数据传输服务（DTS）

产品文档

目录

数据传输服务（DTS） 4
- 产品简介 4
  - 产品概述 4
  - 产品优势 5
  - 应用场景 6
- 操作指南 7
  - 准备工作 7
    - 网络准备 7
      - 网络准备概述 7
      - 本地 IDC 与云平台的互通 10
        - 专线接入 10
      - 其他云厂商与云平台的互通 13
      - 添加 DTS IP 地址至对接数据库白名单 14
        - 单个任务，放通 DTS 访问 IP 14
    - 数据库及权限准备 16
    - 配置自建 MySQL 系的 Binlog 17
  - 数据迁移 19
    - 数据迁移支持的数据库 19
    - 云数据库跨账号实例间迁移 23
    - 迁移至 MySQL 系列 33
      - 迁移至 MariaDB 33
        - MySQL/MariaDB/Percona 迁移至 MariaDB 33
        - TDSQL MySQL 迁移至 MariaDB 41
      - 迁移至 TDSQL MySQL 42
        - TDSQL MySQL 迁移至 TDSQL MySQL 42
        - MySQL/MariaDB/Percona 迁移至 TDSQL MySQL 47
      - 迁移至 Percona 48
        - Percona 迁移至 Percona 48
      - MySQL 系列前置校验不通过处理 49
        - 连接 DB 检查 49
        - 周边检查 50
        - 版本检查 52
        - 源实例权限检查 53
        - 部分实例参数检查 54
        - 目标实例权限检查 58
        - 目标实例内容冲突检查 59
        - 目标实例空间检查 61
        - Binlog 参数检查 62
        - 外键依赖检查 70
        - 视图检查 74
        - 高级对象检查 75
        - 警告项检查 77
        - TDSQL Proxy 检查 79
        - TDSQL 内核检查 80
        - 同库表对象 DDL 环形链路同步检测 81
        - 同目标同库表对象 DDL 同步冲突检测 82
    - 迁移至 MongoDB 83
      - MongoDB 迁移至 MongoDB 83
        - 支持能力 83
        - 使用说明 85
        - 迁移操作指导 87
        - 前置校验不通过处理 93
          - 连接 MongoDB 实例校验 93
          - 库表冲突校验 94
          - 源端节点角色校验 95
          - Oplog 校验 96
          - 源端或目标端账户权限校验 97
          - 实例版本校验 98
          - 实例容量校验 99
          - 目的端负载校验 100
          - 片键校验 101
          - 源端 Balancer 校验 102
          - 时序集合校验 103
          - 压缩算法校验 104
    - 迁移至 Redis® 105
      - Redis® 迁移至 Redis® 105
        - 支持能力 105
        - 使用说明 107
        - 迁移操作指导 109
        - Key 前缀过滤 115
        - 前置校验项不通过处理 117
          - 网络是否可达 117
          - 源实例目标实例版本是否兼容 118
          - 源实例参数检查 119
          - 目标实例容量是否满足要求 120
          - 目标实例状态是否正常 121
          - 源实例是否为从机 122
        - 常见错误处理 123
  - 数据同步 132
    - 数据同步支持的数据库 132
    - 云数据库跨账号实例间同步 136
    - 同步至 MySQL 146
      - MySQL/MariaDB/Percona 同步至 MySQL 146
      - TDSQL MySQL 同步至 MySQL 155
    - 同步至 MariaDB 156
      - MySQL/MariaDB/Percona 同步至 MariaDB 156
      - TDSQL MySQL 同步至 MariaDB 157
    - 同步至 TDSQL MySQL 158
      - TDSQL MySQL 同步至 TDSQL MySQL 158

- 使用说明 158
- 同步操作指导 160
- MySQL/MariaDB/Percona 同步至TDSQL MySQL 165
- MySQL 系列前置校验项 166
- 连接 DB 检查 166
- 周边检查 167
- 版本检查 169
- 源实例权限检查 170
- 部分实例参数检查 171
- 目标实例权限检查 175
- 目标实例内容冲突检查 176
- 目标实例空间检查 178
- Binlog 参数检查 179
- 外键依赖检查 187
- 视图检查 191
- 高级对象检查 192
- 警告项检查 194
- TDSQL Proxy 检查 196
- TDSQL 内核检查 197
- 同库表对象 DDL 环形链路同步检测 198
- 同目标同库表对象 DDL 同步冲突检测 199
- 前置校验项不通过处理 200
- 割接说明 201
- 监控与告警 203
- 支持的监控指标 203
- 支持的事件 210
- 配置指标告警和事件告警 211
- 配置数据迁移指标告警 211
- 配置数据同步指标告警 213
- 配置事件告警推送 215
- 查看告警历史 222
- 查看监控指标 223
- 常见问题 226
- 数据迁移 226
- 数据同步 231
- 错误处理 232
- 常见错误处理 232
- 迁移慢或者进度卡住 237
- 数据同步有延时 239
- 词汇表

产品简介
产品概述

概述

数据传输服务（Data Transfer Service，DTS）支持 MySQL、MariaDB、Percona、TDSQL MySQL、Redis®、MongoDB 多种数据库类型，可帮助用户在业务不停服的前提下轻松完成数据库迁移上云，利用实时同步通道轻松构建高可用的数据库容灾架构等场景需求。

产品功能

- 数据迁移
- 数据迁移功能指不同数据源之间的数据复制，DTS 支持数据库不停机迁移，最小化迁移过程引起的数据库停机对业务造成的影响。支持将外网自建数据库、CVM 自建数据库迁移到云平台。
- 数据同步
- 数据同步功能指两个数据源之间的数据实时同步。 数据同步不同于数据迁移，数据迁移是以整体数据库搬迁为目的，它属于一次性的短期任务，迁移完成后用户一般需要进行手动割接操作，将业务对接到新的数据库上。数据同步适合非一次性的任务，可支持在源和目标端同时进行数据写入，增加了数据冲突处理、任务配置动态调整等功能，更方便用户对任务的管理。

产品优势

数据传输服务 DTS 支持关系型数据库 RDBMS、非关系型数据库 NoSQL 数据源间的数据传输。它提供了数据迁移、实时数据同步等多种数据传输方式。相对于第三方数据流工具，数据传输服务 DTS 提供更丰富多样、高性能、高安全创新的传输链路，同时提供了诸多便利功能，极大地方便了传输链路的创建及管理。

数据同步

DTS 帮助您将数据库迁移至云平台，且几乎不需要停机拷贝。源数据库在迁移期间发生的所有数据更改都会复制到目标数据库，因此迁移期间不影响源数据库对外提供服务。在数据复制完成后，源数据库和目标数据库将保持继续同步，您可自由选择业务切换时间。

高传输性能

DTS 使用高规格服务器来保证每条迁移同步链路都能拥有良好的传输性能。在数据迁移方面，数据传输服务底层采取了多种性能优化措施；相对于传统的数据迁移工具，极大提升了传输性能。

故障自动恢复

DTS 具有极高的可用性，每个节点均具备高度的恢复和自愈能力，故障秒级恢复。DTS 的高可靠性让您可以放心将数据迁移上云，无需担心迁移数据的一致性问题。

可视化操作

DTS 无需单独安装驱动程序或应用程序，无需对源数据库做大幅改动，只需在TCloudFinanceZone提供的可视化管理界面进行简单配置即可开启数据传输。

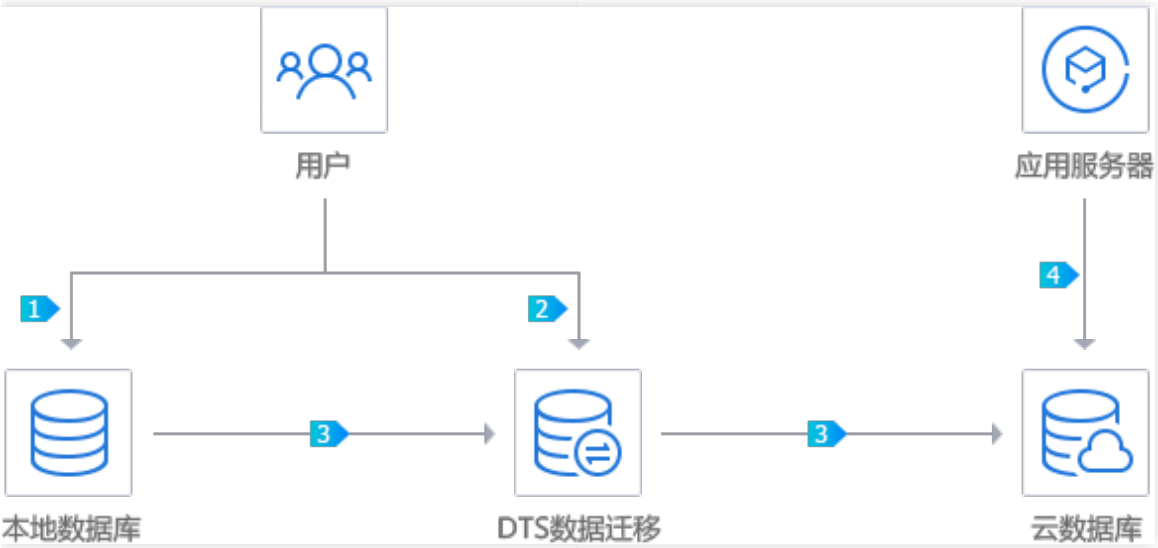
设置简单快速

在TCloudFinanceZone DTS 管理控制台中，只需花费几分钟就能设置一个迁移任务。您可以在迁移任务中定义用来执行迁移的各项参数，其中包括设置源数据库和目标数据库的连接，以及迁移类型和对象。

应用场景

数据迁移上云

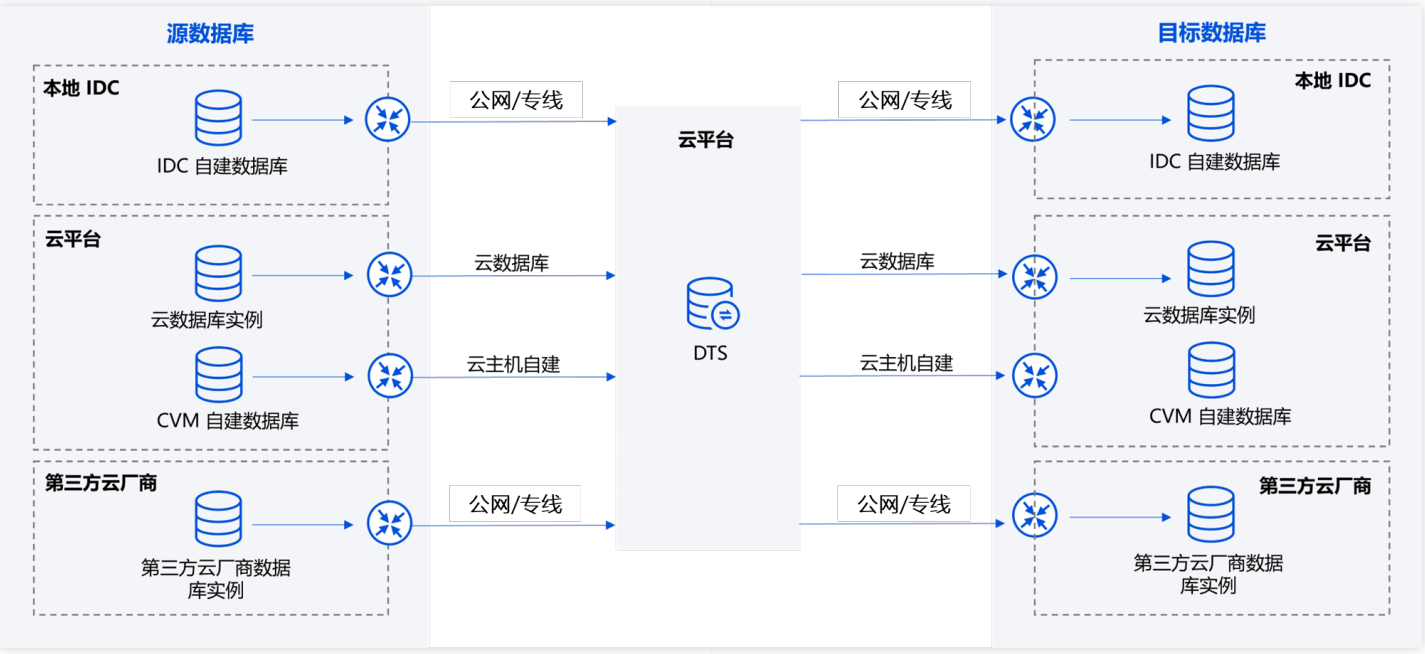
数据传输服务 DTS 提供的数据迁移仅需几步设置，即可帮助您完成数据从本地迁移上云的一系列繁琐工作。迁移过程并不影响源数据库继续对外提供服务，最小化上云过程对您业务的影响。



操作指南
准备工作
网络准备
网络准备概述

操作场景

使用 DTS 可实现本地 IDC、云平台、其他第三方云厂商，这些不同部署形态的数据库之间的同步，方便企业用户进行数据库搬迁、数据库备份、构建云上云下多活架构等。
使用 DTS 进行数据库的同步，需要分别将源/目标数据库所在的网络与云平台网络进行打通，以便 DTS 可以连通源/目标数据库。



DTS 接入类型选择

源/目标数据库采用哪种方式与云平台打通，则在配置 DTS 任务时，接入类型选择对应的方式。

说明

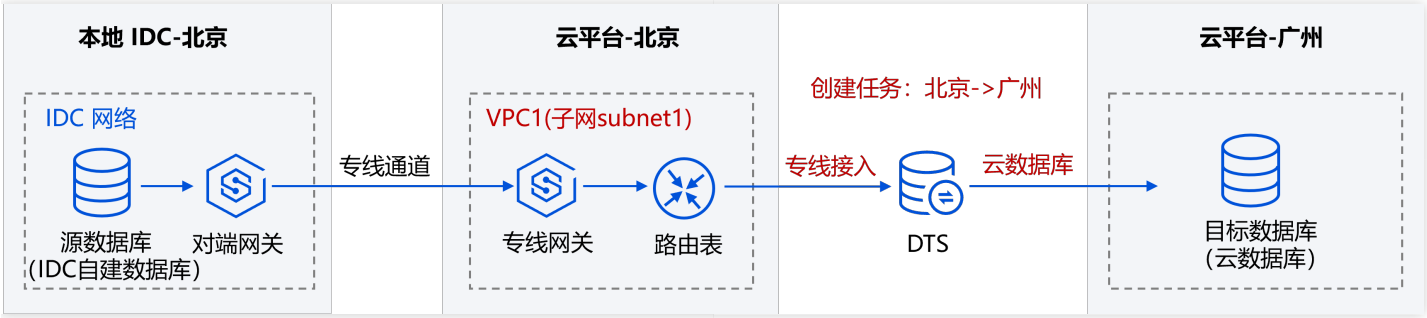
- 添加 DTS IP 地址到对接数据库的安全组或者白名单中，可能会对数据库造成一定的安全风险，请用户在使用过程中加强相关的安全防护，如规范账号密码管理、内部各 API 采用鉴权方式通讯、检查并限制不需要的网段等。使用 DTS 代表您已确认可能会存在的风险，如果用户对安全防护的要求较高，建议选择专线、私有网络 VPC 的接入方式。
- DTS 使用完毕后，建议用户及时删除安全组或防火墙中的 DTS IP 地址。

源/目标库部署类型	接入类型	适用场景	网络配置指引
• IDC 自建数据库 • 其他云厂商数据库	公网	• 数据库可以通过公网 IP 访问。 • 公网无法保证传输带宽，且存在安全隐患，适用于对传输要求不高的场景。	添加 DTS IP 地址至对接数据库白名单（自建数据库通常在防火墙中配置，其他云厂商数据库在安全组中配置）。
	专线接入	• 数据库通过专线接入与私有网络打通。 • 专线接入方式网络链路用户独占，无数据泄露风险，安全性高，满足金融、政企等高等级网络连接要求。	1. 配置通过 VPN 网关实现 VPC 与 IDC 之间的互通。 2. 添加 DTS IP 地址至对接数据库白名单。
CVM 自建数据库	云主机自建	数据库部署在云服务器 CVM 上。	添加 DTS IP 地址至对接数据库白名单。
云数据库实例	云数据库	数据库属于云数据库实例。	添加 DTS IP 地址至对接数据库白名单。
CVM 自建数据库/云数据库实例	私有网络 VPC	数据库和目标数据库都部署在云平台，且有私有网络。	添加 DTS IP 地址至对接数据库白名单。

本地 IDC 与云平台的互通
专线接入

操作场景

使用专线接入方式，需要用户购买一个 VPC 和专线网关，并申请专线通道，将本地 IDC 数据库就近接入到 VPC 中，然后通过 DTS 进行传输任务。
DTS 支持将本地 IDC 数据库作为源/目标进行数据传输，如下示例场景为：本地 IDC 数据库（专线接入） -> 云数据库（云数据库）。



专线接入网络打通

请自行打通专线网络，通过专线建立 VPC 与 IDC 的连接。

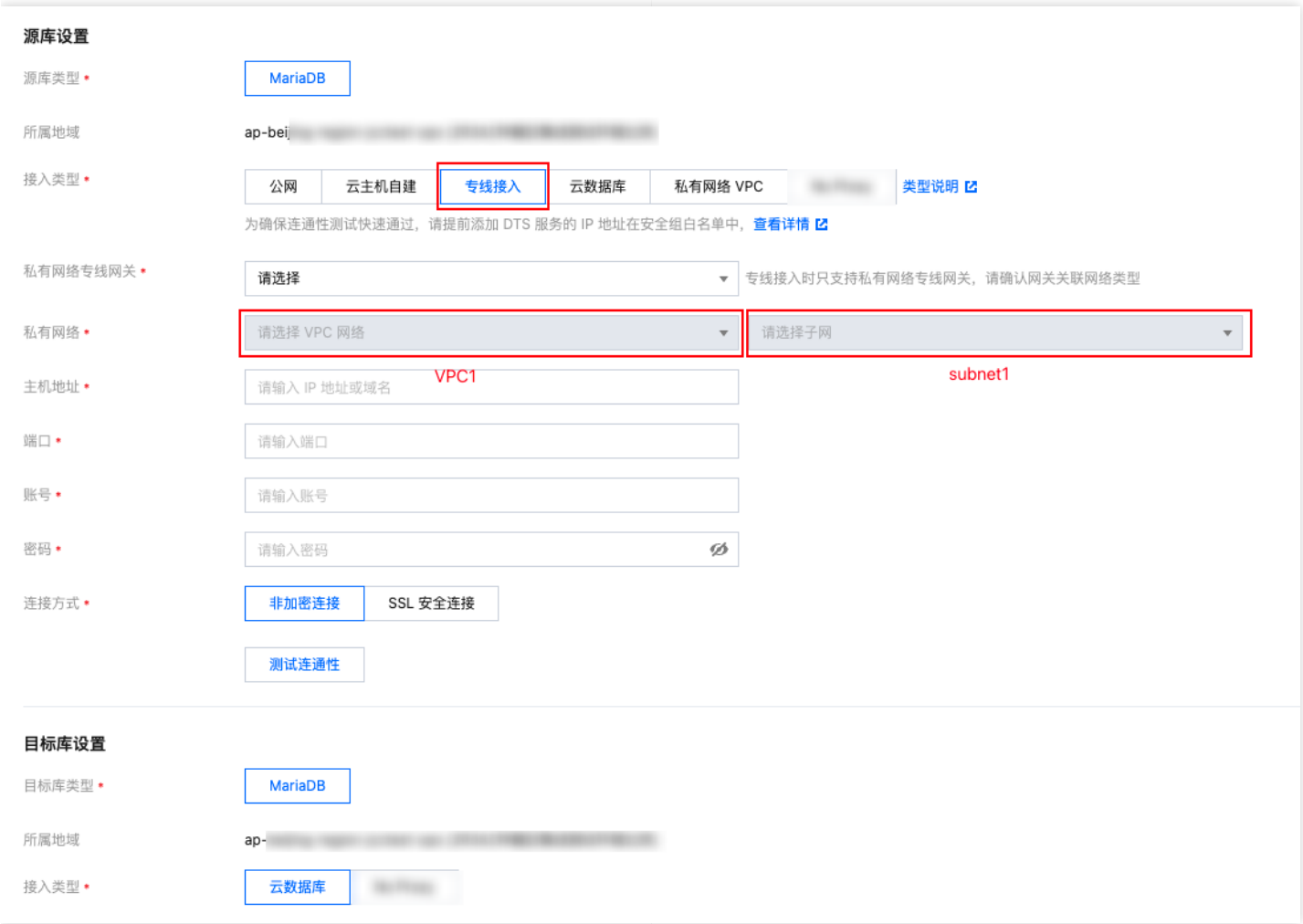
DTS 任务配置

1. 创建 DTS 任务

创建 DTS 任务时，源实例地域选择源库接入 VPC1 所属地域。目标实例地域选择目标数据库所属地域。

2. 配置 DTS 任务

源库设置中，接入类型选择“专线接入”，私有网络选择“VPC1”，并选择其中的一个子网“subnet1”；目标库设置中，接入类型选择“云数据库”。



3. 测试连通性

如果数据库及所属网路配置了安全访问规则，如安全组、防火墙、IP 访问限制等，需要放通 DTS 服务 IP，否则，会出现连通性测试不通过。

放通 DTS 服务 IP

1. 连通性测试不通过时，根据弹窗中的提示，获取“DTS 服务 IP”。
2. 依次检查数据库是否设置了如下网络规则，如果有，请在对应规则中放通 DTS 服务 IP。
    - 数据库所属网络层级是否设置了网络 ACL 或安全组。
    - 数据库所在服务器层级是否设置防火墙（如 Linux 系统的 iptables 规则）。
    - 源数据库层级是否设置了 IP 访问限制。

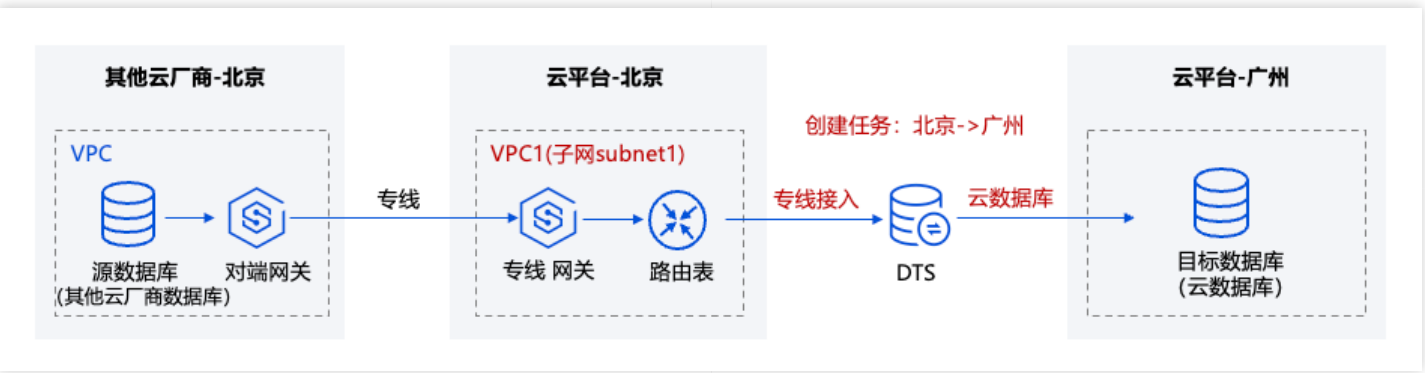
其他云厂商与云平台的互通

操作场景

DTS 支持将其他第三方云厂商数据库作为源/目标库进行数据同步，使用 DTS 进行数据库的同步，需要分别将源/目标数据库所在的网络与云平台网络进行打通。

网络打通操作

如下图中仅展示了第三方云厂商数据库作为源库的接入方式，作为目标库的接入方式类似。
其他云厂商数据库（专线接入） -> 云数据库（云数据库）



DTS 接入数据库的方式支持“公网/专线接入”。

- 选择“公网”方式，只需要在源/目标库上放通 DTS IP 的访问，不需要其他网络打通操作。
- 选择“专线接入”方式，需要用户提前通过专用通道方式，将第三方云厂商数据库所在的网络与云平台网络进行打通。

添加 DTS IP 地址至对接数据库白名单

单个任务，放通 DTS 访问 IP

操作场景

数据迁移、数据同步、数据订阅任务中，需要将 DTS 访问 IP 分别添加到源数据库和目标数据库的白名单中，以便 DTS 可以访问源/目标数据库，否则会出现连通性测试失败。

操作步骤

1. 在配置 DTS 任务时，设置源和目标数据库页面中，参数输入完成后，先进行连通性测试。
2. 如果您的数据库及所属网络设置了安全访问规则，如网络 ACL 和安全组、防火墙（iptables 规则）、数据库账号访问 IP 限制，您需要根据场景分别放通 DTS 访问 IP，否则会出现如下报错，图中提示的地址即为 DTS 服务 IP。
   如果连通性测试通过，则说明数据库没有设置网络限制，可以继续后续任务，无需进行放通操作。

3. 添加 DTS 访问 IP 到数据库的安全规则中。
   不同的接入方式，需要的网络放通操作不同。

接入方式	网络放通排查	说明
公网/专线	- 检查数据库所属网络层级，是否设置了网络 ACL 和安	在相应规则中，放通

接入方式	网络放通排查	说明
	全组规则 - 检查数据库部署服务器层级，是否设置了防火墙（如 iptables）规则 - 检查数据库层级，是否设置了访问 IP 规则（如仅限制授权内的主机地址才可访问数据库）	DTS 服务 IP。
- 云主机自建 - 私有网络 VPC（CVM 自建数据库）	- 检查数据库部署服务器层级，是否设置了防火墙（如 iptables）规则 - 检查数据库层级，是否设置了访问 IP 规则（如限制授权内的主机地址可访问数据库）	在相应规则中，放通 DTS 服务 IP。
- 云数据库 - 私有网络 VPC（云数据库）	- 检查数据库层级，是否设置了访问 IP 规则（如限制授权内的主机地址可访问数据库）	在相应规则中，放通 DTS 服务 IP。

数据库及权限准备

1. 准备源和目标数据库。
2. 分别在源和目标数据库中，创建 DTS 任务账号并授权。

- 建议创建单独用于 DTS 任务的数据库账号，便于区分会话信息以及提升数据安全性。
- 不同链路的授权要求不同，您可以在后续进行 DTS 任务配置时，参考每个链路的配置指导进行授权。

配置自建 MySQL 系的 Binlog

操作场景

当数据迁移、数据同步、数据订阅任务的源库为自建 MySQL 系，即 MySQL/MariaDB/Percona 时，需要用户在自建数据库上设置 Binlog，以满足校验项阶段对源库的要求。

操作影响

本操作需要重启数据库，会对业务造成一定影响，建议在业务低峰阶段操作。

操作步骤

1. 登录源数据库。

2. 参考如下内容修改配置文件 my.cnf 。

> 说明
>
> - my.cnf 配置文件的默认路径为 /etc/my.cnf ，现场以实际情况为准。
> - 建议源端 Binlog 日志至少保留3天及以上，否则可能会因任务暂停/中断时间大于 Binlog 日志保留时间，造成任务无法续传，进而导致任务失败。
> - 在 my.cnf 配置文件中修改会永久生效，如果用户仅想临时生效，请执行 set global expire_logs_days=3 命令修改。
> - MySQL 8.0版本及以上也可以使 binlog_expire_logs_seconds 来修改 Binlog 保留时间，该参数精确到秒级。

```
log_bin = MYSQL_BIN
binlog_format = ROW
server_id = 2 //建议设为大于1的整数，此处仅为示例。
binlog_row_image = FULL
expire_logs_days=3 //修改 binlog 的保留时间，建议大于等于3天
```

3. 重启 MySQL 进程。

```
[\$Mysql_Dir]/bin/mysqladmin -u root -p shutdown
[\$Mysql_Dir]/bin/safe_mysqld &
```

> 说明
>
> [$Mysql_Dir] 指源数据库的安装路径，请替换为实际的源数据库安装目录。

数据迁移
数据迁移支持的数据库

场景概述

数据迁移是将源数据库复制到目标数据库，它以整体数据库搬迁为目的，属于短期的一次性任务，迁移完成后用户一般会进行手动割接，将任务对接到新的数据库上。
DTS 支持的迁移场景示例如下。

场景	说明
自建数据库迁移上云	将本地 IDC 自建数据库、CVM 上的自建数据库，迁移到云平台上。
第三方云厂商迁移到云平台	从其他云厂商（如阿里云、AWS 等）迁移到云平台。
云数据库跨实例间迁移	数据库版本升级，跨地域迁移，不同账号下的数据库实例迁移等。

功能说明

DTS 支持的接入方式有“公网/云主机自建/专线接入/云数据库/私有网络 VPC”，数据库的部署形态不同，可以选择的接入方式也不同。每种接入方式需要具备相应网络条件，请参考 网络准备概述。

- IDC 自建数据库/其他云厂商数据库：接入方式可选择“公网/专线接入”。
- CVM 上的自建数据库：接入方式选择“云主机自建”。
- 云数据库实例：接入方式选择“云数据库”。

迁移至 MariaDB

MySQL > MariaDB

源数据库及版本	目标数据库及版本	迁移类型	参考文档
自建数据库 MySQL（IDC 自建/CVM 上自建）	云数据库 MariaDB	• 结构迁移 • 全量迁移 • 全量 + 增量迁移	MySQL/MariaDB/Percona 迁移至 MariaDB
MySQL （IDC 自建/CVM 上自建）			

MariaDB > MariaDB

源数据库及版本	目标数据库及版本	迁移类型	参考文档
自建数据库 MariaDB	云数据库 MariaDB	• 结构迁移 • 全量迁移 • 全量 + 增量迁移	MySQL/MariaDB/Percona 迁移至 MariaDB
云数据库 MariaDB			

Percona > MariaDB

源数据库及版本	目标数据库及版本	迁移类型	参考文档
自建数据库 Percona（IDC 自建/CVM 上自建）	云数据库 MariaDB	- 结构迁移 - 全量迁移 - 全量 + 增量迁移	MySQL/MariaDB/Percona 迁移至 MariaDB

TDSQL MySQL > MariaDB

源数据库及版本	目标数据库及版本	迁移类型	参考文档
云数据库 TDSQL MySQL	云数据库 MariaDB	• 结构迁移 • 全量迁移 • 全量 + 增量迁移	TDSQL MySQL 迁移至 MariaDB

迁移至 Percona

Percona > Percona

源数据库及版本	目标数据库及版本	迁移类型	参考文档
自建数据库 Percona（IDC 自建/CVM 上自建）	Percona	- 结构迁移 - 全量迁移 - 全量 + 增量迁移	Percona 迁移至 Percona

迁移至 TDSQL MySQL

MySQL > TDSQL MySQL

源数据库及版本	目标数据库及版本	迁移类型	参考文档
自建数据库 MySQL（IDC 自建/CVM 上自建）	云数据库 TDSQL MySQL	- 全量迁移 - 全量 + 增量迁移	MySQL/MariaDB/Percona 迁移至 TDSQL MySQL

MariaDB > TDSQL MySQL

源数据库及版本	目标数据库及版本	迁移类型	参考文档
自建数据库 MariaDB（IDC 自建/CVM 上自建）	云数据库 TDSQL MySQL	• 全量迁移 • 全量 + 增量迁移	MySQL/MariaDB/Percona 迁移至 TDSQL MySQL
云数据库 MariaDB			

Percona > TDSQL MySQL

源数据库及版本	目标数据库及版本	迁移类型	参考文档
自建数据库 Percona（IDC 自建/CVM 上自建）	云数据库 TDSQL MySQL	- 全量迁移 - 全量 + 增量迁移	MySQL/MariaDB/Percona 迁移至 TDSQL MySQL

TDSQL MySQL > TDSQL MySQL

源数据库及版本	目标数据库及版本	迁移类型	参考文档
云数据库 TDSQL MySQL	云数据库 TDSQL MySQL	• 结构迁移 • 全量迁移 • 全量 + 增量迁移	TDSQL MySQL 迁移至 TDSQL MySQL

迁移至 MongoDB

MongoDB > MongoDB

源数据库	目标数据库	迁移类型	参考文档
自建数据库 MongoDB（IDC 自建、CVM 上自建）	云数据库 MongoDB	• 全量迁移 • 全量 + 增量迁移	MongoDB 迁移至 MongoDB
第三方云厂商 MongoDB			
云数据库 MongoDB			

迁移至 Redis

Redis > Redis

源数据库	目标数据库	迁移类型	参考文档
自建数据库 Redis（IDC 自建、CVM 上自建）	云数据库 Redis	全量 + 增量迁移	Redis 迁移至 Redis
第三方云厂商 Redis			
云数据库 Redis®			

云数据库跨账号实例间迁移

操作场景

云数据库跨账号实例间数据迁移，资源库和目标库都属于云数据库实例，但所属不同的主账号名下，这种数据库实例之间的数据迁移。本章节介绍通过 DTS 数据迁移功能实现从其他账号云数据库实例迁移数据至本账号下云数据库实例。

支持范围

源数据库为云数据库实例，具体为：TDSQL MySQL、MariaDB、MongoDB、Redis®。

前提条件

已创建目标数据库实例。

注意事项

本操作中涉及多处账号信息配置，如下列出了主要配置逻辑，以方便用户理解和正确配置。

- 数据迁移方向：源数据库（其他账号的数据库实例）> 目标数据库（本账号的数据库实例）。
- 执行迁移任务的账号可以是目标数据库的主账号，也可以目标数据库的子账号。
  - 使用主账号执行迁移任务。操作任务前，需要请求源数据所属主账号持有人进行角色授权（给目标数据库的主账号），使目标数据库主账号可以访问源数据库。
  - 使用子账号执行迁移任务。操作任务前，需要先请求源数据所属主账号持有人进行角色授权（给目标数据库的主账号），使目标数据库主账号通过角色访问源数据库。然后再请求目标数据库所属主账号持有人进行策略授权（对目标数据库的子账号），使目标数据库子账号可以访问源数据库。

授权账号

使用主账号执行迁移任务，请操作步骤1 - 6，使用子账号执行迁移任务，请操作步骤1 - 11。

1. 使用源数据库所属的主账号登录 访问管理控制台（如果子账号有 CAM 和角色相关的权限，也可以使用子账号登录）。

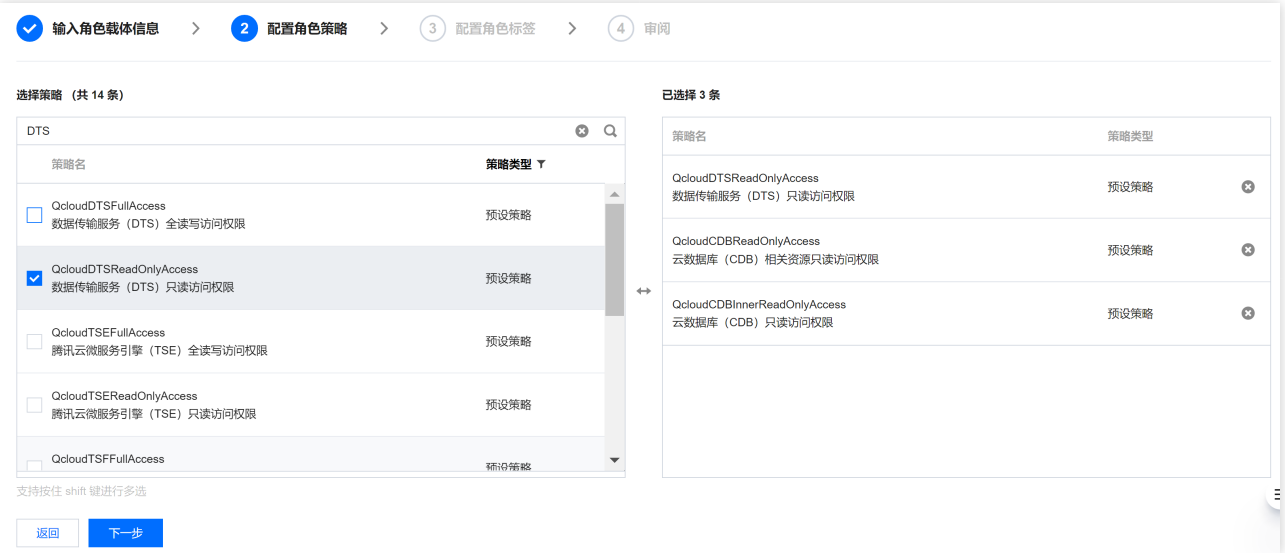
2. 左侧导航单击角色，进入角色管理页面，然后单击新建角色。

3. 配置相关信息，单击下一步。

- 角色名称：配置后请记录，后续配置 DTS 任务时需要输入。
- 角色载体：填入目标数据库所属的主账号 ID。目标数据库实例属于子账号名下时，此处也填写主账号 ID。

4. 在配置角色策略页面，选择 DTS 策略和源数据库服务对应策略，单击下一步。

   i. DTS 服务策略，选择 QcloudDTSReadOnlyAccess。
   ii. 源数据库服务对应的策略。
   - 数据库为云数据库 TDSQL MySQL，选择 QcloudTDSQLReadOnlyAccess（TDSQL MySQL 版（TDSQL for MySQL）只读访问权限）。
   - 数据库为云数据库 MariaDB，选择 QcloudMariaDBReadOnlyAccess（云数据库 MariaDB 只读访问权限）。
   - 源数据库为云 MongoDB，选择 QcloudMongoDBReadOnlyAccess（文档数据库（MongoDB）只读访问权限）。
   - 源数据库为云 Redis®，选择 QcloudRedisReadOnlyAccess（云缓存 Redis® 只读访问权限）。

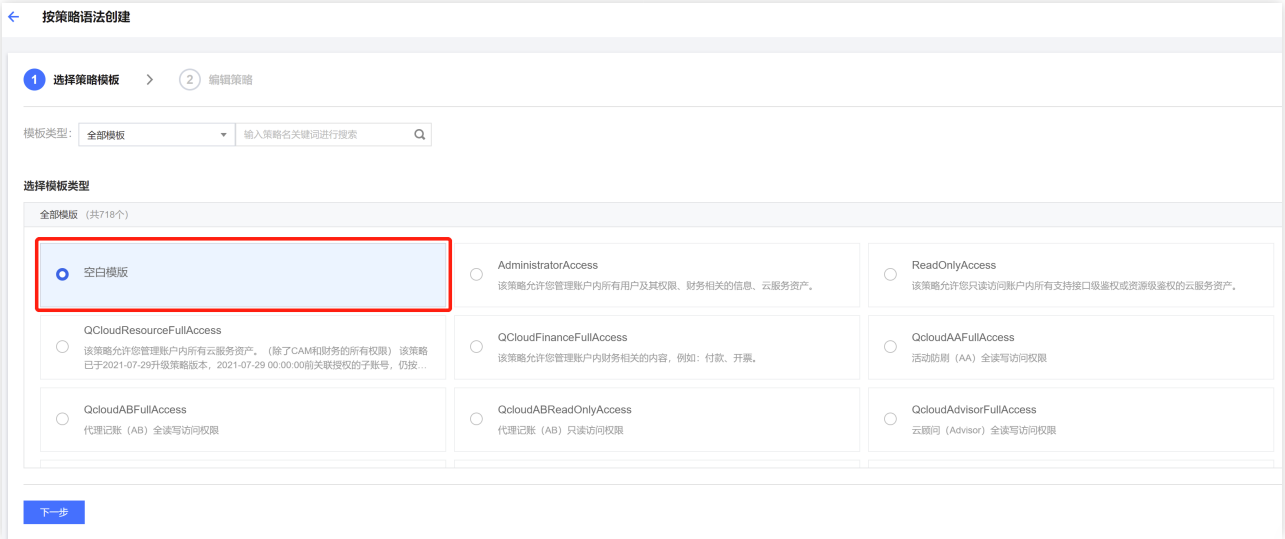


5. 在审阅页面，单击完成后该角色创建完成。

> 说明：
>
> 如果执行迁移任务的账号为主账号，授权步骤到此结束；如果为子账号，还需要继续进行如下步骤7-11，请求当前主账号持有人，对子账号进行策略授权。

6. （可选）使用目标数据库所属的主账号登录访问管理控制台，在左侧导航单击策略，然后在右侧单击新建自定义策略，并选择按策略语法创建。

7.（可选）选择空白模板，然后单击下一步。

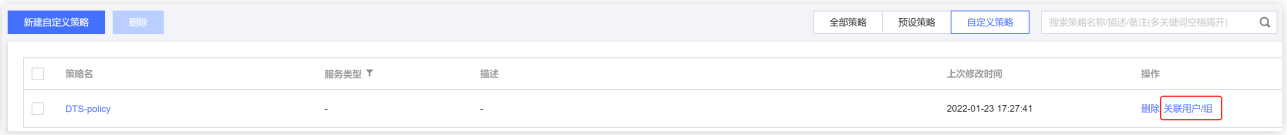


8. （可选）创建一个策略，策略的名称以及描述可以根据自己的需求填写，策略内容复制示例代码后，将红框中的内容替换成对应的信息。

策略语法示例：

```
{
    "version": "2.0",
    "statement": [
    {
        "effect": "allow",
        "action": ["name/sts:AssumeRole"],
  "resource": ["qcs::cam::uin/10*******8:roleName/DTS-role"]
    }
 ]
}
```

9. （可选）单击完成后返回到策略列表页，在列表页中单击关联用户/组。



10. （可选）选择目标数据库实例所属子账号（即执行迁移任务的子账号），单击确定，如下图所示。

创建迁移任务

1. 使用目标数据库实例所属账号，登录 DTS 控制台。
2. 选择数据迁移 > 新建迁移任务，购买一个新的迁移任务。
3. 购买完成后，返回数据迁移列表，单击操作列的配置，进入配置迁移任务页面。
4. 在设置源和目标数据页面，配置源库和目标库信息。

跨账号关键参数配置如下：

- 接入类型：选择云数据库，表示源数据库属于云数据库实例。
- 是否跨账号：选择跨账号。
- 跨账号 ID： 填入源数据库所属的主账号 ID。
- 跨账号授权角色名称。即前文授权账号步骤6中创建的角色名称。
- 外部角色 ID：如果前文授权账号步骤4中设置了外部 ID，则这里则必须按照设置的内容正确填写，否则会发生权限报错。没有设置则不需要填写。

5. 在设置迁移选项和迁移对象页面，对数据迁移选项、迁移对象选项进行设置，在设置完成后单击保存并下一步。
6. 在校验任务页面，完成校验并全部校验项通过后，单击启动任务。

7. 返回数据迁移任务列表，任务开始进入运行中状态。

常见问题

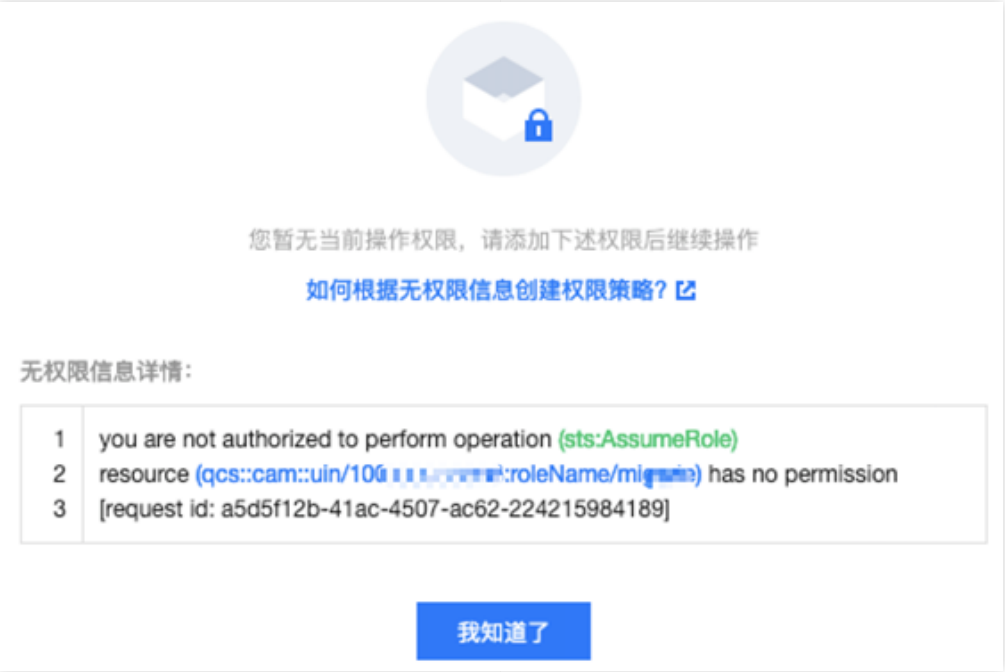
1. 跨账号拉取实例列表报错：role not exist[InternalError.GetRoleError]

请确认跨账号 ID（应该为源数据库的主账号 ID）和跨账号授权角色名称（应该为授权账号步骤6中创建的角色名称）配置是否正确。

2. 获取云数据库实例列表失败：InternalError:InternalInnerCommonError

角色中没有授权源数据库所属的云服务策略，请参考授权账号中的步骤5进行授权。

3. 跨账号拉取实例列表报错：you are not authorized to perform operation (sts:AssumeRole) ，resource (qcs::cam::uin/1xx5:roleName/xxxx) has no permission



