个字节，而真正的 UTF-8 是每个字符最多四个字节。包括 Emoji 表情（Emoji 是一种特殊的 Unicode 编码，常见于手机上），和很多不常用的汉字，以及任何新增的 Unicode 字符等都无法使用 MySQL 的 utf8 字符集存储。在 2010 年 MySQL 发布了“utf8mb4”的字符集。MySQL 在 5.5.3 之后增加了 utf8mb4 的编码，兼容四字节的 unicode。utf8mb4 是 utf8 的超集，除了将编码改为 utf8mb4 外不需要做其他转换。

数据管理服务（Data Admin Service，简称 DAS）是一款专业的简化数据库管理工具，您可以通过数据管理服务（Data Admin Service，简称 DAS）SQL 控制台查看数据库和系统的字符集。

## 操作步骤

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面，选择目标实例，单击操作列的“登录”，进入数据管理服务数据库登录界面。

您也可以在“实例管理”页面，单击目标实例名称，在页面右上角，单击“登录”，进入数据管理服务数据库登录界面。

步骤 5 输入数据库用户名和密码，单击“登录”，即可进入您的数据库并进行管理。

步骤 6 在顶部菜单栏选择“SQL 操作” > “SQL 查询”，打开一个 SQL 窗口。

步骤 7 在“SQL 查询”执行如下 SQL 查看数据库字符集。

```
show variables like '%character%';
```

图4-7 SQL 执行结果



The screenshot shows the SQL query results for the command 'show variables like '%character%';'. The results are displayed in a table with two columns: 'Variable\_name' and 'Value'. There are 8 rows of data.

Variable_name	Value
character_set_client	utf8
character_set_connection	utf8mb4
character_set_database	utf8
character_set_filesystem	binary
character_set_results	utf8mb4
character_set_server	utf8
character_set_system	utf8
character_sets_dir	/usr/local/teurus-2.8.28.1-linux-x86_64/share/charsets/

步骤 8 在“SQL 查询”执行如下 SQL 查看数据库的编码。

```
show variables like 'collation%';
```

图4-8 SQL 执行结果



The screenshot shows the SQL query results for the command 'show variables like 'collation%';'. The results are displayed in a table with two columns: 'Variable\_name' and 'Value'. There are 3 rows of data.

Variable_name	Value
collation_connection	utf8mb4_unicode_ci
collation_database	utf8_general_ci
collation_server	utf8_general_ci

步骤 9 修改字符集为 utf8mb4。

1. 执行如下 SQL 更改数据库字符集。

```
ALTER DATABASE DATABASE_NAME DEFAULT CHARACTER SET  
utf8mb4 COLLATE utf8mb4_general_ci;
```

2. 执行如下 SQL 更改表字符集。

```
ALTER TABLE TABLE_NAME DEFAULT CHARACTER SET utf8mb4
COLLATE utf8mb4_general_ci;
```

#### 说明

SQL 语句修改的是表的字符集，表里面字段的字符集并没有被修改。

3. 执行如下 SQL 更改表中所有字段的字符集。

```
ALTER TABLE TABLE_NAME CONVERT TO CHARACTER SET utf8mb4
COLLATE utf8mb4_general_ci;
```

#### 说明

- `character_set_client`、`character_set_connection` 以及 `character_set_results` 是客户端的设置。
- `character_set_system`、`character_set_server` 以及 `character_set_database` 是服务器端的设置。
- 服务器端的参数优先级是：  
`character_set_database`>`character_set_server`>`character_set_system`。


----结束

## 4.9.4 使用 utf8mb4 字符集存储 emoji 表情到 GaussDB(for MySQL) 实例

如果要实现存储 emoji 表情到 GaussDB(for MySQL)实例，需要如下几方面统一使用或者支持 utf8mb4 字符集。

- 客户端：保证客户端输出的字符串的字符集为 utf8mb4。
- 应用到 GaussDB(for MySQL)实例的连接：支持 utf8mb4 字符集。以常见的 JDBC 连接为例，需要使用 MySQL Connector/J 5.1.13（含）以上的版本，JDBC 的连接串中，建议不配置“characterEncoding”选项。
- GaussDB(for MySQL)实例配置如下：
  - 设置控制台参数“character\_set\_server”为“utf8mb4”

参数名称	是否需要重启	值	允许值	描述
character_set_server	是	utf8mb4	utf8, latin1, gbk, utf8mb4	服务器字符集。

- 登录管理控制台。
  - 单击管理控制台左上角的 ，选择区域和项目。
  - 在“实例管理”页面，选择指定的实例，单击实例名称，进入实例的基本信息页面。
  - 在左侧导航栏中选择“参数修改”，在“参数”页签选择“character\_set\_server”，参数值修改为“utf8mb4”。
  - 单击“保存”，在弹出框中单击“是”，保存修改。
- 配置表的字符集为 utf8mb4

```
([redacted]) [redacted] > create table emoji_01 (id int auto_increment primary key, content varchar(255)) default charset utf8mb4;
Query OK, 0 rows affected (0.01 sec)

([redacted]) [redacted] > show create table emoji_01 \G
***** 1. row *****
Table: emoji_01
Create Table: CREATE TABLE 'emoji_01' (
  'id' int(11) NOT NULL AUTO_INCREMENT,
  'content' varchar(255) DEFAULT NULL,
  PRIMARY KEY ('id')
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
1 row in set (0.00 sec)
```

## 常见问题

如果您对于 JDBC 连接串设置了 `characterEncoding` 为 `utf8`，或者执行过上述操作后，仍旧无法正常插入 emoji 数据，建议您按照如下示例，在代码中指定连接的字符集为 `utf8mb4`：

```
String query = "set names utf8mb4";
stat.execute(query);
```

### 4.9.5 GaussDB(for MySQL)如何设置表名大小写敏感

表名大小写敏感可以通过在管理控制台或 API 创建数据库实例时指定。

已创建完成的实例不支持修改表名大小写敏感。

- 通过管理控制台的购买实例页面设置是否区分表名大小写。

图4-9 数据库设置

The screenshot shows the configuration interface for GaussDB(for MySQL). The 'Table Name Case Sensitivity' (表名大小写敏感) option is highlighted with a red box and set to 'Yes' (是). The 'Parameter Template' (参数模板) is set to 'Default-GaussDB-for-MySQL 8.0'. The 'Admin Username' (管理员用户名) is 'root'. The 'Admin Password' (管理员密码) and 'Confirm Password' (确认密码) fields are empty. The 'Enterprise Project' (企业项目) dropdown is set to 'Please select an enterprise project' (请选择企业项目).

- 通过 API 创建数据库实例设置 “`lower_case_table_names`” 指定大小写是否敏感，默认值是 “1”。

取值范围：

- 0：表名称大小写敏感。
- 1：表名将被存储成小写且表名称大小写不敏感。

### 4.9.6 是否支持使用 SQL 命令修改全局参数

云数据库 GaussDB(for MySQL)不支持在数据库中执行修改全局参数的命令，您可以到控制台修改参数。

#### 操作步骤

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面，选择指定的实例，单击实例名称。

步骤 5 在左侧导航栏，选择“参数修改”。

步骤 6 修改目标参数值，单击“保存”。

步骤 7 在弹框中，单击“是”，完成参数的修改。

----结束

## 4.10 网络安全

### 4.10.1 GaussDB(for MySQL)有哪些安全保障措施

#### 网络

- 云数据库 GaussDB(for MySQL)实例可以设置所属虚拟私有云，从而确保云数据库 GaussDB(for MySQL)实例与其他业务实现网络安全隔离。
- 使用安全组确保访问源为可信的。
- 使用 SSL 通道，确保数据传输加密。

#### 管理

通过统一身份认证服务（Identity and Access Management，简称 IAM），可以实现对云数据库 GaussDB(for MySQL)实例的管理权限控制。

### 4.10.2 如何防止任意源连接数据库

- 数据库开放 EIP 后，如果公网上的恶意人员获取到您的 EIPDNS 和数据库端口，那么便可尝试破解您的数据库并进行进一步破坏。因此，强烈建议您保护好 EIP DNS、数据库端口、数据库帐号和密码等信息，并通过云数据库 GaussDB(for MySQL)实例的安全组限定源 IP，保障只允许可信源连接数据库。
- 为避免恶意人员轻易破解您的数据库密码，请按照云数据库 GaussDB(for MySQL)实例的密码策略设置足够复杂度密码，并定期修改。

### 4.10.3 访问 GaussDB(for MySQL)实例应该如何配置安全组

- 通过内网访问 GaussDB(for MySQL)实例时，设置安全组分为以下两种情况：
  - ECS 与 GaussDB(for MySQL)实例在相同安全组时，默认 ECS 与 GaussDB(for MySQL)实例互通，无需设置安全组规则。
  - ECS 与 GaussDB(for MySQL)实例在不同安全组时，需要为 GaussDB(for MySQL)和 ECS 分别设置安全组规则。
    - 设置 GaussDB(for MySQL)安全组规则：为 GaussDB(for MySQL)所在安全组配置相应的入方向规则。
    - 设置 ECS 安全组规则：安全组默认规则为出方向上数据报文全部放行，此时，无需对 ECS 配置安全组规则。当在 ECS 所在安全组为非默认安全

组且出方向规则非全放通时，需要为 ECS 所在安全组配置相应的出方向规则。

- 通过弹性公网 IP 访问 GaussDB(for MySQL)实例时，需要为 GaussDB(for MySQL) 所在安全组配置相应的入方向规则。

## 4.10.4 将根证书导入 Windows/Linux 操作系统

### 导入 Windows 操作系统

1. 单击“开始”，运行框输入“MMC”，回车。
2. 在 MMC 控制台菜单栏中单击“文件”，选择“添加/删除管理单元”。
3. 在“添加或删除管理单元”对话框，选择“可用管理单元”区域的“证书”。单击“添加”添加证书。
4. 在“证书管理”对话框，选择“计算机账户”，单击“下一步”。
5. 在“选择计算机”对话框，单击“完成”。
6. 在“添加或删除管理单元”对话框，单击“确定”。
7. 在 MMC 控制台，双击“证书”。
8. 右键单击“受信任的根证书颁发机构”，选择“所有任务”，单击“导入”。
9. 单击“下一步”。
10. 单击“浏览”，将文件类型更改为“所有文件 (\*.\*)”。
11. 找到下载根证书 ca.pem 文件，单击“打开”，然后在向导中单击“下一步”。

#### 须知

您必须在浏览窗口中将文件类型更改为“所有文件 (\*.\*)”才能执行此操作，因为“.pem”不是标准证书扩展名。

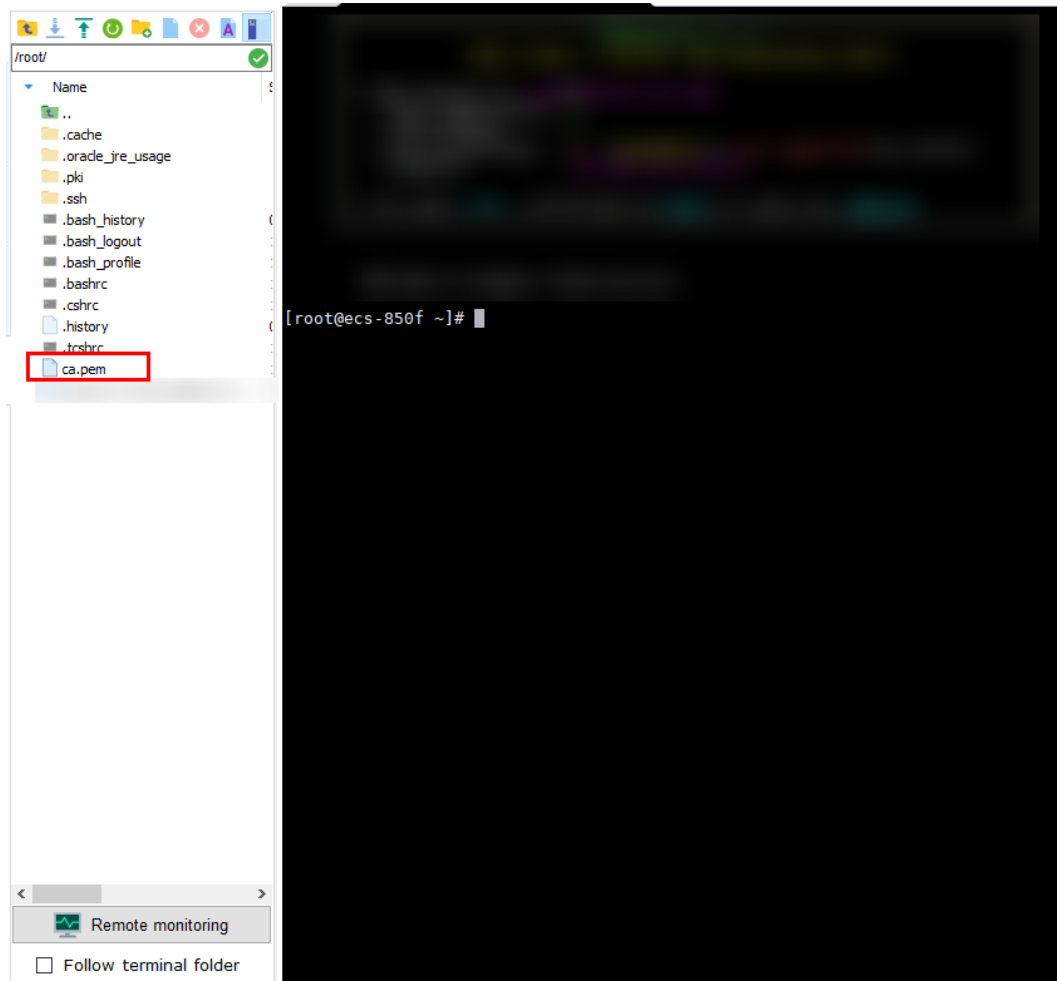
12. 单击“下一步”。
13. 单击“完成”。
14. 单击“确定”，完成根证书导入。

### 导入 Linux 操作系统

您可以使用任何终端连接工具（如 WinSCP、PuTTY 等工具）将证书上传至 Linux 系统任一目录下。

示例：

图4-10 导入证书



#### 4.10.5 如何管理 GaussDB(for MySQL)安全性

若要避免安全性问题，切勿使用用户账户的主天翼云用户名和密码。推荐您使用您的主天翼云账户来创建 IAM 用户，并将这些用户分配给数据库用户账户。如有必要，您还可使用您的主账户创建其他用户账户。

创建账户过程中出错，可能是因为您的账户缺少权限，或者您的账户未正确设置。示例：

您的访问策略中可能缺少执行特定操作的权限，如创建数据库实例。要修复该问题，您的 IAM 管理员需要为您的账户提供必要的角色。



## 4.11 日志管理

### 4.11.1 如何查看 GaussDB(for MySQL)执行过的所有 SQL 日志

您可以通过数据管理服务（Data Admin Service，简称 DAS）这款可视化的专业数据库管理工具，快速查找目标 SQL 执行记录信息。

#### 通过 DAS 查询 SQL 日志

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面，选择目标实例，单击操作列的“登录”，进入数据管理服务登录界面。

步骤 5 正确输入数据库用户名和密码，单击“登录”，即可进入您的数据库。

步骤 6 在顶部菜单栏选择“SQL 操作” > “SQL 执行记录”，打开历史执行记录列表。

步骤 7 在 SQL 执行记录页面，您可通过日期范围、数据库名称、SQL 语句关键字进行搜索，快速查找目标 SQL 执行记录信息。

图4-11 SQL 执行记录




- 单击列表中数据库名称，您可直接进入该数据库管理页面。
- 单击“SQL 语句”，您可在 SQL 语句弹出框中复制使用 SQL。
- 单击“在 SQL 执行窗口打开”，您可在 SQL 窗口中直接使用语句。

----结束

### 4.11.2 如何查看 GaussDB(for MySQL)慢 SQL?

#### 查看日志明细

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面，选择目标实例，单击实例名称，进入“基本信息”页面。

步骤 5 在左侧导航树，单击“日志管理”。

步骤 6 选择“慢日志”页签，查看慢 SQL 语句的详细信息。


慢日志功能支持查看指定执行语句类型或时间段的慢日志记录。

----结束

### 4.11.3 GaussDB(for MySQL)服务如何开启并查看 binlog 文件?

#### 开启 binlog

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 选中要开启 binlog 的实例，单击实例名称，进入实例详情界面。

步骤 5 在实例详情界面，单击参数修改，跳转到参数组界面。

步骤 6 搜索 log-bin 参数，在值列下拉框中选择“ON”，单击保存。

步骤 7 对实例进行重启操作，打开 binlog。

----结束

#### 查看 binlog 文件

步骤 1 连接数据库实例。

步骤 2 执行以下 show binlog 命令可以直接读取 binlog 日志。

```
SHOW BINLOG EVENTS [IN 'log_name'] [FROM pos] [LIMIT [offset,] row_count];
```

##### 说明

如果提示帐号权限不足，请使用 root 帐号登录查询。

----结束

#### 开启 binlog 对 GaussDB(for MySQL)的性能影响

性能损耗和业务压力大相关，开启 Binlog 不会影响查询（SELECT）性能，只会影响写入更新（如 INSERT、UPDATE、DELETE 等）性能。



##### 说明

GaussDB(for MySQL)服务的 binlog 和开源的 MySQL-binlog 两者从使用方法上来说无明显差异，GaussDB(for MySQL)服务的 binlog 完全兼容开源 MySQL 的 binlog 相关语法。

## 4.11.4 如何修改 binlog 保留时长

GaussDB(for MySQL)兼容社区 8.0 版本的 `binlog_expire_logs_seconds` 参数，使用 `binlog_expire_logs_seconds` 参数设置 binlog 保留时长。

### 操作步骤

- 步骤 1 登录管理控制台。
- 步骤 2 单击管理控制台左上角的 ，选择区域和项目。
- 步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。
- 步骤 4 在“实例管理”页面，选择指定的实例，单击实例名称，进入实例的基本信息页面。
- 步骤 5 在左侧导航栏中选择“参数修改”，在“参数”页签查看参数“log-bin”，确认参数值为“ON”，表示 binlog 已经开启。
- 步骤 6 在“参数”页签查看参数“binlog\_expire\_logs\_seconds”，修改为需要的保留时间，单位是秒。

### 说明

- 在新生成一个 binlog 的文件的时候，数据库会启动清理功能，清理已经超过保留时间的 binlog 文件。
- 如果一直未生成新的 binlog 文件，则历史的 binlog 文件一直不会被清理，可能会超过保留时间。如果需要立即清理，可以连接数据库执行 `flush logs`；强制生成新的 binlog 文件，此时会触发清理机制。

----结束

## 4.11.5 如何查看 GaussDB(for MySQL)数据库的死锁日志

数据库的死锁日志不会记录在错误日志中，您可以通过数据管理服务（Data Admin Service，简称 DAS）这款可视化的专业数据库管理工具，快速执行 SQL 语句查看。

### 操作步骤



- 步骤 1 登录管理控制台。
- 步骤 2 单击管理控制台左上角的 ，选择区域和项目。
- 步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。
- 步骤 4 在“实例管理”页面，选择目标实例，单击操作列的“登录”，进入数据管理服务登录界面。
- 步骤 5 正确输入数据库用户名和密码，单击“登录”，即可进入您的数据库。

