

ALIBABA CLOUD

阿里云

数据传输
最佳实践

文档版本：20201111

 阿里云

法律声明

阿里云提醒您阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
 危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 危险 重置操作将丢失用户配置数据。
 警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	 警告 重启操作将导致业务中断，恢复业务时间约十分钟。
 注意	用于警示信息、补充说明等，是用户必须了解的内容。	 注意 权重设置为0，该服务器不会再接受新请求。
 说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击设置> 网络> 设置网络类型。
粗体	表示按键、菜单、页面名称等UI元素。	在结果确认页面，单击确定。
Courier字体	命令或代码。	执行 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	<code>bae log list --instanceid</code> <i>Instance_ID</i>
[] 或者 [a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ } 或者 {a b}	表示必选项，至多选择一个。	<code>switch {active stand}</code>

目录

1.业务切换流程	05
2.通过触发器和函数实现PostgreSQL的DDL增量迁移	07
3.关闭RDS MySQL慢日志提升迁移性能	11
4.RDS MySQL修改字符集（平滑方案）	12
5.源库存在触发器时如何配置同步作业	16
6.不同库名间的数据迁移	19
7.数据订阅SDK容灾	20
8.通过数据订阅实现缓存更新策略	21

1. 业务切换流程

为最大化地减少数据迁移对业务的影响，您可以参照本文的流程执行业务切换并建立回退方案。

前提条件


已配置数据迁移任务，且数据迁移任务处于迁移中或已完成状态，配置案例请参见[DTS数据迁移方案概览](#)。

注意事项

- 由于执行业务切换操作需要停止数据库写入并暂停业务，请选择业务低峰期操作以降低影响。
- 为便于区分会话信息以及提升数据安全性，建议单独创建并授权一个数据库账号以供数据迁移使用。

操作步骤

1. 等待数据迁移任务的进度变更为**增量迁移**，并显示为**无延迟**状态或延迟时间低于5秒。

 **说明** 如果数据迁移任务在配置时未勾选**增量数据迁移**，进度条不会变更为**增量迁移**，且迁移完成后会自动结束任务。此类情况通常约定在执行数据迁移任务之前将业务中断并停止新的数据写入源数据库，所以请直接跳转至步骤5继续操作。

2. 将业务暂时中断，禁止新的数据写入源数据库。
3. 登录源数据库，根据数据库类型选择下述命令查看会话信息，确保没有新的会话执行写入操作。

 **说明** 上述语句查询到的进程或会话信息中，包含DTS连接源数据库的进程或会话。


MySQL
SQL Server
Oracle
PostgreSQL
Redis
MongoDB

```
show processlist;
```

4. 等待迁移任务的**增量迁移**再次进入**无延迟**状态，保持1分钟或以上，然后手动结束迁移任务。

5. 保持业务中断状态，取消源数据库禁止写入的限制。
6. 创建反向数据迁移任务并启动，用于将目标库后续产生增量数据迁移回源数据库。此步骤创建的反向迁移任务的作用是为业务提供回退方案，业务恢复运行后，一旦出现异常可将业务切换至原有的数据库中。


例如原先是将自建MySQL迁移至RDS MySQL，反向数据迁移任务配置案例请参见[从RDS MySQL迁移至自建MySQL](#)（仅选择**增量数据迁移**）。

 **警告** 在配置反向数据迁移任务时，在设置迁移类型及列表环节仅需选择**增量数据迁移**，然后选择需要迁移回源数据库的库或表。

7. 验证源和目标的数据一致后，将业务的数据库服务切换至目标数据库并恢复业务。
8. 由于创建的反向迁移任务会将目标库产生的增量数据实时迁移回源库，如果业务运行出现问题，可随时将业务切换回源库。

后续步骤

业务切换至目标数据库并稳定运行一段时间，测试所有业务涉及的功能并确认无问题，可结束反向数据迁移的任务，详情请参见[结束数据迁移任务](#)。

 **警告** 用于数据迁移的数据库账号拥有读写权限，为保障数据库安全性，请在数据迁移完成后，删除用于数据迁移的数据库账号或回收相关写权限。

常见问题

- Q: 业务切换后，如果业务运行出现问题如何处理？
A: 由于创建的反向迁移任务会将目标库产生的增量数据实时迁移回源库，如果业务运行出现问题，可随时将业务切换回源库。
- Q: 业务切换失败后，源库如何保证数据一致性？
A: 您可以在业务切换前备份源库。
- Q: 业务切换后，由于误操作导致在源数据库中写入了新的数据，应该如何处理？
A: 通过数据校验确认源和目标库的差异数据，进行手动订正。