出错原因：您当前创建迁移任务使用的账号是子账号，并且当前子账号没有 sts:AssumeRole 权限。

解决方法：

- 使用主账号来创建迁移任务。
- 请求目标数据库所属的主账号持有人，参考授权账号对子账号授权，策略语法中的 resource，填写报错框中蓝色字段部分。

4. DTS 任务配置未填写角色外部 ID，或者填写错误，导致权限报错或者拉取数据库实例失败。

出错原因：授权账号时设置了外部 ID，但在 DTS 任务配置中未填写角色外部 ID，或者填写错误，导致权限报错或者拉取数据库示例失败。

解决方案：授权账号时如果设置了外部 ID，则 DTS 任务配置时需要正确填写，未设置则不需要填。

迁移至 MySQL 系列
迁移至 MariaDB
MySQL/MariaDB/Percona 迁移至 MariaDB

本文介绍使用 DTS 数据迁移功能从 MySQL、MariaDB、Percona 迁移数据至云数据库 MariaDB 的操作指导。
源数据库支持的部署类型如下：

- 自建 MySQL、云数据库 MySQL。
- 自建 MariaDB、云数据库 MariaDB。
- 自建 Percona。

> 说明：
>
> 云数据库 MariaDB 支持三种内核 MariaDB、MySQL 和 Percona，用户在使用时不需要区分哪种内核，如果源数据库为云 MariaDB，不论源数据库的内核是 MariaDB、Percona 还是 MySQL，在设置源数据库的类型时，都选择 MariaDB。

因为 MySQL、MariaDB、Percona 迁移数据至云数据库 MariaDB，三种场景的迁移要求和操作步骤基本一致，本章节仅以 MariaDB 到 MariaDB 的数据迁移为例进行介绍，其他场景请参考相关内容。

注意事项

- DTS 在执行全量数据迁移时，会占用一定源端实例资源，可能会导致源实例负载上升，增加数据库自身压力。如果您的数据库配置过低，建议您在业务低峰期进行迁移。
- 默认采用无锁迁移来实现，迁移过程中对源库不加全局锁（FTWRL），仅对无主键的表加表锁，其他不加锁。
- 创建数据一致性校验 时，DTS 会使用执行迁移任务的账号在源库中写入系统库 __tencentdb__ ，用于记录迁移任务过程中的数据对比信息。
  - 为保证后续数据对比问题可定位，迁移任务结束后不会删除源库中的 __tencentdb__ 。
  - __tencentdb__ 系统库占用空间非常小，约为源库存储空间的千分之一到万分之一（例如源库为50GB，则 __tencentdb__ 系统库约为5MB - 50MB），并且采用单线程，等待连接机制，所以对源库的性能几乎无影响，也不会抢占资源。

前提条件

- 已创建云数据库 MariaDB。
- 源数据库和目标数据库符合迁移功能和版本要求，请参见 数据迁移支持的数据库 进行核对。
- 已完成 准备工作。
- 源数据库需要具备的权限如下：
  - “整个实例”迁移：

```
CREATE USER '迁移账号'@'%' IDENTIFIED BY '迁移密码';
```

```
GRANT RELOAD,LOCK TABLES,REPLICATION CLIENT,REPLICATION SLAVE,SHOW DATABASES,SHOW VIE
W,PROCESS ON *.* TO '迁移账号'@'%';
//源端若为云 MariaDB 数据库，需要提交工单进行 RELOAD 授权，其他场景请用户参照代码授权
//源库为阿里云数据库时，不需要授权 SHOW DATABASES，其他场景则需要授权。阿里云数据库授权，请参考
https://help.aliyun.com/document_detail/96101.html
//如果选择迁移触发器和事件，需要同时授权 TRIGGER 和 EVENT 权限
GRANT ALL PRIVILEGES ON `__tencentdb__`.* TO '迁移账号'@'%';
GRANT SELECT ON *.* TO '迁移账号';
```

- “指定对象”迁移：

```
CREATE USER '迁移账号'@'%' IDENTIFIED BY '迁移密码';
GRANT RELOAD,LOCK TABLES,REPLICATION CLIENT,REPLICATION SLAVE,SHOW DATABASES,SHOW VIE
W,PROCESS ON *.* TO '迁移账号'@'%';
//源端若为云 MariaDB 数据库，需要提交工单进行 RELOAD 授权，其他场景请用户参照代码授权
//源库为阿里云数据库时，不需要授权 SHOW DATABASES，其他场景则需要授权。阿里云数据库授权，请参考
https://help.aliyun.com/document_detail/96101.html
//如果选择迁移触发器和事件，需要同时授权 TRIGGER 和 EVENT 权限
GRANT ALL PRIVILEGES ON `__tencentdb__`.* TO '迁移账号'@'%';
GRANT SELECT ON `mysql`.* TO '迁移账号'@'%';
GRANT SELECT ON 待迁移的库.* TO '迁移账号';
```

- 需要具备目标数据库的权限：ALTER, ALTER ROUTINE, CREATE, CREATE ROUTINE, CREATE TEMPORARY TABLES, CREATE USER, CREATE VIEW, DELETE, DROP, EVENT, EXECUTE, INDEX, INSERT, LOCK TABLES, PROCESS, REFERENCES, RELOAD, SELECT, SHOW DATABASES, SHOW VIEW, TRIGGER, UPDATE。

应用限制

- 支持迁移基础表、视图、函数、触发器、存储过程和事件。不支持迁移系统库表，包括 information_schema ， sys ， performance_schema ， __cdb_recycle_bin__ ， __recycle_bin__ ， __tencentdb__ ， mysql 。
- 在迁移视图、存储过程和函数时，DTS 会检查源库中 DEFINER 对应的 user1（ [DEFINER = user1]）和迁移账号 user2 是否一致，如果不一致，迁移后 DTS 会修改 user1 在目标库中的 SQL SECURITY 属性，由 DEFINER 转换为 INVOKER （ [INVOKER = user1]），同时设置目标库中 DEFINER 为迁移账号 user2（[DEFINER = 迁移账号 user2]）。如果源库中视图定义过于复杂，可能会导致任务失败。
- 源端如果是非 GTID 实例，DTS 不支持源端 HA 切换，一旦源端 MySQL 发生切换可能会导致 DTS 增量同步中断。
- 只支持迁移 InnoDB、MyISAM、TokuDB 三种数据库引擎，如果存在这三种以外的数据引擎表则默认跳过不进行迁移。其中，源库如果存在压缩模式的 TokuDB 引擎数据，需要目标库同步支持压缩模式才可以迁移，否则任务会报错。
- 相互关联的数据对象需要同时迁移，否则会导致迁移失败。常见的关联关系：视图引用表、视图引用视图、主外键关联表等。
- 增量迁移过程中，若源库产生了类型为 STATEMENT 格式的 Binlog 语句，则会导致迁移失败。
- 源数据库为 MariaDB 时，应用限制如下。
  - DTS 迁移任务要求源库、目标库的 lower_case_tame_name 参数（表名大小敏感）保持一致，如果源数据库为云数据库 MariaDB，由于云数据库 MariaDB 只能在创建实例时修改 lower_case_tame_name 参数，所以用户需要在创建源库实例时确定大小写敏感规则，并在参数校验不一致时，修改目标库的 lower_case_tame_name 参数。
- 无锁迁移场景，迁移任务步骤为“源库导出”时，不支持 DDL 操作。

- 源数据库 Binlog 的 GTID 如果存在空洞，可能会影响迁移任务的性能并导致任务失败。
- 不支持同时包含 DML 和 DDL 语句在一个事务的场景，遇到该情况任务会报错。
- 不支持 Geometry 相关的数据类型，遇到该类型数据任务报错。
- 不支持 ALTER VIEW 语句，遇到该语句任务跳过不迁移。

操作限制

- 迁移过程中请勿进行如下操作，否则会导致迁移任务失败。
  - 请勿修改、删除源数据库和目标数据库中用户信息（包括用户名、密码和权限）和端口号。
  - 请勿在源库写入 Binlog 格式为 STATEMENT 的数据。
  - 请勿在源库上执行清除 Binlog 的操作。
  - 在库表结构迁移和全量迁移阶段，请勿执行库或表结构变更的 DDL 操作。
  - 在增量迁移阶段，请勿删除系统库表 __tencentdb__ 。
- 如果仅执行全量数据迁移，请勿在迁移过程中向源实例中写入新的数据，否则会导致源和目标数据不一致。针对有数据写入的场景，为实时保持数据一致性，建议选择全量+增量数据迁移。
- 源数据库为阿里云 RDS，PolarDB 时，由于 RDS，PolarDB 在 Binlog 中为无主键或无非空唯一键的表加上附加主键列，但在表结构中不可见，可能会导致 DTS 无法识别，因此建议用户尽量不要迁移无主键的表。

支持的 SQL 操作

操作类型	支持的 SQL 操作
DML	INSERT、UPDATE、DELETE、REPLACE
DDL	- TABLE：CREATE TABLE、ALTER TABLE、DROP TABLE、TRUNCATE TABLE、RENAME TABLE - VIEW：CREATE VIEW、DROP VIEW - INDEX：CREATE INDEX、DROP INDEX - DATABASE：CREATE DATABASE、ALTER DATABASE、DROP DATABASE 暂不支持 CREATE TABLE 表名 AS SELECT 语句。

环境要求

> 说明：
>
> 如下环境要求，系统会在启动迁移任务前自动进行校验，不符合要求的系统会报错。如果用户能够识别出来，可以参考 前置校验项不通过处理 自行修改，如果不能则等系统校验完成，按照报错提示修改。

类型	环境要求
源数据库要求	- 源库和目标库网络能够连通。 - 源库所在的服务器需具备足够的出口带宽，否则将影响迁移速率。 - 实例参数要求： - 源库 server_id 参数需要手动设置，且值不能设置为0。 - 源库表的 row_format 不能设置为 FIXED。

类型	环境要求
	- 源库和目标库 lower_case_table_names 变量必须设置为一致。 - 源库变量 connect_timeout 设置数值必须大于10。 - 建议开启 skip-name-resolve，减少连接超时的可能性。 - Binlog 参数要求： - 源库 log_bin 变量必须设置为 ON。 - 源库 binlog_format 变量必须设置为 ROW。 - 源库 binlog_row_image 变量必须设置为 FULL。 - MySQL 5.6 及以上版本 gtid_mode 变量不为 ON 时会报警告，建议打开 gtid_mode。 - 不允许设置 do_db, ignore_db 过滤条件。 - 源实例为从库时，log_slave_updates 变量必须设置为 ON。 - 建议源库 Binlog 日志至少保留3天及以上，否则可能会因任务暂停/中断时间大于 Binlog 日志保留时间，造成任务无法续传，进而导致任务失败。 - 外键依赖： - 外键依赖只能设置为 NO ACTION，RESTRICT 两种类型。 - 部分库表迁移时，有外键依赖的表必须齐全。 - DTS 对数据类型为 FLOAT 的迁移精度为38位，对数据类型为 DOUBLE 的迁移精度为308位，需要确认是否符合预期。 - 环境变量 innodb_stats_on_metadata 必须设置为 OFF。
目标数据库要求	- 目标库的版本必须大于等于源库的版本。 - 目标库的空间大小须是源库待迁移库表空间的1.2倍以上。（全量数据迁移会并发执行 INSERT 操作，导致目标数据库的表产生碎片，因此全量迁移完成后目标数据库的表存储空间很可能会比源实例的表存储空间大） - 目标库不能有和源库同名的表、视图等迁移对象。 - 目标库 max_allowed_packet 参数设置数值至少为4M。

操作步骤

1. 登录 DTS 控制台，在左侧导航选择数据迁移页，单击新建迁移任务，进入新建迁移任务页面。

2. 在新建迁移任务页面，选择迁移的源实例类型和所属地域，目标实例类型和所属地域，规格等，然后单击立即创建。
   创建完成后，页面自动跳转到数据迁移的任务列表，请选择刚才购买的任务进行配置。如果您购买了多个地域，或者跨地域的 DTS 任务，任务列表按照目标实例的所属地域进行展示，可在上方切换地域后找到已购买的任务。

3. 在设置源和目标数据库页面，完成任务设置、源库设置和目标库设置，测试源库和目标库连通性通过后，单击新建。

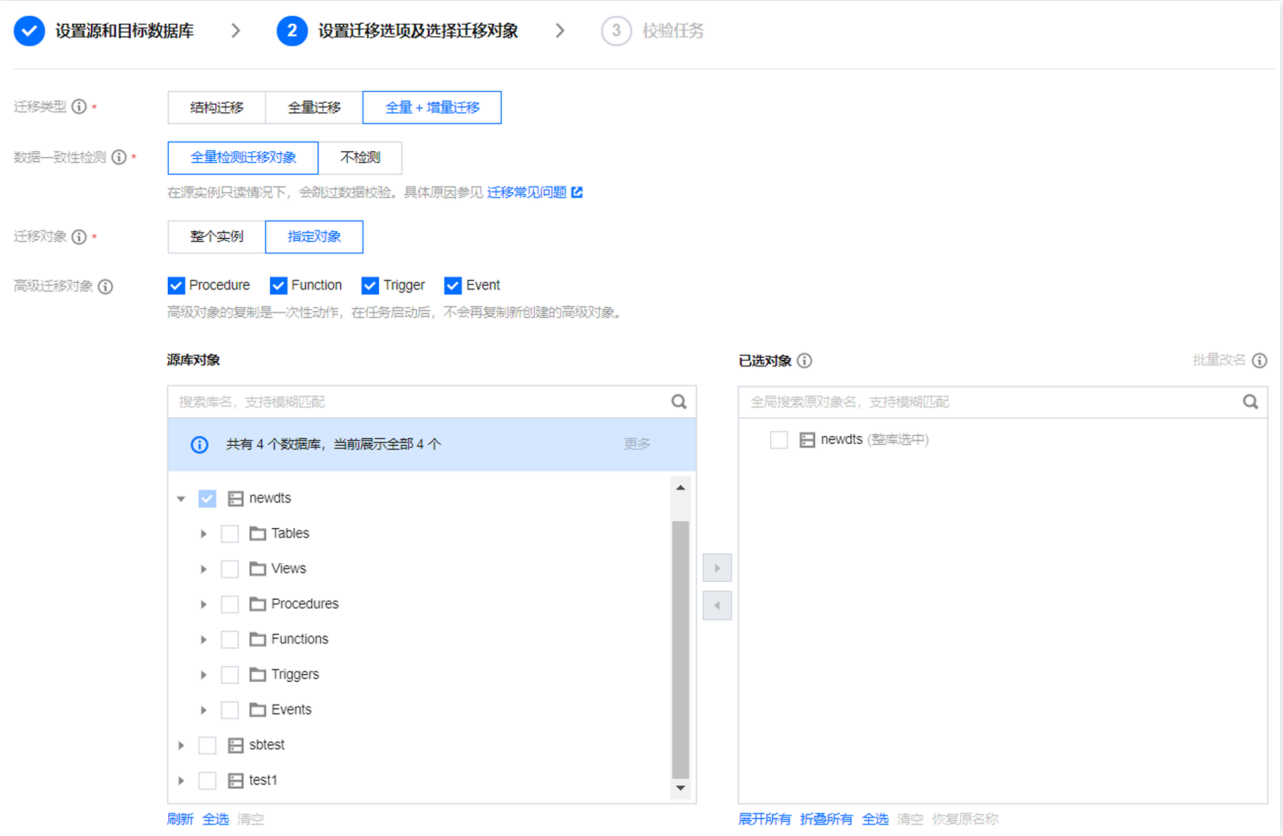
设置类型	配置项	说明
任务设置	任务名称	设置一个具有业务意义的名称，便于任务识别。
	运行模式	◦ 立即执行：完成任务校验通过后立即启动任务。 ◦ 定时执行：需要配置一个任务执行时间，到时间后启动任务。
	自动重试	◦ 设置后，迁移任务因网络异常等引起的任务临时中断，DTS 将在设置的时间范围内自动重试和恢复任务，不需要用户手动操作。 ◦ 支持设置的时间范围为5分钟-720分钟。
源库设置	源库类型	创建时选择的源库类型，不可修改。

设置类型	配置项	说明
	所属地域	创建时选择的源库地域，不可修改。
	接入类型	请根据您的场景选择，以下以专线接入为例，不同接入类型的准备工作请参考网络准备概述。 ◦ 公网：源数据库可以通过公网 IP 访问。 ◦ 云主机自建：源数据库部署在云服务器 CVM 上。 ◦ 专线接入：源数据库可以通过 专线接入 方式与私有网络打通。 ◦ 云数据库：源数据库属于云数据库实例。
	私有网络专线网关	专线接入时只支持私有网络专线网关，请确认网关关联网络类型。
	私有网络	选择VPC网络和子网。
	主机地址	源库访问 IP 地址或域名。
	端口	源库访问端口。
	账号	源库的数据库账号，账号权限需要满足要求。
	密码	源库的数据库账号的密码。
	连接方式	SSL 安全连接指 DTS 与数据库通过 SSL（Secure socket layer）安全连接，对传输链路进行加密。 选择 SSL 安全连接可能会增加数据库的连接响应时间，采用公网/专线等传输方式，并且对数据安全要求较高的场景，需要开启 SSL 安全连接。选择 SSL 安全连接前，需要先在数据库中开启 SSL 加密。
目标库设置	目标库类型	购买时选择的目标库类型，不可修改。
	所属地域	购买时选择的目标库地域，不可修改。
	接入类型	根据您的场景选择，本场景选择“云数据库”。
	数据库实例	选择目标库的实例 ID。
	账号	目标库的数据库账号，账号权限需要满足要求。
	密码	目标库的数据库账号的密码。
	连接方式	SSL 安全连接指 DTS 与数据库通过 SSL（Secure socket layer）安全连接，对传输链路进行加密。 选择 SSL 安全连接可能会增加数据库的连接响应时间，采用公网/专线等传输方式，并且对数据安全要求较高的场景，需要开启 SSL 安全连接。选择 SSL 安全连接前，需要先在数据库中开启 SSL 加密。

4. 在设置迁移选项及选择迁移对象页面，设置迁移类型、对象，单击保存。

> 说明：
>
> 如果用户在迁移过程中确定会对某张表使用 rename 操作（例如将 table A rename 为 table B），则迁移对象需要选择 table A 所在的整个库（或者整个实例），不能仅选择 table A，否则 rename 操作后，table B 的数据不会同步到

目标库。

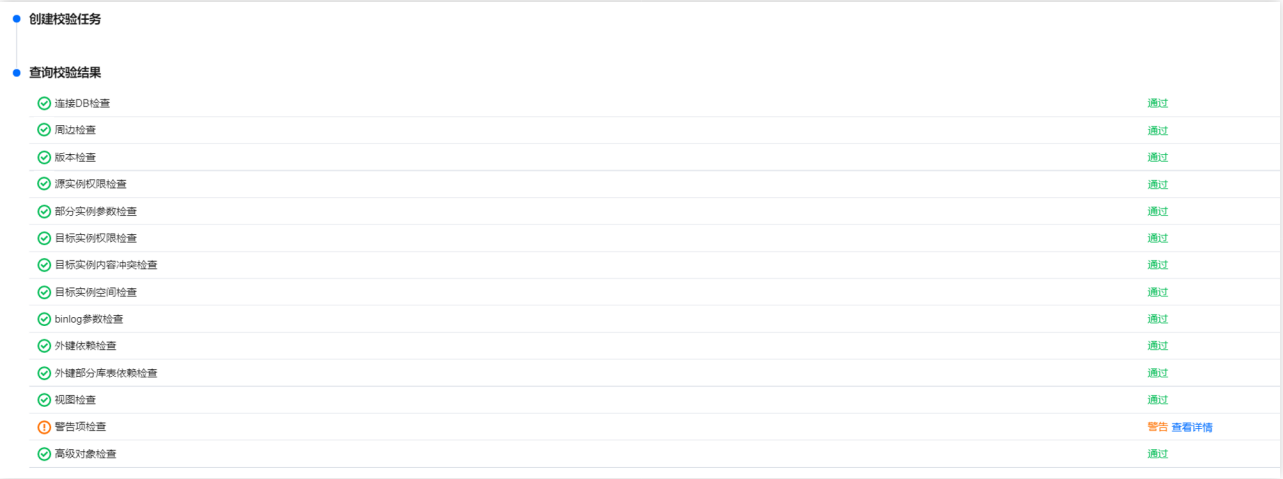


配置项	说明
迁移类型	请根据您的场景选择。 - 结构迁移：迁移数据库中的库、表等结构化的数据。 - 全量迁移：迁移整个数据库，迁移数据仅针对任务发起时，源数据库已有的内容，不包括任务发起后源库实时新增的数据写入。 - 全量 + 增量迁移：迁移数据包括任务发起时源库的已有内容，也包括任务发起后源库实时新增的数据写入。如果迁移过程中源库有数据写入，需要不停机平滑迁移，请选择此场景。
数据一致性检测	当选择“全量 + 增量迁移”时，支持进行数据一致性检测，对迁移后源库和目标库的数据进行详细的对比检测。 - 勾选“全量检测迁移对象”后，迁移任务进行到“同步增量”阶段，目标与源库数据差距为0MB，目标与源库时间延迟也为0秒时，DTS 会自动触发一次一致性校验任务。 - 未勾选“全量检测迁移对象”，用户也可在任务进行到“同步增量”阶段，手动进行创建一致性校验任务。
迁移对象	- 整个实例：迁移整个实例，但不包括系统库，如 information_schema、mysql、performance_schema、sys。 - 指定对象：迁移指定对象。
高级迁移对象	支持迁移存储过程（Procedure）、函数（Function）、触发器（Trigger）、事件（Event）。 - 高级对象的迁移是一次性动作，仅支持迁移在任务启动前源库中已有的高级对象，在任务启动后，新增的高级对象不会同步到目标库中。 - 存储过程和函数，在“源库导出”阶段进行迁移；触发器和事件，没有增量任务，在任务结束时进行迁移，有增量任务，在用户单击完成操作后开始迁移，所以单击完成后，任务的过渡时间会略微

配置项	说明
	增加。
已选对象	- 支持库表映射（库表重命名），将鼠标悬浮在库名、表名上即显示编辑按钮，单击后可在弹窗中填写新的名称。 - 选择高级对象进行迁移时，建议不要进行库表重命名操作，否则可能会导致高级对象迁移失败。
是否同步 Online DDL 临时表	如果使用 gh-ost、pt-osc 工具对源库中的表执行 Online DDL 操作，DTS 支持将 Online DDL 变更产生的临时表迁移到目标库。 - 勾选 gh-ost，DTS 会将 gh-ost 工具产生的临时表名（ _表名_ghc 、 _表名_gho 、 _表名_del ）迁移到目标库。 - 勾选 pt-osc， DTS 会将 pt-osc 工具产生的临时表名（ _表名_new 、 _表名_old ）迁移到目标库。

5. 在校验任务页面，进行校验，校验任务通过后，单击启动任务。

- 如果校验任务不通过，可以参考 前置校验项不通过处理 修复问题后重新发起校验任务。
- 失败：表示校验项检查未通过，任务阻断，需要修复问题后重新执行校验任务。
- 警告：表示检验项检查不完全符合要求，可以继续任务，但对业务有一定的影响，用户需要根据提示自行评估是忽略警告项还是修复问题再继续。



-

6. 返回数据迁移任务列表，任务进入准备运行状态，运行1分钟 - 2分钟后，数据迁移任务开始正式启动。

- 选择结构迁移或者全量迁移：任务完成后会自动结束，不需要手动结束。
- 选择全量 + 增量迁移：全量迁移完成后会自动进入增量数据同步阶段，增量数据同步不会自动结束，需要您手动单击完成结束增量数据同步。
    - 请选择合适时间手动完成增量数据同步，并完成业务切换。
    - 观察迁移阶段为增量同步，并显示无延迟状态，将源库停写几分钟。
    - 目标与源库数据差距为0KB及目标与源库时间延迟为0秒时，手动完成增量同步。

7. （可选）如果您需要进行查看任务、删除任务等操作，请单击对应的任务，在操作列进行操作。
8. 当迁移任务状态变为任务成功时，即可对业务进行正式割接。

TDSQL MySQL 迁移至 MariaDB

TDSQL MySQL 迁移至云数据库 MariaDB 的要求和指导，与 TDSQL MySQL 迁移至 TDSQL MySQL 的内容基本一致，请参考相关内容进行操作。

迁移至 TDSQL MySQL
TDSQL MySQL 迁移至 TDSQL MySQL

本文介绍使用 DTS 数据迁移功能，从 TDSQL MySQL 迁移数据至分布式数据库 TDSQL MySQL 的操作指导。

注意事项

- DTS 在执行全量数据迁移时，会占用一定源端实例资源可能会导致源实例负载上升，增加数据库自身压力。如果您数据库配置过低，建议您在业务低峰期进行。
- 默认采用无锁迁移来实现，迁移过程中对源库不加全局锁（FTWRL），仅对无主键的表加表锁，其他不加锁。
- 创建数据一致性校验时，DTS 会使用执行迁移任务的账号在源库中写入系统库 __tencentdb__ ，用于记录迁移任务过程中的数据对比信息。
  - 为保证后续数据对比问题可定位，迁移任务结束后不会删除源库中的 __tencentdb__ 。
  - __tencentdb__ 系统库占用空间非常小，约为源库存储空间的千分之一到万分之一（例如源库为50GB，则 __tencentdb__ 系统库约为5MB - 50MB），并且采用单线程，等待连接机制，所以对源库的性能几乎无影响，也不会抢占资源。

前提条件

- 需要您在源端 TDSQL MySQL 中提前创建好数据库： __tencentdb__ （参考语法 CREATE DATABASE __tencentdb__; ）。

- 需要具备源数据库的权限。

```
CREATE USER '迁移账号'@'%' IDENTIFIED BY '迁移密码';
GRANT SELECT,RELOAD,LOCK TABLES,REPLICATION CLIENT,REPLICATION SLAVE,SHOW DATABASES,SHOW VIEW,PROCESS ON *.* TO '迁移账号'@'%';
GRANT INSERT, UPDATE, DELETE, DROP, SELECT, INDEX, ALTER, CREATE ON `__tencentdb__`.* TO '迁移账号'@'%';
```

- 需要具备目标数据库的权限：ALTER, ALTER ROUTINE, CREATE, CREATE ROUTINE, CREATE TEMPORARY TABLES, CREATE USER, CREATE VIEW, DELETE, DROP, EVENT, EXECUTE, INDEX, INSERT, LOCK TABLES, PROCESS, REFERENCES, RELOAD, SELECT, SHOW DATABASES, SHOW VIEW, TRIGGER, UPDATE。

应用限制

1. 只支持迁移基础表、视图，不支持迁移函数、触发器、存储过程等对象。
2. 不支持迁移系统库表和用户信息，包括 information_schema ， sysdb ， test ， sys ， performance_schema ， __tencentdb__ ， mysql 。
3. 只支持迁移 InnoDB 数据库引擎，如果存在其他的数据引擎表则默认跳过不进行迁移。

4. 源端 TDSQL MySQL 中对表的数量有限制，整个实例最多为5000个，超出后 DTS 任务会报错；同时，表的数量太多会导致源端的访问耗时变大，引起性能抖动和下降。
5. 相互关联的数据对象需要同时迁移，否则会导致迁移失败。
6. 增量迁移过程中，若源库产生了类型为 STATEMENT 格式的 Binlog 语句，则会导致迁移失败。
7. 不支持迁移二级分区表。
   i. TDSQL MySQL 迁移到 MySQL/MariaDB，遇到二级分区表的迁移，任务报错。
   ii. TDSQL MySQL 迁移到 TDSQL MySQL，如果迁移的库表中包含二级分区表，存量数据会跳过二级分区表的迁移；如果选择整库或全实例迁移，增量过程中遇到二级分区表任务报错暂停。
8. 不支持同时包含 DML 和 DDL 语句在一个事务的场景，遇到该情况任务会报错。
9. 不支持 Geometry 相关的数据类型，遇到该类型数据任务报错。
10. 不支持主键列为某列的截取字段（如：create table tb_length_key_11(c1 char(255), primary key pk(c1(10)))charset=utf8;），遇到这种情况任务会报错。
11. 增量迁移期间不支持修改主键，包括主键列、分区表分布键、对主键列的 COMMENT 注释，增删改列字段和长度。
12. 源端 TDSQL MySQL 为 MariaDB 10.1.x 内核时，使用 timestamp 类型不支持指定精度（例如 timestamp(3)），否则 DTS 任务会报错，需要去掉精度，然后重建任务。
13. 在任务过程中建议不要在目标端进行双写，否则会导致任务异常或者数据同步的结果不一致。

- 全量同步阶段：DTS 将全量初始数据写入目标端时检测到主键冲突，任务会报错。
- 增量同步阶段：DTS 将全量阶段的新增数据、增量阶段的新增数据写入到目标端时，检查到主键冲突，会用源端数据直接覆盖掉目标端。

操作限制

- 迁移过程中请勿进行如下操作，否则会导致迁移任务失败。
  - 请勿修改、删除源数据库和目标数据库中用户信息（包括用户名、密码和权限）和端口号。
  - 请勿在源库写入 Binlog 格式为 STATEMENT 的数据。
  - 请勿在源库上执行清除 Binlog 的操作。
  - 在增量迁移阶段，请勿删除系统库表 __tencentdb__ 。
- 增量迁移过程中，不支持源库新增分片、调整分片规格，否则迁移任务不会同步新增分片的数据或是任务报错暂停，如果需要长期保持增量同步并且支持源库的新增分片、调整分片操作，请使用“数据同步”来进行同步。同时在迁移过程中，建议用户不要单独在目标端写入数据，否则双写会导致数据同步的结果不一致，如果用户需要在目标端写入，也建议使用数据同步功能。

支持的 SQL 操作

操作类型	支持同步的 SQL 操作
DML	INSERT、UPDATE、DELETE、REPLACE
DDL	- TABLE：CREATE TABLE、ALTER TABLE、DROP TABLE、TRUNCATE TABLE - VIEW：CREATE VIEW、DROP VIEW - INDEX：CREATE INDEX、DROP INDEX - DATABASE：CREATE DATABASE、ALTER DATABASE、DROP DATABASE

操作类型	支持同步的 SQL 操作
	暂不支持 CREATE TABLE 表名 AS SELECT 语句。

操作步骤

1. 登录 DTS 控制台，在左侧导航选择数据迁移页，单击新建迁移任务，进入新建迁移任务页面。

2. 在新建迁移任务页面，选择迁移的源实例类型和所属地域，目标实例类型和所属地域，规格等，然后单击立即购买。

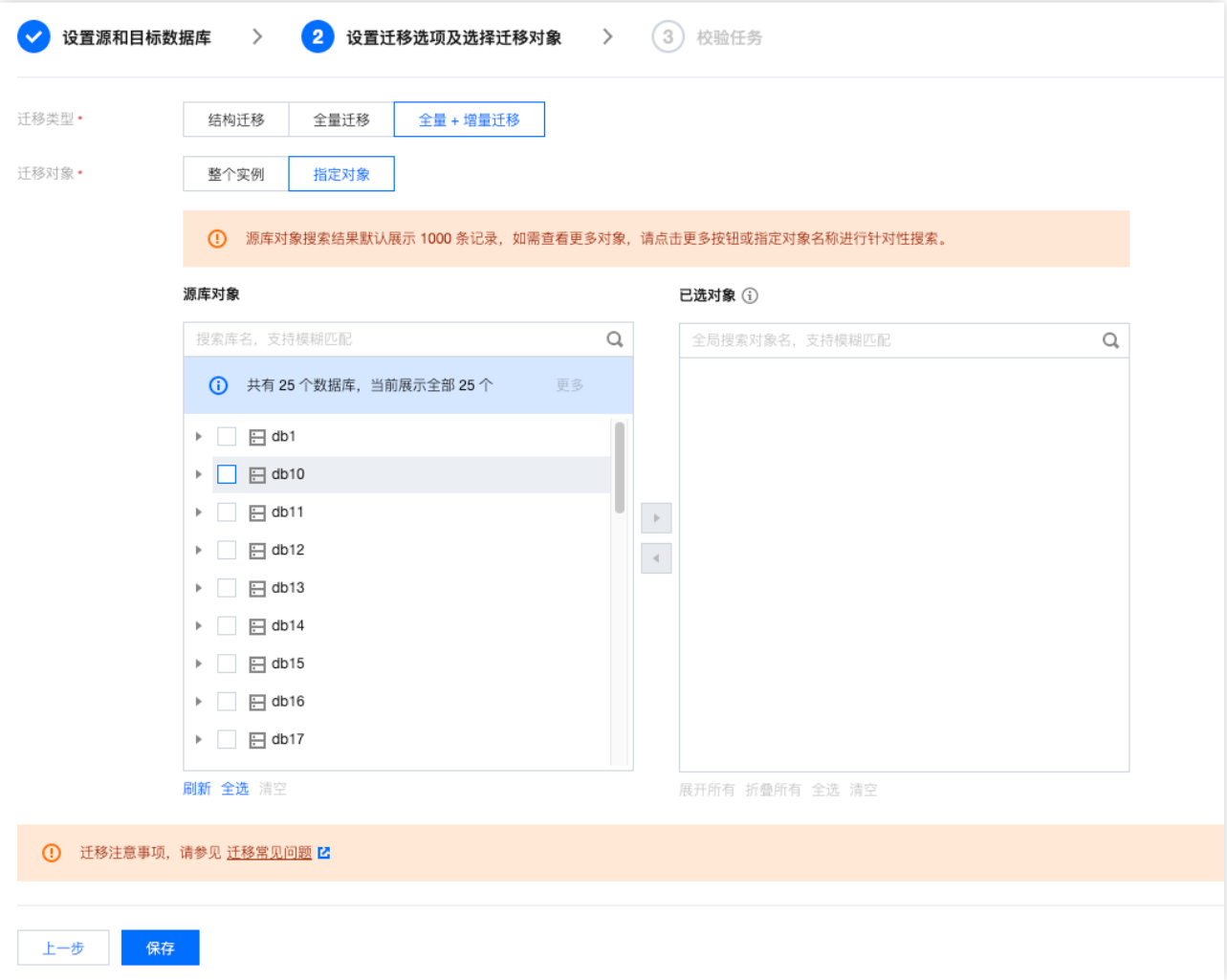
3. 在设置源和目标数据库页面，完成任务设置、源库设置和目标库设置，测试源库和目标库连通性通过后，单击新建。

设置类型	配置项	说明
任务设置	任务名称	设置一个具有业务意义的名称，便于任务识别。
	运行模式	支持立即执行和定时执行：立即执行，则完成任务校验通过后立即启动任务；定时执行，需要配置一个任务执行时间则到时间后启动任务。
	标签	标签用于从不同维度对资源分类管理。如现有标签不符合您的要求，请前往控制台管理标签。
源库设置	源库类型	购买时选择的源库类型，不可修改。
	所属地域	购买时选择的源库地域，不可修改。
	接入类型	选择“云数据库”。
	数据库实例	选择源数据库实例 ID。
	账号	源库 TDSQL MySQL 的数据库账号，账号权限需要满足要求。
	密码	源库 TDSQL MySQL 的数据库账号的密码。
目标库设置	目标库类型	购买时选择的目标库类型，不可修改。
	所属地域	购买时选择的目标库地域，不可修改。
	接入类型	选择“云数据库”。
	数据库实例	选择目标端 TDSQL MySQL 实例 ID。
	账号	目标端 TDSQL MySQL 的数据库账号，账号权限需要满足要求。
	密码	目标端 TDSQL MySQL 的数据库账号的密码。

4. 在设置迁移选项及选择迁移对象页面，设置迁移类型、对象，单击保存。

> 说明：

> 如果用户在迁移过程中确定会对某张表使用 rename 操作（例如将 table A rename 为 table B），则迁移对象需要选择 table A 所在的整个库（或者整个实例），不能仅选择 table A，否则 rename 操作后，table B 的数据不会同步到目标库。



配置项	说明
迁移类型	请根据您的场景选择。 - 结构迁移：迁移数据库中的库、表等结构化的数据。 - 全量迁移：迁移整个数据库，迁移数据仅针对任务发起时，源数据库已有的内容，不包括任务发起后源库实时新增的数据写入。 - 全量 + 增量迁移：迁移数据包括任务发起时源库的已有内容，也包括任务发起后源库实时新增的数据写入。如果迁移过程中源库有数据写入，需要不停机平滑迁移，请选择此场景。
迁移对象	- 整个实例：迁移整个实例，但不包括系统库，如information_schema、mysql、performance_schema、sys。 - 指定对象：迁移指定对象。
指定对象	在源库对象中选择待迁移的对象，然后将其移到已选对象框中。

5. 在校验任务页面，进行校验，校验任务通过后，单击启动任务。

如果校验任务不通过，可以参考“前置校验不通过处理”修复问题后重新发起校验任务。

- 失败：表示校验项检查未通过，任务阻断，需要修复问题后重新执行校验任务。
- 警告：表示检验项检查不完全符合要求，可以继续任务，但对业务有一定的影响，用户需要根据提示自行评估是忽略警告项还是修复问题再继续。

6. 返回数据迁移任务列表，任务进入创建中状态，运行1分钟 - 2分钟后，数据迁移任务开始正式启动。

- 选择结构迁移或者全量迁移：任务完成后会自动结束，不需要手动结束。
- 选择全量 + 增量迁移：全量迁移完成后会自动进入增量数据同步阶段，增量数据同步不会自动结束，需要您手动单击完成结束增量数据同步。
  - 请选择合适时间手动完成增量数据同步，并完成业务切换。
  - 观察迁移阶段为增量同步，并显示无延迟状态，将源库停写几分钟。
  - 目标与源库数据差距为0MB及目标与源库时间延迟为0秒时，手动完成增量同步。

7. （可选）如果您需要进行查看任务、删除任务等操作，请单击对应的任务，在操作列进行操作。

MySQL/MariaDB/Percona 迁移至 TDSQL MySQL

MySQL/MariaDB/Percona/TDSQL-C MySQL 迁移至云数据库 TDSQL MySQL 的要求和指导，与 TDSQL MySQL 迁移至 TDSQL MySQL 的内容基本一致，请参考相关内容进行操作。

迁移至 Percona

Percona 迁移至 Percona

从 Percona 迁移数据至 Percona 的操作指导，与 MariaDB/Percona 迁移到 MySQL 的使用说明和操作步骤一致，请参考 MySQL/MariaDB/Percona 迁移至 MariaDB 进行配置。

MySQL 系列前置校验不通过处理
连接 DB 检查

检查详情

源数据库和目标数据库需要能正常连通，如果未连通，会报连接失败。

问题原因

- 源数据库所在网络或服务器设置了安全组或防火墙。
- 源数据库对来源 IP 地址进行了限制。
- 网络端口未放通。
- 数据库账号或密码不正确。

修复方法

请按照问题原因中的对应链接进行处理。

周边检查

检查详情

1. 建议源数据库环境变量参数 innodb_stats_on_metadata 设置为 OFF 。
2. 目标库为 MySQL/MariaDB/Percona/TDSQL-C MySQL/TDSQL TDStore 时，建议目标库 explicit_defaults_for_timestamp 设置为 ON 。如果目标库为 OFF ，DTS 会尝试修改目标端上该参数设置为 ON ，仅在会话级别。
   - 目标端为 MySQL 8.0版本，DTS 可以成功修改目标端上会话级别的参数为 ON 。
   - 目标端为 MySQL 8.0以下的版本，或者其他数据库类型，DTS 无法修改成功，则初次校验任务不通过。用户可评估影响后跳过该参数检查，然后重新校验。

是否可跳过

出现校验不通过时，用户可评估影响后自行决定是否修改参数，如确认不需要修改，可跳过参数检查。

- innodb_stats_on_metadata ：可跳过校验。
- explicit_defaults_for_timestamp ：可跳过校验。

修复方法

修改 innodb_stats_on_metadata 参数

- innodb_stats_on_metadata 参数开启时，每当查询 information_schema 元数据库里的表，Innodb 就会更新 information_schema.statistics 表，导致访问时间变长。关闭后可加快对于 schema 库表的访问。
- MySQL 5.6.6 之前版本 innodb_stats_on_metadata 参数预设值为 ON，需要修改为 OFF。MySQL 5.6.6 及其以后的版本预设值为 OFF，不存在问题。

1. 登录源数据库。

2. 修改 innodb_stats_on_metadata 为 OFF 。

```
set global innodb_stats_on_metadata = OFF;
```

3. 查看配置是否生效。

```
show global variables like '%innodb_stats_on_metadata%';
```

系统显示结果类似如下：

```
mysql> show global variables like '%innodb_stats_on_metadata%';
+-------------------------+-------+
| Variable_name           | Value |
+-------------------------+-------+
| innodb_stats_on_metadata | OFF   |
+-------------------------+-------+
1 row in set (0.00 sec)
```

4. 重新执行校验任务。

修改 explicit_defaults_for_timestamp 参数

参数说明

explicit_defaults_for_timestamp 用于控制 MySQL 如何处理 TIMESTAMP 列的默认值。取值说明如下：

- OFF：MySQL 会将 TIMESTAMP 列的默认值设置为 CURRENT_TIMESTAMP。插入数据时，如果没有指定 TIMESTAMP 列的值，则该值会被设置为 CURRENT_TIMESTAMP。
- ON：MySQL 不会自动将 TIMESTAMP 列的默认值设置为 CURRENT_TIMESTAMP。

  参数影响

  该参数为 MySQL 5.6.6之后的版本特性，DTS 在迁移/同步过程中，为了保证数据的一致性，建议目标端设置为 ON；如果目标端为 OFF 并且未按照提示修改参数，则选择同步含有 TIMESTAMP 列的数据后，可能会导致源和目标的表结构不一致。

  如果评估后要修改参数，请参考如下命令。

  > 说明：
  >
  > 该参数修改需要重置数据库上的所有连接才能生效，建议不要在 session 级别设置该参数，否则可能无法查询到参数状态。

```
set global explicit_defaults_for_timestamp = ON
```

版本检查

检查详情

1. 建议目标数据库版本大于等于源数据库版本（如从源库5.6.x 到目标库5.7.x 及更高版本），否则版本检查会报警告，提示可能存在兼容性问题。警告不影响任务继续进行，用户评估无影响后，可以忽略警告继续后续任务操作。
2. 源和目标数据库的版本满足 DTS 支持的能力范围。

修复方法

请按照版本要求检查源库和目标库，如果源库或者目标库版本不支持，请升级目标实例版本或者使用更高版本的数据库实例。

源实例权限检查

检查详情

检查用户是否具备对数据库的操作权限，迁移/同步中，每个数据库类型的操作指导有详细的权限要求，请参考对应章节内容。

修复方法

用户若不具备操作权限，请按照检查要求中的对应权限要求对用户进行授权，然后重新执行校验任务。

部分实例参数检查

检查详情

- 源库表的 row_format 不能为 FIXED 。
- 源库和目标库 lower_case_table_names 变量必须一致。
- 目标库 max_allowed_packet 参数设置至少为4M。
- 源库变量 connect_timeout 必须大于10。
- 在 MySQL/TDSQL MySQL/TDSQL-C 迁移到 MySQL 的场景中，如果源实例存在耗时较长的 SQL 在运行，则会触发警告，提示“源实例有耗时较长的 SQL 在运行，可能导致锁表，请稍后重试或对源实例中的 SQL 进行处理”。

修复方法

修改源库 row_format 参数

数据库中表的 row_format 的取值为 FIXED 时，表格中每行的存储长度超过限制值时会溢出，发生报错。因此需要修改为其他模式，如 DYNAMIC ，使每行的存储长度会随内容的长度而变化。
如发生此类报错，请参考如下指导进行修复：

1. 登录源数据库。

2. 修改 row_format 参数为 DYNAMIC 。

```
alter table table_name row_format = DYNAMIC;
```

3. 查看配置是否生效。

```
show table status like 'table_name'\G;
```

系统显示结果类似如下：

```
mysql> show table status like 'table_name'\G;
*************************** 1. row ***************************
        Name: table_name
      Engine: InnoDB
     Version: 10
```

```
  Row_format: Dynamic
        Rows: 5
        ......
1 row in set (0.00 sec)
```

4. 重新执行校验任务。

修改源库和目标库 lower_case_table_names 变量保持一致

lower_case_table_names 是 MySQL 设置大小写是否敏感的一个参数，不同的取值情况如下：
Windows 或 macOS 环境对大小写是不敏感的，但是 Linux 环境却是敏感的，为了保证不同系统的兼容性，需要将大小写敏感规则设置统一。

- 0：表名存储为给定的大小写，比较的时候区分大小写。
- 1：表名存储在磁盘是小写的，比较的时候不区分大小写。
- 2：表名存储为给定的大小写，比较的时候是小写的。

如发生此类报错，请参考如下指导将源库和目标库的参数改为统一。

1. 登录源数据库。

2. 查看源库和目标库中的 lower_case_table_names 取值。

```
show variables like '%lower_case_table_names%';
```

3. 参考如下内容修改源数据库的配置文件 my.cnf 。

> 说明
>
> my.cnf 配置文件的默认路径为 /etc/my.cnf ，现场以实际情况为准。

```
lower_case_table_names = 1
```