图4-12 登录数据库



步骤 6 选择目标数据库，选择“SQL 操作 > SQL 查询”，新建一个 SQL 查询窗口。

步骤 7 查询窗口中使用 `show engine innodb status` 查看当前数据库最新的死锁日志，通过关键字 **LATEST DETECTED DEADLOCK** 快速定位，最新产生的死锁日志会覆盖历史死锁日志记录。

----结束

## 4.12 版本升级

### 4.12.1 如何查看云数据库 GaussDB(for MySQL)实例的版本号

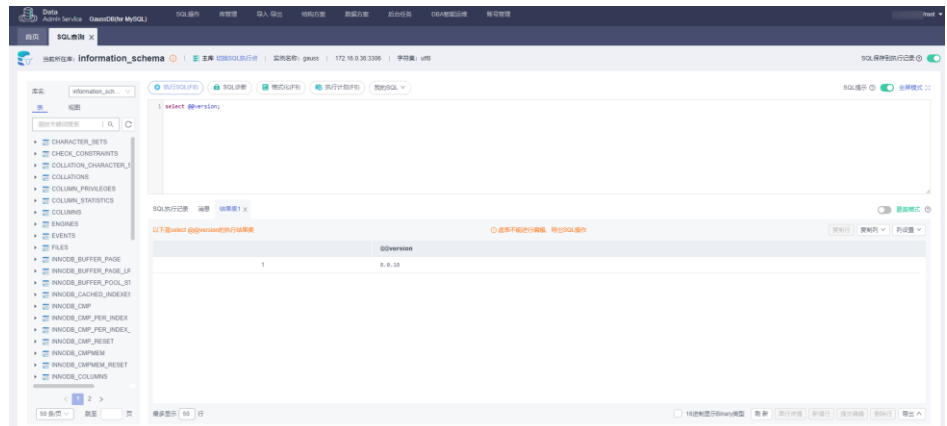
- 通过 GaussDB(for MySQL)目标实例的基本信息页面，查看数据库实例的版本号。

图4-13 基本信息



- 通过 DAS 页面，查看数据库实例的版本号。
  - a. 在 DAS 页面，登录目标实例。
  - b. 在界面上方选择“SQL 操作 > SQL 查询”。
  - c. 执行 `select @@version;`语句，查看该实例的版本号。

图4-14 查看版本号



## 4.12.2 GaussDB(for MySQL)是否支持版本升级

GaussDB(for MySQL)支持手动升级内核小版本，内核小版本的升级涉及性能提升、新功能或问题修复等。

升级实例时，升级顺序依次是只读节点、主节点。

# 5 故障排除

## 5.1 备份恢复

### 5.1.1 mysqldump 导出数据报错权限不足

#### 场景描述

mysqldump 使用指定用户导出数据库数据时，报错：'Access denied; you need (at least one of) the PROCESS privilege(s)'

```
[root@xxxx ~]# mysqldump -h192.168.1.100 -P3306 -uadmin -p123456 --set-gtid-purged=off --skip-lock-tables zzkj > zzkj.sql
mysqldump: [warning] Using a password on the command line interface can be insecure.
mysqldump: Error: 'Access denied; you need (at least one of) the PROCESS privilege(s) for this operation' when trying to dump tablespaces
```

#### 原因分析

mysqldump 使用指定用户导出数据时，需要赋予 PROCESS 权限。

#### 解决方案

使用管理员帐户给相应用户授予 PROCESS 权限。

```
GRANT SELECT, PROCESS ON *.* TO 'dump_user' @' %' ;  
FLUSH PRIVILEGES;
```

### 5.1.2 使用 mysqlbinlog 工具获取 binlog

本文以从弹性云服务器 ECS 上拉取为例，其他环境下方法类似。

1. 在 ECS 上安装 MySQL 客户端，详情请参考[安装 MySQL 客户端](#)。

#### 📖 说明

GaussDB(for MySQL)兼容社区 MySQL 8.0 及以上版本，请勿安装 8.0 以下版本的版本的客户端。

2. 执行命令，下载 binlog 文件。

```
mysqlbinlog -hxxx -uxxx -Pxxx -pxxx binlog.xxxx --read-from-remote-server
```

mysqlbinlog 的常用参数：

- -h: 数据库 host。
- -u: 用户名。
- -P: 端口号。
- -p: 密码。
- --start-position: 表示从指定的起始位置开始解析。
- --start-datetime: 表示从指定的时间开始解析。
- --stop-position: 表示解析到指定的位置。
- --stop-datetime: 表示解析到指定的时间。
- --skip-gtids: 跳过打印 `gtid_log_event`。
- --short-form: 表示只显示 `statements`。
- --result-file: 将 binlog 解析生成 sql 文件。
- --read-from-remote-server: 远程下载 binlog(用于 `mysqlbinlog` 与数据库服务端不再同一台机器的情况)。

### 5.1.3 canal 解析 binlog 报错

#### 场景描述

canal 解析 Binlog 出现错误，导致拉取 Binlog 中断，错误信息如下：

```
com.alibaba.otter.canal.parse.exception.CanalParseException:
java.lang.NumberFormatException: - Caused by: java.lang.NumberFormatException: - at
com.alibaba.fastsql.sql.parser.Lexer.integerValue(Lexer.java:2454)
```

```

219760 | 1 | 1 | EXCEPTION | pid:1 nid:1 | exception:canal:canal1:com.alibaba.otter.canal.parse.exception.CanalParseException: java.lang.NumberFormatException: -
Caused by: java.lang.NumberFormatException: - Lexer.integerValue(Lexer.java:2454)
at com.alibaba.fastsql.sql.parser.SQLStatementParser.parseValueClause(SQLStatementParser.java:5101)
at com.alibaba.fastsql.sql.dialect.mysql.parser.MySQLStatementParser.parseInsert(MySQLStatementParser.java:3674)
at com.alibaba.fastsql.sql.parser.SQLStatementParser.parseStatementList(SQLStatementParser.java:41)
at com.alibaba.fastsql.sql.parser.SQLStatementParser.parseStatementList(SQLStatementParser.java:230)
at com.alibaba.fastsql.sql.parser.SQLStatementParser.parseStatementList(SQLStatementParser.java:93)
at com.alibaba.fastsql.sql.SQUT11s.parseStatements(SQUT11s.java:534)
at com.alibaba.otter.canal.parse.inbound.mysql.doll.DruidDDLParser.parse(DruidDDLParser.java:51)
at com.alibaba.otter.canal.parse.inbound.mysql.dbsync.LogEventConvert.parseRowsQueryEvent(LogEventConvert.java:379)
at com.alibaba.otter.canal.parse.inbound.mysql.dbsync.LogEventConvert.parse(LogEventConvert.java:130)
at com.alibaba.otter.canal.parse.inbound.mysql.dbsync.LogEventConvert.parse(LogEventConvert.java:67)
at com.alibaba.otter.canal.parse.inbound.AbstractEventParser.parseAndProfileIfNeededNecessary(AbstractEventParser.java:409)
at com.alibaba.otter.canal.parse.inbound.AbstractEventParser$Sink.run(AbstractEventParser.java:208)
at com.alibaba.otter.canal.parse.inbound.mysql.MySQLConnection.dump(MySQLConnection.java:168)
at com.alibaba.otter.canal.parse.inbound.AbstractEventParser$Sink.run(AbstractEventParser.java:271)
at java.lang.Thread.run(Thread.java:748)
| 2022-03-30 14:21:17 | 2022-03-30 14:21:17 | EXCEPTION | pid:1 nid:null | exception:cid:1 stop recovery successful for rid:1
219761 | NULL | -1 | -1 |
| 219762 | 1 | 1 | EXCEPTION | pid:1 nid:1 | exception:canal:canal1:com.alibaba.otter.canal.parse.exception.CanalParseException: java.lang.NumberFormatException: -
Caused by: java.lang.NumberFormatException: - Lexer.integerValue(Lexer.java:2454)
at com.alibaba.fastsql.sql.parser.Lexer.integerValue(Lexer.java:2454)
at com.alibaba.fastsql.sql.parser.SQLStatementParser.parseValueClause(SQLStatementParser.java:5101)
at com.alibaba.fastsql.sql.dialect.mysql.parser.MySQLStatementParser.parseInsert(MySQLStatementParser.java:3674)
at com.alibaba.fastsql.sql.parser.SQLStatementParser.parseStatementList(SQLStatementParser.java:41)
at com.alibaba.fastsql.sql.parser.SQLStatementParser.parseStatementList(SQLStatementParser.java:230)
at com.alibaba.fastsql.sql.parser.SQLStatementParser.parseStatementList(SQLStatementParser.java:93)
at com.alibaba.fastsql.sql.SQUT11s.parseStatements(SQUT11s.java:534)
at com.alibaba.otter.canal.parse.inbound.mysql.doll.DruidDDLParser.parse(DruidDDLParser.java:51)
at com.alibaba.otter.canal.parse.inbound.mysql.dbsync.LogEventConvert.parseRowsQueryEvent(LogEventConvert.java:379)
at com.alibaba.otter.canal.parse.inbound.mysql.dbsync.LogEventConvert.parse(LogEventConvert.java:130)
at com.alibaba.otter.canal.parse.inbound.AbstractEventParser.parseAndProfileIfNeededNecessary(AbstractEventParser.java:409)
at com.alibaba.otter.canal.parse.inbound.AbstractEventParser$Sink.run(AbstractEventParser.java:208)
at com.alibaba.otter.canal.parse.inbound.mysql.MySQLConnection.dump(MySQLConnection.java:168)
at com.alibaba.otter.canal.parse.inbound.AbstractEventParser$Sink.run(AbstractEventParser.java:271)
at java.lang.Thread.run(Thread.java:748)
| 2022-03-30 14:21:28 | 2022-03-30 14:21:28 | EXCEPTION | pid:1 nid:null | exception:cid:1 stop recovery successful for rid:1
219763 | NULL | -1 | -1 |

```

#### 原因分析

检查 GaussDB(for MySQL)的参数“`binlog_rows_query_log_events`”的值是否设置为 **1** 或 **ON**。

- 目前 canal 只能支持 ROW 格式的 Binlog 增量订阅。
- 当 GaussDB(for MySQL)的参数“`binlog_rows_query_log_events`”的值设置为 **1** 或 **ON** 时，会在 Binlog 中产生 `Rows_query` 类型的 event，此类 event 非 ROW 格式，一些场景下，会导致 canal 出现 blank topic 问题，引发 Binlog 解析失败。

## 解决方案

将 GaussDB(for MySQL)的参数“binlog\_rows\_query\_log\_events”的值修改为 **OFF**，重启中断的 canal 任务。

### 5.1.4 使用 mysqldump 导出大表的注意事项

在使用 `mysqldump` 导出数据时，倘若添加 `-q(--quick)` 参数时，`select` 出来的结果将不会存放在缓存中，而是直接导出到标准输出中。如果不添加该参数，则会把 `select` 的结果放在本地缓存中，然后再输出给客户端。

- 如果只是备份少量数据，足以放在空闲内存 `buffer` 中的话，禁用 `-q` 参数，则导出速度会快一些。
- 对于大数据集，如果没办法完全储存在内存缓存中时，就会产生 `swap`。对于大数据集的导出，不添加 `-q` 参数，不但会消耗主机的内存，也可能造成数据库主机因无可用内存继而宕机的严重后果。

因此，如果使用 `mysqldump` 来备份数据时，建议添加 `-q` 参数。

导出示例：

```
mysqldump -uroot -p-P8635 -h192.168.0.199 --set-gtid-purged=OFF --single-transaction --flush-logs -q test t1>t1.sql
```

### 5.1.5 mysqldump 的 6 大使用场景的导出命令

#### 背景描述

`mysqldump` 是 MySQL 最常用的逻辑导入导出的工具，下面介绍几种常见使用场景。

#### mysqldump 选项解析

表5-1 配置项说明

选项名称	说明
<code>add-drop-table</code>	每个数据表创建之前添加 <code>drop</code> 数据表语句。
<code>events, E</code>	导出事件。
<code>routines, R</code>	存储过程以及自定义函数。
<code>flush-logs</code>	开始导出之前刷新日志。
<code>no-create-db, n</code>	只导出数据，而不添加 <code>CREATE DATABASE</code> 语句。
<code>add-drop-database</code>	创建数据库之前添加 <code>drop</code> 数据库语句。
<code>no-create-info, t</code>	只导出数据，而不添加 <code>CREATE TABLE</code> 语句。



选项名称	说明
no-data, d	不导出任何数据，只导出数据库表结构。
set-gtid-purged=OFF	不导出 gtid 相关语句。
hex-blob	使用十六进制格式导出二进制字符串字段。

## 场景描述

适用场景举例如下。

1. 导出 db1、db2 两个数据库的所有数据。

```
mysqldump -uroot -p -P8635 -h192.168.0.199 --hex-blob --set-gtid-purged=OFF  
--single-transaction --order-by-primary --flush-logs -q --databases db1  
db2 >db12.sql
```

2. 导出 db1 库的 t1 和 t2 表。

```
mysqldump -uroot -p -P8635 -h192.168.0.199 --hex-blob --set-gtid-purged=OFF  
--single-transaction --order-by-primary --flush-logs -q --databases db1 --tables t1  
t2 >t1_t2.sql
```

3. 条件导出，导出 db1 表 t1 中 id=1 的数据。

```
mysqldump -uroot -p -P8635 -h192.168.0.199 --hex-blob --set-gtid-  
purged=OFF --single-transaction --order-by-primary --flush-logs -q --databases  
db1 --tables t1 --where='id=1'>t1_id.sql
```

4. 导出 db1 下所有表结构，而不导出数据。

```
mysqldump -uroot -p -P8635 -h192.168.0.199 --no-data --set-gtid-purged=OFF -  
-single-transaction --order-by-primary -n --flush-logs -q --databases  
db1 >db1_table.sql
```

5. 除 db1 下的表和数据外，其他对象全部导出。

```
mysqldump -uroot -p -h192.168.0.199 -P8635 --set-gtid-purged=OFF -F -n -t -  
d -E -R db1 > others.sql
```

## 5.1.6 增加表字段后出现运行卡顿现象

### 故障描述

当给 MySQL 实例的表中增加一个字段，出现系统无法访问的现象。

### 解决方案

因增加表字段而引起数据库出现性能问题，有可能是未对新增字段添加索引，数据量大导致消耗了大量的 CPU 资源。为此，提出如下建议恢复数据库性能。

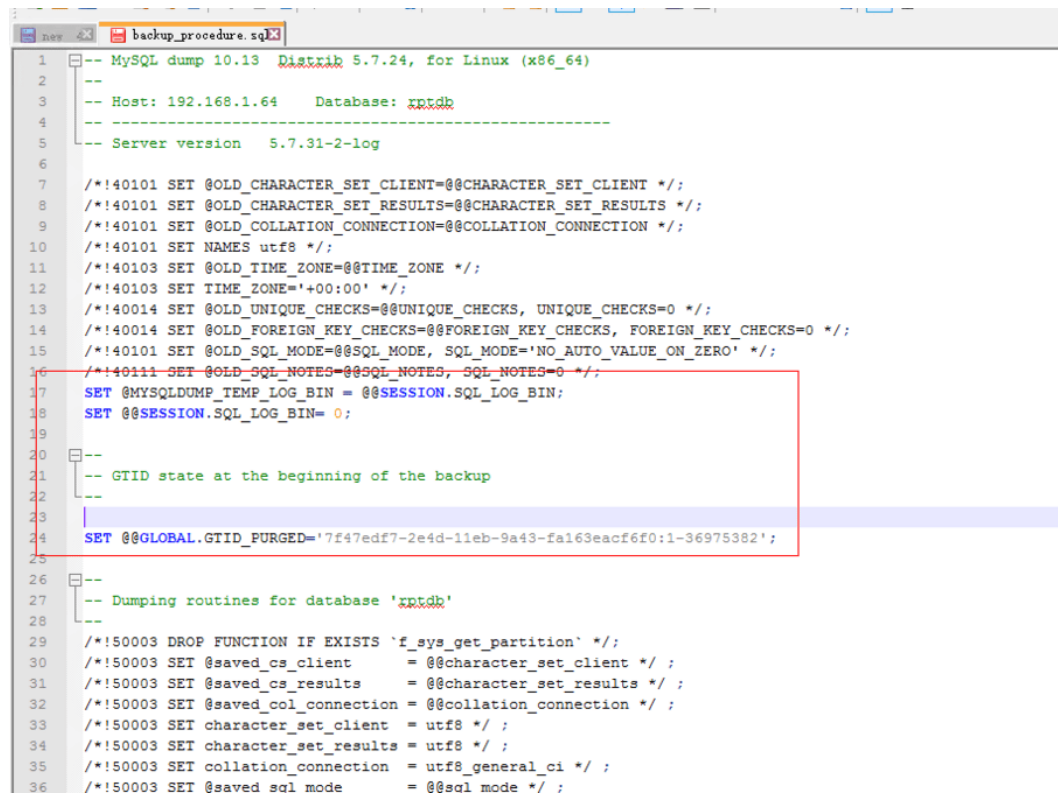
- 添加对应索引、主键。
- 优化慢 SQL 语句。

## 5.1.7 怎么解决执行 mysqldump 出现 SET @@SESSION.SQL\_LOG\_BIN 等 SQL 的问题

### 场景描述

新购买的云数据库 GaussDB(for MySQL)，执行 mysqldump 时，会出现如下如所示代码。

图5-1 代码显示



```
1  -- MySQL dump 10.13 Distrib 5.7.24, for Linux (x86_64)
2  --
3  -- Host: 192.168.1.64 Database: xntdb
4  -----
5  -- Server version  5.7.31-2-log
6
7  /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
8  /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
9  /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
10 SET NAMES utf8 */;
11 /*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
12 /*!40103 SET TIME_ZONE='+00:00' */;
13 /*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
14 /*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
15 /*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
16 /*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
17 SET @@SESSION.SQL_LOG_BIN = @@SESSION.SQL_LOG_BIN;
18 SET @@SESSION.SQL_LOG_BIN= 0;
19
20 --
21 -- GTID state at the beginning of the backup
22 --
23
24 SET @@GLOBAL.GTID_PURGED='7f47edf7-2e4d-11eb-9a43-fa163eac6f0:1-36975382';
25
26 --
27 -- Dumping routines for database 'xntdb'
28 --
29 /*!50003 DROP FUNCTION IF EXISTS `f_sys_get_partition` */;
30 /*!50003 SET @saved_cs_client      = @@character_set_client */ ;
31 /*!50003 SET @saved_cs_results    = @@character_set_results */ ;
32 /*!50003 SET @saved_col_connection = @@collation_connection */ ;
33 /*!50003 SET character_set_client  = utf8 */ ;
34 /*!50003 SET character_set_results = utf8 */ ;
35 /*!50003 SET collation_connection  = utf8_general_ci */ ;
36 /*!50003 SET @saved_sql_mode      = @@sql_mode */ ;
```

### 故障分析

开启了“gtid-mode=ON”参数。

如果一个数据库开启了 GTID，使用 mysqldump 备份或者转储的时候，即使不是 MySQL 全库(所有库)备份，也会备份整个数据库所有的 GTID 号。

### 解决方案

在 GaussDB(for MySQL)数据库进行导出备份和恢复的时候，需要注意是否启用数据库用 GTID 模式。

如果开启，则在 mysqldump 数据时，应该在 mysqldump 命令加上参数“-set-gtid-purged=OFF”。

## 5.1.8 canal 工具报错权限不足

### 场景描述

在搭建 canal 环境，使用指定用户从 GaussDB(for MySQL)获取 Binlog 时，启动 canal 经常会报如下错误：'show master status' has an error! Access denied: you need (at least one of) the SUPER, REPLICATION CLIENT privilege(s) for this operation

完整报错信息如下：

```
2021-01-10 23:58:32.964 [destination = evoicedc , address = /dbus-mysql:3306 ,
EventParser] ERROR com.alibaba.otter.canal.common.alarm.LogAlarmHandler -
destination:evoicedc[com.alibaba.otter.canal.parse.exception.CanalParseException:
command : 'show master status' has an error!
Caused by: java.io.IOException: ErrorPacket [errorNumber=1227, fieldCount=-1,
message=Access denied; you need (at least one of) the SUPER, REPLICATION CLIENT
privilege(s) for this operation, sqlState=42000, sqlStateMarker=#] with command:
show master status at
com.alibaba.otter.canal.parse.driver.mysql.MySqlQueryExecutor.query(MySqlQueryExecu
tor.java:61)
```

### 原因分析

canal 拉取 Binlog 时需要赋予 REPLICATION SLAVE, REPLICATION CLIENT 权限。

### 解决方案

使用管理员帐户给相应用户授予 REPLICATION SLAVE, REPLICATION CLIENT 权限。

```
GRANT SELECT, REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO  
'canal' @' %' ;
```

```
FLUSH PRIVILEGES;
```

## 5.2 连接类

### 5.2.1 root 账号的 ssl\_type 修改为 ANY 后无法登录

#### 场景描述

在控制台以 root 帐号通过 DAS 登录实例时，报错 Access denied。

The screenshot shows a login form with the following fields and elements:

- 登录用户名** (Login Username): Input field containing 'root'.
- 密码** (Password): Input field with masked characters '.....' and a '测试连接' (Test Connection) button.
- 错误信息** (Error Message): A red message indicating '连接失败, 失败原因: Access denied for user 'root'@'100.xxxxxxx' (using password: YES) 请参见 常见连接失败原因及解决方法' (Connection failed, failure reason: Access denied for user 'root'@'100.xxxxxxx' (using password: YES) Please refer to common connection failure reasons and solutions).
- 记住密码** (Remember Password): A checked checkbox with the text '同意DAS使用加密方式记住密码 (建议选中, 否则DAS将无法开启元数据采集功能)' (I agree that DAS uses encryption to remember the password (recommended to be selected, otherwise DAS will be unable to start the metadata collection function)).
- 描述** (Description): Input field containing 'created by sync rds instance'.

