

数据复制服务

常见问题

文档版本 76
发布日期 2025-02-07



版权所有 © 华为技术有限公司 2025。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

华为技术有限公司

地址： 深圳市龙岗区坂田华为总部办公楼 邮编： 518129

网址： <https://www.huawei.com>

客户服务邮箱： support@huawei.com

客户服务电话： 4008302118

安全声明

漏洞处理流程

华为公司对产品漏洞管理的规定以“漏洞处理流程”为准，该流程的详细内容请参见如下网址：

<https://www.huawei.com/cn/psirt/vul-response-process>

如企业客户须获取漏洞信息，请参见如下网址：

<https://securitybulletin.huawei.com/enterprise/cn/security-advisory>

目录

1 产品咨询	1
1.1 区域和可用区	1
1.2 什么是数据复制服务	2
1.3 数据复制服务是否支持关系型数据库的 HA 实例迁移	6
1.4 数据复制服务对源数据库有什么要求	6
1.5 数据复制服务对目标数据库有什么要求	6
1.6 数据复制服务是否采用了并发技术	7
1.7 数据复制服务是否采用了压缩技术	8
1.8 DRS 是否支持公有云和专属云间的数据库迁移	8
1.9 什么是配额	8
1.10 DRS 支持将 DB2 for LUW 迁移到 PostgreSQL 吗	9
1.11 Microsoft SQL Server 数据库能与本地数据库的数据实时同步吗	9
1.12 数据复制服务是否支持某个时间段的数据复制	10
1.13 DRS 支持断点续传吗	10
1.14 什么是单主/双主灾备	10
1.15 实时迁移、实时同步和实时灾备有什么区别	13
1.16 MySQL 迁移出现数据膨胀怎么办	14
1.17 MySQL 数据同步对源数据库和目标数据库有什么影响	14
1.18 DRS 需要源数据库停止业务吗	15
1.19 什么是 SMN 主题	15
1.20 DRS 任务的双 AZ 和单 AZ 有什么区别	16
1.21 DRS 是否支持只迁移表结构不迁移数据	16
1.22 MySQL 迁移、同步、灾备任务怎么迁移账号，是否支持修改密码	16
1.23 影响 DRS 任务速度的因素有哪些以及如何进行耗时预估	16
1.24 DRS 任务支持修改对象吗	17
1.25 DRS 是否支持同一实例不同库之间的同步	18
1.26 源或者目标数据库哪些操作，会影响 DRS 任务状态	18
1.27 数据订阅与 MySQL 到 Kafka 同步的区别	19
1.28 为什么 DRS 不支持部分他云只读实例的备库做源库	19
1.29 DRS 支持 caching_sha2_password 加密用户迁移吗	20
1.30 为什么迁移前后数据库磁盘空间使用量不一致	20
1.31 DRS 发送 Kafka 消息序列化方式是什么	21
1.32 DRS 可支持自建的 MySQL 数据库迁移吗	21

1.33 DRS 会主动清空目标表数据吗?	21
1.34 入云, 出云, 自建有什么不同?	22
1.35 DRS 支持自建 MySQL 到自建 MySQL 的链路吗?	22
1.36 DRS 最小的传输粒度是什么?	22
1.37 在数据迁移过程中, 对源库中目标库进行操作, 会对 DRS 有什么影响?	22
2 网络及安全.....	23
2.1 数据复制服务有哪些安全保障措施.....	23
2.2 如何处理迁移过程中出现的网络中断.....	23
2.3 迁移过程中会使用哪些数据库账户.....	23
2.4 如何通过设置 VPC 安全组, 实现 DRS 任务网络互通.....	24
2.5 如何处理迁移实例和数据库网络连接异常.....	26
2.6 不同 VPC 场景下, 如何实现源库和目标库的网络互通.....	29
2.7 DRS 公网网络的 EIP 带宽是多少.....	30
2.8 DRS 支持跨账号云数据库迁移吗.....	30
3 权限管理.....	31
3.1 如何设置最小化权限且独立的使用 DRS 的 Oracle 账号.....	31
3.2 DRS 要求的 MySQL 权限有哪些.....	32
3.3 如何将源数据库的用户与权限导出, 再导入到目标数据库.....	37
3.4 创建 DRS 任务后执行自动功能失败.....	38
3.5 创建 DRS 任务提示用户权限不足或没有操作权限.....	40
3.6 DRS 要求的 PostgreSQL 权限有哪些.....	41
3.7 当源库有无主键表时需要 lock table 权限, 对源库有什么影响?	44
4 实时迁移.....	45
4.1 如何判断数据迁移任务可以停止.....	45
4.2 MySQL 迁移中 Definer 强制转化后如何维持原业务用户权限体系.....	45
4.3 MySQL 存储过程迁移上云后遇到调用权限的问题, 如何解决.....	46
4.4 如何确保业务数据库的全部业务已经停止.....	48
4.5 迁移日志提示 can not get agency token.....	49
4.6 Oracle 到 MySQL 迁移时, 索引超长如何处理.....	49
4.7 Oracle 等异构数据库迁移到 MySQL 后的数据库排序规则为什么是 utf8mb4_bin.....	50
4.8 RDS for MySQL 不支持 MyISAM 引擎表, 迁移时 MyISAM 如何处理.....	51
4.9 低版本迁移至 MySQL 8.0, 应该注意哪些问题.....	51
4.10 MongoDB 数据库迁移过程中, 源数据库出现内存溢出 (OOM) 是什么原因.....	59
4.11 如何关闭集合均衡器 Balancer.....	60
4.12 如何批量导出、导入事件 (event) 和触发器 (trigger)	60
4.13 源库参数 lower_case_table_names=1 时, 为什么不允许迁移包含大写字母的库或者表.....	61
4.14 分片集群 MongoDB 迁移前清除孤儿文档.....	62
4.15 MySQL 账号迁移后权限显示多了反斜杠 “\”	63
5 备份迁移.....	64
5.1 备份迁移场景的是否最后一个备份文件选择错误该如何处理.....	64
5.2 手动配置信息.....	64

5.3 DRS 备份迁移支持跨区域备份吗？	72
6 实时同步.....	73
6.1 DRS 支持直接同步不同 schema 的表到同一个 schema 吗.....	73
6.2 DRS 实时同步支持使用 Online DDL 工具吗.....	73
6.3 DRS 是否支持双向实时同步.....	75
6.4 源库 Oracle 为 RAC 集群时，为什么建议使用 SCAN IP 连接.....	75
6.5 源库 Oracle 补全日志检查方法.....	75
6.6 同步到 GaussDB(DWS)的使用建议.....	77
6.7 Oracle 到 GaussDB 字符集兼容性说明.....	105
6.8 字符集不兼容导致数据乱码或者同步失败.....	106
6.9 DRS 增量同步怎样填启动位点.....	106
6.10 MySQL 或 MariaDB 不同版本之间的语法差异.....	109
6.11 GaussDB 远程连接配置方法.....	110
6.11.1 配置说明.....	110
6.11.2 GaussDB 集中式版为源任务配置方法.....	110
6.11.3 GaussDB 分布式为源的多任务配置方法.....	111
6.11.4 GaussDB 分布式为源的单任务配置方法.....	113
6.11.5 GaussDB 分布式实例开启备机读.....	114
6.12 GaussDB 为源链路增量同步连接及端口说明.....	115
6.13 DRS 主键或者唯一键不一致导致数据不一致需要怎样处理.....	115
6.14 已经启动的任务，在配置中没有选 DDL，需要重新支持，该怎么办？	116
6.15 DRS 同步任务支持 MySQL8 到 MySQL5.7 吗？	116
6.16 DRS 已经开始的同步任务如何增删表？	116
6.17 GaussDB 到 Kafka 导入对象文件支持表名映射和 topic 映射操作说明.....	116
7 数据订阅.....	118
7.1 修改数据库后，到 SDK 的用户端消费一般有多长时间的延迟.....	118
7.2 订阅 SDK 获取不到数据，且程序异常.....	118
8 实时灾备.....	119
8.1 DRS 灾备的 RPO、RTO 是什么.....	119
8.2 双主灾备的主 1、主 2 如何选择.....	119
8.3 双主灾备子任务的正向和反向怎么理解.....	121
8.4 实时灾备常见异常场景示例.....	121
8.5 灾备任务的主备倒换是自动触发，还是手动触发.....	123
8.6 实时灾备是否支持指定库进行灾备.....	123
8.7 DDM 灾备任务源库填写说明.....	123
8.8 为什么实时灾备任务不支持触发器（TRIGGER）和事件（EVENT）	124
9 数据对比.....	126
9.1 内容对比不支持哪些数据类型.....	126
9.2 DRS 对比任务对数据库有什么影响.....	127
9.3 DRS 对比任务耗时预估.....	128
9.4 DRS 对比任务取消后，数据库有对比任务的查询进程未结束.....	128

10 通用操作.....	129
10.1 DRS 界面信息重叠是什么原因.....	129
10.2 目标库读写设置是实例级还是库级.....	129
10.3 MySQL 源库设置了 global binlog_format = ROW 没有立即生效.....	129
10.4 binlog_row_image 参数设置为 FULL 没有立即生效.....	130
10.5 设置的密码不符合目标库的密码复杂度要求时，如何修改密码强度.....	131
10.6 如何设置 MongoDB 数据库分片集群的分片键.....	132
10.7 扩大带宽是否会对 DRS 正在进行中的任务产生影响.....	134
10.8 为什么 MariaDB 和 SysDB 下的数据不迁移或同步.....	134
10.9 多对一的场景约束及操作建议.....	135
10.10 一对多的场景约束及操作建议.....	138
10.11 数据复制服务的操作日志在哪里查看.....	139
10.12 为什么 DRS 发送提示信息的语言和页面不符.....	140
10.13 为什么 DRS 任务自动结束了.....	140
10.14 DRS 导出任务列表.....	141
10.15 已结束的任务还能重新启动吗.....	141
10.16 重置任务和重新创建任务有什么区别.....	141
10.17 DRS 支持一键反向功能吗.....	141
10.18 已经创建的 SMN 主题为什么选择不了.....	141
10.19 任务创建成功后是否支持修改 SMN 主题.....	142
10.20 DDM 为源链路源库分片数量和信息怎么填写.....	142
10.21 源端 MySQL 发生主备倒换，DRS 任务会丢数据吗.....	143
10.22 DRS 对象选择时的全部迁移、表级迁移、库级迁移有什么区别.....	143
10.23 源库或目标库修改密码后如何操作.....	144
10.24 设置 SMN 主题策略，允许 DRS 发送消息通知.....	144
10.25 DRS 任务表数量太多导致任务超时怎么办.....	145
10.26 DRS 任务创建后支持修改源或目标数据库吗.....	145
10.27 Api V5 公测状态，是可用的吗？.....	145
10.28 DRS 进行暂停，结束任务，续传任务操作，需要多少时间？.....	146
11 计费相关.....	147
11.1 失败任务计费说明.....	147
11.2 暂停任务计费说明.....	147
11.3 未启动任务计费说明.....	148
11.4 包年包月到期说明.....	148
11.5 任务结束后不删除还会计费吗.....	148
11.6 资源冻结/释放/删除/退订.....	148
12 时延相关.....	150
12.1 灾备任务时延增高常见原因.....	150
12.2 MongoDB 链路时延增高可能原因.....	151
12.3 DRS 全量阶段迁移过慢或者进度不更新可能原因.....	151
12.4 DRS 增量阶段时延增高可能原因.....	152

1 产品咨询

1.1 区域和可用区

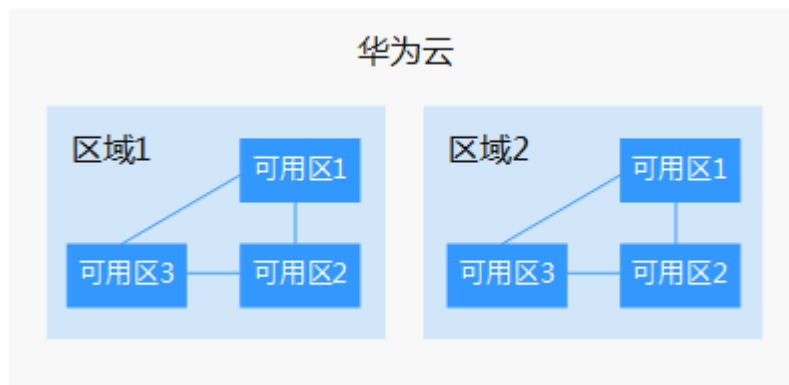
什么是区域、可用区？

我们用区域和可用区来描述数据中心的位置，您可以在特定的区域、可用区创建资源。

- 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。
- 可用区（AZ，Availability Zone）：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集群。一个Region中的多个AZ间通过高速光纤相连，以满足用户跨AZ构建高可用性系统的需求。

图1-1阐明了区域和可用区之间的关系。

图 1-1 区域和可用区



目前，华为云已在全球多个地域开放云服务，您可以根据需求选择适合自己的区域和可用区。更多信息请参见[华为云全球站点](#)。

如何选择区域？

选择区域时，您需要考虑以下几个因素：

- 地理位置

一般情况下，建议就近选择靠近您或者您的目标用户的区域，这样可以减少网络时延，提高访问速度。不过，在基础设施、BGP网络品质、资源的操作与配置等方面，中国大陆各个区域间区别不大，如果您或者您的目标用户在中国大陆，可以不用考虑不同区域造成的网络时延问题。

- 在除中国大陆以外的亚太地区有业务的用户，可以选择“亚太-曼谷”或“亚太-新加坡”区域。
- 在非洲地区有业务的用户，可以选择“南非-约翰内斯堡”区域。
- 在欧洲地区有业务的用户，可以选择“欧洲-巴黎”区域。
- 在拉丁美洲地区有业务的用户，可以选择“拉美-圣地亚哥”区域。

📖 说明

“拉美-圣地亚哥”区域位于智利。

- 资源的价格

不同区域的资源价格可能有差异，请参见[华为云服务价格详情](#)。

如何选择可用区？

是否将资源放在同一可用区内，主要取决于您对容灾能力和网络时延的要求。

- 如果您的应用需要较高的容灾能力，建议您将资源部署在同一区域的不同可用区内。
- 如果您的应用要求实例之间的网络延时较低，则建议您将资源创建在同一可用区内。

区域和终端节点

当您通过API使用资源时，您必须指定其区域终端节点。有关华为云的区域和终端节点的更多信息，请参阅[地区和终端节点](#)。

1.2 什么是数据复制服务

数据复制服务（Data Replication Service，简称DRS）是一种易用、稳定、高效、用于数据库实时迁移和数据库实时同步的云服务。

数据复制服务围绕云数据库，降低了数据库之间数据流通的复杂性，有效地帮助您减少数据传输的成本。

您可通过数据复制服务快速解决多场景下，数据库之间的数据流通问题，以满足数据传输业务需求。

数据复制服务提供了实时迁移、备份迁移、实时同步、数据订阅和实时灾备等多种功能。

实时迁移

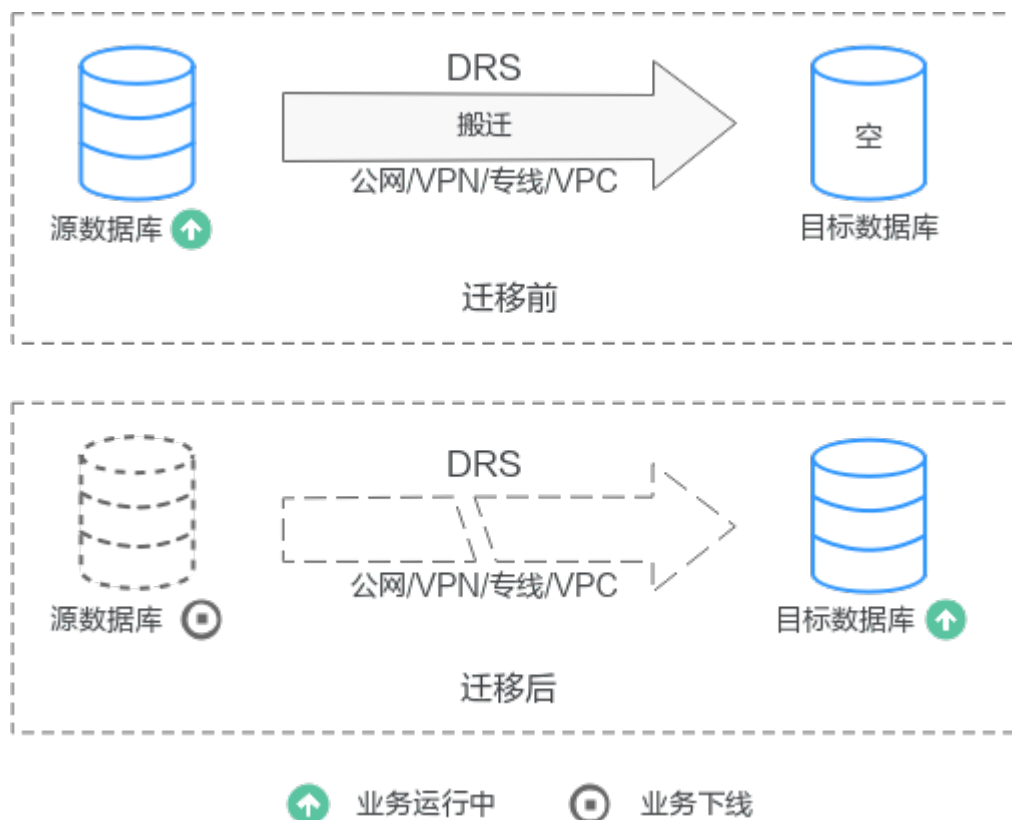
实时迁移是指在数据复制服务能够同时连通源数据库和目标数据库的情况下，只需要配置迁移的源、目标数据库实例及迁移对象即可完成整个数据迁移过程，再通过多项

指标和数据的对比分析，帮助确定合适的业务割接时机，实现最小化业务中断的数据库迁移。

实时迁移支持多种网络迁移方式，如：公网网络、VPC网络、VPN网络和专线网络。通过多种网络链路，可快速实现跨云平台数据库迁移、云下数据库迁移上云或云上跨区域的数据迁移等多种业务场景迁移。

特点：通过增量迁移技术，能够最大限度允许迁移过程中业务继续对外提供使用，有效的将业务系统中断时间和业务影响最小化，实现数据库平滑迁移上云，支持全部数据库对象的迁移。

图 1-2 实时迁移



备份迁移

由于安全原因，数据库的IP地址有时不能暴露在公网上，但是选择专线网络进行数据库迁移，成本又高。这种情况下，您可以选用数据复制服务提供的备份迁移，通过将源数据库的数据导出成备份文件，并上传至对象存储服务，然后恢复到目标数据库。备份迁移可以帮助您在云服务不触碰源数据库的情况下，实现数据迁移。

常用场景：云下数据库迁移上云。

特点：云服务无需碰触源数据库，实现数据迁移。

图 1-3 备份迁移



实时同步

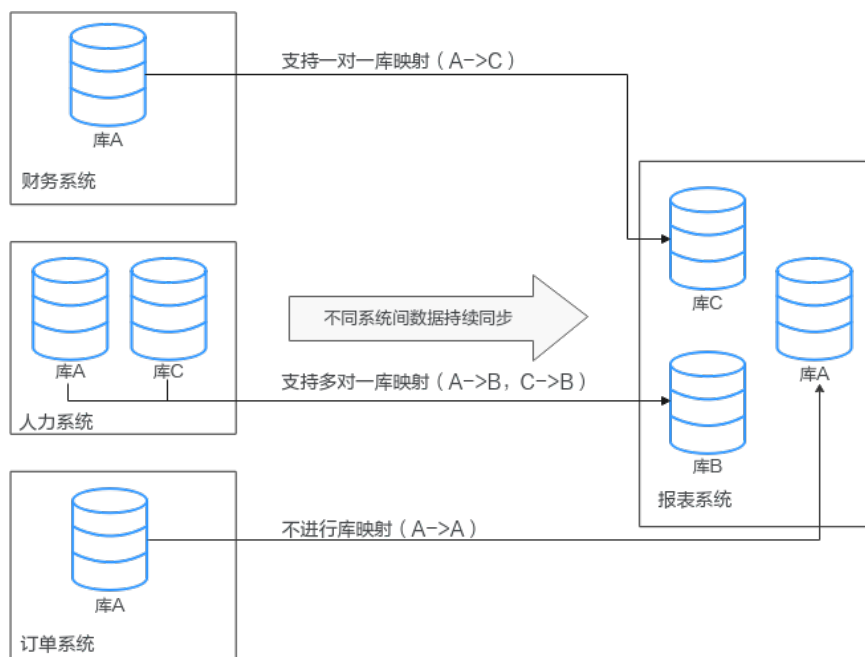
实时同步是指在不同的系统之间，将数据通过同步技术从一个数据源拷贝到其他数据库，并保持一致，实现关键业务的数据实时流动。

实时同步不同于迁移，迁移是以整体数据库搬迁为目的，而实时同步是维持不同业务之间的数据持续性流动。

常用场景：实时分析，报表系统，数仓环境。

特点：实时同步功能聚焦于表和数据，并满足多种灵活性的需求，例如多对一、一对多，动态增减同步表，不同表名之间同步数据等。

图 1-4 多对一实时同步



数据订阅

数据订阅是指获取数据库中关键业务的数据变化信息，这类信息常常是下游业务所需要的。数据订阅将其缓存并提供统一的SDK接口，方便下游业务订阅、获取、并消费，从而实现数据库和下游系统解耦，业务流程解耦。

常用场景：Kafka订阅MySQL增量数据。

图 1-5 数据订阅

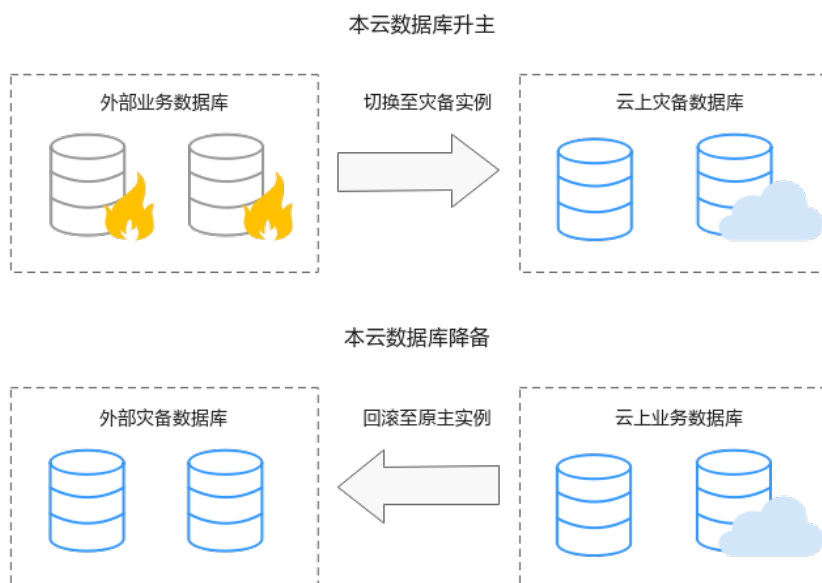


实时灾备

为了解决地区故障导致的业务不可用，数据复制服务推出灾备场景，为用户业务连续性提供数据库的同步保障。您可以轻松地实现云下数据库到云上的灾备、跨云平台的数据库灾备，无需预先投入巨额基础设施。

数据灾备支持两地三中心、两地四中心灾备架构。单边灾备可以利用灾备场景的升主、降备功能从而实现异地主备倒换的效果。

图 1-6 实时灾备倒换



录制回放

录制回放是指将源数据库发生的真实业务流量，在目标数据库模拟执行，从而观察和检验目标数据库的功能和性能表现。

录制回放主要分为录制、回放两个阶段，录制过程是从源数据库上将所需时间段内的全部SQL原语句（包括增、删、改、查）通过binlog下载、录制工具等进行采集，以文件形式缓存起来，并往目标数据库注入模拟数据，等待用户执行回放；用户可以在DRS任务上手工触发回放，观察目标数据库的性能表现。

常用场景：

- 数据库搬迁前，通过录制回放可以提前得知源业务在目标数据库的运行效果。
- 通过控制录制回放的线程及回放速度，来模拟源业务流量放大的效果，从而分析目标数据库对于未来业务激增时的稳定性表现。

图 1-7 录制回放



1.3 数据复制服务是否支持关系型数据库的 HA 实例迁移

数据复制服务的高可用性保障机制，可以支持关系型数据库的单实例和HA实例的迁移。针对HA实例的迁移，DRS的自动重连技术在连接短暂中断后连接可以得到修复，断点续传技术，根据数据库内部连续性标志可以确保实时同步的连续性和一致性。

源数据库的HA设计，满足浮动IP连接效果，且倒换时RPO=0，则DRS完全支持数据库的HA实例迁移，无需人工介入。

源数据库的HA设计，不能满足浮动IP连接且倒换时RPO=0时，存在以下几种情况：

- 使用浮动IP，但不能保证倒换时RPO=0时，可以连接，但DRS会识别出数据断层（如果有主备倒换出现数据丢失的话）并提示任务失败，此时只能根据新的数据情况，使用重置功能重新迁移。
- 使用固定IP，且倒换时RPO=0时，支持迁移（只有在实例正常运行的情况下支持迁移，否则不支持）。
- 使用固定IP，且不能保证切换时RPO=0时，可以连接，但DRS会识别出数据断层并提示任务失败，此时只能根据新的数据情况，使用重置功能重新迁移。

出云迁移且目标端数据库为HA实例时，DRS可以保证源的数据完整的迁移到目标数据库，但由于目标数据库本身的倒换不能保证RPO=0，则目标数据库可能会出现数据断层的情况。

1.4 数据复制服务对源数据库有什么要求

不同链路，数据库和权限的要求也不同，可参考以下链接，选择对应链路查看使用须知：

- [支持的数据库版本](#)
- [实时迁移方案概览](#)
- [备份迁移方案概览](#)
- [实时同步方案概览](#)
- [数据订阅使用须知](#)
- [实时灾备方案概览](#)

1.5 数据复制服务对目标数据库有什么要求

在创建任务时，用户填写的连接目标数据库的用户，需具备一定的权限。不同链路的具体权限可参考使用须知中权限要求。

- 实时迁移请参考[实时迁移方案概览](#)
- 备份迁移请参考[备份迁移方案概览](#)
- 实时同步请参考[实时同步方案概览](#)
- 数据订阅强参考[数据订阅使用须知](#)
- 实时灾备请参考[实时灾备方案概览](#)

1.6 数据复制服务是否采用了并发技术

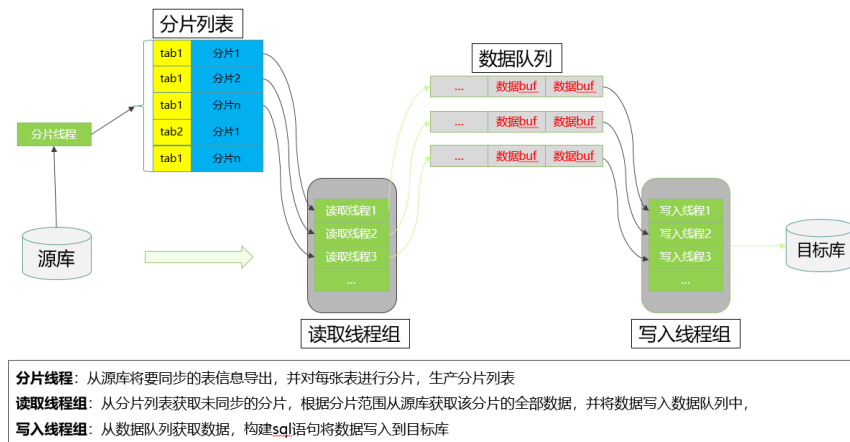
并发技术是提升性能的关键技术，DRS在多个环节采用了并发技术，总体可以分为两个大类并发技术：读写类并发和线程类并发。

读写类并发

- 为了提升全量阶段历史数据的搬迁，同时对抗网络的不稳定性，DRS实现了数据行级并发抽取效果，从而使得历史数据搬迁速度最大化，同时如果出现网络丢包时，不会大面积重来。
- 为加快数据写入，DRS实现了表级并发写入，从而大幅提升了数据传输速度，使得同步时延在同城达到毫秒级。

图 1-8 读写类并发

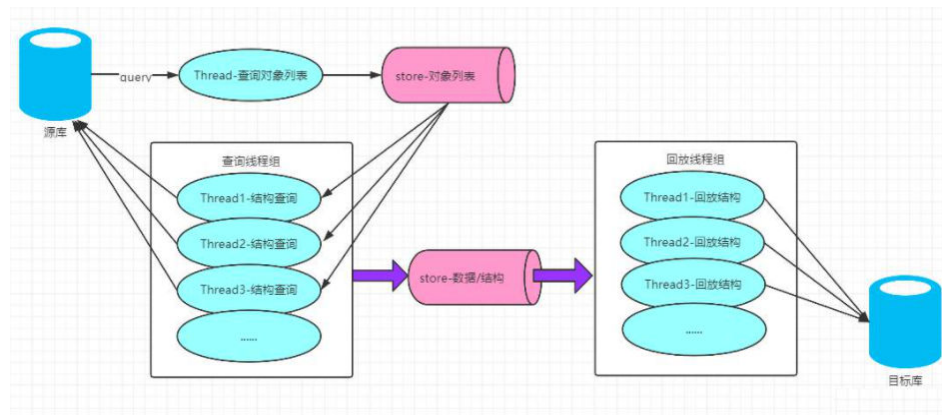
全量同步数据流



线程类并发

在面对超多数据库对象（如万级以上张表）的搬迁时，结构迁移将会成为瓶颈。DRS针对这类场景做了专项并发优化，通过多个线程并发进行结构查询和结构回放，从而大大提升结构搬迁性能。

图 1-9 线程类并发



1.7 数据复制服务是否采用了压缩技术

压缩技术是提升性能和解决数据膨胀的关键技术，DRS在数据传输和增量数据存储两大环节采用了压缩技术。

- 数据传输压缩：在网络带宽不理想的环境中，进行数据同步与传输，DRS可以开启压缩开关，以减少数据传输对带宽的占用量。
- 增量数据存储压缩：增量数据转储成逻辑文件后，将会出现膨胀现象，DRS针对该场景做了存储压缩技术，使得本地逻辑文件缓存在DRS时占用磁盘空间大大降低。

1.8 DRS 是否支持公有云和专属云间的数据库迁移

DRS支持专属云和公有云间的数据库迁移。其中，公有云向专属云的迁移数据流动方向为出云，专属云向公有云的迁移数据流动方向为入云。支持的迁移场景请参见[实时迁移](#)，具体操作方法请参见[实时迁移方案概览](#)。


1.9 什么是配额

为防止资源滥用，平台限定了各服务资源的配额，对用户的资源数量和容量做了限制，如：一个数据复制服务账号，您最多可以创建30个任务。

如果受当前资源配额限制，无法满足业务使用需求，您可以申请扩大配额。

怎样查看我的配额

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 在页面右上角，选择“资源 > 我的配额”，进入“服务配额”页面。

图 1-10 我的配额



步骤4 您可以在“服务配额”页面，查看各项资源的总配额、及使用情况。

----结束

如何申请扩大配额

步骤1 登录管理控制台。

步骤2 在页面右上角，选择“资源 > 我的配额”。系统进入“服务配额”页面，单击“申请扩大配额”。

图 1-11 申请扩大配额



步骤3 在“新建工单”页面，根据您的需求，填写相关参数。其中，“问题描述”项请填写需要调整的内容和申请原因。

图 1-12 新建工单



步骤4 填写完毕后，勾选协议并单击“提交”。

----结束

1.10 DRS 支持将 DB2 for LUW 迁移到 PostgreSQL 吗

DRS暂不支持DB2 for LUW迁移到PostgreSQL，目前仅支持DB2 for LUW到GaussDB的实时同步场景。DRS支持的源数据库和目标数据库类型请参见[支持的数据库](#)。

1.11 Microsoft SQL Server 数据库能与本地数据库的数据实时同步吗

目前Microsoft SQL Server数据库目前不支持与本地数据库数据的实时同步，如果是上云迁移，可以使用备份迁移。备份迁移支持Microsoft SQL Server的增量迁移，可以有效缩短业务割接时间。

如果是上云实时同步，目前支持目标库为GaussDB(DWS)、GaussDB、RDS for SQL Server，该链路仅支持白名单用户使用，需要提交工单申请才能使用。您可以在管理控制台右上角，选择“工单 > 新建工单”，完成工单提交。

1.12 数据复制服务是否支持某个时间段的数据复制

数据复制服务目前只支持全量、全量+增量的迁移场景。

1.13 DRS 支持断点续传吗

针对数据库的迁移、同步场景，在迁移、同步过程中由于不可抗拒因素（例如网络波动）导致的任务失败，DRS通过记录当前解析和回放的位点（该位点同时也是数据库内部一致性的依据），下次从该位点开始回放的方式来实现断点续传，以确保数据的完整性。

增量阶段的迁移、同步，DRS会自动进行多次断点续传的重试，全量阶段的MySQL迁移，系统默认进行三次自动续传，无需人工干预。

当自动重试失败累计一定次数后，任务会显示异常，需要人为根据日志来分析无法继续的原因，并尝试解决阻塞点（例如数据库修改了密码），如果环境无法修复，如所需日志已经淘汰，则使用重置功能可以完全重新开始任务。

1.14 什么是单主/双主灾备

伴随着信息技术的高速发展，数据和信息在现代企业中扮演着越来越重要的角色，信息数据的丢失和损坏将对企业造成难以估量的损失。如何抵御大规模的灾难事件受到人们越来越多的关注，现阶段而言，容灾是较好的解决方案。关键数据的备份与恢复操作已经成为系统日常运行维护的一个重要组成部分。

华为云云数据库服务的双AZ高可用实例，可以满足同城级别的灾备。数据复制服务提供的实时灾备功能具有跨区域、跨云灾备的能力，分为单主灾备和双主灾备。

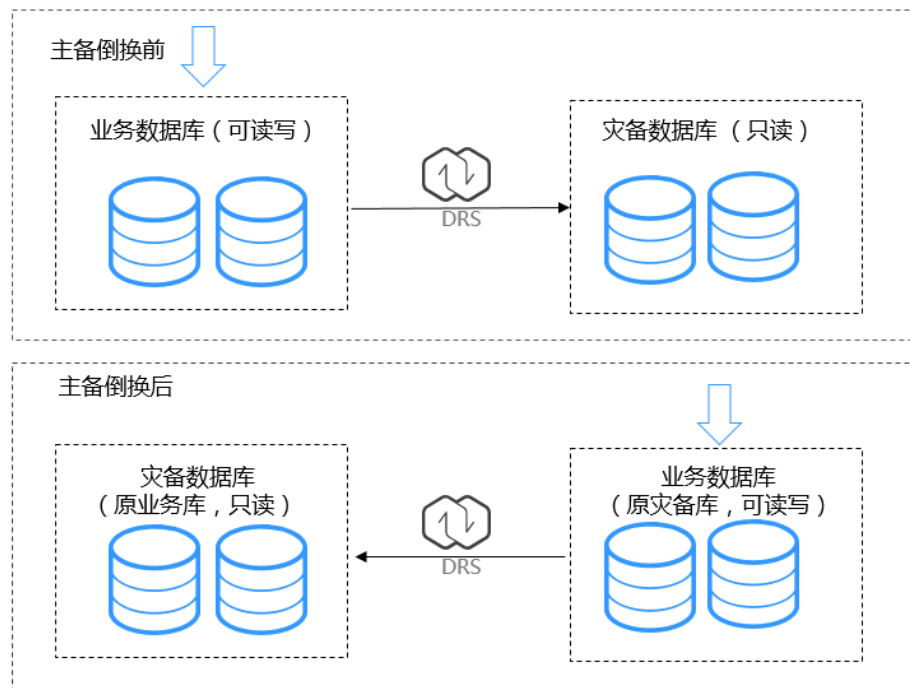
说明

目前仅支持至少单边为华为云RDS for MySQL数据库实例的单主/双主灾备，如果两边均是华为云内的RDS for MySQL数据库实例，可以进行跨区域容灾。

单主灾备

单主灾备即使用一主一备的架构，当灾难发生时原灾备数据库作为业务数据库保证业务正常运行。DRS的主备倒换功能可以实现主备切换，主备倒换前，业务正常运行在业务数据库，并且将数据实时同步至灾备数据库。此时，灾备数据库不可写入数据。主备倒换后，原灾备数据库为可读写状态，可将业务切换到原灾备数据库运行，原业务数据库不可写。

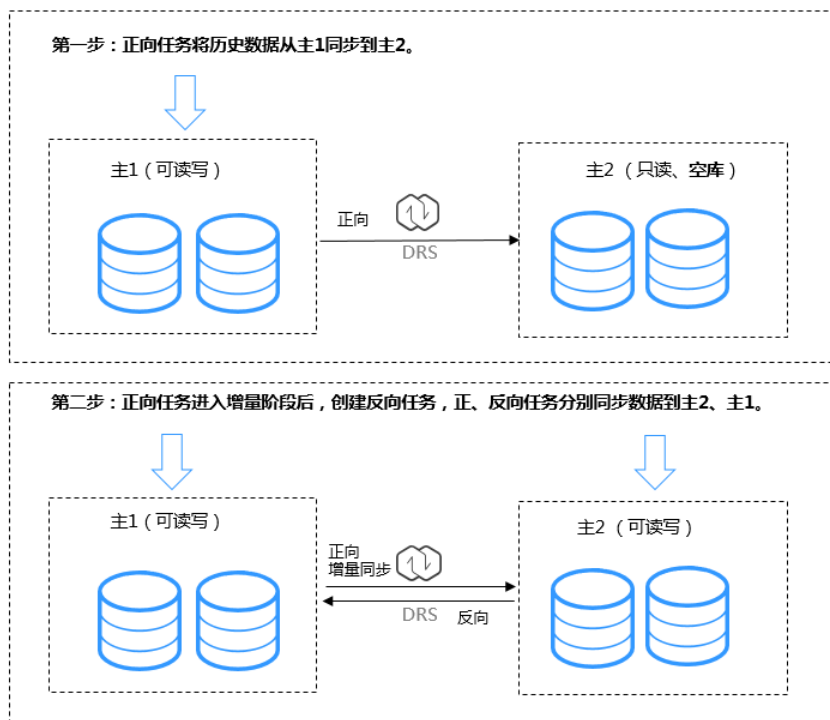
图 1-13 单主灾备



双主灾备

双主灾备多用于灾备双方互为主备，共同承担业务流量的场景。其角色分为主1、主2，使用之前需要首先确立本云（本区）RDS的角色，才可以更好的完成实时灾备的搭建。完整的双主灾备是由正、反两条链路构成的，其创建顺序有着严格的要求。开始时，主1为可读写状态、主2为只读状态，正向灾备将初始化数据全量同步到主2后，方可启动反向灾备。此时，主1、主2均为可读写状态，正/反向灾备分别向主2、主1持续同步增量数据。

图 1-14 双主灾备



特点及使用约束

- **重要！双主灾备是对环境要求比较高的架构，实施部署双主灾备前请务必阅读[约束和设计思考](#)。**
- 双主灾备的搭建过程对步骤是有严格要求的，请按照以下步骤进行部署，以确保双主任务顺利部署。
 - a. 创建正向灾备任务，具体操作步骤请参见[创建灾备任务](#)。创建完成后，会生成两个子任务，即正向灾备和反向灾备任务，此时反向灾备任务为配置状态。
 - b. 当正向灾备任务进入“灾备中”状态时（反向任务操作列出现编辑），配置并启动反向任务。

在“实时灾备管理”页面，选择该灾备任务的反向任务，单击操作列的编辑，进入“创建灾备任务”页面，继续完成创建反向任务。

建议您在主2进行验证，满足预期后，启动反向任务。

图 1-15 双向灾备任务

ID	名称	状态	类型	数据库	创建时间	网络	是否	备注
DRS-7711-Align-01	双主灾备	灾备中	MySQL	MySQL	2020/03/03 10:59:03 GM...	公网网络	否	-
DRS-7711-ch16-01		灾备中	正向	MySQL	2020/03/03 10:59:02 ...	公网网络	否	源库IP地址或域名：10...
DRS-7711-ch16-02		灾备中	反向	MySQL	2020/03/03 10:59:03 ...	公网网络	否	源库实例名称：rds-hj...