4. 参考如下命令重启数据库。

```
[$Mysql_Dir]/bin/mysqladmin -u root -p shutdown
[$Mysql_Dir]/bin/safe_mysqld &
```

5. 查看配置是否生效。

```
show variables like '%lower_case_table_names%';
```

系统显示结果类似如下：

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

6. 重新执行校验任务。

修改目标库 max_allowed_packet 参数

max_allowed_packet 为最大允许的传输包。设置太大，会使用更多内存导致丢包，无法捕捉异常大事物包 SQL；设置太小，可能会导致程序报错，备份失败，也会导致频繁的收发网络包，影响系统性能。
如发生此类报错，请参考如下指导进行修复：

1. 登录目标数据库。

2. 修改 max_allowed_packet 参数。

```
set global max_allowed_packet = 4*1024*1024;
```

3. 查看配置是否生效。

```
show global variables like '%max_allowed_packet%';
```

系统显示结果类似如下：

```
mysql> show global variables like '%max_allowed_packet%';
+-----------------------+---------+
| Variable_name         | Value   |
+-----------------------+---------+
| max_allowed_packet    | 4194304 |
+-----------------------+---------+
1 row in set (0.00 sec)
```

4. 重新执行校验任务。

修改源库变量 connect_timeout

connect_timeout 为数据库的连接时间，超过 connect_timeout 设置值的连接请求将会被拒绝。如果设置过小，会导致数据库连接频繁断开，影响处理效率，因此建议该参数取值大于10。
如发生此类报错，请参考如下指导进行修复：

1. 登录源数据库。

2. 修改 connect_timeout 参数。

```
set global connect_timeout = 10;
```

3. 查看配置是否生效。

```
show global variables like '%connect_timeout%';
```

系统显示结果类似如下：

```
mysql> show global variables like '%connect_timeout%';
+-----------------------+---------+
| Variable_name         | Value   |
+-----------------------+---------+
| connect_timeout       | 10      |
+-----------------------+---------+
1 row in set (0.00 sec)
```

4. 重新执行校验任务。

目标实例权限检查

检查详情

1. 检查用户是否具备对数据库的操作权限，在迁移/同步中，每个数据库类型的操作指导有详细的权限要求，请参考对应章节内容。
2. 目标端为 MySQL8.0.26及之后版本，DTS 会检查 innodb_strict_mode 参数，建议在迁移/同步任务阶段将 innodb_strict_mode 参数设置为 OFF，若不为 OFF，DTS 会在前置校验给出提示。目标端为 MySQL8.0.26 之前的版本，或者其他数据库类型时，DTS 会对该参数进行 session 级别的修改，改为 OFF。
   - 参数说明：innodb_strict_mode 用于控制 InnoDB 存储引擎对一些不规范数据是否启用严格模式。
     - innodb_strict_mode=ON，InnoDB 存储引擎会执行严格的数据验证模式。如创建表（CREATE TABLE）、更改表（ALTER TABLE）和创建索引（CREATE INDEX）时，语法有错误，直接抛出错误。
     - innodb_strict_mode=OFF，InnoDB 存储引擎对数据验证放宽，遇到如上的错误语法不会报错，并且使用默认的语法替代错误的语法。
   - 设置建议：如果源端数据库本身有不规范的数据，建议在迁移/同步任务阶段将 innodb_strict_mode 参数设置为 OFF，这样可以将数据可以正常同步到目标端。

修复方法

用户若不具备操作权限，请按照检查要求中的对应权限要求对用户进行授权，然后重新执行校验任务。

目标实例内容冲突检查

检查详情

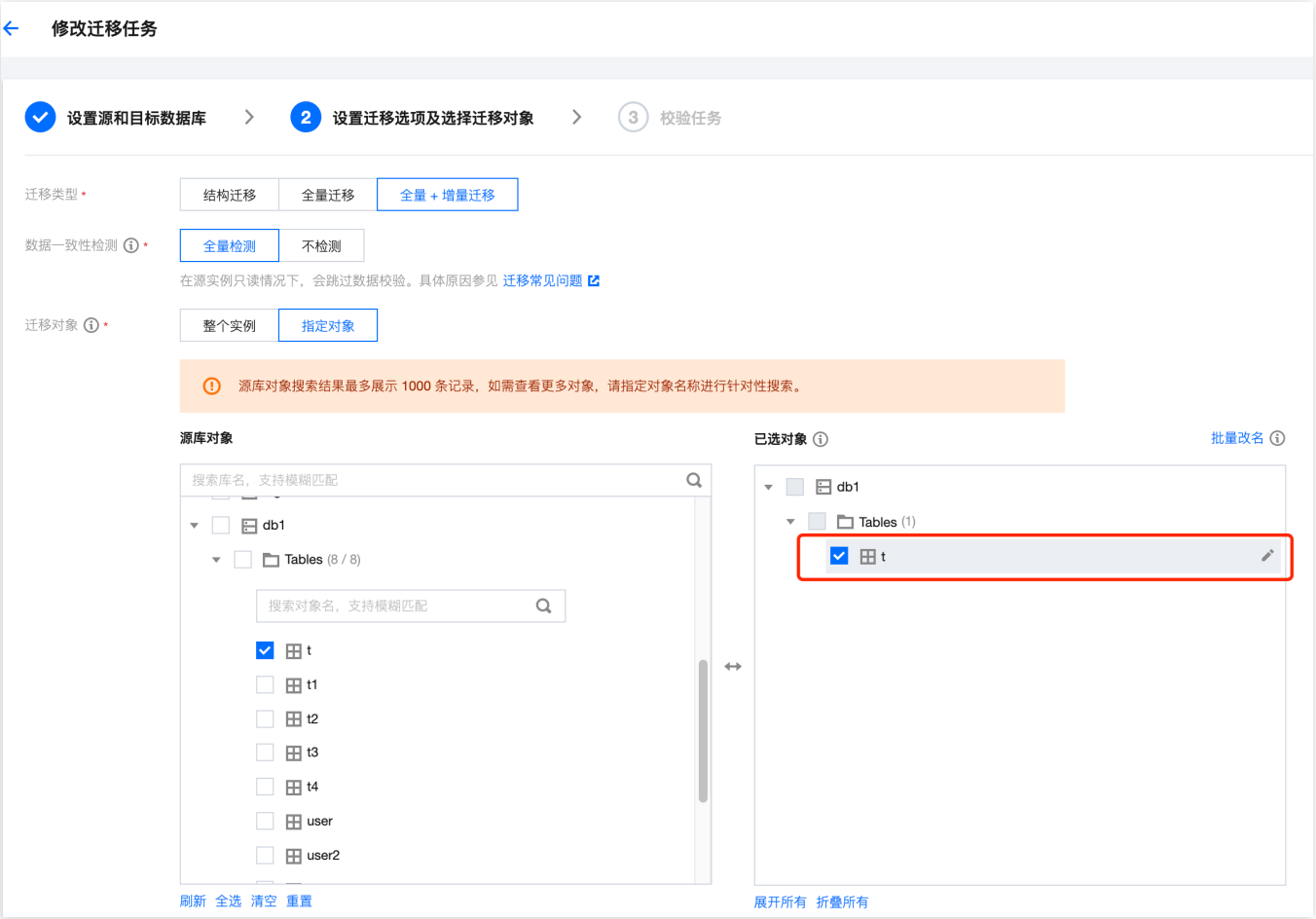
- 目标实例不能有和源库同名的对象。如果存在冲突报错，可任选以下一个方法进行修复。
  - 方法一：使用库表映射。
  - 方法二：修改或删除目标数据库中的同名对象。
  - 方法三：从迁移对象中移除同名对象。
- 整个实例迁移时，目标实例必须为空。如果存在冲突报错，需要删除实例内容。
- 选择高级对象时， 目标库不能有冲突的高级对象。如果存在冲突报错，需要删除冲突的对象。

修复方法

使用库表映射

使用 DTS 库表映射功能，将同名的待迁移对象映射为目标数据库中的其他名称。

1. 登录 DTS 控制台，选择对应的迁移任务，在操作列，选择更多 > 修改。
2. 在选择迁移对象右侧“已选对象”中，将鼠标悬浮在需要修改的对象上即可显示编辑按钮，然后重命名对象。



3. 重新执行校验任务。

修改目标数据库中的同名对象

登录目标数据库，重命名或删除目标数据库中和迁移对象同名的对象。

从迁移对象中移除同名对象

修改迁移任务配置，从迁移对象中移除同名对象，该对象不进行数据迁移。

1. 登录 DTS 控制台，选择对应的迁移任务，在操作列，选择更多 > 修改。
2. 在迁移对象中，移除同名的对象。
3. 重新执行校验任务。

删除目标库中的内容

登录目标数据库，删除目标数据库中的同名对象或者整库内容，然后重新执行校验任务。

目标实例空间检查

检查详情

目标库存储空间需要在源库待迁移库表空间的1.2倍以上。
全量数据迁移会并发执行 INSERT 操作，导致目标数据库的表产生碎片，因此全量迁移完成后目标数据库的表存储空间很可能会比源实例的表存储空间大。

修复方法

- 删除目标库中的部分数据，以便腾出足够的空间。
- 升级目标库存储规格，使用更大容量的实例进行迁移。

Binlog 参数检查

检查详情

源数据库 binlog 相关参数需要按照如下要求配置，如果校验不通过，请参考本文后续指导进行修复。

- log_bin 变量必须设置为 ON 。
- binlog_format 变量必须设置为 ROW 。
- binlog_row_image 必须设置为 FULL 。
- 如果源数据库为 MySQL 5.6 及以上版本， gtid_mode 只支持设置为 ON 和 OFF ，建议将 gtid_mode 设置为 ON ，设置为 OFF 会报警告，设置为 ON_PERMISSIVE 和 OFF_PERMISSIVE 会报错。
- server_id 参数需要手动设置，且值不能设置为0。
- 不允许设置 do_db ， ignore_db 。
- 对于源实例为从库的情况， log_slave_updates 变量必须设置为 ON 。
- 建议源库 Binlog 日志至少保留3天及以上，否则可能会因任务暂停/中断时间大于 Binlog 日志保留时间，造成任务无法续传，进而导致任务失败。

修复方法

开启 binlog

log_bin 是 binlog 的开关控制参数，需要将 binlog 打开，以便记录所有的数据库表结构和表数据变更日志。
如发生类似报错，请参考如下指导进行修复：

1. 登录源数据库。

2. 参考如下内容修改源数据库的配置文件 my.cnf 。
   因 log_bin 参数修改后需要重启数据库后才能生效，如果 binlog_format、server_id、binlog_row_image、expire_logs_days 这几个参数也在校验阶段提示报错，请一并修改完成后再重启数据库，让所有参数都生效。

   > 说明：
   >
   > my.cnf 配置文件的默认路径为 /etc/my.cnf ，现场以实际情况为准。

```
log_bin = MYSQL_BIN
binlog_format = ROW
server_id = 2     //建议设为大于1的整数，此处仅为示例
binlog_row_image = FULL
expire_logs_days = 3     //修改 binlog 的保留时间，建议大于等于3天
```

3. 参考如下命令重启源数据库。

```
[$Mysql_Dir]/bin/mysqladmin -u root -p shutdown
[$Mysql_Dir]/bin/safe_mysqld &
```

> 说明：
>
> [$Mysql_Dir] 指源数据库的安装路径，请替换为实际的源数据库安装目录。

4. 确认 binlog 功能是否已启用。

```
show variables like '%log_bin%';
```

系统显示结果类似如下：

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

请使用如下对应的命令查看其他参数的修改结果。
show variables like '%binlog_format%';
show variables like '%binlog_row_image%';
show global variables like '%server_id%';

5. 重新执行校验任务。

修改 binlog_format 参数

binlog_format 为 binlog 的记录模式，有以下三种：

- STATEMENT ：每一条会修改数据的 SQL 都会记录到 master 的 binlog 中，slave 在复制的时候，会执行和原来 master 端相同的 SQL。该模式可以减少 binlog 日志量，但是对某些特定的函数进行复制时，slave 端不能正确复制。
- ROW ：binlog 日志中会记录成每一行数据修改的形式，然后在 slave 端再对相同的数据进行修改。该模式会保证 master 和 slave 的正确复制，但是 binlog 日志量会增加。
- MIXED ：前两种模式的结合，MySQL 会根据执行的每一条具体的 SQL 语句来区分对待记录的日志形式，在 STATEMENT 和 ROW 之间选择一种。

综上，为了保证 master 和 slave 的正确复制， binlog_format 参数需要设置为 ROW 。如发生类似报错，

请参考如下指导进行修复。

> 说明：
>
> 该参数修改需要重置数据库上的所有连接才能生效，当源库为从库时，还需重启主从同步 SQL 线程，避免当前业务连接继续使用修改前的模式写入。

1. 登录源数据库。

2. 参考如下命令修改 binlog_format 。

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

3. 重启线程使配置生效，然后通过如下命令查看参数修改是否生效。

```
show variables like '%binlog_format%';
```

系统显示结果类似如下：

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

4. 重新执行校验任务。

修改 binlog_row_image 参数

binlog_row_image 参数决定了 binlog 是如何记录前镜像（记录修改前的内容）和后镜像（记录修改后的内容）的，这会直接影响到数据闪回、主从复制等功能。
binlog_row_image 参数只在 binlog_format 配置为 ROW 模式下生效。具体取值影响如下：

- FULL ：在 ROW 模式下，binlog 会记录前镜像和后镜像的所有列的数据信息。
- MINIMAL ：在 ROW 模式下，当表没有主键或唯一键时，前镜像记录所有列，后镜像记录被修改的列；如果存在主键或唯一键，不管是前镜像还是后镜像，都只记录有影响的列。

  综上， binlog_row_image 需要配置为 FULL ，源数据库的 binlog 记录全镜像。如发生报错，请参考如下步骤进行修复。

  > 说明：

> 该参数修改需要重置数据库上的所有连接才能生效，当源库为从库时，还需重启主从同步 SQL 线程，避免当前业务连接继续使用修改前的模式写入。

1. 登录源数据库。

2. 参考如下内容修改 binlog_row_image 。

```
set global binlog_row_image = FULL;
```

3. 重启线程使配置生效，然后通过如下命令查看参数修改是否生效。

```
show variables like '%binlog_row_image%';
```

系统显示结果类似如下：

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

4. 重新执行校验任务。

修改 gtid_mode 参数

GTID（Global Transaction Identifier， 全局事务标识），用于在 binlog 中唯一标识一个事务，使用 GTID 可以避免事务重复执行导致数据混乱或者主从不一致。
GTID 是 MySQL 5.6 的新特性，所以 MySQL 5.6 及之后版本存在此问题。DTS 只支持 gtid_mode 设置为 ON 和 OFF ，建议将 gtid_mode 设置为 ON ，否则校验时会报警告。
警告不影响迁移或同步任务进行，但是会对业务造成一定的影响：设置 GTID 后，在增量数据同步阶段，如果源实例发生 HA 切换，DTS 服务切换重连，任务几乎无感知；反之，任务会中断后失败，且不可恢复。
gtid_mode 的取值如下，在修改 gtid_mode 的值时，只能从以下四个值逐级修改，例如，需要从 OFF 修改为 ON ，则 gtid_mode 修改顺序为 OFF <-> OFF_PERMISSIVE <-> ON_PERMISSIVE <-> ON 。

- OFF ：主库所有新启的事务以及从库的事务都要求是匿名事务。
- OFF_PERMISSIVE ：主库新启的事务是匿名事务，但从库事务允许是匿名的或者是 GTID 事务，但不允许只是 GTID 模式。
- ON_PERMISSIVE ：主库新启的事务是 GTID 事务，从库事务允许是匿名的或者是 GTID 事务。

- ON ：主库新启的事务是 GTID 事务，从库的事务也要求是 GTID 事务。

如果发生类似警告，请按照如下指导进行修复：

1. 登录源数据库。

2. 在主从复制结构两边都设置 gtid_mode = OFF_PERMISSIVE 。
MySQL 5.7.6 之前的版本需要在 my.cnf 配置文件中修改并重启数据库才能生效，5.7.6 及之后的版本通过全局命名修改，不需要重启数据库，但是需要重置所有业务连接。

```
set global gtid_mode = OFF_PERMISSIVE;
```

3. 在主从复制结构两边都设置 gtid_mode = ON_PERMISSIVE 。

```
set global gtid_mode = ON_PERMISSIVE;
```

4. 在各个实例节点上执行如下命令，检查匿名事务是否消耗完毕，参数值为 0 则代表消耗完毕。

```
show variables like '%ONGOING_ANONYMOUS_TRANSACTION_COUNT%';
```

系统显示结果类似如下：

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

5. 在主从复制结构两边都设置 gtid_mode = ON 。

```
set global gtid_mode = ON;
```

6. 在 my.cnf 文件中添加如下内容，后续重启数据库后使初始值生效。

> 说明：
>
> my.cnf 配置文件的默认路径为 /etc/my.cnf ，现场以实际情况为准。

```
gtid_mode = on
enforce_gtid_consistency = on
```

7. 重新执行校验任务。

修改 server_id 参数

server_id 参数需要手动设置，且值不能设置为0。该参数系统预设值为 1 ，如果查询该参数显示为 1 不一定正确，需要手动进行配置。

1. 登录源数据库。

2. 参考如下内容修改 server_id 。

```
set global server_id = 2;  //建议设为大于1的整数，此处仅为示例
```

3. 通过如下命令查看参数修改是否生效。

```
show global variables like '%server_id%';
```

系统显示结果类似如下：

```
mysql> show global variables like '%server_id%';
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| server_id     | 2     |
+---------------+-------+
1 row in set (0.00 sec)
```

4. 重新执行校验任务。

删除 do_db，ignore_db 设置

binlog 会记录数据库所有执行的 DDL 和 DML 语句，而 do_db，ignore_db 则是设置 binlog 记录的过滤条件。

- binlog_do_db ：只记录指定数据库的二进制日志，默认全部记录。
- binlog_ignore_db ：不记录指定的数据库的二进制日志。

  设置 do_db，ignore_db 后，会导致一些跨库操作 binlog 记录不全，主从复制出现异常，因此不建议设置。

  如发生类似报错，请参考如下指导进行修复：

1. 登录源数据库。

2. 修改源数据库的配置文件 my.cnf ，删除 do_db，ignore_db相关设置。

> 说明：
>
> my.cnf 配置文件的默认路径为 /etc/my.cnf ，现场以实际情况为准。

3. 参考如下命令重启源数据库。

```
[$Mysql_Dir]/bin/mysqladmin -u root -p shutdown
[$Mysql_Dir]/bin/safe_mysqld &
```

> 说明：
>
> [$Mysql_Dir] 指源数据库的安装路径，请替换为实际的源数据库安装目录。

4. 确认参数修改是否生效。

```
show master status;
```

系统显示结果类似如下：

```
mysql> show master status;
+---------------+----------+--------------+------------------+-------------------+
| File          | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
+---------------+----------+--------------+------------------+-------------------+
| binlog.000011 | 154      |              |                  |                   |
+---------------+----------+--------------+------------------+-------------------+
```

5. 重新执行校验任务。

修改 log_slave_updates 参数

在主从复用结构中，从库开启 log-bin 参数，直接在从库操作数据时，可以记录在 binlog 中，但是从库从主库上复制数据时，不能记录在 binlog 中，所以从库作为其他从库的主库时，需要打开 log_slave_updates 参数。

1. 登录源数据库。

2. 在源数据库的配置文件 my.cnf 中增加如下内容。

> 说明：
>
> my.cnf 配置文件的默认路径为 /etc/my.cnf ，现场以实际情况为准。

```
log_slave_updates = ON
```

3. 参考如下命令重启源数据库。

```
[$Mysql_Dir]/bin/mysqladmin -u root -p shutdown
[$Mysql_Dir]/bin/safe_mysqld &
```

> 说明：
>
> [$Mysql_Dir] 指源数据库的安装路径，请替换为实际的源数据库安装目录。

4. 查看配置是否生效。

```
show variables like '%log_slave_updates%';
```

系统显示结果类似如下：

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

5. 重新执行校验任务。

外键依赖检查

检查详情

- MySQL/MariaDB/Percona/TDSQL-C MySQL/TDSQL MySQL/TDSQL TDStore 之间的数据迁移：DTS 默认支持 RESTRICT，NO ACTION，如果源库中存在 CASCADE、SET NULL、SET DEFAULT 这三种外键数据，也可以迁移，需要在校验项中开启外键依赖开关。
- MySQL/MariaDB/Percona/TDSQL-C MySQL/TDSQL MySQL/TDSQL TDStore 之间的数据同步：外键依赖只能设置为 NO ACTION、RESTRICT，设置为其他校验将报错，但用户可选择修复报错或者忽略报错继续任务。
- 部分库表迁移时，有外键依赖的表必须齐全。

外键依赖参数说明

MySQL 系类型的数据库在设置外键时，删除和更新有四个值可以选择。

- CASCADE ：父表进行删除或者更新记录时，子表会同步删除或更新关联记录。
- SET NULL ：父表进行删除或者更新记录时，子表会将关联记录的外键字段所在列设为 null （子表外键不能设为 not null ）。
- RESTRICT ：父表执行删除或更新记录时，如果子表中有关联该父表的记录，则拒绝该父表删除请求。
- NO ACTION ：同 RESTRICT，也是首先检查外键。
- SET DEFAULT ：父表进行删除或者更新记录时，子表将外键列设置成一个默认的值，但 InnoDB 引擎不能识别。

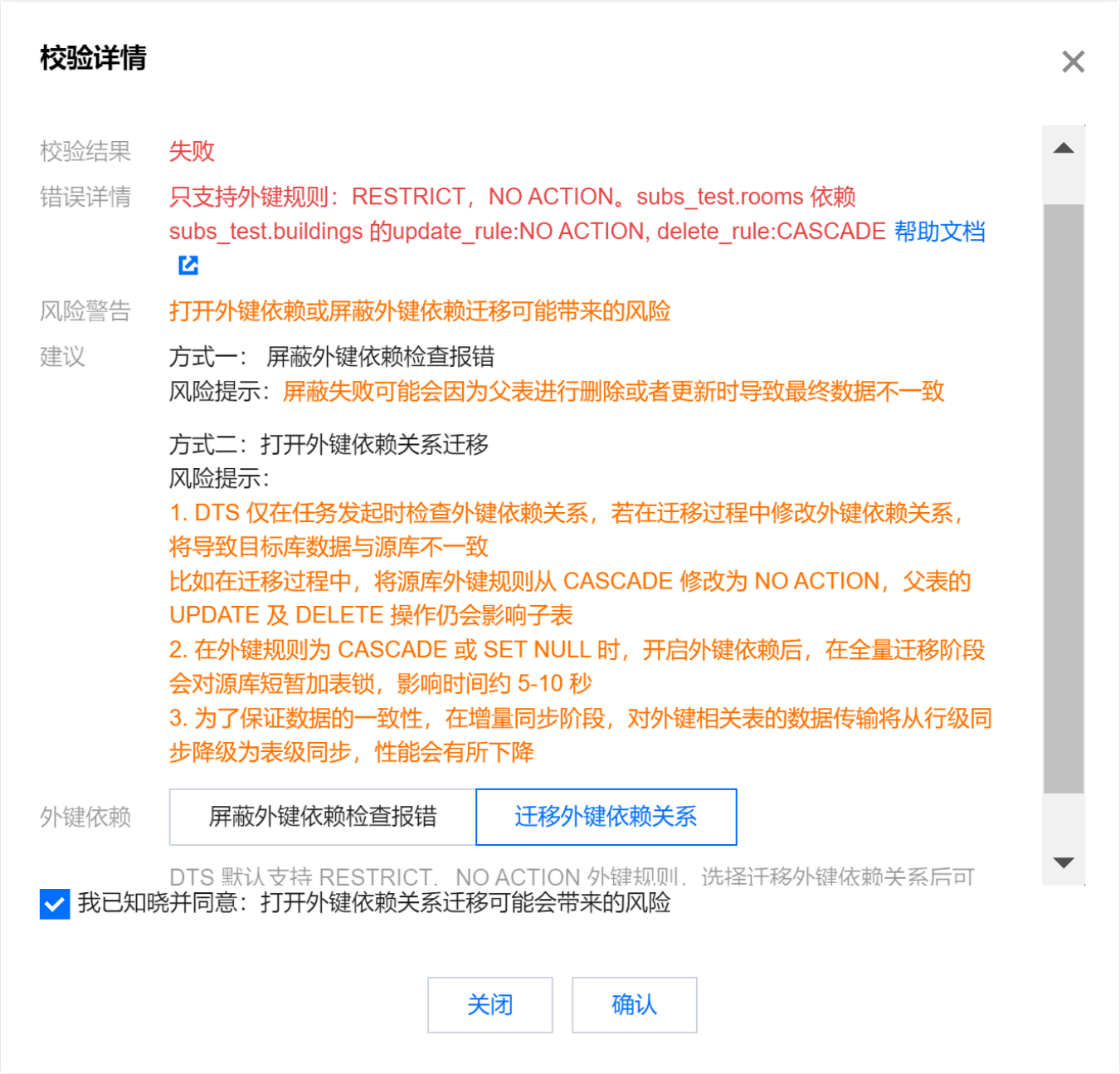
迁移场景-开启外键依赖关系迁移

DTS 默认支持 RESTRICT，NO ACTION，如果源数据库迁移的数据中存在其他类型的外键依赖配置，校验系统会提示报错，需要用户按照如下指导在校验项中开启外键依赖。

1. 在数据迁移操作中的校验任务页面，源库存在外键依赖规则 CASCADE、SET NULL、SET DEFAULT，校验项报错，查看校验项详情。

2. 根据情况选择忽略报错，或者迁移外键依赖关系。确认无误后勾线风险知会，单击确认。

选项	说明
屏蔽外键依赖检查报错	勾选后忽略已有的校验报错，继续迁移任务。 源库中存在外键规则 CASCADE、SET NULL、SET DEFAULT，仍可以进行迁移，但屏蔽报错后可能出现因为父表进行删除或者更新操作，导致目标库和源库的数据结果不一致。
迁移外键依赖关系	勾选后支持迁移 CASCADE、SET NULL、SET DEFAULT 类型的外键依赖关系，但 DTS 仅在任务发起时检查外键依赖关系，保证存量数据迁移到目标库的一致性。 1. 任务迁移过程中不要修改源库中的外键依赖规则，否则将导致目标库数据与源库不一致。如在迁移过程中，将源库外键规则从 CASCADE 修改为 NO ACTION，目标库中父表的更新及删除操作仍会影响子表。 1. 在外键规则为 CASCADE 或 SET NULL 时，勾选迁移外键依赖关系后，在全量迁移阶段会对源库短暂加表锁，影响时间约 5-10 秒。 1. 为了保证数据的一致性，在增量同步阶段，对外键相关表的数据传输将从行级同步降级为表级同步，性能会有所下降。



3. 重新执行校验任务。

同步场景-修改外键参数

数据同步中，外键依赖只能设置为 NO ACTION、RESTRICT，设置为其他校验将报错，但用户可选择修复报错或者忽略报错继续任务，如需修复报错请如下指导操作。

1. 登录源数据库。

2. 删除原来的外键设置。

```
alter table `表名称1` drop foreign key `外键名称1`;
```

3. 重新添加外键设置。

```
alter table `表名称1` add constraint `外键名称2` foreign key `表名称1`(`列名1`) references `表名称2`(`列名1`)
on update no action on delete no action;
```

4. 重新执行校验任务。

完善迁移对象

修改迁移任务配置，在迁移对象中勾选具有关联关系的对象。

1. 在 DTS 控制台，选择对应的迁移任务，在操作列选择更多 > 修改。
2. 在迁移对象中勾选具有关联关系的对象。
3. 重新执行校验任务。

视图检查

检查详情

- 检查要求：在导出视图结构时，DTS 会检查源库中 DEFINER 对应的 user1（ [DEFINER = user1]）和执行任务账号 user2 是否一致。
  - 如果一致则迁移后不做改动。
  - 如果不一致，则迁移后修改 user1 在目标库中的 SQL SECURITY 属性，由 DEFINER 转换为 INVOKER（ [INVOKER = user1]），同时设置目标库中 DEFINER 为执行任务账号 user2（[DEFINER = 执行任务账号 user2]）。
- 检查说明： SQL SECURITY 参数用来表示用户访问指定视图时，系统按照谁的权限来执行。
  - DEFINER ：表示只有定义者才能执行。
  - INVOKER ：表示拥有权限的调用者可以执行。

  默认情况下，系统指定为 DEFINER 。

高级对象检查

MySQL/MariaDB/Percona 检查详情

选择迁移/同步高级对象时，DTS 会对如下内容进行校验。报错项必须要处理才能继续任务，警告项用户评估业务风险后可忽略，继续任务。

- 报错项：目标实例参数 log_bin_trust_function_creators 必须为 ON。
- 警告项：
  - 迁移/同步高级对象与库表重命名功能冲突，选择高级对象后需要取消库表重命名。
  - 选择高级对象的函数，存储过程时，DTS 会检查源库中 DEFINER 对应的 user1（[DEFINER = user1]）和执行任务账号 user2 是否一致。
    - 如果一致则迁移/同步后不做改动。
    - 如果不一致，则迁移/同步后修改 user1 在目标库中的 SQL SECURITY 属性，由 DEFINER 转换为 INVOKER（[INVOKER = user1]），同时设置目标库中 DEFINER 为执行任务账号的 user2（[DEFINER = 执行任务账号user2]）。
  - 高级对象的迁移/同步时间：
    - 存储过程和函数，在“源库导出”阶段进行迁移/同步。
    - 触发器和事件，没有增量任务，在任务结束时进行迁移/同步；有增量任务，在用户单击完成操作后开始迁移/同步，所以单击完成后任务的过渡时间会长一些。

修复方法

修改 log_bin_trust_function_creators 参数。
log_bin_trust_function_creators 用于控制是否信任用户将存储函数写入 binlog 日志中。设置为 OFF ，仅 SUPER 权限的用户可将创建的存储函数操作写入 binlog 日志，设置为 ON ，非 SUPER 权限的用户也可将创建的存储函数操作写入 binlog 日志中。
发生报错时，请参考如下步骤进行修改。

1. 登录源数据库。

2. 参考如下内容修改 log_bin_trust_function_creators 参数。

```
set global log_bin_trust_function_creators = ON;
```

3. 通过如下命令查看参数修改是否生效。

```
show variables like '%log_bin_trust_function_creators%';
```

系统显示结果类似如下：

```
mysql>  show variables like '%log_bin_trust_function_creators%';
+--------------------------------+-------+
| Variable_name                  | Value |
+--------------------------------+-------+
| log_bin_trust_function_creators | ON    |
```

4. 重新执行校验任务。

警告项检查

检查详情

如下参数需要按照要求配置，否则校验时系统会发出警告，警告不影响迁移任务的进行，但是会对业务造成一定的影响，请用户评估后自行决定是否修改。

- 建议目标库 max_allowed_packet 的取值大于源库。
  - 业务影响：目标库的 max_allowed_packet 参数设置小于源库，会导致目标库数据无法写入，从而造成全量迁移失败。
  - 处理建议：修改目标库的 max_allowed_packet 参数，大于源库取值。
- 建议目标库的 max_allowed_packet 设置大于1GB。
  - 业务影响： max_allowed_packet 设置太大，会使用更多内存导致丢包，无法捕捉异常大事务包SQL；设置太小，可能会导致程序报错，备份失败，也会导致频繁的收发网络报，影响系统性能。
  - 处理建议：参考如下命令修改 max_allowed_packet 参数。

    ```
    set global max_allowed_packet = 1GB
    ```
- 建议源库和目标库的字符集保持一致。
  - 业务影响：源库和目标库的字符集不一致可能会导致乱码。
  - 处理建议：参考如下命令将源库和目标库字符集修改为一致。

    ```
    set character_set_server = 'utf8';
    ```
- 建议使用2CPU，4G Mem以上规格的实例。
- 如果仅执行全量数据迁移，请勿在迁移过程中向源实例中写入新的数据，否则会导致源和目标数据不一致。针对有数据写入的场景，为实时保持数据一致性，建议选择全量+增量数据迁移。
- 有锁导出时：源实例需要使用 Flush Table With Read Lock 短暂加锁，其中的 MyISAM 表会锁定到全量数据导出完成。当前等待加锁超时时间设置为60秒, 该时间内无法获取锁将导致任务失败。
- 无锁导出时：仅对没有主键的表会加读锁，其他不加锁。
- 对于既没有主键、也没有非空唯一键的表，有数据重复的风险，建议用户对迁移的表设置主键，或者非空唯一键。
- 对于 TDSQL MySQL 的数据迁移，当源数据库实例为分布式数据库时，需要提前在目标库建立分表，否则这些表被迁移后都将是单表。
- 目标库为 MySQL/MariaDB/Percona/TDSQL-C MySQL/TDSQL TDStore 时，需要检查源库和目标库

explicit_defaults_for_timestamp 的设置。源库为 OFF ，或者源和目标都为 ON 时，任务警告，提醒用户任务运行期间不要随便修改该参数。

- 在全量导出库表结构时检查表的 COLUMN_DEFAULT ， IS_NULLABLE 属性，如果源库表的 COLUMN_DEFAULT 为 NULL ， IS_NULLABLE 为 NOT NULL ，则该表结构不进行迁移/同步，避免数据同步到目标库后，MySQL 系统可能会自动为 TIMESTAMP 的数据类型添加默认参数“CURRENT_TIMESTAMP”。

TDSQL Proxy 检查

检查详情

目标库 TDSQL MySQL 使用 Proxy 方式连接的场景中，DTS 在目标库执行 SQL 的时候，是将多条语句放到一个 SQL 里去执行，需要将 gateway.mode.multi_query.open 参数设置为1，以支持多个 SQL 合并执行。参数如未打开，预校验任务会报错。

修复方法

请按照要求修改参数 进行修改。

TDSQL 内核检查

检查详情

源端为 TDSQL MySQL 时，检查源端的内核版本，不支持 mariadb 内核。如果是 mariadb 内核，预校验任务会报错。

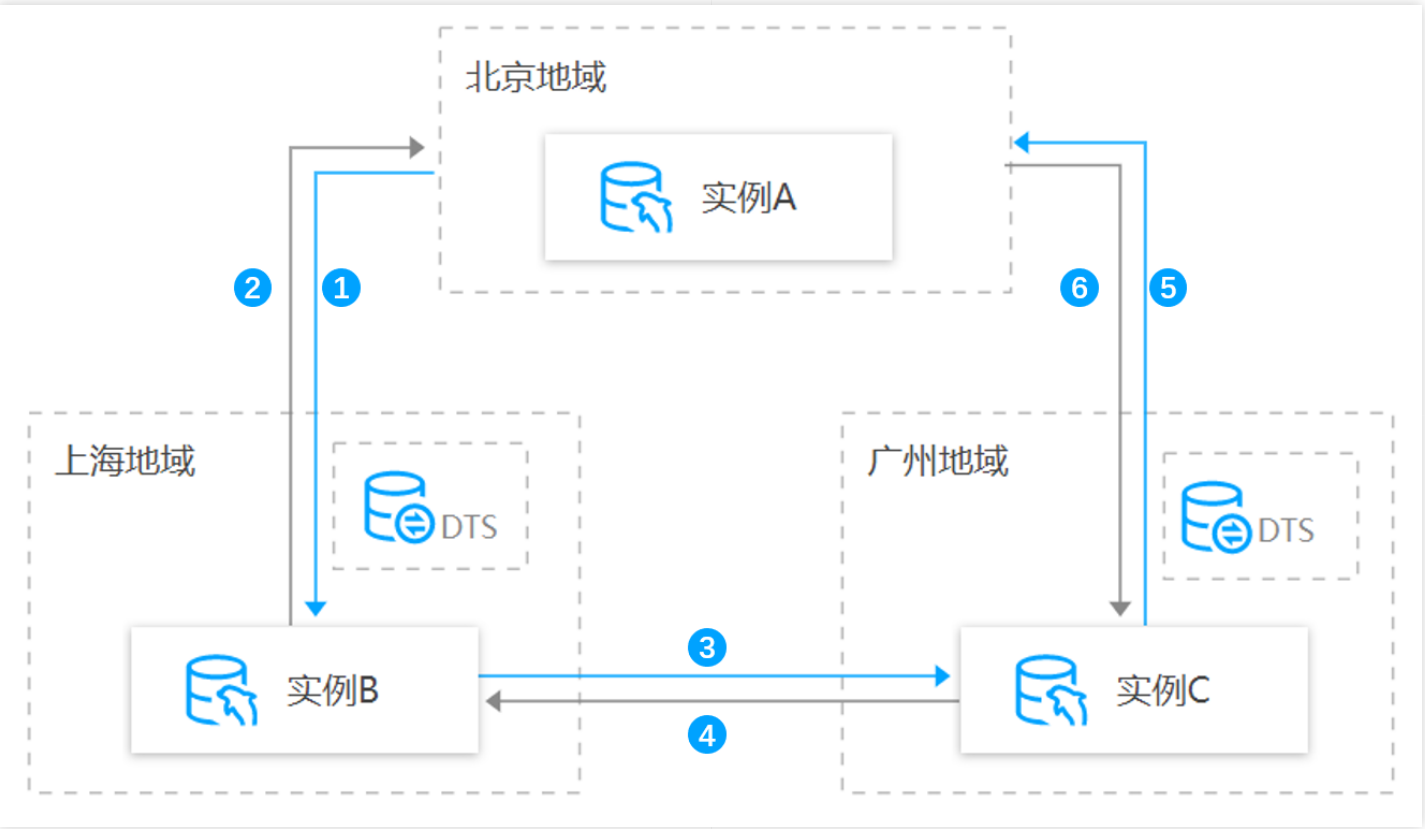
修复方法

无法同步 TDSQL MySQL（mariadb 内核），请选择其他版本进行同步。

同库表对象 DDL 环形链路同步检测

检查详情

在双向同步、多对一同步、多活等需要配置多个同步任务的场景中，DDL 的配置不能形成环形链路，否则可能造成 DDL 语句在系统中循环，进而引发错误。
示例：下图中蓝色线条1、3、5三个同步任务中，最多只能在两个同步任务中选择 DDL，如果选择三个就构成环形链路了。



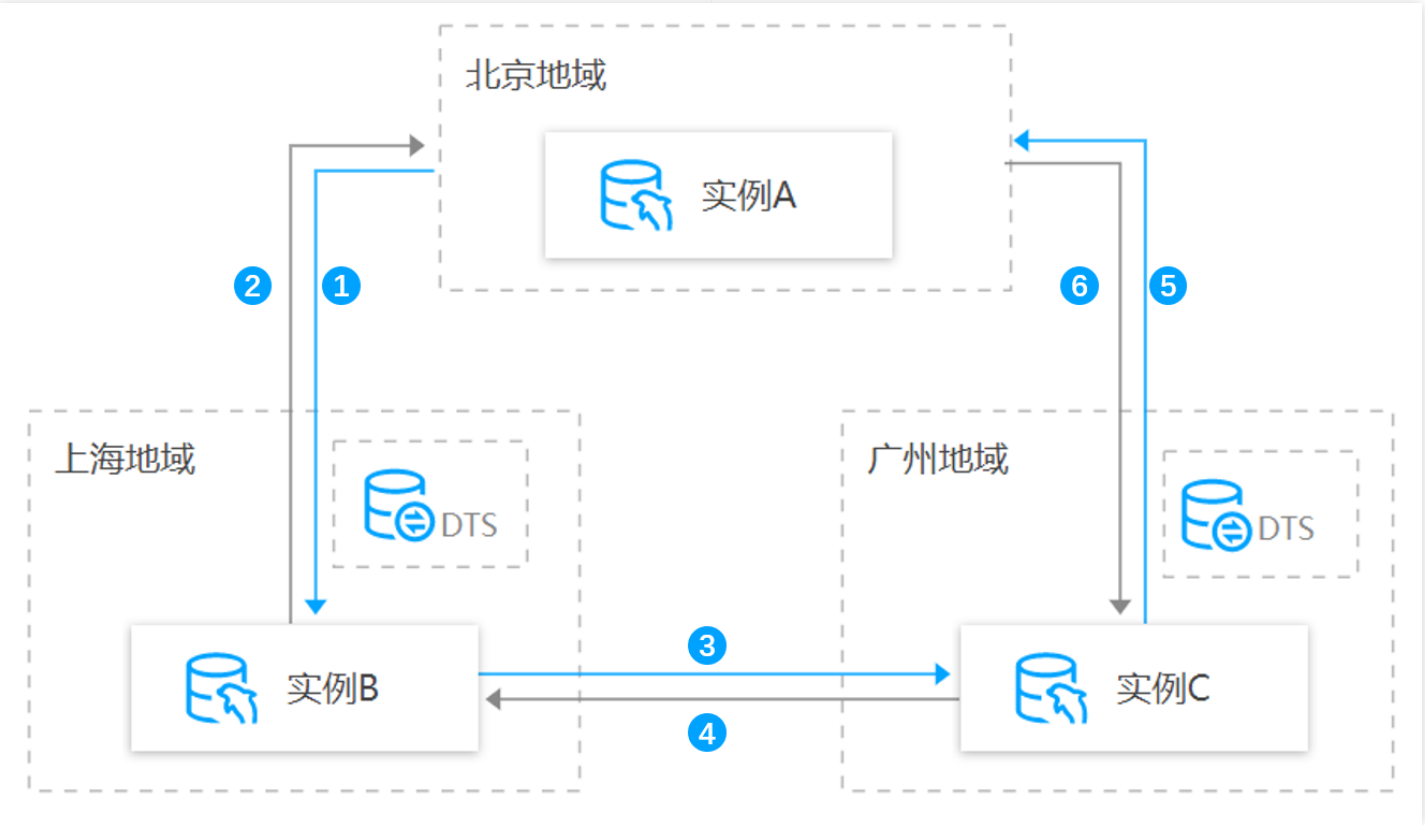
修复方法

修改同步任务配置，在设置同步选项和同步对象 > 数据同步选项 > 同步操作类型中，修改 DDL 参数配置，避免形成环形链路。

同目标同库表对象 DDL 同步冲突检测

检查详情

在多对一同步、多活等需要配置多个同步任务的场景中，同一个库表对象不能接收多个数据中心的 DDL 同步，否则多个数据中心的 DDL 可能在目标端形成冲突，进而引发错误。
示例：下图中实例 A 和 C 有相同名称的表需要同步到实例 B 中，则同步任务1和4中，只能在一个任务选择 DDL。



修复方法

修改同步任务配置，在设置同步选项和同步对象 > 数据同步选项 > 同步操作类型中，修改 DDL 参数配置，避免同一个库表对象接收多个数据中心的 DDL 同步。

迁移至 MongoDB
MongoDB 迁移至 MongoDB
支持能力

支持场景及版本

支持不同架构之间的迁移，副本集 > 副本集/分片集群，分片集群 > 副本集/分片集群，单节点 > 副本集/分片集群。

源数据库	目标数据库	场景说明
自建数据库 MongoDB（IDC 自建、CVM 上自建） 2.6、2.8、3.0、3.2、3.4、3.6、4.0、4.2、4.4、5.0、6.0、7.0	云数据库 MongoDB 4.0、4.2、4.4、5.0、6.0、7.0	云下数据库迁移上云。
第三方云厂商 MongoDB 2.6、2.8、3.0、3.2、3.4、3.6、4.0、4.2、4.4、5.0、6.0、7.0 AWS MongoDB Atlas 6.0、7.0	云数据库 MongoDB 4.0、4.2、4.4、5.0、6.0、7.0	其他云厂商迁移数据库到云数据库。
云数据库 MongoDB 3.2、3.6、4.0、4.2、4.4、5.0、6.0、7.0	云数据库 MongoDB 4.0、4.2、4.4、5.0、6.0、7.0	- 云平台同地域之间、或不同地域之间的迁移。 - 云平台相同主账号之间数据库迁移、不同主账号之间的数据库迁移。 - MongoDB 实例不同版本之间的迁移。 - MongoDB 副本集群与分片集群之间迁移。

支持功能

功能大类	功能子项或说明	支持能力
迁移对象	-	数据库、集合
迁移类型	-	- 全量迁移 - 全量 + 增量迁移
任务管理	重试	支持

功能大类	功能子项或说明	支持能力
关键操作		
	创建类似任务	支持
增量同步	DML 同步（INSERT/UPDATE/DELETE）	支持
	DDL 同步	INDEX：createIndexes、createIndex、dropIndex、dropIndexes COLLECTION：createCollection、drop、collMod、renameCollection、convertToCapped DATABASE：dropDatabase、copyDatabase 支持副本集和分片集群的 DDL 操作。
一致性校验	校验对象	全部迁移对象/自定义对象
	校验方式	行数对比/内容校验/抽样对比

使用说明

源库影响

DTS 在执行全量数据同步时，会占用一定源库资源，可能会导致源库负载上升，增加数据库自身压力。如果您数据库配置过低，建议您在业务低峰期进行数据迁移。

目标库影响

迁移过程中，DTS 会使用系统服务账号在目标端的 TencetDTSData 库下以任务 ID 创建一个表，例如表名为 dts- 用于记录 CHECKPOINT，在任务发生中断时，实现断点续传。