## 原因分析

1. 查看 mysql.user 表中的 root 帐号信息，排查客户端 IP 范围是否正确、是否使用 SSL。

```
SELECT * FROM mysql.user WHERE User='root';
```

如果发现 root 帐号的 **ssl\_type** 被设置为 **ANY**，表明 root 帐号需要使用 SSL 连接。

2. 查看 SSL 开启情况。

```
show variables like '%ssl%';
```

发现该实例未开启 SSL:

以下是 show variables like "ssl%" 的执行结果

Variable_name	Value
have_openssl	DISABLED
have_ssl	DISABLED
ssl_ca	/CA/ca.pem
ssl_capath	
ssl_cert	/CA/server.pem
ssl_cipher	
ssl_cr1	
ssl_cr1path	
ssl_key	/CA/server.key

因此，问题原因是自行修改 root 帐号的 **ssl\_type** 为 **ANY** 后，导致无法登录。

## 解决方案

将 root 帐号的 **ssl\_type** 修改为空即可，参考命令：

```
update mysql.user set ssl_type='' where user = 'root';
```

如果要将其他所有用户帐号的 **ssl\_type** 修改为空，参考命令：

```
update mysql.user set ssl_type='' where user not like 'mysql%';
```

## 5.2.2 SSL 使用与介绍

### 场景描述

使用 SSL 无法连接上数据库。

### 原因分析

优先检查网络是否已经连通，如果不带 SSL 的连接方式可以连接，则可能是 mysql client 或对应的数据库驱动的版本不兼容。

### 解决方案

GaussDB(for MySQL)是兼容社区 8.0 以上版本的，需要使用 8.0 及以上版本的 mysql client 或数据库驱动。

SSL(Secure Socket Layer: 安全套接字层)使用数据加密、身份校验和消息完整性校验，为连接提供安全性保证。

**SSL 提供的功能主要包含：**

1. 加密数据传输：利用对称密钥算法对传输的数据进行加密。

- 身份校验：基于证书使用数字签名的方法对客户端与服务器进行身份验证。
- 消息完整性校验：消息传输过程中使用 MAC 算法来检验消息的完整性。

### 注意

- 当服务端的 SSL 开启时，客户端的身份验证是可选的，即：即使服务端开启了 SSL，客户端也可以不通过 SSL 的方式连接服务端，此时也可以正常通信，只是数据不会被加密。
- 如果不是通过 SSL 的方式，那么其在网络中的传输数据会以明文进行，存在安全隐患。
- GaussDB(for MySQL)数据库服务端会默认开启服务端会默认开启 SSL，客户业务使用时可以在客户端自行选择是否使用 SSL。
- 使用 mysql client 连接时使用 SSL 的方式可参考：[通过客户端连接 GaussDB\(for MySQL\)](#)
- 使用 JDBC 连接时使用 SSL 的方式可参考：[通过 JDBC 连接 MySQL 数据库](#)

## 5.2.3 对各个 IP 地址的解释说明

购买 GaussDB(for MySQL)实例后获得了多个 IP 地址，以 1 主 1 只读为例，在实例基础信息中最多共能找到 4 个 IP，业务可以按自己的需要连接对应的 IP。

### 说明

对于节点读内网地址，如果出现节点故障，故障恢复前 IP 不可访问。

#### 1. 主节点读内网地址（不推荐使用）

IP 与节点绑定，可以从内网（同 VPC 网络内）直接连接 IP 做读写操作，如果发生故障倒换，节点变为只读节点，则该 IP 将只能做读操作，不能做写操作。对该 IP 的操作实际会落到对应的节点上。

节点信息

您可以开启故障转移，通过高可用连接地址访问实例，避免故障导致内网地址发生变化也无需在应用程序中重新配置连接地址。点击故障转移，跳转到数据库代理页面。

节点名称/ID	角色	运行状态	可用区	读内网地址	故障转移优先级	操作
	主节点	正常	可用区1	192.168.0.5	1	<a href="#">查看连接标识</a> <a href="#">重命名</a>
BUG-retest-708a41c7810...	只读节点	正常	可用区1	192.168.0.4	1	<a href="#">查看连接标识</a> <a href="#">只读升主</a> <a href="#">重命名</a>

#### 2. 只读节点读内网地址（不推荐使用）

IP 与节点绑定，可以从内网（同 VPC 网络内）直接连接 IP 做读操作，如果发生故障倒换，节点变为主节点，则该 IP 将能做读写操作。对该 IP 的操作实际会落到对应的节点上。

节点信息

您可以开启故障转移，通过高可用连接地址访问实例，避免故障导致内网地址发生变化也无需在应用程序中重新配置连接地址。点击故障转移，跳转到数据库代理页面。

节点名称/ID	角色	运行状态	可用区	读内网地址	故障转移优先级	操作
	主节点	正常	可用区1	192.168.0.5	1	<a href="#">查看连接标识</a> <a href="#">重命名</a>
	只读节点	正常	可用区1	192.168.0.4	1	<a href="#">查看连接标识</a> <a href="#">只读升主</a> <a href="#">重命名</a>

#### 3. 读写内网地址

浮动 IP，IP 永远与主节点绑定，可以内网（同 VPC 网络内）直接连接 IP 做读写操作。如果发生故障倒换，该 IP 会浮动到新的主节点，依然可以做读写操作。对该 IP 的操作永远会落到当时的主节点上。

网络信息		连接实例>>	
读写内网地址	192.168.0.3	读写公网地址	绑定
数据库端口	3306	建议最大连接数	3,000
虚拟私有云	taurus_check2	子网	subnet_check (192.168.0.0/28)
内网安全组	Sys-default		

#### 4. 读写公网地址(创建实例后需要单独绑定)

创建并绑定公网 IP 后，可以从公网连接 IP 做读写操作。与浮动 IP 相同，也是一直与主节点绑定，且一直可以做读写操作。对该 IP 的操作永远会落到当时的主节点上。

网络信息		连接实例>>	
读写内网地址	192.168.0.3	读写公网地址	10.154.217.41 解绑
数据库端口	3306	建议最大连接数	3,000
虚拟私有云	taurus_check2	子网	subnet_check (192.168.0.0/28)
内网安全组	Sys-default		

#### 说明

故障倒换:

GaussDB(for MySQL)默认最少为 2 个节点，1 主（可读可写）1 只读（只可读不可写），主节点仅允许有一个，只读节点可以有多个。

当主节点遇到故障时，高可用系统会迅速发现，并选择一个只读节点将其升级为主节点，并将原主节点修复为只读节点，这个过程叫故障倒换。

## 5.2.4 客户端 TLS 版本 MySQL 不一致导致 SSL 连接失败

### 场景描述

某业务客户端连接到云上 GaussDB(for MySQL)失败，但是连接到自建环境或其他环境可以成功，均使用了 SSL 连接。

### 原因分析

排查步骤:

1. 查看 GaussDB(for MySQL)的错误日志，观察到如下报错:

```
2021-07-09T10:30:58.476586+08:00 212539 [Warning] SSL errno: 337678594, SSL
errmsg: error:14209102:SSL
routines:tls_early_post_process_client_hello:unsupported protocol2021-07-
09T10:30:58.476647+08:00 212539 [Note] Bad handshake2021-07-
09T10:32:43.535738+08:00 212631 [Warning] SSL errno: 337678594, SSL errmsg:
error:14209102:SSL routines:tls_early_post_process_client_hello:unsupported
protocol2021-07-09T10:32:43.535787+08:00 212631 [Note] Bad handshake2021-07-
09T10:50:03.401100+08:00 213499 [Warning] SSL errno: 337678594, SSL errmsg:
error:14209102:SSL routines:tls_early_post_process_client_hello:unsupported
protocol2021-07-09T10:50:03.401161+08:00 213499 [Note] Bad handshake2021-07-
09T10:53:44.458404+08:00 213688 [Warning] SSL errno: 337678594, SSL errmsg:
error:14209102:SSL routines:tls_early_post_process_client_hello:unsupported
protocol2021-07-09T10:53:44.458475+08:00 213688 [Note] Bad handshake
```

- 2. 从报错信息 `unsupported protocol` 可以看出，很可能和 TLS 版本相关，使用如下命令，分别查看 GaussDB(for MySQL)和自建 MySQL 的 TLS 版本。

```
show variables like '%tls_version%';
```

发现 GaussDB(for MySQL)为 TLS v1.2 版本，自建 MySQL 为 TLS v1.1 版本，存在差异。进一步确认客户端 TLS 版本，与自建 MySQL 一致，因此出现连接自建 MySQL 成功，连接云上 GaussDB(for MySQL)失败。

### 解决方案

客户端升级 TLS 版本到 TLSv1.2。

如果使用官方 JDBC 驱动 `mysql-connector/J`，可参考[官方文档](#)，配置方法：

```
TLS versions: The allowable versions of TLS protocol can be restricted using the connection properties tlsVersions and, for X DevAPI connections and for release 8.0.19 and later, xdevapi.tls-versions (when xdevapi.tls-versions is not specified, it takes up the value of tlsVersions). If no such restrictions have been specified, Connector/J attempts to connect to the server with the TLSv1.2 and TLSv1.3.
```

## 5.2.5 GaussDB(for MySQL)建立连接慢导致客户端超时

### 场景描述

业务在高峰期时，客户端经常出现向 MySQL 建立连接超时，导致系统登录需要十几秒。

### 原因分析

- 1. 查看 GaussDB(for MySQL)的错误日志，观察是否有如下信息：`connection xxx is established slowly`。示例：

The screenshot shows a series of MySQL error log entries. Each entry starts with a timestamp and the word 'Warning'. The main message in each entry is 'This connection (xxxxx) is established slowly (xxxxx)'. To the right of the main message, there is a detailed error description: 'Got an error reading communication packets (launch in 310623 and 310548 microseconds respectively)'. The 'xxxxx' values represent connection IDs and slow log times.

有上述日志，说明存在某些连接超过一定时间仍未被 MySQL 处理，客户端的超时时间大于该时间，就会报错。

- 2. 进一步查看线程池配置（默认开启），可以在控制台查看。

参数名称	是否可修改	重固有有效	值	参数范围
<code>threadpool_oversubscribe</code>	是	否	3	[1-50]
<code>threadpool_size</code>	是	否	1	[1-512]
<code>threadpool_stall_limit</code>	是	否	500	[10-4,294,967,295]

可以看出，`threadpool_size` 为 1，`threadpool_stall_limit` 为 500ms，`threadpool_oversubscribe` 为 3，线程池处理连接等待的时间主要与上述 3 个参数相关：

- 当线程池所有线程在忙碌，线程池中的调度线程会每隔 500ms (`threadpool_stall_limit`) 创建一个新线程，所以此时，每个线程组平均每 500ms 才能处理一个新的连接。如果队列太长，很可能导致客户端超时；
- 所有线程都在忙碌是指，工作线程达到线程池总线程数，在大量建立连接时，总线程数计算方法：`threadpool_size*(threadpool_oversubscribe+1)`

## 解决方案

对于存在大量新建连接，建议调大 `threadpool_oversubscribe` 增加线程总数。

减少线程重复创建与销毁部分的开销，提高性能，同时它也限制了 MySQL 的 `runing` 线程数，关键时刻可以保护系统，防止雪崩。

正常情况下，线程池适用于大量短连接的场景，如果客户是长连接，并且连接数量不多（客户端使用了连接池等情况），线程池的作用不大，此时调整 `threadpool_size` 和 `threadpool_oversubscribe` 两个参数扩大线程总数，或者直接关闭线程池。

## 5.2.6 连接数据库报错 Access denied

### 场景描述

客户端连接数据库异常，返回错误：`Error 1045: Access denied for user xxx`

### 处理方法

#### 1. 连接了错误的主机

问题原因：业务连接了错误的数据库主机，该主机上相应用户或客户端 IP 没有权限访问。

解决方案：仔细检查要连接的数据库主机名，确保正确。

#### 2. 用户不存在

问题原因：客户端连接时，使用的用户不存在。

解决方案：

- 使用管理员帐户登录数据库，执行如下命令检查目标用户是否存在。

```
SELECT User FROM mysql.user WHERE User='xxx';
```

- 如果用户不存在，创建相应用户。

```
CREATE USER 'xxx'@'xxxxxxx' IDENTIFIED BY 'xxxx';
```

#### 3. 用户存在，但客户端 IP 无访问权限

问题原因：客户端使用的用户存在，但是客户端 IP 没有该数据库的访问权限。

解决方案：

- 使用管理员帐户登录数据库，执行如下命令，检查目标用户允许哪些客户端 IP 连接。

```
SELECT Host, User FROM mysql.user WHERE User='xxx';
```

- 如果上述查询出的 Host 不包含客户端 IP 所在网段，则需要赋予相应访问权限。例如，赋予 test 用户 192.168.0 网段访问权限。

```
GRANT ALL PRIVILEGES ON *.* TO 'root'@'192.168.0.%' IDENTIFIED BY 'password' WITH GRANT OPTION;  
FLUSH PRIVILEGES;
```

#### 4. 密码错误

问题原因：用户对应的密码错误，或忘记密码

解决方案：



- 确定目标密码是否错误，由于密码用于身份验证，因此无法从 MySQL 以明文形式读取用户密码，但可以将密码的哈希字符串与目标密码的“PASSWORD”函数值进行比较，确定目标密码是否正确，示例 SQL 语句：

```
mysql> SELECT Host, User, authentication_string, PASSWORD('12345') FROM
mysql.user WHERE User='test';
+-----+-----+-----+-----+
| Host      | User | authentication_string |          |
| PASSWORD('12345') |          |
+-----+-----+-----+-----+
| %         | test | *6A23DC5E7446019DC9C1778554ED87BE6BA61041 |
| *00A51F3F48415C7D4E8908980D443C29C69B60C9 |
+-----+-----+-----+-----+
2 rows in set, 1 warning (0.00 sec)
```

从上面例子可以看出，PASSWORD('12345')的哈希值与 authentication\_string 列不匹配，这表明目标密码“12345”是错误的。

- 如果需要重置用户密码，参考如下 SQL 语句：

```
set password for 'test'@'%' = 'new_password';
```

#### 5. 密码包含特殊字符被 Bash 转义

问题原因：Linux 默认的 Bash 环境下，使用命令行连接数据库，用户密码中包含特殊字符会被环境转义，导致密码失效。

例如，在 Bash 环境下，用户 test 的密码为“test\$123”，使用命令 `mysql-hxxx -u test -ptest$123`，连接数据库会报错 `ERROR 1045 (28000): Access denied.`

解决方案：通过用单引号将密码括起来，防止 Bash 解释特殊字符。

```
mysql -hxxx -u test -p'test$123'
```

#### 6. 用户设置了 REQUIRE SSL，但客户端使用非 SSL 连接

排查思路：

- 排查报错用户名是否强制使用 SSL 连接，执行：`show create user 'xxx'`，如果出现“REQUIRE SSL”属性，说明该用户必须使用 SSL 连接。
- 排查是否使用过如下类似语句给用户授权。

```
GRANT ALL PRIVILEGES ON . TO 'ssluser'@'localhost' IDENTIFIED BY
'password' REQUIRE SSL;
```

- 检查目标用户的 ssl\_type 值，如果不为空，说明该用户需要使用 SSL 连接。

```
SELECT User, Host, ssl_type FROM mysql.user WHERE User='xxx';
```

解决方案：

- 客户端使用 SSL 方式数据库连接。
- 去除用户 SSL 连接权限，参考命令：`ALTER USER 'username'@'host' REQUIRE NONE;`