2.通过触发器和函数实现PostgreSQL的DDL增量迁移

在使用DTS执行PostgreSQL数据库间的数据迁移前，可通过本文介绍的方法在源库创建触发器和函数获取源库的DDL信息，然后再由DTS执行数据迁移，在增量数据迁移阶段即可实现DDL操作的增量迁移。


前提条件

- 源库需为9.4及以上版本的自建PostgreSQL数据库。
- 数据迁移任务需在2020年10月1日之后创建。


背景信息

通过DTS执行PostgreSQL数据库间的数据迁移时，在增量数据迁移阶段，DTS仅支持DML操作（INSERT、DELETE、UPDATE）的同步，不支持DDL操作的同步。


通过本文的方法先在源库中创建触发器和函数来捕获DDL信息，再由DTS执行数据迁移，即可实现DDL操作的同步。

 **说明** 仅支持表级别DDL操作的同步：CREATE TABLE、DROP TABLE、ALTER TABLE（包含RENAME TABLE、ADD COLUMN、DROP COLUMN）。

操作步骤

 **警告** 如果源库中有多个数据库需要执行增量数据迁移，您需要重复执行步骤2到步骤5。

1. 登录源PostgreSQL数据库，相关方法请参见[连接PostgreSQL实例](#)或[psql工具介绍](#)。
2. 切换至待迁移的数据库。

 **说明** 本案例以psql工具为例介绍，您可以使用 `\c <数据库名>` 命令来切换数据库，例如 `\c dts tstdata`。

3. 执行下述命令创建存放DDL信息的表。

```
CREATE TABLE public.dts_ddl_command
(
  ddl_text text COLLATE pg_catalog."default",
  id bigserial primary key,
  event text COLLATE pg_catalog."default",
  tag text COLLATE pg_catalog."default",
  username character varying COLLATE pg_catalog."default",
  database character varying COLLATE pg_catalog."default",
  schema character varying COLLATE pg_catalog."default",
  object_type character varying COLLATE pg_catalog."default",
  object_name character varying COLLATE pg_catalog."default",
  client_address character varying COLLATE pg_catalog."default",
  client_port integer,
  event_time timestamp with time zone,
  txid_current character varying(128) COLLATE pg_catalog."default",
  message text COLLATE pg_catalog."default"
);
```

4. 执行下述命令创建捕获DDL信息的函数。

```
CREATE FUNCTION public.dts_capture_ddl()
  RETURNS event_trigger
  LANGUAGE 'plpgsql'
  COST 100
  VOLATILE NOT LEAKPROOF SECURITY DEFINER
AS $BODY$
declare ddl_text text;
declare max_rows int := 10000;
declare current_rows int;
declare pg_version_95 int := 90500;
declare pg_version_10 int := 100000;
declare current_version int;
declare object_id varchar;
declare alter_table varchar;
declare record_object record;
declare message text;
declare pub RECORD;
begin

select current_query() into ddl_text;

if (tag ~ 'CREATE TABLE|ALTER TABLE|TABLE REPLICATED|IDENTITY FULL
```



```
IF TG_TAG = 'CREATE TABLE' then -- ALTER TABLE schema.TABLE REPLICA IDENTITY FULL;
show server_version_num into current_version;
if current_version >= pg_version_95 then
for record_object in (select * from pg_event_trigger_ddl_commands()) loop
if record_object.command_tag = 'CREATE TABLE' then
object_id := record_object.object_identity;
end if;
end loop;
else
select btrim(substring(ddl_text from '[\t\r\n\v\f]*[cC][rR][eE][aA][tT][eE][\t\r\n\v\f]*.*[\t\r\n\v\f]*[tT][aA][bB][lL][eE][\t\r\n\v\f]+(.*)\(.*)' '\t\r\n\v\f) into object_id;
end if;
if object_id = '' or object_id is null then
message := 'CREATE TABLE, but ddl_text=' || ddl_text || ', current_query=' || current_query();
else
alter_table := 'ALTER TABLE ' || object_id || ' REPLICA IDENTITY FULL';
message := 'alter_sql=' || alter_table;
execute alter_table;
end if;
if current_version >= pg_version_10 then
for pub in (select * from pg_publication where pubname like 'dts_sync_%') loop
raise notice 'pubname=%',pub.pubname;
BEGIN
execute 'alter publication ' || pub.pubname || ' add table ' || object_id;
EXCEPTION WHEN OTHERS THEN
END;
end loop;
end if;
end if;

insert into public.dts_ddl_command(id,event,tag,username,database,schema,object_type,object_name,client_address,client_port,event_time,ddl_text,txid_current,message)
values (default,TG_EVENT,TG_TAG,current_user,current_database(),current_schema,"",inet_client_address(),inet_client_port(),current_timestamp,ddl_text,cast(TXID_CURRENT() as varchar(16)),message);

select count(id) into current_rows from public.dts_ddl_command;
if current_rows > max_rows then
delete from public.dts_ddl_command where id in (select min(id) from public.dts_ddl_command);
end if;
end
$BODY$;
```