迁移限制

> 警告：
>
> 数据迁移是以整体数据库搬迁为目的，它属于一次性的短期任务，不建议用于长期数据同步，如灾备同步等持续性业务场景。如果使用 DTS 进行长期数据同步，可能会由于实例连接信息变更、源端oplog被清理等情况，导致任务失败。

1. 迁移过程中请勿进行如下操作，否则会导致迁移任务失败。
    - 请勿修改、删除源数据库和目标数据库中用户信息（包括用户名、密码和权限）和端口号。
    - 请勿在源库上执行清除 oplog 的操作。
    - 在数据迁移阶段，请勿删除目的端数据库 TencetDTSData。
2. 在数据迁移阶段，请谨慎操作目的端数据，避免最终数据不一致。

迁移架构

1. 分片迁移相关说明如下：
    i. 分片集群迁移前，建议提前清理源端集群孤儿文档（Orphaned Document）。否则，可能导致迁移后数据校验不一致的问题。如何清理孤儿文档，请参见 MongoDB 官方文档 cleanupOrphaned。
    ii. 分片迁移中，请勿在源端对迁移的库表启用分片，避免源端和目标端数据分布不一致。若迁移中，源端对迁移的库表存在启用分片的操作，请在目标端检查分片状态，如果目标端没有启用分片，请手动执行下启用分片的操作。启用分片的具体操作，请参见 MongoDB 官方文档 Shard a Collection。
    iii. 源端为云数据库 MongoDB 3.2版本的分片集群，迁移时默认将所有分片片键作为哈希片键处理；如果希望在目标端使用范围片键，请在数据迁移前，预先在目标端创建范围片键。

2. 由于单节点无 Oplog，所以自建实例是单节点时，不支持增量迁移。
3. 对于副本集及分片集群版（4.2及以上版本） MongoDB 的迁移，增量迁移支持 Change Stream 获取数据变更。
4. 对于分片集群版 MongoDB 迁移，支持 SRV 地址连接 MongoDB 数据库。

迁移操作指导

操作场景

基于 DTS 的 MongoDB 数据迁移，支持全量 + 增量数据的迁移，即迁移前源库的历史数据，和迁移过程中源库新增的写入数据可一并进行迁移。
本文介绍使用 DTS 数据迁移功能从 MongoDB 迁移数据至云数据库 MongoDB 的操作指导。

准备工作

1. 请仔细阅读 使用说明，了解功能约束和注意事项。

2. 请根据您需要使用的接入类型，提前打通 DTS 与数据库之间的访问通道，具体请参考 网络准备工作。
   - IDC 自建数据库/其他云厂商数据库：接入方式可选择“公网/专线接入”。
   - CVM 上的自建数据库：接入方式选择“云主机自建”。
   - 云数据库实例：接入方式选择“云数据库”。

3. 建议在源数据库创建一个只读账号供迁移使用，参考方式如下。

```
# 源数据库为副本集、单节点时的语法示例
use admin
db.createUser({user: "username",pwd: "password",roles:[{role: "readAnyDatabase", db: "admin"},{role: "read",
db: "local"}]})

# 源数据库为分片集群时的语法示例
use admin
db.createUser({user: "username",pwd: "password",roles:[{role: "readAnyDatabase", db: "admin"},{role: "read",
db: "local"},{role: "read", db: "config"}]})
```

4. 目标库为云数据库实例，可以使用 mongouser 进行迁移，也可以自行创建账号，自行创建账号的参考方式如下。

```
db.createUser({user:"username",pwd:"password",roles:[{role:"readWriteAnyDatabase",db:"admin"}]})
```

操作步骤

1. 登录 DTS 控制台，在左侧导航选择数据迁移页，单击新建迁移任务，进入新建迁移任务页面。

2. 在新建迁移任务页面，选择迁移的源实例类型和所属地域，目标实例类型和所属地域，规格等，然后单击立即购买。

配置参数	说明

配置参数	说明
创建模式	- 新建任务：新建一个全新的任务。 - 创建类似任务：快速创建一个和历史任务相同配置的任务，新的任务中数据库类型、接入方式、计费模式、迁移类型等默认填充的选项都和历史任务保持一致，用户如果需要也可根据情况进行修改。
源实例类型	请根据您的源数据库类型选择，购买后不可修改。本场景选择“MongoDB”。
源实例地域	选择源数据库所属地域。如果源库为自建数据库，选择离自建数据库最近的一个地域即可。
目标实例类型	请根据您的目标数据库类型选择，购买后不可修改。本场景选择“MongoDB”。
目标实例地域	选择目标数据库所属地域。
规格	当前仅支持 Medium 规格。
任务名	- 选择创建后命名，默认任务名称与任务 ID 一致。在迁移任务创建完成之后，可重新设置任务名称。 - 选择立即命名，在下面输入框，输入任务名称。

3. 购买完成后，页面自动跳转到数据迁移的任务列表，请选择刚才购买的任务进行配置。
   如果您购买了多个地域，或者进行跨地域的任务配置，任务列表按照目标实例的所属地域进行展示，可在上方切换地域后找到已购买的任务。

4. 在设置源和目标数据库页面，完成任务设置、源库设置和目标库设置。

> 注意：
>
> - 源库账号请填入之前创建的只读账号，否则前置校验步骤将不通过。
> - 如果源库或者目标库为云数据库实例，DTS 在迁移过程中会使用系统服务账号导出数据/写入数据。例如，源库为云数据库实例，则 DTS 使用用户填入的只读账号连接源库外，还会使用系统服务账号导出源库数据；目标库为云数据库实例，则 DTS 使用用户填入的账号连接目标库外，还会使用系统服务账号向目标库写入数据。

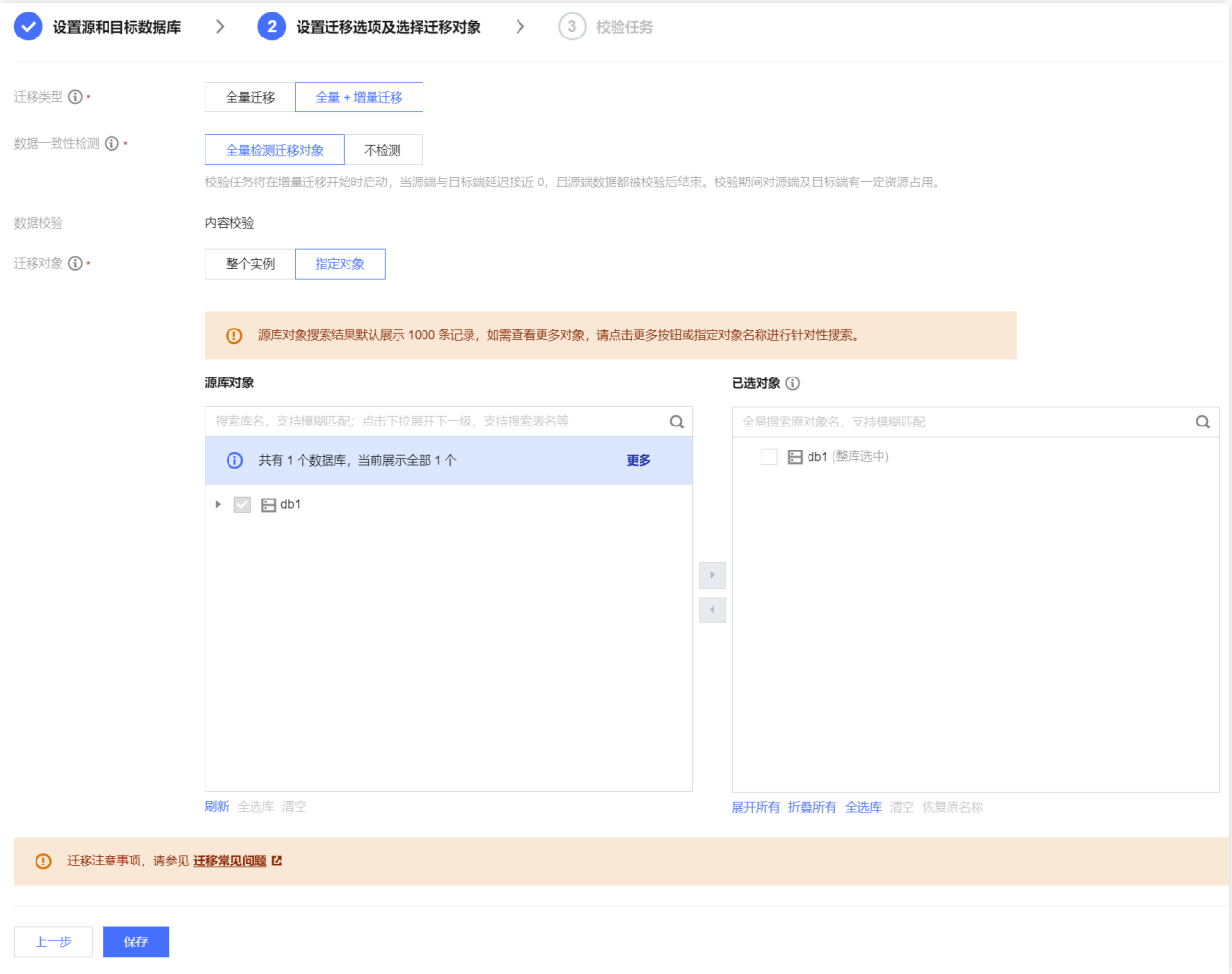
配置参数	说明
任务名称	设置一个具有业务意义的名称，便于任务识别。
运行模式	- 立即执行：前置校验通过后会立即启动任务。 - 定时执行：设置一个任务开始执行的时间，前置校验通过后不启动任务，到设定的时间再启动。

配置参数	说明
接入类型	请根据您的场景选择，选择不同接入类型的准备工作请参考 准备工作概述。 - 公网：源数据库可以通过公网访问。 - 云主机自建：源数据库部署在云服务器 CVM 上。 - 专线接入：源数据库可以通过专线接入方式与私有网络打通。 - 云数据库：源数据库属于云数据库实例。 源库为 IDC 自建数据库/其他云厂商数据库，接入方式可选择“公网/专线接入”。源库为 CVM 上的自建数据库，源库接入方式选择“云主机自建”。源库为云数据库实例，接入方式选择“云数据库”。
架构	接入类型选择“云主机自建/专线接入”时显示该参数。请根据实际情况选择。 - 副本集：指源库为副本集，每一个副本集由一个 Primary 节点和一个或多个 Secondary 节点组成的架构。 - 集群迁移：指源库为分片集群，每一个分片集群由 mongos 节点、config server、shard 节点等组件组成的架构。 - 单节点：指源库集群为单个节点来进行读写操作。 说明： 选择一种架构类型，并且连通性测试通过后，不能再修改为其他架构类型，否则任务会报错。
集群迁移	架构选择“集群迁移”，增量迁移方式为“oplog”时，需要配置如下参数。 - 节点 - mongod：请输入 mongod 节点 IP 和端口，或者域名和端口。多个节点请换行输入；每个 shard 下仅填入一个 mongod 即可，示例：1xx.xx.55.77:6xx9 - 节点 - mongos：请输入 mongos 节点 IP 和端口，或者域名和端口。 - 节点 - Config Server：请输入 Config Server 节点的 IP 和端口，或者域名和端口。 架构选择“集群迁移”，增量迁移方式为“Change Stream”时，需要配置如下参数。 - 连接串类型：支持普通或者 SRV 连接串 - 节点 - mongos：请输入 mongos 节点 IP 和端口，或者域名和端口。 说明： Change Stream 迁移方式以下的 DDL event： createIndexes、dropIndexes、modify、drop、create、shardCollection、reshardCollection、refineCollectionShardKey。 其中：createIndexes、dropIndex、create、modify 仅6.0版本以上支持，refineCollectionShardKey 仅在6.10版本以上支持。
公网	接入类型选择“公网”时，需要配置如下参数。 - 主机地址：源数据库 IP 地址或域名。 - 端口：源数据库使用的端口。
云主机自建	接入类型选择“云主机自建”时，需要配置如下参数。 - 云主机实例：云服务器 CVM 的实例 ID。 - 端口：源数据库使用的端口。
专线接入	接入类型选择“专线接入”时，需要配置如下参数。 - 私有网络专线网关：专线接入时只支持私有网络专线网关，请确认网关关联网络类型。 - 私有网络：选择私有网络和子网。 - 主机地址：源数据库 IP 地址或域名。 - 端口：源数据库使用的端口。
云数据库	接入类型选择“云数据库”时，需要配置如下参数。 云数据库实例：选择源数据库的实例 ID。
是否需要认证	是否需要对源库中用户名和密码的安全性进行认证。选择“需要”则要填写如下参数。 - 认证库：需要认证的库名，即执行迁移任务账号所属的数据库名称，只支持填写 admin。

配置参数	说明
	- 认证机制：当前仅支持 SCRAM-SHA-1。 - 账号及密码选择 - 相同账号及密码：mongod、mongos、Config Server 角色均采用相同账号及密码时选择该参数，并填入统一的账号密码。 - 不同的账号及密码：mongod、mongos、Config Server 角色采用不同账号及密码时选择该参数，并分别填入 mongod、mongos、Config Server 的账号和密码。
账号/密码	账号/密码：源数据库的账号、密码。
连接方式	- 非加密连接：源端为非 AMS Mongo Atlas 时，只能选择非加密连接。 - Mongo Atlas SSL：源端为 AWS Mongo Atlas 时，可以根据需要选择是否 SSL 加密连接，对数据安全要求较高的场景，建议勾选 Mongo Atlas SSL。

5. 测试源实例和目标实例的连通性。

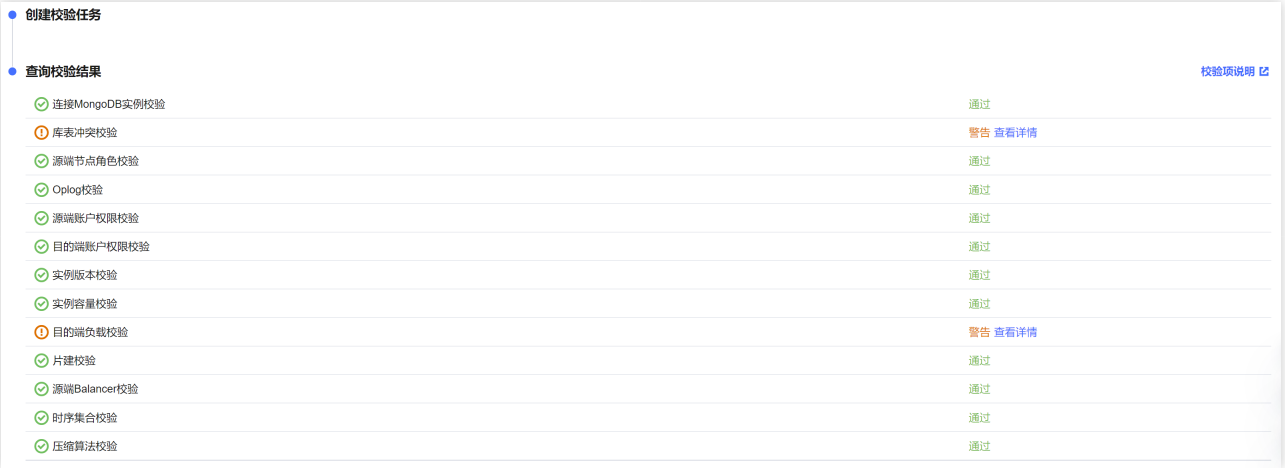
6. 在设置迁移选项及选择迁移对象页面，设置迁移选项和迁移对象。



配置参数	说明
迁移类型	请根据您的场景选择。 - 全量迁移：迁移整个数据库，迁移数据仅针对任务发起时，源数据库已有的内容，不包括任务发起后源库实时新增的数据写入。 - 全量 + 增量迁移：迁移数据包括任务发起时源库的已有内容，也包括任务发起后源库实时新增的数据写入。如果迁移过程中源库有数据写入，需要不停机平滑迁移，请选择此场景。
数据一致性校验	仅当迁移类型配置为“全量 + 增量迁移”时，支持进行一致性校验，对迁移后源库和目标库的数据进行详细对比；当迁移类型配置为“全量迁移”时，固定为不检测。 - 全量检测迁移对象：迁移任务在增量同步完成后，DTS 会自动触发一次一致性校验任务。 - 不检测：即不进行数据校验。用户如果需要可在增量同步完成后，手动进行触发，详情参考 创建数据一致性校验。
数据校验	当数据一致性校验选择“全量检测迁移对象”时，显示默认一致性校验的类型为“内容校验”。
迁移对象	整个实例：迁移整个实例，但不包括系统库：admin、local、config。 指定对象：迁移指定对象。
指定对象	在源库对象中选择待迁移的对象，然后将其移到已选对象框中。

7. 在校验任务页面，完成迁移前校验工作，单击启动任务。

8. 如果校验任务不通过，可以参考 校验不通过处理方法 修复问题后重新发起校验任务。
   - 失败：表示校验项检查未通过，任务阻断，需要修复问题后重新执行校验任务。
   - 警告：表示检验项检查不完全符合要求，可以继续任务，但对业务有一定的影响，用户需要根据提示自行评估是忽略警告项还是修复问题再继续。



9. 返回迁移任务列表，同步任务运行中。
   如果您需要进行查看任务详情、删除任务等操作，请单击对应的任务操作。如果发生任务报错，请参考错误处理。

10. 结束任务。

- 选择全量迁移：任务完成后会自动结束，不需要手动结束。
- 选择全量 + 增量迁移：全量迁移完成后会进入增量数据同步阶段，增量数据同步不会自动结束，需要用户手动来结束任务。
  请在增量同步完成（即状态为“准备完成”），且目标与源库时间延迟为0秒时，在操作列单击完成，即可结束迁移任务。

11. （可选）如果需要割接，在结束任务后，任务状态变为任务成功时，即可对业务进行正式割接。

前置校验不通过处理
连接 MongoDB 实例校验

检查详情

源数据库和目标数据库需要能正常连通，如果未连通，会报连接失败。

问题原因

- 源数据库所在网络或服务器设置了安全组或防火墙。
- 源数据库对来源 IP 地址进行了限制。
- 网络端口未放通。
- 数据库账号或密码不正确。

修复方法

请按照问题原因中的对应原因进行处理。

库表冲突校验

检查要求

MongoDB 迁移场景中，目标实例可以存在与源库同名，但是不能存在数据（只能为空表）。

修复方法

如果存在冲突报错，删除目标库中的对应库表，或者删除目标库同名库表内的数据。

源端节点角色校验

检查详情

- 检查要求：MongoDB 迁移任务，源端为分片时，需要填写对应 mongos，config server，mongod 节点信息。
- 检查说明：mongos，config server 和 mongod 节点信息填写不能混乱，否则会导致数据迁移错乱，例如将 mongos 节点信息填入 mongod 填写框内。注意，每个分片只需要填写一个 mongod 节点。

修复方法

- DTS 任务填写框内填写正确节点信息。
- 每个分片只填写一个 mongod。

Oplog 校验

检查详情

- 检查要求：进行全量 + 增量迁移时，能够从源端获取到 Oplog。
- 检查说明：增量迁移需要通过 Oplog 进行回放，如果源端 local 库下不存在 oplog.rs 或者 oplog.$main 表格，则无法获取 Oplog。

修复方法

将源端以副本集或者主从方式启动，保证操作能够产生 Oplog，并且记录在源端 local 库下。

源端或目标端账户权限校验

检查详情

检查用户是否具备对数据库的操作权限，具体可参考如下对应文档。
数据迁移权限要求：MongoDB 数据迁移。

修复方法

用户若不具备操作权限，请按照检查要求中的对应权限要求对用户进行授权，然后重新执行校验任务。

实例版本校验

源数据库和目标数据的版本符合 MongoDB 支持的版本。

实例容量校验

检查要求

MongoDB 迁移场景，目标库存储空间需要在源库待迁移库表空间的1.3倍以上。

修复方法

- 删除目标库中的部分数据，以便腾出足够的空间。
- 升级目标库存储规格，使用更大容量的实例进行迁移。

目的端负载校验

检查详情

- 检查要求：DTS 迁移会导致目的端负载变高，如果在迁移过程中，目的端有业务使用，则会发出校验警告。警告不会阻塞任务的继续，但会对业务有一定影响，请用户评估后自行决定是否忽略警告。
- 业务影响：MongoDB DTS 采用逻辑同步的方式进行数据迁移，会对目的端 CPU 负载造成一定压力，如果目的端有业务使用，请谨慎评估发起。

修改方法

停止目标端的业务使用，重新执行校验任务。

片键校验

检查详情

- 检查要求：当目的端为分片实例时，可以在目的端预设片键，如果目的端与源端表格片键不一致，则会发出警告提示。警告不会阻塞任务的继续，但会对业务有一定影响，请用户评估后自行决定是否忽略警告。
- 业务影响：部分片键不一致场景会导致迁移或者同步任务失败。

修复方法

如果目的端预设片键，参考如下命令在源端进行分片操作。

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

重新执行校验任务。

源端 Balancer 校验

检查详情

- 检查要求：源端为分片实例情况下，源端必须关闭 Balancer 才能发起迁移。
- 检查说明：增量迁移会获取 Oplog，开启 Balancer 的情况下，源端 moveChunk 可能会导致最终目的端数据不一致。

修复方法

1. 登录源数据库。

2. 使用如下命令关闭源端 Balancer。

```
sh.stopBalancer()
sh.getBalancerState()
```

3. 重新执行校验任务。

时序集合校验

校验详情

MongoDB 5.0及以上版本支持时序集合，从5.0及以上版本迁移至低版本时，如果源库有时序集合时，本校验项会不通过。

修复方法

从5.0及以上版本迁移至低版本的场景中，在配置任务勾选迁移对象时，只选择非时序集合。

压缩算法校验

校验详情

校验源库使用的压缩算法和目标库的是否相同，如果不同，则会产生警告，警告项不会阻塞迁移，用户可忽略警告继续任务。
需要注意的是，在校验目标库使用的压缩算法时，取的是任意一张系统表，修改压缩算法后系统表的压缩算法不会改变，所以可能存在警告不准确的情况，如果确定目标库已经使用了新的压缩算法，则可以忽略警告。

修复方法

同样的数据在不同的压缩算法下占用的磁盘大小是不一样的，如果用户希望目标库采用和源库一样的压缩算法，请修改目标库的压缩算法。

迁移至 Redis®
Redis® 迁移至 Redis®
支持能力

支持场景及版本

源端	目标端	场景说明
自建数据库 Redis（IDC 自建/ CVM 上自建）2.8、3.0、3.2、4.0、5.0、6.2、7.0	Redis® 2.8、3.0、4.0、5.0、6.2、7.0	云下数据库迁移上云。
其他第三方云厂商 Redis - AWS ElasticCache for Redis® 4.0、5.0、6.0、6.2、7.0 - Codis	Redis® 2.8、3.0、4.0、5.0、6.2、7.0	其他云厂商的 Redis 数据库迁移到云数据库实例。
Redis® 2.8、3.0、3.2、4.0、5.0、6.2、7.0	Redis® 2.8、3.0、3.2、4.0、5.0、6.2、7.0	• Redis® 相同地域之间，或者不同地域之间的迁移。 • Redis® 跨版本迁移。 • Redis® 相同主账号下之间，或者不同主账号之间的迁移。 • Redis® 标准架构与集群架构相互之间的迁移。
	自建数据库 Redis（IDC 自建/ CVM 上自建）2.8、3.0、3.2、4.0、5.0、6.2、7.0	Redis® 迁移到自建数据库。

支持不同架构之间的迁移，标准架构 > 标准架构，标准架构 > 集群架构，集群架构 > 集群架构，但异构迁移（如标准架构 > 集群架构）可能会存在兼容性问题。
标准架构指自建的单机版、主从模式以及Redis® 内存版（标准架构）；集群架构包括 Redis Cluster 原生方案、Codis 集群方案、Twemproxy 集群方案。

支持功能

功能大类	功能子项或说明	支持能力
迁移对象	-	数据库、Key
迁移类型	-	仅支持“全量 + 增量迁移”
断点续传	-	不支持
任务管理关键操作	重试	支持
	限速	支持
	创建类似任务	支持
同步拓扑	一对一	支持
	双向同步	不支持

使用说明

源库影响

DTS 在执行全量数据迁移时，会占用一定源端实例资源，可能会导致源实例负载上升，增加数据库自身压力。如果您的数据库配置过低，建议您在业务低峰期进行迁移。

迁移限制和说明

> 警告：
>
> 数据迁移是以整体数据库搬迁为目的，它属于一次性的短期任务，不建议用于长期数据同步，如灾备同步等持续性业务场景。如果使用 DTS 进行长期数据同步，可能会导致以下严重后果：
>
> - 业务持续性风险：源端主备切换、网络中断或实例连接信息变更时，将触发任务重建流程，导致全量数据重新导出导入，重建期间源端同目标端之间，将产生数据延迟，进而导致业务无法切换到目标端，影响业务可用性。

1. DTS 迁移数据需要源实例支持 PSYNC 命令。
2. 源实例在正常运行状态下才支持迁移，未初始化密码或者有其他任务（如扩容等）在执行中时不能迁移。
3. 源库为 AWS ElastiCache for Redis 场景，相应的约束限制如下：
    i. 仅支持迁移 ElastiCache for Redis，其他 AWS Redis 产品暂时不支持。
    ii. 暂不支持 TLS 连接，如果源库开启了 TLS，需要关闭后才能迁移。因为 AWS ElastiCache for Redis 关闭 TLS 后，Auth 用户认证功能也会同时关闭，可能会引发数据暴露风险，所以建议用户先创建安全组，再关闭 TLS，然后将 DTS 的服务 IP 添加到源库的安全组白名单中。
    iii. 因 AWS ElastiCache for Redis 从节点不支持 PSYNC 命令，因此只能使用主节点进行迁移。
4. DTS 任务不能构成环形同步。环形同步写入的数据不会被终止，一直在环路上执行，将会导致写风暴。
5. 不支持迁移模块（Modules）。
6. 迁移过程中请勿进行如下操作，否则会导致迁移任务失败。
    - 请勿修改、删除源数据库和目标数据库中用户信息（包括用户名、密码和权限）和端口号。
    - 在数据迁移阶段，请勿删除目的端数据库。
    - 在数据迁移阶段，请勿操作目的端数据，避免最终数据结果不一致。
7. 迁移过程中，若发生以下情况（包括但不限于），可能导致数据不一致：
    - 源端或目标端 Redis® 发生主备切换或故障切换。
    - 源端或目标端 Redis® 连接地址发生变化。
8. 目标实例为分片集群，DTS 支持 key 路由到对应分片。
    - 目标端 Redis® 发生 HA 或者扩容，会导致 DTS 任务失败，请提前配置自动重试策略。
9. 网络中断或者集群发生 HA 切换等场景会自动触发全量同步，全量同步期间会存在数据延迟。

迁移操作指导

操作场景

基于 DTS 的 Redis 数据迁移，支持全量 + 增量数据的迁移，即迁移前源库的历史数据，和迁移过程中源库新增的写入数据可一并进行迁移。
本文介绍使用 DTS 数据迁移功能从 Redis 迁移数据至云数据库 Redis® 的操作指导。

准备工作

1. 请根据您需要使用的接入类型，提前打通 DTS 与数据库之间的访问通道，具体请参考 网络准备。
   - IDC 自建数据库/其他云厂商数据库：接入方式可选择“公网/专线接入”。
     外网实例迁移时，请确保源实例服务在外网环境下可访问，并且要保持外网连接的稳定性，当网络出现波动或者故障时会导致迁移失败，迁移一旦失败，就需要重新发起迁移任务。
   - CVM 上的自建数据库：接入方式选择“云主机自建”。
   - 云数据库实例：接入方式选择“云数据库”。

2. 迁移前检查
   如下检查需要用户在迁移前自行排查，否则可能会出现迁移失败。

   i. 检查源端是否存在大 Key
      在迁移过程中，大 Key 可能引起缓冲区 client-output-buffer-limit 溢出，导致迁移失败。
      - 云数据库，请使用数据库智能管家的诊断优化功能快速分析大 Key。
      - 非云数据库，请使用 rdbtools 分析 Redis 大 Key。
        评估大 Key 进行拆分或清理，如果保留大 Key，请设置源端缓冲区的大小 client-output-buffer-limit 为无限大。

```
config set client-output-buffer-limit 'slave 0 0 0'
```

   ii. 检查源端 Linux 内核 TCP 连接数的限制
      如果业务并发请求比较大，迁移之前，请检查 Linux 内核对连接数的限制，如果业务请求连接数超出内核限制的连接数，Linux 服务器将会主动断开与 DTS 的连接。

```
echo "net.ipv4.tcp_max_syn_backlog=4096" >> /etc/sysctl.conf
echo "net.core.somaxconn=4096" >> /etc/sysctl.conf
echo "net.ipv4.tcp_abort_on_overflow=0" /etc/sysctl.conf
sysctl -p
```

   iii. 检查源端 RDB 文件目录的访问权限
      迁移之前，请务必检查源端存放 RDB 文件目录的访问权限是否为可读，否则将会因 RDB 文件不可读而引起迁移失败。
      如果 RDB 文件所在目录不可读，请在源端执行如下命令，设置“无盘复制”，直接发送 RDB 文件给 DTS 落盘，而不需要保存在源端的磁盘再发送。

config set repl-diskless-sync yes

iv. （可选）对于标准架构迁移到集群架构的场景，请检查命令兼容性问题。

操作步骤

步骤一：新建迁移任务

1. 登录 DTS 控制台，选择数据迁移页，然后在右侧单击新建迁移任务。

2. 在新建迁移任务页面，选择任务创建参数，然后单击立即购买。

配置参数	说明
创建模式	- 新建任务：新建一个全新的任务。 - 创建类似任务：快速创建一个和历史任务相同配置的任务，新的任务中数据库类型、接入方式、计费模式、迁移类型等默认填充的选项都和历史任务保持一致，用户如果需要也可根据情况进行修改。
源库类型	请根据您的源数据库类型选择，购买后不可修改。此处选择 Redis。
源实例地域	选择源数据库所属地域。如果源库为自建数据库，选择离自建数据库最近的一个地域即可。
目标库类型	请根据您的目标数据库类型选择，购买后不可修改。此处选择 Redis。
目标实例地域	选择目标数据库所属地域。
规格	当前仅支持 Xlarge 规格。
迁移任务名称	- 选择创建后命名，默认任务名称与任务 ID 一致。在迁移任务创建完成之后，可重新设置任务名称。 - 选择立即命名，在下面输入框，输入任务名称。
数量	单次最多支持购买10个迁移任务。

3. 购买完成后，页面自动跳转到数据迁移任务列表。

步骤二：设置源和目标数据库

1. 在数据迁移任务列表中，找到刚才创建的任务，单击操作列的配置。

2. 进入设置源和目标数据库页签，填写源和目标数据库等信息，然后单击测试连通性，测试通过后，单击保存进入下一步。

- 任务设置

配置参数	说明
任务名称	设置一个具有业务意义的名称，便于任务识别。
运行模式	- 立即执行：前置校验通过后会立即启动任务。 - 定时执行：设置一个任务开始执行的时间，前置校验通过后不启动任务，到设定的时间再启动。
自动重试	勾选后，迁移任务因网络异常等引起的任务中断，DTS 将在设置的时间范围内自动重试，不需要用户手动操作。 重试机制为，从头开始重新进行数据迁移，重新迁移时，之前已迁移到目标端的数据会被清空，或者会被覆盖。

- 源库设置

配置参数	说明
源库类型	购买时选择的源数据库类型，不可修改。
所属地域	购买时选择的源数据库地域，不可修改。
接入类型	选择源库接入云网络类型。不同的接入类型需要进行的网络打通操作不同，具体请参考 网络准备。 - 源库为 IDC 自建数据库/其他云厂商数据库，接入方式可选择“公网/专线接入”。 - 公网：源数据库可以通过公网 IP 访问。 - 专线接入：源数据库可以通过专线接入方式与私有网络打通。 - 源库为 CVM 上的自建数据库，接入方式选择“云主机自建”。 - 源库为云数据库实例，接入方式选择“云数据库”。
节点类型	接入类型选择除“云数据库”之外的其他参数时显示该参数。请根据实际情况选择。 - 单节点迁移：指一主多副本的集群架构。 - 集群迁移：指由若干个分片，而每个分片上又存在若干个副本节点的集群架构。勾选集群迁移后，需要填写节点信息。 - 填写源库集群的所有分片节点地址和密码（IP:端口:密码 或 IP:端口），多个节点请换行处理。 - 建议填写的地址为源库的副本节点（从节点），从副本节点迁移可以避免影响源库的业务访问。
公网	接入类型选择“公网”时，需要配置如下参数。 - 主机地址：源数据库 IP 地址或域名。 - 端口：源数据库使用的端口。 - 密码：源数据库迁移账号的密码。
云主机自建	接入类型选择“云主机自建”时，需要配置如下参数。 - 云主机实例：云服务器 CVM 的实例 ID。 - 端口：源数据库使用的端口。 - 密码：源数据库迁移账号的密码。
专线接入	接入类型选择“专线接入”时，需要配置如下参数。 - 私有网络专线网关：专线接入时只支持私有网络专线网关，请确认网关关联网络类型。 - 私有网络：选择私有网络和子网。 - 主机地址：源数据库 IP 地址。 - 端口：源数据库使用的端口。 - 密码：源数据库迁移账号的密码。

配置参数	说明
云数据库	接入类型选择“云数据库”时，需要配置如下参数。 - 是否跨账号 - 本账号：源数据库实例和目标数据库实例所属的主账号为同一个云账号。 - 跨账号：源数据库实例和目标数据库实例所属的主账号为不同的云账号。详细的跨账号操作指导请参见云数据库跨账号实例间迁移。 - 云数据库实例：选择源数据库的实例 ID。 - 密码：源数据库迁移账号的密码。

- 目标库设置

  目标库参数设置与源库类似，如下仅对差异点进行描述。

配置参数	说明
接入类型	根据您的场景选择。不同接入类型的配置原则与源库类似。 支持下云迁移，当源数据库接入类型选择“云数据库”时，目标库可选择“公网/云主机自建/专线接入/云数据库”。
节点类型	接入类型选择除“云数据库”之外的其他参数时显示该参数。 请根据目标库的实际情况选择，支持选择“单节点迁移”、“Redis Cluster 迁移”、“代理集群迁移”。

步骤三：设置迁移选项和选择迁移对象

在设置迁移选项及迁移对象页签，配置参数后，单击下一步。

- 基础选项

配置项	说明
迁移类型	默认选择全量 + 增量迁移，不可修改。全量+增量迁移表示迁移前源库的历史数据，和迁移过程中源库新增的写入数据都一起迁移。
迁移对象	默认选择整个实例，不可修改。
启用 key 过滤	多 key 命令可以设置前缀过滤规则，详情请参考 Key 前缀过滤。
目标库写入模式	- 清空目标实例：源库数据同步到目标库时，目标库已有的数据会被清空，然后重新写入。 - 覆盖写：源库数据同步到目标库时，用源库的 Key 来覆盖目标库的 Key。 对于迁移下云的场景，如果 DTS 任务配置中没有选择“覆盖写”，则在目标端有数据时会触发前置检查无法通过，自动重试会失效。
启用目标库读写	允许 DTS 或者其他服务向目标库写入数据。Redis 迁移场景需要开启目标库读写。
过期 Key 自动淘汰	1. 勾选后目标库上对过期的 Key 自动淘汰，可以节省目标库的空间，但勾选后可能会出现源库和目标库的数据结果不一致，请慎重操作。 1. 正常情况下，目标端本身会设置过期 Key 自动淘汰，用户如果需要保证源和目标的数据结果一致性，在 DTS 任务配置中不勾选“过期 Key 自动淘汰”，则目标端上过期 Key 不会淘汰。在结束 DTS 任务后，这里的配置会还原，目标库上过期 Key 仍会自动淘汰。 1. Redis 迁移场景需要开启目标库读写，如果用户需要迁移源库的 SET NX 命令（键不存在时才执行），建议开启过期 Key 自动淘汰，否则 DTS 向目标库写入 SET NX 命令时，可能会因为目标库上存在过期的 key，导致迁移任务报错。

- 高级选项

配置项	说明
限制数据传输速率	一般不需要设置，如果 DTS 的传输速率对目标库造成负载影响，则可设置限速。 - 目标库分片数：目标库为云数据库实例时，自动获取实例分片数，不支持修改。 - 目标库单分片流量限制：填写单分片流量限制，设置范围为0-100MB/s。 如果目标实例为集群架构，则单实例每秒传输的最大流量=分片数*单分片流量限制(MB/s)

步骤四：校验和启动任务

1. 在校验任务页面，系统自动进行前置校验，校验通过后，单击启动任务，会立即启动数据迁移（如果在配置任务时设置了定时执行时间，这里也可选择稍后启动，任务将在设置的时间点启动）。

> 说明：
>
> 如果出现校验结果失败， 请参考 前置校验项不通过处理。

   - 校验结果为失败：表示校验项检查未通过，任务阻断，需要修复问题后重新执行校验任务。
   - 校验结果为警告：表示检验项检查不完全符合要求，可以继续任务，但对业务有一定的影响，用户需要根据提示自行评估是忽略警告项还是修复问题再继续。

2. 返回数据迁移任务列表，任务进入准备运行状态，运行1分钟 - 2分钟后，数据迁移任务开始正式启动。

步骤五：结束任务

1. 如果您需要进行查看任务进度详情、删除任务等操作，请在操作列单击对应的按钮。
2. 当任务状态为“准备完成”，表示增量同步已完成，用户可以结束任务。在操作列单击完成，结束迁移任务。
   - 建议在目标与源库的数据差距为0KB、目标与源库时间差距为0秒时，再结束任务。
   - 如果需要割接，在结束任务后，任务状态变为任务成功时，即可对业务进行正式割接。

Key 前缀过滤

操作场景

DTS 的 Redis® 数据迁移链路中，支持对源库的多 Key 命令设置前缀过滤规则，方便用户管理数据库中的键值。

Key 前缀过滤规则

- 支持配置单个或者多个 key 前缀匹配规则，多个前缀请用回车键分隔。
- 迁移规则与过滤规则混合使用时，过滤规则优先级高于迁移规则。

Key 前缀过滤规则示例

1. MSET/DEL/UNLINK 命令，采用精准匹配规则。
   - 迁移规则：源端多 key 命令匹配了一个 key，则仅该 key 迁移到目标端，其他 key 不迁移。
   - 过滤规则：源端多 key 命令匹配了一个 key，则仅该 key 过滤，不迁移到目标端，其他 key 会迁移。

前缀类型	源库 key 命令	DTS 设置的前缀	目标端结果
迁移 key 值前缀	MSET dtsnew "111" abcd "222"	dts	MSET dtsnew "111"
过滤 key 值前缀	MSET redisnew "111" efgh "222"	redis	MSET efgh "222"

2. MSET/DEL/UNLINK 以外的其他命令。
   - 迁移规则：源端多 key 命令只要匹配了一个迁移前缀，则该命令全部迁移到目标端。
   - 过滤规则：源端多 key 命令的所有 key 都匹配上，该命令才会被过滤；仅匹配到一个 key，该命令不会过滤，仍会迁移到目标端。

前缀类型	源库 key 命令	DTS 设置的前缀	目标端结果
迁移 key 值前缀	SUNION dtsnew abcd	dts	SUNION dtsnew abcd
过滤 key 值前缀	SUNION redisnew efgh	redis	SUNION redisnew efgh
过滤 key 值前缀	SUNION redisnew efgh	redis efg	不同步

操作步骤

配置 Redis® 迁移任务，在设置迁移选项及选择迁移对象页面中，勾选启用 key 过滤，设置过滤规则后，继续后续配置流程。

前置校验项不通过处理
网络是否可达

检查详情

源数据库和目标数据库需要能正常连通，如果未连通，会报连接失败。

问题原因

- 源数据库所在网络或服务器设置了安全组或防火墙。
- 源数据库对来源 IP 地址进行了限制。
- 网络端口未放通。
- 数据库账号或密码不正确。

修复方法

请按照问题原因中的对应情况进行处理。

源实例目标实例版本是否兼容

检查详情

- Redis® > Redis® 迁移，Redis® 源实例版本需要大于等于2.2.6，2.2.6以下版本不支持 DTS 迁移。
- 建议目标数据库版本大于或等于源数据库版本，否则会报警告，低版本向高版本迁移，有兼容性问题。
- 目标数据库的 Proxy 版本为最新版本。

修复方法

如果源库或者目标库版本不支持，请升级目标实例版本或者使用更高版本的数据库实例。

源实例参数检查

检查详情

- Redis® 迁移场景，目标数据库为 Redis 时，源实例的数据库个数需要小于等于目标实例的数据库个数。
- 源实例的状态是否正常。

修复方法

修改源和目标实例的数据库个数，然后重新启动校验任务。

目标实例容量是否满足要求

检查要求

Redis® > Redis® 迁移，目标库的空间必须大于等于源库待迁移数据所占空间的1.5倍。

修复方法

- 删除目标库中的部分数据，以便腾出足够的空间。
- 升级目标库存储规格，使用更大容量的实例进行迁移。

目标实例状态是否正常

检查详情

检查要迁移的目标库是否已经不存在了，如果不存在则会报错。

修复方法

在迁移过程中不要删除目标库，如果删除了需要重新构建迁移任务。

源实例是否为从机

检查详情

源库建议为 Slave 节点，否则校验项会报警告，警告项不会阻止任务的继续进行，用户可以忽略后继续任务，但是需要评估忽略后的影响。
迁移过程中 DTS 会对源库进行 bgsave 操作，消耗源库的内存和资源，源库为 Master 节点时对业务的写入影响较大。

- 如果源库为 Redis 数据库，默认数据迁移过程中使用 Slave 节点来迁移，用户无需关注。
- 如果源库为本地自建的 Redis 数据库，可能会出现使用 Master 迁移的场景，如果校验报警告，建议修改源库为从库。

修复方法

建议重新配置迁移任务，源库的参数配置为 Slave 节点信息。

常见错误处理

问题1

问题现象

在使用 DTS 迁移过程中，提示如下错误信息：

```
[launch]state:6 #rdb rdbfile:./tmp1600869159_89068.rdb rdbsize:2753701723 rdb_writed_size:1606959104 rdb_parsed_size:0 rdb_parsed_begin:0 rdb_parsed_time:0 #replication master_replid:0549e2f0bdf373cef0c4c89bb0ce9e1757c4b105 repl_offset:1327777565448 write_command_count:0 finish_command_count:0 last_replack_time:0 #queue send_write_pos:0 send_read_pos:0 response_write_pos:0 response_read_pos:0 errtime:1600870264 errmsg:read rdb eof save rdb fail ready shutdown dts
```

问题原因

检查源端 Redis® 数据库日志，是否包含以下信息，说明源端的配置的缓冲区 client-output-buffer-limit 溢出。

```
psync scheduled to be closed ASAP for overcoming of output buffer limits
```

解决方法

请执行如下命令，设置 client-output-buffer-limit 为无限大，重新发起 DTS 任务。

```
config set client-output-buffer-limit 'slave 0 0 0'
```

问题2

问题现象

在使用 DTS 迁移过程中，提示如下错误信息：

```
[launch]state:8 #rdb rdbfile:./tmp1600395232_34851.rdb rdbsize:107994104 rdb_writed_size:107994104 rdb_parsed_size:107994104 rdb_parsed_begin:1600395238 rdb_parsed_time:5 #replication master_replid:995dba8ccffb7cc32a7c85de7b1632b952b74496 repl_offset:23851025 write_command_count:940765 finish_command_count:940763 last_replack_time:1600395298 #queue send_write_pos:440766 send_read_pos:440765 response_write_pos:440765 response_read_pos:440764 errtime:1600395297 errmsg:get rsp error:ERR value is not an integer or out of range command:*2 $4 INCR $35 APP_API_ORDER_CREATION_USER_4260882
```

问题原因

分别在该地区的两个 DTS Syncer 上进行抓包，发现 key 的 value 为字符，而非数字，导致 INCR 执行时失败。

```
[redacted]:8701>
[redacted]:8701> SETEX APP_API_ORDER_CREATION_USER_4308624 600 ......
OK
[redacted]:8701> INCR APP_API_ORDER_CREATION_USER_4308624
(error) ERR value is not an integer or out of range
[redacted]:8701>
```

解决方法

请删除相关 key 后，重新发起 DTS 迁移。

问题3

问题现象

在使用 DTS 迁移过程中，提示如下错误信息：

> errmsg:Error reading bulk length while SYNCing:Operation now in progress read rdb length from src fail save rdb fail ready shutdown dts