## 5.2.7 mariadb-connector SSL 方式连接数据库失败

### 场景描述

使用 jdbc 无法连接数据库，报如下错误：

```
unable to find certification path to requested target
```

```
RELEASE_jar1/:?]  
at com.huawei.devsore.datasource.jdbc.core.router.DefaultClusterRouterExecutor.tryExecute(DefaultClusterRouterExecutor.java:44) ~[devsore-datasource-1.2.2-RELEASE_jar1/:?]  
at com.huawei.devsore.datasource.jdbc.core.router.AbstractRouterExecutor.tryExecute(AbstractRouterExecutor.java:82) ~[devsore-datasource-1.2.2-RELEASE_jar1/:?]  
at com.huawei.devsore.datasource.jdbc.adapter.AbstractDatabaseMetaDataAdapter.getDatabaseProductName(AbstractDatabaseMetaDataAdapter.java:357) ~[devsore-datasource-1.2.2-RELEASE_jar1/:?]  
at org.springframework.jdbc.support.JdbcUtils.extractDatabaseMetaData(JdbcUtils.java:360) ~[spring-jdbc-5.3.21.jar1/:5.3.21]  
... 53 more  
Caused by: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target  
at sun.security.provider.certpath.SunCertPathBuilder.build(SunCertPathBuilder.java:141) ~[?:1.8.0_272]  
at sun.security.provider.certpath.SunCertPathBuilder.engineBuild(SunCertPathBuilder.java:126) ~[?:1.8.0_272]  
at java.security.cert.CertPathBuilder.build(CertPathBuilder.java:288) ~[?:1.8.0_272]  
at sun.security.validator.PKIXValidator.doBuild(PKIXValidator.java:451) ~[?:1.8.0_272]  
at sun.security.validator.PKIXValidator.engineValidate(PKIXValidator.java:323) ~[?:1.8.0_272]  
at sun.security.validator.Validator.validate(Validator.java:271) ~[?:1.8.0_272]  
at sun.security.ssl.X509TrustManagerImpl.validate(X509TrustManagerImpl.java:315) ~[?:1.8.0_272]  
at sun.security.ssl.X509TrustManagerImpl.checkTrusted(X509TrustManagerImpl.java:234) ~[?:1.8.0_272]  
at sun.security.ssl.X509TrustManagerImpl.checkServerTrusted(X509TrustManagerImpl.java:118) ~[?:1.8.0_272]  
at org.mariadb.jdbc.internal.protocol.tls.MariaDbX509TrustManager.checkServerTrusted(MariaDbX509TrustManager.java:243) ~[mariadb-java-client-2.7.5.jar1/:?]  
] ]  
at sun.security.ssl.AbstractTrustManagerWrapper.checkServerTrusted(SSLCertContextImpl.java:1258) ~[?:1.8.0_272]  
at sun.security.ssl.CertificateMessage$T12CertificateConsumer.checkServerCerts(CertificateMessage.java:538) ~[?:1.8.0_272]  
at sun.security.ssl.CertificateMessage$T12CertificateConsumer.onCertificate(CertificateMessage.java:473) ~[?:1.8.0_272]  
at sun.security.ssl.SSLHandshake.consume(SSLHandshake.java:377) ~[?:1.8.0_272]  
at sun.security.ssl.HandshakeContext.dispatch(HandshakeContext.java:444) ~[?:1.8.0_272]  
at sun.security.ssl.HandshakeContext.dispatch(HandshakeContext.java:422) ~[?:1.8.0_272]  
at sun.security.ssl.TransportContext.dispatch(TransportContext.java:162) ~[?:1.8.0_272]  
at sun.security.ssl.SSLTransport.decode(SSLTransport.java:140) ~[?:1.8.0_272]  
at sun.security.ssl.SSLSocketImpl.decode(SSLSocketImpl.java:1142) ~[?:1.8.0_272]  
at sun.security.ssl.SSLSocketImpl.readHandshakeRecord(SSLSocketImpl.java:1054) ~[?:1.8.0_272]  
at sun.security.ssl.SSLSocketImpl.startHandshake(SSLSocketImpl.java:394) ~[?:1.8.0_272]  
at org.mariadb.jdbc.internal.protocol.AbstractConnectProtocol.sslWrapper(AbstractConnectProtocol.java:661) ~[mariadb-java-client-2.7.5.jar1/:?]  
at org.mariadb.jdbc.internal.protocol.AbstractConnectProtocol.createConnection(AbstractConnectProtocol.java:544) ~[mariadb-java-client-2.7.5.jar1/:?]  
at org.mariadb.jdbc.internal.protocol.AbstractConnectProtocol.connectWithoutProxy(AbstractConnectProtocol.java:1389) ~[mariadb-java-client-2.7.5.jar1/:?]  
at org.mariadb.jdbc.internal.util.Utils.retrieveProxy(Utils.java:95) ~[mariadb-java-client-2.7.5.jar1/:?]  
at org.mariadb.jdbc.MariaDbConnection.newConnection(MariaDbConnection.java:158) ~[mariadb-java-client-2.7.5.jar1/:?]  
at org.mariadb.jdbc.Driver.connect(Driver.java:89) ~[mariadb-java-client-2.7.5.jar1/:?]  
at com.zaxxer.hikari.util.DriverDataSource.getConnection(DriverDataSource.java:138) ~[HikariCP-4.0.3.jar1/:?]  
at com.zaxxer.hikari.pool.PoolBase.newConnection(PoolBase.java:364) ~[HikariCP-4.0.3.jar1/:?]  
at com.zaxxer.hikari.pool.PoolBase.newPoolEntry(PoolBase.java:266) ~[HikariCP-4.0.3.jar1/:?]  
at com.zaxxer.hikari.pool.HikariPool.createPoolEntry(HikariPool.java:476) ~[HikariCP-4.0.3.jar1/:?]  
at com.zaxxer.hikari.pool.HikariPool.checkFailFast(HikariPool.java:561) ~[HikariCP-4.0.3.jar1/:?]  
at com.zaxxer.hikari.pool.HikariPool.<init>(HikariPool.java:115) ~[HikariCP-4.0.3.jar1/:?]  
at com.zaxxer.hikari.HikariDataSource.getConnection(HikariDataSource.java:112) ~[HikariCP-4.0.3.jar1/:?]
```

### 原因分析

从错误截图中可以看出，使用的是 mariadb 的 jar 包，而非 MySQL 的官方驱动包，而 mariadb 与官方的使用方法略有区别。

### 解决方案

对于 mariadb 的连接串应该为：

```
String url =  
"jdbc:mysql://xxx.xxx.xxx.xxx:xxxx/mysql?useSsl=true&serverSslCert=D:\\ca.pem&disableSslHostnameVerification=true";
```

注意：GaussDB(for MySQL)实例不支持 hostname 校验，因此需要设置 `disableSslHostnameVerification=true`，不同 mariadb jar 包版本设置方式不同，可查看对应版本的[使用说明](#)。

## 5.2.8 使用 root 帐号连接数据库失败

### 场景描述

使用 root 帐号连接数据库失败。

### 原因分析

1. 查看内核日志 `error.log`，确认是否有拒绝连接的日志。

- 查看 root 权限，发现有两个 root 帐号，其中一个 root 限制 host 必须是 IP 必须是 192 开头。

```
mysql> select * from mysql.user where user='root'\G;
***** 1. row *****
      Host: %
      User: root
      Select_priv: Y
      Insert_priv: Y
      Update_priv: Y
      Delete_priv: Y
      Create_priv: Y
      Drop_priv: Y
      Reload_priv: Y
      Shutdown_priv: N
      Process_priv: Y
      File_priv: N
      Grant_priv: Y
      References_priv: Y
      Index_priv: Y
      Alter_priv: Y
      Show_db_priv: Y
      Super_priv: N
      Create_tmp_table_priv: Y
      password_lifetime: NULL
      account_locked: N
***** 2. row *****
      Host: 192.%
      User: root
      Select_priv: Y
      Insert_priv: Y
      Update_priv: Y
      Delete_priv: Y
      Create_priv: Y
      Drop_priv: Y
      Reload_priv: Y
```

## 解决方案

联系技术支持协助删除多余的 root 帐号。

## 5.2.9 客户端连接实例后会自动断开

### 故障描述

MySQL 客户端连接实例后，会自动断开，报错信息：“ERROR 2013: Lost connection to MySQL server during query”。

### 解决方案

ERROR 2013 是 MySQL 常见错误，一般为配置错误导致。

- “wait\_timeout”：服务器关闭非交互连接之前等待活动的秒数。
- “interactive\_timeout”：服务器关闭交互连接之前等待活动的秒数。

步骤 1 查看实例状态是否处于正常状态。

经查看实例状态正常，继续排查其他问题。

步骤 2 查看错误日志。

步骤 3 使用 MySQL 命令行客户端连接数据库，执行 status 命令，确认数据库实例是否频繁重启。

```
mysql> status
-----
mysql Ver 14.14 Distrib 5.6.34, for Linux (x86_64) using EditLine wrapper

Connection id:          16288
Current database:
Current user:           root@192.168.0.5
SSL:                    Not in use
Current pager:          stdout
Using outfile:          ''
Using delimiter:        ;
Server version:         5.6.34-log MySQL Community Server (GPL)
Protocol version:      10
Connection:             192.168.0.24 via TCP/IP
Server characterset:    utf8
Db characterset:        utf8
Client characterset:    utf8
Conn. characterset:     utf8
TCP port:               8635
Uptime:                 5 hours 5 min 34 sec

Threads: 2  Questions: 62118  Slow queries: 0  Opens: 70  Flush tables: 2  Open tables: 0  Queries per second avg: 3.388
-----
```

Uptime 代表实例的运行时间，从排查结果可知，数据库并没有频繁重启，因而，客户端连接被断开，不是因数据库重启引起的。

步骤 4 查看“wait\_timeout”和“interactive\_timeout”参数设置，MySQL 会自动断开超时的空连接。

步骤 5 您可根据实际应用需求量，修改“wait\_timeout”和“interactive\_timeout”参数值，无需重启实例。

步骤 6 恢复结果确认，等到 10 分钟左右，再次执行 show databases 命令，确认连接是否正常。

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

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

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

mysql>
```

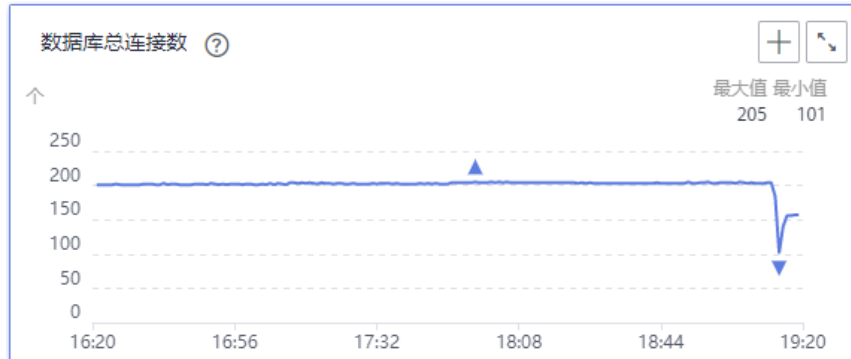
如图所示，说明连接正常。

----结束

## 5.2.10 istio-citadel 证书机制导致每隔 45 天出现断连

### 场景描述

业务侧发现数据库每隔 45 天同一时间，多台数据库实例的连接数骤降。查看服务端连接数监控指标如下：



客户端出现大量报错如下：

```
[2022-11-22T10:34:23.248977Z] [http-nio-000-exec-5-8] [ERROR] [druid.sql.Statement:149] - {conn=110005, stmt=883289} execute error. select 1 from dual
java.sql.SQLException: (conn=12518697) unexpected end of stream, read 0 bytes from 4 (socket was closed by server)
at org.mariadb.jdbc.internal.util.exceptions.ExceptionFactory.create(ExceptionFactory.java:73)
at org.mariadb.jdbc.internal.util.exceptions.ExceptionFactory.create(ExceptionFactory.java:153)
at org.mariadb.jdbc.MariaDbStatement.executeExceptionEpilogue(MariaDbStatement.java:274)
at org.mariadb.jdbc.MariaDbStatement.executeInternal(MariaDbStatement.java:363)
at org.mariadb.jdbc.MariaDbStatement.executeQuery(MariaDbStatement.java:617)
```

### 原因分析

1. 排查业务侧是否有间隔 45 天的定时任务。
2. 客户端如果使用了 istio 等证书加密机制，分析证书相关日志，是否有如下类似信息。如果有，说明是证书过期导致。

```
-----
2021-11-22T10:34:23.248977Z warn istio.io/istio/security/pkg/k8s/controller/workloadsecret.go:236: watch of *v1.Secret
ended with: too old resource version: 228865253 (228865325)
2021-11-22T11:20:50.632458Z info rootCertRotator Check and rotate root cert.
2021-11-22T11:20:50.639274Z info rootCertRotator Root cert is not about to expire, skipping root cert rotation.
2021-11-22T12:11:55.338195Z warn istio.io/istio/security/pkg/k8s/controller/workloadsecret.go:236: watch of *v1.Secret
ended with: too old resource version: 228884272 (228885539)
2021-11-22T12:20:50.632470Z info rootCertRotator Check and rotate root cert.
2021-11-22T12:20:50.635853Z info rootCertRotator Root cert is not about to expire, skipping root cert rotation.
2021-11-22T13:12:05.395613Z warn istio.io/istio/security/pkg/k8s/controller/workloadsecret.go:236: watch of *v1.Secret
```

客户端 istio-citadel 证书每隔 45 天过期，导致主动发起数据库断连请求。

### 解决方案

- 客户端设置合理的 istio-citadel 证书过期时间，并在过期发生时，主动规避操作。
- 客户端排查是否有其他证书过期问题。

## 5.3 SQL 类

### 5.3.1 建表时 timestamp 字段默认值无效

#### 场景描述

客户执行一个建表 SQL 语句失败，详细 SQL 语句及报错如下：

```
CREATE TABLE cluster_membership
(
...
session_start TIMESTAMP DEFAULT '1970-01-01 00:00:01',
...
);
```

执行失败，失败原因：ERROR 1067: Invalid default value for 'session\_start'

## 原因分析

表字段类型是 `TIMESTAMP` 类型，

关于 `timestamp` 字段：MySQL 会把该字段插入的值从当前时区转换成 UTC 时间（世界标准时间）存储，查询时，又将其从 UTC 时间转化为当前时区时间返回

1. `timestamp` 类型字段的时间范围：'1970-01-01 00:00:01' UTC -- '2038-01-19 03:14:07' UTC，详见[官方文档](#)；
2. 使用如下命令，查看当前的时区：

```
show variables like "%zone%";
```

3. 故障场景中使用的是 `utc+8` 时区，如下图，所以 `timestamp` 字段默认值需要加 8 小时才是有效范围，有效支持的范围是从 1970-01-01 08:00:01 开始；

```
mysql> show variables like "%zone%";
+-----+-----+
| Variable_name | Value |
+-----+-----+
| system_time_zone |      |
| time_zone       | +08:00 |
+-----+-----+
2 rows in set, 1 warning (0.00 sec)
```

## 解决方案

执行命令，修改 `timestamp` 字段参数默认值。

```
session_start TIMESTAMP DEFAULT '1970-01-01 08:00:01',
```

## 5.3.2 索引长度限制导致修改 `varchar` 长度失败

### 场景描述

执行 `alter table` 修改表结构失败，报错如下：

```
Specified key was too long; max key length is 3072 bytes
```

### 原因分析

- 在“`innodb_large_prefix`”设置为 `off` 的情况下，InnoDB 表的单字段索引的最大字段长度不能超过 767 字节，联合索引的每个字段的长度不能超过 767 字节，且所有字段长度合计不能超过 3072 字节。

- 当“innodb\_large\_prefix”设置为 on 时，单字段索引最大长度可为 3072 字节，联合索引合计最大长度可为 3072 字节。
- 索引长度与字符集相关。使用 utf8 字符集时，一个字符占用三个字节，在“innodb\_large\_prefix”参数设置为 on 情况下，索引的所有字段的长度合计最大为 1072 个字符。

查看表结构如下：

```
CREATE TABLE `xxxxx` (  
.....  
`subscription_type` varchar(64) NOT NULL DEFAULT 'DEVICE_EXCEPTION' COMMENT '订阅类型',  
`auth_key` varchar(255) DEFAULT '' COMMENT '签名，接口请求头会根据这个值增加 token',  
`create_time` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',  
`update_time` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE  
CURRENT_TIMESTAMP COMMENT '修改时间',  
PRIMARY KEY (`id`) USING BTREE,  
UNIQUE KEY `enterprise_id` (`subscription_type`,`enterprise_id`,`callback_url`)  
USING BTREE)  
ENGINE=InnoDB AUTO_INCREMENT=1039 DEFAULT CHARSET=utf8 ROW_FORMAT=DYNAMIC
```

该表使用了 utf8 字符集，一个字符占用三个字节。联合索引“enterprise\_id”包含了“callback\_url”字段，如果执行 DDL 操作将“callback\_url”修改为 varchar(1024)，会超出联合索引最大长度限制，所以报错。

## 解决方案

MySQL 机约束束，建议修改索引或字段长度。

### 5.3.3 delete 大表数据后，再查询同一张表时出现慢 SQL

#### 场景描述

一次性删除多条宽列数据（每条记录数据长度在 1GB 左右），再次对同一张表进行增删改查时均执行缓慢，20 分钟左右后恢复正常。

#### 场景案例

1. 假定 max\_allowed\_packet 参数大小为 1073741824。
2. 创建表。

```
CREATE TABLE IF NOT EXISTS zstest1  
(  
id int PRIMARY KEY not null,  
c_longtext LONGTEXT  
);
```

3. 向表中插入数据。

```
insert into zstest1 values(1, repeat('a', 1073741800));  
insert into zstest1 values(2, repeat('a', 1073741800));  
insert into zstest1 values(3, repeat('a', 1073741800));  
insert into zstest1 values(4, repeat('a', 1073741800));  
insert into zstest1 values(5, repeat('a', 1073741800));  
insert into zstest1 values(6, repeat('a', 1073741800));
```

```
insert into zstest1 values(7, repeat('a', 1073741800));
insert into zstest1 values(8, repeat('a', 1073741800));
insert into zstest1 values(9, repeat('a', 1073741800));
insert into zstest1 values(10, repeat('a', 1073741800));
```

#### 4. 删除数据。

```
delete from zstest1;
```

#### 5. 执行查询语句。

```
select id from zstest1; //执行缓慢
```

## 原因分析

执行完 `delete` 操作后，后台 `purge` 线程会去清理标记为 `delete mark` 的记录。由于当前删除的数据量较大，`purge` 遍历释放 `page` 的过程中会去获取 `page` 所在索引根节点的 `SX` 锁，导致 `select` 语句无法获取到根节点 `page` 的 `rw-lock`，一直在等待。

## 解决方案

- 该场景为正常现象，等待 `purge` 操作完成后即可恢复正常。
- 扩大实例规格，提高 `purge` 效率。
- 调整优化业务，避免突然删除大量数据。如果需要删除表中所有数据，建议使用 `truncate table`。

## 5.3.4 更新 emoji 表情数据报错 Error 1366

### 场景描述

业务插入或更新带有 emoji 表情的数据时，报错 `Error 1366`。

```
java.sql.SQLException: Incorrect string value: '\xF0\x9F\x90\xB0\xE5\xA4...' for
column 'username' at row 1 ;
uncategorized SQLException for SQL []; SQL state [HY000]; error code [1366];
Incorrect string value: '\xF0\x9F\x90\xB0\xE5\xA4...' for column 'username' at row
1;
```

### 原因分析

原因是字符集配置有误：

- emoji 表情为特殊字符，需要 4 字节字符集存储。
- 该问题场景下，数据库字符集为 `utf-8`，它最多支持 3 个字节；`utf8mb4` 才是支持 4 个字节的字符集；

### 解决方案

1. 将存储 emoji 表情的字段的字符集修改为 `utf8mb4`。

如果涉及的表和字段比较多，建议把对应表、数据库的编码也设置为 `utf8mb4`。参考命令：

```
ALTER DATABASE database_name CHARACTER SET= utf8mb4 COLLATE=
utf8mb4_unicode_ci;
```



```
ALTER TABLE table_name CONVERTTOCHARACTER SET utf8mb4  
COLLATE utf8mb4_unicode_ci;
```

```
ALTER TABLE table_name MODIFY 字段名 VARCHAR(128) CHARSET  
utf8mb4 COLLATE utf8mb4_unicode_ci;
```

2. 若对应字段的字符集已经是 utf8mb4，则为客户端或 MySQL 服务端字符集转换问题，将客户端和 MySQL 服务端的字符集都设置为 utf8mb4。

### 5.3.5 存储过程和相关表字符集不一致导致执行缓慢

#### 场景描述

GaussDB(for MySQL)存储过程执行很慢，处理少量数据耗时 1min 以上，而单独执行存储过程中的 SQL 语句却很快。

#### 原因分析

存储过程和相关表、库的字符集不一致，导致查询结果存在大量字符转换，从而执行缓慢。

排查过程：

使用如下命令查看存储过程和相关表的定义，观察存储过程和表的字符集是否一致。