1.15 实时迁移、实时同步和实时灾备有什么区别

对比项	实时迁移	实时同步	实时灾备
适用场景	跨云平台数据库迁移、云下数据库迁移上云或云上跨区域数据库迁移等多种业务场景迁移。	实时分析，报表系统，数仓环境等。	实现云下数据库到云上的灾备、跨云平台的数据库灾备。
特点	<ol style="list-style-type: none"> 1. 同构数据库间以整体数据库搬迁为目的。 2. 支持包括：表、数据、索引、视图、存储过程、函数、数据库账号、数据库参数等。 3. 可选择表级、库级或全部迁移维度。 	<ol style="list-style-type: none"> 1. 支持同构、异构数据库间的数据同步，维持不同业务之间的数据持续性流动。 2. 聚焦于表和数据，并满足多种灵活性的需求。 3. 可选择表级、库级同步维度。 	<ol style="list-style-type: none"> 1. 支持同构数据库间的数据灾备。 2. 实现异地数据库主备倒换的效果。 3. 实例级灾备，不支持表级、库级对象选择。
支持的数据库	详情请参见 迁移方案概览 。	详情请参见 同步方案概览 。	详情请参见 灾备方案概览 。
功能特性	支持迁移用户。	不支持迁移用户。	默认迁移所有用户，不支持选择。
	支持参数对比。	不支持参数对比。	支持参数对比。
	不支持增量阶段编辑迁移对象。	支持增量阶段编辑同步对象，动态增减同步对象。	不支持增量阶段编辑灾备对象。
	不支持对象名映射。	支持对象名映射，使同步的库表在源数据库和目标数据库中存储名称不同。	不支持对象名映射。
	不支持数据加工能力。	支持数据加工能力，可对选择的对象添加规则，如数据过滤、列加工等。	不支持数据加工能力。
	不支持主备倒换。	不支持主备倒换。	支持主备倒换。
计费方式	仅支持按需计费。	支持按需计费和包年/包月。	支持按需计费和包年/包月。
更多特性	不同链路支持的功能特性也不相同，详情请参见 对应链路使用须知 。	不同链路支持的功能特性也不相同，详情请参见 对应链路使用须知 。	不同链路支持的功能特性也不相同，详情请参见 对应链路使用须知 。

1.16 MySQL 迁移出现数据膨胀怎么办

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RDS 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。

针对这种情况，客户可选择在目标库中执行以下命令，进行优化整理从而缩小空间。

```
optimize table table_name
```

📖 说明

由于命令optimize会进行锁表操作，所以进行优化时要避开表数据操作时间，避免影响正常业务的进行。

1.17 MySQL 数据同步对源数据库和目标数据库有什么影响

📖 说明

以下数据以MySQL2MySQL大规格实时同步任务为例，不同规格对数据库的连接数存在差异，仅供参考。

DRS 对源数据库的压力及影响

- 全量（初始化）阶段，DRS需要从源库将所有存量数据查询一次。DRS查询使用简单SQL语句，对源库影响主要体现在IO上，查询速度也受限于源数据库IO相关的性能以及网络带宽。一般在网络无瓶颈的情况下，会对源数据库增加约50MB/s的查询压力，以及占用2~4个CPU，在并发读取源数据库时，会占用大约6-10个session连接数，其中：
 - 有小于8个连接查询源数据库的一些系统表（如information_schema库下的表信息tables，视图信息views、列信息columns等）；
 - 有小于4个连接查询源数据库表数据分片的SQL，类似如下语句，其中select查询where后的条件只会有主键或者唯一键。

```
select id from *** where id>12345544 and limit 10000,1;
```
 - 有小于4个连接查询数据的SQL，类似如下语句，其中select后为表所有的列名，where后的条件无数据过滤情况下只会有主键或者唯一键。

```
select id,name,msg from *** where id>12345544 and id<=12445544;
```
 - 无主键表的锁表操作SQL，类似如下语句，锁表只是为了获取无主键表的一致性位点，锁表后获取一个连接就会解锁。

```
flush table *** with read lock  
lock table *** read
```
- 增量阶段对源数据库CPU基本无压力，只有一个dump连接实时获取增量日志，产生Binlog日志生成速率等同的IO及带宽影响。

DRS 对目标数据库的压力及影响

- 全量（初始化）阶段，DRS需要将源数据库结构、索引以及存量数据全部写入到目标数据库，顺序为先迁移结构，再迁移数据，最后迁移索引，不同过程不会并行，一般活跃总连接数小于8个，其中：

- 有小于8个连接在批量创建结构。
- 有小于8个连接在批量写数据，类似如下语句：

```
insert into *** (id,name,msg) values (***);
```
- 有小于8个连接在批量创建索引，类似如下语句：

```
alter table *** add index ***;
```
- 增量阶段，DRS会把源数据库日志中的增量数据解析成SQL在目标数据库中执行，一般总连接数小于64个，其中：
 - DDL会单连接串行执行，执行DDL时，不会有其他DML执行。
 - DML最多会有64个连接（短连接，默认超时时间30秒），其中DML只是简单的insert、update、delete、replace语句。

📖 说明

如果需要评估对源数据库的影响，可选择创建测试任务，再通过限速功能或业务低峰期动等来调整迁移策略。

1.18 DRS 需要源数据库停止业务吗

DRS任务分为全量、增量、全量+增量三种模式，不同链路支持的模式不同。

- 全量模式：该模式为数据库一次性迁移，适用于可中断业务的数据库迁移场景，全量迁移将非系统数据库的全部数据库对象和数据一次性迁移至目标端数据库。
- 增量模式：该模式通过解析日志等技术，将源端产生的增量数据持续迁移至目标端数据库。
- 全量+增量模式：该模式为数据库持续性迁移，适用于对业务中断敏感的场景，通过全量迁移过程完成目标端数据库的初始化后，增量迁移阶段通过解析日志等技术，将源端持续迁移至目标端数据库，保持数据一致。

增量模式、全量+增量模式的不会自动结束，会持续将源端产生的增量数据迁移至目标端数据库，需用户判断并手动结束任务，可参考[如何判断数据迁移任务可以停止](#)。

用户在创建任务时，可根据情况选择不同模式，DRS不需要源数据库停止业务，但需注意全量和增量不同阶段对数据库产生的影响，可参考[MySQL数据同步对源数据库和目标数据库有什么影响](#)。

1.19 什么是 SMN 主题

- 什么是SMN
消息通知服务（Simple Message Notification，SMN）是可靠的、可扩展的、海量的消息处理服务。它可以依据用户的需求主动推送通知消息，最终用户可以通过短信、电子邮件、应用等方式接收。
- 对DRS而言
SMN为其他服务均可使用的通用服务，仅需在SMN界面统一配置，如果在其他服务使用时已完成创建和订阅，DRS可以直接选择相应的订阅主题。
如果用户未创建和订阅SMN主题，需要先[创建主题](#)，作为发送消息和订阅通知的信道，为发布者和订阅者提供一个可以相互交流的通道。然后[添加订阅并请求订阅](#)，这样，订阅用户的华为账号冻结时才能够将告警信息通过主题发送给订阅这个主题的订阅者。完成创建和添加订阅后，后续的告警通知即可通过SMN服务发送到用户配置的订阅终端。

1.20 DRS 任务的双 AZ 和单 AZ 有什么区别

目前，DRS实时同步任务在创建任务时支持选择任务类型为“双AZ”或“单AZ”。

- 双AZ任务：一主一备的经典高可用架构，提高了DRS任务的可靠性。双AZ任务创建后，DRS会同时创建一主一备两个子任务，当主AZ任务发生异常，DRS会自动启动备AZ任务继续进行同步，提高同步任务的可靠性。适用于业务数据量较大，需要长期同步，并对中断时间有严格要求的场景。
- 单AZ任务：采用单个节点部署架构，每个任务只创建一个节点，性价比较高。适用于业务数据量较小，短期同步，并对中断时间无严格要求的场景。

目前仅部分链路支持任务类型的选择，详细说明可参考[双AZ任务切换](#)。

1.21 DRS 是否支持只迁移表结构不迁移数据

DRS是用于数据实时传输的云服务，暂不支持只迁移结构而不迁移数据。各链路支持的对象范围可在具体链路说明查看。

[实时迁移方案概览](#)

[实时同步方案概览](#)

[实时灾备方案概览](#)

1.22 MySQL 迁移、同步、灾备任务怎么迁移账号，是否支持修改密码

- MySQL实时迁移：支持在创建任务时选择是否进行账号迁移，详细说明可参考[迁移用户](#)。增量迁移阶段，客户可以通过页面配置支持DCL语句，不支持通过更新mysql.user表进行的账号变更。
- MySQL实时同步：不支持同步账号。
- MySQL实时灾备：不支持对系统库下自定义对象有操作权限的账号灾备；不支持进行灾备对象选择。灾备阶段支持DCL语句，不支持通过更新mysql.user表进行的账号变更。

1.23 影响 DRS 任务速度的因素有哪些以及如何进行耗时预估

影响因素

- 源库的读取吞吐能力
读取吞吐能力越高，迁移速度越快，耗时越少。影响吞吐的因素包括但不限于服务器规格、负载、磁盘IO性能、数据库限流等。
- 目标库的写入吞吐能力
写入吞吐能力越高，迁移速度越快，耗时越少。影响吞吐的因素包括但不限于服务器规格、负载、磁盘IO性能、数据库限流等。
- 可用的网络吞吐量

可用的网络吞吐越高，迁移速度越快，耗时越少。影响网络吞吐量的因素包括但不限于可用带宽、防火墙、网络设备限流设置等。

- 网络质量和网络延迟
网络延迟越小（影响因素包括但不限于源库目标库和DRS实例地理位置的远近程度），迁移速度越快，耗时越少。网络质量差（比如丢包率高）会降低迁移速度。
- DRS实例规格
DRS实例规格越大，迁移速度越快，耗时越少。
- 源数据的模型和分布
此类因素包括是否有主键、是否分区表、是否堆表、平均单行数据量大小、表的数量、索引数量等。
- 目标库是否已有数据
目标库已有数据可能导致迁移时数据冲突，导致性能下降。
- 目标库是否有触发器
目标库有触发器可能导致迁移时写入性能下降。
- 目标库备份和日志等相关设置
如无必要可在迁移期间关闭目标库备份和事务日志，可提升迁移性能。
- 源库增量数据产生速度
源库增量数据产生速度越快，增量数据追平的耗时越长。
- DRS任务数
不考虑其他因素导致的性能瓶颈，通过按表拆分成多个DRS任务，可提升总体迁移性能。

迁移时长预估

由于影响迁移时长的因素很多，因此没有通用的计算方式能准确的计算迁移耗时，用户可先参考[实时同步链路规格说明](#)进行评估。建议用户先使用和待迁移实例相同规格、负载、网络配置、数据模型的环境创建测试任务，以评估迁移时长。

迁移速度提升

全量：

- 如果受限于跟源库或目标库的网络带宽限制，需要扩大带宽；
- 源端负载不高的情况下，可以增加读和写的并发数；
- 扩大DRS的任务规格。

增量：

- 如果受限于跟源库或目标库的网络带宽限制，需要扩大带宽；
- 如果是受限于DRS规格的性能上限，需要改用更大规格的DRS任务；
- 如果受限于回放慢，增加回放并发数。

1.24 DRS 任务支持修改对象吗

实时迁移任务：已创建成功但是未确认启动的编辑状态时，可以修改迁移对象，任务启动后，不支持进行修改。

备份迁移任务：仅支持在创建时编辑备份迁移任务。对于已经启动的备份迁移任务，不支持修改。

实时同步任务：已创建成功但是未确认启动的编辑状态时，可以修改同步对象。同步任务在增量阶段，且为表级同步时，可以进行再次编辑对象的操作，库级同步或其他同步阶段不可以进行再编辑，详细操作可参考[编辑同步对象](#)。

数据订阅任务：已创建成功但是未确认启动的编辑状态时，可以修改选择对象，任务启动后，不支持进行修改。

实时灾备任务：为实例级灾备，不支持对象选择。

1.25 DRS 是否支持同一实例不同库之间的同步

DRS实时同步可通过对象名映射功能，修改目标数据库对象名称，使数据库对象在源数据库和目标数据库中的命名不同，从而实现同一实例不同库之间的同步，详细说明可参考[对象名映射](#)。

1.26 源或者目标数据库哪些操作，会影响 DRS 任务状态

以华为云RDS for MySQL为例，以下操作可能会影响DRS任务状态。

- 实例备份：一般情况下，实例备份对DRS任务无影响。
- 单机转主备：正常情况下对DRS任务无影响。
- 实例重启：重启数据库实例将导致短暂中断，在此期间，实例将不可用，DRS连接短暂中断。这种情况下，DRS会进行自动重试。若重试仍无法恢复，待实例状态正常后，可手动单击任务“操作”列上的“续传”继续任务。
- 主备倒换：实例主备切换期间，可能会造成几秒或几分钟的服务闪断，这种情况下，DRS会进行自动重试。若重试仍无法恢复，待实例状态正常后，可手动单击任务“操作”列上的“续传”继续任务。
- 规格变更：规格变更后，将会重启数据库，重启数据库实例将导致短暂中断，在此期间，实例将不可用，DRS连接短暂中断。这种情况下，DRS会进行自动重试。若重试仍无法恢复，待实例状态正常后，可手动单击任务“操作”列上的“续传”继续任务。
- 实例升级：升级数据库内核小版本会重启实例，重启数据库实例将导致短暂中断，在此期间，实例将不可用，DRS连接短暂中断。这种情况下，DRS会进行自动重试。若重试仍无法恢复，待实例状态正常后，可手动单击任务“操作”列上的“续传”继续任务。
- 实例异常：这种情况下，DRS会进行自动重试。若重试仍无法恢复，待实例状态正常后，可手动单击任务“操作”列上的“续传”继续任务。
- 限制连接会话数：DRS任务连接源和目标端需要一定数量的会话数，详细可参考[MySQL数据同步对源数据库和目标数据库有什么影响](#)。当连接数不足时，会导致DRS任务失败，用户可在调整数据库连接数后，单击任务“操作”列上的“续传”继续任务。
- 网络抖动：针对网络抖动造成的连接失败，DRS会自动进行重试，若重试仍无法恢复，待网络正常后，可手动单击任务“操作”列上的“续传”继续任务。
- 账号密码修改：修改数据库账号密码可能会导致DRS连接失败，详细可参考[源库或目标库修改密码后如何操作](#)。
- 账号权限修改：修改数据库账号权限可能会导致DRS权限不足，迁移数据失败。用户可重新为迁移账号赋权后，单击任务“操作”列上的“续传”继续任务。

- 源库日志清理：当源库日志（例如MySQL的binlog）被清理时，DRS无法从源库获取到和当前同步位点接续的日志，可能导致任务失败（例如[全量或增量阶段失败报错，关键词“binlog is not existed”](#)）。用户可参考[重置同步任务](#)内容进行重置，或重新创建任务开始同步。
- 数据库参数修改：DRS在启动任务前会对源数据库和目标数据库参数进行预检查，在预检查完成后到任务结束前，一般情况下不建议对数据库参数进行修改，避免因参数修改导致的任务失败。如果参数修改导致任务失败，用户可尝试在恢复参数后，单击任务“操作”列上的“续传”继续任务。

1.27 数据订阅与 MySQL 到 Kafka 同步的区别

对比项	MySQL到Kafka同步	数据订阅
支持的源数据库	<ul style="list-style-type: none"> • 华为云RDS for MySQL • 本地自建MySQL数据库 • ECS自建MySQL数据库 • 其他云MySQL数据库 	华为云RDS for MySQL
支持网络	支持公网网络、VPC网络、VPN、专线网络。	仅支持VPC网络，因未开放全网段路由，所以VPC容器形态网络暂不支持。
数据格式	Avro、JSON和JSON-C	SDK
数据类型	全量+增量数据。	仅增量数据。
同步对象	表数据、表结构。	数据更新和结构更新。
稳定性	有任务告警、监控，任务异常自动重试。	无任务告警、监控，任务异常手动重试
性能	支持多规格，可根据不同性能要求选择。	不支持规格选择，性能为极小规格的性能。
商用状态	已经商用，收费。	公测阶段，免费。

由于订阅任务当前存在较多订阅对象和操作约束，推荐使用性能、稳定性更高的[MySQL到Kafka同步链路](#)进行数据订阅。

1.28 为什么 DRS 不支持部分他云只读实例的备库做源库

对于DRS的增量或者全量+增量任务，不支持部分他云只读实例的备库做源库进行数据迁移。DRS增量迁移是通过读取源数据库的原始增量日志数据（例如MySQL的Binlog）后解析转换进行。

以MySQL的Binlog为例，部分他云只读MySQL备库的Binlog并不完整，会导致DRS无法进行增量数据同步。用户可使用mysqlbinlog下载并确认Binlog内容完整性。

下载命令：注意，该命令会把 \$binlogLogName文件后的所有日志下载下来，如果只是确认Binlog内容完整性，可以选择一个Binlog下载。

```
mysqlbinlog --no-defaults -h$sourceHost -u$sourceUsername -P$sourcePort -p$sourcePassword --raw --read-from-remote-server $binlogLogName --to-last-log
```

查看Binlog内容:

```
mysqlbinlog --base64-output=decode-rows -v $binlogLogName
```

- 正常Binlog

```
SET TIMESTAMP=1682563151/*!*/;
BEGIN
/*!*/;
# at 1102
#230427 10:39:11 server id 123453307 end_log_pos 1149 CRC32 0x1f5d6ebe      Table_map: `test`.`t` mapped to number 31
# at 1149
#230427 10:39:11 server id 123453307 end_log_pos 1192 CRC32 0x52c623c4      Write_rows: table id 131 flags: STMT
F
### INSERT INTO `test`.`t`
### SET
###   @1=2
###   @2='2'
# at 1192
#230427 10:39:11 server id 123453307 end_log_pos 1223 CRC32 0x0112f8e5      Xid = 1479589
COMMIT/*!*/;
SET @@SESSION.GTID_NEXT= 'AUTOMATIC' /* added by mysqlbinlog */ /*!*/;
DELIMITER ;
# End of log file
```

1.29 DRS 支持 caching_sha2_password 加密用户迁移吗

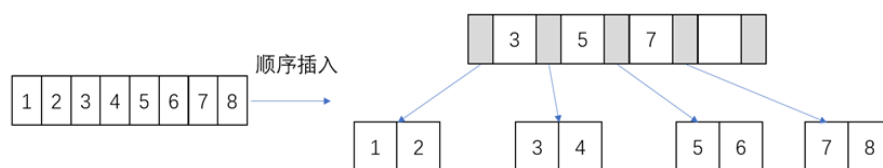
DRS不支持caching_sha2_password插件加密的用户进行迁移。
caching_sha2_password是基于SSL（本地RSA的非对称公私钥）的方式验证的，因此DRS不支持迁移。

1.30 为什么迁移前后数据库磁盘空间使用量不一致

可能原因

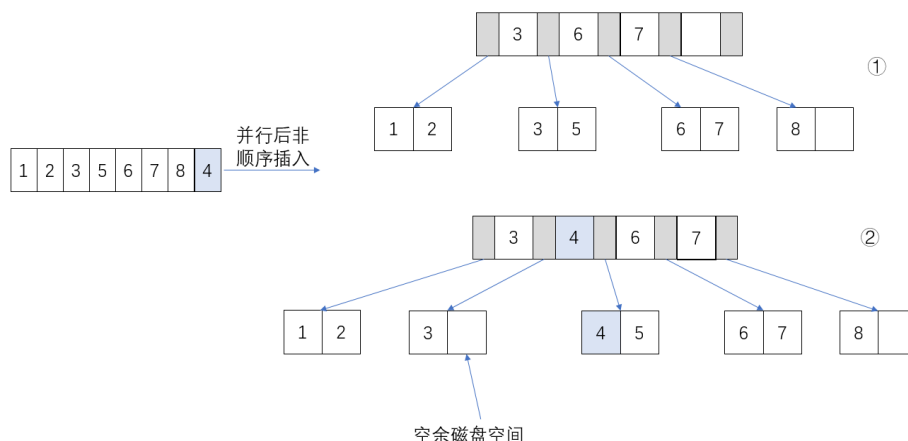
- 原因1：DRS在全量迁移阶段，为了保证迁移性能和传输的稳定性，采用了行级并行的迁移方式。当源数据库数据紧凑的情况下，通过DRS迁移到上云后，可能会出现数据膨胀现象，导致目标数据库磁盘空间使用量大于源数据库磁盘空间使用量。
并行时顺序插入原理可参考图1-16。

图 1-16 顺序插入



当并行后有非顺序插入数据时，如图1-17，当最后数据4插入时，会导致数据3、5叶子节点分裂，分裂后的叶子节点存在磁盘空余空间。

图 1-17 并行非顺序插入



- 原因2：目标库开启Binlog日志时，全量阶段数据的写入会在目标端产生Binlog日志文件，占用部分磁盘空间，导致目标数据库磁盘空间使用量大于源数据库磁盘空间使用量。

解决方法

磁盘空间使用量的大小不能判断迁移前后数据是否一致，可使用DRS的[数据对比](#)功能，对比源数据库和目标数据库迁移前后数据是否一致。

1.31 DRS 发送 Kafka 消息序列化方式是什么

DRS发送到Kafka的数据，key序列化器是 `org.apache.kafka.common.serialization.StringSerializer`，value序列化器是 `org.apache.kafka.common.serialization.ByteArraySerializer`，字符编码为utf8。

1.32 DRS 可支持自建的 MySQL 数据库迁移吗？

DRS支持的MySQL链路方向

- 自建MySQL到华为云RDS for MySQL。
- 华为云RDS for MySQL到自建MySQL。

DRS不支持链路方向

- 自建MySQL到自建MySQL

📖 说明

在华为云创建的ECS上自建的数据库，属于自建数据库。

1.33 DRS 会主动清空目标表数据吗？

DRS不会主动操作目标库数据，只做同步。

- 如果DRS从源库同步数据到目标表，目标表有数据，DRS不感知，只做同步。

- DRS新增一个节点，不会清空目标库数据。
- DRS重置也不会清空目标库数据，如您需要清空数据，请手动操作。

1.34 入云，出云，自建有什么不同？

- 入云：从非本region云上的数据库到本region的云上数据库的链路。
- 出云：从本region的云上数据库到非本region云上的数据库的链路。
- 自建：从非本region云上的数据库到非本region云上的数据库的链路。

1.35 DRS 支持自建 MySQL 到自建 MySQL 的链路吗？

DRS的同步和迁移都没有自建MySQL到自建MySQL的链路。

解决方案：请修改源端和目标库的为云上数据库。

1.36 DRS 最小的传输粒度是什么？

全量和增量均是以行记录为传输粒度。

1.37 在数据迁移过程中，对源库中目标库进行操作，会对DRS有什么影响？

1. 对源库操作，drs任务会正常迁移数据。但是如果对源库操作是迁移中不支持的DDL，DML会导致数据对比不一致。
2. 对目标库操作
 - a. 进行添加字段，drs任务是正常的。
 - b. 对表进行修改字段，减少字段，会导致drs任务失败。
 - c. 进行手动添加记录，修改，删除记录，会导致drs对比数据结果不一致。
 - d. 扩容操作是rds内部操作，不影响drs任务。

2 网络及安全

2.1 数据复制服务有哪些安全保障措施

数据复制服务主要有以下安全保障措施。

网络

- 使用安全组确保访问源可信。
- 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），可以实现对数据复制服务的管理权限控制。

2.2 如何处理迁移过程中出现的网络中断

迁移过程中如果出现网络中断，可先观察任务状态，当如下状态的迁移任务出现失败时，可在任务列表上单击“续传”，进行任务续传。

- 全量迁移
- 增量迁移
- 全量同步
- 增量同步

2.3 迁移过程中会使用哪些数据库账户

MySQL

为了确保用户数据库成功迁移到本云关系型数据库MySQL实例，创建迁移任务时，数据复制服务自动为目标数据库实例创建drsFull和drsIncremental临时账户，任务结束后，数据复制服务会自动删除这两个账户。

须知

- 禁止使用用户自建的drsFull和drsIncremental账号作为DRS任务连接数据库的账号。
- 删除、重命名和修改这些账户的密码和权限，会导致任务出错。

表 2-1 MySQL 实例创建的数据库账户

账户	使用场景	Host	说明
drsFull	全量迁移	关系型数据库 MySQL实例	用户启动全量迁移任务时，系统会在目标数据库中添加该账户，用于迁移数据。
drsIncremental	增量迁移	关系型数据库 MySQL实例	用户启动增量迁移任务时，系统会在目标数据库中添加该账户，用于迁移数据。

2.4 如何通过设置 VPC 安全组，实现 DRS 任务网络互通

基于安全的考虑，本云VPC与外部网络是隔离的，VPC内是无法访问外部的弹性IP（如其他云数据库的弹性IP、云下数据库的弹性IP等）。但数据库迁移场景需要确保本云VPC内的DRS实例与源、目标数据库可连通，为此，您需要分别设置源数据库、目标数据库、DRS实例所在安全组的出或者入方向规则，实现数据库迁移。入方向指从外部访问安全组规则下的实例，出方向指安全组规则下的实例访问安全组外的实例。安全组的出入口规则一般需要满足“严进宽出”的要求。

一般情况下，创建入云任务时，DRS实例默认与目标数据库在同一VPC内，是连通的。此时需要设置[源数据库所在安全组](#)放通DRS实例IP和源数据库自身的端口，设置[DRS实例所在安全组设置](#)（即目标数据库所在安全组）放通源数据库的IP、端口。

同理，创建出云任务时，DRS实例默认与源数据库在同一VPC内，是连通的。此时需要设置[目标数据库所在安全组](#)放通DRS实例IP和目标数据库自身的端口，设置[DRS实例所在安全组设置](#)（即源数据库所在安全组）是否放通目标数据库的IP、端口。

本章节以源和目标数据库均为RDS for MySQL为例，具体排查方法可参考如下内容。

DRS 实例所在安全组设置

DRS实例所在VPC安全组出方向规则需要放通源数据库和目标数据库的IP、端口，允许DRS实例访问安全组外的数据库。

步骤1 在任务列表中，单击DRS任务名称。

步骤2 进入“基本信息”页签，在“实例信息”模块下，单击内网安全组。

迁移实例信息			
引擎类型	MySQL	网络类型	VPC网络
内网IP		迁移模式	全量 + 增量
迁移对象	表级迁移	VPC	default_vpc-
迁移实例所在子网	default_subnet	内网安全组	default

步骤3 进入安全组基本信息页，选择“出方向规则”页签。

步骤4 单击“添加规则”进行配置。

DRS实例所在VPC安全组出方向规则需要放通源数据库和目标数据库的IP、端口（IP地址和端口只要包含目标数据库和源数据库的IP和端口即可）。



----结束

目标数据库所在安全组

目标数据库所在VPC安全组入方向规则需要放DRS实例IP和目标数据库自身的端口，允许DRS实例通过端口访问。

步骤1 在RDS的“实例管理”页面，单击目标实例名称。

步骤2 进入“基本信息”页签，在“连接信息”模块下，单击安全组。



步骤3 进入安全组基本信息页，选择“入方向规则”页签。

步骤4 单击“添加规则”进行配置。

目标数据库所在VPC安全组入方向规则需要放DRS实例IP和目标数据库自身的端口（IP地址和端口只要包含DRS的IP和自身的端口即可）。



----结束

源数据库所在安全组

源数据库所在VPC安全组入方向规则需要放DRS实例IP和源数据库自身的端口，允许DRS实例通过端口访问。

步骤1 在RDS的“实例管理”页面，单击目标实例名称。

步骤2 进入“基本信息”页签，在“连接信息”模块下，单击安全组。



步骤3 进入安全组基本信息页，选择“入方向规则”页签。

步骤4 单击“添加规则”进行配置。

源数据库所在VPC安全组入方向规则需要放DRS实例IP和源库自身的端口（IP地址和端口只要包含DRS的IP和自身的端口即可）。



----结束

2.5 如何处理迁移实例和数据库网络连接异常

数据迁移前请确保完成网络准备和安全规则设置。如果连接异常，请按照本节方法排查网络配置是否正确。

本节将以MySQL到RDS for MySQL的迁移为示例，从三种迁移场景（跨云数据库实时迁移、本地数据库实时迁移、ECS自建数据库实时迁移）进行说明。

跨云数据库实时迁移

1. 网络准备。

源数据库需要开放公网访问。

- 源数据库的网络设置：

源数据库MySQL实例需要开放外网域名的访问。

以阿里云云数据库RDS MySQL迁移到本云云数据库 RDS for MySQL为例，一般情况下，阿里云RDS MySQL不提供外网地址，需要通过申请外网地址来允许外部的应用对接，具体的操作及注意事项可以参考源数据库所在云提供的相关指导。

- 目标数据库的网络设置：

目标数据库默认与DRS迁移实例处在同一个VPC内，网络是互通的，不需要进行任何设置。

2. 安全规则准备。

- 源数据库的安全规则设置：

源数据库MySQL实例需要将DRS迁移实例的弹性公网IP添加到其网络白名单中，确保源数据库MySQL实例可以与上述弹性公网IP连通。

以上讲述的是精细配置白名单的方法，还有一种简单设置白名单的方法，在安全允许的情况下，可以将源数据库MySQL实例的网络白名单设置为0.0.0.0/0，代表允许任何IP地址访问该实例。

上述的网络白名单是为了进行数据迁移设置的，迁移结束后可以删除。

- **目标数据库安全规则设置：**

- 目标数据库默认与DRS迁移实例处在同一个VPC，网络是互通的，DRS可以直接写入数据到目标数据库。
- 设置目标数据库所在VPC的安全组，确保DRS实例IP的出方向到本地数据数据库的IP、监听端口放通。具体设置如下图所示：

图 2-1 出口规则



本地数据库实时迁移

1. **网络准备：**

- **源数据库的网络设置：**

本地MySQL数据库迁移至本云云数据库 RDS for MySQL的场景，一般可以使用VPN网络和公网网络两种方式进行迁移，您可以根据实际情况为本地MySQL数据库开放公网访问或建立VPN访问。一般推荐使用公网网络进行迁移，该方式下的数据迁移过程较为方便和经济。

- **目标数据库的网络设置：**

- 若通过VPN访问，请先开通VPN服务，确保源数据库MySQL和目标端本云云数据库 RDS for MySQL的网络互通。
- 若通过公网网络访问，本云云数据库 RDS for MySQL实例不需要进行任何设置。

2. **安全规则准备：**

a. **源数据库的安全规则设置：**

- 若通过公网网络进行迁移，源数据库MySQL需要将DRS迁移实例的弹性公网IP添加到其网络白名单内，使源数据库与本云的网络互通。在设置网络白名单之前，需要获取DRS迁移实例的弹性公网IP，具体方法如下：

DRS迁移实例创建成功后，可在“源库及目标库”页面获取DRS迁移实例的弹性公网IP。

- 若通过VPN网络进行迁移，源数据库MySQL需要将DRS迁移实例的私有IP添加到其网络白名单内，使源数据库与本云的网络互通。DRS迁移实例创建成功后，可在“源库及目标库”页面获取DRS迁移实例的私有IP。

以上白名单是为了进行迁移针对性设置的，迁移结束后可以删除。

b. 目标数据库安全规则设置：

- 目标数据库默认与DRS迁移实例处在同一个VPC，网络是互通的，DRS可以直接写入数据到目标数据库。
- 设置目标数据库所在VPC的安全组，确保DRS实例IP的出方向到本地数据库的IP、监听端口放通。具体设置如下图所示：

图 2-2 出口规则



ECS 自建数据库实时迁移

1. 网络准备：

- 源数据库所在的region要和目标端本云云数据库 RDS for MySQL实例所在的region保持一致。
- 源数据库可以与目标端本云云数据库 RDS for MySQL实例在同一个VPC，也可以不在同一个VPC。
 - 当源库和目标库处于同一个VPC时，网络默认是互通的。
 - 当不在同一个VPC的时候，要求源数据库实例和目标端本云云数据库 RDS for MySQL实例所处的子网处于不同网段，此时需要通过建立对等连接实现网络互通。具体建立VPC对等连接的方法，可参考《[虚拟私有云用户指南](#)》。

2. 安全规则准备：

- 同一VPC场景下，默认网络是连通的，不需要单独设置安全组。
- 不同VPC场景下，通过建立对等连接就可以实现网络互通，不需要单独设置安全组。

排查 iptables 设置

以源数据库为本云ECS自建数据库为例，如果在上述的操作后，仍无法连通，此时需要额外排查iptables设置，因为HOSTGUARD服务在DRS发起频繁连接请求失败时，会将请求IP加入黑名单中。

1. 登录弹性云服务器。

2. 执行以下命令，排查是否有DENY相关的项目包含DRS实例的IP，一般项目名称为IN_HIDS_MYSQLD_DENY_DROP。
iptables --list
3. 如果存在，执行以下命令，查询iptables入方向规则列表，获取具体规则编号（line-numbers）。
iptables -L INPUT --line-numbers
4. 执行以下命令，删除DRS实例的IP相关的入方向规则（注意：必须从后往前删，不然line-numbers会更新，需要重新查询）。
iptables -D 规则项目名 具体规则编号
5. 删除相关iptables规则后重新进行测试连接即可。

相关文档

更多关于DRS源数据库部署在不同位置的网络设置说明，可参考[这里](#)。

2.6 不同 VPC 场景下，如何实现源库和目标库的网络互通

DRS支持通过多种方式的网络进行数据迁移，包括：VPC网络、VPN网络、专线网络和公网网络，其中VPC网络适合云上同Region数据库之间的迁移。

- 源数据库所在的区域要和目标数据库实例所在的区域保持一致。
- 源数据库可以和目标数据库在同一VPC内，也可以在不同VPC内。
- 当源数据库和目标数据库处于同一个VPC内的时候，默认网络是连通的，不需要单独设置安全组。
- 当源数据库和目标数据库不在同一个VPC内的时候，要求源数据库和目标数据库所处的子网处于不同网段，不能重复或交叉，此时需要通过对等连接实现网络互通。
- 目前DRS暂不支持跨租户通过VPC网络实现源库、目标库的网络互通。如果有需要，用户可通过手动创建对等连接后，选择“VPN网络”实现网络互通。

对等连接限制

- 有重叠子网网段的VPC建立的对等连接，可能不生效。
- 两个VPC之间不能同时建立多个VPC对等连接。
- 不同区域的VPC不能创建对等连接。
- 如果两个VPC的CIDR有重叠，建立对等连接时，只能针对子网建立对等关系。如果两个VPC下的子网网段有重叠，那么该对等关系不生效。建立对等连接时，请确保两个VPC之间没有重叠的子网。
- 对等连接建立后，需要在本端VPC、对端VPC分别添加对方子网的路由才能通信。
- VPC A与B、C分别建立对等连接，如果B、C两个VPC的网段有重叠，A中无法添加具有相同目的网段的路由。
- 为了安全起见，请不要接受来自未知账号的对等连接申请。
- 对等连接双方账号都有权限删除对等连接，一方删除对等连接后，对等连接的所有信息会被立刻删除，包括对等连接关联的路由信息。
- VPC对等连接路由存在时，VPC无法被删除。
- 支持在华为云中国站和国际站同一区域创建对等连接。

- VPC1与VPC2创建对等连接，默认情况下VPC2不能通过VPC1的EIP访问公网。您可以通过使用NAT网关服务或[配置SNAT服务器](#)，使得VPC2下的弹性云服务器可以通过VPC1下绑定了EIP的弹性云服务器访问Internet。具体实现方式请参见[无公网IP的弹性云服务器访问Internet](#)。