问题原因

查看源实例的报错信息，可发现 rdb 文件没有目录的访问权限。

```
3762:M 17 Sep 17:42:31.386 * Starting BGSAVE for SYNC with target: disk
3762:M 17 Sep 17:42:31.387 * Background saving started by pid 66272
66272:C 17 Sep 17:42:31.387 # Failed opening .rdb for saving: Permission denied
3762:M 17 Sep 17:42:31.452 # Background saving error
3762:M 17 Sep 17:42:31.452 # Connection with slave :<unknown-slave-port> lost.
3762:M 17 Sep 17:42:31.452 # SYNC failed. BGSAVE child returned an error
3762:M 17 Sep 18:42:32.707 * Replication backlog freed after 3600 seconds without connected slaves.
3762:M 17 Sep 19:45:41.579 * Slave :<unknown-slave-port> asks for synchronization
3762:M 17 Sep 19:45:41.579 * Full resync requested by slave :<unknown-slave-port>
3762:M 17 Sep 19:45:41.579 * Starting BGSAVE for SYNC with target: disk
3762:M 17 Sep 19:45:41.581 * Background saving started by pid 75128
75128:C 17 Sep 19:45:41.581 # Failed opening .rdb for saving: Permission denied
3762:M 17 Sep 19:45:41.675 # Background saving error
3762:M 17 Sep 19:45:41.675 # Connection with slave :<unknown-slave-port> lost.
3762:M 17 Sep 19:45:41.675 # SYNC failed. BGSAVE child returned an error
3762:M 17 Sep 19:50:25.741 * Slave :<unknown-slave-port> asks for synchronization
3762:M 17 Sep 19:50:25.741 * Full resync requested by slave :<unknown-slave-port>
3762:M 17 Sep 19:50:25.741 * Starting BGSAVE for SYNC with target: disk
3762:M 17 Sep 19:50:25.742 * Background saving started by pid 75457
75457:C 17 Sep 19:50:25.743 # Failed opening .rdb for saving: Permission denied
3762:M 17 Sep 19:50:25.806 # Background saving error
3762:M 17 Sep 19:50:25.806 # Connection with slave :<unknown-slave-port> lost.
3762:M 17 Sep 19:50:25.806 # SYNC failed. BGSAVE child returned an error
[root@rds1.car.bj2.yongche.com redis]#
```

解决方法

执行如下命令，设置“无盘复制”，重新发起 DTS 任务。

```
config set repl-diskless-sync yes
```

问题4

问题现象

在使用 DTS 迁移过程中，提示如下错误信息：

```
[launch]state:6 #rdb rdbfile:./tmp1597977351_20216.rdb rdbsize:24282193511 rdb_writed_size:18683334200 rdb_parsed_size:0 rdb_parsed_begin:0 rdb_parsed_time:0 #replication master_replid:1b0da9f595cc40b795803eba3c9bea3aad1a1d68 repl_offset:921330115650 write_command_count:0 finish_command_count:0 last_replack_time:0 #queue send_write_pos:0 send_read_pos:0 response_write_pos:0 response_read_pos:0 errtime:1597978778 errmsg:write rdb data fail:456!=1696 error:No space left on device save rdb fail ready shutdown dts
```

问题原因

DTS Syncer 机器上的磁盘空间不足。

解决方法

清理 DTS Syncer 机器上的磁盘，或者挂一块新盘，然后重新发起 DTS 任务。

问题5

问题现象

在使用 DTS 迁移过程中，提示如下错误信息：

```
[launch]state:4/5 #rdb rdbfile: rdbsize:0 rdb_writed_size:0 rdb_parsed_size:0 rdb_parsed_begin:0 rdb_parsed_time:0 #replication master_replid:d3e707ec0e72c3908b0ce70dd2460f48086c5386 repl_offset:683087907631 write_command_count:0 finish_command_count:0 last_replack_time:0 #queue send_write_pos:0 send_read_pos:0 response_write_pos:0 response_read_pos:0 errtime:1654369638 errmsg:Error reading bulk length while SYNCing:Operation now in progress read rdb length from src fail save rdb fail ready shutdown dts
```

问题原因

源端内存不足导致 rdb 生成失败或者是网络不稳定。

解决方法

1. 调整源端内核连接限制和缓冲区限制。
   - 断开源端连接，调整源端系统内核连接限制。

     ```
     echo "net.ipv4.tcp_max_syn_backlog=4096" >> /etc/sysctl.conf
     echo "net.core.somaxconn=4096" >> /etc/sysctl.conf
     echo "net.ipv4.tcp_abort_on_overflow=0" /etc/sysctl.conf
     sysctl -p
     ```

   - 执行如下命令，调整源端的 client-output-buffer-limit 无限大。

     ```
     config set client-output-buffer-limit 'slave 0 0 0'
     ```

2. 如果仍未解决，则增加源端节点所在机器的内存，保证节点有足够的资源产生 rdb 文件。

问题6

问题现象

在使用 DTS 从 Redis® 标准架构（内存版）迁移到集群架构迁移过程中，提示如下错误信息：

```
[launch]state:8 #rdb rdbfile:./tmp1645683629_34614.rdb rdbsize:781035471 rdb_writed_size:781035471 rdb_parsed_size:781035471 rdb_parsed_begin:1645683632 rdb_parsed_time:25 #replication master_replid:5abe7987b1e263582c68835412d2963eeb0a3d60 repl_offset:895807918761 write_command_cou
```

nt:6102523 finish_command_count:6102137 last_replack_time:1645683656 #queue send_write_pos:101832 send_read_pos:101742 response_write_pos:101742 response_read_pos:101357 errtime:1645683657 errmsg:get rsp error:CROSSSLOT Keys in request don't hash to the same slot command:*3 $6 RENAME $16 dispatch:km:pool $34 dispatch:km:tmp-pool:1645683651224 ready shutdown dts send replconf ack to src fail:Bad file descriptor

问题原因

数据库含有多 key 操作或有事务操作，涉及跨 slot 操作。

解决方法

请先迁移数据至云上标准架构的实例，或者更改业务逻辑清理多 key 操作。

问题7

问题现象

在使用 DTS 迁移过程中，提示如下错误信息：

[launch]state:7 #rdb rdbfile:./tmp1633836033_79441.rdb rdbsize:1008499748 rdb_writed_size:1008499748 rdb_parsed_size:607311937 rdb_parsed_begin:1633836038 rdb_parsed_time:0 #replication master_replid:d42935b9537b1d76ddd9e99e7cb8d4bc22a3e0c3 repl_offset:4649070152868 write_command_count:1569934 finish_command_count:1546843 last_replack_time:1633836088 #queue send_write_pos:69933 send_read_pos:69934 response_write_pos:69934 response_read_pos:46843 errtime:1633836089 errmsg:send replconf ack to src fail:Connection reset by peer rdb parse error: Wrong RDB checksum rdb load fail ready shutdown dts

在源节点运行日志里可以看到如下信息：

```
44:M 05 Jun 03:31:06.728 * Starting BGSAVE for SYNC with target: disk
44:M 05 Jun 03:31:06.978 * Background saving started by pid 89
89:C 05 Jun 03:32:08.417 # Error moving temp DB file temp-89.rdb on the final destination 20617.20324.rdb (in server root dir /opt/data/dump): No such file or directory
44:M 05 Jun 03:32:08.698 # Background saving error
44:M 05 Jun 03:32:08.698 # Connection with slave 10.xx.xx.119:<unknown-slave-port> lost.
44:M 05 Jun 03:32:08.698 # SYNC failed. BGSAVE child returned an error
44:M 05 Jun 03:50:24.626 * Slave 10.xx.xx.119:<unknown-slave-port> asks for synchronization
44:M 05 Jun 03:50:24.626 * Full resync requested by slave 10.xx.xx.119:<unknown-slave-port>
44:M 05 Jun 03:50:24.626 * Starting BGSAVE for SYNC with target: disk
44:M 05 Jun 03:50:24.880 * Background saving started by pid 90
90:C 05 Jun 03:51:22.585 * DB saved on disk
90:C 05 Jun 03:51:22.739 * RDB: 280 MB of memory used by copy-on-write
44:M 05 Jun 03:51:23.008 * Background saving terminated with success
44:M 05 Jun 03:51:27.898 * Synchronization with slave 10.xx.xx.119:<unknown-slave-port> succeeded
```

```
44:M 05 Jun 03:52:19.531 # Connection with slave client id #317862457 lost.
```

问题原因

这种现象常见为网络环境存在问题、数据库存在大 key 或者源端的 client-output-buffer-limit 溢出导致 DTS 任务连接源节点超时。

解决方法

- 排查源端网络环境是否存在问题。
- 清理源端数据库存在的大 key。
- 执行如下命令，调整源端的 client-output-buffer-limit 无限大。

```
config set client-output-buffer-limit 'slave 0 0 0'
```

问题8

问题现象

在使用 DTS 迁移过程中，提示如下错误信息：

```
[launch]state:7 #rdb rdbfile:./tmp1654365384_70581.rdb rdbsize:1664871634 rdb_writed_size:1664871634 rdb_parsed_size:1266531 rdb_parsed_begin:1654365387 rdb_parsed_time:0 #replication master_replid:d3e707ec0e72c3908b0ce70dd2460f48086c5386 repl_offset:683001122815 write_command_count:17818 finish_command_count:11224 last_replack_time:0 #queue send_write_pos:30251 send_read_pos:17767 response_write_pos:17767 response_read_pos:11213 errtime:1654365387 errmsg:rdb parse error: Short read or OOM loading DB. Unrecoverable error rdb load fail ready shutdown dts
```

问题原因

一般在 DTS 任务失败后，再次重试时提示此错误信息，该信息说明目标节点非空或内存已经存满。

解决方法

请清空目标节点后再重试。

问题9

问题现象

在使用 DTS 迁移过程中，提示如下错误信息：

```
[launch]state:8 #rdb rdbfile:./tmp1653290250_19158.rdb rdbsize:1721160435 rdb_writed_size:1721160
435 rdb_parsed_size:1721160435 rdb_parsed_begin:1653290255 rdb_parsed_time:124 #replication mas
ter_replid:ed87c56060bc5f9b28da6d7ef2f83a15d56a4827 repl_offset:239048673513 write_command_c
ount:360526495 finish_command_count:360520725 last_replack_time:1654350624 #queue send_write_
pos:406694 send_read_pos:406694 response_write_pos:406694 response_read_pos:400925 errtime:165
4350625 errmsg:redisBufferRead read rsp from target fail:1:Connection reset by peer ready shutdown
dts send replconf ack to src fail:Bad file descriptor
```

问题原因

目标实例 Redis 节点发生了 HA 主备切换，或者 Proxy 节点发生了故障切换，导致同步任务失败。

解决方法

请重新创建 DTS 任务，配置 HA 切换后的新节点为数据迁移的目标节点。

问题10

问题现象

在使用 DTS 迁移过程中，目标实例的内存驱逐策略设置为 allkey-lru，提示如下错误信息：

```
[launch]state:8 #rdb rdbfile:./tmp1638263556_29975.rdb rdbsize:597343276 rdb_writed_size:59734327
6 rdb_parsed_size:428299275 rdb_parsed_begin:1638263575 rdb_parsed_time:7 #replication master_re
plid:ae0dfc45f72f3ee8642c8e31e493b6442179734f repl_offset:34832262785 write_command_count:68
11 finish_command_count:6798 last_replack_time:1638263582 #queue send_write_pos:6811 send_read
_pos:6811 response_write_pos:6811 response_read_pos:6799 errtime:1638263583 errmsg:get rsp error:
OOM command not allowed when used memory > 'maxmemory'. command:*3 $3 SET $26 all_busines
s_newmikoxmsong $1189783 [{"id":3,"label":"AI\u5e73\u53f0\u90e8","node_level":0,"leaf":false,"childre
n":[{"id":"887381","label":"[OMG][\u4f53\u80b2\u641c\u7d22][CMDB]","node_level":1,"leaf":false,"child
ren":[{"id":"722605","label":"[\u4e2a\u6027\u5316\u63a8\u8350\u4e2d\u5fc3][\u817e\u8baf\u7f51\u4f
53\u80b2APP\u63a8\u8350]","node_level":2,"leaf":false,"children":[],"collet":0}],"collet":0},{"id":"46087
1","label":"[OMG][\u817e\u8baf\u89c6\u9891\u641c\u7d22][CMDB]","node_level":1,"leaf":false,"childre
n":[{"id":"383393","label":"[\u641c\u7d22\u4e1a\u52a1\u4e2d\u5fc3][\u817e\u8baf\u89c6\u9891\u641
c\u7d22]","node_level":2,"leaf":false,"children":[],"collet":0}],"collet":0}]},{"id":8,"label":"IDC\u5e73\u53f0
\u90e8","node_level":0,"leaf":false,"children":[{"id":"452519","label":"IDC\u5e73\u53f0\u90e8\u81ea\u7
528[CMDB]","node_level":1,"leaf":false,"children":[{"id":"453099","label":"IDC\u7cfb\u7edf\u5f00\u53d
1","node_level":2,"leaf":false,"children":[],"collet":0}],"collet":0}]},{"id":14 ready shutdown dts send replc
onf ack to src fail:Bad file descriptor
```

或出现如下报错，均属于同一类报错：

问题原因

目标实例内存容量小于源库待迁移数据所占内存。

解决方法

扩容目标实例内存容量后再重新发起新 DTS 迁移任务。

问题11

问题现象

在使用 DTS 迁移过程中，提示如下错误信息：

问题原因

使用代理方式启动 DTS 迁移任务，在任务启动阶段报错，通常是由于代理的带宽容量不足导致。

解决方法

扩容代理带宽或者串行执行迁移任务。

问题12

问题现象

在使用 DTS 迁移过程中，提示如下错误信息：

```
[launch]SrcInstance nodes has changed.
```

问题原因

源节点发生 HA 切换主备节点，导致 DTS 任务同步失败。

解决方法

重新创建新的 DTS 任务，配置 HA 切换后的新节点为数据迁移的目标节点。

数据同步
数据同步支持的数据库

场景概述

数据同步功能指两个数据源之间的数据实时同步，数据同步属于持续性任务，任务创建后会一直同步数据（几乎实时同步），保持源库和目标库的数据一致性。
数据同步和数据迁移的基本功能类似，但数据同步可支持在源和目标端同时进行数据写入，增加了数据冲突处理、任务配置动态调整、构建复杂拓扑结构等功能，更方便用户对长期任务的管理。
DTS 支持源端/目标端为自建数据库、云数据库、第三方云厂商数据库的同步，典型场景示例如下。

场景	说明
云上云下同步	将本地 IDC 自建数据库、CVM 自建数据库，同步到云数据库实例，并且支持建立反向同步，实现云上云下双向同步。
多云厂商间同步	将第三方云厂商数据库同步到云数据库实例，实现双云同步。
云数据库实例间同步	跨地域、跨境数据库同步，以及不同云账号下的数据库实例同步等。
IDC 数据库之间同步	本地 IDC 自建数据库之间的相互同步，实现数据打通和汇聚。

功能说明

同步数据类型

支持结构同步、全量同步和增量同步。

- 结构同步：同步任务配置中，初始化类型选择“结构初始化”。
- 全量同步：同步任务配置中，初始化类型选择“结构初始化+全量初始化”。
- 增量同步：同步任务配置中，初始化类型都不勾选，然后在同步操作类型中选择 DML 和 DDL 。

接入方式

DTS 支持的接入方式有“公网/云主机自建/专线接入/云数据库/私有网络 VPC”，数据库的部署形态不同，可以选择的接入方式也不同。每种接入方式需要具备相应网络条件，请参考 网络准备概述。

- IDC 自建数据库/其他云厂商数据库：接入方式可选择“公网/专线接入”。
- CVM 上的自建数据库：接入方式选择“云主机自建”。

- 云数据库实例：接入方式选择“云数据库”。

同步至 MySQL

MySQL > MySQL

源数据库及版本	目标数据库及版本	双向同步	参考文档
自建数据库 MySQL（IDC 自建/CVM 自建）	- 自建数据库 MySQL - 第三方云厂商 MySQL	支持	MySQL/MariaDB/Percona 同步至 MySQL
第三方云厂商 MySQL	• 自建数据库 MySQL • 第三方云厂商 MySQL		

MariaDB > MySQL

源数据库及版本	目标数据库及版本	双向同步	参考文档
自建数据库 MariaDB（IDC 自建/CVM 自建）	自建数据库 MySQL	支持	MySQL/MariaDB/Percona 同步至 MySQL
云数据库 MariaDB	自建数据库 MySQL		

Percona > MySQL

源数据库及版本	目标数据库及版本	双向同步	参考文档
自建数据库 Percona（IDC 自建/CVM 自建）	自建数据库 MySQL	支持	MySQL/MariaDB/Percona 同步至 MySQL

TDSQL MySQL > MySQL

源数据库	目标数据库	双向同步	参考文档
云数据库 TDSQL MySQL	- 自建数据库 MySQL - 云数据库 MySQL	不支持	TDSQL MySQL 同步至 MySQL

同步至 MariaDB

MySQL > MariaDB

源数据库及版本	目标数据库及版本	双向同步	参考文档
自建数据库 MySQL（IDC 自建/CVM 自建）	- 自建数据库 MariaDB - 云数据库 MariaDB	支持	MySQL/MariaDB/Percona 同步至 MariaDB

MariaDB > MariaDB

源数据库及版本	目标数据库及版本	双向同步	参考文档
自建数据库 MariaDB（IDC 自建/CVM 自建）	- 自建数据库 MariaDB - 云数据库 MariaDB	支持	MySQL/MariaDB/Percona 同步至 MariaDB
云数据库 MariaDB	• 自建数据库 MariaDB • 云数据库 MariaDB		

Percona > MariaDB

源数据库及版本	目标数据库及版本	双向同步	参考文档
自建数据库 Percona（IDC 自建/CVM 自建）	- 自建数据库 MariaDB - 云数据库 MariaDB	支持	MySQL/MariaDB/Percona 同步至 MariaDB

TDSQL MySQL > MariaDB

源数据库及版本	目标数据库及版本	双向同步	参考文档
云数据库 TDSQL MySQL	- 自建数据库 MariaDB - 云数据库 MariaDB	不支持	TDSQL MySQL 同步至 MariaDB

同步至 TDSQL MySQL

MySQL > TDSQL MySQL

源数据库及版本	目标数据库及版本	双向同步	参考文档
自建数据库 MySQL （IDC 自建/CVM 自建）	云数据库 TDSQL MySQL	不支持	MySQL/MariaDB/Percona 同步至TDSQL MySQL

MariaDB > TDSQL MySQL

源数据库及版本	目标数据库及版本	双向同步	参考文档
自建数据库 MariaDB（IDC 自建/CVM 自建）	云数据库 TDSQL MySQL	不支持	MySQL/MariaDB/Percona 同步至TDSQL MySQL
云数据库 MariaDB	云数据库 TDSQL MySQL		

Percona > TDSQL MySQL

源数据库及版本	目标数据库及版本	双向同步	参考文档
自建数据库 Percona（IDC 自建/CVM 自建）	云数据库 TDSQL MySQL	不支持	MySQL/MariaDB/Percona 同步至TDSQL MySQL

TDSQL MySQL > TDSQL MySQL

源数据库及版本	目标数据库及版本	双向同步	参考文档
云数据库 TDSQL MySQL	云数据库 TDSQL MySQL	不支持	TDSQL MySQL 同步至 TDSQL MySQL

云数据库跨账号实例间同步

操作场景

云数据库跨账号实例间数据同步，指源库和目标库都属于云数据库实例，但所属不同的主账号名下，这种数据库实例之间的数据同步。本章节介绍通过 DTS 数据同步功能实现从其他账号云数据库实例同步数据至本账号下云数据库实例。

支持范围

支持云数据库实例 TDSQL MySQL/MariaDB 进行跨账号之间的数据同步。

前提条件

已创建目标数据库实例。

注意事项

本操作中涉及多处账号信息配置，如下列出了主要配置逻辑，以方便用户理解和正确配置。

- 数据同步方向：源数据库（其他账号的数据库实例）> 目标数据库（本账号的数据库实例）。
- 执行同步任务的账号可以是目标数据库的主账号，也可以目标数据库的子账号。
  - 使用主账号执行同步任务。操作任务前，需要请求源数据所属主账号持有人进行角色授权（给目标数据库的主账号），使目标数据库主账号可以访问源数据库。
  - 使用子账号执行同步任务。操作任务前，需要先请求源数据所属主账号持有人进行角色授权（给目标数据库的主账号），使目标数据库主账号可以访问源数据库。然后再请求目标数据库所属主账号持有人进行策略授权（对目标数据库的子账号），使目标数据库子账号可以访问源数据库。

授权账号

使用主账号执行同步任务，请操作步骤1 - 6，使用子账号执行同步任务，请操作步骤1 - 11。

1. 使用源数据库所属的主账号登录 访问管理控制台（如果子账号有 CAM 和角色相关的权限，也可以使用子账号登录）。

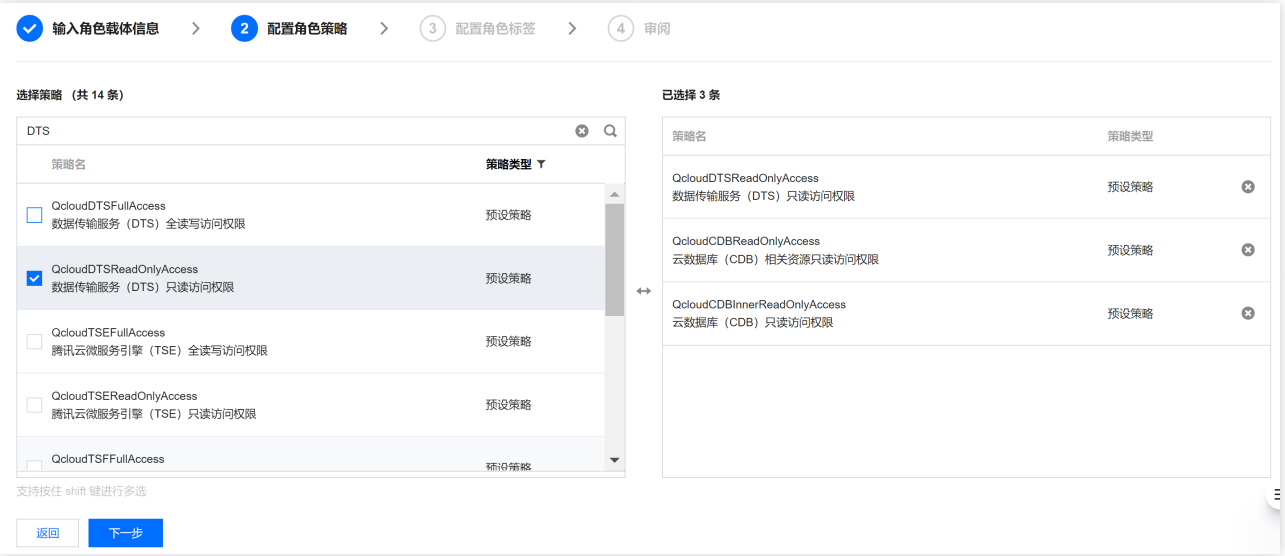
2. 左侧导航单击角色，进入角色管理页面，然后单击新建角色。

3. 配置相关信息，单击下一步。

- 角色名称：配置后请记录，后续配置 DTS 任务时需要输入。
- 角色载体：填入目标数据库所属的主账号 ID。目标数据库实例属于子账号名下时，此处也填写主账号 ID。

4. 在配置角色策略页面，选择 DTS 策略和源数据库服务对应策略，单击下一步。

   i. DTS 服务策略，选择 QcloudDTSReadOnlyAccess。
   ii. 源数据库服务对应的策略。
   - 源数据库为云数据库 MariaDB，选择 QcloudMariaDBReadOnlyAccess（云数据库 MariaDB 只读访问权限）。
   - 源数据库为云数据库 TDSQL MySQL，选择 QcloudTDSQLReadOnlyAccess（TDSQL MySQL 版（TDSQL for MySQL）只读访问权限）。

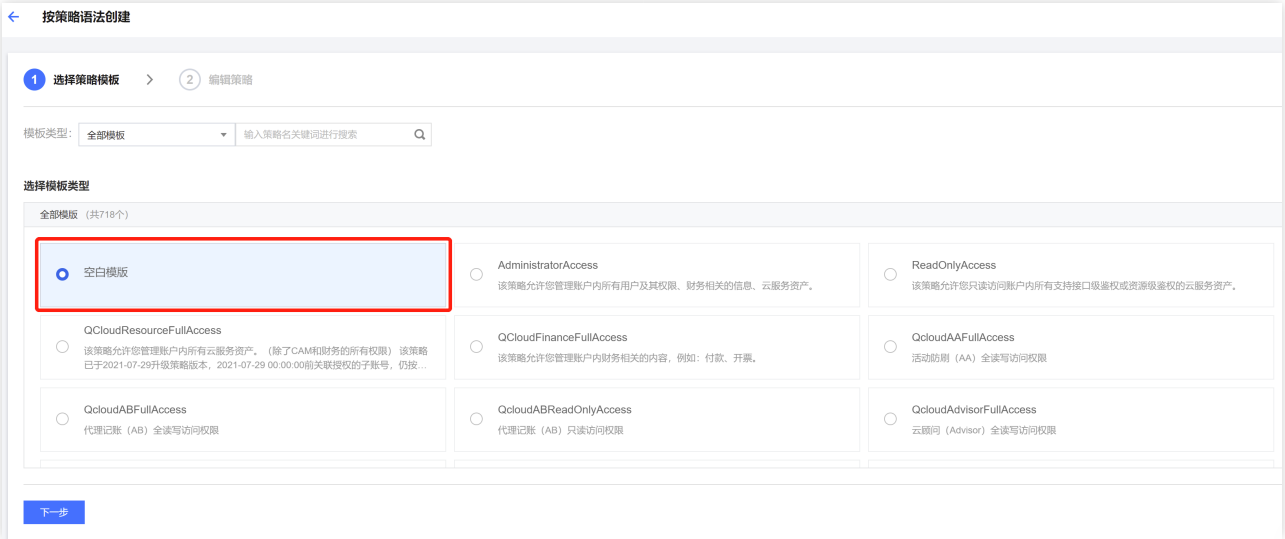


5. 在审阅页面，单击完成后该角色创建完成。

> 说明：
>
> 如果执行同步任务的账号为主账号，授权步骤到此结束；如果为子账号，还需要继续进行如下步骤7-11，请求当前主账号持有人，对子账号进行策略授权。

6. （可选）使用目标数据库所属的主账号登录访问管理控制台，在左侧导航单击策略，然后在右侧单击新建自定义策略，并选择按策略语法创建。

7.（可选）选择空白模板，然后单击下一步。

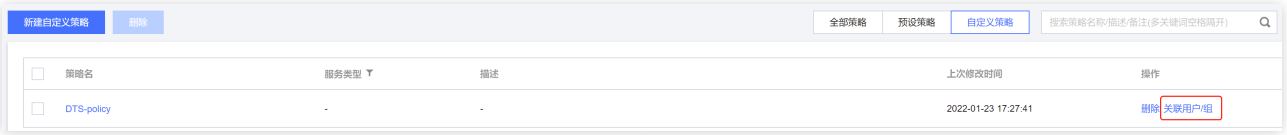


8. （可选）创建一个策略，策略的名称以及描述可以根据自己的需求填写，策略内容复制示例代码后，将红框中的内容替换成对应的信息。

策略语法示例：

```
{
    "version": "2.0",
    "statement": [
    {
        "effect": "allow",
        "action": ["name/sts:AssumeRole"],
  "resource": ["qcs::cam::uin/10*******8:roleName/DTS-role"]
    }
 ]
}
```

9. （可选）单击完成后返回到策略列表页，在列表页中单击关联用户/组。



10. （可选）选择目标数据库实例所属子账号（即执行同步任务的子账号），单击确定，如下图所示。

创建同步任务

1. 使用目标数据库实例所属账号，登录 DTS 控制台。
2. 选择数据同步 > 新建同步任务，购买一个新的同步任务。
3. 购买完成后，返回数据同步列表，单击操作列的配置，进入配置同步任务页面。
4. 在设置源和目标数据页面，配置源库和目标库信息。

跨账号关键参数配置如下：

- 接入类型：选择云数据库，表示源数据库属于云数据库实例。
- 是否跨账号：选择跨账号。
- 跨云账号 ID： 填入源数据库所属的主账号 ID。
- 跨账号授权角色名称。即前文授权账号步骤6中创建的角色名称。
- 外部角色 ID：如果前文授权账号步骤4中设置了外部 ID，则这里则必须按照设置的内容正确填写，否则会发生权限报错。没有设置则不需要填写。

5. 在设置同步选项和同步对象页面，对数据初始化选项、数据同步选项、同步对象选项进行设置，在设置完成后单击保存并下一步。

6. 在校验任务页面，完成校验并全部校验项通过后，单击启动任务。
7. 返回数据同步任务列表，任务开始进入运行中状态。

常见问题

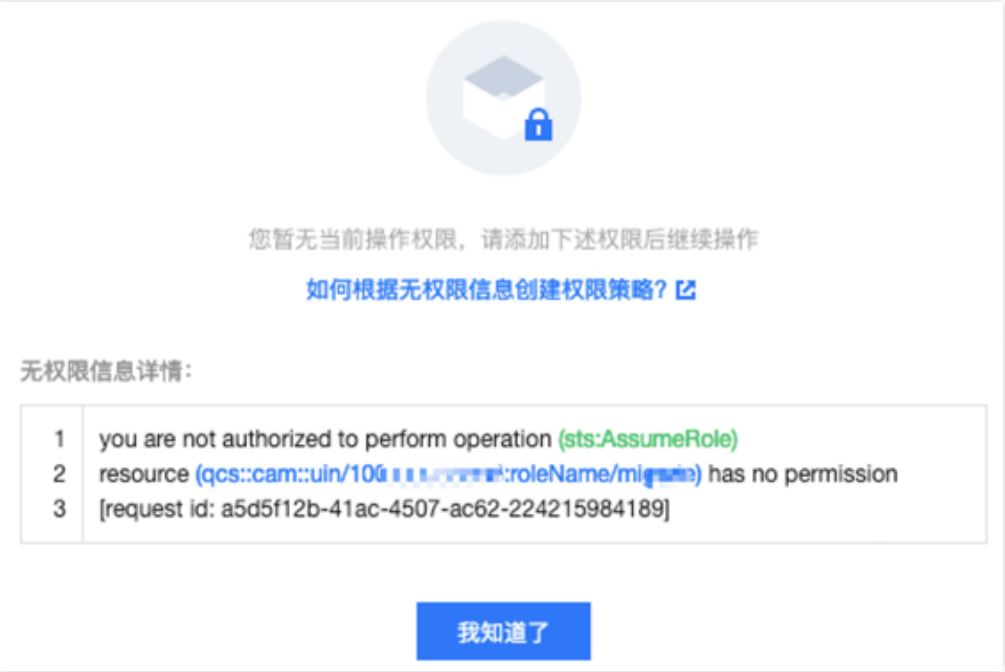
1. 跨账号拉取实例列表报错：role not exist[InternalError.GetRoleError]

请确认跨云账号 ID（应该为源数据库的主账号 ID）和跨账号授权角色名称（应该为授权账号步骤6中创建的角色名称）配置是否正确。如果还无法拉取，可能没有授权源数据库的服务权限（参考授权账号中的步骤5）。

2. 获取云数据库实例列表失败：InternalError:InternalInnerCommonError

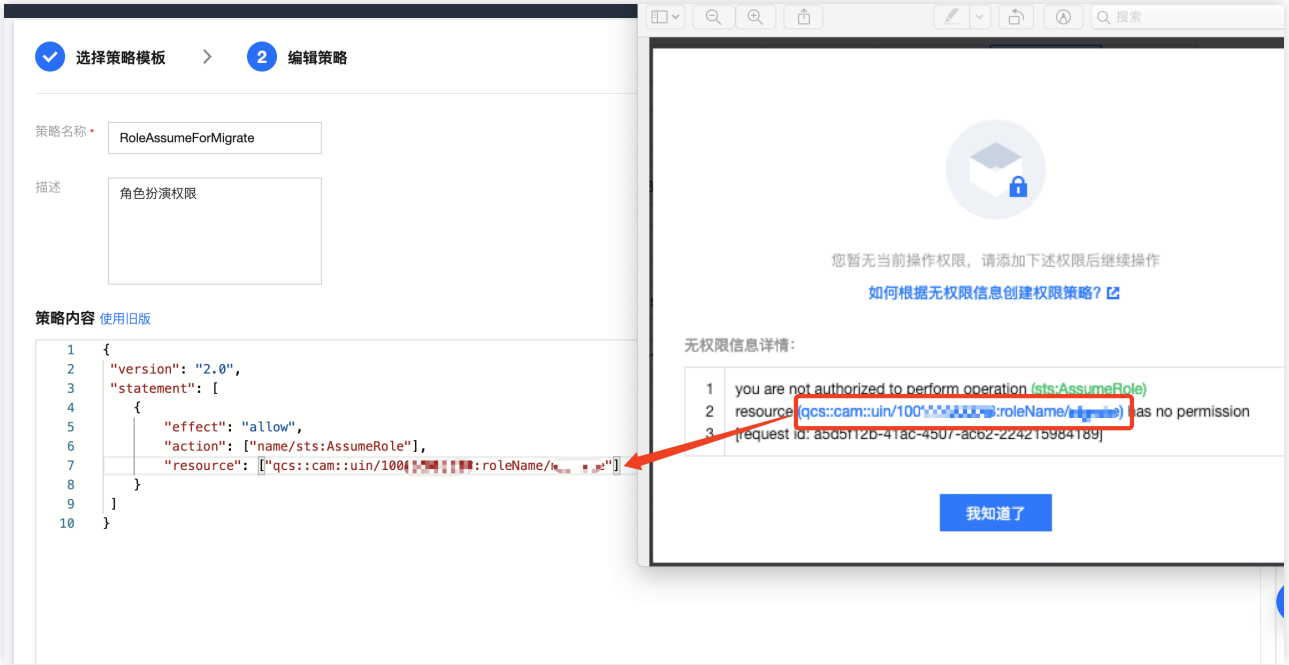
角色中没有授权源数据库所属的云服务策略，请参考授权账号中的步骤5进行授权。

3. 跨账号拉取实例列表报错：you are not authorized to perform operation (sts:AssumeRole) ，resource (qcs::cam::uin/1xx5:roleName/xxxx) has no permission



出错原因：您当前创建同步任务使用的账号是子账号，并且当前子账号没有 sts:AssumeRole 权限。
解决方法：

- 使用主账号来创建同步任务。
- 请求目标数据库所属的主账号持有人，参考授权账号对子账号授权，策略语法中的 resource，填写报错框中蓝色字段部分。



4. DTS 任务配置未填写角色外部 ID，或者填写错误，导致权限报错或者拉取数据库实例失败。

出错原因：授权账号时设置了外部 ID，但在 DTS 任务配置中未填写外部 ID，或者填写错误，导致权限报错或者拉取数据库示例失败。

解决方案：授权账号时如果设置了外部 ID，则 DTS 任务配置时需要正确填写，未设置则不需要填。

同步至 MySQL
MySQL/MariaDB/Percona 同步至 MySQL

操作场景

本文为您介绍使用数据传输服务 DTS 从 MySQL、MariaDB、Percona 数据库同步数据至 MySQL 的过程。

业务影响

1. DTS 在执行全量数据同步时，会将源库的全量数据全部读取一次，所以会增加源库的负载。如果您的数据库规格过低，建议您在业务低峰期进行同步任务，或者在任务启动前降低 DTS 的速率。
   源库的规格不同，影响也不同，以源库规格为 8 核 16G 为例，DTS 任务默认采用 8 线程并发（可调整），在网络无瓶颈的情况下，DTS 任务对源库的性能影响如下：
   - DTS 全量导出阶段：占用源库约 18%-45% 的 CPU，增加源库约 40-60MB/s 的查询压力，占用约 8 个活跃 session 连接数。
   - DTS 增量导出阶段：对源数据库基本无压力，只有一个连接实时监听源库的 binlog 日志。
2. 默认采用无锁方式，同步过程中对源库不加全局锁（FTWRL），仅对无主键的表加表锁，其他不加锁。
3. 数据同步时，DTS 会使用执行同步任务的账号在源库中写入系统库 __tencentdb__ ，用于记录同步任务过程中的数据对比信息，请勿删除该系统库。
   - 为保证后续数据对比问题可定位，同步任务结束后不会删除源库中的 __tencentdb__ 。
   - __tencentdb__ 系统库占用空间非常小，约为源库存储空间的千分之一到万分之一（例如源库为50GB，则 __tencentdb__ 系统库约为5MB-50MB），并且采用单线程，等待连接机制，所以对源库的性能几乎无影响，也不会抢占资源。

准备工作

1. 请根据您需要使用的接入类型，提前打通 DTS 与数据库之间的访问通道，具体请参考 网络准备。
   - IDC 自建数据库/其他云厂商数据库：接入方式可选择“公网/专线接入”。
   - CVM 上的自建数据库：接入方式选择“云主机自建”。
   - 云数据库实例：接入方式选择“云数据库”。

2. 在源数据库中对执行任务账号进行授权，参考如下：

```
GRANT RELOAD,LOCK TABLES,REPLICATION CLIENT,REPLICATION SLAVE,SHOW DATABASES,SHOW VIEW,PROCESS,SELECT ON *.* TO '账号'@'%' IDENTIFIED BY '密码';
```

```
//源库为阿里云数据库时，不需要授权 SHOW DATABASES，但需要创建至少一个非系统库，否则前置校验任务会失败。其他非阿里云数据库场景则需要授权。阿里云数据库授权，请参考 https://help.aliyun.com/document_detail/96101.html
GRANT ALL PRIVILEGES ON `__tencentdb__`.* TO '账号'@'%';
FLUSH PRIVILEGES;
```

3. 在目标库中对执行任务账号进行授权，需要具备的权限如下。
   ALTER, ALTER ROUTINE, CREATE, CREATE ROUTINE, CREATE TEMPORARY TABLES, CREATE USER, CREATE VIEW, DELETE, DROP, EVENT, EXECUTE, INDEX, INSERT, LOCK TABLES, PROCESS, REFERENCES, RELOAD, SELECT, SHOW DATABASES, SHOW VIEW, TRIGGER, UPDATE。

使用说明

同步对象

1. 只支持同步基础表、视图、存储过程和函数。
2. 相互关联的数据对象需要一起同步，否则会导致同步失败。常见的关联关系：视图引用表、视图引用视图、主外键关联表等。
3. 为了避免数据重复，建议同步有主键或者非空唯一键的表。
4. 在同步视图、存储过程和函数时，DTS 会检查源库中 DEFINER 对应的 user1（[DEFINER = user1]）和同步账号 user2 是否一致，如果不一致，同步后 DTS 会修改 user1 在目标库中的 SQL SECURITY 属性，由 DEFINER 转换为 INVOKER（[INVOKER = user1]），同时设置目标库中 DEFINER 为同步账号 user2（[DEFINER = 同步账号 user2]）。如果源库中视图定义过于复杂，可能会导致任务失败。
5. 源数据库为阿里云 MySQL，则阿里云 MySQL 5.6 版本待同步表不能存在无主键表，MySQL 5.7 及以后版本不限制。
6. 只支持同步 InnoDB、MyISAM、TokuDB 三种数据库引擎，如果存在这三种以外的数据引擎表则默认跳过不进行同步。其中，源库如果存在压缩模式的 TokuDB 引擎数据，需要目标库同步支持压缩模式才可以同步，否则任务会报错。

数据类型

1. 增量同步过程中，若源库产生了类型为 STATEMENT 格式的 Binlog 语句，则会导致同步失败。
2. 源数据库 Binlog 的 GTID 如果存在空洞，可能会影响同步任务的性能并导致任务失败。
3. 不支持同时包含 DML 和 DDL 语句在一个事务的场景，遇到该情况任务会报错。
4. 不支持 Geometry 相关的数据类型，遇到该类型数据任务报错。
5. 不支持 ALTER VIEW 语句，遇到该语句任务跳过不同步。

HA 切换

源端如果是非 GTID 实例，DTS 不支持源端 HA 切换，一旦源端 MySQL 发生切换可能会导致 DTS 增量同步中断。

主键冲突策略

主键冲突策略选择“冲突报错”时，在任务的全量同步阶段，请不要在目标端进行双写，否则发生的主键冲突问题，DTS 不会

提示报错，会用源端的数据覆盖掉目标端。

操作限制

同步过程中请勿进行如下操作，否则会导致同步任务失败。

1. 选择结构初始化和全量数据初始化场景中，当同步任务步骤为“全量导出”、“全量导入”时，不支持 DDL 操作。
2. 请勿修改、删除源数据库和目标数据库中用户信息（包括用户名、密码和权限）和端口号。
3. 请勿在源库上执行清除 Binlog 的操作。

支持同步的 SQL 操作

操作类型	SQL 操作语句
DML	INSERT、UPDATE、DELETE
DDL	CREATE DATABASE、DROP DATABASE、ALTER DATABASE、CREATE TABLE、ALTER TABLE、DROP TABLE、TRUNCATE TABLE、RENAME TABLE、CREATE VIEW、DROP VIEW、CREATE INDEX、DROP INDEX

> 说明：
>
> 暂不支持 CREATE TABLE 表名 AS SELECT 语句。

操作步骤

1. 登录 DTS 控制台， 选择“数据同步-> 创建同步任务”，进行相应配置后，单击立即创建。
2. 创建完成后，返回数据同步列表，可看到刚创建的数据同步任务，单击操作列的配置，进入配置同步任务页面。

3. 在配置同步任务页面，配置源端实例、账号密码，配置目标端实例、账号和密码，测试连通性后，单击下一步。

- 任务设置

参数	描述
任务名称	DTS 会自动生成一个任务名称，建议用户修改为一个具有业务意义的名称，便于任务识别。
运行模式	- 立即执行：前置校验通过后会立即启动任务。 - 定时执行：设置一个任务开始执行的时间，前置校验通过后不启动任务，到设定的时间再启动。

参数	描述
自动重试	设置后，同步任务因网络异常等引起的任务临时中断，DTS 将在设置的时间范围内自动重试和恢复任务，不需要用户手动操作。 支持设置的时间范围为5分钟-720分钟。

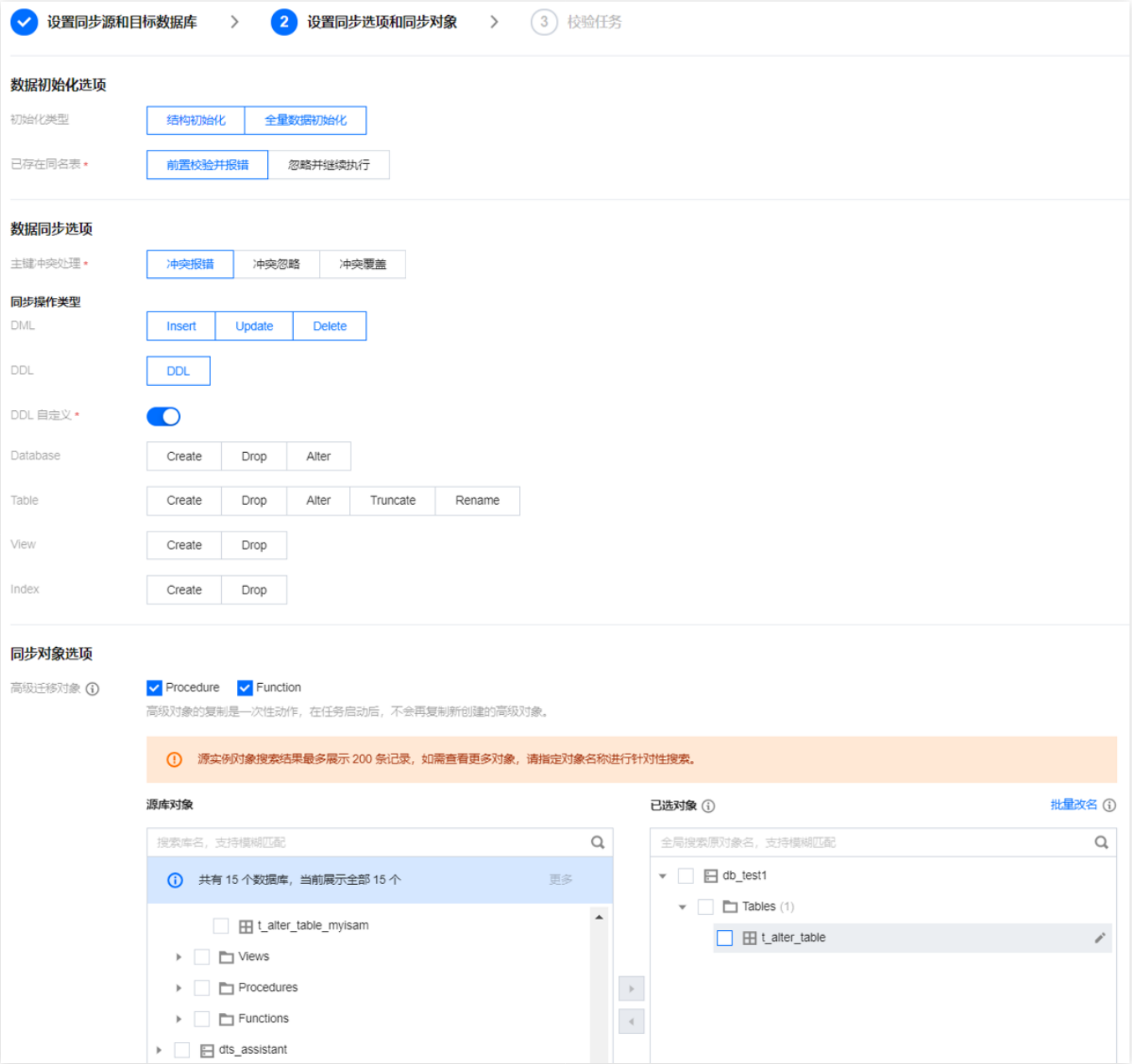
- 源库设置

参数	描述
源实例类型	购买时所选择的源实例类型，不可修改。
源实例地域	购买时选择的源实例所在地域，不可修改。
服务提供商	- 自建数据库（包括云服务器上的自建）、云数据库实例，请选择“普通”。 - 第三方云厂商数据库，请选择对应的服务商。
接入类型	请根据您的场景选择，不同的场景需要打通 DTS 与数据库之间的网络连接，具体请参考 网络准备概述。 - 公网：源数据库可以通过公网 IP 访问。 - 云主机自建：源数据库部署在云服务器 CVM 上。 - 专线接入：源数据库可以通过专线接入方式与私有网络打通。 - 云数据库：源数据库属于云数据库实例。 - 私有网络 VPC：源数据和目标数据库都部署在云平台上，且有私有网络。 选择原则如下： - IDC 自建数据库/其他云厂商数据库：接入方式可选择“公网/专线接入”。 - CVM 上的自建数据库：接入方式选择“云主机自建”。 - 云数据库实例：接入方式选择“云数据库”。
公网	接入类型选择“公网”时，需要配置如下参数。 - 主机地址：源数据库 IP 地址或域名。 - 端口：源数据库使用的端口。
云主机自建	接入类型选择“云主机自建”时，需要配置如下参数。 - 云主机实例：云服务器 CVM 的实例 ID。 - 端口：源数据库使用的端口。
专线接入	接入类型选择“专线接入”时，需要配置如下参数。 - 私有网络专线网关：专线接入时只支持私有网络专线网关，请确认网关关联的网络类型。 - 私有网络：选择私有网络和子网。 - 主机地址：源数据库 IP 地址。 - 端口：源数据库使用的端口。
云数据库	接入类型选择“云数据库”时，需要配置如下参数。 - 云数据库实例：源数据库的实例 ID。
账号/密码	账号/密码：源数据库的账号、密码。