```
SHOW CREATE PROCEDURE xxx;  
SHOW CREATE TABLE xxx
```

示例：

```
mysql> SHOW CREATE PROCEDURE testProc \G  
***** 1. row *****  
Procedure: showstuscore  
sql_mode: STRICT_TRANS_TABLES,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION  
Create Procedure: xxx  
character_set_client: utf8mb4  
collation_connection: utf8mb4_general_ci  
Database Collation: utf8_general_ci  
1 row in set (0.01 sec)
```

可以看出，上述存储过程 collation 为 utf8mb4\_general\_ci，而所在库 collation 默认为 utf8\_general\_ci，collation 值不一致，容易导致性能问题。

#### 解决方案

将存储过程和相关表、库的字符集改成一致后，执行缓慢问题解决。

### 5.3.6 报错 ERROR [1412]的解决方法

#### 场景描述

连接 GaussDB(for MySQL)执行 SQL 时，出现如下报错：

```
ERROR[1412]:Table definition has changed, please retry transaction``
```

## 原因分析

启动一致性快照事务后，其他会话（session）执行 DDL 语句导致。问题复现步骤：

1. 会话 1 启动一致性快照事务。

```
mysql> start transaction with consistent snapshot;  
Query OK, 0 rows affected (0.00 sec)
```

2. 会话 2 执行 DDL 操作，修改表结构。

```
mysql> alter table t_sec_user add test int;  
Query OK, 0 rows affected (0.01 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

3. 会话 1 执行普通的查询语句。

```
mysql> select count(*) from t_sec_user;  
ERROR 1412 (HY000): Table definition has changed, please retry transaction  
mysql>
```

也可以通过 Binlog 或者审计日志，分析业务侧是否有同一个表 DDL 和一致性快照事务一起执行的情况。

## 解决方案

若经排查，是由上述原因引起的报错，需要业务侧避免同一个表的 DDL 语句和一致性快照事务同时执行。

### 5.3.7 存在外键的表无法删除

#### 场景描述

删除 MySQL 表时，如果表中有外键（foreign key），会出现如下报错，且和用户权限无关：

```
ERROR 1451 (23000): Cannot delete or update parent row: a foreign key constraint  
fails .....
```

#### 原因分析

这个表和其他表有外键关系，在 MySQL 中，设置了外键关联，会造成无法更新或删除数据，避免破坏外键的约束。

可以通过设置变量 FOREIGN\_KEY\_CHECKS 值为 off，来关闭上述机制，详见[官方文档](#)。

#### 解决方案

通过设置变量 FOREIGN\_KEY\_CHECKS 值为 off，来关闭上述机制：

```
set session foreign_key_checks=off;  
drop table table_name;
```

### 5.3.8 GROUP\_CONCAT 结果不符合预期

#### 场景描述

SQL 语句中使用 GROUP\_CONCAT()函数时，出现结果不符合预期的情况。

#### 原因分析

GROUP\_CONCAT()函数返回一个字符串结果，该结果由分组中的值连接组合而成。需要注意的是：这个函数的结果长度是有限制的，由 group\_concat\_max\_len 参数决定。

示例：

```
mysql> show variables like 'group_concat_max_len';
+-----+-----+
| Variable_name      | Value |
+-----+-----+
| group_concat_max_len | 1024  |
+-----+-----+
1 row in set (0.01 sec)

mysql> select GROUP_CONCAT(c1,c2,c3) from dis;
+-----+
| GROUP_CONCAT(c1,c2,c3) |
+-----+
| 111,222,322           |
+-----+
```

```
mysql> set session group_concat_max_len=8;
Query OK, 0 rows affected (0.00 sec)

mysql> show variables like 'group_concat_max_len';
+-----+-----+
| Variable_name      | Value |
+-----+-----+
| group_concat_max_len | 8     |
+-----+-----+
1 row in set (0.01 sec)

mysql> select GROUP_CONCAT(c1,c2,c3) from dis;
+-----+
| GROUP_CONCAT(c1,c2,c3) |
+-----+
| 111,222,                |
+-----+
```

#### 解决方案

调整 group\_concat\_max\_len 参数值，适配 GROUP\_CONCAT()函数的结果长度。

## 5.3.9 创建二级索引报错 Too many keys specified

### 场景描述

创建二级索引失败，报错：Too many keys specified; max 64 keys allowed.

### 故障分析

MySQL 对 InnoDB 每张表的二级索引的数量上限有限制，限制上限为 64 个，超过限制会报错 “Too many keys specified; max 64 keys allowed”。详见[官方文档](#)。

[MySQL 8.0 Reference Manual / The InnoDB Storage Engine / InnoDB Limits](#)

### 15.22 InnoDB Limits

This section describes limits for InnoDB tables, indexes, tablespaces, and other aspects of the InnoDB storage engine.

- A table can contain a maximum of 1017 columns. Virtual generated columns are included in this limit.
- A table can contain a maximum of 64 secondary indexes.
- The index key prefix length limit is 3072 bytes for InnoDB tables that use DYNAMIC or COMPRESSED row format.

### 解决方案

MySQL 机制导致，建议优化业务，避免单表创建过多索引。

#### 📖 说明

InnoDB 表的其他限制：

1. 一个表最多可以包含 1017 列（包含虚拟生成列）。
2. InnoDB 对于使用 DYNAMIC 或 COMPRESSED 行格式的表，索引键前缀长度限制为 3072 字节。
3. 多列索引最多允许 16 列，超过限制会报错。

## 5.3.10 distinct 与 group by 优化

### 场景描述

使用 distinct 或 group by 的语句执行比较慢。

### 原因分析

大部分情况下，distinct 是可以转化成等价的 group by 语句。在 MySQL 中，distinct 关键字的主要作用就是去重过滤。

distinct 进行去重的原理是先进分组操作，然后从每组数据中取一条返回给客户端，分组时有两种场景：

- distinct 的字段全部包含于同一索引：该场景下 MySQL 直接使用索引对数据进行分组，然后从每组数据中取一条数据返回。

- distinct 字段未全部包含于索引：该场景下索引不能满足去重分组需要，会用到临时表（首先将满足条件的数据写入临时表中，然后在临时表中对数据进行分组，返回合适的的数据）。因为使用临时表会带来额外的开销，所以一般情况下性能会较差。

综上，在使用 distinct 或 group by 的时候，尽量在合理的情况下设置可以包含所有依赖字段的索引，优化示例：

- 没有合适索引，导致需要用到临时表。

```
mysql> show create table test;
+-----+
| Table | Create Table
+-----+
| test  | CREATE TABLE `test` (
  `id` int NOT NULL,
  `c1` int DEFAULT NULL,
  `c2` int DEFAULT NULL,
  `c3` int DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `c1` (`c1`),
  KEY `c2` (`c2`),
  KEY `c3` (`c3`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 |
+-----+
1 row in set (0.00 sec)

mysql> explain select distinct c1,c2,c3 from test;
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1  | SIMPLE      | test | NULL       | ALL | NULL          | NULL | NULL    | NULL | 1    | 100.00   | Using temporary
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set, 1 warning (0.00 sec)

mysql> explain select c1,c2,c3 from test group by c1,c2,c3;
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1  | SIMPLE      | test | NULL       | ALL | NULL          | NULL | NULL    | NULL | 1    | 100.00   | Using temporary
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set, 1 warning (0.03 sec)
```

- 有合适的索引，不会使用临时表，直接走索引。

```
mysql> alter table test add key(c1,c2,c3);
Query OK, 0 rows affected (0.10 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> show create table test;
+-----+
| Table | Create Table
+-----+
| test  | CREATE TABLE `test` (
  `id` int NOT NULL,
  `c1` int DEFAULT NULL,
  `c2` int DEFAULT NULL,
  `c3` int DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `c1` (`c1`),
  KEY `c2` (`c2`),
  KEY `c3` (`c3`),
  KEY `c1_2` (`c1`,`c2`,`c3`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 |
+-----+

mysql> explain select c1,c2,c3 from test group by c1,c2,c3;
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1  | SIMPLE      | test | NULL       | index | c1_2          | c1_2 | 15      | NULL | 1    | 100.00   | Using index
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set, 1 warning (0.00 sec)

mysql> explain select distinct c1,c2,c3 from test;
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1  | SIMPLE      | test | NULL       | index | c1_2          | c1_2 | 15      | NULL | 1    | 100.00   | Using index
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set, 1 warning (0.01 sec)
```

## 解决方案

在使用 `distinct` 或 `group by` 的时候，尽量在合理的情况下，创建可以包含所有依赖字段的索引。

### 5.3.11 为什么有时候用浮点数做等值比较查不到数据

#### 原因分析

浮点数的等值比较问题是一种常见的浮点数问题。因为在计算机中，浮点数存储的是近似值而不是精确值，所以等值比较、数学运算等场景很容易出现预期外的情况。

MySQL 中涉及浮点数的类型有 `float` 和 `double`。如下示例中遇到的问题：

```
mysql> create table f(fnum float, dnum double);
Query OK, 0 rows affected (0.26 sec)

mysql> insert into f values(1.1, 1.2);
Query OK, 1 row affected (0.07 sec)

mysql> insert into f values(2.1, 2.2);
Query OK, 1 row affected (0.00 sec)

mysql> insert into f values(2.1, 3.2);
Query OK, 1 row affected (0.00 sec)

mysql> insert into f values(3.1, 3.2);
Query OK, 1 row affected (0.03 sec)

mysql> select * from f;
+-----+-----+
| fnum | dnum |
+-----+-----+
| 1.1 | 1.2 |
| 2.1 | 2.2 |
| 2.1 | 3.2 |
| 3.1 | 3.2 |
+-----+-----+
4 rows in set (0.00 sec)

mysql> select * from f where fnum = 1.1;
Empty set (0.03 sec)

mysql> select * from f where fnum < 2;
+-----+-----+
| fnum | dnum |
+-----+-----+
| 1.1 | 1.2 |
+-----+-----+
1 row in set (0.00 sec)
```

## 解决方案

1. 使用精度的方法处理，使用字段与数值的差值的绝对值小于可接受的精度的方法。示例：

```
mysql> select * from f where fnum = 0.01;
Empty set (0.00 sec)

mysql> select * from f where abs(fnum - 1.1) < 0.01;
+-----+-----+
| fnum | dnum |
+-----+-----+
| 1.1 | 1.2 |
+-----+-----+
1 row in set (0.00 sec)
```

2. 使用定点数类型(DECIMAL)取代浮点数类型，示例：

```
mysql> create table d(d1 DECIMAL(5,2), d2 DECIMAL(5,2));
Query OK, 0 rows affected (0.09 sec)

mysql> insert into d values(1.1, 1.2);
Query OK, 1 row affected (0.02 sec)

mysql> insert into d values(2.1, 2.2);
Query OK, 1 row affected (0.01 sec)

mysql> insert into d values(3.1, 3.2);
Query OK, 1 row affected (0.01 sec)

mysql> select * from d;
+-----+-----+
| d1  | d2  |
+-----+-----+
| 1.10 | 1.20 |
| 2.10 | 2.20 |
| 3.10 | 3.20 |
+-----+-----+
3 rows in set (0.00 sec)

mysql> select * from d where d1 = 1.1;
+-----+-----+
| d1  | d2  |
+-----+-----+
| 1.10 | 1.20 |
+-----+-----+
1 row in set (0.00 sec)
```

### 5.3.12 表空间膨胀问题

#### 场景描述

在使用 GaussDB(for MySQL)过程中，经常遇到表空间膨胀问题，例如：表中只有 11774 行数据，表空间却占用 49.9GB，将该表导出到本地只有 800M。

## 原因分析

### 场景 1: DRS 全量迁移阶段并行迁移导致

原因: DRS 在全量迁移阶段, 为了保证迁移性能和传输的稳定性, 采用了行级并行的迁移方式。当源端数据紧凑情况下, 通过 DRS 迁移到云上 GaussDB(for MySQL)后, 可能会出现数据膨胀现象, 使得磁盘空间使用远大于源端。

### 场景 2: 大量删除操作后在表空间留下碎片所致

原因: 当删除数据时, mysql 并不会回收被删除数据占据的存储空间, 而只做标记删除, 尝试供后续复用, 等新的数据来填补相应空间, 如果短时间内没有数据来填补这些空间, 就造成了表空间膨胀, 形成大量碎片;

可以通过如下 SQL 语句, 查询某个表详细信息, DATA\_FREE 字段表示表空间碎片大小:

```
select * from information_schema.tables where table_schema='db_name' and table_name = 'table_name'\G
```

```
mysql> select * from information_schema.tables where table_schema='mall19wo' and table_name='deliveryman_track'\G
***** 1. row *****
TABLE_CATALOG: def
TABLE_SCHEMA: mall19wo
TABLE_NAME: deliveryman_track
TABLE_TYPE: BASE TABLE
ENGINE: InnoDB
VERSION: 10
ROW_FORMAT: Dynamic
TABLE_ROWS: 11968
AVG_ROW_LENGTH: 4479273
DATA_LENGTH: 53607940096
MAX_DATA_LENGTH: 0
INDEX_LENGTH: 802816
DATA_FREE: 54668558336
AUTO_INCREMENT: 94507
CREATE_TIME: 2022-06-28 23:39:00
UPDATE_TIME: 2022-07-07 11:03:22
CHECK_TIME: NULL
TABLE_COLLATION: utf8mb4_general_ci
CHECKSUM: NULL
CREATE_OPTIONS: row_format=DYNAMIC
TABLE_COMMENT: 骑手轨迹
1 row in set (0.00 sec)
```

## 解决方案

针对表空间膨胀的问题, 可以进行表空间优化整理, 从而缩小空间, 执行如下 SQL 命令:

```
optimize table table_name;
```

### 说明

optimize table 命令会有短暂锁表操作, 所以进行表空间优化时, 建议避开业务高峰期, 避免影响正常业务的进行。

## 5.3.13 MySQL 创建用户提示服务器错误(ERROR 1396)

### 场景描述

用户帐号在控制台界面上消失, 创建不了同名帐号, 但使用帐号名和旧密码还能连接。

创建用户失败的报错信息:



```
ERROR 1396 (HY000): Operation CREATE USER failed for xxx。
```

## 问题分析

1. 查询确认后，发现消失的账号在 `mysql.user` 表中已经被删除，故控制台不再显示；
2. 使用帐号名和旧密码还能连接登录，说明使用的是 `delete from mysql.user` 方式删除用户。使用这种方式删除用户，需要执行 `flush privileges` 后，才会清理内存中相关数据，该用户才彻底不能登录。
3. 使用 `delete from mysql.user` 方式删除用户，无法重新创建相应账户（报错 `ERROR 1396`），原因是内存中相关数据仍然存在。

```
mysql> CREATE USER 'test1'@'localhost' IDENTIFIED BY 'test1';
Query OK, 0 rows affected (0.03 sec)

mysql> DELETE FROM mysql.user WHERE Host='localhost'AND User='test1';
Query OK, 1 row affected (0.02 sec)

mysql> CREATE USER 'test1'@'localhost' IDENTIFIED BY 'test1';
ERROR 1396 (HY000): Operation CREATE USER failed for 'test1'@'localhost'
```

正确删除用户的方式为 `drop user` 语句，注意以下几点：

- `drop user` 语句可用于删除一个或多个用户，并撤销其权限。
- 使用 `drop user` 语句必须拥有 MySQL 数据库的 `DELETE` 权限或全局 `CREATE USER` 权限。
- 在 `drop user` 语句的使用中，若没有明确地给出帐户的主机名，则该主机名默认为“%”。

故障场景恢复示例：

创建用户后用 `delete` 删除用户，再创建同名用户时报错 `ERROR 1396`。通过执行 `flush privileges` 后，可正常创建同名用户。

```
mysql> CREATE USER 'test1'@'localhost' IDENTIFIED BY 'test1';
ERROR 1396 (HY000): Operation CREATE USER failed for 'test1'@'localhost'
mysql> FLUSH HOSTS;
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE USER 'test1'@'localhost' IDENTIFIED BY 'test1';
ERROR 1396 (HY000): Operation CREATE USER failed for 'test1'@'localhost'
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE USER 'test1'@'localhost' IDENTIFIED BY 'test1';
Query OK, 0 rows affected (0.01 sec)
```

## 解决方案

- 方式一（推荐）：在业务低峰期，使用管理员帐户执行 `drop user user_name` 删除用户，再重新创建该用户，修复该问题。

- 方式二：在业务低峰期，使用管理员帐户执行 **flush privileges** 后，再重新创建该用户，修复该问题。建议开启数据库全量 sql 洞察功能，便于分析是哪个客户端删除了用户。

### 5.3.14 执行 alter table xxx discard/import tablespace 报错

#### 场景描述

在 GaussDB(for MySQL)中执行 alter table xxx discard/import tablespace 会报错：ERROR 3658 (HY000): Feature IMPORT/DISCARD TABLESPACE is unsupported ()。

#### 原因分析

alter table xxx discard/import tablespace 是社区 MySQL 一种基于本地 .ibd 的表空间文件物理的做数据表内容替换（多用于数据迁移、备份恢复等）的方法。

GaussDB(for MySQL)是存储计算分离架构，实际数据存储在共享存储上，本地没有 .ibd 文件，所以不支持相应的物理操作。

#### 解决方案

使用其他如导入导出、DRS 同步、备份恢复等方式做数据表内容的替换。

### 5.3.15 数据库报错 Native error 1461 的解决方案

#### 场景描述

MySQL 用户通常在并发读写、大批量插入 sql 语句或数据迁移等场景出现如下报错信息：

```
mysql_stmt_prepare failed! error(1461)Can't create more than max_prepared_stmt_count statements (current value: 16382)
```

#### 故障分析

“max\_prepared\_stmt\_count” 的取值范围为 0~1048576，默认为“16382”，该参数限制了同一时间在 mysqld 上所有 session 中 prepared 语句的上限，用户业务超过了该参数当前值的范围。

#### 解决方案

请您调大 “max\_prepared\_stmt\_count” 参数的取值，建议调整为 “65535”。

### 5.3.16 创建表失败报错 Row size too large 的解决方案

#### 场景描述

MySQL 用户创建表失败，出现如下报错信息：

**Row size too large. The maximum row size for the used table type, not counting BLOBs, is 65535. This includes storage overhead, check the manual. You have to change some columns to TEXT or BLOBs**

## 故障分析

“varchar” 的字段总和超过了 65535，导致创建表失败。

## 解决方案

1. 缩减长度，如下所示。

```
CREATE TABLE t1 (a VARCHAR(10000),b VARCHAR(10000),c VARCHAR(10000),d
VARCHAR(10000),e VARCHAR(10000),f VARCHAR(10000) ) ENGINE=MyISAM CHARACTER SET
latin1;
```

2. 请参考[官方文档](#)修改一个字段为 TEXT 类型。

## 5.4 参数类

### 5.4.1 客户端修改全局参数失败

#### 场景描述

客户端修改全局参数，报错 “ERROR 1227 (42000): Access denied”。

#### 原因分析

GaussDB(for MySQL)不支持在数据库中执行修改全局参数的命令，需要通过管理控制台修改。

#### 解决方案

登录管理控制台修改参数。

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面，选择指定的实例，单击实例名称，进入实例的基本信息页面。