5. 将刚创建的函数的所有者修改为DTS连接源库的账号，以postgres为例。

```
ALTER FUNCTION public.dts_capture_ddl()  
OWNER TO postgres;
```

6. 执行下述命令创建全局事件触发器。

```
CREATE EVENT TRIGGER dts_intercept_ddl ON ddl_command_end  
EXECUTE PROCEDURE public.dts_capture_ddl();
```

后续步骤

根据源库的版本，选择下述步骤配置数据迁移任务：

- [从自建PostgreSQL（9.4-10.0版本）增量迁移至RDS PostgreSQL](#)
- [从自建PostgreSQL（10.1-12版本）增量迁移至RDS PostgreSQL](#)

 **说明** 数据迁移任务释放后，您需要登录源PostgreSQL数据库，执行下述命令删除触发器和函数。

```
drop EVENT trigger dts_intercept_ddl;  
drop function public.dts_capture_ddl();  
drop table public.dts_ddl_command;
```

3. 关闭RDS MySQL慢日志提升迁移性能

如果目标RDS MySQL的规格较低（CPU核数低于2核）且待迁移的数据量较大时，建议您关闭RDS MySQL实例的慢日志以提升数据迁移的性能。

前提条件


RDS MySQL实例的数据库版本为5.6、5.7或8.0。

背景信息

DTS在将数据迁移至目标RDS MySQL时，会在目标RDS MySQL实例中产生大量的慢日志，而RDS MySQL实例的慢日志采集程序会扫描查询慢日志表，导致RDS MySQL实例的CPU负载升高，从而影响数据迁移的性能。

操作步骤

1. 登录[RDS管理控制台](#)。
2. 在页面左上角，选择实例所在地域。
3. 找到目标实例，单击实例ID。
4. 在左侧导航栏，单击参数设置。
5. 找到slow_query_log参数，然后将参数的值修改为OFF。

 **说明** 数据迁移任务完成后，您可以再将slow_query_log参数的值设置为ON。

4.RDS MySQL修改字符集（平滑方案）

由于业务需求，需要变更表的字符集（例如从gbk变更为utf8mb4），如果直接使用ALTER命令修改字符集会锁表，当数据表过大时，对业务的影响较大。

本文介绍的操作方法需要先在目标实例中创建表结构信息（新字符集），再通过DTS将源实例的数据迁移至目标实例，同时可建立简易回滚方案或双写方案，平滑地完成RDS MySQL字符集的变更。

注意事项

- 在执行变更字符集之前，请对业务和SQL进行充分的兼容性测试和功能验证，确认正常后再执行本操作。
- DTS在执行全量数据迁移时将占用源实例和目标实例一定的读写资源，可能会导致数据库的负载上升。因此您需要在执行数据迁移前评估源实例和目标实例的性能，同时建议您在业务低峰期执行数据迁移。
- 如果源实例的待迁移对象没有主键或唯一约束，且所有字段没有唯一性，可能会导致目标实例中出现重复数据。