具体设置VPC对等连接的方法，可参考《[虚拟私有云用户指南](#)》。

对等连接建立后，需要在本端VPC、对端VPC分别添加对方子网的路由才能通信，详细操作可参考[添加VPC对等连接路由](#)。

在添加对等路由时，建议添加网段路由信息。如果添加点对点路由，DRS任务重建后实例IP会发生改变，此时需要重新添加路由，否则会导致网络不通。

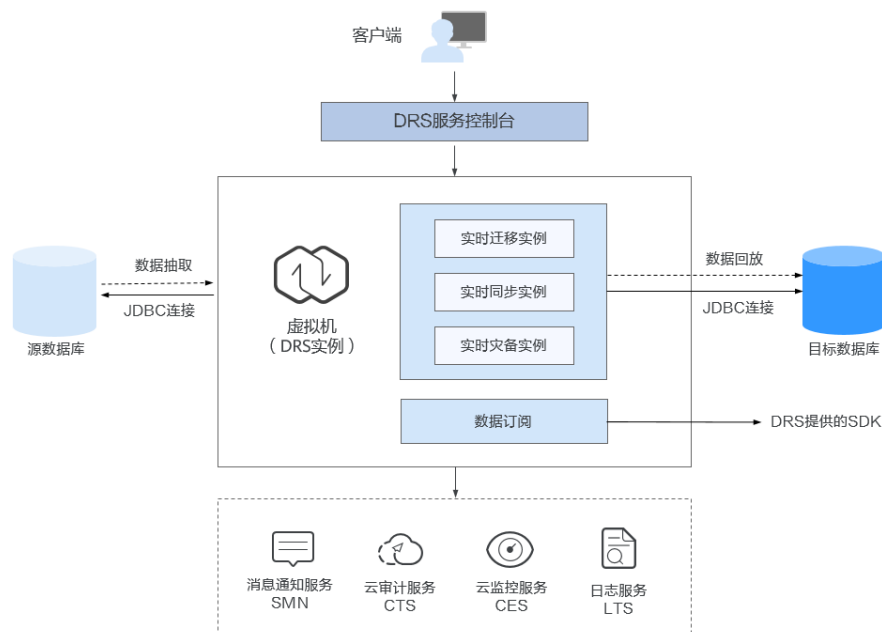
2.7 DRS 公网网络的 EIP 带宽是多少

DRS创建公网网络类型任务时，需要用户指定EIP，因此DRS公网网络的EIP带宽为用户选择绑定的EIP带宽。

更多关于弹性公网IP的说明请参见：[申请弹性公网IP](#)。

2.8 DRS 支持跨账号云数据库迁移吗

图 2-3 DRS 产品架构



目前，DRS支持的网络类型有公网网络、VPC网络、VPN网络和专线网络，不同场景下，用户可根据情况进行选择。

原理上DRS采用JDBC连接，无需在用户的源数据库、目标数据库节点部署程序，只需用户的源数据库和目标数据库允许DRS实例节点通过所选择的网络访问和连接即可。

例如，用户需要将A账号的RDS-A实例迁移到B账号的RDS-B，可选择为RDS-A实例申请弹性公网IP并绑定，在B账号创建入云的DRS任务，选择通过公网网络进行迁移。

3 权限管理

3.1 如何设置最小化权限且独立的使用 DRS 的 Oracle 账号

源数据库为Oracle的全量迁移至少需要CREATE SESSION, SELECT ANY TRANSACTION, SELECT ANY TABLE, SELECT ANY DICTIONARY权限（当目标库为PostgreSQL时，还需要SELECT ANY SEQUENCE权限），源数据库为Oracle的增量迁移还需要日志解析权限。本小节主要介绍如何使用DRS设置最小化权限且独立的Oracle账号的具体操作。

- 全量迁移模式。

- a. 创建一个用户用于迁移，此处以User1为例。

参考命令：CREATE USER *User1* IDENTIFIED BY *pwd*

📖 说明

其中User1为用户名，pwd为密码。

- b. 使用sys用户或者具有DBA权限的用户执行以下语句对User1用户赋予需要的权限。

参考命令：GRANT CREATE SESSION, SELECT ANY TRANSACTION, SELECT ANY TABLE, SELECT ANY DICTIONARY TO *User1*

- 全量+增量迁移模式。

- a. 创建一个用户用于迁移，此处以User1为例。

参考命令：CREATE USER *User1* IDENTIFIED BY *pwd*

- b. 使用sys用户或者具有DBA权限的用户执行以下语句对User1用户赋予需要的权限。

参考命令：GRANT CREATE SESSION, SELECT ANY TRANSACTION, SELECT ANY TABLE, SELECT ANY DICTIONARY TO *User1*

- c. 使用sys用户或者具有DBA权限的用户执行以下语句对User1用户赋予日志解析权限。

- Oracle版本小于12C。

参考命令：GRANT EXECUTE_CATALOG_ROLE TO *User1*

- Oracle版本大于等于12C。

参考命令：GRANT EXECUTE_CATALOG_ROLE TO *User1*

参考命令：GRANT LOGMINING TO *User1*

3.2 DRS 要求的 MySQL 权限有哪些

DRS在迁移、同步、灾备过程中，对账号有一定的权限要求，本章节主要介绍MySQL引擎的权限要求。

权限要求

- 源和目标库的连接账号需要具有登录权限，如果没有该账号，可以通过如下方式创建，以user1为例。
参考语句：**CREATE USER** 'user1'@'host' **IDENTIFIED BY** 'password';
- DRS的实时迁移、实时同步、实时灾备功能的权限要求，[表1 权限要求](#)中以user1为例提供参考语句。

表 3-1 权限要求及参考语句

功能模块	源/业务数据库	目标/灾备数据库
实时迁移	<p>全量迁移权限要求: SELECT、SHOW VIEW、EVENT。 参考语句：GRANT SELECT, SHOW VIEW, EVENT ON *.* TO 'user1';</p> <p>全量+增量迁移权限要求: SELECT、SHOW VIEW、EVENT、LOCK TABLES、REPLICATION SLAVE、REPLICATION CLIENT。</p> <ul style="list-style-type: none"> 其中，REPLICATION SLAVE、REPLICATION CLIENT是全局权限，必须单独开启。参考语句如下：GRANT REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO 'user1'; SELECT、SHOW VIEW、EVENT、LOCK TABLES是非全局权限，参考语句如下：GRANT SELECT, SHOW VIEW, EVENT, LOCK TABLES ON [待迁移数据库].* TO 'user1'; 当业务数据库为8.0.2以上版本时，还需要有XA_RECOVER_ADMIN权限。 	<p>全量迁移权限要求: SELECT、CREATE、ALTER、DROP、DELETE、INSERT、UPDATE、INDEX、EVENT、CREATE VIEW、CREATE ROUTINE、TRIGGER、REFERENCES、WITH GRANT OPTION。当目标库为8.0.14-8.0.18版本时，还需要有SESSION_VARIABLES_ADMIN权限。 参考语句：GRANT SELECT, CREATE, ALTER, DROP, DELETE, INSERT, UPDATE, INDEX, EVENT, CREATE VIEW, CREATE ROUTINE, TRIGGER ON *.* TO 'user1' WITH GRANT OPTION;</p> <p>全量+增量迁移权限要求: SELECT、CREATE、ALTER、DROP、DELETE、INSERT、UPDATE、INDEX、EVENT、CREATE VIEW、CREATE ROUTINE、TRIGGER、REFERENCES、WITH GRANT OPTION。当目标库为8.0.14-8.0.18版本时，还需要有SESSION_VARIABLES_ADMIN权限。 参考语句：GRANT SELECT, CREATE, ALTER, DROP, DELETE, INSERT, UPDATE, INDEX, EVENT, CREATE VIEW, CREATE ROUTINE, TRIGGER, REFERENCES ON [待迁移数据库].* TO 'user1' WITH GRANT OPTION;</p>

功能模块	源/业务数据库	目标/灾备数据库
实时同步	<p>SELECT、SHOW VIEW、EVENT、LOCK TABLES、REPLICATION SLAVE、REPLICATION CLIENT。全量+增量任务，当源数据库为8.0.2及以上版本时，还需要授予XA_RECOVER_ADMIN权限，以防启动或编辑时未提交的XA事务导致数据有损。</p> <ul style="list-style-type: none"> 其中，REPLICATION SLAVE、REPLICATION CLIENT是全局权限，必须单独开启。参考语句如下： GRANT REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO 'user1'; SELECT、SHOW VIEW、EVENT、LOCK TABLES是非全局权限，参考语句如下： GRANT SELECT, SHOW VIEW, EVENT, LOCK TABLES ON [待同步数据库].* TO 'user1'; 当业务数据库为8.0.2以上版本时，还需要有XA_RECOVER_ADMIN权限。 	<p>SELECT、CREATE、DROP、DELETE、INSERT、UPDATE、ALTER、CREATE VIEW、CREATE ROUTINE、REFERENCES。</p> <p>参考语句：GRANT SELECT, CREATE, DROP, DELETE, INSERT, UPDATE, ALTER, REFERENCES ON [待同步数据库].* TO 'user1';</p>

功能模块	源/业务数据库	目标/灾备数据库
实时灾备	<p>SELECT、CREATE、ALTER、DROP、DELETE、INSERT、UPDATE、TRIGGER、REFERENCES、SHOW VIEW、EVENT、INDEX、LOCK TABLES、CREATE VIEW、CREATE ROUTINE、ALTER ROUTINE、CREATE USER、RELOAD、REPLICATION SLAVE、REPLICATION CLIENT、WITH GRANT OPTION，RDS for MySQL实例的root账户默认已具备上述权限。当业务数据库为8.0.14-8.0.18版本时，还需要有SESSION_VARIABLES_ADMIN权限。当业务数据库为8.0.2及以上版本时，还需要授予XA_RECOVER_ADMIN权限，以防启动时未提交的XA事务导致数据有损。RDS for MySQL实例的root账户默认已具备上述权限。</p> <p>当业务数据库为8.0.2以上版本时，还需要有XA_RECOVER_ADMIN权限。</p> <p>参考语句：GRANT SELECT,CREATE,ALTER,DROP,DELETE,INSERT,UPDATE,TRIGGER,SHOW VIEW,EVENT,INDEX,LOCK TABLES,CREATE VIEW,CREATE ROUTINE,ALTER ROUTINE,CREATE USER,RELOAD,REPLICATION SLAVE,REPLICATION CLIENT ON *.* TO 'user1';</p>	<p>SELECT、CREATE、ALTER、DROP、DELETE、INSERT、UPDATE、TRIGGER、REFERENCES、SHOW VIEW、EVENT、INDEX、LOCK TABLES、CREATE VIEW、CREATE ROUTINE、ALTER ROUTINE、CREATE USER、RELOAD、REPLICATION SLAVE、REPLICATION CLIENT、WITH GRANT OPTION，RDS for MySQL实例的root账户默认已具备上述权限。当灾备数据库为8.0.14-8.0.18版本时，还需要有SESSION_VARIABLES_ADMIN权限。</p> <p>参考语句：GRANT SELECT,CREATE,ALTER,DROP,DELETE,INSERT,UPDATE,TRIGGER,REFERENCES,SHOW VIEW,EVENT,INDEX,LOCK TABLES,CREATE VIEW,CREATE ROUTINE,ALTER ROUTINE,CREATE USER,RELOAD,REPLICATION SLAVE,REPLICATION CLIENT ON *.* TO 'user1'@'%' WITH GRANT OPTION;</p>

📖 说明

请在以上参考语句后执行**flush privileges;**使授权生效。

- 用户迁移权限要求

用户迁移时，如果源数据库为8.0版本，需要有mysql.user表的SELECT权限；如果为5.7及以下版本，则需要mysql系统库的SELECT权限，源数据库为阿里云数据库，则账户需要同时具有mysql.user和mysql.user_views的SELECT权限。

参考语句：

```
GRANT SELECT ON mysql.user TO 'user1'@'host' ;
```

```
GRANT SELECT ON mysql.* TO 'user1'@'host' ;
```

```
GRANT SELECT ON mysql.user_view TO 'user1';
```

目标数据库账号需要有所有库的SELECT, INSERT, UPDATE, DELETE, WITH GRANT OPTION权限。

参考语句: **GRANT SELECT, INSERT, UPDATE, DELETE ON *.* TO 'user1' WITH GRANT OPTION;**

- 函数、存储过程迁移权限要求

对于源数据库为MySQL 5.6、5.7, 如果您需要迁移函数、存储过程, 则需要MySQL系统库的proc表的SELECT权限。

参考语句:

```
GRANT SELECT ON `mysql`.`proc` TO 'user1'@'host';
```

对于源数据库为MySQL 8.0以上版本的时候, 如果您需要迁移函数、存储过程, 则需要整个实例的SELECT权限。对于MySQL 8.0.20版本以上, 也可以使用SHOW_ROUTINE权限来迁移函数、存储过程。参考语句:

```
GRANT SELECT ON *.* TO 'user1'@'host';
```

或者:

```
GRANT SHOW_ROUTINE ON *.* TO 'user1'@'host';
```

参考资料: https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_show-routine

DRS迁移完成后, 可通过[数据对比](#)功能查看迁移前后数据是否一致。

授权操作说明

- 创建用户

操作方式:

```
CREATE USER 'username'@'host' IDENTIFIED BY 'password';
```

· username: 待创建的账号。

· host: 允许该账号登录的主机, 如果允许该账号从任意主机登录数据库, 可以使用%。

· password: 账号的密码。

例如: 授予drsmigration账号具备所有数据库和表的所有权限, 并允许从任意主机登录数据库, 命令如下。

```
CREATE USER 'drsmigration'@'%' IDENTIFIED BY 'Drs123456';
```

- 授予相应权限

操作方式:

```
GRANT privileges ON databasename.tablename TO 'username'@'host' WITH GRANT OPTION;
```

```
flush privileges;
```

· privileges: 授予该账号的操作权限, 如SELECT、INSERT、UPDATE等, 如果要授予该账号所有权限, 则使用ALL

· databasename: 数据库名。如果要授予该账号具备所有数据库的操作权限, 则使用*。

· tablename: 表名。如果要授予该账号具备所有表的操作权限, 则使用*。

- username: 待授权的账号。
- host: 允许该账号登录的主机, 如果允许该账号从任意主机登录, 则使用%。
- WITH GRANT OPTION: 授予该账号使用GRANT命令的权限, 该参数为可选。

例如: 创建一个账号, 账号名为drsmigration, 密码为Drs123456, 并允许从任意主机登录数据库, 命令如下。

```
GRANT ALL ON *.* TO 'drsmigration'@'%';
```

3.3 如何将源数据库的用户与权限导出, 再导入到目标数据库

步骤1 选择一台可以访问源数据库的虚拟机。

步骤2 执行如下命令后, 输入密码并回车, 将源库用户导出到临时文件“users.sql”中。

```
mysql -h 'host' -u 'user' -p -N $@ -e "SELECT CONCAT('SHOW GRANTS FOR '", user, "'@'", host, "';") AS query FROM mysql.user" > /tmp/users.sql
```

其中的'host'替换为源数据库的访问IP地址, 'user'替换为源数据库的用户名。

步骤3 执行如下命令, 将源数据库中原有用户的授权信息导出到文件“grants.sql”中。

```
mysql -h 'host' -u 'user' -p -N $@ -e "source /tmp/users.sql" > /tmp/grants.sql  
sed -i 's/$/;/g' /tmp/grants.sql
```

其中的'host'替换为源数据库的访问IP地址, 'user'替换为源数据库的用户名。

步骤4 命令运行成功后, 打开“grants.sql”文件可以看到类似以下的结果。

```
-- Grants for root@%  
GRANT ALL PRIVILEGES ON *.* TO 'root'@'%';  
  
-- Grants for testt@%  
GRANT SELECT, INSERT, UPDATE, DELETE ON *.* TO 'testt'@'%';  
  
-- Grants for debian-sys-maint@localhost  
GRANT ALL PRIVILEGES ON *.* TO 'debian-sys-maint'@'localhost' WITH GRANT OPTION;  
  
-- Grants for mysql.session@localhost  
GRANT SUPER ON *.* TO 'mysql.session'@'localhost';  
GRANT SELECT ON `performance_schema`.* TO 'mysql.session'@'localhost';  
GRANT SELECT ON `mysql`.`user` TO 'mysql.session'@'localhost';  
  
-- Grants for mysql.sys@localhost  
GRANT USAGE ON *.* TO 'mysql.sys'@'localhost';  
GRANT TRIGGER ON `sys`.* TO 'mysql.sys'@'localhost';  
GRANT SELECT ON `sys`.`sys_config` TO 'mysql.sys'@'localhost';  
  
-- Grants for root@localhost  
GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost' WITH GRANT OPTION;  
GRANT PROXY ON '@' TO 'root'@'localhost' WITH GRANT OPTION;
```

步骤5 在**步骤4**显示的结果中, 可以看到源数据库中所有的用户以及对应的权限, 请选择所有需要的用户, 逐个添加到本云关系型数据库MySQL中, 具体操作请参考《关系型数据库用户指南》中的“[创建账号](#)”章节。

----结束

3.4 创建 DRS 任务后执行自动功能失败

当用户创建DRS任务使用的是子账号，执行定时启动任务、全量完成自动结束、自动续传失败任务、定时对比任务等自动功能时，可能会访问全局服务或是Region级别服务，需要使用“账户委托”，否则会导致执行失败。

说明

因为缺少委托而导致定时启动失败的任务，需要在增加委托后重建任务。

解决方案

- 方法一：使用主账号重新创建一次任务，主账号默认有Security Administrator权限，可在创建任务后将委托创建出来。
- 方法二：使用主账号在子账号所在的用户组添加Security Administrator权限后，重新创建任务。添加权限的具体操作请参见：[创建用户并授权使用DRS](#)。
- 方法三：手动添加“账户委托”，添加步骤如下：
 - a. 使用主账号登录华为云，在右上角单击“控制台”。
 - b. 在控制台页面，鼠标移动至右上方的账号名，在下拉列表中选择“统一身份认证”。
 - c. 在统一身份认证页面，单击左侧导航窗格中“委托”，进入“委托”页面。
 - d. 在“委托”页面，单击右上方的“创建委托”进行委托创建。
 - e. 填写委托名称为“DRS_AGENCY”，委托类型为“普通账号”时，委托的账号为“op_svc_rds”；委托类型为“云服务”时，选择“关系型数据库MySQL”；持续时间为“永久”，完成后单击“完成”。

图 3-1 创建委托

* 委托名称

* 委托类型 普通账号
将账号内资源的操作权限委托给其他华为云账号。
 云服务
将账号内资源的操作权限委托给华为云服务。

* 委托的账号

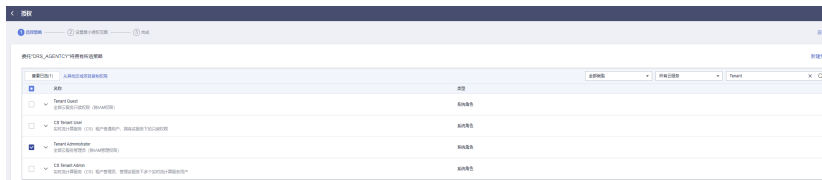
* 持续时间

描述

0/255

- f. 在“选择策略”页面，选择DRS_AGENCY的授权策略，委托权限需配置全局的 Tenant Administrator，完成后单击右下角的“下一步”。

图 3-2 选择策略

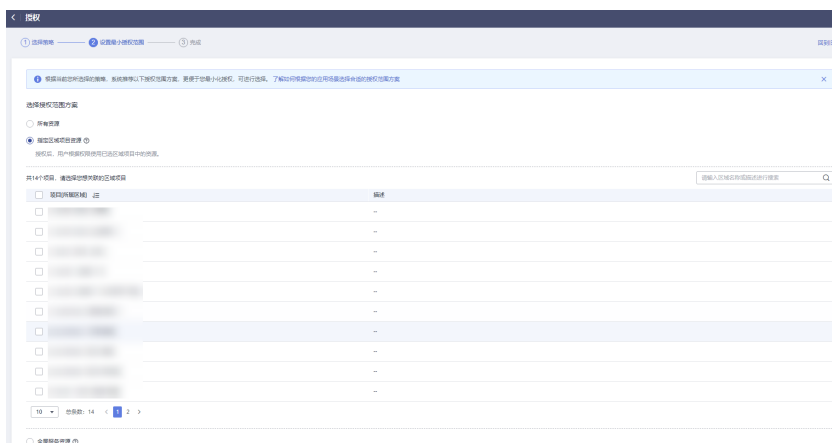


- g. 在“设置最小授权范围”页面，先选择全局服务资源授权后，再基于指定区域设置最小授权范围，完成后单击右下角的“确定”。

图 3-3 全局服务资源授权



图 3-4 指定区域项目授权



- h. 授权完成后，单击委托名称，在“授权记录”中可看到全局服务和指定区域两条授权记录。

图 3-5 授权记录



- i. 权限生效时间提醒，您选中的OBS权限由于系统设计的原因，授权后需等待15-30分钟才可生效，权限生效后重新创建即可。

3.5 创建 DRS 任务提示用户权限不足或没有操作权限

场景描述

在创建DRS任务时，提示当前用户权限不足或者没有操作权限。



可能原因

当前用户使用的是IAM子账号，子账号操作权限不足导致。

解决方案

步骤1 管理员（主账号）登录IAM控制台。

步骤2 在用户列表中，单击用户右侧的“授权”。

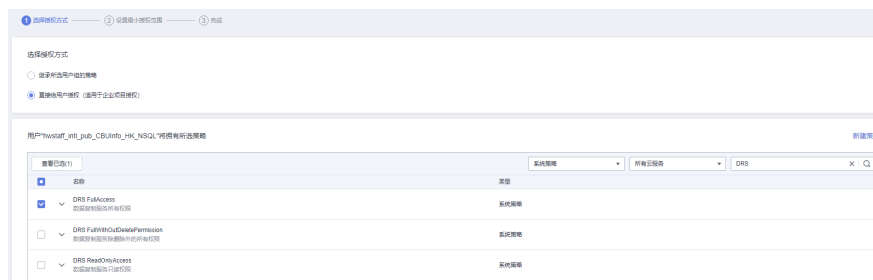
图 3-6 IAM 用户授权



步骤3 在授权页面，选择授权方式和需要的权限。此处以DRS FullAccess权限为例，DRS支持的系统权限可参见：[DRS系统权限](#)。

- 继承所选用用户组的策略：将IAM用户加入拥有DRS FullAccess权限的用户组中，用户将拥有所选用用户组的所有权限。
- 直接给用户授权（适用于企业项目授权）：直接给IAM用户授予DRS FullAccess权限。

图 3-7 直接给用户授权



步骤4 （如授权方式选择“继承所选用用户组的策略”，无需执行此步骤）在“设置最小授权范围”页面，选择授权IAM用户使用的企业项目。

步骤5 单击“确定”，完成IAM用户授权。

授权完成后，管理员可以在“权限管理>授权管理”页面查看、修改该IAM用户的权限。

----结束

3.6 DRS 要求的 PostgreSQL 权限有哪些

DRS在同步过程中，对账号有一定的权限要求，本章节主要介绍PostgreSQL引擎的权限要求。

权限要求

- 源和目标库的连接账号需要具有登录权限，如果没有该账号，可以通过如下方式创建，以用户ptd为例。
参考语句：**CREATE USER ptd PASSWORD 'password';**
- DRS实时同步功能的权限要求，[表1 权限要求](#)中以ptd为例提供参考语句。

表 3-2 权限要求及参考语句

同步模式	源数据库	目标数据库
全量同步	<p>数据库的CONNECT权限，模式的USAGE权限，表的SELECT权限，序列的SELECT权限，系统表pg_catalog.pg_authid的SELECT权限（用于同步用户的密码）。</p> <p>参考语句：</p> <pre>GRANT CONNECT ON DATABASE database_name TO ptd;</pre> <pre>GRANT USAGE ON SCHEMA schema_name TO ptd;</pre> <pre>GRANT SELECT ON TABLE table_name TO ptd;</pre> <pre>GRANT SELECT ON SEQUENCE sequence_name TO ptd;</pre> <pre>GRANT SELECT ON pg_catalog.pg_authid TO ptd;</pre>	<p>库级同步权限要求：</p> <ul style="list-style-type: none"> 如果目标库不是postgres，需要具有CREATEDB权限。 参考语句：ALTER ROLE ptd WITH CREATEDB; 如果目标库是postgres，需要具有postgres库的CONNECT和CREATE权限、对模式public的USAGE和CREATE权限。 参考语句：GRANT CONNECT, CREATE ON DATABASE postgres TO ptd; GRANT USAGE, CREATE ON SCHEMA public TO ptd; <p>表级同步权限要求：</p> <ul style="list-style-type: none"> 如果需要同步库，需要具有CREATEDB权限。 参考语句：ALTER ROLE ptd WITH CREATEDB; 如果需要同步模式，需要具有模式所在库的CONNECT权限、模式所在库上的CREATE权限。 参考语句：GRANT CONNECT, CREATE ON DATABASE database_name TO ptd; 如果需要同步模式下的对象，需要具有模式所在库的CONNECT权限、对象所在模式的USAGE权限、对象所在模式上的CREATE权限。 参考语句：GRANT USAGE, CREATE ON SCHEMA schema_name TO ptd; <p>同步用户权限要求： 需要具有CREATEROLE权限。 参考语句：ALTER USER ptd CREATEROLE;</p>

同步模式	源数据库	目标数据库
<p>全量+增量同步</p>	<ul style="list-style-type: none"> 数据库的CONNECT权限，模式的USAGE权限，表的SELECT权限，序列的SELECT权限，系统表pg_catalog.pg_authid的SELECT权限（用于同步用户的密码），无主键表的UPDATE、DELETE和TRUNCATE权限（用于对无主键表的短暂加锁，保证迁移后数据一致）。 参考语句： GRANT CONNECT ON DATABASE <i>database_name</i> TO <i>ptd</i>; GRANT USAGE ON SCHEMA <i>schema_name</i> TO <i>ptd</i>; GRANT SELECT ON TABLE <i>table_name</i> TO <i>ptd</i>; GRANT SELECT ON SEQUENCE <i>sequence_name</i> TO <i>ptd</i>; GRANT SELECT ON pg_catalog.pg_authid TO <i>ptd</i>; GRANT UPDATE, DELETE, TRUNCATE ON TABLE <i>table_name</i> TO <i>ptd</i>; REPLICATION连接权限。 添加方法： <ol style="list-style-type: none"> 在源数据库的“pg_hba.conf”配置文件的所有配置前增加一行配置“host replication <src_user_name> <drs_instance_ip>/32 <认证方式>”； 认证方式可参考PostgreSQL官方文档pg_hba.conf文件配置，常见的认证方式有md5、scram-sha-256等。 在源库使用SUPERUSER用户执行语句“select pg_reload_conf();”或重 	

同步模式	源数据库	目标数据库
	启数据库实例使设置生效。	

3.7 当源库有无主键表时需要 lock table 权限，对源库有什么影响？

- 在全量阶段，源库会锁无主建表大概10s，对有主键表无影响。
- 在增量阶段，不影响源库。

4 实时迁移

4.1 如何判断数据迁移任务可以停止

您可参考以下方法，确认任务是否可以结束。结束之前您需要确认完成以下几点：

1. 请您确认至少在业务低峰期有过一次完整的数据对比。
2. 完成业务割接。
 - a. 先中断业务（如果业务负载非常轻，也可以尝试不中断业务）。
 - b. 在源数据库端执行如下语句（此处以MySQL为例），并观察在1-5分钟内若无任何新会话执行SQL，则可认为业务已经完全停止。

```
show processlist;
```

📖 说明

上述语句查询到的进程列表中，包括DRS迁移实例的连接，您需要确认除DRS迁移实例的连接外无任何新会话执行SQL，即可认为业务已经完全停止。

- c. 同步时延为0，并稳定保持一段时间；同时，您可以使用数据级对比功能，进行割接前的最后一次数据级对比，耗时可参考之前的对比记录。
 - 如果时间允许，则选择全部对比。
 - 如果时间不允许，则推荐对比活跃表，关键业务表，第二步对比多次存在差异的表等。
 - d. 确定系统割接时机，业务系统指向目标数据库，业务对外恢复使用。
3. 结束迁移任务，该操作仅删除了迁移实例，迁移任务仍显示在任务列表中，您可以进行查看或删除。

4.2 MySQL 迁移中 Definer 强制转化后如何维持原业务用户权限体系

Definer的使用主要应用在视图、存储过程、触发器、事件等对象里，Definer并不会限制对象被调用的权限，但会限制对象访问数据库的权限。本场景下，用户在MySQL迁移过程中选择了“所有Definer迁移到该用户下”，则源库用户体系下其他用户账号在完成用户迁移后，如果用户迁移和权限授权都执行成功，则无需授权便可继续使用原业务（使用DRS用户迁移功能可以实现用户、权限、密码迁移），否则如果想在原来

的用户权限体系下沿用原业务，则需要进行授权后才具有Definer相关数据库对象的访问使用权限，从而保证原业务正常。

本章节主要介绍如何通过数据库命令行对用户账号进行授权的方法。

步骤1 确保新用户（Definer统一使用指定账号）具备足够的权限执行视图、存储过程等相关SQL。

步骤2 通过MySQL官方客户端或者其它工具登录目标数据库。

步骤3 通过如下命令查看需要授权的用户user当前权限详情。

```
show grants for 'user'@'host';
```

步骤4 为了保证原业务不报错，使用如下命令给用户user授予涉及的数据库对象缺失的操作权限。

```
grant select,insert,update,delete on db_name.* to 'user'@'host';
```

一般情况下，访问数据库的权限包括：SELECT、CREATE、DROP、DELETE、INSERT、UPDATE、INDEX、EVENT、CREATE VIEW、CREATE ROUTINE、TRIGGER、EXECUTE。您需要根据具体的数据库对象查看缺少哪些权限，再进行授权操作。

对于存储过程和函数，必须保证用户user对其有拥有EXECUTE权限，授权SQL命令如下：

```
grant execute on db_name.function_name to 'user'@'host';
```

步骤5 使用授权后的用户账号访问目标库对象，无异常报错表示授权成功。需要注意：在java项目工程中调用存储过程、函数如果出现 `Java.sql.SQLException: User does not have access to metadata required to determine stored procedure parameter types. If rights can not be granted, configure connection with "noAccessToProcedureBodies=true" to have driver generate parameters that represent INOUT strings irregardless of actual parametertypes`，则需要单独执行用户user对mysql.proc库的授权：

```
grant select on mysql.proc to 'user'@'host';
```

----结束

4.3 MySQL 存储过程迁移上云后遇到调用权限的问题，如何解决

MySQL存储过程迁移上云后，可能会因为权限问题导致调用存储过程或函数出错。

针对该情况，不同的Definer策略有不同的处理方法。本章节主要以user1为示例，介绍两种迁移Definer的策略下的处理方法。

策略一

在测试连接页面的目标库信息中填写数据库用户名user1，所有Definer迁移到该用户下选“是”。

图 4-1 策略一

目标库信息

数据库实例名称	<input type="text"/>
数据库用户名	<input type="text" value="user1"/>
数据库密码	<input type="password"/>
所有Definer迁移到该用户下	<input checked="" type="radio"/> 是 [?] <input type="radio"/> 否 [?]
<input type="button" value="测试连接"/>	

这种策略下，源库所有存储过程和方法的Definer迁移到目标库后账号都会自动修改为user1，host改为%。若在目标库上出现调用存储过程失败的情况，可执行如下操作：

- 步骤1** 使用uesr1账号登录到目标库RDS for MySQL实例。
- 步骤2** 如果需要使用其他账号调用存储过程，则该账号需要具有execute权限。
- 步骤3** 通过如下语句，使用user1授予其他账号执行存储过程的权限。

其中user表示需要调用存储过程的其他账号：

```
GRANT EXECUTE ON db.* TO user;
```

- 步骤4** 如果需要通过Java调用存储过程，则需要通过如下语句，使用user1授予其他账号查询mysql.proc表的权限。

授权语句可参考如下语句，user表示需要调用存储过程的账号：

```
GRANT SELECT ON mysql.proc TO 'user'@'%';
```

----结束

策略二

在测试连接页面的目标库信息中填写数据库用户名user1，所有Definer迁移到该用户下选“否”。

图 4-2 策略二

目标库信息

数据库实例名称

数据库用户名

数据库密码

所有Definer迁移到该用户下 是 否

这种策略下，源库所有存储过程和方法的Definer迁移到目标库后账号和host保持不变，选择此选项，需要配合[用户权限迁移](#)功能，将源数据库的用户全部迁移，这样才能保持源数据库的权限体系完全不变。

如果您未选择用户权限迁移或者用户权限迁移时存在不支持迁移的账号，建议选择[策略一](#)来处理。

4.4 如何确保业务数据库的全部业务已经停止

业务切换时可通过如下方法确保业务数据库的全部业务已经停止：

- 步骤1** 在源数据库端执行如下语句，查看当前是否还存在有业务连接。
- ```
show processlist;
```

图 4-3 查看是否存在业务连接

```
mysql> show processlist;
+----+-----+-----+-----+-----+-----+-----+-----+
| Id | User | Host | db | Command | Time | State | Info |
+----+-----+-----+-----+-----+-----+-----+-----+
| 3882888 | root | 192.168.0.117:41686 | NULL | Binlog Dump GTID | 58680 | Master has sent all binlog to slave; waiting for more updates | NULL |
| 3886358 | root | 192.168.0.64:55546 | NULL | Query | 0 | starting | show processlist |
+----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

- 步骤2 可选：**如果源数据库有业务连接，则通过结果中Host列的值来查找对应的业务进程并将其停止。

- 步骤3** 在源库执行如下语句，查看binlog位置并记录该值（file列取值:position列取值），此处将该值记为ckpt1。

```
show master status;
```

图 4-4 查看 binlog 位置

```
mysql> show master status;
+-----+-----+-----+-----+-----+
| File | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
+-----+-----+-----+-----+-----+
| mysql-bin.005290 | 197 | | | 67811045-de76-11e9-84c5-fa163e7a0434:1-592564543 |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

- 步骤4** 等待30s以上，在源库执行如下语句，查看binlog位置并记录该值（file列取值:position列取值），此处将该值记为ckpt2。ckpt1=ckpt2时，表示源数据库业务已基本停写。

```
show master status;
```

----结束

## 4.5 迁移日志提示 can not get agency token

### 可能原因

当前创建任务使用的子账号缺少委托，导致执行定时启动任务等自动功能时失败，常见场景如下：

- 创建任务并设置定时启动时，需要使用“账户委托”，否则任务启动失败，迁移日志报：can not get agency token。
- 创建全量+增量任务，任务启动并全量结束后，需要使用“账户委托”，否则任务不会进入增量，迁移日志报：can not get agency token。

### 解决方案

目前针对该情况，分别提供如下解决方案：

- 方法一：使用主账号重新创建任务，启动方式选择“定时启动”。
- 方法二：使用主账号在子账号所在的用户组添加Security Administrator权限后，重新创建任务，启动方式选择“定时启动”。添加权限的具体操作请参见：[创建用户并授权使用DRS](#)。
- 方法三：重新创建任务，启动方式选择“立即启动”。

## 4.6 Oracle 到 MySQL 迁移时，索引超长如何处理

### 索引长度说明

MySQL引擎对索引长度有一些限制，最主要的因素就是存储引擎和字符集。不同的字符集，单个字符包含的最大字节数有所不同。例如UTF8字符集，一个字符最多包含3个字节。而UTF8MB4一个字符最多包含4个字节。

- 如果是单字段索引，则字段长度不应超过[表4-1](#)中的“单字段最大字符数”。例如，MySQL 5.7.6版本的InnoDB引擎，单字段索引不应超过767个字节（字符数=767/最大字节数）。
- 如果是联合索引，则每个字段长度均不能超过[表4-1](#)中的“单字段索引最大字符数”，且所有字段长度合计不应超过“联合索引合计最大字符数”。例如，MySQL 5.7.6版本的InnoDB引擎，每个字段索引不应超过767个字节（字符数=767/最大字节数），且所有字段索引长度总和不超过3072个字节（字符数=3072/最大字节数）。



表 4-1 索引长度说明

| 引擎     | MySQL 版本             | 字符集     | 最大字节数 | 单字段索引最大字符数 | 联合索引合计最大字符数 |
|--------|----------------------|---------|-------|------------|-------------|
| InnoDB | MySQL 5.7.6及<br>以下版本 | UTF8MB4 | 4     | 191        | 768         |
|        | MySQL 5.7.7及<br>以上版本 | UTF8MB4 | 4     | 768        | 768         |

## 索引超长的处理方法

- 方法一  
不迁移含有超长索引的表。
- 方法二  
修改源库索引长度满足以上索引长度说明中的要求，该操作可能导致迁移后数据不完整，请谨慎使用。以目标库为MySQL 5.7.6及以下版本的UTF8MB4为例，可通过如下方式修改长度。

```
alter table tablename modify columnname varchar2 (768) ;
```

其中，tablename请用实际表名代替，columnname请用实际列名代替。

- 方法三  
在源库删除该索引及其约束。以目标库为MySQL 5.7.6及以下版本的UTF8MB4为例，可通过如下方式删除索引及其约束。

```
drop index indexname;
alter table tablename drop constraint constraintname;
```

其中，indexname请用实际索引名代替，tablename请用实际表名代替，constraintname请用实际约束名代替。

## 4.7 Oracle 等异构数据库迁移到 MySQL 后的数据库排序规则为什么是 utf8mb4\_bin

由于不同的数据库支持的字符集有差异，Oracle等数据库迁移到MySQL后，会转换成支持字符相对全面的UTF8MB4字符集。此字符集对应的默认排序字符集为utf8\_general\_ci，是大小写不敏感的，即认为"abc"和"ABC"是相同数据。为了避免Oracle等大小写敏感的数据库到MySQL迁移失败（主键冲突等），或者迁移后对业务有影响（查询结果和Oracle有区别），请参考以下建议。

1. 使用DRS做目标端为MySQL的异构迁移/同步时，DRS自动将数据库（仅涉及迁移的库）字符集排序为utf8mb4\_bin默认字符集，可以后续确保同库内新增表、列与迁移表、列的排序规格一致，从而避免因表、列的排序规则不一致时进行关联查询，执行计划中因索引数据排序不同而无法利用索引的情况。

2. 可在实例级设置参数`collation_server = utf8mb4_bin`，该设定可以确保全实例默认字符集（有别于建议1，建议1是仅迁移的库，这里是全局）一致，均为`utf8mb4_bin`，需要用户根据业务情况决策是否应用于全实例。