步骤 5 在左侧导航栏中选择“参数修改”，在“参数”页签查看并修改对应参数。

----结束

## 5.4.2 客户端超时参数设置导致连接超时退出

### 场景描述

使用数据库时，经常遇到连接退出，导致后续语句执行失败的情况。

### 原因分析

在使用连接器或 API 连接数据库时，客户端会有一些默认的参数配置。其中有一些比较重要的参数如 `socketTimeout`、`connectTimeout` 等，会影响客户端连接的超时时间。如果超过这个时间，一直没使用的连接就会断开。

### 解决方案

- 将 `socketTimeout`、`connectTimeout` 等参数的默认值调整为合适的值。
- 在程序中注意处理断线重连的功能。
- 推荐直接使用连接池。

## 5.4.3 修改全局变量成功但未生效

### 场景描述

使用 Console 上的参数修改功能修改 `long_query_time` 成功，但未生效。

### 原因分析

使用 Console 修改参数时，系统实际使用“`set global 变量名=新的变量值;`”修改全局参数。

在使用 `set global` 命令修改全局变量值时需要注意，该参数在当前连接和已经连接上数据库的其他连接中是不生效的，只对新连接生效，所以此时将所有连接断开重连，即可看到变量修改生效。

### 示例

举例中使用的是命令的方式做描述。

#### 1. 创建会话 1。

```
# 查看参数值。
show variables like 'long_query_time';
+-----+-----+
| Variable_name | Value      |
+-----+-----+
| long_query_time | 10.000000 |
+-----+-----+
1 row in set (0.08 sec)
# 修改变量值
set global long_query_time=1;
Query OK, 0 rows affected (0.02 sec)
# 重新查看，发现未生效。
```

```

show variables like 'long_query_time';
+-----+-----+
| Variable_name | Value   |
+-----+-----+
| long_query_time | 10.000000 |
+-----+-----+
1 row in set (0.01 sec)
    
```

2. 创建会话 2。

```

show variables like 'long_query_time';
+-----+-----+
| Variable_name | Value   |
+-----+-----+
| long_query_time | 10.000000 |
+-----+-----+
1 row in set (0.01 sec)
    
```

3. 在会话 1 中执行。

```

#会话 1 中执行 set global 后，再次查看，变量未生效。
show variables like 'long_query_time';
+-----+-----+
| Variable_name | Value   |
+-----+-----+
| long_query_time | 10.000000 |
+-----+-----+
1 row in set (0.01 sec)
# 会话 1 断开，重新连接，发现修改生效。
show variables like 'long_query_time';
+-----+-----+
| Variable_name | Value   |
+-----+-----+
| long_query_time | 1.000000 |
+-----+-----+
1 row in set (0.00 sec)
    
```

4. 会话 2 断开，重新连接，发现修改生效。

```

show variables like 'long_query_time';
+-----+-----+
| Variable_name | Value   |
+-----+-----+
| long_query_time | 1.000000 |
+-----+-----+
1 row in set (0.01 sec)
    
```

## 5.4.4 GaussDB(for MySQL) timeout 相关参数简介

MySQL 中有多种 timeout 参数，GaussDB(for MySQL)也将相关参数提供给用户设置，如下表：

表5-2 参数说明

参数名称	修改是否需要重启	参数含义
connect_timeout	否	GaussDB(for MySQL)服务器在回 Bad

参数名称	修改是否需要重启	参数含义
		handshake 响应之前等待连接数据包的时间（秒）。
innodb_flush_log_at_timeout	否	每 N 秒写入并刷新日志。当 innodb_flush_log_at_trx_commit 值为 2 时，此设置有效。
innodb_lock_wait_timeout	否	该变量控制 innodb 事务获取行锁等待的最长时间，如果超过该时间还未获取到锁资源，则会返回执行失败
parallel_queue_timeout	否	请求并行执行的查询的等待时间。如果超过该等待时间后，系统中并行执行的线程数仍然大于 parallel_max_threads，则不再等待而进入单线程执行。
lock_wait_timeout	否	试图获得元数据锁的超时时间（秒）。
net_read_timeout	否	中止读数据之前从一个连接等待更多数据的秒数。
net_write_timeout	否	中止写之前等待一个块被写入连接的秒数。
interactive_timeout	否	服务器在关闭交互式连接之前等待活动的秒数。
wait_timeout	否	服务器关闭连接之前等待非交互式连接活动的秒数。

## 5.5 性能资源类

### 5.5.1 冷热数据问题导致 SQL 执行速度慢

#### 场景描述

从自建 MySQL 或友商 MySQL 迁移到云上 GaussDB(for MySQL)实例，发现同一条 SQL 语句执行性能远差于原数据库。

#### 原因分析

同一条 SQL 语句在数据库中执行第一次和第二次可能会性能差异巨大，这是由数据库的 buffer\_pool 机制决定的：

- 第一次执行时，数据在磁盘上，称之为冷数据，读取需要一定的耗时。

- 读取完，数据会被存放于内存的 `buffer_pool` 中，称为热数据，读取迅速；对于热数据的访问速度极大的超过冷数据，所以当数据是热数据时，SQL 语句的执行速度会远快于冷数据。

该场景中，源端数据库中常用的数据一般是热数据，所以访问时速度极快。当数据迁移到云上 GaussDB(for MySQL)时，第一次执行同样的 SQL 语句，很可能是冷数据，就会访问较慢，但再次访问速度就会得到提升。

## 解决方案

该场景是正常现象，在同一个数据库中，我们经常会遇到第一次执行一条语句时很慢，但再次执行就很快，也是因为受到了 `buffer_pool` 的冷热数据原理的影响。

## 5.5.2 复杂查询造成磁盘满

### 场景描述

主机或只读节点偶尔出现磁盘占用高或磁盘占用满，其他只读节点磁盘空间占用正常。

### 原因分析

MySQL 内部在执行复杂 SQL 时，会借助临时表进行分组 (`group by`)、排序 (`order by`)、去重 (`distinct`)、Union 等操作，当内存空间不够时，便会使用磁盘空间。

排查思路：

1. 因为其他只读节点磁盘占用空间正常，且是偶尔出现，说明该实例磁盘占用高，与承载的业务相关。
2. 获取该实例的慢日志，分析磁盘占用高期间，是否有对应的慢 SQL。
3. 如果有慢 SQL，执行 `explain [慢 SQL 语句]`，分析相应慢 SQL 语句。
4. 观察 `explain` 语句输出的 `extra` 列，是否有 `using temporary`、`using filesort`，如果有，说明该语句用到了临时表或临时文件，数据量大的情况下，会导致磁盘占用高。

### 解决方案

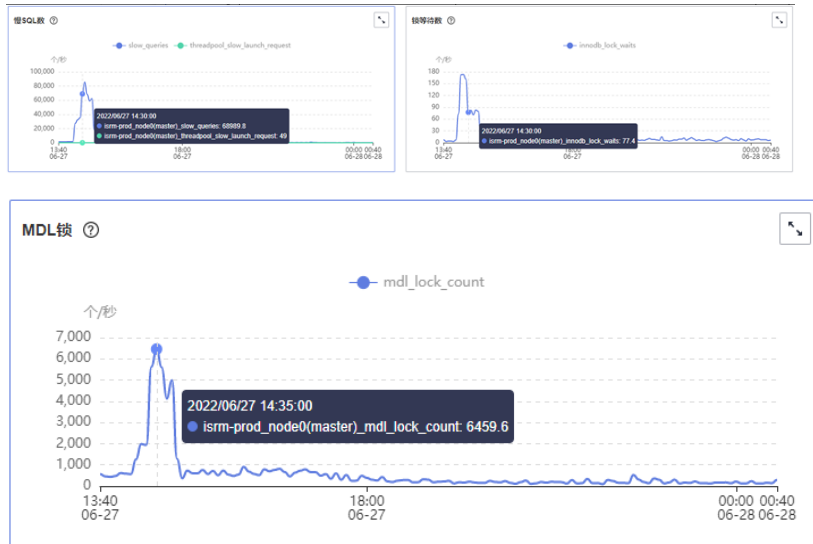
1. 复杂查询语句导致磁盘打满，建议客户从业务侧优化响应查询语句，常见优化措施：
  - 加上合适的索引。
  - 在 `where` 条件中过滤更多的数据。
  - 重写 SQL，优化执行计划。
  - 如果不得不使用临时表，那么一定要减少并发度。
2. 临时规避措施：考虑业务侧优化复杂查询语句需要一定时间，可以通过临时扩容磁盘空间规避。

## 5.5.3 业务死锁导致响应变慢

### 场景描述

14 点~15 点之间数据库出现大量行锁冲突，内核中大量 update/insert 会话在等待行锁释放，导致 CPU 使用率达到 70% 左右，数据库操作变慢。

查看 CES 指标行锁等待个数、MDL 锁数量，下图仅供参考：



发生死锁的表：

```
***** 1. row *****
Table: table_test Create Table: CREATE TABLE table_test (
...
CONSTRAINT act_fk_exe_parent FOREIGN KEY (parent_id_) REFERENCES act_ru_execution
(id_) ON DELETE CASCADE,
CONSTRAINT act_fk_exe_procdef FOREIGN KEY (proc_def_id_) REFERENCES act_re_procdef
(id_),
CONSTRAINT act_fk_exe_procinstant FOREIGN KEY (proc_inst_id_) REFERENCES
act_ru_execution (id_) ON DELETE CASCADE ON UPDATE CASCADE, CONSTRAINT
act_fk_exe_super FOREIGN KEY (super_exec_) REFERENCES act_ru_execution (id_) ON
DELETE CASCADE ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_bin
```

### 原因分析

1. 部分表发生死锁，导致 CPU 一定幅度抬升。
2. 死锁的表中有大量的外键，这些表的记录在更新时，不仅需要获取本表的行锁，还需要检查外键关联表的记录，获取相应锁。高并发情况下，比普通表更容易锁冲突或死锁，详解[官方文档](#)。
3. 当 MySQL 检查到死锁的表时，会进行事务的回滚。其影响范围不仅是某个表，还会影响外键所在的表，最终导致数据库相关操作变慢。



## 解决方案

建议排查并优化死锁表相关的业务，业务上合理使用外键，避免更新冲突，避免产生死锁。

### 5.5.4 GaussDB(for MySQL)实例 CPU 升高定位思路

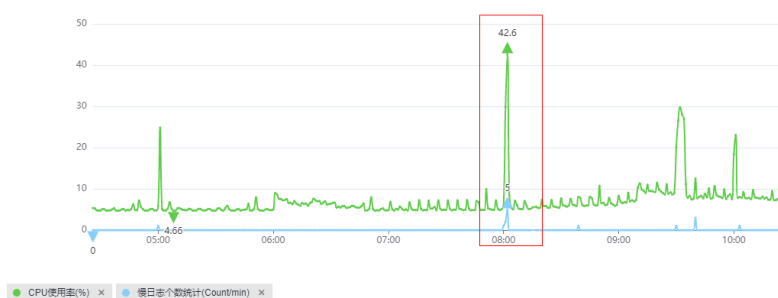
GaussDB(for MySQL)实例 CPU 升高或 100%，引起业务响应慢，新建连接超时等。

#### 场景 1 慢查询导致 CPU 升高

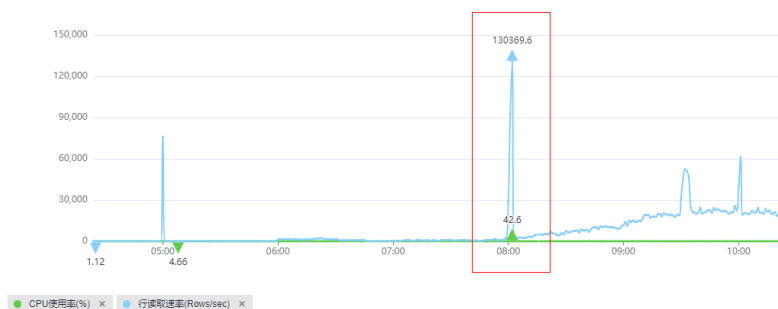
问题原因：大量慢 SQL 导致实例 CPU 升高，需要优化相应的慢 SQL。

排查思路：

查看 CPU 使用率和慢日志个数统计监控指标。



- 如果慢日志个数很多，且与 CPU 曲线吻合，可以确定是慢 SQL 导致 CPU 升高。
- 如果慢日志个数不多，但与 CPU 使用率基本一致，进一步查看行读取速率指标是否与 CPU 曲线吻合。



如果吻合，说明是少量慢 SQL 访问大量行数据导致 CPU 升高：由于这些慢 SQL 查询执行效率低，为获得预期的结果需要访问大量的数据导致平均 IO 高，因此在 QPS 并不高的情况下（例如网站访问量不大），也会导致实例的 CPU 使用率偏高。

解决方案：

1. 根据 CPU 使用率过高的时间点，查看对应时间段的慢日志信息。
2. 重点关注扫描行数、返回结果行数超过百万级别的慢查询，以及锁等待时间长的慢查询。
3. 慢查询用户可自行分析，或使用数据管理服务(DAS)对慢查询语句进行诊断。

4. 通过分析数据库执行中的会话来定位执行效率低的 SQL。
  - a. 连接数据库。
  - b. 执行 **show full processlist;**
  - c. 分析执行时间长、运行状态为 **Sending data**、**Copying to tmp table**、**Copying to tmp table on disk**、**Sorting result**、**Using filesort** 的会话，均可能存在性能问题，通过会话来分析其正在执行的 SQL。

## 场景 2 连接和 QPS 升高导致 CPU 上升

问题原因：业务请求增高导致实例 CPU 升高，需要从业务侧分析请求变化的原因。

排查思路：

查看 QPS、当前活跃连接数、数据库总连接数、CPU 使用率监控指标是否吻合。

QPS 的含义是每秒查询数，QPS 和当前活跃连接数同时上升，且 QPS 和 CPU 使用率曲线变化吻合，可以确定是业务请求增高导致 CPU 上升，如下图：



该场景下，SQL 语句一般比较简单，执行效率也高，数据库侧优化余地小，需要从业务源头优化。

解决方案：

1. 单纯的 QPS 高导致 CPU 使用率过高，往往出现在实例规格较小的情况下，建议升级实例 CPU 规格。
2. 优化慢查询，优化方法参照[场景 1 慢查询导致 CPU 升高](#)的解决方案。若优化慢查询后效果不明显，建议升级实例 CPU 规格。
3. 对于数据量大的表，建议通过分库分表减小单次查询访问的数据量。



## 解决方案

步骤 1 申请 kill 权限，间歇性批量执行 kill select count(0)慢操作，定位 select count(0)触发来源，停止来源，并拆分优化 sql。

批量 kill 动作：

```
mysql> select concat('kill ',id,':') from information_schema.processlist where info like 'select count(0)';
Empty set (0.00 sec)

mysql> select concat('kill ',id,':') from information_schema.processlist where info like 'select count(0)%';
+-----+
| concat('kill ',id,':') |
+-----+
| kill 7161;              |
| kill 7163;              |
| kill 7154;              |
| kill 7149;              |
| kill 7162;              |
| kill 7278;              |
| kill 8540;              |
| kill 8369;              |
| kill 8368;              |
| kill 8375;              |
| kill 8410;              |
| kill 8551;              |
| kill 8374;              |
| kill 8409;              |
| kill 8432;              |
| kill 8394;              |
| kill 8646;              |
| kill 8656;              |
| kill 8615;              |
| kill 8644;              |
| kill 8618;              |
| kill 8619;              |
| kill 8612;              |
| kill 8397;              |
| kill 8890;              |
| kill 8691;              |
| kill 8613;              |
| kill 8614;              |
| kill 8419;              |
| kill 8435;              |
| kill 8452;              |
| kill 7644;              |
| kill 8093;              |
| kill 7973;              |
| kill 7647;              |
| kill 7639;              |
| kill 7646;              |
| kill 7961;              |
| kill 7645;              |
| kill 7964;              |
+-----+
```

步骤 2 CPU idle 恢复：

```
dm-0      0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
dm-1      0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           25.28    0.00    0.38    0.00    0.00   74.34

Device:            rrqm/s   wrqm/s     r/s     w/s    kB/s    kB/s avgrq-sz avgrq-sz   await  r_await  w_await  svctm  %util
xvda              0.00    0.00    0.00  117.00    0.00 14704.00   251.35    1.42  12.12    0.00   12.12  0.21  2.50
xvdb              0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  0.00
xvdc              0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  0.00
dm-0              0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  0.00
dm-1              0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  0.00

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           27.94    0.00    2.38    1.75    0.00   67.92

Device:            rrqm/s   wrqm/s     r/s     w/s    kB/s    kB/s avgrq-sz avgrq-sz   await  r_await  w_await  svctm  %util
xvda              0.00    4.00    2.00  23.00   36.00  152.00   15.04    0.33  13.32  133.00    2.91  7.24  18.10
xvdb              0.00   16.00    0.00    5.00    0.00   92.00   36.80    0.01  1.20    0.00    1.20    1.00  0.50
xvdc              0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  0.00
dm-0              0.00    0.00    0.00   21.00    0.00   92.00    8.76    0.04  1.67    0.00    1.67    0.24  0.50
dm-1              0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  0.00

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           25.50    0.00    0.12    0.00    0.00   74.38

Device:            rrqm/s   wrqm/s     r/s     w/s    kB/s    kB/s avgrq-sz avgrq-sz   await  r_await  w_await  svctm  %util
xvda              0.00    0.00    0.00    1.00    0.00   20.00   40.00    0.00    1.00    0.00    1.00    1.00  0.10
xvdb              0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  0.00
xvdc              0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  0.00
dm-0              0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  0.00
dm-1              0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  0.00

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           25.41    0.00    0.00    0.00    0.00   74.59

Device:            rrqm/s   wrqm/s     r/s     w/s    kB/s    kB/s avgrq-sz avgrq-sz   await  r_await  w_await  svctm  %util
xvda              0.00    0.00    0.00    1.00    0.00    4.00    8.00    0.00    1.00    0.00    1.00    1.00  0.10
xvdb              0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  0.00
xvdc              0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  0.00
dm-0              0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  0.00
dm-1              0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  0.00
```

----结束


## 5.6 基本使用类

### 5.6.1 查看 GaussDB(for MySQL)的存储容量

GaussDB(for MySQL)是存储计算分离架构，数据存储在共享存储系统中，共享存储容量可以通过管理控制台看到，详情请参考如下步骤操作，数据每 30 分钟更新一次。

#### 操作步骤

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在实例列表中，单击目标实例名称，进入实例的“基本信息”页面。

步骤 5 在基本信息页面的“存储“存储/备份空间”模块可以看到当前实例占用的共享存储容量。

#### 说明

GaussDB(for MySQL)存储容量的计算与传统 MySQL 有一定的区别，与传统 MySQL 使用（数据大小+索引大小+空闲空间）计算的容量数据会有一些的差别。

如果要查询精确的存储使用量，可以使用管理控制台查询或者连接 GaussDB(for MySQL)数据库后，执行 show spaceusage;命令查看当前数据使用的存储容量，该值为精确值，非估算值。