- 目标库设置
  目标库参数设置与源库类似，接入类型根据实际的情况选择，此处不再赘述。

4. 在设置同步选项和同步对象页面，将对数据初始化选项、数据同步选项、同步对象选项进行设置，在设置完成后单击保存并下一步。

> 说明：
>
> - 当初始化类型仅选择全量数据初始化，系统默认用户在目标库已经创建了表结构，不会进行表结构同步，也不会校验源库和目标库是否有同名表，所以当用户同时在已存在同名表中选择前置校验并报错，则校验并报错功能不生效。
> - 如果用户在同步过程中确定会对某张表使用 rename 操作（例如将 table A rename 为 table B），则同步对象需要选择 table A 所在的整个库（或者整个实例），不能仅选择 table A，否则 rename 操作后，table B 的数据不会同步到目标库。



设置项	参数	描述
数据初始化选项	初始化类型	• 结构初始化：同步任务执行时会先将源实例中表结构初始化到目标实例中。 • 全量数据初始化：同步任务执行时会先将源实例中数据初始化到目标实例中。仅选择全量数据初始化的场景，用户需要提前在目标库创建好表结构。 默认两者都勾上，可根据实际情况取消。
	已存在同名表	• 前置校验并报错：存在同名表则报错，流程不再继续。 • 忽略并继续执行：全量数据和增量数据直接追加目标实例的表中。

设置项	参数	描述
数据同步选项	冲突处理机制	• 冲突报错：在同步时发现表主键冲突，报错并暂停数据同步任务。 • 冲突忽略：在同步时发现表主键冲突，保留目标库主键记录。 • 冲突覆盖：在同步时发现表主键冲突，用源库主键记录覆盖目标库主键记录。
	同步操作类型	支持操作：Insert、Update、Delete、DDL。勾选“DDL 自定义”，可以根据需要选择不同的 DDL 同步策略。
同步对象选项	源实例库表对象	选择待同步的对象，支持基础库表、视图、存储过程和函数。高级对象的同步是一次性动作，仅支持同步在任务启动前源库中已有的高级对象，在任务启动后，新增的高级对象不会同步到目标库中。
	已选对象	• 支持库表映射（库表重命名），将鼠标悬浮在库名、表名上即显示编辑按钮，单击后可在弹窗中填写新的名称。 • 选择高级对象进行同步时，建议不要进行库表重命名操作，否则可能会导致高级对象同步失败。
	是否同步 Online DDL 临时表	如果使用 gh-ost、pt-osc 工具对源库中的表执行 Online DDL 操作，DTS 支持将 Online DDL 变更产生的临时表同步到目标库。 • 勾选 gh-ost，DTS 会将 gh-ost 工具产生的临时表名（`_表名_ghc`、`_表名_gho`、`_表名_del`）同步到目标库。 • 勾选 pt-osc， DTS 会将 pt-osc 工具产生的临时表名（`_表名_new`、 `_表名_old`）同步到目标库。

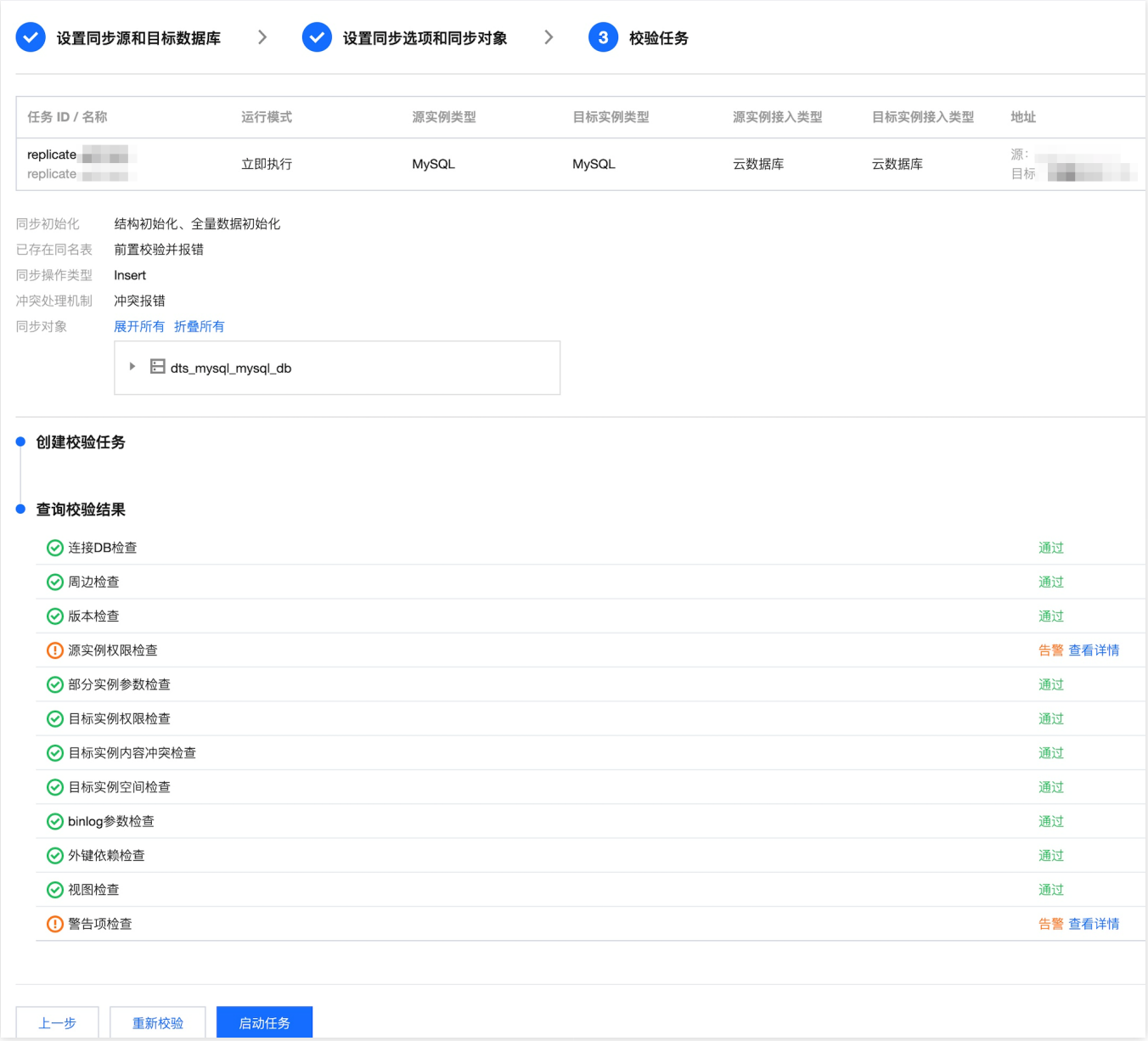
高级选项

参数	描述

参数	描述
限制传输速率	一般情况下任务配置这里不需要勾选，仅在数据库的配置较低，无法负载 DTS 默认并发线程数和 RPS 时，需要配置。

5. 在校验任务页面，完成校验并全部校验项通过后，单击启动任务。
6. 如果校验任务不通过，可以参考 前置校验项不通过处理 修复问题后重新发起校验任务。

- 失败：表示校验项检查未通过，任务阻断，需要修复问题后重新执行校验任务。
- 警告：表示检验项检查不完全符合要求，可以继续任务，但对业务有一定的影响，用户需要根据提示自行评估是忽略警告项还是修复问题再继续。



7. 返回数据同步任务列表，任务开始进入运行中状态。

> 说明：
>
> 选择操作列的更多 > 结束可关闭同步任务，请您确保数据同步完成后再关闭任务。

8. （可选）您可以单击任务名，进入任务详情页，查看任务初始化状态和监控数据。

TDSQL MySQL 同步至 MySQL

TDSQL MySQL 同步至云数据库 MySQL 的要求和指导，与 TDSQL MySQL 同步至 TDSQL MySQL 的内容基本一致，请参考相关内容进行操作。

同步至 MariaDB

MySQL/MariaDB/Percona 同步至 MariaDB

从 MySQL/MariaDB/Percona 同步数据至 MariaDB 的操作指导，与 MySQL/MariaDB/Percona 同步至 MySQL 的使用说明和操作步骤一致，请参考进行配置。

TDSQL MySQL 同步至 MariaDB

TDSQL MySQL 同步至云数据库 MariaDB 的要求和指导，与 TDSQL MySQL 同步至 TDSQL MySQL 的内容基本一致，请参考相关内容进行操作。

同步至 TDSQL MySQL
TDSQL MySQL 同步至 TDSQL MySQL
使用说明

类别	说明
同步对象	1. 只支持同步基础表、视图，不支持同步函数、触发器、存储过程等对象。 1. 相互关联的数据对象需要同时同步，否则会导致同步失败。 1. 源端 TDSQL MySQL 中对表的数量有限制，整个实例最多为5000个，超出后 DTS 任务会报错；同时，表的数量太多会导致源端的访问耗时变大，引起性能抖动和下降。 1. 增量同步阶段，源库的表名如含有“TDSQLagent”、“tdsql_sub”字符可能会被过滤或者引起同步异常，因为这些表名与 TDSQL 系统的临时表名相同，TDSQLagent 为扩容时的临时表，tdsql_sub 表为 hash-lish 和 hash-range 的子表，因此建议源端待同步的表名不要设置为这些类型。
源库影响	1. DTS 在执行全量数据同步时，会占用一定源库资源，可能会导致源库负载上升，增加数据库自身压力。如果您数据库配置过低，建议您在业务低峰期进行。 1. 数据同步时，DTS 会使用执行同步任务的账号在源库中写入系统库 __tencentdb__ ，用于记录事务标记 ID 等元信息，需要确保源库对 __tencentdb__ 的读写权限。 - 为保证后续数据对比问题可定位，同步任务结束后不会删除源库中的 __tencentdb__ - __tencentdb__ 系统库占用空间非常小，约为源库存储空间的千分之一到万分之一（例如源库为 50GB，则 __tencentdb__ 系统库约为5MB - 50MB），并且采用单线程，等待连接机制，所以对源库的性能几乎无影响，也不会抢占资源。 1. 默认采用无锁同步方式，全量数据导出阶段不会对源库加全局锁（FTWRL），仅对无主键的表加表锁。
同步功能说明	1. 目前主键冲突处理策略只支持冲突覆盖，对于全量、增量阶段的主键数据冲突，都会进行冲突覆盖。 1. 关于 DDL 的操作约束如下： - 在全量导出阶段，请勿在源库上执行库或表结构变更的 DDL 操作。 - 同步增量阶段，TDSQL MySQL 分布式为源的链路中，源端执行 DDL 时，建议不要在短时间内大并发的执行，有可能会造成链路延迟或者中断。 1. 只支持同步 InnoDB 数据库引擎，如果存在其他数据引擎表则任务校验时会报错。 1. 增量同步期间不支持修改主键，包括主键列、分区表分布键、对主键列的 COMMENT 注释，增删改列字段和长度。 1. 增量同步过程中，若源库产生了类型为 STATEMENT 格式的 Binlog 语句，则会导致同步失败。
操作限制	同步过程中请勿进行如下操作，否则会导致同步任务失败。 1. 同步任务过程中，请勿修改、删除源数据库和目标数据库中用户信息（包括用户名、密码和权限）和端口号。 1. 请勿在源库上执行清除 Binlog 的操作。
数据类型	1. 不支持 Geometry 相关的数据类型，遇到该类型数据任务报错。 1. 对于存在浮点类型的表，可能因为全量和增量同步的精度存在差异，导致同步结果的精度不一致。 1. 源端 TDSQL MySQL 为 MariaDB 10.1.x 内核时，使用 timestamp 类型不支持指定精度（例如

类别	说明
	timestamp(3)），否则 DTS 任务会报错，需要去掉精度，然后重建任务。
事务	1. 不支持同时包含 DML 和 DDL 语句在一个事务的场景，遇到该情况任务会报错。 1. 源库为 TDSQL MySQL（内核 MariaDB 5.6）时，不支持 XA 事务，遇到 XA 事务任务会报错。
HA 切换和扩容	1. 源库如果是非 GTID 数据库，DTS 不支持源端 HA 切换，一旦源端 TDSQL MySQL 发生切换可能会导致 DTS 增量同步中断。 1. 源库如果是自建数据库，采用 SET 连接 TDSQL MySQL 的场景，当同步任务启动后，源端进行了增加或者删除 SET 节点，DTS 同步任务会报错，需要用户修改 DTS 中源数据库的 SET 配置信息（与源端实际的 SET 保持一致），然后重启任务，这样才能同步到新增或者删除 SET 的信息。
分区表同步	1. 全量阶段支持同步一级/二级分区表，但分区语法需要符合 TDSQL MySQL 的规范，一级 Hash 分区表仅支持通过 shardkey 方式创建。 TDSQL MySQL 创建分区表的关键语法如下。 - 一级 Hash 分区：shardkey - 一级 Range 分区：TDSQL_DISTRIBUTED BY RANGE - 一级 List 分区：TDSQL_DISTRIBUTED BY LIST - 一级 Hash 分区 + 二级 Range/List 分区：shardkey + PARTITION BY RANGE/LIST - 一级 Range 分区 + 二级 Range/List 分区：TDSQL_DISTRIBUTED BY RANGE + PARTITION BY RANGE/LIST - 一级 List 分区 + 二级 Range/List 分区：TDSQL_DISTRIBUTED BY LIST + PARTITION BY RANGE/LIST 1. 增量同步阶段，不支持进行密集的 create 二级分区表，再 drop 二级分区表，再 create 二级分区表的操作，否则可能会由于表类型冲突导致任务异常；先 drop 不存在的二级分区表，再 create 二级分区表，可能会导致死锁，任务无报错，需要手动解锁。 1. TDSQL MySQL 同步到 MySQL/MaraDB/Percona 链路，如果源端待同步的库表中包含二级分区表，则同步到目标端后为单表。

同步操作指导

操作场景

本文为您介绍使用数据传输服务 DTS 从 TDSQL MySQL 数据库同步数据至 TDSQL MySQL 数据库的过程。支持的场景如下：

准备工作

1. 请根据您需要使用的接入类型，提前打通 DTS 与数据库之间的访问通道，具体请参考 网络准备概述。
2. 需要您在源端 TDSQL MySQL 中提前创建好数据库： __tencentdb__ （参考语法 CREATE DATABASE __tencentdb__; ）。
3. 在源数据库中对执行任务账号进行授权，参考如下：

```
GRANT RELOAD,LOCK TABLES,REPLICATION CLIENT,REPLICATION SLAVE,SHOW VIEW,PROCESS,SELECT
ON *.* TO '同步账号'@'%' IDENTIFIED BY '同步密码';
GRANT ALL PRIVILEGES ON `__tencentdb__`.* TO '同步账号'@'%';
FLUSH PRIVILEGES;
```

4. 在目标库中对执行任务账号进行授权，需要具备目标数据库的权限：
   ALTER, ALTER ROUTINE, CREATE, CREATE ROUTINE, CREATE TEMPORARY TABLES, CREATE USER, CREATE VIEW, DELETE, DROP, EVENT, EXECUTE, INDEX, INSERT, LOCK TABLES, PROCESS, REFERENCES, RELOAD, SELECT, SHOW DATABASES, SHOW VIEW, TRIGGER, UPDATE。

操作步骤

1. 登录 DTS 控制台， 选择“数据同步-> 创建同步任务”，进行相应配置后，单击立即创建。
2. 创建完成后，返回数据同步列表，可看到刚创建的数据同步任务，刚创建的同步任务需要进行配置后才可以使用。
3. 在数据同步列表，单击操作列的配置，进入配置同步任务页面。

4. 在配置同步任务页面，配置源端实例、账号密码，配置目标端实例、账号和密码，测试连通性后，单击下一步。

设置项	参数	描述
任务设置	任务名称	DTS 会自动生成一个任务名称，用户可以根据实际情况进行设置。
	运行模式	支持立即执行和定时执行两种模式。
	自动重试	勾选后，同步任务因网络异常等引起的任务中断，DTS 将在设置的时间范围内自动重试，不需要用户手动操作。 重试机制为，从头开始重新进行数据同步，重新同步时，之前已同步到目标端的数据会被清空，或者会被覆盖。
源实例设置	源实例类型	购买时所选择的源实例类型，不可修改。
	源实例地域	购买时选择的源实例所在地域，不可修改。
	接入类型	请根据您的场景选择，不同接入类型的准备工作请参考 网络准备概述。 ◦ 公网：源数据库可以通过公网 IP 访问。 ◦ 云主机自建：源数据库部署在云服务器 CVM 上。 ◦ 专线接入：源数据库可以通过专线接入方式与私有网络打通。 ◦ 云数据库：源数据库属于云数据库实例。 ◦ 私有网络 VPC：源数据和目标数据库都部署在云平台上，且有私有网络。
	私有网络专线网关	专线接入时只支持私有网络专线网关，请确认网关关联网络类型。
	私有网络	选择私有网络和子网。
	主机 - Proxy	填入 Proxy 地址和端口。
	节点 - SET	填入 TDSQL MySQL 分片节点 IP，每分片需输入一个节点 IP，多个节点请换行输入。 源库如果是自建数据库，首次配置 SET 节点，连接保存后，无法再增加或者删除，如果用户需要增加或者删除 SET 节点，需要重新建立同步任务。
	账号	源实例账号，账号权限需要满足要求。
	密码	源实例账号的密码。

设置项	参数	描述
目标实例设置	目标实例类型	购买时选择的目标实例类型，不可修改。
	目标实例地域	购买时选择的目标实例地域，不可修改。
	接入类型	根据您的场景选择，本场景选择“云数据库”。如果源实例接入类型选择了云数据库，目标实例接入类型这里可以选择公网/云主机自建/专线接入/云数据库等方式。
	实例 ID	选择目标实例 ID。
	账号	目标实例账号，账号权限需要满足要求。
	密码	目标实例账号的密码。

5. 在设置同步选项和同步对象页面，将对数据初始化选项、数据同步选项、同步对象选项进行设置，在设置完成后单击保存并下一步。

> 说明：
>
> 如果用户在同步过程中确定会对某张表使用 rename 操作（例如将 table A rename 为 table B），则同步对象需要选择 table A 所在的整个库（或者整个实例），不能仅选择 table A，否则 rename 操作后，table B 的数据不会同步到目标库。

设置项	参数	描述
数据初始化选项	初始化类型	• 结构初始化：同步任务执行时会先将源实例中表结构初始化到目标实例中。 • 全量数据初始化：同步任务执行时会先将源实例中数据初始化到目标实例中。仅选择全量数据初始化的场景，用户需要提前在目标库创建好表结构。 默认两者都勾上，可根据实际情况取消。仅选择“全量数据初始化”时，用户需要提前在目标库创建好表结构。
	已存在同名表	• 前置校验并报错：存在同名表则报错，流程不再继续。 • 忽略并继续执行：全量数据和增量数据直接追加目标实例的表中。
数据同步选项	冲突处理机制	冲突覆盖：在同步时发现表主键冲突，用源库主键记录覆盖目标库主键记录。
	同步操作类型	支持操作：Insert、Update、Delete、DDL。
同步对象选项	源实例库表对象	选择待同步的对象，支持库级别和表级别。

设置项	参数	描述
	已选对象	展示已选择的同步对象，将鼠标悬浮在库、表对象上，可以编辑映射规则。 1. 支持库名、表名的映射。 2. 目标端为分布式 TDSQL MySQL 的同步链路中，可以设置源端的表同步到目标端类型。 3. 建表类型：支持 Hash、Range、List。 请注意，设置 Range 或者 List 分区时，需要保证分区策略可以覆盖源端全量+增量的所有数据，否则会导致数据无法按照分区规则写入，从而导致任务报错。 • 一级 Shardkey：选择您需要指定的 Shardkey。

6. 在校验任务页面，完成校验并全部校验项通过后，单击启动任务。
7. 如果校验任务不通过，可以参考 前置校验项不通过处理 修复问题后重新发起校验任务。

- 失败：表示校验项检查未通过，任务阻断，需要修复问题后重新执行校验任务。
- 警告：表示检验项检查不完全符合要求，可以继续任务，但对业务有一定的影响，用户需要根据提示自行评估是忽略警告项还是修复问题再继续。

8. 返回数据同步任务列表，任务开始进入运行中状态。

> 说明：
>
> 选择操作列的更多 > 结束可关闭同步任务，请您确保数据同步完成后再关闭任务。

9. （可选）您可以单击任务 ID，进入任务详情页，查看任务初始化状态和监控数据。

MySQL/MariaDB/Percona 同步至TDSQL MySQL

MySQL/MariaDB/Percona 同步至云数据库 TDSQL MySQL 的要求和指导，与 TDSQL MySQL 同步至 TDSQL MySQL 的内容基本一致，请参考相关内容进行操作。

MySQL 系列前置校验项
连接 DB 检查

检查详情

源数据库和目标数据库需要能正常连通，如果未连通，会报连接失败。

问题原因

- 源数据库所在网络或服务器设置了安全组或防火墙。
- 源数据库对来源 IP 地址进行了限制。
- 网络端口未放通。
- 数据库账号或密码不正确。

修复方法

请按照问题原因中的对应链接进行处理。

周边检查

检查详情

1. 建议源数据库环境变量参数 innodb_stats_on_metadata 设置为 OFF 。
2. 目标库为 MySQL/MariaDB/Percona/TDSQL-C MySQL/TDSQL TDStore 时，建议目标库 explicit_defaults_for_timestamp 设置为 ON 。如果目标库为 OFF ，DTS 会尝试修改目标端上该参数设置为 ON ，仅在会话级别。
   - 目标端为 MySQL 8.0版本，DTS 可以成功修改目标端上会话级别的参数为 ON 。
   - 目标端为 MySQL 8.0以下的版本，或者其他数据库类型，DTS 无法修改成功，则初次校验任务不通过。用户可评估影响后跳过该参数检查，然后重新校验。

是否可跳过

出现校验不通过时，用户可评估影响后自行决定是否修改参数，如确认不需要修改，可跳过参数检查。

- innodb_stats_on_metadata ：可跳过校验。
- explicit_defaults_for_timestamp ：可跳过校验。

修复方法

修改 innodb_stats_on_metadata 参数

- innodb_stats_on_metadata 参数开启时，每当查询 information_schema 元数据库里的表，Innodb 就会更新 information_schema.statistics 表，导致访问时间变长。关闭后可加快对于 schema 库表的访问。
- MySQL 5.6.6 之前版本 innodb_stats_on_metadata 参数预设值为 ON，需要修改为 OFF。MySQL 5.6.6 及其以后的版本预设值为 OFF，不存在问题。

1. 登录源数据库。

2. 修改 innodb_stats_on_metadata 为 OFF 。

```
set global innodb_stats_on_metadata = OFF;
```

3. 查看配置是否生效。

```
show global variables like '%innodb_stats_on_metadata%';
```

系统显示结果类似如下：

```
mysql> show global variables like '%innodb_stats_on_metadata%';
+-------------------------+-------+
| Variable_name           | Value |
+-------------------------+-------+
| innodb_stats_on_metadata | OFF   |
+-------------------------+-------+
1 row in set (0.00 sec)
```

4. 重新执行校验任务。

修改 explicit_defaults_for_timestamp 参数

参数说明

explicit_defaults_for_timestamp 用于控制 MySQL 如何处理 TIMESTAMP 列的默认值。取值说明如下：

- OFF：MySQL 会将 TIMESTAMP 列的默认值设置为 CURRENT_TIMESTAMP。插入数据时，如果没有指定 TIMESTAMP 列的值，则该值会被设置为 CURRENT_TIMESTAMP。
- ON：MySQL 不会自动将 TIMESTAMP 列的默认值设置为 CURRENT_TIMESTAMP。

  参数影响

  该参数为 MySQL 5.6.6之后的版本特性，DTS 在迁移/同步过程中，为了保证数据的一致性，建议目标端设置为 ON；如果目标端为 OFF 并且未按照提示修改参数，则选择同步含有 TIMESTAMP 列的数据后，可能会导致源和目标的表结构不一致。

  如果评估后要修改参数，请参考如下命令。

  > 说明：
  >
  > 该参数修改需要重置数据库上的所有连接才能生效，建议不要在 session 级别设置该参数，否则可能无法查询到参数状态。

```
set global explicit_defaults_for_timestamp = ON
```

版本检查

检查详情

- 检查要求：目标数据库版本必须大于或等于源数据库版本，且所有迁移和同步的版本符合版本要求。
- 检查说明：此处版本以大版本号区分，如5.6.x支持迁移或同步到5.6.x、5.7.x及以后版本，最后一位属于小版本号，小版本号不限制，如5.6.5可以迁移或同步到5.6.4，但是可能会有兼容性问题。

修复方法

请按照版本要求检查源库和目标库，如果源库或者目标库版本不支持，请升级目标实例版本或者使用更高版本的数据库实例。

源实例权限检查

检查详情

检查用户是否具备对数据库的操作权限，迁移/同步中，每个数据库类型的操作指导有详细的权限要求，请参考对应章节内容。

修复方法

用户若不具备操作权限，请按照检查要求中的对应权限要求对用户进行授权，然后重新执行校验任务。

部分实例参数检查

检查详情

- 源库表的 row_format 不能为 FIXED 。
- 源库和目标库 lower_case_table_names 变量必须一致。
- 目标库 max_allowed_packet 参数设置至少为4M。
- 源库变量 connect_timeout 必须大于10。
- 在 MySQL/TDSQL MySQL/TDSQL-C 迁移到 MySQL 的场景中，如果源实例存在耗时较长的 SQL 在运行，则会触发警告，提示“源实例有耗时较长的 SQL 在运行，可能导致锁表，请稍后重试或对源实例中的 SQL 进行处理”。

修复方法

修改源库 row_format 参数

数据库中表的 row_format 的取值为 FIXED 时，表格中每行的存储长度超过限制值时会溢出，发生报错。因此需要修改为其他模式，如 DYNAMIC ，使每行的存储长度会随内容的长度而变化。
如发生此类报错，请参考如下指导进行修复：

1. 登录源数据库。

2. 修改 row_format 参数为 DYNAMIC 。

```
alter table table_name row_format = DYNAMIC;
```

3. 查看配置是否生效。

```
show table status like 'table_name'\G;
```

系统显示结果类似如下：

```
mysql> show table status like 'table_name'\G;
*************************** 1. row ***************************
        Name: table_name
      Engine: InnoDB
     Version: 10
```

```
  Row_format: Dynamic
        Rows: 5
        ......
1 row in set (0.00 sec)
```

4. 重新执行校验任务。

修改源库和目标库 lower_case_table_names 变量保持一致

lower_case_table_names 是 MySQL 设置大小写是否敏感的一个参数，不同的取值情况如下：
Windows 或 macOS 环境对大小写是不敏感的，但是 Linux 环境却是敏感的，为了保证不同系统的兼容性，需要将大小写敏感规则设置统一。

- 0：表名存储为给定的大小写，比较的时候区分大小写。
- 1：表名存储在磁盘是小写的，比较的时候不区分大小写。
- 2：表名存储为给定的大小写，比较的时候是小写的。

如发生此类报错，请参考如下指导将源库和目标库的参数改为统一。

1. 登录源数据库。

2. 查看源库和目标库中的 lower_case_table_names 取值。

```
show variables like '%lower_case_table_names%';
```

3. 参考如下内容修改源数据库的配置文件 my.cnf 。

> 说明
>
> my.cnf 配置文件的默认路径为 /etc/my.cnf ，现场以实际情况为准。

```
lower_case_table_names = 1
```

4. 参考如下命令重启数据库。

```
[$Mysql_Dir]/bin/mysqladmin -u root -p shutdown
[$Mysql_Dir]/bin/safe_mysqld &
```

5. 查看配置是否生效。

```
show variables like '%lower_case_table_names%';
```

系统显示结果类似如下：

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

6. 重新执行校验任务。

修改目标库 max_allowed_packet 参数

max_allowed_packet 为最大允许的传输包。设置太大，会使用更多内存导致丢包，无法捕捉异常大事物包 SQL；设置太小，可能会导致程序报错，备份失败，也会导致频繁的收发网络包，影响系统性能。
如发生此类报错，请参考如下指导进行修复：

1. 登录目标数据库。

2. 修改 max_allowed_packet 参数。

```
set global max_allowed_packet = 4*1024*1024;
```

3. 查看配置是否生效。

```
show global variables like '%max_allowed_packet%';
```

系统显示结果类似如下：

```
mysql> show global variables like '%max_allowed_packet%';
+-----------------------+---------+
| Variable_name         | Value   |
+-----------------------+---------+
| max_allowed_packet    | 4194304 |
+-----------------------+---------+
1 row in set (0.00 sec)
```

4. 重新执行校验任务。

修改源库变量 connect_timeout

connect_timeout 为数据库的连接时间，超过 connect_timeout 设置值的连接请求将会被拒绝。如果设置过小，会导致数据库连接频繁断开，影响处理效率，因此建议该参数取值大于10。
如发生此类报错，请参考如下指导进行修复：

1. 登录源数据库。

2. 修改 connect_timeout 参数。

```
set global connect_timeout = 10;
```

3. 查看配置是否生效。

```
show global variables like '%connect_timeout%';
```

系统显示结果类似如下：

```
mysql> show global variables like '%connect_timeout%';
+-----------------------+---------+
| Variable_name         | Value   |
+-----------------------+---------+
| connect_timeout       | 10      |
+-----------------------+---------+
1 row in set (0.00 sec)
```

4. 重新执行校验任务。

目标实例权限检查

检查详情

1. 检查用户是否具备对数据库的操作权限，在迁移/同步中，每个数据库类型的操作指导有详细的权限要求，请参考对应章节内容。
2. 目标端为 MySQL8.0.26及之后版本，DTS 会检查 innodb_strict_mode 参数，建议在迁移/同步任务阶段将 innodb_strict_mode 参数设置为 OFF，若不为 OFF，DTS 会在前置校验给出提示。目标端为 MySQL8.0.26 之前的版本，或者其他数据库类型时，DTS 会对该参数进行 session 级别的修改，改为 OFF。
   - 参数说明：innodb_strict_mode 用于控制 InnoDB 存储引擎对一些不规范数据是否启用严格模式。
     - innodb_strict_mode=ON，InnoDB 存储引擎会执行严格的数据验证模式。如创建表（CREATE TABLE）、更改表（ALTER TABLE）和创建索引（CREATE INDEX）时，语法有错误，直接抛出错误。
     - innodb_strict_mode=OFF，InnoDB 存储引擎对数据验证放宽，遇到如上的错误语法不会报错，并且使用默认的语法替代错误的语法。
   - 设置建议：如果源端数据库本身有不规范的数据，建议在迁移/同步任务阶段将 innodb_strict_mode 参数设置为 OFF，这样可以将数据可以正常同步到目标端。

修复方法

用户若不具备操作权限，请按照检查要求中的对应权限要求对用户进行授权，然后重新执行校验任务。

目标实例内容冲突检查

检查详情

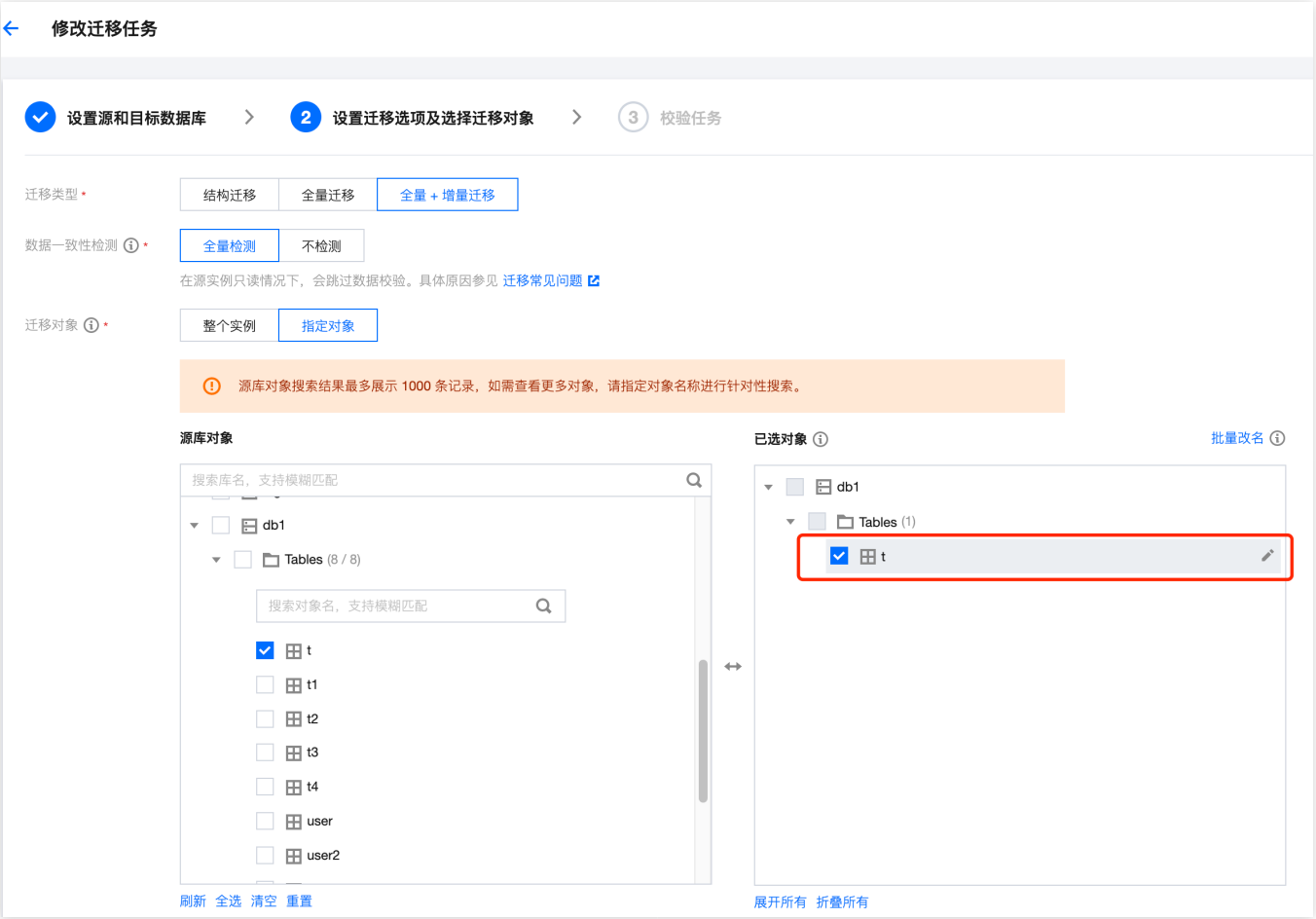
- 目标实例不能有和源库同名的对象。如果存在冲突报错，可任选以下一个方法进行修复。
  - 方法一：使用库表映射。
  - 方法二：修改或删除目标数据库中的同名对象。
  - 方法三：从迁移对象中移除同名对象。
- 整个实例迁移时，目标实例必须为空。如果存在冲突报错，需要删除实例内容。
- 选择高级对象时， 目标库不能有冲突的高级对象。如果存在冲突报错，需要删除冲突的对象。

修复方法

使用库表映射

使用 DTS 库表映射功能，将同名的待迁移对象映射为目标数据库中的其他名称。

1. 登录 DTS 控制台，选择对应的迁移任务，在操作列，选择更多 > 修改。
2. 在选择迁移对象右侧“已选对象”中，将鼠标悬浮在需要修改的对象上即可显示编辑按钮，然后重命名对象。



3. 重新执行校验任务。

修改目标数据库中的同名对象

登录目标数据库，重命名或删除目标数据库中和迁移对象同名的对象。

从迁移对象中移除同名对象

修改迁移任务配置，从迁移对象中移除同名对象，该对象不进行数据迁移。

1. 登录 DTS 控制台，选择对应的迁移任务，在操作列，选择更多 > 修改。
2. 在迁移对象中，移除同名的对象。
3. 重新执行校验任务。

删除目标库中的内容

登录目标数据库，删除目标数据库中的同名对象或者整库内容，然后重新执行校验任务。

目标实例空间检查

检查详情

目标库存储空间需要在源库待迁移库表空间的1.2倍以上。
全量数据迁移会并发执行 INSERT 操作，导致目标数据库的表产生碎片，因此全量迁移完成后目标数据库的表存储空间很可能会比源实例的表存储空间大。

修复方法

- 删除目标库中的部分数据，以便腾出足够的空间。
- 升级目标库存储规格，使用更大容量的实例进行迁移。

Binlog 参数检查

检查详情

源数据库 binlog 相关参数需要按照如下要求配置，如果校验不通过，请参考本文后续指导进行修复。

- log_bin 变量必须设置为 ON 。
- binlog_format 变量必须设置为 ROW 。
- binlog_row_image 必须设置为 FULL 。
- 如果源数据库为 MySQL 5.6 及以上版本， gtid_mode 只支持设置为 ON 和 OFF ，建议将 gtid_mode 设置为 ON ，设置为 OFF 会报警告，设置为 ON_PERMISSIVE 和 OFF_PERMISSIVE 会报错。
- server_id 参数需要手动设置，且值不能设置为0。
- 不允许设置 do_db ， ignore_db 。
- 对于源实例为从库的情况， log_slave_updates 变量必须设置为 ON 。
- 建议源库 Binlog 日志至少保留3天及以上，否则可能会因任务暂停/中断时间大于 Binlog 日志保留时间，造成任务无法续传，进而导致任务失败。

修复方法

开启 binlog

log_bin 是 binlog 的开关控制参数，需要将 binlog 打开，以便记录所有的数据库表结构和表数据变更日志。
如发生类似报错，请参考如下指导进行修复：

1. 登录源数据库。

2. 参考如下内容修改源数据库的配置文件 my.cnf 。
   因 log_bin 参数修改后需要重启数据库后才能生效，如果 binlog_format、server_id、binlog_row_image、expire_logs_days 这几个参数也在校验阶段提示报错，请一并修改完成后再重启数据库，让所有参数都生效。

   > 说明：
   >
   > my.cnf 配置文件的默认路径为 /etc/my.cnf ，现场以实际情况为准。

```
log_bin = MYSQL_BIN
binlog_format = ROW
server_id = 2     //建议设为大于1的整数，此处仅为示例
binlog_row_image = FULL
expire_logs_days = 3     //修改 binlog 的保留时间，建议大于等于3天
```

3. 参考如下命令重启源数据库。

```
[$Mysql_Dir]/bin/mysqladmin -u root -p shutdown
[$Mysql_Dir]/bin/safe_mysqld &
```

> 说明：
>
> [$Mysql_Dir] 指源数据库的安装路径，请替换为实际的源数据库安装目录。

4. 确认 binlog 功能是否已启用。

```
show variables like '%log_bin%';
```

系统显示结果类似如下：

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

请使用如下对应的命令查看其他参数的修改结果。
show variables like '%binlog_format%';
show variables like '%binlog_row_image%';
show global variables like '%server_id%';

5. 重新执行校验任务。

修改 binlog_format 参数

binlog_format 为 binlog 的记录模式，有以下三种：

- STATEMENT ：每一条会修改数据的 SQL 都会记录到 master 的 binlog 中，slave 在复制的时候，会执行和原来 master 端相同的 SQL。该模式可以减少 binlog 日志量，但是对某些特定的函数进行复制时，slave 端不能正确复制。
- ROW ：binlog 日志中会记录成每一行数据修改的形式，然后在 slave 端再对相同的数据进行修改。该模式会保证 master 和 slave 的正确复制，但是 binlog 日志量会增加。
- MIXED ：前两种模式的结合，MySQL 会根据执行的每一条具体的 SQL 语句来区分对待记录的日志形式，在 STATEMENT 和 ROW 之间选择一种。

综上，为了保证 master 和 slave 的正确复制， binlog_format 参数需要设置为 ROW 。如发生类似报错，

请参考如下指导进行修复。

> 说明：
>
> 该参数修改需要重置数据库上的所有连接才能生效，当源库为从库时，还需重启主从同步 SQL 线程，避免当前业务连接继续使用修改前的模式写入。

1. 登录源数据库。

2. 参考如下命令修改 binlog_format 。

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

3. 重启线程使配置生效，然后通过如下命令查看参数修改是否生效。

```
show variables like '%binlog_format%';
```

系统显示结果类似如下：

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

4. 重新执行校验任务。

修改 binlog_row_image 参数

binlog_row_image 参数决定了 binlog 是如何记录前镜像（记录修改前的内容）和后镜像（记录修改后的内容）的，这会直接影响到数据闪回、主从复制等功能。
binlog_row_image 参数只在 binlog_format 配置为 ROW 模式下生效。具体取值影响如下：

- FULL ：在 ROW 模式下，binlog 会记录前镜像和后镜像的所有列的数据信息。
- MINIMAL ：在 ROW 模式下，当表没有主键或唯一键时，前镜像记录所有列，后镜像记录被修改的列；如果存在主键或唯一键，不管是前镜像还是后镜像，都只记录有影响的列。

  综上， binlog_row_image 需要配置为 FULL ，源数据库的 binlog 记录全镜像。如发生报错，请参考如下步骤进行修复。

  > 说明：

> 该参数修改需要重置数据库上的所有连接才能生效，当源库为从库时，还需重启主从同步 SQL 线程，避免当前业务连接继续使用修改前的模式写入。

1. 登录源数据库。

2. 参考如下内容修改 binlog_row_image 。

```
set global binlog_row_image = FULL;
```

3. 重启线程使配置生效，然后通过如下命令查看参数修改是否生效。

```
show variables like '%binlog_row_image%';
```

系统显示结果类似如下：

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

4. 重新执行校验任务。

修改 gtid_mode 参数

GTID（Global Transaction Identifier， 全局事务标识），用于在 binlog 中唯一标识一个事务，使用 GTID 可以避免事务重复执行导致数据混乱或者主从不一致。
GTID 是 MySQL 5.6 的新特性，所以 MySQL 5.6 及之后版本存在此问题。DTS 只支持 gtid_mode 设置为 ON 和 OFF ，建议将 gtid_mode 设置为 ON ，否则校验时会报警告。
警告不影响迁移或同步任务进行，但是会对业务造成一定的影响：设置 GTID 后，在增量数据同步阶段，如果源实例发生 HA 切换，DTS 服务切换重连，任务几乎无感知；反之，任务会中断后失败，且不可恢复。
gtid_mode 的取值如下，在修改 gtid_mode 的值时，只能从以下四个值逐级修改，例如，需要从 OFF 修改为 ON ，则 gtid_mode 修改顺序为 OFF <-> OFF_PERMISSIVE <-> ON_PERMISSIVE <-> ON 。

- OFF ：主库所有新启的事务以及从库的事务都要求是匿名事务。
- OFF_PERMISSIVE ：主库新启的事务是匿名事务，但从库事务允许是匿名的或者是 GTID 事务，但不允许只是 GTID 模式。
- ON_PERMISSIVE ：主库新启的事务是 GTID 事务，从库事务允许是匿名的或者是 GTID 事务。