## 4.8 RDS for MySQL 不支持 MyISAM 引擎表，迁移时 MyISAM 如何处理

基于以下原因，RDS for MySQL目前不支持MyISAM引擎。

- MyISAM引擎表不支持事务，仅支持表级别锁，导致读写操作相互冲突。
- MyISAM对数据完整性的保护存在缺陷，且这些缺陷会导致数据库数据的损坏甚至丢失。
- MyISAM在出现数据损害情况下，很多都需要手动修复，无法通过产品服务提供的恢复功能进行数据恢复。
- MyISAM向InnoDB的迁移透明，大多数情况不需要改动建表的代码，云数据库自动转换InnoDB即可完成迁移。

DRS在迁移过程中，会自动将MyISAM转换为InnoDB。针对MyISAM引擎表不支持事务这一特点，为了确保MyISAM表的数据一致性，DRS会借助主键来实现最终数据的一致。如果需要迁移没有主键的MyISAM表，建议选择无业务期启动迁移任务，以确保数据的一致性。

## 4.9 低版本迁移至 MySQL 8.0，应该注意哪些问题

MySQL 8.0较MySQL 5.7增加了一些新的特性，并在性能表现上存在差异。迁移前，需要做兼容性分析并给出解决方案。可以从兼容性、系统变量等方面考虑。

- 兼容性分析：  
针对MySQL8.0社区版与MySQL5.7社区版进行分析，包括以下两方面：
  - a. 不影响迁移，但使用方法出现差异。

| 兼容性     | 检查项             | 作用 | 状态 | 解决方案              |
|---------|-----------------|----|----|-------------------|
| 数据类型或函数 | ENCODE()函数      | 加密 | 移除 | AES_ENCRYPT()函数代替 |
|         | DECODE()函数      | 解密 | 移除 | AES_DECRYPT()函数代替 |
|         | ENCRYPT()函数     | 加密 | 移除 | SHA2()函数代替        |
|         | DES_ENCRYPT()函数 | 加密 | 移除 | AES_ENCRYPT()函数代替 |
|         | DES_DECRYPT()函数 | 解密 | 移除 | AES_DECRYPT()函数代替 |

| 兼容性      | 检查项                                                                                                                              | 作用           | 状态 | 解决方案                                                                                                                                                                                                                                                                                       |
|----------|----------------------------------------------------------------------------------------------------------------------------------|--------------|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|          | JSON_APPEND()函数                                                                                                                  | 增加json元素     | 移除 | JSON_ARRAY_APPEND()函数代替                                                                                                                                                                                                                                                                    |
|          | PASSWORD()函数                                                                                                                     | 修改用户密码       | 移除 | ALTER USER user IDENTIFIED BY 'auth_string';                                                                                                                                                                                                                                               |
|          | JSON_MERGE()函数                                                                                                                   | 将多个json合并为一个 | 废弃 | JSON_MERGE_PRESERVE()函数代替                                                                                                                                                                                                                                                                  |
| SQL MODE | NO_AUTO_CREATE_USER、DB2, MAXDB, MSSQL, MYSQL323, MYSQL40, ORACLE, POSTGRESQL, NO_FIELD_OPTIONS, NO_KEY_OPTIONS, NO_TABLE_OPTIONS | -            | 移除 | -                                                                                                                                                                                                                                                                                          |
| 外键约束长度   | 外键约束名称不能超过64个字符                                                                                                                  | -            | -  | SELECT TABLE_SCHEMA, TABLE_NAME FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME IN (SELECT LEFT(SUBSTR(ID,INSTR(ID, '/')+1), INSTR(SUBSTR(ID,INSTR(ID, '/')+1),'_ibfk_')-1) FROM INFORMATION_SCHEMA.INNODB_SYS_FOREIGN WHERE LENGTH(SUBSTR(ID,INSTR(ID, '/')+1))>64);<br>使用ALTER TABLE调整长度 |
| features | GRANT创建用户                                                                                                                        | -            | 移除 | CREATE USER                                                                                                                                                                                                                                                                                |
|          | GRANT修改用户信息                                                                                                                      | -            | 移除 | ALTER USER                                                                                                                                                                                                                                                                                 |

| 兼容性 | 检查项                                  | 作用                        | 状态 | 解决方案                                         |
|-----|--------------------------------------|---------------------------|----|----------------------------------------------|
|     | IDENTIFIED BY PASSWORD 'auth_string' | 设置密码                      | 移除 | IDENTIFIED WITH auth_plugin AS 'auth_string' |
|     | SQL语句中的\n                            | NULL                      | 移除 | NULL代替                                       |
|     | PROCEDURE ANALYSE()语法                | 对MySQL字段值进行统计分析后给出建议的字段类型 | 移除 | -                                            |
|     | 空间函数                                 | -                         | -  | -                                            |
|     | mysql_install_db                     | 初始化                       | 移除 | mysqld --initialize或--initialize-insecure    |

- b. 影响迁移，需要提前做检查。

| 兼容性   | 检查项                                                                                                                                                                               | 作用    | 状态 | 解决方案                                                                                                                                                                                                                                                                 | 原始用法                                 |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------|
| 保留关键字 | cume_dist、dense_rank、empty、except、first_value、grouping、groups、json_table、lag、last_value、lateral、lead、nth_value、ntile、of、over、percent_rank、rank、recursive、row_number、system、window | -     | 新增 | SET sql_mode = 'ANSI_QUOTES'                                                                                                                                                                                                                                         | 名称：数据库、表、索引、列、alias、view、存储过程、分区、表空间 |
| 字符集   | UTF8MB3                                                                                                                                                                           | -     | 废弃 | 使用UTF8MB4代替                                                                                                                                                                                                                                                          | -                                    |
| 分区表   | 不得出现不支持本地分区的存储引擎的分区表                                                                                                                                                              | -     | 移除 | SELECT TABLE_SCHEMA, TABLE_NAME FROM INFORMATION_SCHEMA.TABLES WHERE ENGINE NOT IN ('innodb', 'ndbcluster') AND CREATE_OPTIONS LIKE '%partitioned%';<br>可按照下述两种方式解决：<br>(1) ALTER TABLE table_name ENGINE=INNODB;<br>(2) ALTER TABLE table_name REMOVE PARTITIONING; | 不支持 MyISAM                           |
| 语法    | group by ... asc/desc                                                                                                                                                             | 升序/降序 | 移除 | 使用order by子句代替                                                                                                                                                                                                                                                       | view、function等                       |

| 兼容性  | 检查项                      | 作用                | 状态 | 解决方案                                                                                                                                                                                                                                                                       | 原始用法     |
|------|--------------------------|-------------------|----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| 名称长度 | view的列名称不能超过64个字符        | -                 | -  | alter处理                                                                                                                                                                                                                                                                    | 最多255个字符 |
|      | enum或set元素的总长度不能超过255个字符 | -                 | -  | 用户处理                                                                                                                                                                                                                                                                       | 最大64K    |
| 大小写  | lower_case_table_names   | MySQL 设置字母大小写是否敏感 | -  | 升级过程中，如果设置该参数为1，则必须确保schema和table名称必须是小写的<br><br>SELECT TABLE_NAME FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME != LOWER(TABLE_NAME) AND TABLE_TYPE = 'BASE TABLE';<br><br>SELECT SCHEMA_NAME FROM INFORMATION_SCHEMA.SCHEMATA WHERE SCHEMA_NAME != LOWER(SCHEMA_NAME); | -        |
| 触发器  | 是否有空定义或者无效的创建上下文         | -                 | -  | show triggers查看，检测 character_set_client、collation_connection、Database Collation 属性                                                                                                                                                                                         | -        |

- 系统变量默认值变更

针对社区版MySQL5.7与8.0版本的默认值作对比，默认值不影响迁移，但对迁移后的业务会产生影响。

| 序号     | parameter /option    | community |         | 作用 | 备注     |
|--------|----------------------|-----------|---------|----|--------|
|        |                      | 原默认值      | 新默认值    |    |        |
| Server |                      |           |         |    |        |
| 1      | character_set_server | latin1    | utf8mb4 | -  | 和源保持一致 |

|        |                                  |                                                                    |                                                       |                                      |        |
|--------|----------------------------------|--------------------------------------------------------------------|-------------------------------------------------------|--------------------------------------|--------|
| 2      | collation_server                 | latin1_swedish_ci                                                  | utf8mb4_0900_ai_ci                                    | -                                    | 和源保持一致 |
| 3      | explicit_defaults_for_timestamp  | OFF                                                                | ON                                                    | 更新某一行时是否更新timestamp列                 | 和源保持一致 |
| 4      | optimizer_trace_max_mem_size     | 16KB                                                               | 1MB                                                   | -                                    | 和源保持一致 |
| 5      | validate_password_check_username | OFF                                                                | ON                                                    | -                                    | 和源保持一致 |
| 6      | back_log                         | -1 (autosize) changed from : back_log = 50 + (max_connections / 5) | -1 (autosize) changed to : back_log = max_connections | 在MySQL暂时停止回答新请求之前的短时间内多少个请求可以被存在堆栈中。 | 和源保持一致 |
| 7      | max_allowed_packet               | 4194304 (4MB)                                                      | 67108864 (64MB)                                       | 限制Server接受的数据包大小                     | 按默认值   |
| 8      | max_error_count                  | 64                                                                 | 1024                                                  | 控制显示告警的个数                            | 和源保持一致 |
| 9      | event_scheduler                  | OFF                                                                | ON                                                    | -                                    | 和源保持一致 |
| 10     | table_open_cache                 | 2000                                                               | 4000                                                  | -                                    | 和源保持一致 |
| 11     | log_error_verbosity              | 3 (Notes)                                                          | 2 (Warning)                                           | -                                    | 按默认值   |
| INNODB |                                  |                                                                    |                                                       |                                      |        |
| 1      | innodb_undo_tablespace           | 0                                                                  | 2                                                     | -                                    | 按默认值   |
| 2      | innodb_undo_log_truncate         | OFF                                                                | ON                                                    | -                                    | 按默认值   |

|                    |                                |                 |                                    |                                        |                  |
|--------------------|--------------------------------|-----------------|------------------------------------|----------------------------------------|------------------|
| 3                  | innodb_flush_method            | NULL            | fsync (Unix), unbuffered (Windows) | 控制 innodb 数据文件及 redo log 的打开、刷写模式      | 按SQL默认值 O_DIRECT |
| 4                  | innodb_autoinc_lock_mode       | 1 (consecutive) | 2 (interleaved)                    | 控制着在向有 auto_increment 列的表插入数据时，相关锁的行为； | 和源保持一致           |
| 5                  | innodb_flush_neighbors         | 1 (enable)      | 0 (disable)                        | 从缓冲池刷新页面是否也刷新相同范围内的其他脏页。               | 和源保持一致           |
| 6                  | innodb_max_dirty_pages_pct_lwm | 0 (%)           | 10 (%)                             | 影响 innodb 刷新脏页行为                       | 按默认值             |
| 7                  | innodb_max_dirty_pages_pct     | 75 (%)          | 90 (%)                             | 影响 innodb 刷新脏页行为                       | 按默认值             |
| PERFORMANCE_SCHEMA | 整体是不是开的                        | -               | -                                  | -                                      | 和源保持一致           |
| REPLICATION        |                                |                 |                                    |                                        |                  |
| 1                  | log_bin                        | OFF             | ON                                 | -                                      | 默认打开             |
| 2                  | server_id                      | 0               | 1                                  | -                                      | 如果是0, 则设为1       |
| 3                  | log-slave-updates              | OFF             | ON                                 | -                                      | 默认打开             |
| 4                  | expire_log_days                | 0               | 30                                 | -                                      | 按默认值1            |
| 5                  | master-info-repository         | FILE            | TABLE                              | -                                      | 默认TABLE          |



|   |                                  |                           |                          |   |         |
|---|----------------------------------|---------------------------|--------------------------|---|---------|
| 6 | relay-log-info-repository        | FILE                      | TABLE                    | - | 默认TABLE |
| 7 | transaction-write-set-extraction | OFF                       | XXHASH64                 | - | 按默认值    |
| 8 | slave_rows_search_algorithms     | INDEX_SCAN,<br>TABLE_SCAN | INDEX_SCAN,<br>HASH_SCAN | - | 按默认值    |

- 移除系统变量

针对社区版MySQL 5.7与8.0进行分析，移除系统变量不影响迁移。

| 移除变量                                  |
|---------------------------------------|
| innodb_locks_unsafe_for_binlog        |
| log_builitn_as_identified_by_password |
| old_passwords                         |
| query_cache_limit                     |
| query_cache_min_res_unit              |
| query_cache_size                      |
| query_cache_type                      |
| query_cache_wlock_invalidate          |
| ndb_cache_check_time                  |
| ignore_db_dirs                        |
| tx_isolation                          |
| tx_read_only                          |
| sync_frm                              |
| secure_auth                           |
| multi_range_count                     |
| log_error_verbosity                   |
| sql_log_bin                           |
| metadata_locks_cache_size             |
| metadata_locks_hash_instances         |
| date_format                           |

| 移除变量                             |
|----------------------------------|
| datetime_format                  |
| time_format                      |
| max_tmp_tables                   |
| ignore_builtin_innodb            |
| innodb_support_xa                |
| innodb_undo_logs                 |
| innodb_undo_tablespace           |
| internal_tmp_disk_storage_engine |

## 4.10 MongoDB 数据库迁移过程中，源数据库出现内存溢出（OOM）是什么原因

### 场景描述

在进行MongoDB数据库迁移的过程中，出现源数据库内存溢出（OOM），导致源数据库不可用，迁移失败。

### 问题分析

出现上述内存溢出可能存在如下原因：

- 源数据库的mongod服务单独部署在一台机器上，如果这种情况下在迁移过程中出现内存溢出，一般就是因为是在迁移过程中源库在执行会大量消耗内存的操作，比如：创建索引，排序查询等。
- 源数据库的mongod服务和其他服务同时部署在一台机器上，而且没有设置cacheSizeGB的大小，这种情况下，如果因为其他服务消耗掉内存导致不能给wiredTiger引擎保证的内存，则会出现内存溢出的情况。

#### 说明

一般默认情况下，mongod的wiredTiger引擎可以使用整个机器内存减一的50%（3.2的版本）或者60%（3.4以后的版本）。

### 解决方案

- 如果mongod服务是单独部署在一台机器上，则在迁移过程中最好不要执行会大量消耗内存的操作，比如：创建索引，排序查询等。
- 如果mongod服务和其他服务共同部署在一台机器上，则建议给mongod的wiredTiger引擎加上cacheSizeGB的参数，设置的值为机器最小空闲内存的一半，保证所有服务在高峰期所使用的内存不会超过分配给wiredTiger引擎的内存。

## 4.11 如何关闭集合均衡器 Balancer

使用DRS服务进行MongoDB数据库分片集群到分片集群的迁移，必须关闭要迁移集合的均衡器Balancer。

### 📖 说明

- DDS集群关闭集合均衡器Balancer需调用API接口，参考[设置集群均衡开关](#)，具体方法可咨询DDS技术支持。自建MongoDB设置可参考[如下步骤](#)。
- 迁移结束后请开启Balancer，因为在迁移期间关闭了Balancer，源数据库的不同shard可能产生了不等量的块（chunk），在Balancer开启之后集群shard之间的块（chunk）移动会暂时影响源数据库的性能。

### 关闭 Balancer 的步骤

**步骤1** 通过Mongo Shell 登录数据库。

**步骤2** 在mongos节点命令窗口中，使用如下命令，切换至config数据库。

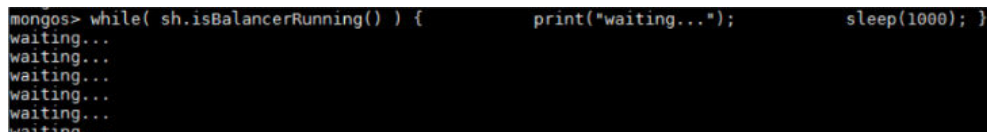
**use config**

**步骤3** 执行如下命令，判断是否可以关闭Balancer。

```
while(sh.isBalancerRunning()) {
 print("waiting...");
 sleep(1000);
}
```

- 如果返回结果是waiting，则表示当前Balancer正在执行块（chunk）迁移，此时不能执行关闭Balancer的命令，否则可能引起数据不一致。

图 4-5 查看输出结果



```
mongos> while(sh.isBalancerRunning()) { print("waiting..."); sleep(1000); }
waiting...
waiting...
waiting...
waiting...
waiting...
waiting...
```

- 如果返回结果是空，则表示当前Balancer没有在进行块（chunk）迁移，此时可以执行下一步的关闭Balancer的命令。

**步骤4** 关闭Balancer。

- 如果是整个实例的迁移，则执行如下命令，可以关闭整个实例的Balancer。  
`sh.stopBalancer()`
- 如果要关闭待迁移且已经开启了分片的集合的Balancer，则执行如下命令：  
`sh.disableBalancing("database.collection")`

其中database.collection表示要关闭的集合的namespace。

----结束

## 4.12 如何批量导出、导入事件（event）和触发器（trigger）

在进行MySQL到MySQL的迁移时，若任务结束后发现迁移日志中提示迁移事件和触发器失败，可手动迁移。

本小节主要介绍批量导出导入事件和触发器的具体操作。

#### 步骤1 从源库批量导出触发器。

1. 在源库执行以下语句，获取TRIGGER\_SCHEMA和TRIGGER\_NAME。  

```
SELECT TRIGGER_SCHEMA,TRIGGER_NAME FROM INFORMATION_SCHEMA.TRIGGERS
WHERE TRIGGER_SCHEMA in ('DB1','DB2','DB3') order by TRIGGER_NAME;
```

上述语句中，DB1，DB2，DB3分别表示从源库待迁移到目标库的数据库。
2. 在源库执行如下语句，从字段SQL Original Statement中获取源库创建触发器的语句。  

```
SHOW CREATE TRIGGER TRIGGER_SCHEMA.TRIGGER_NAME \G;
```

上述语句中，TRIGGER\_SCHEMA.TRIGGER\_NAME填写的为[步骤1.1](#)中查询到的TRIGGER\_SCHEMA和TRIGGER\_NAME具体值。

#### 步骤2 从源库批量导出事件。

1. 在源库执行以下语句，获取EVENT\_SCHEMA和EVENT\_NAME。  

```
SELECT EVENT_SCHEMA,EVENT_NAME FROM INFORMATION_SCHEMA.EVENTS WHERE
EVENT_SCHEMA in ('DB1','DB2','DB3') order by EVENT_NAME;
```

上述语句中，DB1，DB2，DB3分别表示从源库待迁移到目标库的数据库。
2. 在源库执行如下语句，从字段SQL Original Statement中获取源库创建事件的语句。  

```
SHOW CREATE EVENT EVENT_SCHEMA.EVENT_NAME \G;
```

上述语句中，EVENT\_SCHEMA.EVENT\_NAME填写的为[步骤2.1](#)中查询到的EVENT\_SCHEMA和EVENT\_NAME具体值。

#### 步骤3 导入触发器和事件。

在目标库重新执行从源库导出的创建触发器和创建事件语句。

----结束

## 4.13 源库参数 lower\_case\_table\_names=1 时，为什么不允许迁移包含大写字母的库或者表

### 场景描述

当源库参数lower\_case\_table\_names=1时，无法迁移包含大写字母的库或者表。

### 问题分析

当源库的lower\_case\_table\_names 参数值为1时，MySQL会将库名或者表名转换成小写再进行查找。若存在以大写字母形式创建的库或者表，那么在lower\_case\_table\_names参数值为1的情况下，MySQL将无法找到这个库或表，报告查询失败。也就是说，若lower\_case\_table\_names的参数值为1时，大写字母的库或表很可能是不可访问的。

### 解决方案

目前针对该情况，分别提供如下解决方案：

## 方法一

修改源库lower\_case\_table\_names的参数值为0 (即大小写敏感), 并且保证源库与目标库的该参数值一致。

## 方法二

若无法永久修改lower\_case\_table\_names, 可临时将源库lower\_case\_table\_names修改为0, 然后执行如下操作。

- 对于表, 可以使用如下语句将表名转换为小写:  

```
alter table `BigTab` rename to `bigtab`
```
- 对于库, 则需要导出后, 修改库名为小写, 再进行导入。

### 注意

修改库名或表名之后, 需要维护权限的一致性, 以免影响应用访问。

## 方法三

对象选择时不迁移该库或者该表。

## 4.14 分片集群 MongoDB 迁移前清除孤儿文档

### 什么是孤儿文档

MongoDB负载均衡器 (Balancer) 会根据集合的分片键(Shard key)均衡数据。Balancer的工作原理是: 需要Balancer的数据块 (Chunk) 先复制到目标Shard, 成功后再删除原Shard上的Chunk, 来完成一次Chunk迁移, 通过多次Chunk迁移来实现均衡。在Chunk迁移时, 如果发生网络闪断等不可预知的场景, 完成了复制但没有完成删除, 那么对同一条文档会同时存在于两个Shard上。因为Chunk迁移在MongoDB上是感知的, config会更新这条文档应该在哪个Shard上, 那么另一个Shard上的文档会存在但不会被感知, 后续的update、delete操作都不会作用于这个错误的Shard上的文档, 那么这条文档被称为孤儿文档 (Orphaned Document)。

### 迁移影响

DRS在迁移集群时, 会从Shard上抽取全量数据。正常文档和孤儿文档在不同的Shard上, DRS不会感知, 都会迁移到目标库。DRS针对MongoDB迁移的冲突策略为忽略, 因此最终目标库上的文档取决于哪个文档先被迁过去, 会造成数据内容或行数不一致。

### 操作步骤

- 步骤1** 下载用于清除孤儿文档的**cleanupOrphaned.js**脚本文件。
- 步骤2** 修改cleanupOrphaned.js脚本文件, 将test替换为待清理孤儿文档的数据库名。
- 步骤3** 执行以下命令, 清理Shard节点下指定的数据库中所有集合的孤儿文档。

```
mongo --host ShardIP --port Primaryport --authenticationDatabase database -u username -p password cleanupOrphaned.js
```

### 📖 说明

- ShardIP: Shard节点的IP地址。
- Primaryport: Shard节点中的Primary节点的服务端口。
- database: 鉴权数据库名, 即数据库账号所属的数据库。
- username: 登录数据库的账号。
- password: 登录数据库的密码。

### 📖 说明

如果您有多个数据库, 您需要重复执行步骤**步骤2**和步骤**步骤3**, 分别为每个数据库的每个Shard节点清理孤立文档。

----结束

## 4.15 MySQL 账号迁移后权限显示多了反斜杠 “\”

其他云MySQL使用DRS进行入云迁移后, 迁移后的账号在RDS的控制台上显示带有反斜杠 “\”。

迁移前在其他控制台查看, 显示如下:

| 账号 | 类型   | 状态    | 所属数据库           | 账号描述 |
|----|------|-------|-----------------|------|
| aa | 普通账号 | ✓ 已激活 | a_a读写 (DDL+DML) | --   |

迁移后在RDS控制台查看, 显示如下:

创建帐号

| 帐号 | 状态   | 已授权数据库   |
|----|------|----------|
| aa | ✓ 激活 | a\_a(读写) |

是因为MySQL GRANT语法支持通配符, 详细可参考[MySQL官网文档](#)。DRS迁移逻辑是按照SQL查询结果进行权限迁移, 实际目标端与源端账号权限是一致的, 只是其他云控制台单独对 “\” 做了处理, 在界面上不显示转义字符 “\” 而已。

# 5 备份迁移

## 5.1 备份迁移场景的是否最后一个备份文件选择错误该如何处理

备份迁移过程中，根据选择“最后一个备份文件”来判断是否为最后一次备份，对于人工操作中不可控的误选择，有以下两种情况及处理方法：

- 选择“是”，但期望为否，即仍然希望继续做增量备份迁移。但由于SQL Server本身的设计，数据库一旦收到还原已完成的信号，便会做一系列的内部工作并把数据库置为可用，已无法继续增量备份迁移。此时，只能删除备份数据库重新进行全量+增量的备份还原。
- 选择“否”，但期望为是，即不希望继续恢复增量备份迁移。其实SQL Server没有严格意义上的最后一个备份文件，此时可以再做一个增量备份（即使没有数据改变也可以备份），在该次增量备份时选择“是”即可完成迁移，相关数据库将会变为可用。

## 5.2 手动配置信息

### 操作场景

目前从本地或虚拟机通过DRS备份迁移功能直接迁移到本云RDS for SQL Server实例上，在迁移完成后还需要针对**Login账号**，**DBLink**，**AgentJOB**，**关键配置**进行识别，并手动完成相关同步工作。

### Login 账号

Login账号即SQL Server的实例级账号，主要用于用户管理用户服务器权限与数据库权限。一个用户通常会有多个该类型账号，用户迁移到RDS for SQL Server实例后，需要手动将自己本地的Login账号同步在实例上进行创建，以下方法将介绍如何在本云RDS for SQL Server实例上创建同名，同密码的Login账号，并进行授权操作。

- 步骤1** 通过以下脚本获取本地实例Login账号创建脚本，获取到的脚本可以直接在目标端上执行，以创建同名，同密码的Login账号。

```
SELECT 'IF (SUSER_ID('+QUOTENAME(SP.name,'')+') IS NULL) BEGIN CREATE LOGIN '
+QUOTENAME(SP.name)+
```

```
CASE
WHEN SP.type_desc = 'SQL_LOGIN' THEN ' WITH PASSWORD = '
+CONVERT(NVARCHAR(MAX),SL.password_hash,1)+ ' HASHED,SID='
+CONVERT(NVARCHAR(MAX),SP.SID,1)+' ,CHECK_EXPIRATION = '
+ CASE WHEN SL.is_expiration_checked = 1 THEN 'ON' ELSE 'OFF' END +', CHECK_POLICY = '
+CASE WHEN SL.is_policy_checked = 1 THEN 'ON,' ELSE 'OFF,' END
ELSE ' FROM WINDOWS WITH'
END
+' DEFAULT_DATABASE=[' +SP.default_database_name+ '], DEFAULT_LANGUAGE=['
+SP.default_language_name+ '] END;' as CreateLogin
FROM sys.server_principals AS SP LEFT JOIN sys.sql_logins AS SL
ON SP.principal_id = SL.principal_id
WHERE SP.type = 'S'
AND SP.name NOT LIKE '###'
AND SP.name NOT LIKE 'NT AUTHORITY%'
AND SP.name NOT LIKE 'NT SERVICE%'
AND SP.name NOT IN ('rdsadmin','rdsbackup','rdsuser','rdsmirror','public')
```

**步骤2** 执行**步骤1**脚本可获取如下执行脚本。

图 5-1 获取执行脚本



```
CreateLogin
1 IF (SUSER_ID('sa') IS NULL) BEGIN CREATE LOGIN [sa] WITH PASSWORD = 0x010039EF2EFAD6A3DB4E2AEB941E8ED32E5189A4BE757...
2 IF (SUSER_ID('rdsuser2') IS NULL) BEGIN CREATE LOGIN [rdsuser2] WITH PASSWORD = 0x0100E8BCBC25FC67008D4EE75AD660D1...
3 IF (SUSER_ID('csidbo') IS NULL) BEGIN CREATE LOGIN [csidbo] WITH PASSWORD = 0x0100A508789C15CE688664E162A5EDF4F4D2E...
4 IF (SUSER_ID('TestLogin7') IS NULL) BEGIN CREATE LOGIN [TestLogin7] WITH PASSWORD = 0x010073DA9A79E6677E8AF7077EF67...
5 IF (SUSER_ID('rdsuser3') IS NULL) BEGIN CREATE LOGIN [rdsuser3] WITH PASSWORD = 0x01009448FEDBC8D5B5B2529384029CA0...
6 IF (SUSER_ID('Test2') IS NULL) BEGIN CREATE LOGIN [Test2] WITH PASSWORD = 0x0100130953CEEAEC997D08B6EAF65F84EBCAA44...
7 IF (SUSER_ID('Test3') IS NULL) BEGIN CREATE LOGIN [Test3] WITH PASSWORD = 0x0100EE98873948E02595EDCD953B426637281B7...
8 IF (SUSER_ID('Test4') IS NULL) BEGIN CREATE LOGIN [Test4] WITH PASSWORD = 0x01000EE91B9EF087741F16A44E70AA813D08A8B...
9 IF (SUSER_ID('Test5') IS NULL) BEGIN CREATE LOGIN [Test5] WITH PASSWORD = 0x010056EEF845DF098D2DF9395AF7B7618A20735...
```

**步骤3** 复制**步骤2**中的执行脚本在目标端直接执行，创建出来的Login账号跟原实例密码一致。

**步骤4** 将新建的Login账号跟用户当前RDS SQL Server实例上的迁移过来的数据库用户权限进行映射（mapping），以保证该账号在当前实例上的权限一致性，执行脚本如下。

```
declare @DBName nvarchar(200)
declare @Login_name nvarchar(200)
declare @SQL nvarchar(MAX)
set @Login_name = 'TestLogin7' //输入Login名称逐个执行
declare DBName_Cursor cursor for
select quotename(name)from sys.databases where database_id > 4 and state = 0
and name not like '%$%'
and name <> 'rdsadmin'
open DBName_Cursor
fetch next from DBName_Cursor into @DBName
WHILE @@FETCH_STATUS= 0
begin
SET @SQL=' USE '+ (@DBName)+ '
if exists(select top 1 1 from sys.sysusers where name = ''+ @Login_name + '')
begin
ALTER USER '+@Login_name+' with login = '+@Login_name+';
end
'
print @SQL
EXEC (@SQL)
fetch next from DBName_Cursor into @DBName
end
close DBName_Cursor
deallocate DBName_Cursor
```



### 说明

以上脚本执行完成后，用户即可在自己的新实例上看到同名的登录账号，并且密码跟权限是完全跟本地一致的。

----结束

## DBLink 连接

DBLink连接指SQL Server支持用户通过创建DBLink连接的方式，跟外部实例上的数据库进行交互，这种方式可以极大的方便用户不同实例间，不同数据库类型之间的数据库查询，同步，比较，所以大部分用户都会在本地产例上用到该服务，但是迁移上云后，本地DBLink是不会自动同步到云上实例的，还需要简单的手动进行同步。

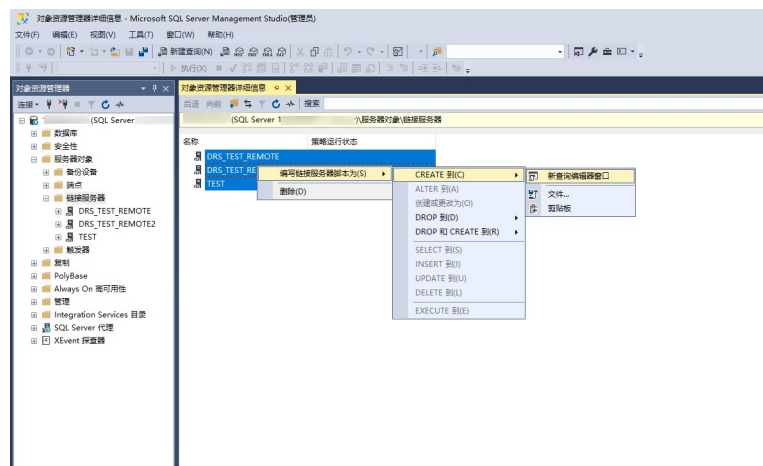
- 步骤1** 通过微软提供的官方Microsoft SQL Server Management Studio客户端工具连接本地实例与云上实例，同时在“服务器对象 > 链接服务器”下找到当前实例的DBLink链接。

图 5-2 查看 DBLink 链接



- 步骤2** 选中链接服务器，然后按F7，会自动弹出对象资源管理信息页，在该页面中可以方便您快速的自动创建脚本。

图 5-3 自动创建脚本



**步骤3** 在新窗口中，可以看到当前实例上所有DBLink的创建脚本，仅需复制该脚本到目标实例上，并修改@rmtpassword上的密码即可执行创建操作。

```
USE [master]
GO

/***** Object: LinkedServer [DRS_TEST_REMOTE] Script Date: 2019/5/25 17:51:50 *****/
EXEC master.dbo.sp_addlinkedserver @server = N'DRS_TEST_REMOTE', @srvproduct=N'',
@provider=N'SQLNCLI', @datasrc=N'DESKTOP-B18JH5T\SQLSERVER2016EE'
/* For security reasons the linked server remote logins password is changed with ##### */
EXEC master.dbo.sp_addlinkedsrvlogin
@rmtsrvname=N'DRS_TEST_REMOTE',@useself=N'False',@locallogin=NULL,@rmtuser=N'sa',@r
mtpassword='#####'
GO
```

**说明**

以上脚本为范例，创建的脚本可能包含大量系统默认配置项，但是每个DBLink仅需保留以下两个关键脚本即可执行成功，同时需要注意重新输入账号连接密码。

----结束

## Agent JOB

Agent JOB又名SQL Server代理服务，可以方便用户快速的在实例上创建定时任务，帮助用户进行日常运维和数据处理工作，用户在本地的JOB需要手动进行脚本迁移。

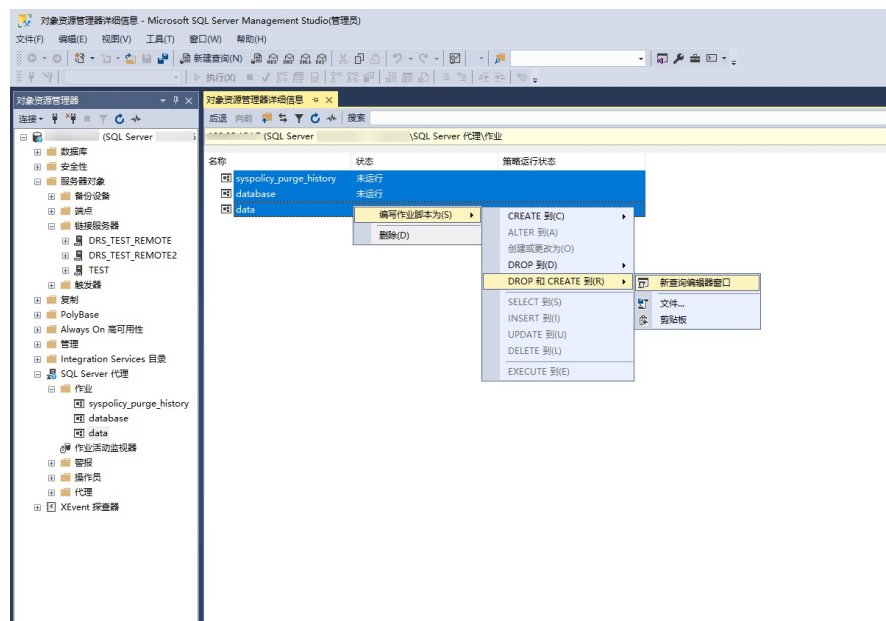
**步骤1** 通过微软提供的官方Microsoft SQL Server Management Studio客户端工具连接本地实例与云上实例，同时在“SQL Server代理 > 作业”下找到当前实例上的所有JOB任务。

图 5-4 查看作业



**步骤2** 选择SQL Server代理下的作业，然后按F7，可以在对象资源管理器中看到所有的作业（JOB），全部选中后创建脚本到新窗口。

图 5-5 创建脚本



**步骤3** 复制新窗口中的T-SQL创建脚本到新实例上，然后注意修改如下几个关键项，以保障您的创建成功。

- 注意修改每个JOB上的Owner账号：  
例如：  
@owner\_login\_name=N'rdsuser'
- 注意修改每个JOB上的实例名称：  
例如：  
@server=N'实例IP'  
@server\_name = N'实例IP'

#### 说明

新建JOB的Owner账号十分重要，在RDS SQL Server上，仅有该JOB的Owner可以看到实例上自己的JOB，别的Login账号是看不到无法操作的，所以建议所有的JOB Owner尽量是同一个账号方便管理。

---结束

## 关键配置项

用户将数据库还原到RDS for SQL Server实例上之后，本地的一些重要配置项也需要进行同步确认，避免影响业务的正常使用。

1. tempdb：临时数据库的文件配置需要进行同步。

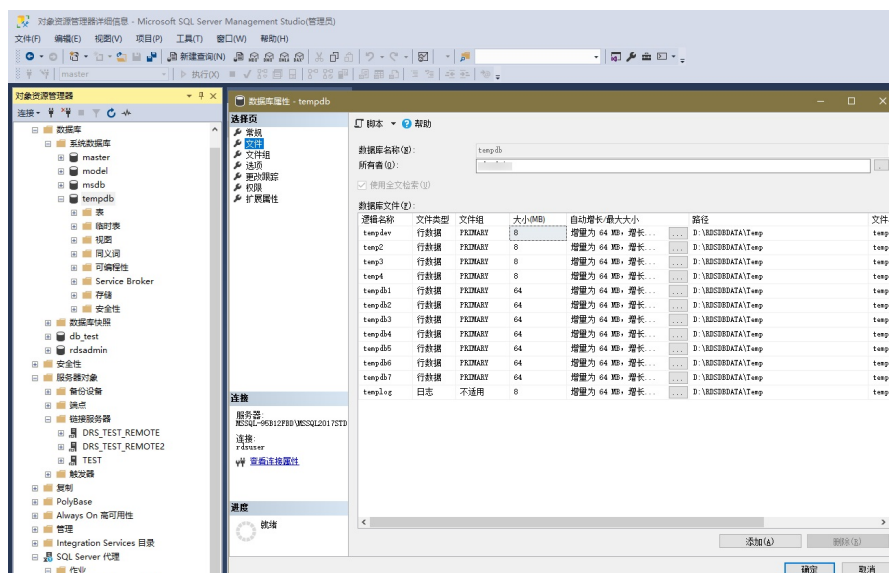
推荐配置为8个临时文件，注意路径一定要确保在D:\RDSDBDATA\Temp\