图5-2 查看存储容量



- 共享存储
  - 显示的使用状况就是该实例购买的包周期的共享存储容量及目前数据已占用的容量。
  - 如果已使用空间超过购买的共享存储容量，GaussDB(for MySQL)会自动扩容，无需担心磁盘满带来的业务问题。

- 自动扩容的空间会按照按需使用的收费标准收取，建议超出后使用磁盘容量变更功能扩展包周期的存储容量。
- 备份空间：  
系统会赠送一份与包周期或按需存储容量相同大小的备份空间。

#### ----结束

执行 `show spaceusage;` 命令查看存储容量，其值等于表数据、表预分配空间、分区预分配空间、Binlog、Redolog 和 Undolog 之和，详情见下表：

条目	查看方式	说明
表数据	<code>select sum(data_length+index_length+data_free) from information_schema.tables;</code>	传统 MySQL 的容量计算方式，该语句依赖统计数据精确度，在统计数据未更新时可能会有偏差。
表预分配空间	<code>select count(*) from information_schema.tables;</code>	每张表会预分配 4MB 空间，该语句查询出表的数量乘以 4MB 就是总的表预分配空间。
分区预分配空间	<code>select count(*) from INFORMATION_SCHEMA.PARTITIONS where PARTITION_NAME is not null;</code>	每个分区会预分配 4MB 空间，该语句查询出分区数量乘以 4MB 就是总的分区预分配空间。
Binlog	<code>show binary logs;</code>	将所有 binlog 的文件大小相加。
Redolog	<code>show lsnninfo;</code>	<code>flushed_to_disk_lsn-truncate_lsn</code>
Undolog	无法直接查看	需要时可咨询客服人员。

## 5.6.2 修改库名和修改表名

对于库重命名和表重命名，GaussDB(for MySQL)与社区 MySQL 的用法是相同的。

- 支持修改表名：`rename table a to b;` 注意，该语句是可以跨库执行的，比如：`rename table da.ta to db.ta;` 是将 ta 表从 da 库移动到 db 库。
- 不支持修改库名，如果有修改库名的需求，可以先创建新的库名，然后借助 `rename table` 的跨库执行将所有表从原库移动到新库，然后删除原库。语句示例：

```
# 进入原库
use ta;
# 列出原库的所有表名
Show tables;
# 查看原库的创建语句
Show create database ta;
# 使用原库的创建语句创建新库(只改库名，其他参数照抄，这样能尽量保证新库与原库的各类参数相同)
```

```
create database tb;
# 将原库所有表移动至新库
rename table da.ta to db.ta;
rename table da.tb to db.tb;
rename table da.tc to db.tc;
...
# 删除原库
Drop database ta;
```

## 5.6.3 字符集和字符序的默认选择方式

### 相关变量设置

参数组中默认 `character_set_server=utf8`、`collation_server=utf8_general_ci`，可以在界面修改参数值。

### 默认选择方式

- 在创建数据库时，如果未显式指定库的字符集和字符序，则库的字符集和字符序采用 `character_set_server` 和 `collation_server` 参数的值；如果显式指定，则使用指定的字符集和字符序。
- 在创建数据表时，如果未显式指定表的字符集和字符序，则表默认字符集和字符序使用所在数据库的字符集和字符序；如果显式指定，则使用指定的字符集和字符序。
- 在创建数据表时，如果未显式指定字段的字符集和字符序，则字段使用所在表的字符集和字符序；如果显式指定，则使用指定的字符集和字符序。

示例 1：不显式指定字符集、字符序的情况下创建数据库和数据表。

```
mysql> show variables like 'character_set_server';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| character_set_server | utf8 |
+-----+-----+
1 row in set (0.01 sec)

mysql> show variables like 'collation_server';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| collation_server | utf8_general_ci |
+-----+-----+
1 row in set (0.01 sec)

mysql> create database test_default;
Query OK, 1 row affected (0.26 sec)

mysql> show create database test_default;
+-----+-----+
| Database | Create Database |
+-----+-----+
| test_default | CREATE DATABASE `test_default` /*!40100 DEFAULT CHARACTER SET utf8 */ /*!80016 DEFAULT ENCRYPTION='N' */ |
+-----+-----+
1 row in set (0.00 sec)

mysql> create table test_default.t_default(name varchar(20));
Query OK, 0 rows affected (0.23 sec)

mysql> show create table test_default.t_default;
+-----+-----+
| Table | Create Table |
+-----+-----+
| t_default | CREATE TABLE `t_default` (
  `name` varchar(20) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 |
+-----+-----+
1 row in set (0.01 sec)
```

示例 2：显式指定库的字符集、字符序的情况下创建数据库。

```
mysql> create database test_define CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci;
Query OK, 1 row affected (0.00 sec)

mysql> show create database test_define;
+-----+-----+
| Database | Create Database |
+-----+-----+
| test_define | CREATE DATABASE `test_define` /*!40100 DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci */ /*!80016 DEFAULT ENCRYPTION='N' */ |
+-----+-----+
1 row in set (0.00 sec)

mysql> create table test_define.t_default(name varchar(20));
Query OK, 0 rows affected (0.08 sec)

mysql> show create table test_define.t_default;
+-----+-----+
| Table | Create Table |
+-----+-----+
| t_default | CREATE TABLE `t_default` (
  `name` varchar(20) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-----+-----+
1 row in set (0.01 sec)
```

示例 3：显式指定表的字符集、字符序的情况下创建数据表。

```
mysql> create table test_define.t_define(name varchar(20)) CHARACTER SET utf8 COLLATE utf8_bin;
Query OK, 0 rows affected, 2 warnings (0.05 sec)

mysql> show create table test_define.t_define;
+-----+-----+
| Table | Create Table |
+-----+-----+
| t_define | CREATE TABLE `t_define` (
  `name` varchar(20) COLLATE utf8_bin DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin |
+-----+-----+
1 row in set (0.01 sec)
```

示例 4：显式指定字段的字符集、字符序的情况下创建数据表。

```
mysql> create table test_define.t_v_define(name varchar(20) CHARACTER SET gbk COLLATE gbk_bin, str char(32)) CHARACTER SET utf8 COLLATE utf8_bin;
Query OK, 0 rows affected, 2 warnings (0.06 sec)

mysql> show create table test_define.t_v_define;
+-----+-----+
| Table | Create Table |
+-----+-----+
| t_v_define | CREATE TABLE `t_v_define` (
  `name` varchar(20) CHARACTER SET gbk COLLATE gbk_bin DEFAULT NULL,
  `str` char(32) COLLATE utf8_bin DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin |
+-----+-----+
1 row in set (0.01 sec)
```

## 5.6.4 自增字段值跳变的原因

数据表中的自增字段取值不是连续的，自增值跳变。

出现表中的自增字段取值不连续的情况，可能原因有以下几种：

- 初值与步长问题，步长不为 1 会导致自增字段取值不连续。

```
mysql> show variables like 'auto_inc%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| auto_increment_increment | 2 |
| auto_increment_offset | 1 |
+-----+-----+

mysql> select * from auto_test1;
+----+
| id |
+----+
| 2 |
| 4 |
| 6 |
| 8 |
+----+
```

- 直接修改表的 AUTO\_INCREMENT，会导致自增字段取值跳变。

```
mysql> select * from animals;
+-----+-----+
```



```

| id | name      |
+----+-----+
|  1 | dog       |
|  2 | cat       |
|  3 | penguin   |
+----+-----+
mysql> show create table animals;
+-----+-----+
| Table | Create Table |
+-----+-----+
| animals | CREATE TABLE `animals` (
`id` mediumint NOT NULL AUTO_INCREMENT,
`name` char(30) NOT NULL,
PRIMARY KEY (`id`))
ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8 |
+-----+-----+
mysql> alter table animals AUTO_INCREMENT=100;
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0
mysql> show create table animals;
+-----+-----+
| Table | Create Table |
+-----+-----+
| animals | CREATE TABLE `animals` (
`id` mediumint NOT NULL AUTO_INCREMENT,
`name` char(30) NOT NULL,
PRIMARY KEY (`id`))
) ENGINE=InnoDB AUTO_INCREMENT=100 DEFAULT CHARSET=utf8 |
+-----+-----+
mysql> INSERT INTO animals (id,name) VALUES(0,'rabbit');
Query OK, 1 row affected (0.00 sec)
mysql> select * from animals;
+----+-----+
| id | name      |
+----+-----+
|  1 | dog       |
|  2 | cat       |
|  3 | penguin   |
| 100 | rabbit    |
+----+-----+
9 rows in set (0.00 sec)
    
```

- 插入数据时直接指定自增字段的取值，会导致自增字段取值跳变。

```

mysql> select * from animals;
+----+-----+
| id | name      |
+----+-----+
|  1 | dog       |
|  2 | cat       |
|  3 | penguin   |
+----+-----+
mysql> INSERT INTO animals (id,name) VALUES(100,'rabbit');
Query OK, 1 row affected (0.00 sec)
mysql> select * from animals;
+----+-----+
| id | name      |
    
```

```
+-----+-----+
|  1 | dog      |
|  2 | cat      |
|  3 | penguin  |
| 100 | rabbit   |
+-----+-----+
9 rows in set (0.00 sec)
```

- 未提交的事务或回滚的事务，会导致 `AUTO_INCREMENT` 增长，但回滚后不会下降。后续如果再次插入数据就会导致数据中的自增字段发生跳变。

```
mysql> show create table auto_test1;
+-----+-----+
| Table      | Create Table                               |
+-----+-----+
| auto_test1 | CREATE TABLE `auto_test1` (
`id` int NOT NULL AUTO_INCREMENT,
PRIMARY KEY (`id`))
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8 |
+-----+-----+
1 row in set (0.00 sec)
mysql> select * from auto_test1;
+-----+
| id |
+-----+
|  1 |
|  2 |
|  3 |
+-----+
mysql> begin;
Query OK, 0 rows affected (0.02 sec)
mysql> insert into auto test1 values (0),(0),(0);
Query OK, 3 rows affected (0.00 sec)
Records: 3 Duplicates: 0 Warnings: 0
mysql> select * from auto_test1;
+-----+
| id |
+-----+
|  1 |
|  2 |
|  3 |
|  4 |
|  5 |
|  6 |
+-----+
6 rows in set (0.00 sec)
mysql> show create table auto_test1;
+-----+-----+
| Table      | Create Table                               |
+-----+-----+
| auto_test1 | CREATE TABLE `auto_test1` (
`id` int NOT NULL AUTO_INCREMENT,
PRIMARY KEY (`id`))
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8 |
+-----+-----+
1 row in set (0.00 sec)
```

```
mysql> rollback;
Query OK, 0 rows affected (0.05 sec)
mysql> select * from auto_test1;
+----+
| id |
+----+
| 1 |
| 2 |
| 3 |
+----+
3 rows in set (0.00 sec)
mysql> show create table auto_test1;
+-----+-----+
| Table      | Create Table                               |
+-----+-----+
| auto_test1 | CREATE TABLE `auto_test1` (
`id` int NOT NULL AUTO_INCREMENT,
PRIMARY KEY (`id`))
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8 |
+-----+-----+
mysql> insert into auto_test1 values (0),(0),(0);
Query OK, 3 rows affected (0.01 sec)
Records: 3 Duplicates: 0 Warnings: 0
mysql> select * from auto_test1;
+----+
| id |
+----+
| 1 |
| 2 |
| 3 |
| 7 |
| 8 |
| 9 |
+----+
6 rows in set (0.00 sec)
mysql> show create table auto_test1;
+-----+-----+
| Table      | Create Table                               |
+-----+-----+
| auto_test1 | CREATE TABLE `auto_test1` (
`id` int NOT NULL AUTO_INCREMENT,
PRIMARY KEY (`id`))
) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=utf8 |
+-----+-----+
```

- 数据插入后，AUTO\_INCREMENT 变化，然后删除对应的数据行，AUTO\_INCREMENT 不会下降，后续如果再次插入数据就会导致数据中的自增字段发生跳变。

```
mysql> show create table auto_test1;
+-----+-----+
| Table      | Create Table                               |
+-----+-----+
| auto_test1 | CREATE TABLE `auto_test1` (
`id` int NOT NULL AUTO_INCREMENT,
PRIMARY KEY (`id`))
) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=utf8 |
+-----+-----+
```

```

) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8 |
+-----+-----+
1 row in set (0.00 sec)
mysql> select * from auto_test1;
+----+
| id |
+----+
| 1 |
| 2 |
| 3 |
+----+
mysql> insert into auto_test1 values (0),(0),(0);
Query OK, 3 rows affected (0.00 sec)
Records: 3 Duplicates: 0 Warnings: 0
mysql> select * from auto_test1;
+----+
| id |
+----+
| 1 |
| 2 |
| 3 |
| 4 |
| 5 |
| 6 |
+----+
6 rows in set (0.00 sec)
mysql> show create table auto_test1;
+-----+-----+
| Table      | Create Table                               |
+-----+-----+
| auto_test1 | CREATE TABLE `auto_test1` (
`id` int NOT NULL AUTO_INCREMENT,
PRIMARY KEY (`id`))
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8 |
+-----+-----+
1 row in set (0.00 sec)
mysql> delete from auto_test1 where id>3;
mysql> select * from auto_test1;
+----+
| id |
+----+
| 1 |
| 2 |
| 3 |
+----+
3 rows in set (0.00 sec)
mysql> show create table auto_test1;
+-----+-----+
| Table      | Create Table                               |
+-----+-----+
| auto_test1 | CREATE TABLE `auto_test1` (
`id` int NOT NULL AUTO_INCREMENT,
PRIMARY KEY (`id`))
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8 |
+-----+-----+

```

```
mysql> insert into auto_test1 values (0),(0),(0);
Query OK, 3 rows affected (0.01 sec)
Records: 3 Duplicates: 0 Warnings: 0
mysql> select * from auto_test1;
+----+
| id |
+----+
| 1 |
| 2 |
| 3 |
| 7 |
| 8 |
| 9 |
+----+
6 rows in set (0.00 sec)
mysql> show create table auto_test1;
+-----+-----+
| Table      | Create Table                               |
+-----+-----+
| auto_test1 | CREATE TABLE `auto_test1` (
`id` int NOT NULL AUTO_INCREMENT,
PRIMARY KEY (`id`))
) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=utf8 |
+-----+-----+
```

- 因为一些原因（比如唯一键冲突），使得插入数据最终未成功的，有可能导致 AUTO\_INCREMENT 跳变。

```
mysql> create table auto_test7(`id` int NOT NULL AUTO_INCREMENT, cred_id int
UNIQUE, PRIMARY KEY (`id`));
Query OK, 0 rows affected (0.64 sec)
mysql> insert into auto test7 values(null, 1);
Query OK, 1 row affected (0.03 sec)
mysql> show create table auto_test7;
+-----+-----+
| Table      | Create Table                               |
+-----+-----+
| auto_test7 | CREATE TABLE `auto_test7` ( `id` int NOT NULL AUTO_INCREMENT,
`cred_id` int DEFAULT NULL, PRIMARY KEY (`id`), UNIQUE KEY `cred_id`
(`cred_id`)) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8 |
+-----+-----+
1 row in set (0.00 sec)
mysql> insert into auto_test7 values(null, 1);
ERROR 1062 (23000): Duplicate entry '1' for key 'auto_test7.cred_id'
mysql> show create table auto_test7;
+-----+-----+
| Table      | Create Table                               |
+-----+-----+
| auto_test7 | CREATE TABLE `auto_test7` ( `id` int NOT NULL AUTO_INCREMENT,
`cred_id` int DEFAULT NULL, PRIMARY KEY (`id`), UNIQUE KEY `cred_id`
(`cred_id`)) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8 |
+-----+-----+
```

- 批量插入数据时（如 insert...select、load file 等），自增键的申请是分批申请的，每批申请 2 的 n 次方个序号，用完继续申请，没用完也不会退回，所以可能会导致 AUTO\_INCREMENT 跳变。

```
mysql> create table auto_test5_tmp(id tinyint not null AUTO_INCREMENT, name
varchar(8), PRIMARY KEY (`id`));
Query OK, 0 rows affected (0.08 sec)
mysql> select * from auto_test5;
+----+-----+
| id | name |
+----+-----+
| 1 | A |
| 2 | B |
| 3 | C |
| 4 | X |
| 5 | Y |
| 6 | Z |
| 8 | A |
| 9 | B |
| 10 | C |
| 11 | X |
| 12 | Y |
| 13 | Z |
+----+-----+
12 rows in set (0.00 sec)
mysql> insert into auto_test5 tmp select 0,name from auto_test5;
Query OK, 12 rows affected (0.01 sec)
Records: 12 Duplicates: 0 Warnings: 0
mysql> select * from auto_test5_tmp;
+----+-----+
| id | name |
+----+-----+
| 1 | A |
| 2 | B |
| 3 | C |
| 4 | X |
| 5 | Y |
| 6 | Z |
| 7 | A |
| 8 | B |
| 9 | C |
| 10 | X |
| 11 | Y |
| 12 | Z |
+----+-----+
12 rows in set (0.00 sec)
mysql> show create table auto_test5_tmp;
+-----+-----+
| Table | Create Table |
+-----+-----+
| auto_test5_tmp | CREATE TABLE `auto_test5_tmp` ( `id` tinyint NOT NULL
AUTO_INCREMENT, `name` varchar(8) DEFAULT NULL, PRIMARY KEY (`id`))
ENGINE=InnoDB AUTO_INCREMENT=16 DEFAULT CHARSET=utf8 |
+-----+-----+
```

### 5.6.5 表的自增 AUTO\_INCREMENT 初值与步长


数据表中自增字段的 AUTO\_INCREMENT 的初值与步长由 auto\_increment\_increment 和 auto\_increment\_offset 参数决定。

- auto\_increment\_offset: AUTO\_INCREMENT 值的初值。
- auto\_increment\_increment: AUTO\_INCREMENT 值每次增长的步长。
- 当 auto\_increment\_offset > auto\_increment\_increment 时，实际使用时初值会变为 auto\_increment\_increment。
- 当 auto\_increment\_offset <= auto\_increment\_increment，自增值计算方式如下：  
自增值 = auto\_increment\_offset + N \* auto\_increment\_increment (N 为插入的数据条数)

在 GaussDB(for MySQL)中这两个参数默认值都为 1，参考如下步骤修改。如需修改时需要在控制台-实例详情-参数修改中修改。

步骤 1 登录管理控制台。

步骤 2 单击管理控制台左上角的 ，选择区域和项目。

步骤 3 在页面左上角单击 ，选择“数据库 > 云数据库 GaussDB(for MySQL)”。

步骤 4 在“实例管理”页面，选择指定的实例，单击实例名称，进入实例的基本信息页面。

步骤 5 在左侧导航栏中选择“参数修改”，在“参数”页签修改相应参数。

----结束

示例:

1. auto\_increment\_offset=1, auto\_increment\_increment=1, 表示初值为 1，步长为 1。

```
show variables like 'auto_inc%';
+-----+-----+
| Variable_name          | Value |
+-----+-----+
| auto_increment_increment | 1     |
| auto_increment_offset   | 1     |
+-----+-----+
```

2. 修改 auto\_increment\_increment=2, 步长变为 2。

```
set session auto_increment_offset=2;
Query OK, 0 rows affected (0.02 sec)
show variables like 'auto_inc%';
+-----+-----+
| Variable_name          | Value |
+-----+-----+
| auto_increment_increment | 2     |
| auto_increment_offset   | 1     |
+-----+-----+
```