- ON ：主库新启的事务是 GTID 事务，从库的事务也要求是 GTID 事务。

如果发生类似警告，请按照如下指导进行修复：

1. 登录源数据库。

2. 在主从复制结构两边都设置 gtid_mode = OFF_PERMISSIVE 。
MySQL 5.7.6 之前的版本需要在 my.cnf 配置文件中修改并重启数据库才能生效，5.7.6 及之后的版本通过全局命名修改，不需要重启数据库，但是需要重置所有业务连接。

```
set global gtid_mode = OFF_PERMISSIVE;
```

3. 在主从复制结构两边都设置 gtid_mode = ON_PERMISSIVE 。

```
set global gtid_mode = ON_PERMISSIVE;
```

4. 在各个实例节点上执行如下命令，检查匿名事务是否消耗完毕，参数值为 0 则代表消耗完毕。

```
show variables like '%ONGOING_ANONYMOUS_TRANSACTION_COUNT%';
```

系统显示结果类似如下：

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

5. 在主从复制结构两边都设置 gtid_mode = ON 。

```
set global gtid_mode = ON;
```

6. 在 my.cnf 文件中添加如下内容，后续重启数据库后使初始值生效。

> 说明：
>
> my.cnf 配置文件的默认路径为 /etc/my.cnf ，现场以实际情况为准。

```
gtid_mode = on
enforce_gtid_consistency = on
```

7. 重新执行校验任务。

修改 server_id 参数

server_id 参数需要手动设置，且值不能设置为0。该参数系统预设值为 1 ，如果查询该参数显示为 1 不一定正确，需要手动进行配置。

1. 登录源数据库。

2. 参考如下内容修改 server_id 。

```
set global server_id = 2;  //建议设为大于1的整数，此处仅为示例
```

3. 通过如下命令查看参数修改是否生效。

```
show global variables like '%server_id%';
```

系统显示结果类似如下：

```
mysql> show global variables like '%server_id%';
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| server_id     | 2     |
+---------------+-------+
1 row in set (0.00 sec)
```

4. 重新执行校验任务。

删除 do_db，ignore_db 设置

binlog 会记录数据库所有执行的 DDL 和 DML 语句，而 do_db，ignore_db 则是设置 binlog 记录的过滤条件。

- binlog_do_db ：只记录指定数据库的二进制日志，默认全部记录。
- binlog_ignore_db ：不记录指定的数据库的二进制日志。

  设置 do_db，ignore_db 后，会导致一些跨库操作 binlog 记录不全，主从复制出现异常，因此不建议设置。

  如发生类似报错，请参考如下指导进行修复：

1. 登录源数据库。

2. 修改源数据库的配置文件 my.cnf ，删除 do_db，ignore_db相关设置。

> 说明：
>
> my.cnf 配置文件的默认路径为 /etc/my.cnf ，现场以实际情况为准。

3. 参考如下命令重启源数据库。

```
[$Mysql_Dir]/bin/mysqladmin -u root -p shutdown
[$Mysql_Dir]/bin/safe_mysqld &
```

> 说明：
>
> [$Mysql_Dir] 指源数据库的安装路径，请替换为实际的源数据库安装目录。

4. 确认参数修改是否生效。

```
show master status;
```

系统显示结果类似如下：

```
mysql> show master status;
+---------------+----------+--------------+------------------+-------------------+
| File          | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
+---------------+----------+--------------+------------------+-------------------+
| binlog.000011 | 154      |              |                  |                   |
+---------------+----------+--------------+------------------+-------------------+
```

5. 重新执行校验任务。

修改 log_slave_updates 参数

在主从复用结构中，从库开启 log-bin 参数，直接在从库操作数据时，可以记录在 binlog 中，但是从库从主库上复制数据时，不能记录在 binlog 中，所以从库作为其他从库的主库时，需要打开 log_slave_updates 参数。

1. 登录源数据库。

2. 在源数据库的配置文件 my.cnf 中增加如下内容。

> 说明：
>
> my.cnf 配置文件的默认路径为 /etc/my.cnf ，现场以实际情况为准。

```
log_slave_updates = ON
```

3. 参考如下命令重启源数据库。

```
[$Mysql_Dir]/bin/mysqladmin -u root -p shutdown
[$Mysql_Dir]/bin/safe_mysqld &
```

> 说明：
>
> [$Mysql_Dir] 指源数据库的安装路径，请替换为实际的源数据库安装目录。

4. 查看配置是否生效。

```
show variables like '%log_slave_updates%';
```

系统显示结果类似如下：

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

5. 重新执行校验任务。

外键依赖检查

检查详情

- MySQL/MariaDB/Percona/TDSQL MySQL/TDSQL TDStore 之间的数据迁移：DTS 默认支持 RESTRICT，NO ACTION，如果源库中存在 CASCADE、SET NULL、SET DEFAULT 这三种外键数据，也可以迁移，需要在校验项中开启外键依赖开关。
- MySQL/MariaDB/Percona/TDSQL MySQL/TDSQL TDStore 之间的数据同步：外键依赖只能设置为 NO ACTION、RESTRICT，设置为其他校验将报错，但用户可选择修复报错或者忽略报错继续任务。
- 部分库表迁移时，有外键依赖的表必须齐全。

外键依赖参数说明

MySQL 系类型的数据库在设置外键时，删除和更新有四个值可以选择。

- CASCADE ：父表进行删除或者更新记录时，子表会同步删除或更新关联记录。
- SET NULL ：父表进行删除或者更新记录时，子表会将关联记录的外键字段所在列设为 null （子表外键不能设为 not null ）。
- RESTRICT ：父表执行删除或更新记录时，如果子表中有关联该父表的记录，则拒绝该父表删除请求。
- NO ACTION ：同 RESTRICT，也是首先检查外键。
- SET DEFAULT ：父表进行删除或者更新记录时，子表将外键列设置成一个默认的值，但 InnoDB 引擎不能识别。

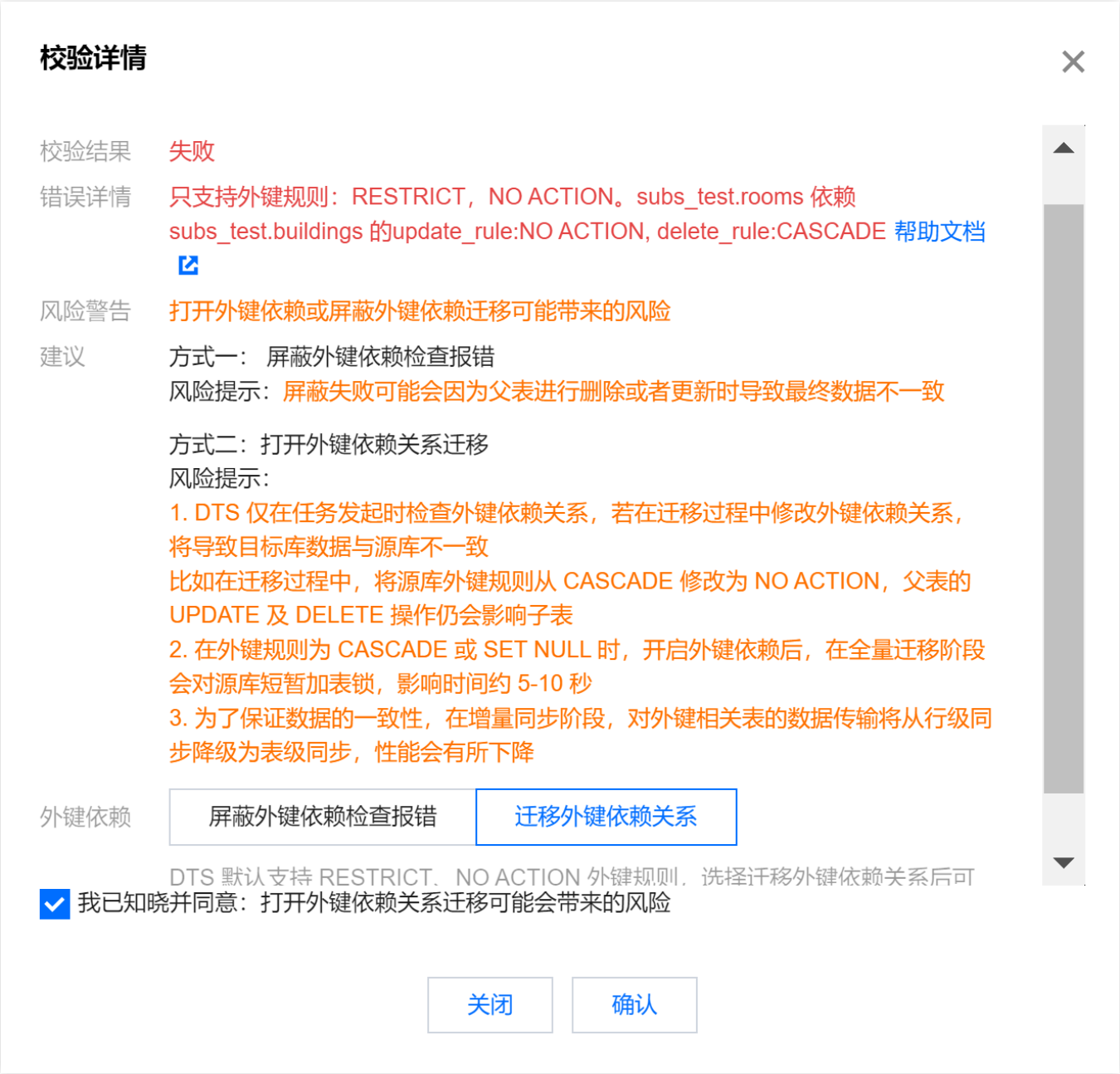
迁移场景-开启外键依赖关系迁移

DTS 默认支持 RESTRICT，NO ACTION，如果源数据库迁移的数据中存在其他类型的外键依赖配置，校验系统会提示报错，需要用户按照如下指导在校验项中开启外键依赖。

1. 在数据迁移操作中的校验任务页面，源库存在外键依赖规则 CASCADE、SET NULL、SET DEFAULT，校验项报错，查看校验项详情。

2. 根据情况选择忽略报错，或者迁移外键依赖关系。确认无误后勾线风险知会，单击确认。

选项	说明
屏蔽外键依赖检查报错	勾选后忽略已有的校验报错，继续迁移任务。 源库中存在外键规则 CASCADE、SET NULL、SET DEFAULT，仍可以进行迁移，但屏蔽报错后可能出现因为父表进行删除或者更新操作，导致目标库和源库的数据结果不一致。
迁移外键依赖关系	勾选后支持迁移 CASCADE、SET NULL、SET DEFAULT 类型的外键依赖关系，但 DTS 仅在任务发起时检查外键依赖关系，保证存量数据迁移到目标库的一致性。 1. 任务迁移过程中不要修改源库中的外键依赖规则，否则将导致目标库数据与源库不一致。如在迁移过程中，将源库外键规则从 CASCADE 修改为 NO ACTION，目标库中父表的更新及删除操作仍会影响子表。 1. 在外键规则为 CASCADE 或 SET NULL 时，勾选迁移外键依赖关系后，在全量迁移阶段会对源库短暂加表锁，影响时间约 5-10 秒。 1. 为了保证数据的一致性，在增量同步阶段，对外键相关表的数据传输将从行级同步降级为表级同步，性能会有所下降。



3. 重新执行校验任务。

同步场景-修改外键参数

数据同步中，外键依赖只能设置为 NO ACTION、RESTRICT，设置为其他校验将报错，但用户可选择修复报错或者忽略报错继续任务，如需修复报错请如下指导操作。

1. 登录源数据库。

2. 删除原来的外键设置。

```
alter table `表名称1` drop foreign key `外键名称1`;
```

3. 重新添加外键设置。

```
alter table `表名称1` add constraint `外键名称2` foreign key `表名称1`(`列名1`) references `表名称2`(`列名1`)
on update no action on delete no action;
```

4. 重新执行校验任务。

完善迁移对象

修改迁移任务配置，在迁移对象中勾选具有关联关系的对象。

1. 在 DTS 控制台，选择对应的迁移任务，在操作列选择更多 > 修改。
2. 在迁移对象中勾选具有关联关系的对象。
3. 重新执行校验任务。

视图检查

检查详情

- 检查要求：在导出视图结构时，DTS 会检查源库中 DEFINER 对应的 user1（ [DEFINER = user1]）和执行任务账号 user2 是否一致。
  - 如果一致则迁移后不做改动。
  - 如果不一致，则迁移后修改 user1 在目标库中的 SQL SECURITY 属性，由 DEFINER 转换为 INVOKER（ [INVOKER = user1]），同时设置目标库中 DEFINER 为执行任务账号 user2（[DEFINER = 执行任务账号 user2]）。
- 检查说明： SQL SECURITY 参数用来表示用户访问指定视图时，系统按照谁的权限来执行。
  - DEFINER ：表示只有定义者才能执行。
  - INVOKER ：表示拥有权限的调用者可以执行。

  默认情况下，系统指定为 DEFINER 。

高级对象检查

MySQL/MariaDB/Percona 检查详情

选择迁移/同步高级对象时，DTS 会对如下内容进行校验。报错项必须要处理才能继续任务，警告项用户评估业务风险后可忽略，继续任务。

- 报错项：目标实例参数 log_bin_trust_function_creators 必须为 ON。
- 警告项：
  - 迁移/同步高级对象与库表重命名功能冲突，选择高级对象后需要取消库表重命名。
  - 选择高级对象的函数，存储过程时，DTS 会检查源库中 DEFINER 对应的 user1（[DEFINER = user1]）和执行任务账号 user2 是否一致。
    - 如果一致则迁移/同步后不做改动。
    - 如果不一致，则迁移/同步后修改 user1 在目标库中的 SQL SECURITY 属性，由 DEFINER 转换为 INVOKER（[INVOKER = user1]），同时设置目标库中 DEFINER 为执行任务账号的 user2（[DEFINER = 执行任务账号user2]）。
  - 高级对象的迁移/同步时间：
    - 存储过程和函数，在“源库导出”阶段进行迁移/同步。
    - 触发器和事件，没有增量任务，在任务结束时进行迁移/同步；有增量任务，在用户单击完成操作后开始迁移/同步，所以单击完成后任务的过渡时间会长一些。

修复方法

修改 log_bin_trust_function_creators 参数。
log_bin_trust_function_creators 用于控制是否信任用户将存储函数写入 binlog 日志中。设置为 OFF ，仅 SUPER 权限的用户可将创建的存储函数操作写入 binlog 日志，设置为 ON ，非 SUPER 权限的用户也可将创建的存储函数操作写入 binlog 日志中。
发生报错时，请参考如下步骤进行修改。

1. 登录源数据库。

2. 参考如下内容修改 log_bin_trust_function_creators 参数。

```
set global log_bin_trust_function_creators = ON;
```

3. 通过如下命令查看参数修改是否生效。

```
show variables like '%log_bin_trust_function_creators%';
```

系统显示结果类似如下：

```
mysql>  show variables like '%log_bin_trust_function_creators%';
+--------------------------------+-------+
| Variable_name                  | Value |
+--------------------------------+-------+
| log_bin_trust_function_creators | ON    |
```

4. 重新执行校验任务。

警告项检查

检查详情

如下参数需要按照要求配置，否则校验时系统会发出警告，警告不影响迁移任务的进行，但是会对业务造成一定的影响，请用户评估后自行决定是否修改。

- 建议目标库 max_allowed_packet 的取值大于源库。
  - 业务影响：目标库的 max_allowed_packet 参数设置小于源库，会导致目标库数据无法写入，从而造成全量迁移失败。
  - 处理建议：修改目标库的 max_allowed_packet 参数，大于源库取值。
- 建议目标库的 max_allowed_packet 设置大于1GB。
  - 业务影响： max_allowed_packet 设置太大，会使用更多内存导致丢包，无法捕捉异常大事务包SQL；设置太小，可能会导致程序报错，备份失败，也会导致频繁的收发网络报，影响系统性能。
  - 处理建议：参考如下命令修改 max_allowed_packet 参数。

    ```
    set global max_allowed_packet = 1GB
    ```

- 建议源库和目标库的字符集保持一致。
  - 业务影响：源库和目标库的字符集不一致可能会导致乱码。
  - 处理建议：参考如下命令将源库和目标库字符集修改为一致。

    ```
    set character_set_server = 'utf8';
    ```

- 建议使用2CPU，4G Mem以上规格的实例。
- 如果仅执行全量数据迁移，请勿在迁移过程中向源实例中写入新的数据，否则会导致源和目标数据不一致。针对有数据写入的场景，为实时保持数据一致性，建议选择全量+增量数据迁移。
- 有锁导出时：源实例需要使用 Flush Table With Read Lock 短暂加锁，其中的 MyISAM 表会锁定到全量数据导出完成。当前等待加锁超时时间设置为60秒, 该时间内无法获取锁将导致任务失败。
- 无锁导出时：仅对没有主键的表会加读锁，其他不加锁。
- 对于既没有主键、也没有非空唯一键的表，有数据重复的风险，建议用户对迁移的表设置主键，或者非空唯一键。
- 对于 TDSQL MySQL 的数据迁移，当源数据库实例为分布式数据库时，需要提前在目标库建立分表，否则这些表被迁移后都将是单表。
- 目标库为 MySQL/MariaDB/Percona/TDSQL TDStore 时，需要检查源库和目标库

explicit_defaults_for_timestamp 的设置。源库为 OFF ，或者源和目标都为 ON 时，任务警告，提醒用户任务运行期间不要随便修改该参数。

- 在全量导出库表结构时检查表的 COLUMN_DEFAULT ， IS_NULLABLE 属性，如果源库表的 COLUMN_DEFAULT 为 NULL ， IS_NULLABLE 为 NOT NULL ，则该表结构不进行迁移/同步，避免数据同步到目标库后，MySQL 系统可能会自动为 TIMESTAMP 的数据类型添加默认参数“CURRENT_TIMESTAMP”。

TDSQL Proxy 检查

检查详情

目标库 TDSQL MySQL 使用 Proxy 方式连接的场景中，DTS 在目标库执行 SQL 的时候，是将多条语句放到一个 SQL 里去执行，需要将 gateway.mode.multi_query.open 参数设置为1，以支持多个 SQL 合并执行。参数如未打开，预校验任务会报错。

修复方法

请按照要求修改参数 进行修改。

TDSQL 内核检查

检查详情

源端为 TDSQL MySQL 时，检查源端的内核版本，不支持 mariadb 内核。如果是 mariadb 内核，预校验任务会报错。

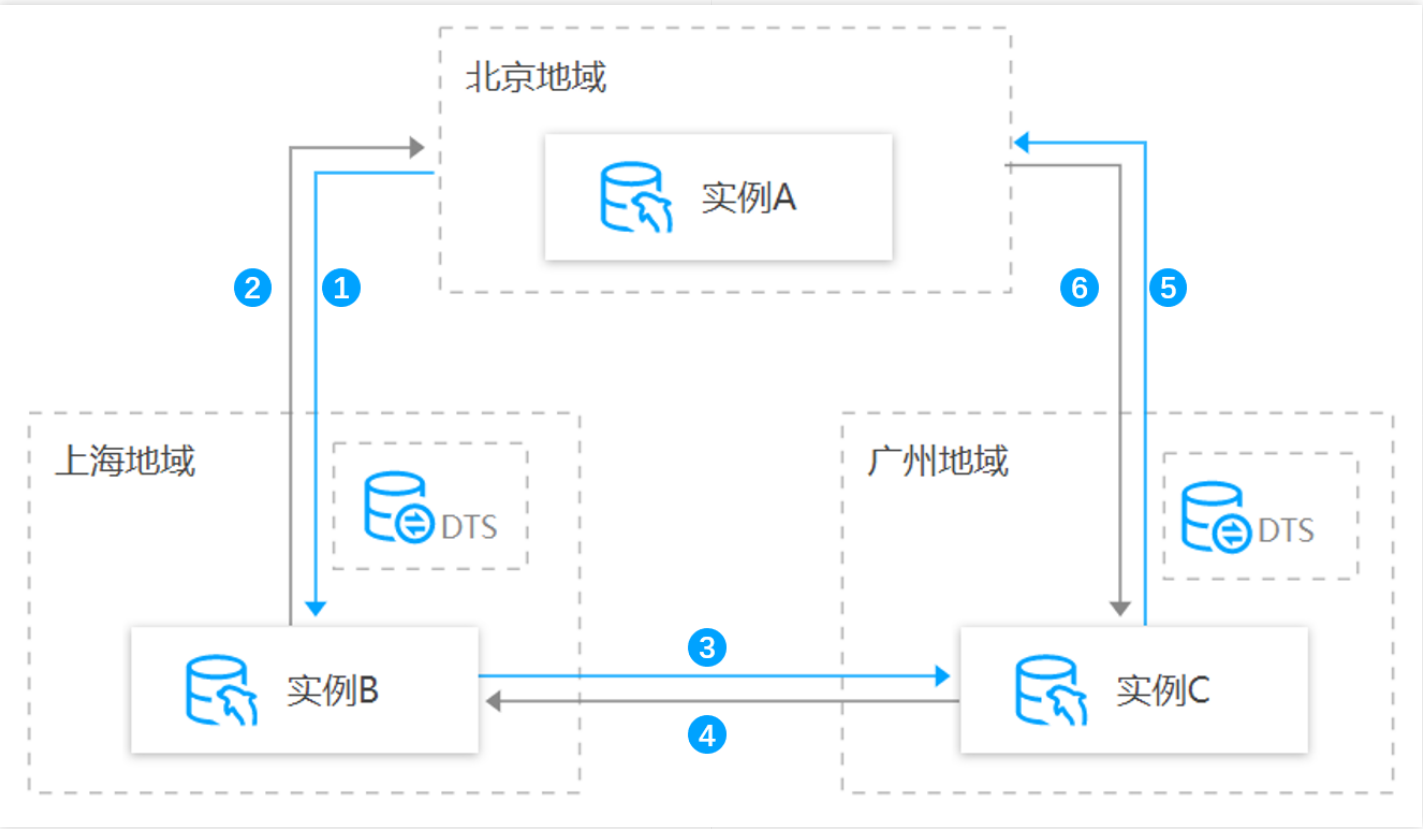
修复方法

无法同步 TDSQL MySQL（mariadb 内核），请选择其他版本进行同步。

同库表对象 DDL 环形链路同步检测

检查详情

在双向同步、多对一同步、多活等需要配置多个同步任务的场景中，DDL 的配置不能形成环形链路，否则可能造成 DDL 语句在系统中循环，进而引发错误。
示例：下图中蓝色线条1、3、5三个同步任务中，最多只能在两个同步任务中选择 DDL，如果选择三个就构成环形链路了。



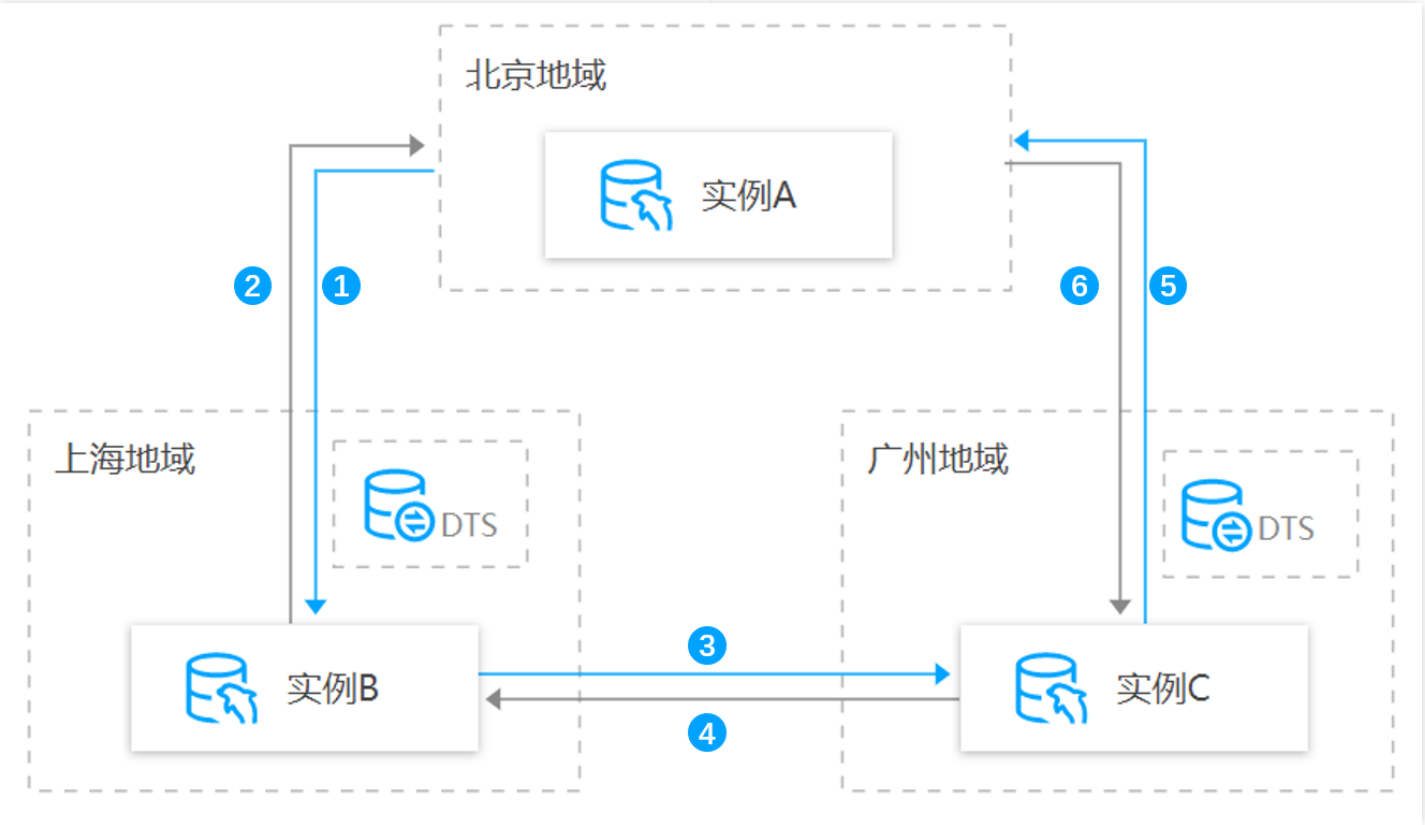
修复方法

修改同步任务配置，在设置同步选项和同步对象 > 数据同步选项 > 同步操作类型中，修改 DDL 参数配置，避免形成环形链路。

同目标同库表对象 DDL 同步冲突检测

检查详情

在多对一同步、多活等需要配置多个同步任务的场景中，同一个库表对象不能接收多个数据中心的 DDL 同步，否则多个数据中心的 DDL 可能在目标端形成冲突，进而引发错误。
示例：下图中实例 A 和 C 有相同名称的表需要同步到实例 B 中，则同步任务1和4中，只能在一个任务选择 DDL。



修复方法

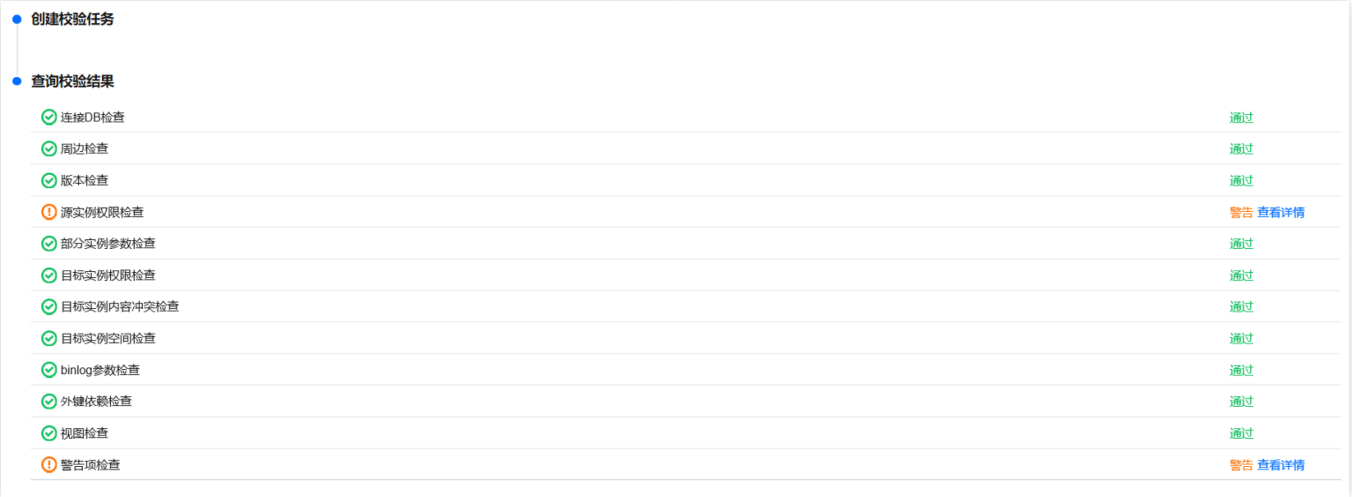
修改同步任务配置，在设置同步选项和同步对象 > 数据同步选项 > 同步操作类型中，修改 DDL 参数配置，避免同一个库表对象接收多个数据中心的 DDL 同步。

前置校验项不通过处理

问题现象

用户在任务校验阶段，出现校验项结果不通过或者出现警告。

- 结果为失败：表示校验项检查未通过，任务阻断，需要修复问题后重新执行校验任务。
- 结果为警告：表示检验项检查不完全符合要求，可以继续任务，但对业务有一定的影响，用户需要根据提示自行评估是忽略警告项还是修复问题再继续。



处理方法

检查项不符合要求，请参考如下链接中的对应指导进行处理。

数据迁移

1. MySQL/MariaDB/Percona/TDSQL MySQL：MySQL 系列前置校验不通过处理
2. MongoDB：MongoDB 前置校验不通过处理
3. Redis®：Redis 前置校验项不通过处理

数据同步

MySQL/MariaDB/Percona/TDSQL MySQL：MySQL 系列前置校验不通过处理

割接说明

注意事项

- 为便于区分会话信息以及提升数据安全性，建议单独创建一个数据库账号以供数据迁移使用。
- 由于割接需要暂停源库的数据写入，建议您选择一个业务低峰期进行业务割接。

操作步骤

1. 登录 DTS 控制台，根据有无增量迁移，选择如下步骤：

- 有增量迁移：请执行步骤 2。
- 无增量迁移：请执行步骤 6。

2. 等待数据迁移任务的迁移步骤显示为同步增量，并且目标与源库数据差距为0KB，目标与源库时间延迟为0秒。

3. 暂停源库业务，停止新的数据写入。
4. 根据源数据库的不同类型，选择如下对应代码查看是否有新的会话信息。如果1分钟 - 5分钟内显示结果除 DTS 迁移实例的连接外，无任何新的会话执行，即可认为业务已经完全停止。

- MySQL

```
show processlist
```

- MongoDB

```
use admin
db.runCommand({currentOp: 1, $all:[{"active" : true}]})
```

5. 结束增量迁移任务。
   再次查看迁移任务，等待目标与源库数据差距为0KB，目标与源库时间延迟为0秒，并保持1分钟以上，单击完成，结束增量迁移任务。

6. 验证源库和目标库的数据一致后，确定割接时机，将业务系统指向目标数据库，恢复业务使用。

监控与告警
支持的监控指标

指标说明

- 统计维度：迁移 app_id、migratejob_id；同步 appid、replicationjobid，订阅 appId、subscribeid。
- 统计粒度：60s、300s。
- BPS：表示每秒传输的数据量。
- RPS：表示每秒传输的行数。

数据迁移

MySQL/MariaDB/Percona/TDSQL MySQL

指标分类	指标中文名称	指标英文名称	单位	说明
BPS-全量阶段	源实例全量导出 BPS	MigrateDumperBps	MB/s	全量阶段，DTS 每秒从源实例导出的数据量。
	目标实例全量导入 BPS	MigrateLoaderBps	MB/s	全量阶段，DTS 每秒导入到目标实例的数据量。
BPS-增量阶段	源实例增量导出 BPS	MigrateRiverBps	MB/s	增量阶段，DTS 每秒从源实例导出的数据量。
	目标实例增量导入 BPS	MigrateSinkerBps	MB/s	增量阶段，DTS 每秒导入到目标实例的数据量。
RPS-全量阶段	源实例全量导出 RPS	MigrateDumperRps	Count/s	全量阶段，DTS 每秒从源实例导出的数据行数。
	目标实例全量导入 RPS	MigrateLoaderRps	Count/s	全量阶段，DTS 每秒导入到目标实例的数据行数。
RPS-增量阶段	源实例数据抽取 RPS（同源实例增量导出 RPS）	MigrateCaptureRps	Count/s	该指标为过渡阶段，后续仅保留源实例增量导出 RPS。

指标分类	指标中文名称	指标英文名称	单位	说明
	源实例增量导出 RPS	MigrateRiverRps	Count/s	增量阶段，DTS 每秒从源实例导出的数据行数。
	目标数据装载 RPS（同目标实例增量导入 RPS）	MigrateLoadRps	Count/s	该指标为过渡阶段，后续仅保留目标实例增量导入 RPS。
	目标实例增量导入 RPS	MigrateSinkerRps	Count/s	增量阶段，DTS 每秒导入到目标实例的数据行数。
网络延迟-全量阶段	DTS 全量导出时与源实例网络延迟	MigrateDumperNetworkLag	ms	全量阶段，DTS 数据导出与源实例的网络延迟。
	DTS 全量导入时与目标实例网络延迟	MigrateLoaderNetworkLag	ms	全量阶段，DTS 数据导入与目标实例的网络延迟。
网络延迟-增量阶段	DTS 增量导出时与源实例网络延迟	MigrateRiverNetworkLag	ms	增量阶段，DTS 数据导出与源实例的网络延迟。
	DTS 增量导入时与目标实例网络延迟	MigrateSinkerNetworkLag	ms	增量阶段，DTS 数据导入与目标实例的网络延迟。
	数据迁移延迟时间	MigrateLag	s	增量阶段，目标实例和源实例的时间延迟。 计算方法：源实例当前时间减去目标实例当前正在执行的最新一条源实例 Binlog Event 中记录的时间。 源库为只读的场景，当源库长时间没有数据写入时，该指标会出现延迟。 “数据迁移延迟时间”指标的计算依赖于源库的增量 Binlog，当源库长时间没有 DDL 或者 DML 操作时，该指标会逐步增大，无法反应真实的同步延迟时间（例如出现值为“-1”，表示存量数据迁移完成，一直没有增量数据的刷新）。DTS 为了解决这个问题，在源库中插入心跳让增量数据持续更新，但源库为只读的场景无

指标分类	指标中文名称	指标英文名称	单位	说明
				法写入心跳，所以仍会存在延迟的情况。
	数据迁移延迟数据量	MigrateLagData	MBytes	增量阶段，目标实例和源实例的数据差距。 计算方法：源实例最新的 Binlog Event 文件位点减去目标实例当前正在执行的最新一条源实例 Binlog Event 的文件位点。 当这两个位点跨越 Binlog 文件时，该值有估算成分。 源库为只读的场景，当源库长时间没有数据写入时，该指标会出现延迟（例如出现值为“-1”，表示存量数据同步完成，一直没有增量数据的刷新）。
RPS 使用率	目标实例增量导入 RPS 使用率	MigrateSinkerRpsUsage	%	RPS 使用率 = 实时 RPS/ RPS 上限，RPS 上限为链路配置中选择的规格，不同的规格上限不同。 • 当 RPS 使用率接近 100%，则增量速率无法再提升，需要升级更高的规格。 • 当链路配置为最大规格时，因为最大规格中 RPS 无上限限制，所以 RPS 使用率可能超过100%。

Redis®

> 注意：
>
> 2025年4月14日后启动的任务支持查看监控数据。对于2025年4月14日之前创建的任务，如果在2025年4月14日后启动，也会有监控数据。

指标分类	指标中文名称	指标英文名称	单位	说明
BPS-全量阶段	源实例全量导出 BPS	MigrateDumperBps	MB/s	全量阶段，DTS 每秒从源实例导出的数据量。

指标分类	指标中文名称	指标英文名称	单位	说明
	目标实例全量导入BPS	MigrateLoaderBps	MB/s	全量阶段，DTS 每秒导入到目标实例的数据量。
BPS-增量阶段	源实例增量导出BPS	MigrateRiverBps	MB/s	增量阶段，DTS 每秒从源实例导出的数据量。
	目标实例增量导入BPS	MigrateSinkerBps	MB/s	增量阶段，DTS 每秒导入到目标实例的数据量。
RPS-全量阶段	目标实例全量导入RPS（每秒命令数）	MigrateLoaderRps	Count/s	在全量阶段，每秒导入数据到目标库的命令数
RPS-增量阶段	源实例增量导出RPS（每秒命令数）	MigrateRiverRps	Count/s	在增量阶段从源库导出的数据命令数。 初始值为“0”，在全量文件接收完毕后刷新。
	目标实例增量导入RPS（每秒命令数）	MigrateSinkerRps	Count/s	在增量阶段导入进目标库的数据命令数。 初始值为“0”，在任意一个分片进入增量阶段后刷新。
网络延迟-全量阶段	DTS 全量导出时与源实例网络延迟	MigrateDumperNetworkLag	ms	全量阶段，DTS 数据导出与源实例的网络延迟。
	DTS 全量导入时与目标实例网络延迟	MigrateLoaderNetworkLag	ms	全量阶段，DTS 数据导入与目标实例的网络延迟。
网络延迟-增量阶段	DTS 增量导出时与源实例网络延迟	MigrateRiverNetworkLag	ms	增量阶段，DTS 数据导出与源实例的网络延迟。
	DTS 增量导入时与目标实例网络延迟	MigrateSinkerNetworkLag	ms	增量阶段，DTS 数据导入与目标实例的网络延迟。
	数据迁移延迟时间	MigrateLag	s	目标实例和源实例的时间延迟。 计算方法：增量延迟数据量/当前目标实例增量导入BPS 在任意一个分片进入增量阶段后刷新。

指标分类	指标中文名称	指标英文名称	单位	说明
	数据迁移延迟数据量	MigrateLagData	MBytes	目标实例和源实例的数据差距。 计算方法：源实例的 repl_offset 同 DTS 已导入完成的数据量差值。 在任意一个分片进入增量阶段后刷新。

数据同步

MySQL/MariaDB/Percona/TDSQL MySQL

指标分类	指标中文名	指标英文名	单位	说明
BPS-全量阶段	源实例全量导出 BPS	ReplicationDumperBps	MB/s	全量阶段，DTS 每秒从源实例导出的数据量。
	目标实例全量导入 BPS	ReplicationLoaderBps	MB/s	全量阶段，DTS 每秒导入到目标实例的数据量。
BPS-增量阶段	源实例增量导出 BPS	ReplicationRiverBps	MB/s	增量阶段，DTS 每秒从源实例导出的数据量。
	目标实例增量导入 BPS	ReplicationSinkerBps	MB/s	增量阶段，DTS 每秒导入到目标实例的数据量。
RPS-全量阶段	源实例全量导出 RPS	ReplicationDumperRps	Count/s	全量阶段，DTS 每秒从源实例导出的数据行数。
	目标实例全量导入 RPS	ReplicationLoaderRps	Count/s	全量阶段，DTS 每秒导入到目标实例的数据行数。
RPS-增量阶段	源实例数据抽取 RPS（同源实例增量导出 RPS）	ReplicationCaptureRps	Count/s	该指标为过渡阶段，后续仅保留源实例增量导出 RPS。
	源实例增量导出 RPS	ReplicationRiverRps	Count/s	增量阶段，DTS 每秒从源实例导出的数据行数。
	目标数据装载 RPS（同	ReplicationLoadRps	Count/s	该指标为过渡阶段，后续仅保留目标实例增量导入 RPS。

指标分类	指标中文名	指标英文名	单位	说明
	目标实例增量导入RPS）			
	目标实例增量导入 RPS	ReplicationSinkerRps	Count/s	增量阶段，DTS 每秒导入到目标实例的数据行数。
网络延迟-全量阶段	DTS 全量导出时与源实例网络延迟	ReplicationDumperNetworkLag	ms	全量阶段，DTS 数据导出与源实例的网络延迟。
	DTS 全量导入时与目标实例网络延迟	ReplicationLoaderNetworkLag	ms	全量阶段，DTS 数据导入与目标实例的网络延迟。
网络延迟-增量阶段	DTS 增量导出时与源实例网络延迟	ReplicationRiverNetworkLag	ms	增量阶段，DTS 数据导出与源实例的网络延迟。
	DTS 增量导入时与目标实例网络延迟	ReplicationSinkerNetworkLag	ms	增量阶段，DTS 数据导入与目标实例的网络延迟。
	数据同步延迟时间	DtsReplicationLag	s	增量阶段，目标实例和源实例的时间同步延迟。 计算方法：源实例当前时间减去目标实例当前正在执行的最新一条源实例 Binlog Event 中记录的时间。 源库为只读的场景，当源库长时间没有数据写入时，该指标会出现延迟。“数据同步延迟时间”指标的计算依赖于源库的增量 Binlog，当源库长时间没有 DDL 或者 DML 操作时，该指标会逐步增大，无法反应真实的同步延迟时间（例如出现值为“-1”，表示存量数据同步完成，一直没有增量数据的刷新）。DTS 为了解决这个问题，在源库中插入心跳让增量数据持续更新，但源库为只读的场景无法写入心跳，所以仍会存在延迟的情况。

指标分类	指标中文名	指标英文名	单位	说明
	数据同步延迟数据量	 DtsReplicationLagData 	MBytes	增量阶段，目标实例和源实例的数据同步差距。 计算方法：源实例最新的 Binlog Event 文件位点减去目标实例当前正在执行的最新一条源实例 Binlog Event 的文件位点。 当这两个位点跨越 Binlog 文件时，该值有估算成分。 源库为只读的场景，当源库长时间没有数据写入时，该指标会出现延迟（例如出现值为“-1”，表示存量数据同步完成，一直没有增量数据的刷新）。
RPS 使用率	目标实例增量导入 RPS 使用率	ReplicationSinkerRpsUsage	%	RPS 使用率 = 实时 RPS/RPS 上限，RPS 上限为链路配置中选择的规格，不同的规格上限不同。 • 当 RPS 使用率接近 100%，则增量速率无法再提升，需要升级更高的规格。 • 当链路配置为最大规格时，因为最大规格中 RPS 无上限限制，所以 RPS 使用率可能超过 100%。

支持的事件

概述

DTS 支持对数据迁移、数据同步任务过程中的事件进行监控并设置报警规则，在事件触发或者指标达到设定阈值时，及时通知用户采取措施。

> 说明：
>
> 当前支持事件告警的场景为 MySQL/MariaDB/Percona/TDSQL TDStore 的迁移和同步链路。

支持的事件告警

事件名称	说明
数据迁移任务中断	数据迁移任务过程中，当任务发生异常中断时（不含用户主动中断任务的场景），触发告警。
数据同步任务中断	数据同步任务过程中，当任务发生异常中断时（不含用户主动中断任务的场景），触发告警。
DTS 服务维护即将启动	DTS 服务即将进行维护升级，在用户设置的维护时间前24小时，触发告警，通知用户 DTS 将会进行服务维护升级。
DTS 任务中断时间过长	DTS 任务中断时间大于1天，小于14天，触发告警。
DTS 任务状态扭转为失败	DTS 任务中断时间达到14天，触发告警，通知用户任务无法救起，任务状态将扭转为结束。
云 API 操作事件（基于操作审计投递）	对云 API 操作过程中的异常中断触发告警。
控制台操作事件（基于操作审计投递）	对控制台操作过程中的异常中断触发告警。
小程序操作事件（基于操作审计投递）	对小程序操作过程中的异常中断触发告警。

配置指标告警和事件告警
配置数据迁移指标告警

操作场景

用户可以通过云监控对数据迁移任务中重要的指标设置告警规则，在指标发生异常时，云监控平台及时通知用户采取措施。
本操作用于指导用户设置指标告警的通知规则，包括指标告警触发的条件、指标告警的范围、通知的形式、通知的时段、通知的用户群组等。

新增告警策略

1. 登录云监控。

2. 在左侧导航选择告警管理 > 策略管理，进入告警策略配置页面。

3. 单击新建策略，配置告警策略，配置说明如下：

配置类型	配置项	说明
基本信息	策略名称	自定义策略名称。
	备注	自定义策略备注。
	监控类型	云产品监控。
	策略类型	选择您需要监控的云产品策略类型，此处选择数据传输服务/数据迁移。 ○ 数据迁移：监控数据迁移场景的指标。 ○ 数据同步：监控数据同步场景的指标。
配置告警规则	告警对象	○ 选择实例 ID，则该告警策略绑定用户选中的实例。 ○ 选择实例组，则该告警策略绑定用户选中的实例分组。 ○ 选择全部对象，则该告警策略绑定当前账号拥有权限的全部实例。
	手动配置-指标告警	○ 告警触发条件：可选择任意或者所有指标达到设置条件时触发警告。 ○ 配置示例：指标为“源实例数据抽取 RPS”、统计粒度1分钟、比较关系为“小于”、阈值为“1” 、持续监控数据为“持续3个数据点”。 ○ 配置效果：每1分钟收集一次“源实例数据抽取 RPS”，若 DTS 每秒读取源实例数据的行数连续3次小于1，则触发告警。