通过在目标数据库端执行如下脚本添加临时数据库的文件配置：

```
USE [master]
GO
ALTER DATABASE [tempdb] ADD FILE (NAME = N'tempdb1', FILENAME =
N'D:\RDSDBDATA\Temp\tempdb1.ndf', SIZE = 65536KB, FILEGROWTH = 65536KB)
GO
ALTER DATABASE [tempdb] ADD FILE (NAME = N'tempdb2', FILENAME =
```

```
N'D:\RDSDBDATA\Temp\tempdb2.ndf' , SIZE = 65536KB , FILEGROWTH = 65536KB)
GO
ALTER DATABASE [tempdb] ADD FILE (NAME = N'tempdb3', FILENAME =
N'D:\RDSDBDATA\Temp\tempdb3.ndf' , SIZE = 65536KB , FILEGROWTH = 65536KB)
GO
ALTER DATABASE [tempdb] ADD FILE (NAME = N'tempdb4', FILENAME =
N'D:\RDSDBDATA\Temp\tempdb4.ndf' , SIZE = 65536KB , FILEGROWTH = 65536KB)
GO
ALTER DATABASE [tempdb] ADD FILE (NAME = N'tempdb5', FILENAME =
N'D:\RDSDBDATA\Temp\tempdb5.ndf' , SIZE = 65536KB , FILEGROWTH = 65536KB)
GO
ALTER DATABASE [tempdb] ADD FILE (NAME = N'tempdb6', FILENAME =
N'D:\RDSDBDATA\Temp\tempdb6.ndf' , SIZE = 65536KB , FILEGROWTH = 65536KB)
GO
ALTER DATABASE [tempdb] ADD FILE (NAME = N'tempdb7', FILENAME =
N'D:\RDSDBDATA\Temp\tempdb7.ndf' , SIZE = 65536KB , FILEGROWTH = 65536KB)
GO
```

图 5-6 检查临时文件



2. 数据库隔离级别：请确认原实例上数据库的隔离级别是否开启，并同步到RDS SQL Server实例，快照隔离参数有2个，分别是：

- 读提交快照 ( Is Read Committed Snapshot On )
- 允许快照隔离 ( Allow Snapshot Isolation )

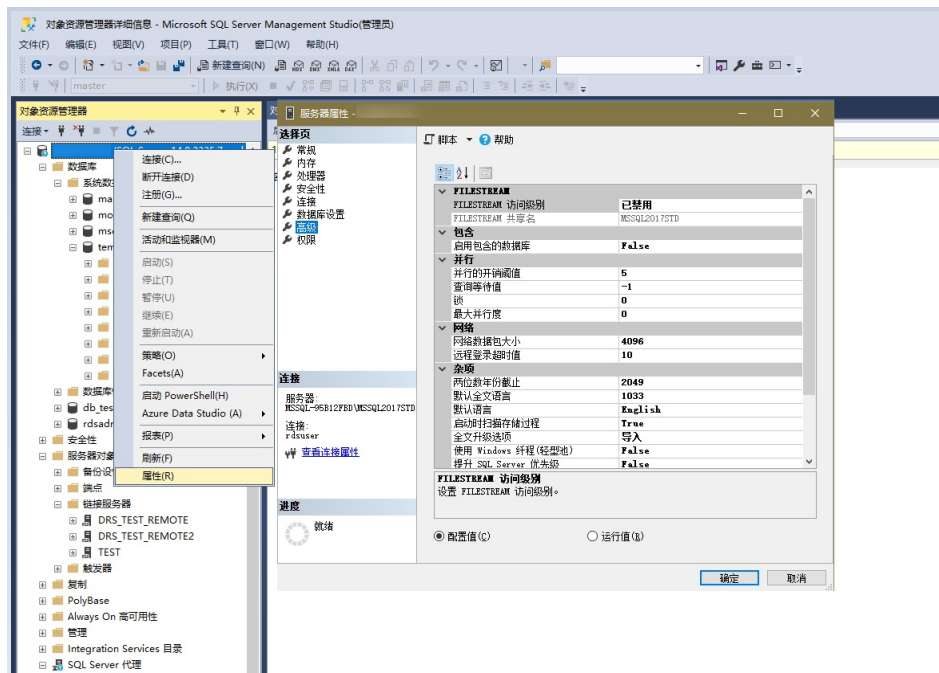
若原实例上数据库的隔离级别是开启的，您可以通过在目标数据库端执行如下脚本开启数据库的隔离级别：

```
USE [DBName]
GO
ALTER DATABASE [DBName] SET READ_COMMITTED_SNAPSHOT ON WITH NO_WAIT
GO
ALTER DATABASE [DBName] SET ALLOW_SNAPSHOT_ISOLATION ON
GO
```

3. 实例最大并行度：实例最大并行度在RDS for SQL Server实例上默认设置为0，用户也可以根据自己本地原来的设置项进行同步设置，避免不同环境下业务场景出现异常。

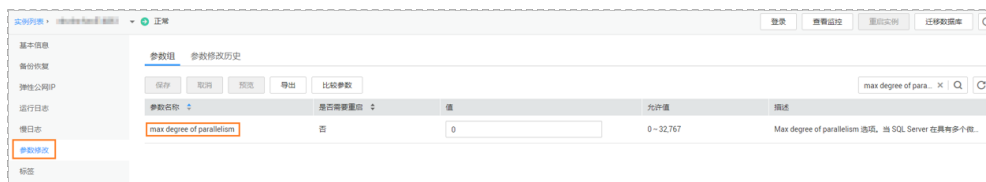
右击本地实例选择属性，在服务器属性弹出框中选择高级，然后在右侧找到最大并行度 ( max degree of parallelism ) 设置项，确认本地实例设置值，并同步在目标RDS for SQL Server实例管理的参数组中进行修改。

图 5-7 查看本地实例最大并行度值



登录本云实例控制台，在实例管理页，单击目标实例名称，进入基本信息页签，切换至“参数修改”，搜索最大并行度（max degree of parallelism）并进行修改。

图 5-8 修改目标 RDS for SQL Server 实例的最大并行度



4. 迁移上云的数据库恢复模式是否为完整（FULL）模式，如果不是需要进行修改。右击数据库选择属性，在弹出数据库属性框中选择选项，并在右侧确认该数据库恢复模式为完整（FULL），保证该数据库高可用和备份策略可执行。

图 5-9 检查数据库恢复模式



## 5.3 DRS 备份迁移支持跨区域备份吗？

DRS备份迁移有两种模式：自建OBS桶，RDS全量备份。

1. 自建OBS桶：通过上传其他区域RDS的全量备份的数据，可以实现跨区域备份。
2. RDS全量备份：获取的当前区域的RDS全量备份，此种模式不支持跨区域备份。

# 6 实时同步

## 6.1 DRS 支持直接同步不同 schema 的表到同一个 schema 吗

### 使用场景

不同的schema需要合并到一个schema。

### 使用限制

不同的schema中，不能包含相同的表名。

### 解决方案

DRS支持直接同步不同schema的表到同一个schema。

## 6.2 DRS 实时同步支持使用 Online DDL 工具吗

### 使用场景

MySQL为源的迁移或同步任务，在增量阶段使用第三方拷表方式实现的Online DDL工具（如PT-OSC、GH-OST）在源库执行DDL，会先创建临时表，通过临时表做DDL，针对这个情况，使用DRS进行迁移或同步时，可分为以下三种场景：

- 对于库级和实例级迁移、同步任务，由于Online DDL使用的临时表在同步列表中，DRS会自动同步DDL操作，因此不需要特殊处理。
- 对于表级迁移或同步任务，如果创建DRS任务时，迁移或同步对象列表中已经提前添加了第三方Online DDL涉及的临时表的场景，DRS也会自动同步DDL操作，不需要特殊处理。
- 对于表级迁移或同步任务，如果创建DRS任务时，只选中表的数据，由于Online DDL使用的临时表不在选中的列表中，因此不会被DRS同步。可以参考以下[使用限制](#)和[操作步骤](#)，通过手动在目标库执行DDL的方式，避免源库Online DDL操作导致的源库和目标库表结构不一致，引起的DRS任务失败。



## 使用限制

- 本方案为不能使用DRS库级或实例级迁移、同步的场景下的替代方案，建议优先选择库级和实例级，尽量避免使用本方案。
- 不同DDL语句，在源库和目标库的操作顺序不同，请严格参照[操作步骤](#)中的顺序执行，避免因执行顺序错误导致DRS任务失败。
- 在源库和目标库执行的DDL语句要语义一致，包括但不限于对象名、列类型、长度等要相同或兼容。

## 操作步骤

**步骤1** 查看DRS任务状态，确认任务状态为增量中且增量时延在10秒以内。

**步骤2** 确认需要执行的DDL操作，不同操作在源库和目标库执行的顺序不同。

- 加列：先在目标库执行，执行成功后在源库执行。
- 删列：先在源库执行，执行成功后在目标库执行。
- 增加、修改、删除列默认值：跟操作顺序无关。
- 修改列类型：先在目标库执行，执行成功后在源库执行。
- 修改字符集：先在目标库执行，执行成功后在源库执行。
- 修改列名：先在源库执行，执行成功后等待DRS任务因列不存在失败，然后在目标库执行，续传DRS任务。
- 加分区：先在目标库执行，执行成功后在源库执行。
- 删分区：先在源库执行，执行成功后在目标库执行。
- 加索引：跟操作顺序无关。
- 删索引：跟操作顺序无关。
- 加约束（主键、唯一键、check等）：先在源库执行，执行成功后在目标库执行。
- 删约束（主键、唯一键、check等）：先在目标库执行，执行成功后在源库执行。
- 增加字段长度：先在目标库执行，执行成功后在源库执行。
- 减少字段长度：先在源库执行，执行成功后在目标库执行。

### 📖 说明

如果一条DDL中包含多个操作的，所有操作除去跟操作顺序无关的操作外（比如修改默认值），剩余操作的要求顺序一样，则按要求顺序操作；否则需要拆成多条DDL操作。如加列的同时修改默认值，先在目标库执行，执行成功后在源库执行。

表 6-1 总结

| DDL操作                         | 操作顺序                |
|-------------------------------|---------------------|
| 加列，修改列类型，修改字符集，加分区，删约束，增加字段长度 | 先在目标库执行，执行成功后在源库执行。 |
| 删列，删分区，加约束，减少字段长度             | 先在源库执行，执行成功后在目标库执行。 |
| 增加、修改、删除列默认值，加索引，删索引          | 跟操作顺序无关。            |

| DDL操作 | 操作顺序                                         |
|-------|----------------------------------------------|
| 修改列名  | 先在源库执行，执行成功后等待DRS任务因列不存在失败，然后在目标库执行，续传DRS任务。 |

**步骤3** 根据**步骤2**顺序中完成DDL操作后，查看DRS任务状态是否正常。

----结束

## 6.3 DRS 是否支持双向实时同步

DRS目前支持MySQL->MySQL的双向同步，为白名单功能，需要提交工单申请才能使用。您可以在管理控制台右上角，选择“工单 > 新建工单”，完成工单提交。

双向同步任务在创建过程对步骤是有严格要求的，请按照以下步骤进行部署，以确保双向任务顺利部署。

1. 先创建双向同步任务，创建完成后，会生成两个子任务，即正向任务和反向任务，此时任务为配置状态。
2. 先配置正向任务，当正向任务进入“增量中”状态并且时延低于60s后，再配置并启动反向任务。
3. 等待反向任务状态到增量中，双向任务任务配置完成。

## 6.4 源库 Oracle 为 RAC 集群时，为什么建议使用 SCAN IP 连接

源库Oracle为RAC集群时，建议使用SCAN IP+ SERVICE\_NAMES方式创建任务，因为SCAN IP具有更强的容错性，更好的负载能力，更快的同步体验。

- 如果需要使用SCAN IP，需要保证SCAN IP与源库的所有VIP互通，否则无法通过测试连接检查。
- 若不使用SCAN IP，可以使用某一节点的VIP，其他节点异常不影响同步。

关于SCAN IP的说明，可参考[Oracle官网文档](#)。

## 6.5 源库 Oracle 补全日志检查方法

Oracle数据库在Physical Standby模式下，日志会从主库直接复制，而自身不产生任何日志。针对Oracle为源的增量同步链路，DRS需要用户提前手动在主库检查补全日志是否符合要求，以保证任务的正常运行。以下检查和设置方法中，

表级：针对指定表的设置。

库级：指整个数据库级别的设置。

PK/UI：每一行日志中除了记录变更的列以外，还额外记录了该行中主键和唯一键的值。

ALL：每一行日志中记录了该行所有列的值。

## 说明

以下三项检查，满足其中一项即可符合DRS增量同步的基本要求。

## 表级补全日志 PK/UI 检查（最低要求）

针对用户选择的待同步的表级对象，检查补全日志是否满足要求。

**步骤1** 在源库中执行以下sql语句。

```
select * from ALL_LOG_GROUPS where (LOG_GROUP_TYPE='UNIQUE KEY LOGGING' or
LOG_GROUP_TYPE='PRIMARY KEY LOGGING') and OWNER='大写SCHEMA名' and TABLE_NAME='大写表名';
```

该表名在查询结果中能同时对应到LOG\_GROUP\_TYPE值为UNIQUE KEY LOGGING和PRIMARY KEY LOGGING的两条记录，即可满足DRS增量同步要求。

**步骤2** 如果不满足要求，可执行以下sql语句开启表级PK/UI级别补全日志。

```
alter database add supplemental log data;
alter table SCHEMA名.表名 add supplemental log data(primary key,unique) columns;
```

### 须知

其中SCHEMA名.表名需要根据实际使用进行替换。

----结束

## 表级补全日志 ALL 检查

针对用户选择的待同步的表级对象，检查补全日志是否满足要求。

**步骤1** 在源库中执行以下sql语句。

```
select * from ALL_LOG_GROUPS where LOG_GROUP_TYPE='ALL COLUMN LOGGING' and OWNER='大写
SCHEMA名' and TABLE_NAME='大写表名';
```

该表名在查询结果中有记录，即可满足DRS增量同步要求。

**步骤2** 如果不满足要求，可执行以下sql语句开启表级ALL级别补全日志。

```
alter database add supplemental log data;
alter table SCHEMA名.表名 add supplemental log data(all) columns;
```

### 须知

其中SCHEMA名.表名需要根据实际使用进行替换。

----结束

## 库级补全日志检查

针对待同步的库级对象，检查补全日志是否满足要求。

**步骤1** 在源库执行以下sql语句。

```
select SUPPLEMENTAL_LOG_DATA_MIN MIN, SUPPLEMENTAL_LOG_DATA_PK PK,
SUPPLEMENTAL_LOG_DATA_UI UI, SUPPLEMENTAL_LOG_DATA_ALL ALL_LOG from v$database;
```

**步骤2** 满足以下其中一项要求即可。

- PK和UI同时为YES，即可满足DRS增量同步要求。  
如果不满足要求，可执行以下sql语句开启库级PK/UI级别补全日志。  
alter database add supplemental log data(primary key, unique) columns;
- ALL\_LOG为YES，即可满足DRS增量同步要求。  
如果不满足要求，可执行以下sql语句开启库级ALL级别补全日志。  
alter database add supplemental log data(all) columns;

----结束

## 6.6 同步到 GaussDB(DWS)的使用建议

### 增量 DDL 支持度

MySQL、Oracle等数据库语法与GaussDB(DWS)不同，增量阶段DDL同步存在失败的可能性，部分DDL转换规则可参考表6-2，建议在方案实施前联系GaussDB(DWS)技术支持评估DDL的支持度。

表 6-2 DDL 转换规则

| 语法转换点        | 转换结果   | 示例原语句                                                                                                          | 转换结果                                                                                                                                                                                                                                                           | 备注 |
|--------------|--------|----------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 表名、列名有反引号    | 替换为双引号 | create table `t1`(`c1` int, `c2` varchar(10));                                                                 | CREATE TABLE "public"."t1"("c1" INTEGER, "c2" VARCHAR(40)) WITH ( ORIENTATION = ROW, COMPRESSION = NO ) NOCOMPRESS DISTRIBUTE BY HASH ("c1");                                                                                                                  | -  |
| key 字段名(字段名) | 建索引    | Create Table mall_order_dc (id bigint NOT NULL AUTO_INCREMENT,order_id varchar(50) NOT NULL,key order_id(id)); | CREATE TABLE "public"."mall_order_dc"("id" BIGSERIAL NOT NULL, "order_id" VARCHAR(200) NOT NULL) WITH ( ORIENTATION = ROW, COMPRESSION = NO ) NOCOMPRESS DISTRIBUTE BY HASH ("id");<br>CREATE INDEX "order_id" ON "public"."mall_order_dc" USING BTREE ("id"); | -  |

| 语法转换点                          | 转换结果    | 示例原语句                                                                                                                                                                                                              | 转换结果                                                                                                                                                                                                                                                                                                                                                                                                    | 备注                                                                                                                      |
|--------------------------------|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------|
| 自定义的字段名和系统隐藏字段重名               | 加_new后缀 | <pre>Create Table mall_order_dc ( xc_node_id bigint NOT NULL AUTO_INCREMENT, tableoid varchar(50) NOT NULL, cmax int, xmax int, cmin char, xmin varchar(10), ctid smallint, tid time, tidd int, ctidd int );</pre> | <pre>CREATE TABLE "public"."mall_order_dc"( "xc_node_id_new" BIGSERIAL NOT NULL, "tableoid_new" VARCHAR(200) NOT NULL, "cmax_new" INTEGER, "xmax_new" INTEGER, "cmin_new" CHAR(4), "xmin_new" VARCHAR(40), "ctid_new" SMALLINT, "tid_new" TIME WITHOUT TIME ZONE, "tidd" INTEGER, "ctidd" INTEGER) WITH ( ORIENTATION = ROW, COMPRESSION = NO ) NOCOMPRESS DISTRIBUTE BY HASH ("xc_node_id_new");</pre> | GaussDB(DWS)有如下系统字段：<br>xc_node_id,<br>tableoid<br>, cmax,<br>xmax,<br>cmin,<br>xmin,<br>ctid,<br>tid, 最终会加_new后缀进行重命名。 |
| GaussDB(DWS)关键字做字段名, 字段名需要加双引号 | 加双引号    | -                                                                                                                                                                                                                  | -                                                                                                                                                                                                                                                                                                                                                                                                       | 所有表名、字段名均会添加双引号, 关键字如 desc, checksum, operator, size 等。                                                                 |
| GaussDB(DWS)关键字做表名, 表名需要加双引号   | 加双引号    | -                                                                                                                                                                                                                  | -                                                                                                                                                                                                                                                                                                                                                                                                       | 所有表名、字段名均会添加双引号, 关键字如 user。                                                                                             |

| 语法转换点        | 转换结果 | 示例原语句                                                                                                                                                                                                                                                                                                   | 转换结果                                                                                                                                                                                                                                                                                                                                                                          | 备注                                                              |
|--------------|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------|
| RENAME TABLE | -    | <pre> RENAME TABLE department TO newdept; RENAME TABLE employee TO pepole; RENAME TABLE newdept TO newdept_02,pepole to pepole_02;                     </pre>                                                                                                                                           | <pre> ALTER TABLE "public"."department" RENAME TO "newdept"; ALTER TABLE "public"."employee" RENAME TO "pepole"; ALTER TABLE "public"."newdept" RENAME TO "newdept_02"; ALTER TABLE "public"."pepole" RENAME TO "pepole_02";                     </pre>                                                                                                                       | <p>并列<br/>rename<br/>语句会转<br/>换成为多<br/>个<br/>rename<br/>语句。</p> |
| SET DEFAULT  | -    | <pre> ALTER TABLE runoob_alter_test ALTER dataType2 SET DEFAULT 1; ALTER TABLE runoob_alter_test ALTER COLUMN dataType2 SET DEFAULT 3;  ALTER TABLE runoob_alter_test ALTER dataType2 SET DEFAULT '1'; ALTER TABLE runoob_alter_test ALTER COLUMN dataType2 SET DEFAULT '3';                     </pre> | <pre> ALTER TABLE "public"."runoob_alter_test" ALTER COLUMN "datatype2" SET DEFAULT '1'; ALTER TABLE "public"."runoob_alter_test" ALTER COLUMN "datatype2" SET DEFAULT '3';  ALTER TABLE "public"."runoob_alter_test" ALTER COLUMN "datatype2" SET DEFAULT '1'; ALTER TABLE "public"."runoob_alter_test" ALTER COLUMN "datatype2" SET DEFAULT '3';                     </pre> | -                                                               |

| 语法转换点            | 转换结果 | 示例原语句                                                                                                                                                                                                                                                                                                                                                                               | 转换结果                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 备注 |
|------------------|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| CHANGE           | -    | ALTER TABLE<br>runoob_alter_test CHANGE<br>dataType1 dataType1New<br>VARCHAR(50);<br>ALTER TABLE<br>runoob_alter_test CHANGE<br>dataType2 dataType2New<br>VARCHAR(50) NOT NULL;<br>ALTER TABLE<br>runoob_alter_test CHANGE<br>dataType3 dataType3New<br>VARCHAR(100) FIRST;<br>ALTER TABLE<br>runoob_alter_test CHANGE<br>dataType4 dataType4New<br>VARCHAR(50) AFTER<br>dataType1; | ALTER TABLE<br>"public"."runoob_alter_test"<br>CHANGE COLUMN<br>"datatype1"<br>"datatype1new"<br>VARCHAR(200);<br>ALTER TABLE<br>"public"."runoob_alter_test"<br>CHANGE COLUMN<br>"datatype2"<br>"datatype2new"<br>VARCHAR(200) NOT NULL;<br>ALTER TABLE<br>"public"."runoob_alter_test"<br>CHANGE COLUMN<br>"datatype3"<br>"datatype3new"<br>VARCHAR(400);<br>ALTER TABLE<br>"public"."runoob_alter_test"<br>CHANGE COLUMN<br>"datatype4"<br>"datatype4new"<br>VARCHAR(200); | -  |
| MODIFY           | -    | ALTER TABLE<br>runoob_alter_test modify<br>datatype1 char default 1;<br>ALTER TABLE<br>runoob_alter_test modify<br>datatype1 char default '1';<br>ALTER TABLE<br>runoob_alter_test modify<br>datatype1 char default "1";                                                                                                                                                            | ALTER TABLE<br>"public"."runoob_alter_test"<br>MODIFY "datatype1"<br>CHAR(4) DEFAULT '1';<br>ALTER TABLE<br>"public"."runoob_alter_test"<br>MODIFY "datatype1"<br>CHAR(4) DEFAULT '1';<br>ALTER TABLE<br>"public"."runoob_alter_test"<br>MODIFY "datatype1"<br>CHAR(4) DEFAULT '1';                                                                                                                                                                                           | -  |
| ADD PRIMARY KEY  | -    | ALTER TABLE<br>runoob_alter_test ADD<br>PRIMARY KEY (dataType1);                                                                                                                                                                                                                                                                                                                    | ALTER TABLE<br>"public"."runoob_alter_test"<br>ADD PRIMARY<br>KEY("datatype1");                                                                                                                                                                                                                                                                                                                                                                                               | -  |
| DROP PRIMARY KEY | -    | ALTER TABLE<br>runoob_alter_test DROP<br>PRIMARY KEY;                                                                                                                                                                                                                                                                                                                               | ALTER TABLE<br>"public"."runoob_alter_test"<br>DROP CONSTRAINT IF<br>EXISTS<br>runoob_alter_test_pkey;                                                                                                                                                                                                                                                                                                                                                                        | -  |
| ADD COLUMN       | -    | ALTER TABLE<br>runoob_alter_test ADD<br>dataType1_1 INT NOT NULL<br>AFTER dataType1;                                                                                                                                                                                                                                                                                                | ALTER TABLE<br>"public"."runoob_alter_test"<br>ADD COLUMN<br>"datatype1_1" INTEGER<br>NOT NULL;                                                                                                                                                                                                                                                                                                                                                                               | -  |

| 语法转换点                 | 转换结果              | 示例原语句                                                                                                                                                                                                                                                                                                             | 转换结果                                                                                                                                                                                                                                                                                                                  | 备注                                                                                                        |
|-----------------------|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------|
| ADD INDEX             | -                 | ALTER TABLE runoob_tbl<br>ADD INDEX<br>idex_runoob_id(runoob_id)<br>USING BTREE;                                                                                                                                                                                                                                  | CREATE INDEX<br>"idex_runoob_id" ON<br>"public"."runoob_tbl"<br>("runoob_id");                                                                                                                                                                                                                                        | -                                                                                                         |
| ADD UNIQUE INDEX      | unique key 生成普通索引 | ALTER TABLE runoob_tbl<br>ADD UNIQUE KEY<br>IDEX_runoob_id USING<br>BTREE (runoob_id);                                                                                                                                                                                                                            | CREATE INDEX<br>"idex_runoob_id" ON<br>"public"."runoob_tbl"<br>("runoob_id");                                                                                                                                                                                                                                        | -                                                                                                         |
| DROP INDEX            | -                 | ALTER TABLE runoob_tbl<br>DROP KEY IDEX_runoob_id;                                                                                                                                                                                                                                                                | DROP INDEX<br>"public"."idex_runoob_id"<br>RESTRICT;                                                                                                                                                                                                                                                                  | -                                                                                                         |
| ALGORITHM             | 删除                | ALTER TABLE<br>runoob_alter_test<br>ALGORITHM=DEFAULT;<br>ALTER TABLE<br>runoob_alter_test<br>ALGORITHM=INPLACE;<br>ALTER TABLE<br>runoob_alter_test<br>ALGORITHM=COPY;                                                                                                                                           | -                                                                                                                                                                                                                                                                                                                     | -                                                                                                         |
| DEFAULT CHARACTER SET | 删除                | ALTER TABLE<br>runoob_alter_test<br>CHARACTER SET=utf8;<br>ALTER TABLE<br>runoob_alter_test DEFAULT<br>CHARACTER SET=utf8;                                                                                                                                                                                        | -                                                                                                                                                                                                                                                                                                                     | -                                                                                                         |
| COLLATE               | 删除                | -                                                                                                                                                                                                                                                                                                                 | -                                                                                                                                                                                                                                                                                                                     | COLLATE<br>表示默认<br>的数据库<br>排序规<br>则。                                                                      |
| DELAY_KEY_WRITE       | 删除                | CREATE TABLE<br>`public`.`runoob_tbl_test`(<br>`runoob_id` VARCHAR(30),<br>`runoob_title`<br>VARCHAR(100) NOT NULL,<br>`runoob_author`<br>VARCHAR(40) NOT NULL,<br>`submission_date`<br>VARCHAR(30)<br>) ENGINE=MyISAM,<br>DELAY_KEY_WRITE=0;<br>ALTER TABLE<br>`public`.`runoob_tbl_test6`<br>DELAY_KEY_WRITE=1; | CREATE TABLE<br>"PUBLIC"."RUNOOB_TBL_TEST"<br>(<br>"RUNOOB_ID"<br>VARCHAR(30),<br>"RUNOOB_TITLE"<br>VARCHAR(100) NOT NULL,<br>"RUNOOB_AUTHOR"<br>VARCHAR(40) NOT NULL,<br>"SUBMISSION_DATE"<br>VARCHAR(30)<br>) WITH ( ORIENTATION =<br>ROW, COMPRESSION = NO )<br>NOCOMPRESS<br>DISTRIBUTE BY HASH<br>("RUNOOB_ID"); | DELAY_K<br>EY_WRI<br>TE只对<br>MyISAM<br>引擎表有<br>作用, 根据<br>DELAY_K<br>EY_WRI<br>TE的值来<br>延迟更新<br>直至表关<br>闭。 |



| 语法转换点         | 转换结果 | 示例原语句                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | 转换结果                                                                                                                                                                                                                                                                                                                                                                                                                                               | 备注 |
|---------------|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| DIRECTORY     | 删除   | <pre>CREATE TABLE `public`.`runoob_tbl_test1` ( `dataType1` int NOT NULL AUTO_INCREMENT, `dataType2` DOUBLE(20,8), PRIMARY KEY(`dataType1`) ) ENGINE=MYISAM DATA DIRECTORY = 'D:\input' INDEX DIRECTORY= 'D:\ input'; CREATE TABLE `public`.`runoob_tbl_test2` ( `dataType1` int NOT NULL AUTO_INCREMENT, `dataType2` DOUBLE(20,8), PRIMARY KEY(`dataType1`) ) ENGINE=INNODB DATA DIRECTORY = 'D:\input'</pre>                                                                                                | <pre>CREATE TABLE "public"."runoob_tbl_test1"( "datatype1" SERIAL NOT NULL, "datatype2" FLOAT(20), PRIMARY KEY ("datatype1") ) WITH ( ORIENTATION = ROW, COMPRESSION = NO ) NOCOMPRESS DISTRIBUTE BY HASH ("datatype1"); CREATE TABLE "public"."runoob_tbl_test2"( "datatype1" SERIAL NOT NULL, "datatype2" FLOAT(20), PRIMARY KEY ("datatype1") ) WITH ( ORIENTATION = ROW, COMPRESSION = NO ) NOCOMPRESS DISTRIBUTE BY HASH ("datatype1");</pre> | -  |
| INSERT_METHOD | 删除   | <pre>CREATE TABLE `public`.`runoob_alter_test` ( `dataType1` int NOT NULL AUTO_INCREMENT, `dataType2` DOUBLE(20,8), `dataType3` TEXT NOT NULL, PRIMARY KEY(`dataType1`) ) INSERT_METHOD=LAST; ALTER TABLE runoob_alter_test INSERT_METHOD NO; ALTER TABLE runoob_alter_test INSERT_METHOD=NO; ALTER TABLE runoob_alter_test INSERT_METHOD FIRST; ALTER TABLE runoob_alter_test INSERT_METHOD=FIRST; ALTER TABLE runoob_alter_test INSERT_METHOD LAST; ALTER TABLE runoob_alter_test INSERT_METHOD=LAST;</pre> | <pre>CREATE TABLE "public"."runoob_alter_test"( "datatype1" SERIAL NOT NULL, "datatype2" FLOAT(20), "datatype3" TEXT NOT NULL, PRIMARY KEY ("datatype1") ) WITH ( ORIENTATION = ROW, COMPRESSION = NO ) NOCOMPRESS DISTRIBUTE BY HASH ("datatype1");</pre>                                                                                                                                                                                         | -  |

| 语法转换点          | 转换结果 | 示例原语句                                                                                                                                                                                                                                                                                                             | 转换结果                                                                                                                                                                                                                                                                               | 备注 |
|----------------|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| KEY_BLOCK_SIZE | 删除   | <pre>CREATE TABLE `public`.`runoob_tbl_test`( `runoob_id` VARCHAR(30), `runoob_title` VARCHAR(100) NOT NULL, `runoob_author` VARCHAR(40) NOT NULL, `submission_date` VARCHAR(30) ) ENGINE=MyISAM KEY_BLOCK_SIZE=8; ALTER TABLE runoob_tbl_test ENGINE=InnoDB; ALTER TABLE runoob_tbl_test KEY_BLOCK_SIZE=0;</pre> | <pre>CREATE TABLE "public"."runoob_tbl_test"( "runoob_id" VARCHAR(30), "runoob_title" VARCHAR(100) NOT NULL, "runoob_author" VARCHAR(40) NOT NULL, "submission_date" VARCHAR(30) ) WITH ( ORIENTATION = ROW, COMPRESSION = NO ) NOCOMPRESS DISTRIBUTE BY HASH ("runoob_id");</pre> | -  |
| MAX_ROWS       | 删除   | <pre>CREATE TABLE `public`.`runoob_alter_test`( `dataType1` int NOT NULL AUTO_INCREMENT, `dataType2` DOUBLE(20,8), `dataType3` TEXT NOT NULL, PRIMARY KEY(`dataType1`) ); ALTER TABLE runoob_alter_test MAX_ROWS 100000; ALTER TABLE runoob_alter_test MAX_ROWS=100000;</pre>                                     | <pre>CREATE TABLE "public"."runoob_alter_test"( "datatype1" SERIAL NOT NULL, "datatype2" FLOAT(20), "datatype3" TEXT NOT NULL, PRIMARY KEY ("datatype1") ) WITH ( ORIENTATION = ROW, COMPRESSION = NO ) NOCOMPRESS DISTRIBUTE BY HASH ("datatype1");</pre>                         | -  |
| MIN_ROWS       | 删除   | <pre>CREATE TABLE `public`.`runoob_alter_test`( `dataType1` int NOT NULL AUTO_INCREMENT, `dataType2` DOUBLE(20,8), `dataType3` TEXT NOT NULL, PRIMARY KEY(`dataType1`) ); ALTER TABLE runoob_alter_test MIN_ROWS 10000; ALTER TABLE runoob_alter_test MIN_ROWS=10000;</pre>                                       | <pre>CREATE TABLE "public"."runoob_alter_test"( "datatype1" SERIAL NOT NULL, "datatype2" FLOAT(20), "datatype3" TEXT NOT NULL, PRIMARY KEY ("datatype1") )WITH ( ORIENTATION = ROW, COMPRESSION = NO ) NOCOMPRESS DISTRIBUTE BY HASH ("datatype1");</pre>                          | -  |

| 语法转换点     | 转换结果 | 示例原语句                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 转换结果                                                                                                                                                                                                                                                                                         | 备注 |
|-----------|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| PACK_KEYS | 删除   | <pre>CREATE TABLE `public`.`runoob_alter_test`( `dataType1` int NOT NULL AUTO_INCREMENT, `dataType2` DOUBLE(20,8), `dataType3` TEXT NOT NULL, PRIMARY KEY(`dataType1`) ) ENGINE=MyISAM PACK_KEYS=1; ##A ALTER TABLE runoob_alter_test PACK_KEYS 0; ALTER TABLE runoob_alter_test PACK_KEYS=0; ##B ALTER TABLE runoob_alter_test PACK_KEYS 1; ALTER TABLE runoob_alter_test PACK_KEYS=1; ##C ALTER TABLE runoob_alter_test PACK_KEYS DEFAULT; ALTER TABLE runoob_alter_test PACK_KEYS=DEFAULT;</pre> | <pre>CREATE TABLE "public"."runoob_alter_test"( "datatype1" SERIAL NOT NULL, "datatype2" FLOAT(10), "datatype3" FLOAT(20), "datatype4" TEXT NOT NULL, PRIMARY KEY ("datatype1") )WITH ( ORIENTATION = ROW, COMPRESSION = NO ) NOCOMPRESS DISTRIBUTE BY HASH ("datatype1"); --A --B --C</pre> | -  |
| PASSWORD  | 删除   | <pre>CREATE TABLE `public`.`runoob_alter_test`( `dataType1` int NOT NULL AUTO_INCREMENT, `dataType2` DOUBLE(20,8), `dataType3` TEXT NOT NULL, PRIMARY KEY(`dataType1`) ); ALTER TABLE runoob_alter_test PASSWORD 'HELLO';</pre>                                                                                                                                                                                                                                                                     | <pre>CREATE TABLE "public"."runoob_alter_test" ( "datatype1" SERIAL NOT NULL, "datatype2" FLOAT(20), "datatype3" TEXT NOT NULL, PRIMARY KEY ("datatype1") ) WITH ( ORIENTATION = ROW, COMPRESSION = NO ) NOCOMPRESS DISTRIBUTE BY HASH ("datatype1");</pre>                                  | -  |

| 语法转换点                             | 转换结果 | 示例原语句                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | 转换结果                                                                                                                                                                                                                                                                                                    | 备注 |
|-----------------------------------|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| STAT<br>S_AU<br>TO_R<br>ECAL<br>C | 删除   | <pre>CREATE TABLE `public`.`runoob_alter_t est`( `runoob_id` VARCHAR(30), `runoob_title` VARCHAR(100) NOT NULL, `runoob_author` VARCHAR(40) NOT NULL, `submission_date` VARCHAR(30) ) ENGINE=InnoDB, STATS_AUTO_RECALC=D EFAULT; ## A. ALTER TABLE runoob_alter_test STATS_AUTO_RECALC DEFAULT; ALTER TABLE runoob_alter_test STATS_AUTO_RECALC=D EFAULT; ## B. ALTER TABLE runoob_alter_test STATS_AUTO_RECALC 0; ALTER TABLE runoob_alter_test STATS_AUTO_RECALC=0; ## C. ALTER TABLE runoob_alter_test STATS_AUTO_RECALC 1; ALTER TABLE runoob_alter_test STATS_AUTO_RECALC=1;</pre> | <pre>CREATE TABLE "public"."runoob_alter_t est" ( "runoob_id" VARCHAR(30), "runoob_title" VARCHAR(100) NOT NULL, "runoob_author" VARCHAR(40) NOT NULL, "submission_date" VARCHAR(30) ) WITH ( ORIENTATION = ROW, COMPRESSION = NO ) NOCOMPRESS DISTRIBUTE BY HASH ("runoob_id"); -- A. -- B. -- C</pre> | -  |

| 语法转换点                        | 转换结果 | 示例原语句                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 转换结果                                                                                                                                                                                                                                                                           | 备注 |
|------------------------------|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| STAT<br>S_PER<br>SISTE<br>NT | 删除   | <pre>CREATE TABLE `public`.`runoob_alter_test`( `dataType1` int NOT NULL AUTO_INCREMENT, `dataType2` DOUBLE(20,8), `dataType3` TEXT NOT NULL, PRIMARY KEY(`dataType1`) ) ENGINE=InnoDB, STATS_PERSISTENT=0; ## A. ALTER TABLE runoob_alter_test STATS_PERSISTENT DEFAULT; ALTER TABLE runoob_alter_test STATS_PERSISTENT=DEF AULT; ## B. ALTER TABLE runoob_alter_test STATS_PERSISTENT 0; ALTER TABLE runoob_alter_test STATS_PERSISTENT=0; ## C. ALTER TABLE runoob_alter_test STATS_PERSISTENT 1; ALTER TABLE runoob_alter_test STATS_PERSISTENT=1</pre> | <pre>CREATE TABLE "public"."runoob_alter_t est" ( "datatype1" SERIAL NOT NULL, "datatype2" FLOAT(20), "datatype3" TEXT NOT NULL, PRIMARY KEY ("datatype1") ) WITH ( ORIENTATION = ROW, COMPRESSION = NO ) NOCOMPRESS DISTRIBUTE BY HASH ("datatype1"); -- A. -- B. -- C.</pre> | -  |

| 语法转换点                              | 转换结果 | 示例原语句                                                                                                                                                                                                                                                                                                                              | 转换结果                                                                                                                                                                                                                                                         | 备注 |
|------------------------------------|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| STAT<br>S_SA<br>MPLE<br>_PAG<br>ES | 删除   | <pre>CREATE TABLE `public`.`runoob_alter_t est`( `dataType1` int NOT NULL AUTO_INCREMENT, `dataType2` DOUBLE(20,8), `dataType3` TEXT NOT NULL, PRIMARY KEY(`dataType1`) ) ENGINE=InnoDB,STATS_ SAMPLE_PAGES=25; ALTER TABLE runoob_alter_test STATS_SAMPLE_PAGES 100; ALTER TABLE runoob_alter_test STATS_SAMPLE_PAGES= 100;</pre> | <pre>CREATE TABLE "public"."runoob_alter_t est" ( "datatype1" SERIAL NOT NULL, "datatype2" FLOAT(20), "datatype3" TEXT NOT NULL, PRIMARY KEY ("datatype1") ) WITH ( ORIENTATION = ROW, COMPRESSION = NO ) NOCOMPRESS DISTRIBUTE BY HASH ("datatype1");</pre> | -  |