步骤一：将源实例中表的结构导入至新实例中

1. 导出源实例中需要变更字符集的表的结构脚本。

- i. 通过DMS登录源实例，详情请参见[通过DMS登录RDS数据库](#)。
- ii. 在页面上方选择**数据方案 > 导出**。

iii. 选择**新增导出任务 > 导出数据库**。

iv. 设置导出信息。

 **说明** 选择目标数据库，然后选择导出内容为结构并选择目标表，其余配置可根据业务需求调整。

v. 单击**确定**，然后在弹出的提示框中单击**YES**。

2. 解压下载的文件，然后修改表结构脚本，调整字符集配置信息，将其更换为需要变更的目标字符集。

 **说明** 本案例中，将gbk字符集更换为utf8mb4。


3. 新购一个与源实例规格相同的RDS MySQL实例，详情请参见[创建RDS MySQL实例](#)。

4. 将修改后的表结构脚本导入至新的RDS MySQL实例中。

- i. 通过DMS登录新的RDS MySQL实例，详情请参见[通过DMS登录RDS数据库](#)。
- ii. 在页面上方选择**数据方案 > 导入**。

iii. 单击**新增任务**。

iv. 按下图设置导入表结构脚本，单击开始。

 **说明** 导入成功后，您可以通过执行 `show create table <表名>;` 命令来确认表的字符集信息。

步骤二：将源实例中表的数据迁移至新的实例

1. 登录[数据传输控制台](#)。
2. 在左侧导航栏，单击[数据迁移](#)。
3. 在迁移任务列表页面顶部，选择迁移的目标集群所属地域。

4. 单击页面右上角的[创建迁移任务](#)。
5. 配置迁移任务的源库及目标库信息。

类别	配置	说明
无	任务名称	DTS会自动生成一个任务名称，建议配置具有业务意义的名称（无唯一性要求），便于后续识别。
源库信息	实例类型	选择RDS实例。
	实例地区	选择源RDS实例所属地域。
	RDS实例ID	选择源RDS实例ID。
	数据库账号	填入源RDS实例的数据库账号（需具备待迁移库的读写权限）。
	数据库密码	填入该数据库账号对应的密码。  说明 源库信息填写完毕后，您可以单击数据库密码后的 测试连接 来验证填入的源库信息是否正确。源库信息填写正确则提示 测试通过 ；如果提示 测试失败 ，单击 测试失败后的诊断 ，根据提示调整填写的源库信息。
	连接方式	根据需求选择 非加密连接 或 SSL安全连接 。如果设置为 SSL安全连接 ，您需要提前开启RDS实例的SSL加密功能，详情请参见 设置SSL加密 。
	实例类型	选择RDS实例。
	实例地区	选择目标RDS实例所属地域。
	RDS实例ID	选择目标RDS实例ID。
	数据库账号	填入目标RDS实例的数据库账号（需具备迁入的目标库的读写权限）。

源数据库信息	配置	说明
	数据库密码	填入该数据库账号对应的密码。 <div style="border: 1px solid #ccc; padding: 5px; background-color: #e6f2ff;"> ? 说明 目标库信息填写完毕后，您可以单击数据库密码后的测试连接来验证填入的目标库信息是否正确。目标库信息填写正确则提示测试通过；如果提示测试失败，单击测试失败后的诊断，根据提示调整填写的目标库信息。 </div>
	连接方式	根据需求选择非加密连接或SSL安全连接。如果设置为SSL安全连接，您需要提前开启RDS实例的SSL加密功能，详情请参见 设置SSL加密 。

6. 配置完成后，单击页面右下角的授权白名单并进入下一步。

? **说明** 此步骤会将DTS服务器的IP地址自动添加到RDS MySQL的白名单中，用于保障DTS服务器能够正常连接源和目标实例。

7. 选择迁移类型和迁移对象。

配置	说明
迁移类型	由于已经在 步骤一 中完成了结构迁移，此处无需勾选结构迁移。 <ul style="list-style-type: none"> ○ 如果只需要全量迁移，仅勾选全量数据迁移。 ○ 如果需要不停机迁移，同时勾选全量数据迁移和增量数据迁移。 <div style="border: 1px solid #ccc; padding: 5px; background-color: #e6f2ff; margin-top: 10px;"> ? 说明 <ul style="list-style-type: none"> ■ 如果没有勾选增量数据迁移，为保障数据一致性，数据迁移期间请勿在源库中写入新的数据。 ■ 增量数据迁移阶段支持同步的SQL操作为：INSERT、UPDATE、DELETE、REPLACE、CREATE FUNCTION、CREATE INDEX、CREATE PROCEDURE、CREATE TABLE、CREATE VIEW、DROP INDEX、DROP TABLE、RENAME TABLE、TRUNCATE TABLE。 </div>
迁移对象	在 迁移对象 框中选择待迁移的对象（该对象的结构须在 步骤一 中完成了迁移），然后单击 <input type="text"/> 将其移动至已选择对象框。 <div style="border: 1px solid #ccc; padding: 5px; background-color: #fff9c4; margin-top: 10px;"> 🔔 警告 由于已完成了结构迁移，请勿执行对象名映射操作，否则将导致迁移失败。 </div>