配置类型	配置项	说明
		○ 告警频率：您可以定义告警以特定的频率重复通知。如每1小时告警一次、每2小时告警一次、每1天告警一次……等重复频率。
	手动配置-事件告警	选择需要上报告警的事件。 此处配置与事件总线的配置效果一致，如已参考“配置事件告警”进行了相关配置，则此处不需要重复配置。
	选择模板	选择模板按钮，并在下拉列表选择已配置的模板 。若新建的模板没有显示，则单击右侧的刷新，即可刷新触发告警模板选择列表。
配置告警通知	通知模板	支持选择系统预设通知模板和用户自定义通知模板，每个告警策略最多只能绑定三个通知模板。

4. 配置完以上信息后单击保存，即成功创建告警策略。

修改告警策略

1. 登录云监控。
2. 在左侧导航选择告警管理 > 策略管理，然后单击需要修改的策略名称 ID，进入管理告警策略页面。
3. 修改触发条件、告警对象、告警通知等信息。

配置数据同步指标告警

操作场景

用户可以通过云监控对数据同步任务中重要的指标设置告警规则，在指标发生异常时，云监控平台及时通知用户采取措施。
本操作用于指导用户设置指标告警的通知规则，包括指标告警触发的条件、指标告警的范围、通知的形式、通知的时段、通知的用户群组等。

新增告警策略

1. 登录云监控。

2. 在左侧导航选择告警管理 > 策略管理，进入告警策略配置页面。

3. 单击新建策略，配置告警策略，配置说明如下：

配置类型	配置项	说明
基本信息	策略名称	自定义策略名称。
	备注	自定义策略备注。
	监控类型	云产品监控。
	策略类型	选择您需要监控的云产品策略类型，此处选择数据传输服务/数据同步。 ○ 数据迁移：监控数据迁移场景的指标。 ○ 数据同步：监控数据同步场景的指标。
配置告警规则	告警对象	○ 选择实例 ID，则该告警策略绑定用户选中的实例。 ○ 选择实例组，则该告警策略绑定用户选中的实例分组。 ○ 选择全部对象，则该告警策略绑定当前账号拥有权限的全部实例。
	手动配置-指标告警	○ 告警触发条件：可选择任意或者所有指标达到设置条件时触发警告。 ○ 配置示例：指标为“源实例数据抽取 RPS”、统计粒度1分钟、比较关系为“小于”、阈值为“1” 、持续监控数据为“持续3个数据点”。 ○ 配置效果：每1分钟收集一次“源实例数据抽取 RPS”，若 DTS 每秒读取源实例数据的行数连续3次小于1，则触发告警。 ○ 告警频率：您可以定义告警以特定的频率重复通知。如每1小时告警一次、每2小时告警一次、每1天告警一次……等重复频率。

配置类型	配置项	说明
	手动配置-事件告警	选择需要上报告警的事件。 此处配置与事件总线的配置效果一致，如已参考“配置事件告警”进行了相关配置，则此处不需要重复配置。
	选择模板	选择模板按钮，并在下拉列表选择已配置的模板。若新建的模板没有显示，则单击右侧的刷新，即可刷新触发告警模板选择列表。
配置告警通知	通知模板	支持选择系统预设通知模板和用户自定义通知模板，每个告警策略最多只能绑定三个通知模板。

4. 配置完以上信息后单击保存，即成功创建告警策略。

修改告警策略

1. 登录云架空。
2. 在左侧导航选择告警管理 > 策略管理，然后单击需要修改的策略名称 ID，进入管理告警策略页面。
3. 修改触发条件、告警对象、告警通知等信息。

配置事件告警推送

操作场景

开通云事件总线服务后，事件总线会为您自动在广州地域创建云服务默认事件集，DTS 发生的所有事件（任务异常中断、任务升级、任务状态扭转等）都会自动投递到默认事件集。
您需要通过本操作设置告警推送规则，即从总的事件中，筛选需要收到的告警项，并设置告警通知形式、接收用户群组等。

前提条件

已开通事件总线。

操作步骤

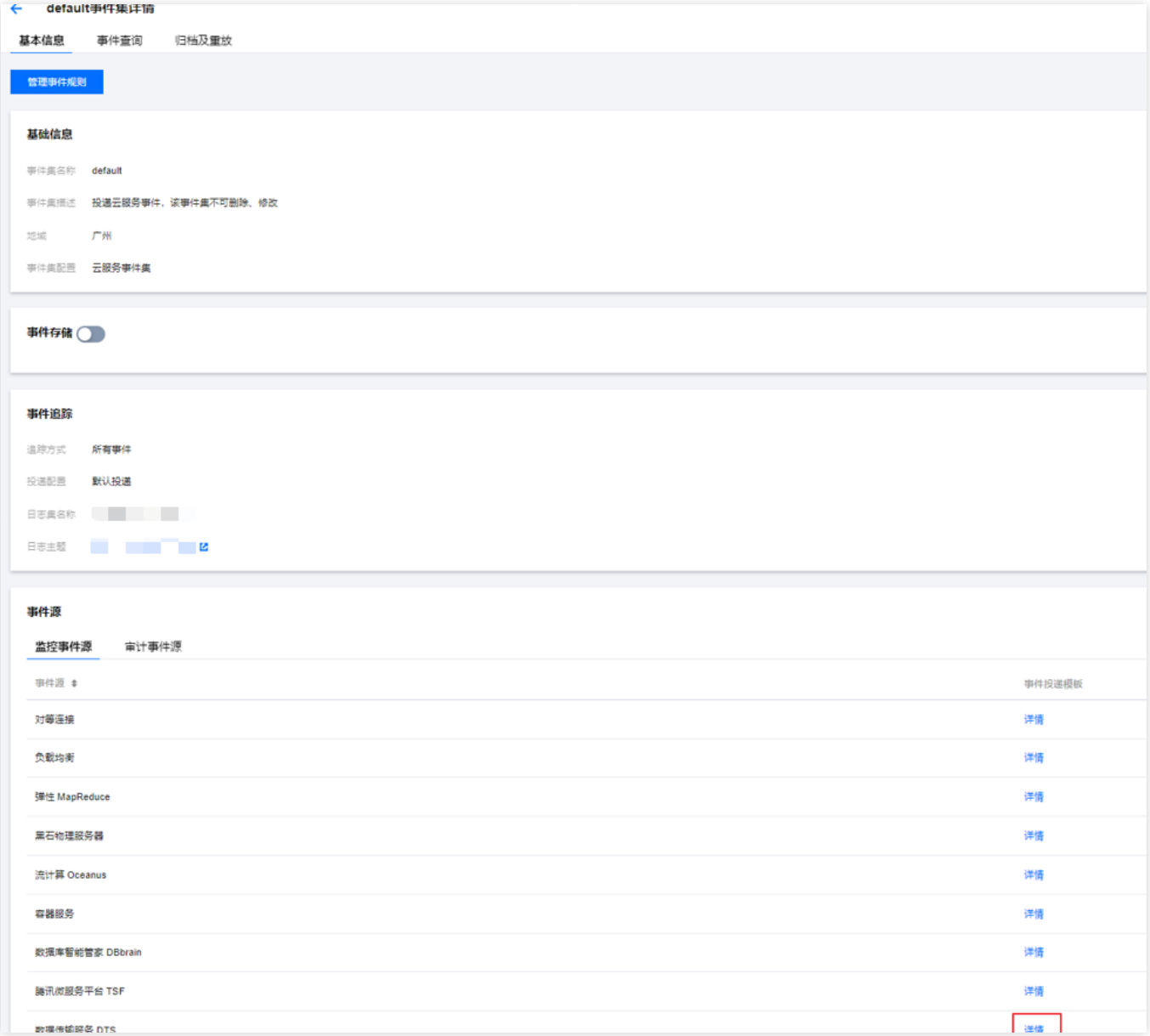
步骤一：查看事件列表

1. 登录事件总线控制台。

2. 单击左侧导航栏事件集，default 事件集统一存储在广州地域，无需修改**。**

> 说明：
>
> 首次登录系统会提醒用户进行授权，如果已授权请跳过此步骤。

3. 单击事件集 ID，在基本信息 > 监控事件源中查看 DTS 监控事件。



步骤二：配置事件告警规则

1. 登录事件总线控制台，单击左侧导航栏的事件规则。

2. 在事件规则页面，选择地域和事件集，单击新建。default 事件集统一存储在广州地域，无需修改。

3. 在基础信息页面中，输入规则名称、规则描述后，在基础信息页面下方的事件匹配模块，用户可按需选择如下任一方式配置：

i. 选择表单模式，配置云服务类型和事件类型，勾选对应的事件类型即可，事件类型可选择全部事件告警，也可选择指定的事件告警。

事件名称	说明
数据迁移任务中断	数据迁移任务过程中，当任务发生异常中断时（不含用户主动中断任务的场景），触发告警。
数据同步任务中断	数据同步任务过程中，当任务发生异常中断时（不含用户主动中断任务的场景），触发告警。
DTS 服务维护即将启动	DTS 服务即将进行维护升级，在用户设置的维护时间前24小时，触发告警，通知用户 DTS 将会进行服务维护升级。
DTS 任务中断时间过长	DTS 任务中断时间大于1天，小于14天，触发告警。

事件名称	说明
DTS 任务状态扭转为失败	DTS 任务中断时间达到14天，触发告警，通知用户任务无法救起，任务状态将扭转为结束。
云 API 操作事件（基于操作审计投递）	对云 API 操作过程中的异常中断触发告警。
控制台操作事件（基于操作审计投递）	对控制台操作过程中的异常中断触发告警。
小程序操作事件（基于操作审计投递）	对小程序操作过程中的异常中断触发告警。

ii. 选择自定义事件，根据需要，灵活设置告警规则，如仅接收指定任务 ID 的告警，不同场景的语法规则请参见本章节后续的“自定义事件语法示例”。

4. 事件模式页面配置完成后，单击下一步。

5. 在新建事件规则 > 事件目标页面，配置如下参数后，单击完成。

参数	说明
触发方式	此处选择消息推送。
消息模板	默认选择通用通知模板。

参数	说明
通知方式	通知方式可选如下两种，也可以选择全部方式。 - 接口回调：通过接口回调，接口回调具备将告警信息通过 GET 请求推送到可访问公网 URL 的功能。 - 渠道推送：需选择接收对象、通知时段、接收渠道。 可配置云账号下其他子账号为接收对象，如需新增接收用户/用户组，请先在访问管理中进行配置，然后在本步骤中才可选择接收对象。
添加	如果需要配置不同的触发方式，可以单击最下方的添加增加事件目标。
立即启用事件规则	勾选后，在单击完成后，立即启用事件规则。

6. 返回事件规则列表，确认创建的事件规则已启动 。后续当任务异常触发告警时，用户即可接收到消息通知。

自定义事件告警语法示例

不同事件告警的规则语法示例如下：

- 接收全部 DTS 事件告警。如下语法表示所有来自 DTS 的告警事件均可以通过规则匹配进行告警推送。

```
{
  "source":"dts.cloud.tencent"
}
```

- 接收 DTS 数据迁移、数据同步事件告警。如下语法 type 中的两个字段，分别表示数据迁移任务中断、数据同步任务中断，如果用户不需要接收哪种类型的告警，删除对应内容即可。

```
{
  "source":"dts.cloud.tencent",
  "type":[
    "dts:ErrorEvent:MigratejobInterruption",
    "dts:ErrorEvent:ReplicationInterruption"
  ]
}
```

- 接收单个 DTS 任务事件告警。如下语法表示 DTS 事件中仅 ID 为 sync-jt12XXgt 的任务产生的事件才可以通过规则

匹配进行告警推送，其它事件将被丢弃，无法触达用户。

```
{
  "source":"dts.cloud.tencent",
  "subject":"sync-jt12XXgt"
}
```

- 接收多个 DTS 任务事件告警。

```
{
  "source":"dts.cloud.tencent",
  "subject":["sync-jt12XXgt","dts-a5uqXXhs"]
}
```

- 接收 DTS 数据迁移指定地域的事件告警。这里的地域为目标端实例的所属地域。

```
{
  "source":"dts.cloud.tencent",
  "type":"dts:ErrorEvent:MigratejobInterruption",
  "region":"ap-guangzhou"
}
```

查看告警历史

操作场景

用户可以查看历史指标告警和事件告警，以便了解系统的各项指标性能。

操作步骤

1. 登录云监控。
2. 单击告警管理 > 告警历史，即可查看告警历史，可以选择高级筛选，输入关键字后查看对应告警内容。

查看监控指标

操作场景

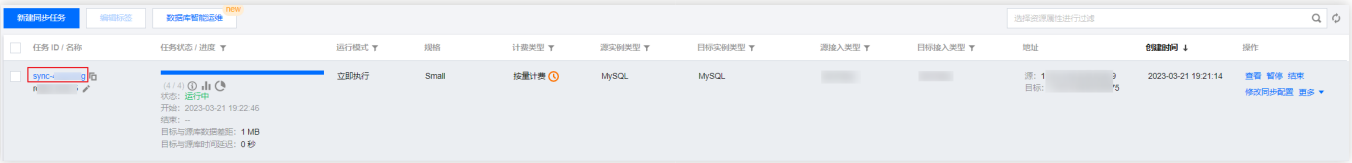
用户可以实时查看指标监控，以便了解任务运行中的各项指标性能。

操作步骤

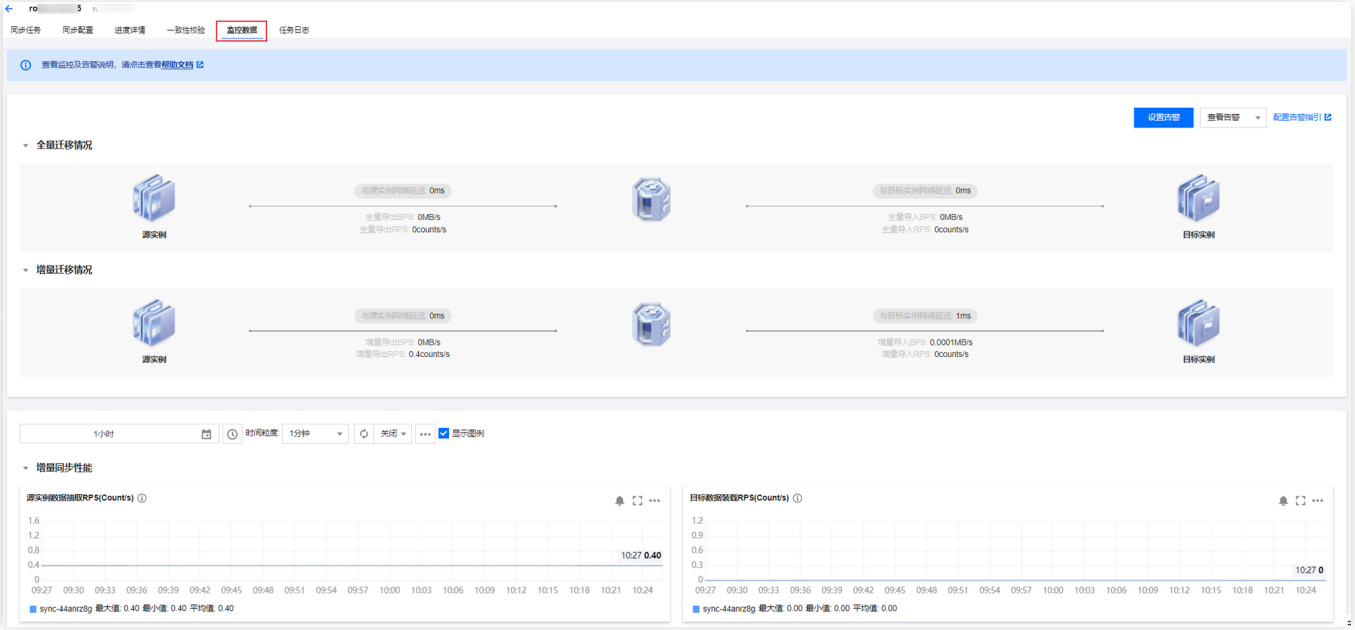
1. 登录 DTS 控制台，在左侧选择任务场景：数据迁移或数据同步。
2. 您可通过如下两种方式查看监控指标。

- 方式一：选择指定的迁移任务，单击任务状态下的查看视图按钮 ，可查看监控视图。单击查看任务监控按钮 ，可查看详细监控指标信息。

- 方式二：选择指定的迁移任务，单击任务 ID，进入任务详情页。

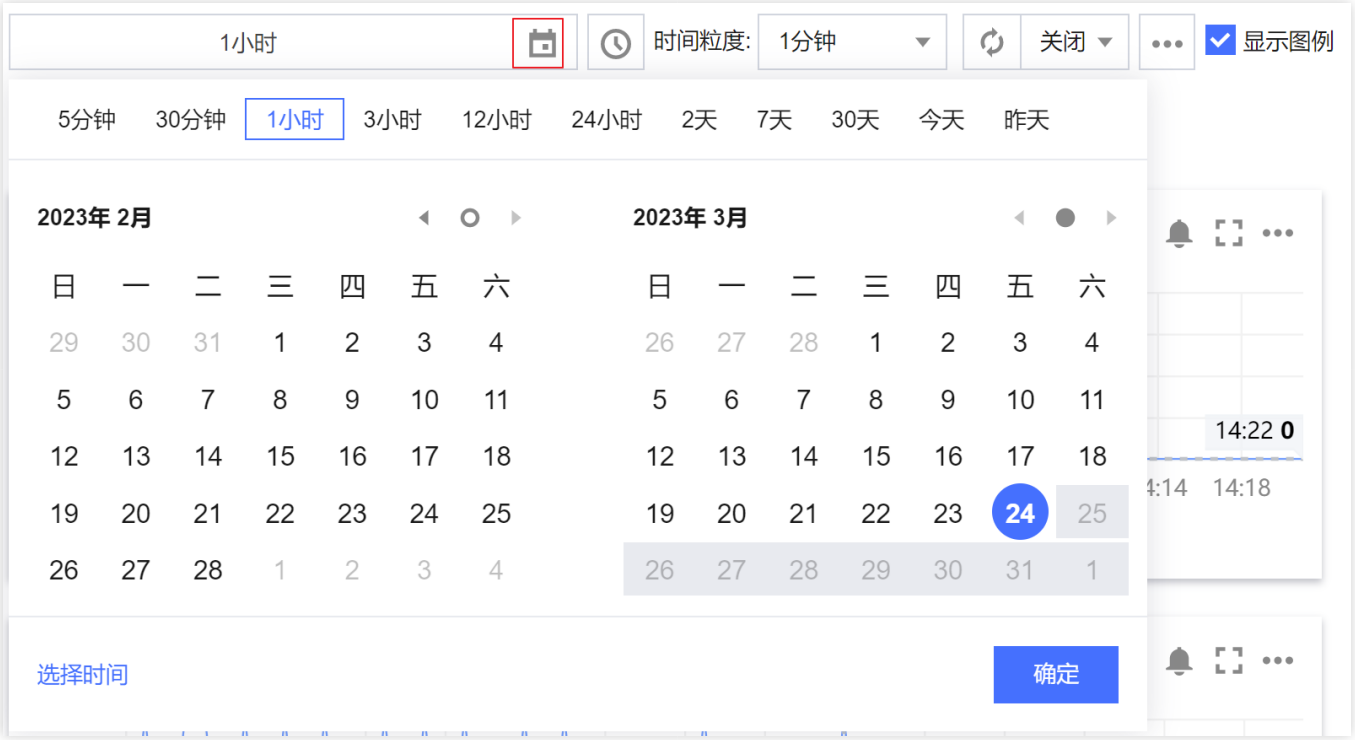


切换页签后，单击监控数据，查看对应的指标数据。

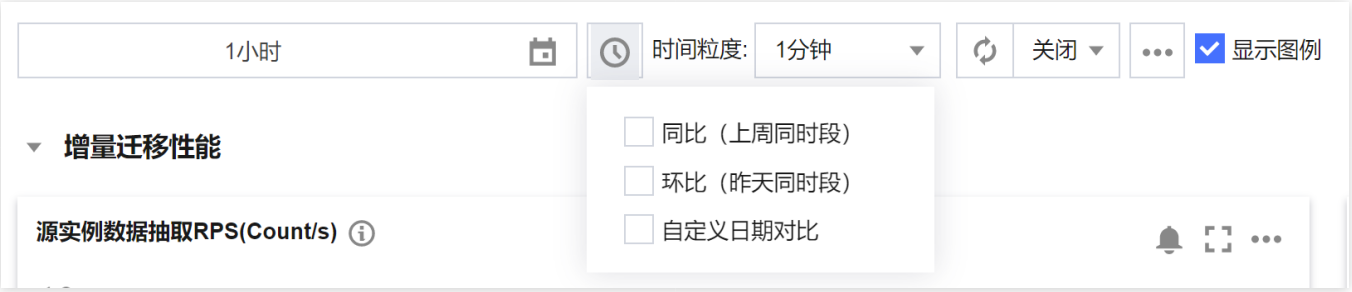


3. 时间筛选。

- 支持自定义对比时间范围，或者快捷选择时间范围。



- 支持时间对比维度，同比、环比、自定义日期对比。


1小时
时间粒度:
1分钟
关闭
显示图例
同比（上周同时段）
环比（昨天同时段）
自定义日期对比
增量迁移性能
源实例数据抽取RPS(Count/s)

常见问题
数据迁移

通用问题

使用 DTS 进行数据迁移/同步，对源数据库有啥影响？

1. 对源数据库的数据内容无影响，源库可正常进行业务写入。DTS 进行数据迁移时，其实是复制了源数据库的一份数据，不会删除源数据库的内容，所以对源数据库的数据没有影响。
2. 对源数据库的性能有一定影响，影响主要在 CPU。
   DTS 在执行全量数据迁移/同步时，会将源库的全量数据全部读取一次，所以会增加源库自身的压力。
   以源库为 MySQL，规格8核16G 为例，DTS 任务默认采用8线程并发，在网络无瓶颈的情况下，DTS 任务对源库的性能影响如下：
   - DTS 全量导出阶段：占用源库约18%-45%的 CPU，增加源库约40-60MB/s的查询压力，占用约8个活跃 session 连接数。
   - DTS 增量导出阶段：对源数据库基本无压力，只有一个连接实时监听源库的 binlog 日志。
3. 全量导出阶段源库的各连接详情。

   i. （可选）分块阶段。源库表的预估行数小于200万行，不进行分块，大于200万行则需要分块，增加如下连接。分块阶段时间很短，所以对源库增加的连接也是在短时间内。

      - 源库表的主键是整形类型，并且基本连续，会有类似如下的少量 SQL。

        `SELECT MIN(`id`), MAX(`id`) FROM `db`.`table`

      - 源库表的主键是字符串或者不连续的整形类型，会有类似如下 SQL。ORDER BY id limit语句会略微增加对源库的负载压力，DTS 默认使用3线程，同时会根据源数据库的规格调整，确保对源端的负载压力可控。

        SELECT `id` FROM `db`.`table` WHERE `id` > `某个id` ORDER BY `id` limit 500000, 1

   ii. 任务开始导出阶段会有1个线程查询系统表获取导出信息，会有类似如下 SQL。

   SELECT TABLE_NAME, TABLE_TYPE, ENGINE, TABLE_ROWS FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_SCHEMA='db'

   SELECT TABLE_NAME FROM INFORMATION_SCHEMA.TABLE_CONSTRAINTS WHERE ABLE_

SCHEMA='db' and TABLE_NAME='table' and CONSTRAINT_TYPE='PRIMARY KEY';

iii. 在结构导出阶段，会有导出线程数（默认8）的连接执行如下 SQL。

```
SELECT COLUMN_NAME, DATA_TYPE, COLUMN_TYPE, NUMERIC_PRECISION, NUMERIC_SC
ALE, CHARACTER_SET_NAME, COLUMN_DEFAULT, IS_NULLABLE from information_schema.
COLUMNS "
  "where TABLE_SCHEMA='db' and TABLE_NAME='db';
```

iv. 数据导出阶段，会有导出线程数（默认8）的连接执行如下 SQL。

```
SELECT /*!40001 SQL_NO_CACHE */ column FROM db.table where id>= 'id' AND id <
'id';
```

v. 无锁导出时，会有如下类似锁表 SQL，锁表只是为了获取无主键表的一致性位点，获取后即会解锁。

```
lock table xxx read
```

vi. 有锁导出时，会有类似如下 SQL 的全局锁。

```
flush table xxx with read lock
```

使用 DTS 进行数据迁移/同步，对目标数据库有啥影响？

1. 全量导入阶段，DTS 写入目标库时，对目标库的主要影响在 CPU 和 IOPS。
   i. 以源库为 MySQL，规格8核16G 为例，DTS 任务默认采用8线程并发，在网络无瓶颈的情况下，DTS 全量导入阶段对目标库的性能影响：占用目标库 CPU 约20%-49%，占用 IOPS 约1200-3100，占用小于8个活跃 session 连接数。
   ii. 全量导入阶段目标库的连接详情。
   - 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 xxx (id,name,msg) values (xxx);
2. 增量导入阶段，DTS 会把源数据库 binlog 中的增量数据解析成 SQL，然后在目标数据库中执行，总连接数小于32个 session，具体如下：
   - DDL 会串行执行，执行 DDL 时，不会有其他 DML 执行。
   - DML 最多会有32个连接（都为短连接，超时时间30秒），其中 DML 只是简单的 insert、update、delete、replace 语句。

使用 DTS 进行数据迁移，目标库是否需要为空？

数据迁移到目标库时，可以选择整个实例迁移，也可以选择指定对象迁移。

- 整个实例迁移：要求目标数据库为空，需要先清空目标库，不为空系统校验不通过，无法发起任务。
- 指定对象迁移：选择指定的库、表等对象进行迁移，系统会校验源库和目标库是否有同名的库表，如果有，则会校验不通过，提示用户修改后再进行迁移任务。

DTS 是否支持源库使用只读实例进行迁移？

支持。源库为自建数据库的场景，在 DTS 连接源库的配置中，输入只读实例的 IP 即可。

DTS 是否支持源库使用备库进行迁移？

源库为自建数据库的场景，在 DTS 连接源库的配置中，输入备库的 IP 即可；源库为云数据库实例的场景，在 DTS 连接源库的配置中，只能选择实例 ID，无法连接备库。

数据迁移过程中源库是否支持写入？

支持，迁移过程中源库可以正常进行数据写入，但在结构迁移、全量迁移阶段不能对源库进行 DDL 操作，否则可能会导致任务失败。等到增量迁移阶段，DDL 和 DML 都可以正常操作。

数据迁移过程中目标库是否支持写入？

目标库不需要设置为只读，但是建议不要对目标库进行写入，如果在迁移过程中，用户同时向目标库写入，可能会导致最终源和目标的数据不一致。

DTS 是否支持同一个数据库实例内不同库表的迁移？

不支持，DTS 仅支持源库和目标库为不同的数据库实例之间的数据迁移，不支持源库和目标库为同一个数据库实例（同一个数据库实例内不同库表对象的迁移）。

同一个源端数据库是否支持配置多个 DTS 任务，往不同的云数据库实例上迁移？

支持，支持同一个源端迁移到多个目标端（任务可并行处理），也支持多个源端迁移到同一个目标端（需要等前一个任务进行到增量阶段才可发起新一个任务）。需要注意多个任务并行，可能会增加源端或者目标端的访问压力，影响迁移速率。如果您确定需要对同一个源端数据库创建多个任务，可以在创建第一个迁移任务后，通过操作列更多 > 创建类似任务，快捷创建相同的任务。

DTS 是否支持定时自动迁移？

支持，您可以创建 DTS 后，在修改配置时将选择定时执行的选项，并配置定时迁移时间。

迁移过程中可以监控任务的进度吗？

可以，您可以在 DTS 控制台 数据迁移页面查看迁移任务进度。

为什么数据增量迁移服务会有15天设置?

数据增量迁移服务目前使用的是就近代理服务器接入，通过内网专线降低了走公网的抖动问题，确保了数据传输的质量，15天的设置是为了能有效降低代理的服务器的连接压力，如果超过15天，现有阶段并不会进行强制断开，只是从合理使用迁移资源出发做了时间设置规定。

数据校验为什么需要源数据库实例不为只读？

数据校验需要在源实例中创建新库 __tencentdb__ ，并在该库下写入 CheckSum 表，在该实例只读时将会跳过数据校验阶段。

DTS 数据迁移能指定库表进行迁移么？

可以，迁移对象可以选择整个实例，也可以选择指定库表对象。

数据迁移什么时候结束？

用户选择增量迁移时，任务长时间没有结束，有可能需要用户自己进行结束操作。

- 如果迁移类型选择结构迁移或者全量迁移，则任务完成后会自动结束，不需要用户手动结束。
- 如果迁移类型选择全量 + 增量迁移，则全量迁移完成后会自动进入增量数据同步阶段，增量数据同步不会自动结束，需要您手动单击完成结束增量数据同步。
    - 请选择合适时间手动完成增量数据同步，并完成业务切换。
    - 观察迁移阶段为增量同步，并显示无延迟状态，将源库停写几分钟。
    - 目标与源库数据差距为0MB及目标与源库时间延迟为0秒时，手动完成增量同步。

为什么全量迁移前后数据的大小不一致？

因为源库和目标库的碎片空间不一样，源库可能存在一些空洞数据，因此全量迁移完成后目标库的表存储空间很可能会比源库的表存储空间小。建议用户迁移完成后使用数据一致性校验来核对源库和目标库内容是否一致。

DTS 数据传输服务，是否支持跨国数据库迁移？

支持，接入方式选择公网可以实现跨国数据传输。

DTS 迁移过程中任务异常中断，是否可以重新启动？

可以，用户在配置任务时，可以设置自动重试策略，在任务发生异常中断（如源库或者目标库短暂不可用，网络问题等），DTS 支持在设置的时间范围内进行自动重试，无需手动干预。

MySQL 常见问题

MySQL 迁移过程中是否会加锁？

MySQL 迁移过程中的加锁，指的是对源数据库加全局锁（FTWRL），选择全量数据迁移的场景中，才会涉及到加锁处理。
当前 DTS 在 MySQL/MariaDB/Percona/TDSQL MySQL 之间的数据迁移中，默认使用无锁迁移。无锁迁移指对源

库不加全局锁（FTWRL），仅对无主键的表加表锁。

是否支持迁移无主键的表？

支持，但是建议用户选择有主键的表，无主键的表也可以发起任务，但是会造成以下影响：

- 迁移/同步无主键表，可能会导致数据重复。
- 无主键表在有 DML 操作时，容易导致数据同步延迟。
- 源数据库为阿里云 RDS，PolarDB 时，由于 RDS，PolarDB 在 Binlog 中为无主键或无非空唯一键的表加上附加主键列，但在表结构中不可见，可能会导致 DTS 无法识别，数据结果异常。

对源库设置 binlog_format 为 row 之后，如何确保源库 binlog_format 格式立刻生效？

设置 binlog_format 为 row 后，需重置当前数据库上的所有业务连接（当源库在从机时，还需重置主从同步 SQL 线程），避免当前业务连接继续使用老格式写入。
在上述操作未结束之前，请不要创建或者启动迁移任务，避免产生数据不一致。

如果迁移的源实例里面有 Toku 引擎，迁移会有什么需要注意的？

如果源实例涉及了 Toku 引擎，我们会在迁移时默认转成 InnoDB，然后 Cluster Index 和 TokuDB 的压缩特性的表需要提前做处理，目前迁移不支持，另外对 Toku 引擎的 DDL 操作也是迁移不支持。

数据同步

数据同步对目标库有啥影响，是否要求目标库为空？

使用 DTS 进行数据同步对目标库没有影响，不需要目标库为空。DTS 支持检测源库和目标库是否有同名对象，并支持在发生同名对象冲突时按照设置的策略进行处理，将任务报错提醒用户，或者忽略报错继续任务。

源/目标实例发生 HA（High Availability）切换时，同步服务是否会受影响？

- 源实例支持并开启 GTID（Global Transaction Identifier）时，在增量同步阶段，如果源实例发生 HA，那么服务会自动重连，同步数据流在 HA 完成后迅速恢复。
- 目标实例在增量同步阶段发生目标实例 HA，服务也会自动重连，同步数据流在 HA 完成后迅速恢复。

支持将高版本实例的数据同步到低版本的实例吗？

不支持。同一个类型的数据库，目标实例的大版本必须不小于源实例。以 MySQL 实例为例，不支持 MySQL 5.7 的实例同步到 MySQL 5.6。

源/目标能否为云下实例？

可以。可以将云上的数据库同步到云下。

在双向同步拓扑中，是否可以对 DDL 进行双向同步？

不可以。创建同步实例时，只能允许其中一个实例进行 DDL 同步，否则检测算法检测到 DDL 循环，会禁止其中一个实例的创建。

是否支持非事务引擎？

当前技术方案采用在事务中打上路由信息来标记事务的来源，依赖于事务的原子性。基于非事务引擎的库表，会破坏了事务的原子性，无法保证数据一致，不建议用户使用。

同步任务启动后，是否支持追加同步对象或者删除已勾选的同步对象？

支持。选择需要修改的同步任务，通过操作 > 修改同步配置，可以修改同步任务配置。
可支持增加/删除同步对象、修改主键冲突策略、SQL 同步策略等。修改同步配置时，已有的同步任务不会暂停，也不会受影响。

错误处理
常见错误处理

本章节提供了在任务运行过程中，常见的错误及处理方法，用户可按照对应指导进行错误处理。

MySQL 常见错误

如下为 MySQL 数据库在迁移、同步过程中的报错和处理方法，所列错误码为 MySQL 系统的错误码。

错误码	报错码说明	报错场景	报错示例	分析和处理方法
1227	权限问题	数据迁移、数据同步	Error 1227: Access denied.	- 问题分析 执行任务的账号没有连接源库/目标库的权限。 - 处理方法 给执行任务的账号进行授权。具体账号权限要求请参考操作指导中的对应文档。
1040	数据库连接数太多	数据迁移、数据同步	Error 1040: Too many connections.	- 问题分析 数据库连接数太多。 - 处理方法 修改源数据库的最大连接数 max_connections 为更大值，关闭源库不使用的连接，或者稍后在业务量少时进行任务重试。
1045	操作被拒绝	数据迁移、数据同步	Error 1045 (28000): Access denied for user '{{xx}}'@'{{xx}}' (using password: xx)	- 问题分析 - 在任务运行中用户修改了账号权限或密码信息。 - 在源库或者目标库上未对 DTS 的服务 IP 进行授权。 - 处理方法 - 检查是否有修改账号或者密码操作，如果有请撤销或者改回，具体账号权限要求请参考操作指导中的对应文档。 - 对 DTS 的服务 IP 进行授权。
1050	库表已存在，重复执行 DDL 语句	数据迁移、数据同步	Error 1050: Table {{}} *already exists, binlog position:<{{*}}>, gtid: {{}}*, related tables: {{*}}	- 问题分析 - 创建多合一迁移/同步任务，多个源库执行了相同的 DDL 操作导致在目标库 DDL 操作重复。多合一场景中，仅支持在一个同步任务中勾选 DDL 操作。 - 任务过程中，用户同时在目标库中也创建了该表，导致源库 DDL 操作同步到目

错误码	报错码说明	报错场景	报错示例	分析和处理方法
				标库中，重复执行。 - 网络异常或语句执行时间太长，任务重试过程中造成 DDL 重复。 - 处理方法 按照如上分析排查。
1054	相关表中包含未知列	数据迁移、数据同步	Error 1054: Unknown column {{*}} *related tables: {{}}*	- 问题分析 - 在任务启动前，未选择迁移/同步该表结构，目标库中不包含对应列。 - 在任务运行中，用户同时操作了目标库，删除了该列。 - 处理方法 请在目标库上确认该列是否存在；如果不存在，请补齐该列后进行任务重试。
1062	主键冲突报错	数据同步	Error 1062: Duplicate entry '{{xx}}' for key 'PRIMARY', related tables: '{{xx}}'.	- 问题分析 同步场景中，如果主键冲突处理机制选择冲突报错，则 DTS 遇到目标库与源库同步的数据发生主键冲突会报错。 - 在任务运行过程中，手动在目标库中写入数据，导致目标库中已存在相同主键的记录。 - 在任务运行前，源库关闭了唯一键检查，在源库本身已经存在重复主键数据。 - 在任务运行过程中，未同步 Delete 操作，导致源库数据删除后未同步至目标库，造成源库插入数据时与目标库主键冲突。 - 处理方法 - 检查源库是否有重复主键，如果有请先处理。 - 修改或者删除目标库中对应数据表的主键，然后重试任务。
1071	索引字段长度太长	数据迁移、数据同步	Error 1071 (42000): Specified key was too long; max key length is 767 bytes.	- 问题分析 默认情况下，InnoDB 引擎单一字段索引的长度最大为767bytes，即：767/2<384个双字节的字段，或者767/3<256个三字节的字段。GBK 是双字节，UTF-8是三字节，utf8mb4_unicode_ci 是四字节。MySQL 5.6 及其以上版本，所有 MyISAM 表都会被自动转换为 InnoDB，所以在自建数据库上有超过767bytes的组合索引列，同样的建表语句在自建库上运行没问题，但是在 MySQL 5.6 版本以上就会有问题。

错误码	报错码说明	报错场景	报错示例	分析和处理方法
				- 处理方法 修改文件中出错行组合索引列的长度。 示例-建表时索引长度定义最大为255： create table test(test varchar(255) primary key) char set utf8;
1146	库表不存在	数据迁移、 数据同步	Error 1146: Table '{{xx}}' doesn't exist on query.	- 问题分析 - 任务过程中，目标库上删除了该表。 - 源库在数据导出阶段进行了库表结构变更的 DDL。 - 任务启动前，未选择迁移/同步该表结构。 - 处理方法 请到目标库上执行 show create table xxx ，确认该表是否存在；如果不存在，请在目标库上手动创建该表。
1213	源库和目标库双写引起死锁	数据迁移、数据同步	Error 1213: Deadlock found when trying to get lock; try restarting transaction, related tables: '{{xx}}'.	- 问题分析 DTS 在目标库的写操作与用户在目标库的写操作冲突，造成死锁。 - 处理方法 - 终止死锁进程，重建任务。 - 建议实例控制下更新操作的加锁逻辑，给表加索引，尽量走行锁，减少锁开销。
1236	源端 Binlog 日志问题	数据迁移、数据同步	Error 1236 (HY000): Cannot replicate because the master purged required binary logs. Replicate the missing transactions from elsewhere, or provision a new slave from backup……	- 问题分析 源库 Binlog 日志保存时间短，DTS 拉取时已经被清理，或者拉取到的 Binlog 位点不正确。 - 处理方法 确认源库 Binlog 的保存时间（ expire_logs_days ）设置符合业务需求，建议保存时间大于3天，然后重建任务。
1414	数据导出阶段源库进行了变更库表结构的 DDL	数据迁移	Error 1414: Table definition has changed, please retry transaction.	- 问题分析 在源库数据导出阶段，不能进行库表结构变更的 DDL 操作，否则会可能会导致报错。 - 处理方法 重建迁移任务。

DTS 常见错误

如下为 DTS 系统在迁移、同步过程中的常见报错及处理方法。

报错说明	报错场景	报错示例	分析和处理方法
数据库连接异常	数据迁移、数据同步	- {{}}*invalid connection*{{}}. - driver: bad connection,{{}} - *dial tcp* {{}}: connect: connection refused.	问题分析 - 源/目标库被隔离或下线。 - 源/目标库长时间重启不成功。 - 源/目标库发生主从切换长时间不成功。 - 源/目标库负载过高。 - 人工或者程序定期/不定期地 Kill 源/目标库上的连接。 - 其他可能导致网络连接失败的情况，例如对源/目标库设置了网络安全策略阻止了接入请求。 处理方法 请按照上述分析逐个排查并解决。 对于云数据库，可通过实例控制台的监控信息协助排查和解决问题，解决成功后可在控制台重试任务进行恢复。
数据库连接异常	数据迁移、数据同步	dial tcp {{*}}: connect: connection refused.	问题分析 - 源/目标库被隔离或下线。 - 源/目标库长时间重启不成功。 - 源/目标库发生主从切换长时间不成功。 - 源/目标库负载过高。 - 其他可能导致网络连接失败的情况，对源/目标库设置了网络安全策略阻止了接入请求。 处理方法 请按照上述分析逐个排查并解决。 对于云数据库，可通过实例控制台的监控信息协助排查和解决问题，解决成功后可在控制台重试任务进行恢复。
源数据库存在较长时间的 SQL 在运行导致加锁失败	数据迁移、数据同步	- Find Resumable Error, src db has long query sql, fix it and try it later. - Find Resumable Error: Task failed due to table lock failure caused by time-consuming SQL query statements in source instance.	问题分析 - 如果源数据库存在较长时间的 SQL 在运行（大于5s），为保证源数据业务不受影响，DTS 需要等慢 SQL 运行结束后再加锁进行数据导出，默认加锁时间为60s，超时后加锁失败，任务报错。 处理方法 - 处理源库中的慢 SQL 或者等慢 SQL 运行结束后进行任务重建。

报错说明	报错场景	报错示例	分析和处理方法
Binlog 参数格式不符合要求	数据迁移、数据同步	- Statement binlog format unsupported:{{xx}}. - binlog must ROW format, but MIXED now. - binlog row before/after image not full, missing column {{xx}}, binlog position:{{xx}}, gtid:{{*}}.	问题分析 - 为了保证数据的正确性和完整性，DTS 会对源库 Binlog 相关参数做如下要求，在检查阶段也会对源库的 Binlog 参数进行检查，不符合要求则会报错，无法启动任务。 - 检查通过任务启动后，如果用户修改了源库 Binlog 参数也会导致任务报错，请确保源库 Binlog 满足如下要求。 - binlog_format 需要设置为 ROW 。 - binlog_row_image 需要设置为 FULL 。 处理方法 - 根据提示修改报参数，详细操作指导请参考 Binlog 参数检查，然后重建任务。 注意：参数修改后需要重启线程才能生效，数据库重启后参数会恢复初始配置，所以重启后请确认参数配置正确。

迁移慢或者进度卡住

现象描述

用户在进行迁移/同步任务时，时间过长或者进度卡住。

可能原因

- 迁移数据量比较大。
- 源库有大事务或者慢 SQL 正在执行。
- 源数据内容不合规。
- 网络问题，带宽限制或者网络抖动。
- 增量迁移或者同步场景中源库无数据写入。

> 说明：
>
> 迁移场景中，如果迁移类型选择了“全量 + 增量迁移”，当全量迁移任务完成后，因为还一直有增量迁移任务，需要用户自行结束任务（在任务列表操作列单击完成），否则任务一直在运行，这不属于进度卡住的情况。

处理方法

迁移数据量比较大

数据量较大导致迁移/同步进度慢。

源库有大事务或者慢 SQL 在执行

检查源库是否有大事务或者慢 SQL，因为源库上的大事务或慢 SQL 本身执行时间较长，DTS 解析后再同步到目标端也需要一段时间，所以会导致进度卡住或者延迟。如有这种情况，需要等大事务或者慢 SQL 执行结束后延迟会慢慢降下来，或者避免迁移/同步过程中源库有慢 SQL 或大事务。

源数据库内容不合格

源数据库中的内容不合规，如源端存在无主键表，存在这些表的大查询会造成进度慢。建议给源库的表都加上主键，或者不要迁移无主键的表。

网络问题

如果是自建数据库，需要检查网络带宽是否有限制。

增量迁移或者同步场景中源库无数据写入

增量迁移或者同步场景中，源库长时间没有数据写入，或者有空的 Binlog，在源库进行数据写入即可恢复任务。

数据同步有延时

问题现象

源数据库和目标数据库的同步内容有延时。

可能原因

- DTS 选择的传输链路规格较低。
- 目标库负载过大。
- 目标库规格较低。
- 源库有大事务或者慢 SQL 正在执行。
- 源端为只读，并且长时间没有数据写入。
- 网络问题，带宽限制或者网络抖动。

处理方法

DTS 选择的传输链路规格较低

请先查看监控数据中源库和目标库的 RPS，如果用户选择的传输链路规格较低，并且 RPS 已达到该规格的上限，请升级传输链路规格，不同规格的上限请参考数据迁移规格说明；如果用户已经选择了最高的规格，RPS 仍然达到规格上限，则按照以下办法进行排查。

目标库负载过大

目标库负载大时，可以等业务量小的时候看情况是否正常，或者对目标库的规格进行升级。

目标库规格较低

对目标库的规格进行升级。

源库有大事务或者慢 SQL 在执行

检查源库是否有大事务或者慢 SQL，因为源库上的大事务或慢 SQL 本身执行时间较长，DTS 解析后再同步到目标端也需要一段时间，所以会导致进度卡住或者延迟。如有这种情况，需要等大事务或者慢 SQL 执行结束后延迟会慢慢降下来，或者避免迁移/同步过程中源库有慢 SQL 或大事务。

增量迁移或者同步场景中源库无数据写入

在源端为只读的场景中，如果源库长时间没有数据写入，或者有空的 Binlog，就会导致数据延时，请在源库进行数据写入即可恢复正常。源端非只读，不存在该问题。

网络问题

如果是自建数据库，需要检查网络带宽是否有限制。