| 语法转换点     | 转换结果 | 示例原语句                                                                                                                                                                                                                                                                                                                                                                                                                                             | 转换结果                                                                                                                                                                                                                                                 | 备注                                                                                                                                                           |
|-----------|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 创建表指定引擎类型 | 删除   | <pre>CREATE TABLE `public`.`runoob_alter_test`( `dataType1` int NOT NULL, `dataType2` DOUBLE(20,8), PRIMARY KEY(`dataType1`) )ENGINE=MYISAM; ## A. ALTER TABLE runoob_alter_test ENGINE INNODB; ALTER TABLE runoob_alter_test ENGINE=INNODB; ## B. ALTER TABLE runoob_alter_test ENGINE MYISAM; ALTER TABLE runoob_alter_test ENGINE=MYISAM; ## C. ALTER TABLE runoob_alter_test ENGINE MEMORY; ALTER TABLE runoob_alter_test ENGINE=MEMORY</pre> | <pre>CREATE TABLE "public"."runoob_alter_t est" ( "datatype1" INTEGER NOT NULL, "datatype2" FLOAT(20), PRIMARY KEY ("datatype1") ) WITH ( ORIENTATION = ROW, COMPRESSION = NO ) NOCOMPRESS DISTRIBUTE BY HASH ("datatype1"); -- A. -- B. -- C.</pre> | <p>在MySQL中，ENGINE指定表的存储引擎。当存储引擎为ARCHIVE、BLACKHOLE、CSV、FEDERATED、INNODB、MYISAM、MEMORY、MRG_MYISAM、NDB、NDBCLUSTER和PERFORMANCE_SCHEMA时，支持该属性迁移，迁移过程中会将该属性删除。</p> |

| 语法转换点    | 转换结果 | 示例原语句                                                                                                                                                                                                                                                                                                                                                           | 转换结果                                                                                                                                                                                                                                                      | 备注 |
|----------|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| CHECKSUM | 删除   | <pre>CREATE TABLE `public`.`runoob_alter_test`( `dataType1` int NOT NULL AUTO_INCREMENT, `dataType2` FLOAT(10,2), `dataType3` DOUBLE(20,8), PRIMARY KEY(`dataType1`) ) CHECKSUM=1; ALTER TABLE runoob_alter_test CHECKSUM 0; ALTER TABLE runoob_alter_test CHECKSUM=0; ALTER TABLE runoob_alter_test CHECKSUM 1; ALTER TABLE runoob_alter_test CHECKSUM=1</pre> | <pre>CREATE TABLE "public"."runoob_alter_test" ( "datatype1" SERIAL NOT NULL, "datatype2" REAL, "datatype3" DOUBLE PRECISION, PRIMARY KEY ("datatype1") ) WITH ( ORIENTATION = ROW, COMPRESSION = NO ) NOCOMPRESS DISTRIBUTE BY HASH ("datatype1");</pre> | -  |



| 语法转换点                          | 转换结果 | 示例原语句                                                                                                                                                                                                               | 转换结果                                                                                                                                                                                                                                                                                                                                     | 备注                          |
|--------------------------------|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------|
| ON UPDATE CURRENT_TIMESTAMP    | 删除   | drop table if exists<br>unsupport_parse_test;<br>create table<br>`unsupport_parse_test` (<br>`username` int,<br>`update` timestamp not<br>null default<br>current_timestamp on<br>update<br>current_timestamp<br>); | DROP TABLE IF EXISTS<br>"public"."unsupport_parse_test";<br>CREATE TABLE<br>"public"."unsupport_parse_test"<br>(<br>"username" INTEGER,<br>"update" TIMESTAMP<br>WITH TIME ZONE NOT<br>NULL DEFAULT<br>CURRENT_TIMESTAMP<br>)<br>WITH ( ORIENTATION =<br>ROW, COMPRESSION =<br>NO )<br>NOCOMPRESS<br>DISTRIBUTE BY HASH<br>("username"); | -                           |
| SET FOREIGN_KEY_CHECKS=0或1     | 删除   | -                                                                                                                                                                                                                   | -                                                                                                                                                                                                                                                                                                                                        | -                           |
| PRIMARY KEY (`id`) USING BTREE | 删除   | Create Table<br>mall_order_dc (<br>id int,<br>name varchar(10),<br>primary key(`id`) using<br>btree<br>);                                                                                                           | CREATE TABLE<br>"public"."mall_order_dc"<br>(<br>"id" INTEGER,<br>"name" VARCHAR(40),<br>PRIMARY KEY ("id")<br>)<br>WITH ( ORIENTATION =<br>ROW, COMPRESSION =<br>NO )<br>NOCOMPRESS<br>DISTRIBUTE BY HASH<br>("id");                                                                                                                    | 默认<br>btree,<br>不需要此<br>关键字 |

| 语法转换点                      | 转换结果                | 示例原语句 | 转换结果 | 备注                |
|----------------------------|---------------------|-------|------|-------------------|
| 行列存转换                      | 默认行存                | -     | -    | -                 |
| create table if not exists | 支持if not exists 关键字 | -     | -    | -                 |
| charset                    | 迁移时会<br>将该关键字删除。    | -     | -    | CHARSET指定表的默认字符集。 |

| 语法转换点 | 转换结果           | 示例原语句                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | 转换结果                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 备注 |
|-------|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 分区表   | 支持 RANG<br>E分区 | <pre>drop table if exists `runoob_tbl_part_test`; CREATE TABLE IF NOT EXISTS `runoob_tbl_part_test`( `runoob_id` INT NOT NULL, `runoob_title` VARCHAR(100) NOT NULL, `runoob_author` VARCHAR(40) NOT NULL, `submission_date` INT )ENGINE=InnoDB DEFAULT CHARSET=utf8 PARTITION BY RANGE COLUMNS(runoob_id, submission_date)( PARTITION p0 VALUES LESS THAN(123, MAXVALUE), PARTITION p1 VALUES LESS THAN(200, MAXVALUE), PARTITION p2 VALUES LESS THAN(300, MAXVALUE), PARTITION p3 VALUES LESS THAN(400, MAXVALUE), PARTITION p4 VALUES LESS THAN(500, MAXVALUE), PARTITION p5 VALUES LESS THAN(MAXVALUE, MAXVALUE) );</pre> | <pre>CREATE TABLE IF NOT EXISTS "public"."runoob_tbl_par t_test" ( "runoob_id" INTEGER NOT NULL, "runoob_title" VARCHAR(400) NOT NULL, "runoob_author" VARCHAR(160) NOT NULL, "submission_date" INTEGER ) WITH ( ORIENTATION = ROW, COMPRESSION = NO ) NOCOMPRESS DISTRIBUTE BY HASH ("runoob_id") PARTITION BY RANGE ("runoob_id", "submission_date") ( PARTITION p0 VALUES LESS THAN (123, MAXVALUE), PARTITION p1 VALUES LESS THAN (200, MAXVALUE), PARTITION p2 VALUES LESS THAN (300, MAXVALUE), PARTITION p3 VALUES LESS THAN (400, MAXVALUE), PARTITION p4 VALUES LESS THAN (500, MAXVALUE),</pre> | -  |

| 语法转换点    | 转换结果 | 示例原语句                                                                                                                                                            | 转换结果                                                                                                                                                                                                      | 备注                                     |
|----------|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------|
|          |      |                                                                                                                                                                  | PARTITION p5 VALUES LESS THAN (MAXVALUE, MAXVALUE);                                                                                                                                                       |                                        |
| 字段指定字符编码 | 删除   | CREATE TABLE `FCT_TRADE_***_RT` ( `KID` INT(11) NOT NULL AUTO_INCREMENT, `YM` VARCHAR(7) CHARSET UTF8MB4 COLLATE UTF8MB4_BIN NOT NULL COMMENT '年月, 例如2019-04' ); | CREATE TABLE "public"."fct_trade_***_rt" ( "kid" SERIAL NOT NULL, "ym" VARCHAR(28) NOT NULL COMMENT '年月, 例如2019-04' ) WITH ( ORIENTATION = ROW, COMPRESSION = NO ) NOCOMPRESS DISTRIBUTE BY HASH ("kid"); | -                                      |
| 带数据宽度的整型 | 删除宽度 | CREATE TABLE `FCT_TRADE_***_RT` ( `KID` INT(11) NOT NULL AUTO_INCREMENT, `YM` SMALLINT(15), `c3` BIGINT(50) );                                                   | CREATE TABLE "public"."fct_trade_***_rt" ( "kid" SERIAL NOT NULL, "ym" SMALLINT, "c3" BIGINT ) WITH ( ORIENTATION = ROW, COMPRESSION = NO ) NOCOMPRESS DISTRIBUTE BY HASH ("kid");                        | 数字类型 int、smallint、bigint 不支持带宽度, 统一删除。 |

| 语法转换点             | 转换结果                                   | 示例原语句                                                                                                                                                                                                                                           | 转换结果                                                                                                                                                                                                                                                                        | 备注 |
|-------------------|----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| unsigned zerofill | 删除                                     | <pre>CREATE TABLE IF NOT EXISTS `runoob_dataType_test`( `dataType_1` DEC, `dataType_2` DEC(10), `dataType_3` DEC(10, 2) UNSIGNED ZEROFILL, `dataType_4` DEC(10, 2) ZEROFILL )ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_general_c i;</pre> | <pre>CREATE TABLE IF NOT EXISTS "public"."runoob_datatype_test" ( "datatype_1" DECIMAL, "datatype_2" DECIMAL(10), "datatype_3" DECIMAL(10,2), "datatype_4" DECIMAL(10,2) ) WITH ( ORIENTATION = ROW, COMPRESSION = NO ) NOCOMPRESS DISTRIBUTE BY HASH ("datatype_1");</pre> | -  |
| auto_increment    | 不支持该参数，迁移时会设置该属性的字段迁移为SERIAL类型，并删除关键字。 | <pre>CREATE TABLE `public`.`job_instance` ( `job_sche_id` int(11) NOT NULL AUTO_INCREMENT, `task_name` varchar(100) NOT NULL DEFAULT ", PRIMARY KEY (`job_sche_id`) ) ENGINE=InnoDB AUTO_INCREMENT=219 DEFAULT CHARSET=utf8</pre>               | <pre>CREATE TABLE "public"."job_instance" ( "job_sche_id" SERIAL NOT NULL, "task_name" VARCHAR(100) NOT NULL DEFAULT ", PRIMARY KEY ("job_sche_id") ) WITH ( ORIENTATION = ROW, COMPRESSION = NO ) NOCOMPRESS DISTRIBUTE BY HASH ("job_sche_id");</pre>                     | -  |

| 语法转换点          | 转换结果                             | 示例原语句                                                                                                                                                                                                            | 转换结果                                                                                                                                                                                                                                                                                | 备注 |
|----------------|----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| AVG_ROW_LENGTH | 不支持此属性，该属性将迁移到SERIAL，并在迁移过程中被删除。 | <pre>CREATE TABLE `public`.`runoob_tbl_test ` ( `runoob_id` VARCHAR(30), `runoob_title` VARCHAR(100) NOT NULL, `runoob_author` VARCHAR(40) NOT NULL, `submission_date` VARCHAR(30) )AVG_ROW_LENGTH=10 000;</pre> | <pre>CREATE TABLE "public"."runoob_tbl_test " ( "runoob_id" VARCHAR(30), "runoob_title" VARCHAR(100) NOT NULL, "runoob_author" VARCHAR(40) NOT NULL, "submission_date" VARCHAR(30) ) WITH ( ORIENTATION = ROW, COMPRESSION = NO ) NOCOMPRESS DISTRIBUTE BY HASH ("runoob_id")</pre> | -  |

| 语法转换点 | 转换结果                        | 示例原语句                                                                                                                                                                                                                                                                                                                                               | 转换结果                                                                                                                                                                                                                                                                                                                                                                                                             | 备注                                                                       |
|-------|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------|
| UNION | 迁移时会将该特性转为 Gauss DB 视图创建语句。 | <pre>CREATE TABLE t1 (   a INT NOT NULL   AUTO_INCREMENT   PRIMARY KEY,   message CHAR(20) ) ENGINE=MyISAM; CREATE TABLE t2 (   a INT NOT NULL   AUTO_INCREMENT   PRIMARY KEY,   message CHAR(20) ) ENGINE=MyISAM; CREATE TABLE total (   a INT NOT NULL   AUTO_INCREMENT,   message CHAR(20)) ENGINE=MERGE UNION=(t1,t2) INSERT_METHOD=LAST;</pre> | <pre>CREATE TABLE t1 (   a SERIAL NOT NULL   PRIMARY KEY,   message CHAR(20) ) WITH ( ORIENTATION = ROW, COMPRESSION = NO ) NOCOMPRESS DISTRIBUTE BY HASH ("a"); CREATE TABLE t2 (   a SERIAL NOT NULL   PRIMARY KEY,   message CHAR(20) ) WITH ( ORIENTATION = ROW, COMPRESSION = NO ) NOCOMPRESS DISTRIBUTE BY HASH ("a"); CREATE VIEW total(a, message) AS SELECT * FROM t1 UNION ALL SELECT * FROM t2;</pre> | UNION 是 MERGE 引擎的建表参数。通过该关键字建表类似于创建普通视图。新创建的表将在逻辑上合并 UNION 关键字限定的多个表的数据。 |

| 语法转换点                   | 转换结果               | 示例原语句                                                                                                                                                                                                                             | 转换结果                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 备注 |
|-------------------------|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| LIKE<br>表克隆             | 迁移时会添加额外的表属性信息。    | <pre>CREATE TABLE IF NOT EXISTS `public`.`runoob_tbl_old` ( `dataType_1` YEAR, `dataType_2` YEAR(4), `dataType_3` YEAR DEFAULT '2018', `dataType_4` TIME DEFAULT NULL ); CREATE TABLE `runoob_tbl` (like `runoob_tbl_old`);</pre> | <pre>CREATE TABLE "public"."runoob_tbl_old" ( "datatype_1" VARCHAR(4), "datatype_2" VARCHAR(4), "datatype_3" VARCHAR(4) DEFAULT '2018', "datatype_4" TIME WITHOUT TIME ZONE DEFAULT NULL ) WITH ( ORIENTATION = ROW, COMPRESSION = NO ) NOCOMPRESS DISTRIBUTE BY HASH ("datatype_1"); CREATE TABLE "public"."runoob_tbl"( LI KE "public"."runoob_tbl_old"  INCLUDING COMMENTS INCLUDING CONSTRAINTS INCLUDING DEFAULTS INCLUDING INDEXES INCLUDING STORAGE);</pre> | -  |
| drop<br>表               | 默认 schema 为 public | <pre>DROP TABLE `test_create_table01`;</pre>                                                                                                                                                                                      | <pre>DROP TABLE "public"."test_create_tabl e01";</pre>                                                                                                                                                                                                                                                                                                                                                                                                             | -  |
| TRUN<br>CATE<br>删除<br>表 | -                  | <pre>TRUNCATE TABLE `test_create_table01`;</pre>                                                                                                                                                                                  | <pre>TRUNCATE TABLE "public"."test_create_tabl e01" CONTINUE IDENTITY RESTRICT;</pre>                                                                                                                                                                                                                                                                                                                                                                              | -  |



| 语法转换点   | 转换结果                          | 示例原语句                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 转换结果                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 备注 |
|---------|-------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| HASH 索引 | 迁移时会根据 Gauss DB 的特性将其迁移为普通索引。 | <pre>CREATE TABLE `public`.`test_create_table03` ( `DEMAND_ID` INT(11) NOT NULL AUTO_INCREMENT, `DEMAND_NAME` CHAR(100) NOT NULL, `THEME` VARCHAR(200) NULL DEFAULT NULL, `SEND_ID` INT(11) NULL DEFAULT NULL, `SEND_NAME` CHAR(20) NULL DEFAULT NULL, `SEND_TIME` DATETIME NULL DEFAULT NULL, `DEMAND_CONTENT` TEXT NOT NULL, PRIMARY KEY(`DEMAND_ID`), INDEX CON_INDEX(DEMAND_C ONTENT(100)) USING HASH , INDEX SEND_INFO_INDEX USING HASH (SEND_ID,SEND_NAME( 10),SEND_TIME) ); ALTER TABLE runoob_alter_test ADD KEY alterTable_addKey_index Type(dataType1) USING HASH;</pre> | <pre>CREATE TABLE "public"."test_create_table03" ( "demand_id" SERIAL NOT NULL, "demand_name" CHAR(100) NOT NULL, "theme" VARCHAR(200) DEFAULT NULL, "send_id" INTEGER(11) DEFAULT NULL, "send_name" CHAR(20) DEFAULT NULL, "send_time" TIMESTAMP WITHOUT TIME ZONE DEFAULT NULL, "demand_content" TEXT NOT NULL, PRIMARY KEY ("demand_id") ) WITH ( ORIENTATION = ROW, COMPRESSION = NO ) NOCOMPRESS DISTRIBUTE BY HASH ("demand_id"); CREATE INDEX "con_index" ON "public"."test_create_table03" ("demand_content"); CREATE INDEX "send_info_index" ON "public"."test_create_table03" ("send_id","send_name", "send_time"); CREATE INDEX "altertable_addkey_index type" ON</pre> | -  |

| 语法转换点   | 转换结果                      | 示例原语句                                                                                       | 转换结果                                                                                  | 备注 |
|---------|---------------------------|---------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|----|
|         |                           |                                                                                             | "public"."runoob_alter_test" ("datatype1");                                           |    |
| BTREE索引 | 迁移时会根据 Gauss DB的特性进行相应适配。 | ALTER TABLE runoob_alter_test ADD KEY alterTable_addKey_index Type (dataType1) USING BTREE; | CREATE INDEX "altertable_addkey_index" ON "public"."runoob_alter_test" ("datatype1"); | -  |

| 语法转换点        | 转换结果                       | 示例原语句                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 转换结果                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 备注 |
|--------------|----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| SPATIAL 空间索引 | 迁移时会根据 Gauss DB 的特性进行相应适配。 | <pre>CREATE TABLE `public`.`test_create_table04` ( `ID` INT(11) NOT NULL AUTO_INCREMENT PRIMARY KEY, `A` POINT NOT NULL, `B` POLYGON NOT NULL, `C` GEOMETRYCOLLECTION NOT NULL, `D` LINESTRING NOT NULL, `E` MULTILINESTRING NOT NULL, `F` MULTIPOINT NOT NULL, `G` MULTIPOLYGON NOT NULL, SPATIAL INDEX A_INDEX(A), SPATIAL INDEX B_INDEX(B), SPATIAL INDEX C_INDEX(C), SPATIAL KEY D_INDEX(D), SPATIAL KEY E_INDEX(E), SPATIAL KEY F_INDEX(F), SPATIAL INDEX G_INDEX(G) );</pre> | <pre>CREATE TABLE "public"."test_create_table04" ( "id" SERIAL NOT NULL PRIMARY KEY, "a" POINT NOT NULL, "b" POLYGON NOT NULL, "c" CIRCLE NOT NULL, "d" POLYGON NOT NULL, "e" BOX NOT NULL, "f" BOX NOT NULL, "g" POLYGON NOT NULL ) WITH ( ORIENTATION = ROW, COMPRESSION = NO ) NOCOMPRESS DISTRIBUTE BY HASH ("id"); CREATE INDEX "a_index" ON "public"."test_create_table04" USING GIST ("a"); CREATE INDEX "b_index" ON "public"."test_create_table04" USING GIST ("b"); CREATE INDEX "c_index" ON "public"."test_create_table04" USING GIST ("c"); CREATE INDEX "d_index" ON "public"."test_create_table04" USING GIST ("d"); CREATE INDEX "e_index" ON "public"."test_create_table04" USING GIST ("e");</pre> | -  |

| 语法转换点 | 转换结果 | 示例原语句 | 转换结果                                                                                                                                                            | 备注 |
|-------|------|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
|       |      |       | <pre>CREATE INDEX "f_index" ON "public"."test_create_table04" USING GIST ("f"); CREATE INDEX "g_index" ON "public"."test_create_table04" USING GIST ("g")</pre> |    |

| 语法转换点        | 转换结果                       | 示例原语句                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 转换结果                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 备注 |
|--------------|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| FULLTEXT全文索引 | 迁移时会根据 Gauss DB 的特性进行相应适配。 | <pre>CREATE TABLE `public`.`test_create_table02` ( `ID` INT(11) NOT NULL PRIMARY KEY, `TITLE` CHAR(255) NOT NULL, `CONTENT` TEXT NULL, `CREATE_TIME` DATETIME NULL DEFAULT NULL, FULLTEXT (`CONTENT` ));  CREATE TABLE IF NOT EXISTS `public`.`runoob_dataType_test` ( `id` INT PRIMARY KEY AUTO_INCREMENT, `name` VARCHAR(128) NOT NULL, FULLTEXT INDEX (name) );  CREATE TABLE IF NOT EXISTS `public`.`runoob_dataType_test` ( `id` INT PRIMARY KEY AUTO_INCREMENT, `name` VARCHAR(128) NOT NULL, FULLTEXT INDEX (name ASC) );</pre> | <pre>CREATE TABLE "public"."test_create_table02" ( "id" INTEGER(11) NOT NULL PRIMARY KEY, "title" CHAR(255) NOT NULL, "content" TEXT, "create_time" TIMESTAMP WITHOUT TIME ZONE DEFAULT NULL ) WITH ( ORIENTATION = ROW, COMPRESSION = NO ) NOCOMPRESS DISTRIBUTE BY HASH ("id");  CREATE INDEX "idx_test_create_table02_content" ON "public"."test_create_table02" USING GIN(to_tsvector(coalesce("content","")));  CREATE TABLE "public"."runoob_datatype_test" ( "id" SERIAL PRIMARY KEY, "name" VARCHAR(128) NOT NULL ) WITH ( ORIENTATION = ROW, COMPRESSION = NO ) NOCOMPRESS DISTRIBUTE BY HASH ("id");</pre> | -  |

| 语法转换点 | 转换结果 | 示例原语句 | 转换结果                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 备注 |
|-------|------|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
|       |      |       | <pre>CREATE INDEX "idx_runoob_datatype_test_name" ON "public"."runoob_datatype_test" USING GIN(to_tsvector(coalesce("name",""))); CREATE TABLE "public"."runoob_datatype_test" ( "id" SERIAL PRIMARY KEY, "name" VARCHAR(128) NOT NULL ) WITH ( ORIENTATION = ROW, COMPRESSION = NO ) NOCOMPRESS DISTRIBUTE BY HASH ("id"); CREATE INDEX "idx_runoob_datatype_test_name" ON "public"."runoob_datatype_test" USING GIN(to_tsvector(coalesce("name","")));</pre> |    |

| 语法转换点               | 转换结果                       | 示例原语句                                                                                                                                                                                                                                                                                                                                                       | 转换结果                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 备注 |
|---------------------|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| CHECK约束             | 迁移时会根据 Gauss DB 的特性进行相应适配。 | <pre>drop table if exists `t1`; create table if not exists t1 (   id int(25) not null   primary key check (id &gt;   1 and id &lt; 100),   city varchar(255) not   null unique check   (city='CityA' or   city='CityB' or   city='CityC'),   population int(25) not   null ,   constraint t1_1 check   (population&gt;0 and   population&lt;10000) );</pre> | <pre>DROP TABLE IF EXISTS "public"."t1"; CREATE TABLE IF NOT EXISTS "public"."t1" (   "id" INTEGER NOT NULL   PRIMARY KEY CHECK (   id &gt; 1   and id &lt; 100   ),   "city" VARCHAR(1020)   NOT NULL CHECK (   city = 'CityA'   or city = 'CityB'   or city = 'CityC'   ),   "population" INTEGER   NOT NULL,   CONSTRAINT t1_1   CHECK (   population &gt; 0   and population &lt; 10000   ) ) WITH (ORIENTATION = ROW, COMPRESSION = NO) NOCOMPRESS DISTRIBUTE BY HASH ("id</pre> | -  |
| GENERATED ALWAYS AS | 删除                         | <pre>create table `createTable_others11` (   id INT PRIMARY KEY,   price DECIMAL(10, 2),   discount DECIMAL(4, 2),   discounted_price   DECIMAL(10, 2)   GENERATED ALWAYS AS   (price - price * discount)   STORED );</pre>                                                                                                                                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | -  |

## 多表归一场景注意事项

DRS支持多表对一的数据聚合场景，最佳实践是推荐只选择同步加列DDL，其他大部分DDL同步都可能会因目标表修改而导致其他任务失败或数据不一致的情况发生，常见情况有：同步truncate导致目标数据全部被清空；同步创建索引导致目标表被锁定；同步rename导致其他任务找不到目标表而失败；同步改列导致其他任务因数据类型不兼容而失败。

- **多表归一高风险操作：**

- DROP、TRUNCATE、RENAME等高危DDL未过滤，导致目标表DROP、TRUNCATE、RENAME。
- 随意执行DDL，导致同步任务中断未提前识别业务数据冲突，出现同步后数据不一致。
- 多对一的表不支持Online DDL工具。Online DDL通常会使用临时表和执行rename操作，多对一任务如果未过滤rename会导致目标数据丢失。

- **多表归一DDL操作技巧：**

- 增加列ADD COLUMN：

配置DRS任务时过滤ADD COLUMN：先在目标库目标表增加列，再在每个多对一任务的源库源表分别增加列。同步任务会自动兼容源表列少于目标表的场景，因此任务不受影响。顺序相反则会导致任务中断。

配置DRS任务时未过滤ADD COLUMN：DRS会自动识别只执行一次DDL，不会导致任务失败。

- 修改列MODIFY COLUMN：

配置DRS任务时过滤MODIFY COLUMN：需要确保所有源表已经完成同步无时延并且无业务写入，先在目标库目标表修改列，再在每个多对一任务的源库源表分别修改列。

配置DRS任务时未过滤MODIFY COLUMN：DRS会自动识别只执行一次DDL，不会导致任务失败。

- 删除列DROP COLUMN：

配置DRS任务时过滤DROP COLUMN：先在每个源库源表删除列，多对一任务的所有源表删除列后，再在目标库表删除列。同步任务会自动兼容源表列少于目标表的场景，因此任务不受影响。顺序相反则会导致任务中断。

配置DRS任务时未过滤DROP COLUMN：DRS会自动识别只执行一次DDL，不会导致任务失败。

- DROP、TRUNCATE类操作：

高危操作建议配置DRS任务时过滤掉，手工执行。如果未选择过滤，多对一任务一张源表的DROP、TRUNCATE操作会同步到目标表执行DROP、TRUNCATE操作，比如多对一任务DROP掉其中一个表，在目标库就把多对一的表DROP掉了，DROP索引、DROP约束是类似的。

## 6.7 Oracle 到 GaussDB 字符集兼容性说明

Oracle到GaussDB同步时，当源库和目标库字符集不兼容时，可能会导致数据乱码或者同步失败的场景，可参考[表6-3](#)进行设置。



表 6-3 字符集兼容性

| 源库/目标库           | UTF8 | GBK | GB2312 | GB18030 |
|------------------|------|-----|--------|---------|
| US7ASCII         | √    | √   | √      | √       |
| UTF8             | √    | ×   | ×      | ×       |
| AL32UTF8         | √    | ×   | ×      | ×       |
| ZHS16GBK         | √    | √   | ×      | √       |
| WE8ISO8859P1     | √    | ×   | ×      | ×       |
| WE8ISO8859P2     | √    | ×   | ×      | ×       |
| WE8ISO8859P4     | √    | ×   | ×      | ×       |
| WE8ISO8859P5     | √    | ×   | ×      | ×       |
| WE8ISO8859P7     | √    | ×   | ×      | ×       |
| WE8ISO8859P9     | √    | ×   | ×      | ×       |
| WE8ISO8859P13    | √    | ×   | ×      | ×       |
| WE8ISO8859P15    | √    | ×   | ×      | ×       |
| WE8MSWIN125<br>2 | √    | ×   | ×      | ×       |

## 6.8 字符集不兼容导致数据乱码或者同步失败

### 背景

当源数据库和目标数据库字符集不兼容时，可能导致

- 部分数据乱码。
- 数据同步不一致。
- 目标库写入失败。

### 解决方案

针对这种情况，建议先修改目标库字符集后再进行同步。若同步任务已经报错，请修改目标库字符集后再重置任务进行同步。

## 6.9 DRS 增量同步怎样填启动位点

### MySQL 为源

MySQL的单增量同步任务，在“设置同步”页面需要填写启动位点，任务源库日志从位点后开始获取（不含当前启动位点）。

图 6-1 启动位点

启动位点  ;  ?

当数据库的gtid\_mode参数是on的情况下，位点信息由三部分构成：文件名（File）、位置（Position）和已执行的Global Transaction Identifiers Set（Executed\_Gtid\_Set），这些值可通过show master status命令获取（如果源库为MySQL 5.5版本，则不支持）。填写Executed\_Gtid\_Set的时候，需要删换其中的换行符再进行填入。

例如，社区版MySQL获取信息如下：

```
mysql> show master status\G
***** 1. row *****
 File: binlog.1011289
 Position: 1611713
 Binlog_Do_DB:
 Binlog_Ignore_DB:
 Executed_Gtid_Set: c8cd8ff7-da6f-11ec-a945-00d861ef1161:1-3,
f4f9537f-0c8b-11ec-a284-00d861ef116a:1-10820402
1 row in set (0.00 sec)
```

根据获取信息，任务的启动位点设置为如下，其中需要填入的Executed\_Gtid\_Set应为c8cd8ff7-da6f-11ec-a945-00d861ef1161:1-3,f4f9537f-0c8b-11ec-a284-00d861ef116a:1-10820402。

图 6-2 设置启动位点

启动位点  ;  ?

## SQLServer 为源

SQLServer的单增量同步任务，在“设置同步”页面需要填写启动位点，任务源库日志从位点后开始获取（不含当前启动位点）。

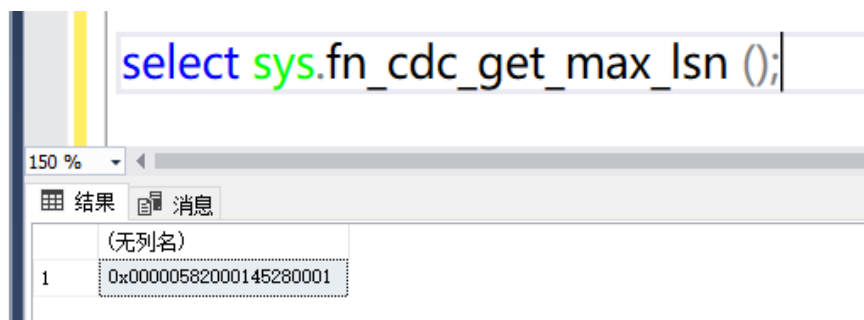
图 6-3 启动位点

启动位点

SQLServer增量数据依靠CDC官方能力，启动位点为SQLServer数据库的LSN，可通过select sys.fn\_cdc\_get\_max\_lsn(); 查询数据库的当前位点

例如，查询数据库当前LSN位点

图 6-4 查询 lsn



根据获取信息，任务的启动位点设置为如下：

图 6-5 设置启动位点



## DDS 为源

DDS到MongoDB和DDS到Kafka的同步出云任务，支持设置启动位点，在“设置同步”页面用户可以选择启动位点，同步任务从启动位点开始获取源库增量日志（含当前启动位点）。如果用户没有手动设置启动位点，DRS将会自动获取源库最新位点。

图 6-6 设置启动位点

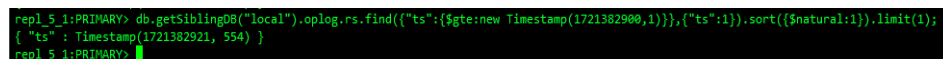


增量同步的启动位点，格式为timestamp:inre（timestamp代表unix时间戳，单位为秒；inre代表当前命令在同一秒内执行的次序）。同步任务从启动位点开始获取源库增量日志（含当前启动位点）。

- 副本集通过`db.getSiblingDB("local").oplog.rs.find({"ts":{"$gte:new Timestamp(<秒级时间戳>,1)}},{"ts":1}).sort({"natural":1}).limit(1)`命令查看源库oplog，查到的结果格式为`{"ts": Timestamp(<timestamp>, <inre> )}`，转换成timestamp:inre格式即可。
- 集群无法从mongos上查询oplog，timestamp可以根据需要启动的时间，转换成unix timestamp格式，inre为1。

例如，源库为DDS副本集，期望增量开始的秒级时间戳为1721382900（2024-07-19T17:55:00+08:00），通过上述命令去源库查询到对应的结果如下图所示。根据获取信息，任务的启动位点可以设置为：1721382921:554。

图 6-7 oplog 查询结果示例



## Oracle 为源

Oracle为的单增量同步任务，在“设置同步”页面需要填写启动位点，任务源库日志从位点后开始获取。

图 6-8 抓取启动位点

抓取启动位点

若不指定，以当前数据库位点启动(Oracle为当前数据库SCN)

增量同步的启动位点为源库SCN。

可以通过select current\_scn from v\$database; 获取数据库当前SCN

```
SQL> select current_scn from v$database;

CURRENT_SCN

22485552
```

## 6.10 MySQL 或 MariaDB 不同版本之间的语法差异

对于源数据库和目标数据库的版本有差异的迁移和同步链路，由于不同版本之间的部分特性差异，可能会出现语法兼容性问题。

DRS接收到的源数据库Binlog日志中里保存的是原始的执行SQL语句，这些SQL语句将会直接在目标数据库中执行，如果目标数据库对于源数据库中执行SQL中有不支持的语法约束，那么DRS任务将会报错并失败。

例如以下几种源数据库和目标数据库的语法不兼容的场景（非全部场景）：

- 源数据库MySQL 5.7.34版本中执行create table t1 (id int, c1 varchar(16), FULLTEXT INDEX (c1 ASC)) ENGINE = INNODB创建表结构的DDL，该DDL在源数据库Binlog保存的为以上原生SQL，增量过程中在目标数据库MySQL 8.0.27版本中执行会报错，错误信息为“1221 - Incorrect usage of spatial/fulltext/hash index and explicit index order”，错误原因为目标数据库MySQL 8.0.27版本不支持在全文索引上加排序语法规则。
- 源数据库MariaDB 10.5.8版本执行create table t1(id INT, c1 VARCHAR(32)) ENGINE = INNODB DEFAULT CHARSET = utf8mb4 COLLATE = utf8mb4\_nopad\_bin创建表结构的DDL，该DDL在源数据库Binlog保存的为以上原生SQL，增量过程中在目标数据库MySQL 8.0.27版本中执行会报错，错误信息为“1273 - Unknown collation: 'utf8mb4\_nopad\_bin'”，错误原因为目标数据库MySQL 8.0.27版本不支持utf8mb4\_nopad\_bin排序字符集。
- 源数据库MariaDB 10.5.8版本执行 create table t1(c1 INT, c2 VARCHAR(32) NOT NULL DEFAULT UUID()) ENGINE = INNODB创建表结构的DDL，该DDL在源数据库Binlog保存的为以上原生SQL，增量过程中在目标数据库MySQL 8.0.27版本中执行会报错，错误信息为“1064 - 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 UUID()”，错误原因为目标数据库MySQL 8.0.27版本不支持默认值为UUID()函数。

- 源数据库MariaDB 10.5.8版本执行 create table t1(c1 int, c2 inet6) ENGINE = INNODB 创建表结构的DDL，该DDL在源数据库Binlog保存的为以上原生SQL，增量过程中在目标数据库MySQL 8.0.27版本中执行会报错，错误信息为“1064 - 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 inet6”，错误原因为目标数据库MySQL 8.0.27版本不支持inet6类型。

## 6.11 GaussDB 远程连接配置方法

### 6.11.1 配置说明

#### 配置方式

通过修改GaussDB数据库后台pg\_hba.conf或者gs\_hba.conf文件进行配置。

#### pg\_hba.conf 或者 gs\_hba.conf 配置约束

配置文件时，请依次从上到下配置，配置的优先级由高到低。

#### pg\_hba.conf 或者 gs\_hba.conf 配置信息格式

每个白名单规则都要遵循格式：TYPE DATABASE USER ADDRESS METHOD；

- TYPE：取值local或host，local仅用于Unix域套接字连接，外部使用都配置为host。
- DATABASE：允许连接的database名称。逻辑复制连接跟普通连接有区别，逻辑复制连接时必须配置为replication。
- USER：连接用户名。
- ADDRESS：发起连接的客户端地址，可以是全放开，也可以是指定IP放开。
- METHOD：身份验证方法，取值trust或sha256。trust用于内部节点连接，比如CN与DN节点之间的连接取值trust；外部连接节点配置为sha256。

### 6.11.2 GaussDB 集中式版为源任务配置方法

GaussDB集中式版为源任务配置方法有两种，修改配置文件、通过gs\_guc修改。

#### 修改配置文件操作步骤

**步骤1** 登录GaussDB集中式版数据库所有节点。

**步骤2** 查询配置文件位置：

```
find / -name "pg_hba.conf"
或者
find / -name "gs_hba.conf"
```

**步骤3** 打开pg\_hba.conf或者gs\_hba.conf，增加配置如下。

```
host all <user> 0.0.0.0/0 sha256
host replication <user> 0.0.0.0/0 sha256
```

**步骤4** 在源数据库使用系统管理员用户执行以下语句，或重启数据库实例，使修改生效。

```
select pg_reload_conf();
```

----结束

## 通过 gs\_guc 修改操作步骤

**步骤1** 云上数据库，进入沙箱需执行以下命令；自建数据库，source gauss\_env\_file环境变量文件即可

**登录数据库实例任一节点，切换至Ruby用户，su - Ruby**

1. 2.0(V500R002C00)以前版本  

```
/usr/sbin/chroot /var/chroot /bin/bash
source /etc/profile && source ~/.bashrc
```
2. 2.0(V500R002C00)及以后版本  

```
chroot /var/chroot
source /etc/profile; source /home/Ruby/.bashrc ;source /home/Ruby/gauss_env_file
```