8. 启动迁移任务。

- i. 单击页面右下角的**预检查并启动**。

说明

- 在迁移任务正式启动之前，会先进行预检查。只有通过预检查，DTS才能迁移数据。
- 如果预检查失败，单击具体检查项后的 ，查看失败详情。根据提示修复后，重新进行预检查。

- ii. 预检查通过后，单击**下一步**。
- iii. 在弹出的**购买配置确认**对话框，选择**链路规格**并勾选**数据传输（按量付费）服务条款**。
- iv. 单击**购买并启动**，迁移任务正式开始。

步骤三：业务切换

您可以根据业务需求选择下述方案执行业务切换：

- 简易回退方案（有一定的回滚失败风险，无需改造业务的应用程序）
 - i. 在目标RDS MySQL实例中完成迁移后的数据校验。
 - ii. 执行业务切换并建立回退方案（将目标库的增量数据实时迁移回源库中），详情请参见[业务切换流程](#)。
 - iii. 回退方案保持一个业务周期，测试完所有功能，确认无误后可结束回退方案中创建的反向数据迁移任务。
 - iv. （可选）如不再需要源实例，可将其释放（按量付费实例）或退订（包年包月实例）。
- 双写方案（可保障回滚成功率，业务的应用程序改造量大）
 - i. 对业务的应用程序进行改造，让其实现双写的逻辑（即数据变更会同时向源和目标实例写入）。
 - ii. 结束数据迁移任务。
 - 全量数据迁移
请勿手动结束迁移任务，否则可能导致数据不完整。您只需等待迁移任务完成即可，迁移任务会自动结束。
 - 增量数据迁移
迁移任务不会自动结束，您需要手动结束迁移任务。
 - a. 观察迁移任务的进度变更为**增量迁移**，并显示为**无延迟**状态时，将源库停写几分钟，此时**增量迁移**的状态可能会显示**延迟**的时间。
 - b. 等待迁移任务的**增量迁移**再次进入**无延迟**状态后，手动结束迁移任务。

说明 您可以登录源实例，执行 `show processlist;` 查看会话信息，确保没有新的会话执行写入操作。


- iii. 在目标RDS MySQL实例中完成迁移后的数据校验。
- iv. 将业务的应用程序设置为双写，即数据变更会同时写入源和目标实例。
- v. 双写状态保持一个业务周期，测试完所有功能，确认无误后可调整为只写入目标实例。
- vi. （可选）如不再需要源实例，可将其释放（按量付费实例）或退订（包年包月实例）。

5.源库存在触发器时如何配置同步作业

当同步对象为整个库，且库中的触发器（TRIGGER）会更新库内某个表时，可能导致源和目标库的数据不一致。本文介绍触发器存在情况下如何配置同步作业。

案例介绍

数据库triggertestdata中存在着两个表：parent和child。其中，表parent中有一个触发器（定义：当INSERT一条数据到表parent之后，将其INSERT写入的数据写入一份至表child中）。

 说明 表结构及触发器定义如下表所示。

对象类型	名称	结构语句
表	parent	<pre>CREATE TABLE `parent` (`user_vs_id` int(11) NOT NULL AUTO_INCREMENT, `name` varchar(30) DEFAULT NULL, PRIMARY KEY (`user_vs_id`)) ENGINE=InnoDB AUTO_INCREMENT=2001 DEFAULT CHARSET=utf8</pre>
表	child	<pre>CREATE TABLE `child` (`sys_child_id` int(11) NOT NULL AUTO_INCREMENT, `user_vs_id` int(11) DEFAULT NULL, `name` varchar(30) DEFAULT NULL, PRIMARY KEY (`sys_child_id`)) ENGINE=InnoDB AUTO_INCREMENT=2001 DEFAULT CHARSET=utf8</pre>


对象类型	名称	结构语句
触发器	data_check	<pre>CREATE TRIGGER `triggertestdata`.`data_check` AFTER INSERT ON triggertestdata.parent FOR EACH ROW begin insert into child(user_vs_id, name) values(new.user_vs_id, new.name); end;</pre>