3. auto\_increment\_offset=10, auto\_increment\_increment=2, 由于 auto\_increment\_offset > auto\_increment\_increment, 因此初值为 2, 步长为 2。

```
set session auto_increment_offset=10;
set session auto_increment_increment=2;
show variables like 'auto_inc%';
+-----+-----+
| Variable_name          | Value |
+-----+-----+
```

```
| auto_increment_increment | 2 |
| auto_increment_offset   | 10 |
+-----+-----+
create table auto_test2(id int NOT NULL AUTO_INCREMENT, PRIMARY KEY (`id`));
Query OK, 0 rows affected (0.08 sec)
show create table auto_test2;
CREATE TABLE `auto_test2` ( `id` int NOT NULL AUTO_INCREMENT, PRIMARY KEY
(`id`) ) ENGINE=InnoDB DEFAULT CHARSET=utf8
1 row in set (0.01 sec)
insert into auto_test2 values(0), (0), (0);
Query OK, 3 rows affected (0.00 sec)
Records: 3 Duplicates: 0 Warnings: 0
select * from auto_test2;
+----+
| id |
+----+
| 2 |
| 4 |
| 6 |
+----+
3 rows in set (0.01 sec)
```

4. `auto_increment_offset=5`, `auto_increment_increment=10`, 初值为 5, 步长为 10。

```
set session auto_increment_offset=5;
set session auto_increment_increment=10;
show variables like 'auto_inc%';
+-----+-----+
| Variable_name          | Value |
+-----+-----+
| auto_increment_increment | 10 |
| auto_increment_offset   | 5 |
+-----+-----+
create table auto_test3(id int NOT NULL AUTO_INCREMENT, PRIMARY KEY (`id`));
insert into auto_test3 values(0), (0), (0);
Query OK, 3 rows affected (0.00 sec)
Records: 3 Duplicates: 0 Warnings: 0
select * from auto_test3;
+----+
| id |
+----+
| 5 |
| 15 |
| 25 |
```

## 5.6.6 修改表的自增 AUTO\_INCREMENT 值

AUTO\_INCREMENT 修改时, 遵循如下约束限制:

1. 当 AUTO\_INCREMENT 大于表中数据的最大值时, 可以在取值范围内任意修改为更大的值。

```
show create table animals;
+-----+-----+
| Table | Create Table |
+-----+-----+
| animals | CREATE TABLE `animals` (
```



```

`id` mediumint NOT NULL AUTO_INCREMENT, `name` char(30) NOT NULL,
PRIMARY KEY (`id`) ) ENGINE=InnoDB AUTO_INCREMENT=101 DEFAULT CHARSET=utf8 |
+-----+-----+
1 row in set (0.00 sec)
mysql> select * from animals;
+-----+-----+
| id | name |
+-----+-----+
| -50 | -middle |
| 1 | dog |
| 2 | cat |
| 50 | middle |
| 100 | rabbit |
+-----+-----+
11 rows in set (0.00 sec)
alter table animals AUTO_INCREMENT=200;
Query OK, 0 rows affected (0.22 sec)
Records: 0 Duplicates: 0 Warnings: 0
show create table animals;
+-----+-----+
| Table | Create Table |
+-----+-----+
| animals | CREATE TABLE `animals` (
`id` mediumint NOT NULL AUTO_INCREMENT, `name` char(30) NOT NULL,
PRIMARY KEY (`id`) ) ENGINE=InnoDB AUTO_INCREMENT=200 DEFAULT CHARSET=utf8 |
+-----+-----+

```

2. 当 **AUTO\_INCREMENT** 大于表中数据的最大值时，如果修改后的指定值仍大于数据的最大值，则修改为指定值成功。否则，默认会修改为数据最大值+1。

```

mysql> select * from animals;
+-----+-----+
| id | name |
+-----+-----+
| -50 | -middle |
| 1 | dog |
| 2 | cat |
| 50 | middle |
| 100 | rabbit |
+-----+-----+
mysql> show create table animals;
+-----+-----+
| Table | Create Table |
+-----+-----+
| animals | CREATE TABLE `animals` (
`id` mediumint NOT NULL AUTO_INCREMENT, `name` char(30) NOT NULL,
PRIMARY KEY (`id`) ) ENGINE=InnoDB AUTO_INCREMENT=200 DEFAULT CHARSET=utf8 |
+-----+-----+
mysql> alter table animals AUTO_INCREMENT=150;
Query OK, 0 rows affected (0.05 sec)
Records: 0 Duplicates: 0 Warnings: 0
mysql> show create table animals;
+-----+-----+
| Table | Create Table |
+-----+-----+
| animals | CREATE TABLE `animals` (
`id` mediumint NOT NULL AUTO_INCREMENT, `name` char(30) NOT NULL,

```

```
PRIMARY KEY (`id`) ) ENGINE=InnoDB AUTO_INCREMENT=150 DEFAULT CHARSET=utf8 |
+-----+-----+
mysql> alter table animals AUTO_INCREMENT=50;
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0
mysql> show create table animals;
+-----+-----+
| Table | Create Table |
+-----+-----+
| animals | CREATE TABLE `animals` (
`id` mediumint NOT NULL AUTO_INCREMENT, `name` char(30) NOT NULL,
PRIMARY KEY (`id`) ) ENGINE=InnoDB AUTO_INCREMENT=101 DEFAULT CHARSET=utf8 |
+-----+-----+
mysql> delete from animals where id=100;
Query OK, 1 row affected (0.00 sec)
mysql> select * from animals;
+-----+-----+
| id | name |
+-----+-----+
| -50 | -middle |
| 1 | dog |
| 2 | cat |
| 50 | middle |
+-----+-----+
10 rows in set (0.00 sec)
mysql> alter table animals AUTO_INCREMENT=50;
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0
mysql> show create table animals;
+-----+-----+
| Table | Create Table |
+-----+-----+
| animals | CREATE TABLE `animals` (
`id` mediumint NOT NULL AUTO_INCREMENT, `name` char(30) NOT NULL,
PRIMARY KEY (`id`) ) ENGINE=InnoDB AUTO_INCREMENT=51 DEFAULT CHARSET=utf8 |
+-----+-----+
1 row in set (0.00 sec)
```

### 3. AUTO\_INCREMENT 无法修改为负数。

```
alter table animals AUTO_INCREMENT=-1;
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 near '-1'
at line 1
```

## 5.6.7 自增主键达到上限，无法插入数据

### 场景描述

插入数据时报错 ERROR 1062 (23000): Duplicate entry 'xxx' for key 'xxx'。

### 原因分析

自增主键的字段取值达到上限，无法继续增长，导致新插入的数据生成的自增主键值与表中上一条数据相同，因为自增主键的值不可重复，插入失败报错。

## 解决方案

1. 如果数据变化较多，表中实际数据量远小于自增主键的容量，则可以考虑将该表的数据全量导入新表，删除原表，然后 `rename` 将新表名改回原表名。(使用数据导入导出的方法有多种实现方法，此处仅举其中一种例子)

- a. 创建表 `auto_test5_tmp`。

```
create table auto_test5_tmp(id tinyint not null AUTO INCREMENT, name
varchar(8), PRIMARY KEY (`id`));
Query OK, 0 rows affected (0.07 sec)
```

- b. 插入数据。

```
insert into auto_test5_tmp select 0,name from auto_test5;
Query OK, 6 rows affected (0.01 sec)
Records: 6 Duplicates: 0 Warnings: 0
```

- c. 查询表数据。

```
select * from auto_test5_tmp;
+----+-----+
| id | name |
+----+-----+
| 1  | A    |
| 2  | B    |
| 3  | C    |
| 4  | X    |
| 5  | Y    |
| 6  | Z    |
+----+-----+
```

- d. 删除表。

```
drop table auto_test5;
```

- e. 重命名。

```
rename table auto_test5_tmp to auto_test5;
Query OK, 0 rows affected (0.12 sec)
```

2. 如果自增主键的取值范围不够，则修改自增主键的字段类型。

```
alter table auto_test6 modify column id int NOT NULL AUTO_INCREMENT;
Query OK, 6 rows affected (0.15 sec)
Records: 6 Duplicates: 0 Warnings: 0
```

## 5.6.8 自增字段取值

GaussDB(for MySQL)对自增字段的赋值有以下几种方法:

```
# 表结构
CREATE TABLE animals (
  id MEDIUMINT NOT NULL AUTO_INCREMENT,
  name CHAR(30) NOT NULL,
  PRIMARY KEY (id)
);
```

1. 不对自增字段赋值，数据库会自动将自增值填入字段中，`AUTO_INCREMENT` 自增。

- a. 插入数据。

```
INSERT INTO animals (name) VALUES
('dog'),('cat'),('penguin'),('lax'),('whale'),('ostrich');
```

b. 查询表数据。

```
select * from animals;
+----+-----+
| id | name  |
+----+-----+
| 1  | dog   |
| 2  | cat   |
| 3  | penguin |
| 4  | lax   |
| 5  | whale |
| 6  | ostrich |
+----+-----+
```

c. 查询表结构。

```
show create table animals;
+-----+-----+
| Table | Create Table |
+-----+-----+
| animals | CREATE TABLE `animals` ( `id` mediumint NOT NULL AUTO_INCREMENT, `name` char(30) NOT NULL, PRIMARY KEY (`id`)) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8 |
+-----+-----+
```

2. 对自增字段赋 0 或 null 值，数据库会自动将自增值填入字段中。  
AUTO\_INCREMENT 自增。

a. 插入数据。

```
INSERT INTO animals (id,name) VALUES (0,'groundhog');
INSERT INTO animals (id,name) VALUES (NULL,'squirrel');
```

b. 查询数据。

```
select * from animals;
+----+-----+
| id | name  |
+----+-----+
| 1  | dog   |
| 2  | cat   |
| 3  | penguin |
| 4  | lax   |
| 5  | whale |
| 6  | ostrich |
| 7  | groundhog |
| 8  | squirrel |
+----+-----+
8 rows in set (0.00 sec)
```

c. 查询表结构。

```
show create table animals;
+-----+-----+
-+
| Table | Create Table |
|
+-----+-----+
-+
```

```
| animals | CREATE TABLE `animals` ( `id` mediumint NOT NULL  
AUTO_INCREMENT, `name` char(30) NOT NULL, PRIMARY KEY (`id`))  
ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=utf8 |  
+-----+-----+
```

3. 直接使用大于 AUTO\_INCREMENT 的值 A，数据库会将 A 填入字段并修改 AUTO\_INCREMENT=A+1。

- a. 插入数据。

```
INSERT INTO animals (id,name) VALUES(100,'rabbit');
```

- b. 查询数据。

```
select * from animals;  
+-----+-----+  
| id | name      |  
+-----+-----+  
|  1 | dog       |  
|  2 | cat       |  
|  3 | penguin   |  
|  4 | lax       |  
|  5 | whale     |  
|  6 | ostrich   |  
|  7 | groundhog |  
|  8 | squirrel  |  
| 100 | rabbit    |  
+-----+-----+  
9 rows in set (0.00 sec)
```

- c. 查询表结构。

```
show create table animals;  
+-----+-----+  
-----+  
| Table | Create Table  
|  
+-----+-----+  
-----+  
| animals | CREATE TABLE `animals` ( `id` mediumint NOT NULL  
AUTO_INCREMENT, `name` char(30) NOT NULL, PRIMARY KEY (`id`))  
ENGINE=InnoDB AUTO_INCREMENT=101 DEFAULT CHARSET=utf8 |  
+-----+-----+  
-----+
```

4. 使用小于 AUTO\_INCREMENT，但不冲突的值。数据可以插入，但 AUTO\_INCREMENT 不变。

```
mysql> INSERT INTO animals (id,name) VALUES(50,'middle');  
Query OK, 1 row affected (0.00 sec)  
mysql> select * from animals;  
+-----+-----+  
| id | name      |  
+-----+-----+  
|  1 | dog       |  
|  2 | cat       |  
|  3 | penguin   |  
|  4 | lax       |  
|  5 | whale     |  
|  6 | ostrich   |  
|  7 | groundhog |
```

```

| 8 | squirrel |
| 50 | middle |
| 100 | rabbit |
+-----+-----+
10 rows in set (0.00 sec)
mysql> show create table animals;
+-----+-----+
| Table | Create Table
+-----+-----+
| animals | CREATE TABLE `animals` ( `id` mediumint NOT NULL AUTO_INCREMENT,
`name` char(30) NOT NULL, PRIMARY KEY (`id`)) ENGINE=InnoDB AUTO_INCREMENT=101
DEFAULT CHARSET=utf8 |
+-----+-----+
    
```

5. 使用负值数据可以插入，但 AUTO\_INCREMENT 不变。

a. 插入数据。

```
INSERT INTO animals (id,name) VALUES (-50,'-middle');
```

b. 查询数据。

```

select * from animals;
+-----+-----+
| id | name |
+-----+-----+
| -50 | -middle |
| 1 | dog |
| 2 | cat |
| 3 | penguin |
| 4 | lax |
| 5 | whale |
| 6 | ostrich |
| 7 | groundhog |
| 8 | squirrel |
| 50 | middle |
| 100 | rabbit |
+-----+-----+
11 rows in set (0.00 sec)
    
```

c. 查询表结构。

```

show create table animals;
+-----+-----+
| Table | Create Table
+-----+-----+
| animals | CREATE TABLE `animals` ( `id` mediumint NOT NULL
AUTO_INCREMENT,
`name` char(30) NOT NULL, PRIMARY KEY (`id`)) ENGINE=InnoDB
AUTO_INCREMENT=101 DEFAULT CHARSET=utf8 |
+-----+-----+
    
```

## 5.6.9 自增属性 AUTO\_INCREMENT 为何未在表结构中显示

### 场景描述

创建表时，添加了自增属性 AUTO\_INCREMENT，执行 **show create table**，自增属性未在表结构中显示。

创建表：

```
mysql> CREATE TABLE test (
-> id int(10) NOT NULL AUTO_INCREMENT COMMENT '自增id',
-> account_id bigint(20) unsigned NOT NULL DEFAULT '0' COMMENT '关联account_base主键',
-> account_name varchar(128) NOT NULL DEFAULT '' COMMENT '用户名',
-> identity varchar(64) NOT NULL DEFAULT '' COMMENT '身份证',
-> mobile varchar(32) NOT NULL DEFAULT '' COMMENT '手机号',
-> score float(6,2) NOT NULL DEFAULT '0.00' COMMENT '分数',
-> utime bigint(20) unsigned NOT NULL DEFAULT '0' COMMENT '创建时间',
-> PRIMARY KEY (id),
-> UNIQUE KEY uique_index_account_id (account_id)
-> ) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8mb4
Query OK, 0 rows affected (0.01 sec)
```

执行 **show create table xxx**，未显示自增属性 AUTO\_INCREMENT：

```
| aml_stats3 | CREATE TABLE test (
| "id" int(10) NOT NULL COMMENT '自增id',
| "account_id" bigint(20) unsigned NOT NULL DEFAULT '0' COMMENT '关联account_base主键',
| "account_name" varchar(128) NOT NULL DEFAULT '' COMMENT '用户名',
| "identity" varchar(64) NOT NULL DEFAULT '' COMMENT '身份证',
| "mobile" varchar(32) NOT NULL DEFAULT '' COMMENT '手机号',
| "score" float(6,2) NOT NULL DEFAULT '0.00' COMMENT '分数',
| "utime" bigint(20) unsigned NOT NULL DEFAULT '0' COMMENT '创建时间',
| PRIMARY KEY ("id"),
| UNIQUE KEY "uique_index_account_id" ("account_id")
| )
+-----+
```

### 原因分析

经过排查，是因为参数“sql\_mode”设置了 NO\_FIELD\_OPTIONS 属性。

sql\_mode 相关属性介绍：

- NO\_FIELD\_OPTIONS：不要在 SHOW CREATE TABLE 的输出中打印 MySQL 专用列选项。
- NO\_KEY\_OPTIONS：不要在 SHOW CREATE TABLE 的输出中打印 MySQL 专用索引选项。
- NO\_TABLE\_OPTIONS：不要在 SHOW CREATE TABLE 的输出中打印 MySQL 专用表选项（例如 ENGINE）。

### 解决方案

将 sql\_mode 的 NO\_FIELD\_OPTIONS 属性去掉即可。

## 5.6.10 空用户的危害

MySQL 中是允许用户名为 "" 的用户存在，本章节介绍数据库中存在这种空用户时的危害。

MySQL 中使用空用户时，它将可以匹配任何用户名。这一特性也会带来多种安全性、功能性危害。所以，在实际使用过程中应避免使用空用户。

- 安全性危害
  - 当存在空用户时，连接时可以使用任意用户名进行登录。
  - 如果空用户有密码，则使用任意用户名和空用户的密码即可登录数据库，并获得空用户所拥有的所有权限。示例：

```
#没有空用户时，使用非法用户名'abcd'，连接失败
mysql> select user,host from mysql.user;
+-----+-----+
| user          | host          |
+-----+-----+
| root          | %             |
| mysql.infoschema | localhost    |
| mysql.session  | localhost    |
| mysql.sys      | localhost    |
+-----+-----+
mysql -uabcd -h127.0.0.1 -P3306 -pTest_1234
mysql: [Warning] Using a password on the command line interface can be
insecure.
ERROR 1045 (28000): Access denied for user 'abcd'@'localhost' (using
password: YES)

# 创建空用户后，使用非法用户名'abcd'，密码用空用户的密码，连接成功
mysql> create user ''@'localhost' IDENTIFIED BY 'Test_1234';
mysql> select user,host from mysql.user;
+-----+-----+
| user          | host          |
+-----+-----+
|              | localhost    |
| mysql.infoschema | localhost    |
| mysql.session  | localhost    |
| mysql.sys      | localhost    |
+-----+-----+
mysql -uabcd -h127.0.0.1 -P3306 -pTest_1234
mysql: [Warning] Using a password on the command line interface can be
insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 37Server version: 8.0.22-debug Source
distribution
Copyright (c) 2000, 2020, 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>
```



- 如果空用户没有密码，则使用任意用户名即可免密登录数据库，并获得空用户所拥有的所有权限。示例：

```
#存在无密码的空用户时，可以使用任意用户免密登录数据库。
mysql> create user ''@'localhost';
Query OK, 0 rows affected (8.87 sec)
mysql> select user,host from mysql.user;
+-----+-----+
| user          | host          |
+-----+-----+
| root          | %             |
|               | localhost    |
| mysql.infoschema | localhost    |
| mysql.session  | localhost    |
| mysql.sys      | localhost    |
+-----+-----+
mysql -uabcd -h127.0.0.1 -P3306
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 39Server version: 8.0.22-debug Source
distribution
Copyright (c) 2000, 2020, 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>
#-----
mysql -usdhsjdkdshk -h127.0.0.1 -P3306
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 40Server version: 8.0.22-debug Source
distribution
Copyright (c) 2000, 2020, 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>
```

- 功能性危害

当存在空用户时，可能因为匹配出错，导致正常的用户名无法登录。

示例：存在空用户与 root 用户的 host 有重叠时，导致 root 用户无法使用密码登录，或者使用空用户的密码登录后无法进入 root 的权限。

```
mysql> create user ''@'localhost';
Query OK, 0 rows affected (8.87 sec)
mysql> select user,host from mysql.user;
+-----+-----+
| user          | host          |
+-----+-----+
| root          | %             |
|               | localhost    |
| mysql.infoschema | localhost    |
| mysql.session  | localhost    |
+-----+-----+
```