### 说明

华为云上GaussDB实例，需要数据库oncall同事实施变更

**步骤2** 1) 普通连接

如果放开所有用户、ip普通连接到dn节点：

```
gs_guc reload -N all -I all -h "host all all 0.0.0.0/0 sha256"
```

如果放开指定<user>、<ip>普通连接到dn节点：

```
gs_guc reload -N all -I all -h "host all <user> <ip>/32 sha256"
```

2) 逻辑复制连接

如果放开所有用户、ip复制连接到dn节点：

```
gs_guc reload -N all -I all -h "host replication all 0.0.0.0/0 sha256"
```

如果放开指定<user>、<ip>复制连接到dn节点：

```
gs_guc reload -N all -I all -h "host replication <user> <ip>/32 sha256"
```

----结束

## 6.11.3 GaussDB 分布式为源的多任务配置方法

GaussDB分布式为源的多任务配置方法有两种，修改配置文件、通过gs\_guc修改。

### 修改配置文件操作步骤

**步骤1** 登录GaussDB分布式数据库所有CN节点。

**步骤2** 查询配置文件位置：

```
find / -name "pg_hba.conf"
或者
find / -name "gs_hba.conf"
```

**步骤3** 打开pg\_hba.conf或者gs\_hba.conf，增加配置如下：

```
host all <user> 0.0.0.0/0 sha256
```

**步骤4** 登录GaussDB分布式数据库所有DN节点。

**步骤5** 执行命令查询配置文件位置。

```
find / -name "pg_hba.conf"
或者
find / -name "gs_hba.conf"
```

**步骤6** 打开pg\_hba.conf 或者gs\_hba.conf，增加配置如下：

```
host all <user> 0.0.0.0/0 sha256
host replication <user> 0.0.0.0/0 sha256
```

**步骤7** 在源数据库使用系统管理员用户执行以下语句，或重启数据库实例，使修改生效。

```
select pg_reload_conf();
```

----结束

## 通过 gs\_guc 修改操作步骤

**步骤1** 云上数据库，进入沙箱需执行以下命令；自建数据库，source gauss\_env\_file环境变量文件即可。

**登录数据库实例任一节点，切换至Ruby用户，su - Ruby**

- 2.0(V500R002C00)以前版本  

```
/usr/sbin/chroot /var/chroot /bin/bash
source /etc/profile && source ~/.bashrc
```
- 2.0(V500R002C00)及以后版本  

```
chroot /var/chroot
source /etc/profile; source /home/Ruby/.bashrc ;source /home/Ruby/gauss_env_file
```

### 说明

华为云上GaussDB实例，需要数据库oncall同事实施变更

**步骤2** 1) 普通连接

如果放开所有用户、ip普通连接到cn、dn节点：

```
gs_guc reload -Z datanode -Z coordinator -N all -l all -h "host all all 0.0.0.0/0
sha256"
```

如果放开指定<user>、<ip>普通连接到cn、dn节点：

```
gs_guc reload -Z datanode -Z coordinator -N all -l all -h "host all <user> <ip>/32
sha256"
```

2) 逻辑复制连接

如果放开所有用户、ip复制连接到dn节点：

```
gs_guc reload -Z datanode -N all -l all -h "host replication all 0.0.0.0/0 sha256"
```

如果放开指定<user>、<ip>复制连接到dn节点：

```
gs_guc reload -Z datanode -N all -l all -h "host replication <user> <ip>/32 sha256"
```

----结束

## 6.11.4 GaussDB 分布式为源的单任务配置方法

方式一：修改配置文件

### 操作步骤

**步骤1** 登录GaussDB分布式数据库所有节点。

**步骤2** 查询配置文件位置：

```
find / -name "pg_hba.conf"
或者
find / -name "gs_hba.conf"
```

**步骤3** 打开pg\_hba.conf，建议原有的sha256不动，在sha256的配置前一行配置指定单任务同步用户的trust。

- CN节点配置：

```
host replication <user> 0.0.0.0/0 sha256
```

图 6-9 CN 节点配置图

```
"local" is for Unix domain socket connections only
local all all trust
IPv4 local connections:
host all all all 127.0.0.1/32 trust
host all all 10.10.0.102/32 sha256
host all all 10.10.0.104/32 sha256
host all all 10.10.0.108/32 trust
host all all 10.10.0.134/32 sha256
host all all 10.10.0.149/32 trust
host all all 10.10.0.224/32 trust
host all all 11.10.28.145/32 trust
host all all 11.10.39.147/32 trust
host all all 11.10.48.65/32 trust
host all rdsMetric 0.0.0.0/0 reject
host all rdsRepl 0.0.0.0/0 reject
host all rdsBackup 0.0.0.0/0 reject
host all all 0.0.0.0/0 sha256
IPv6 local connections:
host all all all ::1/128 trust
host replication all 0.0.0.0/0 sha256
Disallow replication connections from localhost by a user with the
replication privilege.
#local replication rdsAdmin trust
#host replication rdsAdmin 127.0.0.1/32 trust
#host replication rdsAdmin ::1/128 trust
```

- DN节点配置：

```
host replication <user> 0.0.0.0/0 trust
```

图 6-10 DN 节点配置图

```
replication privilege.
#local replication rdsAdmin trust
#host replication rdsAdmin 127.0.0.1/32 trust
host replication hcs_auto_csn 10.10.0.0/16 trust
host replication all 10.10.0.0/16 sha256
#host replication rdsAdmin ::1/128 trust
```

### 📖 说明

DN节点配置中，建议不要将<user>指定为all，否则会影响分布式为源多任务模式下直连DN解码。

**步骤4** 在源数据库使用系统管理员用户执行以下语句，或重启数据库实例，使修改生效。



```
select pg_reload_conf());
```

----结束

方式二：gs\_guc修改

**步骤1** 云上数据库，进入沙箱需执行以下命令；自建数据库，source gauss\_env\_file环境变量文件即可

**登录数据库实例任一节点，切换至Ruby用户，su - Ruby**

1. 2.0(V500R002C00)以前版本  

```
/usr/sbin/chroot /var/chroot /bin/bash
source /etc/profile && source ~/.bashrc
```
2. 2.0(V500R002C00)及以后版本  

```
chroot /var/chroot
source /etc/profile; source /home/Ruby/.bashrc ;source /home/Ruby/gauss_env_file
```

#### 说明

华为云上GaussDB实例，需要数据库oncall同事实施变更

**步骤2** 如果用户只使用单任务模式，以单任务迁移模式用户drs1为例：

```
gs_guc reload -Z coordinator -N all -I all -h "host all drs1 0.0.0.0/0 sha256"
```

```
gs_guc reload -Z coordinator -N all -I all -h "host replication drs1 0.0.0.0/0
sha256"
```

配置cn连接dn的复制连接，以cn节点网段192.168.0.0/16为例

```
gs_guc reload -Z datanode -N all -I all -h 'host replication drs1 192.168.0.0/16
trust'
```

如果用户只使用多任务模式，以多任务迁移模式用户drs2为例：

```
gs_guc reload -Z datanode -Z coordinator -N all -I all -h "host all drs2 0.0.0.0/0
sha256"
```

```
gs_guc reload -Z datanode -N all -I all -h 'host replication drs2 0.0.0.0/0 sha256'
```

#### 说明

如果同一个源数据库实例，用户同时使用多任务模式和单任务模式，建议规划好迁移用户，按照用户进行配置。

----结束

## 6.11.5 GaussDB 分布式实例开启备机读

**步骤1** 登录GaussDB分布式实例每个数据节点的后台。

**步骤2** 执行以下命令，获取数据节点的配置文件路径。

```
find / -name "postgresql.conf"|grep dn
```

或者

```
find / -name "gaussdb.conf"|grep dn
```

**步骤3** 依次打开**步骤2**中获取的文件，在listen\_addresses='xxxx'配置下面新增listen\_address\_ext='xxxx'，其中xxxx代表监听地址，listen\_address\_ext配置的监听地址与listen\_addresses相同。

**步骤4** 重启数据库实例，使修改生效。

----结束

## 6.12 GaussDB 为源链路增量同步连接及端口说明

GaussDB为源的链路在进行增量同步时，需要保证两类连接可用：普通连接和复制连接。其中普通连接进行数据查询，复制连接进行逻辑复制。

为保证DRS能够正常连接进行同步，首先要保证数据库连接的放开，配置方式可参考[GaussDB远程连接配置方法](#)，其次要保证安全组和防火墙策略放开对应的普通连接及复制连接端口。

GaussDB集中式版的复制连接端口为普通连接端口+1。GaussDB分布式的连接分为CN连接和DN连接，对应的复制连接端口也为普通连接端口+1。

## 6.13 DRS 主键或者唯一键不一致导致数据不一致需要怎样处理

### 场景描述

MySQL到MySQL实时同步，目标数据库已经存在表结构，增量阶段发现数据不一致，同时目标数据库结构与源数据库不同。

### 可能原因

目标数据库的唯一索引或者主键与源数据库不一致，尤其是目标数据库唯一索引（主键）比源数据库对应唯一键的列少的情况下，在增量阶段，源数据库不同的两条数据写入时，目标数据库会因为唯一键冲突相互覆盖，导致目标库数据比源库数据少，数据发生不一致。

### 处理方法

**步骤1** 单击任务操作列的“编辑”按钮，在“设置同步”页面，修改同步的对象，去掉数据不一致的表后启动任务。

**步骤2** 清理目标库对应表的数据，修改表结构为兼容源库形式。

**步骤3** 再次单击任务操作列的“编辑”按钮，修改同步的对象，添加该表后启动任务，重新该表数据。

----结束

## 6.14 已经启动的任务，在配置中没有选 DDL，需要重新支持，该怎么办？

已经启动的任务，不支持修改迁移的DDL。如果需要添加或减少支持的DDL，请重新创建任务。

费用说明：

1. 包年/包月任务，不会重新收费。
2. 按需任务，是按需收费。

## 6.15 DRS 同步任务支持 MySQL8 到 MySQL5.7 吗？

DRS任务不支持MySQL8到MySQL5.7的同步。

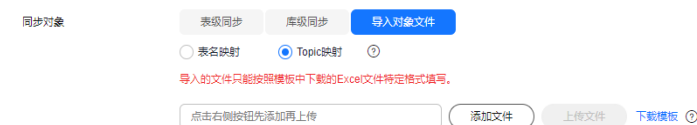
## 6.16 DRS 已经开始的同步任务如何增删表？

可以使用编辑同步对象功能。

具体参看[编辑同步对象](#)章节。

## 6.17 GaussDB 到 Kafka 导入对象文件支持表名映射和 topic 映射操作说明

图 6-11 导入对象文件



显示有表名映射和Topic映射,映射方式只支持选择一种，将以最后导入的文件为准。

- 表名映射填写示例：第一列表示要同步的表，第二列表示映射后的表，此列为空或者和两列保持一致表示不映射。库名不支持映射，schema和表名支持映射。

映射后的对象名在Kafka消息体中体现。

图 6-12 表名映射填写示例

| SourceDatabaseName. SourceSchemaName. SourceTableName | MappingName    |
|-------------------------------------------------------|----------------|
| db.sch1.t1                                            | db.sch1.t1_map |
| db.sch2.t2                                            | db.sch2_map.t2 |
| db.sch3.t3                                            |                |

- Topic映射填写示例：第一列表示要同步的表，第二列表示该表写入的Topic。此列为空表示Topic不映射，按照页面选定的Topic策略进行投递。

映射后的Topic名称，按照业务实际需要，建议提前在Kafka中创建好。

图 6-13 Topic 映射填写示例

| SourceDatabaseName. SourceSchemaName. SourceTableName | MappingName |
|-------------------------------------------------------|-------------|
| db.sch1.t1                                            | topic1      |
| db.sch2.t2                                            | TOPIC2      |
| db.sch3.t3                                            |             |

# 7 数据订阅

## 7.1 修改数据库后，到 SDK 的用户端消费一般有多长时间的延迟

如果用户消费没有延迟，1s内就可以收到这条变更。

## 7.2 订阅 SDK 获取不到数据，且程序异常

### 问题现象

创建DRS数据订阅任务，使用订阅SDK获取不到数据，且程序异常。

### 排查方法

- 订阅SDK接口排查：查看SDK接口中参数填写是否准确，详细可查看[SDK接口介绍](#)。
- 网络排查：目前数据订阅仅支持VPC网络，但是不支持VPC内的容器网络。如果客户订阅端运行在容器环境中，因DRS安全原因，未开启全网段路由，因此可能会网络不通无法获取数据。

# 8 实时灾备

## 8.1 DRS 灾备的 RPO、RTO 是什么

- RPO (Recovery Point Objective) 指当前业务数据库一个事务的提交时间, 与该事务送达DRS的时间差 (该事务通常也是DRS收到的最新的一个事务)。

是主实例与DRS实例数据差的一种度量方式, RPO=0时, 意味着业务数据库的最新数据已经全部到达DRS实例。
- RTO (Recovery Time Objective) 指当前DRS实例上事务, 传输至灾备实例且执行成功的时间差 (该事务通常也是DRS收到的最新的一个事务)。RTO是处在传输中数据量的一种度量方式, RTO=0时, 意味着DRS实例上的事务已经全部在灾备数据库上执行完毕。

图 8-1 RPO 和 RTO



## 8.2 双主灾备的主 1、主 2 如何选择

双主灾备要求灾备双方至少有一方为本云RDS实例, 另一方可以为本云RDS实例、其他云数据库、ECS自建库或本地自建数据库。为了明确本云 (本区) RDS在灾备中所扮演的角色, DRS使用主1、主2来区分角色。选择了本云RDS角色后, 也意味着确定了另一方的角色。

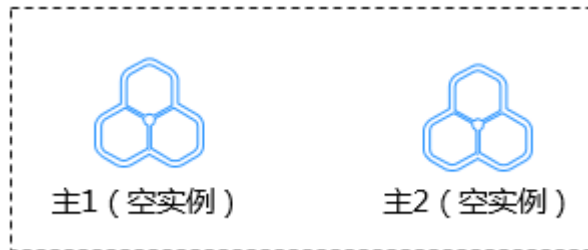
- 主1：一般具有业务数据，创建任务时选择主1则表示本云RDS具有初始数据。
- 主2：必须为空数据库，创建任务时选择主2则表示本云RDS为空库，等待接收数据。

创建灾备任务时，选择主1、主2应遵循以下原则

- 灾备双方均为本云RDS实例。
  - 有一方为空实例，空实例作为主2，非空实例作为主1。



- 双方均为空实例，主1、主2角色无差别，推荐选主2。



- 灾备双方一方为本云RDS实例，另一方为其他云数据库、ECS自建库或本地自建数据库。
  - 一方数据库存在初始数据，另一方为空库。

- 本云RDS实例为空库，则选择主2。



- 本云RDS实例有初始数据，另一方为空库，则选择主1。



- 双方均为空库，主1、主2角色无差别，推荐选主2。

## 8.3 双主灾备子任务的正向和反向怎么理解

为了明确本云（本区）RDS在灾备中所扮演的角色，在创建双主灾备任务时，DRS使用主1、主2来区分角色。主1表示选择的本云RDS具有初始数据，主2则表示选择本云RDS为空库，等待接收数据。

如图8-2，

- 选择主2，表示创建任务时选择的本云数据库实例为空库，待接收数据，创建成功后，需先配置表示入云的正向任务，待正向任务进入“灾备中”状态时，再配置并启动反向任务。
- 选择主1，表示创建任务时选择的本云数据库实例具有初始数据，待同步数据，创建成功后，需先配置表示出云的反向任务，待反向任务进入“灾备中”状态时，配置并启动正向任务。

图 8-2 双主灾备实例信息

## 8.4 实时灾备常见异常场景示例

由于一些不可控原因，实时灾备在同时操作双边数据时，存在一定的数据不一致的可能性，本章节主要介绍常见的数据异常场景。以下场景均以dr1、dr2库进行示例。

### 场景一：双主灾备，双边同时操作同一行操作，导致多数据

- 初始数据如下图所示（seqno为主键，column1为非主键）。

图 8-3 dr1、dr2 库初始数据

| seqno | column1 |
|-------|---------|
| 1     | 1       |
| 2     | 2       |
| 3     | 8       |

- 双方数据库同时分别执行如下操作：



- dr1库: update dr1 set seqno=5 where column1=8;
- dr2库: update dr2 set seqno=6 where column1=8;
- 执行操作后, 双方数据库数据一致, 并且多数据。

图 8-4 dr1、dr2 库数据

| seqno | column1 |
|-------|---------|
| 1     | 1       |
| 2     | 2       |
| 5     | 8       |
| 6     | 8       |

## 场景二：双主灾备，同时操作双边数据库，导致数据不一致

- 初始数据如下图所示 (seqno为主键, column1为非主键)。

图 8-5 初始数据

| seqno | column1 |
|-------|---------|
| 1     | 1       |
| 2     | 2       |
| 3     | 8       |

- 双方数据库同时分别执行如下操作：
  - dr1库: insert into dr1 values(101,100);
  - dr2库: insert into dr2 values(101,102);
- 执行操作后, 双方数据库数据如下图所示。

图 8-6 dr1 库数据

| seqno | column1 |
|-------|---------|
| 1     | 1       |
| 2     | 2       |
| 3     | 8       |
| 101   | 102     |

图 8-7 dr2 库数据

| seqno | column1 |
|-------|---------|
| 1     | 1       |
| 2     | 2       |
| 3     | 8       |
| 101   | 100     |

## 场景三：双主灾备，业务期执行 DDL 操作，导致数据不一致

- 双方数据库同时分别执行如下操作：
  - dr1库: truncate table dr1;
  - dr2库: insert into dr2 values(5,5,5);
- 执行操作后, 双方数据不一致, 如下图所示。

图 8-8 dr1 库数据

| seqno | column1 | column2 |
|-------|---------|---------|
| 5     | 5       | 5       |
|       |         |         |
|       |         |         |

图 8-9 dr2 库数据

| seqno | column1 | column2 |
|-------|---------|---------|
|       |         |         |
|       |         |         |
|       |         |         |

更多场景，持续补充中。

## 8.5 灾备任务的主备倒换是自动触发，还是手动触发

### 背景

实时灾备中的任务，当业务库故障后，实时灾备不会自动进行主备倒换。

原因是需要用户自行判断业务是否已经停止。

### 解决方案

需手动进行主备倒换，可参考[主备倒换任务](#)进行操作。

## 8.6 实时灾备是否支持指定库进行灾备

### 问题

实时灾备是按照实例维度进行灾备，不支持选择指定库。

### 解决方案

实时迁移和实时同步支持选择指定的表或者库。

。

## 8.7 DDM 灾备任务源库填写说明

DRS支持DDM灾备任务的源端通过输入IP方式进行连接，如[图8-10](#)。DDM灾备的原理是源端和目标端DDM中的DN一一对应灾备，因此需要正确填写源端和目标端的DN的对应关系。关于DDM逻辑库和逻辑库下的DN实例说明可参考[DDM产品文档](#)。

在创建DDM灾备任务前，建议在DDM实例目标端的控制台上，导入源端DDM的逻辑库信息，详细操作可参[“导入逻辑库”](#)。

DRS页面上“数据库实例”中按照DN对应关系填写DDM的DN信息，也就是挂载在DDM中的RDS for MySQL信息。例如源端DDM1有一个DN节点，挂载的RDS for MySQL实例1，IP为192.168.1.10；目标端DDM2中有一个DN节点，挂载RDS for MySQL实例2，IP为192.168.6.150，则填写方式如图8-10，建立DN数据灾备的对应关系。

如果DDM下有多个DN，需要用户确保每个目标端的DN都和源端DN正确对应。

图 8-10 源库信息

## 8.8 为什么实时灾备任务不支持触发器（TRIGGER）和事件（EVENT）

数据库的触发器（TRIGGER）和事件（EVENT）操作会记录binlog，DRS通过解析binlog同步数据，如果业务侧写入与触发器、事件存在同样对象，就会发生重复执行导致数据不一致或者任务失败，所以在灾备场景下，触发器和事件是不支持的。

假如源库表user存在这样触发器，当表user写入数据时，触发器会同步往另一个log表中写入一条日志数据。

当源库业务侧写入了一条数据到user表。

```
mysql> insert into user values(1, "xiaoming");
Query OK, 1 row affected (0.02 sec)
```

触发器同步了这条数据到log表，此时binlog中会有两条数据，如下图，第一条就是业务插入user表的数据，第二条是触发器被触发，写入到log中的数据。

```
binlog.000133 | 1392 | Table_map | 123453307 | 1451 | table_id: 573 (test_db.user)
binlog.000133 | 1451 | Table_map | 123453307 | 1508 | table_id: 574 (test_db.log)
```

DRS同步数据的时候可能会出现以下情况：

- 如果先同步了插入的数据到目标库user表，这时候目标数据库的触发器被触发，目标库log表会自动写入，当同步第二条log表数据时，就无法将这条log表数据写到目标库了，数据冲突任务报错。
- 如果先同步log表中的数据，再同步user表的数据时，这时候目标数据库的触发器被触发，会导致log表多一条数据，发生数据不一致问题。

同理，事件（EVENT）也会记录binlog并在目标库再次执行，这也会导致上述问题。

DRS实时迁移支持触发器和事件，是因为DRS会在结束任务的时候去迁移触发器和事件，在结束任务成功的时候保证目标库和源库的对象一致。

# 9 数据对比

## 9.1 内容对比不支持哪些数据类型

DRS提供的数据比对功能可以清晰反馈出源数据库和目标数据库的数据是否存在差异。

目前对于以下数据类型，DRS不支持内容对比，进行内容对比时会自动跳过。

表 9-1 不支持内容对比的数据类型

| 源数据库类型               | 数据类型                                                                                                          |
|----------------------|---------------------------------------------------------------------------------------------------------------|
| MySQL                | TINYBLOB、BLOB、MEDIUMBLOB、LONGBLOB、TINYTEXT、TEXT、MEDIUMTEXT、LONGTEXT                                           |
| GaussDB              | TEXT、CLOB、BLOB、BYTEA、INTERVAL DAY TO SECOND、INTERVAL                                                          |
| Oracle               | BLOB、NCLOB、CLOB、LONG RAW、LONG、INTERVAL DAY TO SECOND、INTERVAL YEAR TO MONTH、UROWID、BFILE、XMLTYPE、SDO_GEOMETRY |
| MongoDB              | _id为BINDATA类型。                                                                                                |
| Microsoft SQL Server | TEXT、NTEXT、IMAGE、BINARY、VARBINARY、HIERARCHYID、XML、TIMESTAMP                                                   |

对于以下数据类型作为主键，DRS也不支持内容对比，进行内容对比时会归到无法比对的表中。

表 9-2 不支持内容对比的主键类型

| 源数据库类型               | 数据类型                                                                                                                                                                                                                                                                                                                                                                      |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MySQL                | TINYBLOB、BLOB、MEDIUMBLOB、LONGBLOB、TINYTEXT、TEXT、MEDIUMTEXT、LONGTEXT、FLOAT、TIMESTAMP、DATE、DATETIME、BINARY、VARBINARY                                                                                                                                                                                                                                                        |
| GaussDB              | TEXT、CLOB、BLOB、BYTEA、INTERVAL DAY TO SECOND、INTERVAL、REAL、DOUBLE PRECISION、BOOL、TIME、TIMETZ、TIMESTAMP、TIMESTAMPTZ、DATE                                                                                                                                                                                                                                                    |
| Oracle               | BLOB、NCLOB、CLOB、LONG RAW、LONG、INTERVAL DAY TO SECOND、INTERVAL YEAR TO MONTH、UROWID、BFILE、XMLTYPE、SDO_GEOMETRY、BINARY_FLOAT、BINARY_DOUBLE、FLOAT、RAW、TIMESTAMP、TIMESTAMP WITH TIME ZONE、TIMESTAMP WITH LOCAL TIME ZONE、DATE                                                                                                                                                 |
| PostgreSQL           | REAL、DOUBLE PRECISION、MONEY、TEXT、BYTEA、TIMESTAMP WITHOUT TIME ZONE、TIMESTAMP WITH TIME ZONE、DATE、TIME WITHOUT TIME ZONE、TIME WITH TIME ZONE、INTERVAL、BOOLEAN、ENUMERATED TYPES、POINT、LINE、LSEG、BOX、PATH、POLYGON、CIRCLE、CIDR、INET、MACADDR、MACADDR8、BIT、BIT VARYING、TSVECTOR、TSQUERY、XML、JSON、ARRAY、COMPOSITE TYPES、INT4RANGE、INT8RANGE、NUMRANGE、TSRANGE、TSTZRANGE、DATERANGE |
| Microsoft SQL Server | FLOAT、REAL、DATE、DATETIME、DATETIME2、DATETIMEOFFSET、TIME、TIMESTAMP、TEXT、NTEXT、IMAGE、BINARY、VARBINARY、HIERARCHYID、XML、BIT                                                                                                                                                                                                                                                    |

## 9.2 DRS 对比任务对数据库有什么影响

- 对象对比：会查询源库及目标库的系统表，占用10个左右的session的连接数，正常情况下不会对数据库产生影响。但是如果对象数量巨大（比如几十万张表），可能会对数据库产生一定的查询压力。
- 行数对比：会查询源库及目标库的表行数，占用10个左右的session的连接数，正常的select count语句基本不会对数据库产生影响。但是如果表数据量巨大（亿级）会对数据库产生一定的查询压力，返回查询结果会比较慢。
- 内容对比：会查询源库及目标库的全部数据，涉及每个字段都会对比，所以对数据库产生一定的查询压力，主要体现在IO上，查询速度受限于源库和目标库的IO和网络带宽。占用1-2个CPU，占用10个左右的session的连接数。
- 用户对比：会查询源库及目标库的账户和权限，基本不会对数据库产生影响。

## 9.3 DRS 对比任务耗时预估

- 对象对比：根据源库查询性能，一般会在几分钟内返回结果，如果对象数据量特别巨大，对比时长可能达到几十分钟。
- 行数对比：使用select count方式，查询速度跟数据库性能相关。
- 内容对比：在数据库没有压力并且网络正常的情况下，对比速度大概是5M/s。
- 用户对比：与对象对比同时返回结果，如果对象不多的情况下，基本会在几分钟就会返回结果。

## 9.4 DRS 对比任务取消后，数据库有对比任务的查询进程未结束

以源库Mysql数据库为例，进行行数对比，取消对比任务，在源库有对比任务的查询进程未及时结束，清理残留进程，按照如下步骤操作：

- 步骤1** 使用管理员权限用户登录Mysql数据库，执行show processlist;
  - 步骤2** 按照Host和Command获取Id,其中Host是drs任务节点IP，Command是select count(1)开头的查询语句.
  - 步骤3** 步骤2获取到的id是112773，执行kill 112773;
  - 步骤4** 执行show processlist;再次确认残留进程是否删掉.
- 结束

# 10 通用操作

## 10.1 DRS 界面信息重叠是什么原因

### 背景

DRS界面出现信息重叠通常是页面缩放率过小导致的。

### 解决方案

建议将页面缩放率调整为100%即可显示正常。

## 10.2 目标库读写设置是实例级还是库级

配置迁移任务时，目标数据库实例可以选择设置为“只读”或者“读写”状态。

- 只读：目标数据库**整个实例**将转化为只读、不可写入的状态，迁移任务**结束后**恢复可读写状态，此选项可有效地确保数据迁移的完整性和成功率，推荐此选项。
- 读写：目标数据库可以读写，但需要避免操作或接入应用后会更改迁移中的数据（注意：无业务的程序常常也有微量的数据操作），进而形成数据冲突、任务故障、且无法修复续传，充分了解要点后可选择此选项。

只读保护优点是避免用户对正在迁移的数据库或表进行DDL或DML误操作，造成数据不一致，可提高迁移完整性和数据一致性。

- 任务启动后，DRS不支持修改目标数据库状态。
- 待所有设置该目标库为“只读”状态的迁移任务结束后，可恢复读写。

## 10.3 MySQL 源库设置了 global binlog\_format = ROW 没有立即生效

使用DRS进行MySQL的迁移或同步时，必须确保源库的binlog\_format是ROW格式的，否则就会导致任务失败甚至数据丢失。在源库设置了global级别的binlog\_format=ROW之后，还需要中断之前所有的业务连接，因为设置之前的连接使用的还是非ROW格式的binlog写入。



## 安全设置 global 级 binlog\_format=ROW 的步骤

**步骤1** 通过MySQL官方客户端或者其它工具登录源数据库。

**步骤2** 在源数据库上执行全局参数设置命令。

```
set global binlog_format = ROW;
```

**步骤3** 在源数据库上执行如下命令确认上面操作已执行成功。

```
select @@global.binlog_format;
```

**步骤4** 您可以通过如下两种方式确保修改后的源库binlog\_format格式立即生效。

### 方法一：

1. 选择一个非业务的时间段，中断当前数据库上的所有业务连接。

a. 通过如下命令查询当前数据库上的所有业务连接(所有的binlog Dump连接及当前连接除外)。

```
show processlist;
```

b. 中断上面查出的所有业务连接。

### 📖 说明

在上述操作未结束之前，请不要创建或者启动迁移任务，否则会导致数据不一致。

2. 为了避免源库binlog\_format格式因为数据库重启失效，请在源库的启动配置文件(my.ini或my.cnf等)中添加或修改配置参数binlog\_format并保存。

```
binlog_format=ROW
```

### 方法二：

1. 为了避免源库binlog\_format格式因为数据库重启失效，请在源库的启动配置文件(my.ini或my.cnf等)中添加或修改配置参数binlog\_format并保存。

```
binlog_format=ROW
```

2. 确保上述配置参数binlog\_format添加或修改成功后，选择一个非业务时间段，重启源数据库即可。

----结束

## 10.4 binlog\_row\_image 参数设置为 FULL 没有立即生效

使用DRS进行MySQL迁移时，必须确保源库的binlog\_row\_image参数设置为FULL，否则就会导致任务失败。在源库设置了binlog\_row\_image=FULL之后，只对新的session生效，为了关闭旧的session，需选择一个非业务时间段，重启源数据库并重置任务即可。

### 设置 binlog\_row\_image 为 FULL 步骤

- 如果源数据库为云上RDS实例，可通过RDS管理界面的参数配置，将binlog\_row\_image修改为FULL，完成修改后重启源数据库并重置任务即可。

### 📖 说明

云数据库RDS for MySQL 8.0版本的binlog\_row\_image参数默认值为FULL，您无需进行设置。

- 如果源数据库为本地自建库，请参考如下步骤修复。

- a. 登录MySQL源数据库所在服务器。
- b. 手动修改my.cnf配置文件，将binlog\_row\_image参数值修改为FULL后保存。  
binlog\_row\_image=full
- c. 为了关闭旧的session，需选择一个非业务时间段，重启源数据库并重置任务。

## 10.5 设置的密码不符合目标库的密码复杂度要求时，如何修改密码强度

### 操作场景

用户在设置迁移用户密码时，设置的密码不符合目标库的密码复杂度要求，需要按照用户密码复杂度的要求进行密码设置。

### 操作步骤

以下操作适用于目标数据库为RDS实例的情况。

- 步骤1 登录关系型数据库服务控制台。
- 步骤2 选择指定目标数据库实例。
- 步骤3 单击实例名称。
- 步骤4 页面跳转至“基本信息”页签，切换至“参数修改”页面。
- 步骤5 在页面右上角搜索框，输入关键字“password”，查看搜索结果。

图 10-1 修改密码



- 步骤6 在步骤5的搜索结果中，对于表1 密码参数列举的参数，需要根据密码复杂度要求进行修改，确保各参数在密码复杂度允许的范围。

表 10-1 密码参数

| 参数                                 | 允许值               | 说明                                                            |
|------------------------------------|-------------------|---------------------------------------------------------------|
| validate_password_length           | 0 ~ 2,147,483,647 | validate_password插件校验的密码的最小字符数。                               |
| validate_password_mixed_case_count | 0 ~ 2,147,483,647 | 指定当密码策略为MEDIUM（中）或更高时，为通过validate_password校验，密码至少需包含多少个大小写字符。 |

| 参数                                   | 允许值                 | 说明                                                              |
|--------------------------------------|---------------------|-----------------------------------------------------------------|
| validate_password_number_count       | 0 ~ 2,147,483,647   | 指定当密码策略为MEDIUM（中）或更高时，为通过validate_password校验，密码至少需包含多少个数字。      |
| validate_password_policy             | LOW, MEDIUM, STRONG | validate_password插件执行的密码策略。                                     |
| validate_password_special_char_count | 0 ~ 2,147,483,647   | 指定当密码策略为MEDIUM（中）或更高时，为通过validate_password校验，密码至少需包含多少个非字母数字字符。 |

**步骤7** 密码复杂度修改完成后，保存修改结果。

图 10-2 保存修改结果



**步骤8** 返回数据复制服务的“迁移模式”页面，继续执行下一步操作即可。

----结束

## 10.6 如何设置 MongoDB 数据库分片集群的分片键

MongoDB数据库中的数据的分片是以集合为基本单位的，集合中的数据通过片键被分成多部分。

对集合进行分片时，您需要选择一个片键，片键是每条记录都必须包含的，且建立了索引的单个字段或复合字段，MongoDB数据库按照片键将数据划分到不同的数据块中，并将数据块均衡地分布到所有分片中。为了按照片键划分数据块，MongoDB数据库使用基于范围的分片方式或者基于哈希的分片方式。

表 10-2 分片键分类

| 分片键类型    | 描述                                                                                                                                                        | 使用场景                                              |
|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------|
| 基于范围的分片键 | <p>基于范围的分片键是根据分片键值把数据分成一个个邻接的范围，如果没有指定特定的分片类型，则基于范围的分片键是默认的分片类型。</p> <p>特点：基于范围的分片键对于范围类型的查询比较高效，给定一个片键的范围，分发路由可以很简单地确定哪个数据块存储了请求需要的数据，并将请求转发到相应的分片中。</p> | 建议在分片键基数较大，频率较低，并且分片键值不是单调变化的情况下使用基于范围的分片键。       |
| 基于哈希的分片键 | <p>基于哈希的分片键是指MongoDB数据库计算一个字段的哈希值，并用这个哈希值来创建数据块。</p> <p>特点：保证了集群中数据的均衡。哈希值的随机性使数据随机分布在每个数据块中，因此也随机分布在不同分片中。</p>                                           | 如果分片键值的基数较大，拥有大量不一样的值，或者分片键值是单调变化的，则建议使用基于哈希的分片键。 |

集合设置分片并插入文档之后，其中的每个文档的分片的键和值都是不可更改的。如果需要修改文档的分片键，必须先删除文档，再修改分片键，然后重新插入文档。

 说明

分片键不支持数组索引，文本索引和地理空间索引。

## 基于范围的分片键设置

**步骤1** 使用如下命令，开启数据库分片开关。

```
sh.enableSharding (database)
```

 说明

参数database表示要开启分片集合的数据库。

**步骤2** 设置分片键。

```
sh.shardCollection (namespace, key)
```

 说明

- 参数namespace表示需要进行分片的目标集合的完整命名空间<database>.<collections>。
- key表示要设置分片键的索引。
- 如果需要进行分片的目标集合是空集合，可以不创建索引直接进行下一步的分片设置，该操作会自动创建索引。

```
sh.shardCollection()
```

- 如果需要进行分片的目标集合是非空集合，则需要先创建索引key。然后使用如下命令设置分片键。

```
sh.shardCollection()
```

----结束

## 基于哈希的分片键设置

**步骤1** 使用如下命令，开启数据库分片开关。

```
sh.enableSharding(database)
```

### 📖 说明

参数database表示要开启分片集合的数据库。

**步骤2** 设置基于哈希的分片键。

```
sh.shardCollection("<database>.<collection>", { <shard key> : "hashed" }, false, {numInitialChunks: 预置的chunk个数})
```

其中numInitialChunks值的估算方法是： $db.collection.stats().size / 10 * 1024 * 1024 * 1024$ 。

如果集合已经包含数据，则需要先使用如下命令对需要创建的基于哈希的分片键先创建哈希索引：

```
db.collection.createIndex()
```

然后再使用如下命令创建基于哈希的分片键：

```
sh.shardCollection()
```

----结束

## 10.7 扩大带宽是否会对 DRS 正在进行中的任务产生影响

### 背景

扩大云连接带宽时需要重建带宽链路，则会导致网络断开，此时是否会对DRS任务产生影响取决于网络断开的时间以及源库IP有没有发生变化。例如针对MySQL引擎而言，如果网络断开1天，而在这1天时间内源库binlog被清理了（MySQL都有binlog清理策略，用户侧自己配置的），就无法进行任务续传，需要重置任务。

### 解决方案

如果网络中断的时间很短，并且带宽链路更换完成后源库的VPN内的IP地址没有变，则是可以继续续传任务，不会对DRS任务产生影响。

## 10.8 为什么 MariaDB 和 SysDB 下的数据不迁移或同步

### 背景

由于某些MariaDB的版本把SysDB库作为其系统库（类似于MySQL官方版5.7的sys库），所以DRS默认也将SysDB作为所有MariaDB的系统库来处理（等同于MySQL、information\_schema、performance\_schema等库）。

## 解决方案

如果SysDB确实是业务库，您可以通过工单申请处理。

## 10.9 多对一的场景约束及操作建议

因业务需要，不同实例、不同表的数据需要进行合并时，数据复制服务提供的数据迁移和实时同步均支持多对一的场景。

多对一场景分为实例级多对一和库表级多对一。其中实例级多对一可创建多个同步任务实现；库表级多对一仅实时同步支持，通过实时同步的对象名映射功能，实现库级或者表级数据多对一同步。

## 操作建议

- 为避免创建任务过程中出现空间不足问题，建议提前计算源数据库的数据量总和，根据该总和一次性规划目标实例的磁盘空间，剩余磁盘空间需大于源库实际数据量大小的总和（例如“源系统1”数据量大小为1GB，“源系统2”数据量大小为3GB，“源系统3”数据量大小为6GB，则目标实例的剩余磁盘空间应该大于10GB）。
- 对于MySQL引擎，目标端参数的设置需要考虑整体资源的提升，建议使用第一个任务的参数对比功能中“常规参数”的“一键修改”（其中max\_connections除外），而“性能参数”应该结合目标端实际规格做相应的手工设置。
- 对于多对一同步任务场景，由于该场景是一个一个任务逐步创建的，后面创建任务时可能会造成已创建任务的同步阻塞，为了避免这个情况发生，请注意创建技巧。每个同步任务都会涉及创建索引步骤，而创建索引时数据库可能会导致Schema锁进而阻塞Schema下的其他表的数据同步，从而导致后创建的任务可能在索引创建阶段对已经同步中的任务阻塞一段时间，我们可以选择在创建同步任务最后设置为“稍后启动”，这样设定在业务低峰期后创建任务，从而避免后创建任务的索引创建对已有任务的同步阻塞。
- 对于多对一同步任务场景，不建议使用内容对比，因为内容对比无法保证一致性。
- 如果涉及表级汇集的多对一同步任务，则不支持DDL，否则会导致同步全部失败。

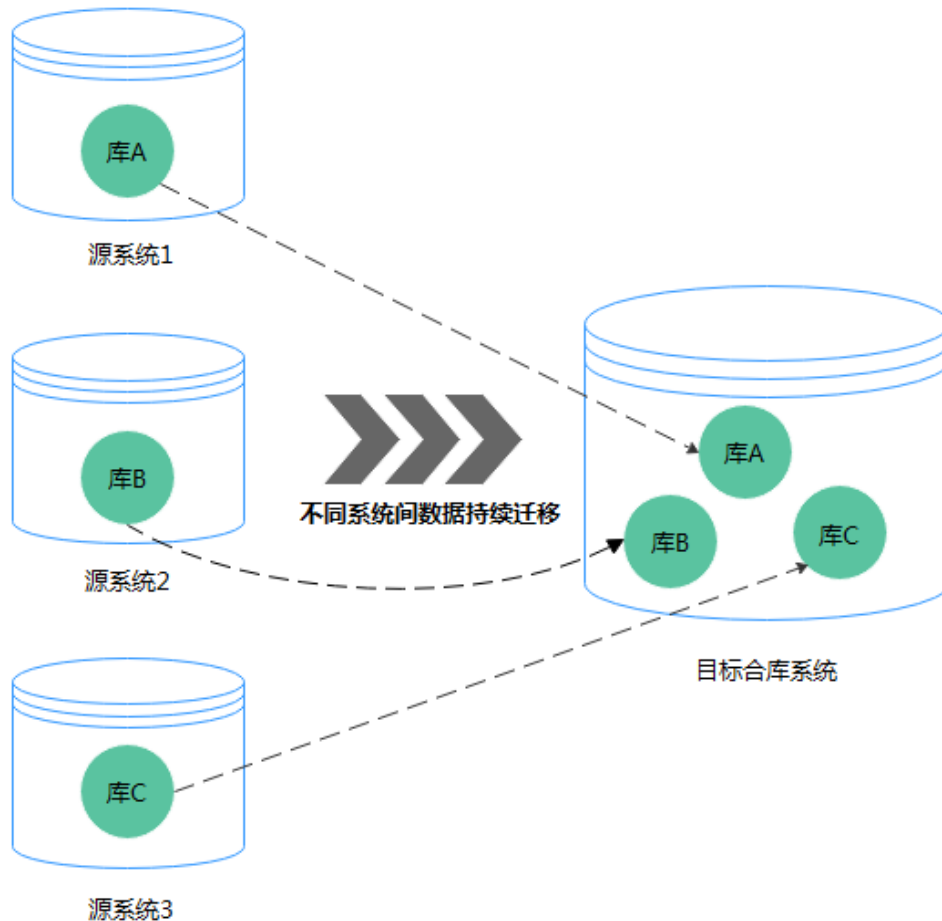
图 10-3 参数对比

| 参数名                                                     | 源库值               | 目标库值              | 对比结果 |
|---------------------------------------------------------|-------------------|-------------------|------|
| <input type="checkbox"/> character_set_client           | latin1            | latin1            | 一致   |
| <input type="checkbox"/> character_set_connection       | latin1            | latin1            | 一致   |
| <input type="checkbox"/> character_set_results          | latin1            | latin1            | 一致   |
| <input type="checkbox"/> collation_connection           | latin1_swedish_ci | latin1_swedish_ci | 一致   |
| <input type="checkbox"/> collation_server               | latin1_swedish_ci | latin1_swedish_ci | 一致   |
| <input type="checkbox"/> connect_timeout                | 10                | 10                | 一致   |
| <input checked="" type="checkbox"/> event_scheduler     | OFF               | ON                | 不一致  |
| <input type="checkbox"/> innodb_flush_log_at_trx_commit | 1                 | 1                 | 一致   |
| <input type="checkbox"/> innodb_lock_wait_timeout       | 50                | 50                | 一致   |
| <input type="checkbox"/> max_allowed_packet             | 1048576           | 1048576           | 一致   |
| <input type="checkbox"/> max_connections                | 151               | 151               | 一致   |
| <input type="checkbox"/> net_read_timeout               | 30                | 30                | 一致   |
| <input type="checkbox"/> net_write_timeout              | 60                | 60                | 一致   |
| <input type="checkbox"/> tx_isolation                   | REPEATABLE-READ   | REPEATABLE-READ   | 一致   |

## 场景一：实例级多对一数据迁移

数据迁移是以整体数据库搬迁为目的，可以实现实例级多对一迁移（如图10-4，源1、2、3中的数据到目标库的迁移），不支持源端具有同名的数据库，不支持库名映射。

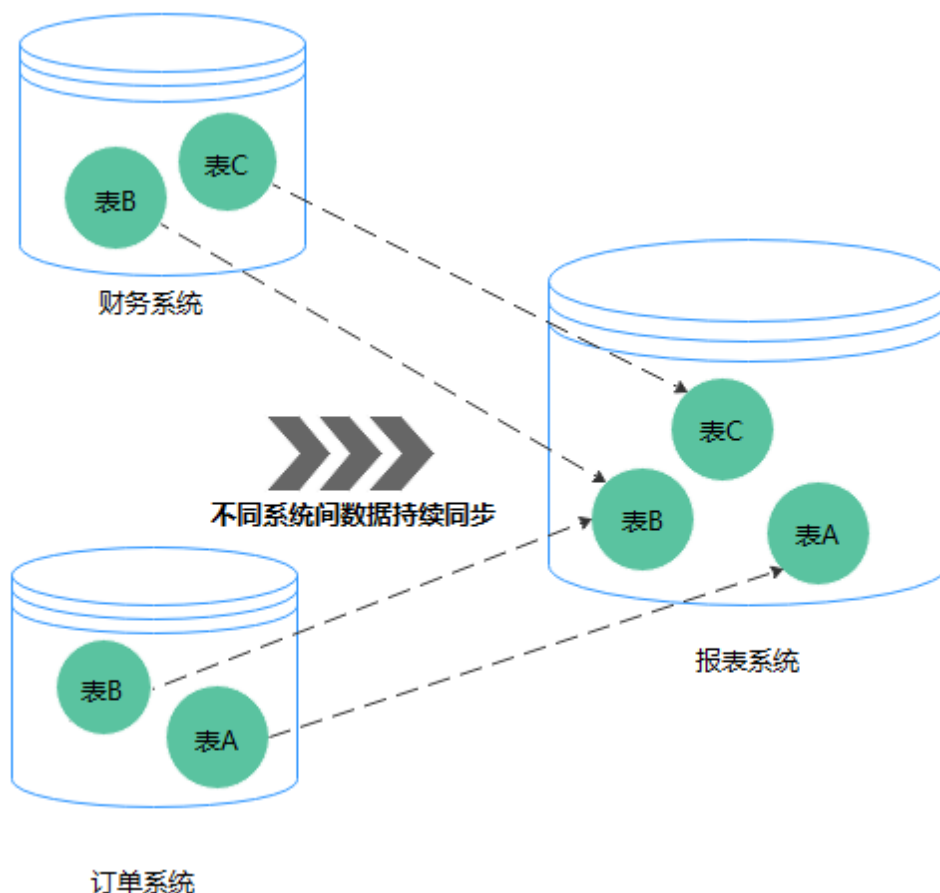
图 10-4 多对一数据迁移



## 场景二：多对一实时同步

实时同步不同于数据迁移，是维持不同业务之间的数据持续性流动，除了可以实现实例级多对一同步（如图10-5，财务系统、订单系统到报表系统的数据同步）外，还可以通过对象名映射功能，实现库级或者表级数据多对一同步（如图10-5，财务系统、订单系统中表B的数据到报表系统表B的数据同步）。

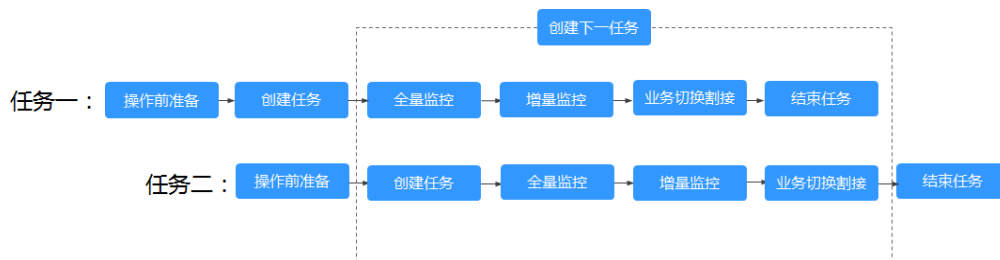
图 10-5 多对一实时同步



### 实例级多对一任务操作流程

创建任务时，为方便多对一任务间的相互识别，请在创建顺序上确保第一个任务进入全量迁移后再创建第二个任务，具体操作请参见：[方案概览](#)。

图 10-6 操作流程



### 库表级多对一任务操作流程

实时同步过程中，可以使用对象名映射功能修改库、表名称，实现库表级多对一。

以库级多对一为例：

**步骤1** 在“设定同步”页面，同步对象右侧已选对象框中，选择需要进行映射的数据库，单击“编辑”按钮。

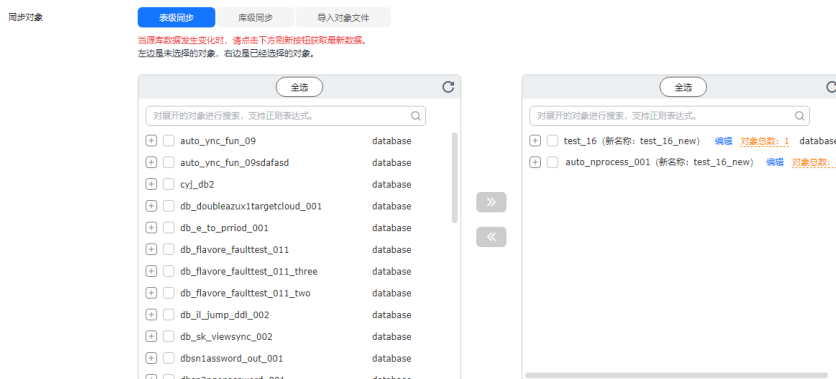


**步骤2** 在“编辑库名”的弹出框中，填写新的数据库名，修改后的名称即为保存在目标数据库中的库名。

例如，将源库中的库A和库B，修改为同样的库C后，即可实现库A、库B到库C的同步。

**步骤3** 库名修改成功后，您会看到修改前的库名和修改后的新数据库名，至此，表示完成库映射的配置。

图 10-7 查看库名修改结果



**步骤4** 配置完成后，等待任务启动并完成同步。

----结束

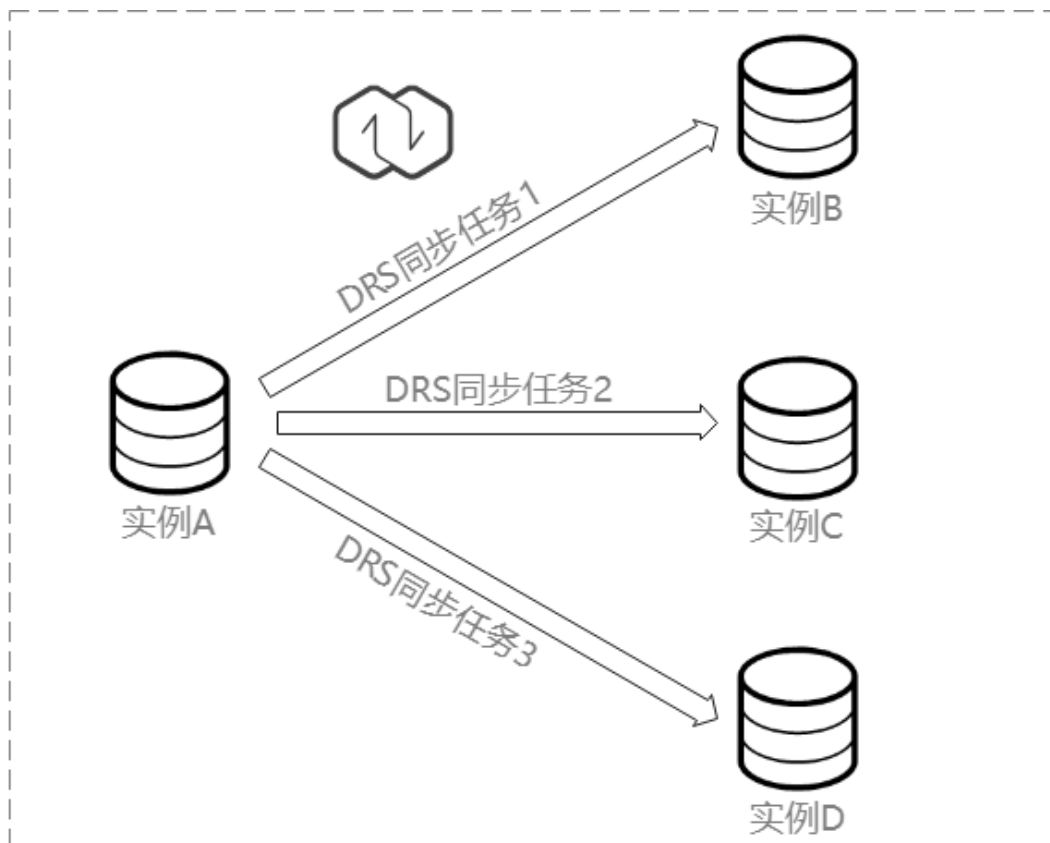
## 10.10 一对多的场景约束及操作建议

因业务需要，存在同一实例或同一对象的数据需要同步到多个实例或多个对象的场景，数据复制服务提供的数据库迁移和实时同步均支持一对多的场景。

### 操作建议

- 由于一对多场景需要多个DRS任务来配合完成，所以会对源库产生的压力也是成倍数的增加，所以为避免多个DRS对同一个源库造成的压力影响到源库业务，需要客户提前对源库的负载情况进行分析。如果源库负载已经很沉重，建议减少同步任务。
- 建议该场景一个一个任务逐步创建。

## 实例级一对多实时同步

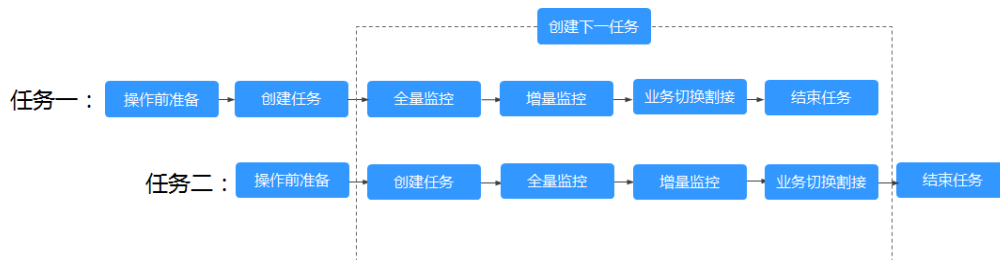


用户需要创建多个同步任务来实现一对多的实时同步。例如从实例A同步到实例B、实例C和实例D，需要创建三个同步任务。

## 操作流程

创建任务时，为方便一对多任务间的相互识别，请在创建顺序上确保第一个任务进入全量阶段后再创建第二个任务，具体操作请参见：[方案概览](#)。

图 10-8 操作流程



## 10.11 数据复制服务的操作日志在哪里查看

- 数据复制服务的操作日志属于操作审计类日志。
- 用户可以登录到云审计服务（Cloud Trace Service，简称CTS）页面，查看当前用户在Console页面单击的页面操作，主要是涉及任务变更的管理类操作。

- 请单击界面右上角的用户名，在下拉菜单选择“操作日志”进行查看。

## 10.12 为什么 DRS 发送提示信息的语言和页面不符

DRS发送邮件和短信的语言是根据用户首选项中的设置决定的。登录控制台页面，单击右上角账号进入“账号中心”页面，单击“首选项设置”选择语言和时区。

图 10-9 首选项设置

首选项设置

消息接收设置

1. 系统默认在08:00-18:00发送的消息通知会根据您设置的时区对应时间发送；  
2. 您收到的短信、邮件中的时间会根据您设置的时区展示。

语言  设置您的邮件/短信接收语言。

时区  设置您所在的时区。

## 10.13 为什么 DRS 任务自动结束了

DRS为避免不必要的计费情况发生，支持按需计费任务在创建任务阶段设置“任务异常自动结束时间（天）”，输入值必须在14到100之间。设置任务异常自动结束天数后，异常且超时的任务将会自动结束，以免产生不必要的费用。

DRS包年/包月任务不支持设置任务异常自动结束时间。

图 10-10 设置任务异常自动结束时间

启动时间   ⓘ

全量阶段DRS会占用一些性能资源，为保证业务和任务正常运行，强烈建议在业务低峰期启动，同时预留2-3天时间校对数据。

任务异常通知和设置  ⓘ

\* SMN主题  ⓘ

时延阈值(s)  ⓘ

数据异常通知

\* 任务异常自动结束时间  ⓘ 任务处于异常状态一段时间后，将会自动结束。单位为天。

## 10.14 DRS 导出任务列表

DRS支持查询和导出详细的任务信息列表，包括任务名称、ID、状态、网络类型、源库和目标库IP地址或域名、端口、告警配置等信息，方便客户从不同维度查看和管理任务。在任务管理界面，选择查询条件，单击“导出”按钮，导出当前的查询结果。

图 10-11 导出任务查询结果



## 10.15 已结束的任务还能重新启动吗

### 背景

- DRS任务已经结束。
- 需要重新启动。

### 解答

DRS已结束任务无法重新启动。

针对已结束但是没有删除的同步任务，用户可通过[克隆功能](#)，快速复制现有同步任务的配置，创建新任务。

## 10.16 重置任务和重新创建任务有什么区别

重置功能一般在任务暂停和失败场景使用，DRS重置功能不会清空目标库，客户需要根据自己的需求选择是否清空目标库。

任务重置后会重新进行全量同步，不需要再次配置任务。

更多重置功能详细说明可参考[这里](#)。

## 10.17 DRS 支持一键反向功能吗

DRS实时迁移和实时同步暂不支持切换反向任务。

单主灾备任务可通过[主备倒换任务](#)功能实现。

双主灾备不同于单主灾备，分为正向和反向灾备任务，无需进行主备倒换。

## 10.18 已经创建的 SMN 主题为什么选择不了

您可能没有请求订阅主题。在创建主题后，还需要请求订阅该主题，添加订阅后，消息通知服务会向订阅终端发送订阅确认信息，信息中包含订阅确认的链接。订阅确认

的链接在48小时内有效，用户需要及时在手机端、邮箱或其他协议终端确认订阅。具体操作可参见[请求订阅](#)。

## 10.19 任务创建成功后是否支持修改 SMN 主题

支持修改，单击任务名称进入“基本信息”页签，在“任务异常通知设置”模块下，可对SMN主题进行修改。

图 10-12 修改 SMN 主题



## 10.20 DDM 为源链路源库分片数量和信息怎么填写

DRS支持DDM为源的迁移和同步，在创建任务时，需要先按照源端DDM绑定的RDS实例个数选择分片数量，以实时同步为例，如[图10-13](#)。例如源端DDM有两个DN节点，挂载的RDS for MySQL实例1和实例2，则“源端实例数据库实例个数”设置为“2”。关于DDM逻辑库和逻辑库下的DN实例说明可参考[DDM产品文档](#)。

图 10-13 同步实例信息



任务创建完成后，需要在“测试连接”页面填写源库信息，如[图10-14](#)，假设源端DDM实例IP地址为192.168.5.11，挂载的RDS for MySQL实例1，其IP地址为192.168.1.10；挂载的RDS for MySQL实例2，其IP地址为192.168.6.150，则填写方式如[表10-3](#)。

图 10-14 源库信息

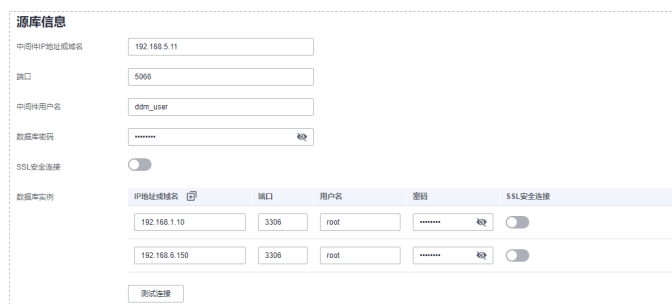


表 10-3 源库信息

| 参数         | 描述                                                                            |
|------------|-------------------------------------------------------------------------------|
| 中间件IP地址或域名 | 填写源数据库DDM中间件的IP地址或域名，本示例中为192.168.5.11。                                       |
| 端口         | 源数据库DDM中间件服务端口，可输入范围为1~65535间的整数，一般DDM端口默认为5066，请根据实际情况填写。                    |
| 中间件用户名     | 源数据库DDM中间件的用户名，可参考《DDM快速入门》。                                                  |
| 数据库密码      | 源数据库DDM中间件用户名所对应的密码。                                                          |
| SSL安全连接    | 通过该功能，用户可以选择是否开启对迁移链路的加密。如果开启该功能，需要用户上传SSL CA根证书。                             |
| 数据库实例      | 根据DDM实际的DN分片数据库，填写对应的数据库信息。<br>本示例中两个分片数据库IP地址分别填写192.168.1.10和192.168.6.150。 |

## 10.21 源端 MySQL 发生主备倒换，DRS 任务会丢数据吗

如果源端MySQL已开启了GTID模式，在发生主备倒换时，DRS会记录binlog位点信息，在主备倒换后接续位点继续进行任务，不会丢数据。

源端MySQL可通过修改数据库配置文件方式将GTID开启，重启数据库后生效。

参考命令如下：

```
gtid_mode = on
log_slave_updates = true
enforce_gtid_consistency = on
```

## 10.22 DRS 对象选择时的全部迁移、表级迁移、库级迁移有什么区别

目前DRS实时迁移和实时同步任务支持对象选择，实时灾备是按照实例维度进行灾备，不支持选择指定库或表。

实时迁移任务支持“全部迁移”、“表级迁移”或者“库级迁移”。

实时同步任务支持“表级同步”或者“库级同步”。

- 全部迁移：仅迁移任务支持选择，选择后源数据库中的所有对象全部迁移至目标数据库，对象迁移到目标数据库实例后，对象名将会保持与源数据库实例对象名一致且无法修改。
- 表级迁移或同步：以表为单位的迁移或同步，库中新增表不会自动加入迁移或同步关系中。如果同步任务，可以通过再次编辑同步对象，将新的表追加到同步关系中；迁移任务启动后不支持编辑迁移对象。
- 库级迁移或同步：以库为单位的迁移或同步，不可单独选择表，且已选库中新增的表会自动加入迁移或同步关系之中。

## 10.23 源库或目标库修改密码后如何操作

DRS任务进行过程中，可能会因为源数据库或者目标数据库修改密码信息，导致连接失败，此时需要通过数据复制服务控制台更新为正确的信息，然后续传任务。

### 操作步骤

- 步骤1** 在任务列表选中指定任务，单击任务名称。
  - 步骤2** 进入“基本信息”页签，在“连接信息”模块下，单击“修改连接信息”。
  - 步骤3** 在“修改连接信息”弹出框中对源库和目标库的密码进行更新，更新完成后，单击“确认”即可。
  - 步骤4** 更新完成后，可在“基本信息”页签下查看修改结果。
- 结束

## 10.24 设置 SMN 主题策略，允许 DRS 发送消息通知

该章节指导用户通过手动设置SMN主题策略的方式，授权DRS服务发送消息通知。

### 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** 单击“服务列表”，选择“管理与监管 > 消息通知服务”。  
进入消息通知服务页面。
- 步骤4** 在左侧导航栏，选择“主题管理 > 主题”。  
进入主题页面。
- 步骤5** 在主题列表中，选择一个主题，单击右侧“操作”栏的“更多”，在下拉列表中选择“设置主题策略”。
- 步骤6** 在弹出的对话框中，设置允许DRS服务发布消息，如[图10-15](#)。

图 10-15 设置主题策略

设置主题策略

主题名称

访问策略 ? 基本模式

可发布消息的用户

仅自己(主题创建者)

所有人

仅如下用户

[了解如何获取账号ID 点击这里。](#)

可发布消息的服务

CAD  OBS  DWS  apm  VOD  Moderation

CIE  MPC  DRS  LTS  CTS

步骤7 设置完成后，单击“确定”。

----结束

## 10.25 DRS 任务表数量太多导致任务超时怎么办

当DRS任务表数量太多时，可能占用过多的内存等资源，导致在进行预检查、数据对比等操作时超时。针对这种情况，建议用户通过以下方式解决：

- 创建任务时选择大规格，表数量建议不超过5W张。
- 如果表数量过多（超过5W张），建议拆分多个任务进行同步。

## 10.26 DRS 任务创建后支持修改源或目标数据库吗

DRS在任务创建后、启动前的配置状态时，支持修改源或者目标数据库。

- 入云链路：支持修改源数据库，不支持修改目标数据库。目标数据库为创建任务时选择的实例，不支持更换，仅支持修改数据库连接信息（用户名和密码）。
- 出云链路：支持修改目标数据库信息，不支持修改源数据库。源数据库为创建任务时选择的实例，不支持更换，仅支持修改数据库连接信息（用户名和密码）。
- 自建-自建链路：支持修改源和目标数据库信息。

DRS在任务提交启动后，源和目标数据库均不支持更换，仅支持修改数据库连接密码。

## 10.27 Api V5 公测状态，是可用的吗？

DRS api v5接口是可用的。



## 10.28 DRS 进行暂停，结束任务，续传任务操作，需要多少时间？

- 暂停任务：10秒以内。
- 结束任务：不同链路在结束任务阶段需要做的操作不同，一般不会超过1分钟。
- 续传任务：10秒以内完成进程重启，如果任务在增量阶段，重启后会在一段时间内任务状态为“启动中”，这个时间取决于续传时的延迟时间，延迟追上之后任务状态会变成“增量同步”。

# 11 计费相关

## 11.1 失败任务计费说明

目前DRS对客户创建的计费任务，从任务启动开始进行计费，任务结束终止收费，在此期间的失败场景（如全量失败、增量失败等）均不会停止收费。为避免不必要的计费情况发生，需要客户在创建任务阶段设置“任务异常自动结束时间（天）”，输入值必须在14到100之间。设置任务异常自动结束天数后，异常且超时的任务将会自动结束，以免产生不必要的费用。

## 11.2 暂停任务计费说明

数据复制服务支持“按需计费”和“包年/包月”两种计费模式，两种方式均针对DRS实例从以下方面收取费用：

表 11-1 计费项

| 计费项          | 计费项说明                                       | 计费规则                                                                                                                           |
|--------------|---------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|
| 配置费（必选）      | 指计算和存储资源及数据处理的费用。                           | <ul style="list-style-type: none"><li>“按需计费”配置费用是按照实际使用时长计费，以自然小时为单位整点计费，不足一小时按照实际使用时长计费。</li><li>“包年/包月”配置费用是一次性计费。</li></ul> |
| 弹性公网IP费用（可选） | 指公网访问产生的数据处理和数据流量费用。如果创建公网网络任务，您需要购买弹性公网IP。 | 详细说明请参考 <a href="#">弹性公网IP计费说明</a> 。                                                                                           |

所以已暂停的“按需计费”任务仍然会收取配置费用。

## 11.3 未启动任务计费说明

DRS任务的启动时间可以根据业务需求，设置为“立即启动”或“稍后启动”。DRS任务是从启动开始计费，当用户选择“稍后启动”时，在设置的启动时间之前DRS不会收取费用。

## 11.4 包年包月到期说明

“包年/包月”任务到期后，如果没有及时的进行续费或充值，将进入宽限期。如宽限期满仍未续费或充值，将进入保留期。在保留期内，无法在DRS管理控制台进行该任务的操作，相关接口也无法调用，自动化监控或告警等运维也会停止。如果在保留期结束时您没有续费，实例将终止服务，系统中的数据也将被永久删除。

关于宽限期和保留期的时长限制说明，可参考[这里](#)。若您购买的任务已到期，请参见费用中心用户指南中[资源停止服务或逾期释放说明](#)章节进行处理。

## 11.5 任务结束后不删除还会计费吗

DRS任务结束后，不删除列表中的任务，不会产生费用。已结束的任务无法重新启动。

## 11.6 资源冻结/释放/删除/退订

### DRS 资源为什么被释放了？

客户在华为云购买产品后，如果没有及时的进行续费或充值，将进入宽限期。如宽限期满仍未续费或充值，将进入保留期。在保留期内资源将停止服务。保留期满仍未续费或充值，存储在云服务中的数据将被删除、云服务资源将被释放。请参见[资源停止服务或逾期释放说明](#)。

### DRS 资源为什么被冻结了？

资源冻结的类型有多种，最常见类型为欠费冻结。

单击了解[资源冻结的类型、冻结后对续费、退订的影响](#)。

### 怎样将资源解冻？

欠费冻结：用户可通过续费或充值来解冻资源，恢复DRS正常使用。欠费冻结的DRS允许续费、释放 或删除；已经到期的包周期DRS不能发起退订，未到期的包周期DRS可以退订。

### 冻结、解冻、释放资源时对业务的影响

- 资源冻结时：
  - 资源将被限制访问和使用，会导致您的业务中断。例如DRS被冻结时，会使得用户无法再迁移数据。
  - 包周期资源被冻结后，将被限制进行变更操作。

- 资源被冻结后，可以手动进行退订/删除。
- 资源解冻时：资源将被解除限制，用户可以迁移数据。
- 资源释放时：资源将被释放，实例将被删除。

## 怎样续费？

包年/包月方式购买的DRS到期后，请在管理控制台[续费管理](#)页面进行续费操作。详细操作请参考[续费管理](#)。

## 资源被释放了能否恢复？/退订错了可以找回吗？

实例被删除，无法找回。

退订资源前请一定要仔细确认资源信息。如果退订错了建议重新购买使用。

## 怎样删除 DRS 实例？

- DRS按需实例需要先结束才能删除，以实时迁移为例，详细操作请参考[删除迁移任务](#)。
- DRS包周期实例需要先退订才能删除，以实时同步为例，详细操作请参考[退订包周期任务](#)。

# 12 时延相关

## 12.1 灾备任务时延增高常见原因

### RTO 增大常见原因

RTO是增量同步期间DRS实例上的事务传输到灾备目标库且回放成功的时间差，RTO值较大说明DRS上需要回放到目标实例的事务有积压。一般有以下几种原因：

1. 灾备任务初始化完成不久，启动灾备任务到当前时间累积的增量数据需要回放。
2. 业务数据库对无主键表进行了批量操作，DRS灾备实例正在同步变更数据较大的无主键表。为了确保无主键表数据一致性，对所有执行操作都进行位点记录，因此效率相比有主键表低。同时，如果目标表无索引，数据更新效率会更低。
3. 业务数据库执行了DDL操作，DRS灾备实例需要等待DDL在灾备目标实例执行完成后，再执行数据回放。
4. 业务数据库对热点表进行频繁操作。DRS灾备实例会对热点表的事务进行合并后再进行回放，减少频繁对目标库操作。
5. 灾备数据库访问异常导致增量数据无法回放成功。

### RTO 增大处理建议

**步骤1** 在“实时灾备管理”页面，选择指定的灾备任务，单击任务名称进入“基本信息”页签。

**步骤2** 在“基本信息”页签，单击“灾备监控”页签，查看时延监控的RTO的变化情况。

- 如果RTO逐渐降低或者只是短时间内增加，则无需关注。
- 如果RTO持续增加，可在灾备数据库执行以下语句，查看是否有执行时间比较长的sql或者正在执行的DDL。  

```
show processlist
```
- 如果灾备数据库异常，需要联系数据库运维工程师解决。

----结束

## RPO 增大常见原因

RPO是增量同步期间业务数据库一个事务的提交时间与送达DRS实例的时间差，RPO值较大，说明业务数据库最新的变更数据还没有抽取到DRS灾备实例。一般有以下几种原因：

1. 业务数据库和DRS灾备实例之间的网络不稳定，读取业务数据库端的变更日志比较慢。
2. 业务数据库访问异常导致增量数据无法抽取成功。

## RPO 增大处理建议

**步骤1** 在“实时灾备管理”页面，选择指定的灾备任务，单击任务名称进入“基本信息”页签。

**步骤2** 在“基本信息”页签，单击“灾备监控”页签，查看时延监控的RPO的变化情况。

- 如果RPO逐渐降低或者只是短时间内增加，则无需关注。
- 如果业务数据库异常，需要联系数据库运维工程师解决。

----结束

## 12.2 MongoDB 链路时延增高可能原因

### 涉及链路

- MongoDB->DDS入云迁移
- DDS->MongoDB出云迁移
- DDS->MongoDB出云同步

### 可能原因

DRS在任务增量阶段，为保证迁移/同步/灾备的性能，会集合级的进行并发回放。如果出现以下特殊情况，DRS只支持单线程写入，不支持并发回放。

- 集合的索引中有unique key时；
- 集合属性的capped为true时；

若出现时延增高的这种情况，用户可排查是否以上原因导致。

## 12.3 DRS 全量阶段迁移过慢或者进度不更新可能原因

### 问题现象

DRS在进行全量迁移时，出现迁移耗时很久或者迁移任务进度不更新的情况。

### 排查方法

- 排查源库迁移数据大小：  
数据迁移进度是按表数量来计算百分比，当数据迁移进度长时间不更新，可能因为某几个表数据量过大，迁移时间较长。可以在“进度明细”下，单击目标迁移对象“详细信息”列的“查看详情”，查看具体对象的迁移进度。

- 排查源库表主键和索引情况：  
源库大表缺少主键和非空唯一索引，以MySQL为例，可以在源库执行show create table <数据库名>.<表名>查询是否有主键或非空唯一索引。
- 排查源库是否有结束长连接现象：  
源库为其他云上数据库时，可能存在自动结束长连接的现象，导致全量迁移数据过慢或者进度不更新的情况。
- 排查目标库索引迁移：  
索引迁移进度长时间不更新，可能是某些大表的索引创建比较慢，目标库一直在创建索引中。可以登录目标库，执行show processlist查看DRS在目标库的执行情况。
- 排查目标库是否有死锁：  
目标库如果有死锁，可能导致全量数据无法写入。以MySQL为例，可执行以下命令查看、删除死锁。
  - 查看是否锁表：show OPEN TABLES where In\_use > 0;
  - 查看锁表进程：show processlist;
  - 删除锁表的进程：KILL [CONNECTION | QUERY] <thread\_id>。
- 排查源库和目标库网络连接：  
检查网络的连通性是否正常，网络带宽是否存在限制。可通过使用ping等OS命令测试网络的连通性及网络延迟。
- 检查DRS任务是否设置限速：  
可单击任务名称，在“基本信息”页签的“限速信息”区域查看是否进行了全量限速。

## 12.4 DRS 增量阶段时延增高可能原因

### 问题现象

DRS在进行增量迁移或同步时，任务出现延迟高的现象。

### 可能原因

- 原因1：全量阶段刚结束，增量迁移延迟高。全量迁移过程中，源库增量数据还在不断地写入，DRS需要在全量结束之后，将这部分增量数据同步到目标端去，因此时延较高。
- 原因2：源库有大批量数据变更(导入、更新、删除)。大事务写入在源库执行时间本来就很长，而DRS把数据同步到目标库也需要一段时间，所以此时延迟会慢慢增加，如果此大事务的表没有主键或者索引，恢复时间会相应拉长。
- 原因3：源库存在大批量DDL操作，DDL执行比较慢，同步到目标导致时延增高。
- 原因4：DRS任务规格限制。DRS不同规格对应的性能上限不同，详细可参考[规格说明](#)，当源库数据写入量过大达到瓶颈时，就会导致任务出现延迟。
- 原因5：目标库规格受限，达到写入瓶颈。以目标实例为RDS for MySQL为例，用户可以在RDS控制台查看数据库性能指标。
- 原因6：可能存在热点更新。无主键表的写入会导致热点更新，源库对单一表或单一行的高频更新也会导致热点更新，从而导致时延增大。以RDS for MySQL为例，可以通过RDS的审计日志进行确认，详情可参考[RDS审计日志](#)。

- 原因7：网络不稳定导致的时延增大。

## 解决方法

- 解决方法1：针对这种情况，DRS会自动追平时延，无需处理，用户可观察增量时延是否降低。
- 解决方法2：存在大事务写入的情况时，等待更新完毕，或者从业务上考虑尽量避免大事务写入的情况。用户可查看源库执行历史，看是否有大事务的写入，同时可以在目标库查看DRS数据回放情况。以MySQL为例，可执行show processlist查看。
- 解决方法3：源库应尽量避免执行批量DDL，如果必须执行DDL，请尽量在业务低峰期执行。
- 解决方法4：重新创建任务，选择较大规格提升同步性能（目前，DRS同步仅支持任务类型为单AZ的同步任务升级实例规格，不支持降低规格，具体操作及约束限制可参考[同步规格变更](#)）。
- 解决方法5：升级目标库的实例规格，提高目标库实例写入性能。以目标实例为RDS for MySQL为例，参考[变更规格](#)，升级实例配置。
- 解决方法6：如果存在热点更新，需要等待热点更新完毕，或者从业务上尽量避免热点更新。
- 解决方法7：网络延迟无法避免，用户可以通过专线网络接入源库和目标库实例减少延迟。