在此案例中，如果在数据同步时，在源库的表parent中执行了INSERT操作，将导致源和目标库中的表child数据不一致。为解决该问题，需要删除被迁移到目标库中的触发器。

操作步骤

本文以自建MySQL同步至RDS MySQL为例介绍配置流程，更多配置案例请参见[DTS数据同步方案概览](#)。

1. 创建一个数据迁移任务，将源库的结构迁移至目标库，相关案例请参见[从自建MySQL迁移至RDS MySQL（仅选择结构迁移）](#)。

 **说明** 在配置数据迁移任务时，在设置迁移类型及列表环节仅需选择结构迁移，然后选择待迁移的库或表。

2. 由于执行结构迁移时，触发器也会被迁移至目标库中。当迁移任务完成后，您需要登录目标数据库，执行下述命令删除被迁移过去的触发器。

```
drop trigger <触发器名称>;
```

示例：

```
drop trigger data_check;
```

3. 创建数据同步作业，将源库的数据同步至目标库中，配置案例请参见[从ECS上的自建MySQL同步至RDS MySQL（仅选择全量数据初始化）](#)。

② 说明

- 在同步作业的选择同步对象环节中，需和步骤1中选择的同步对象一致。
- 由于已经完成了结构迁移，在同步作业的高级设置环节中，您仅需选中全量数据初始化。

数据一致性测试

1. 登录源数据库，在表parent中插入一条记录。

```
insert into parent values(1,'testname');
```

执行该操作后，由于源库触发器的缘故，该条记录也会被写入至源库的child表中。

2. 登录源和目标数据库，查询表child的数据，比对其数据是否一致。

- 源库查询结果

- 目标库查询结果

经比对，数据一致。

6.不同库名间的数据迁移

DTS支持在配置数据迁移时映射数据库名称，通过该功能即可将源实例中某个数据库迁移至目标实例的某个不同名的数据库（例如从数据库A迁移至数据库B）。

适用场景

- 实例内不同数据库之间的数据迁移，例如实例中的数据库A迁移至数据库B。
- 跨实例不同数据库间的数据迁移，例如从源实例的数据库A迁移至目标实例的数据库B。

操作步骤

在配置数据迁移任务的设置迁移类型及列表环节中执行库名映射的操作，详细操作方法请参见[库表列映射](#)。

7.数据订阅SDK容灾

为了简化用户使用SDK的复杂度，DTS实现了SDK容灾机制。这个机制依赖于SDK消费数据时，向DTS中控节点实时汇报消费时间点。本小节简单介绍如何使用DTS提供的SDK容灾机制。

1. SDK消费代码中，每消费一个message要调用一次ackAsConsumed，向DTS汇报消费时间点。
2. 同一个订阅通道至少启动2个SDK下游互为主备。

消费架构图

□

填写链接方式

DTS的一个订阅通道只能有一个SDK拉取到增量数据。如果在一个订阅通道中接多个SDK下游，那么只有其中一个订阅SDK能够接收到增量数据。根据这个原理，可以搭建SDK灾备架构。

在上图这个示例架构中，对一个订阅通道中，接了两个SDK消费下游，这两个SDK互为主备，同一个时间内，只有主SDK能订阅并消费到增量数据。当主SDK异常或是出现网络连接异常时，DTS会自动failover漂移到备SDK上，同时使用最后消费的时间点启动备SDK。主SDK每消费一条数据，会向DTS控制系统汇报一个ACK，即向控制系统汇报消费时间点。

8.通过数据订阅实现缓存更新策略

本文介绍通过DTS的数据订阅功能，结合MongoDB或Redis实现高可靠、低延迟的缓存更新策略。

前提条件


已完成数据订阅通道（旧版）的配置，详情请参见[创建RDS MySQL数据订阅通道（旧版）](#)或[创建DRDS数据订阅通道](#)。

背景信息

为提高业务访问速度，提升业务读并发，在业务架构中引入了缓存层。业务的读请求路由到缓存层，通过缓存的内存读取机制提升业务读取性能；同时，为保证数据的完整性，业务的更新数据持久化存储至RDS MySQL中。此时，业务需要实现缓存更新策略，保证业务数据更新后，缓存中的数据能够被同步更新。

准备工作

创建AccessKey并获取AccessKey ID和AccessKey Secret信息，详情请参见[创建AccessKey](#)。

 **注意** 如果使用子账号（RAM用户）来订阅数据，该账号需具备AliyunDTSFullAccess权限，授权方法请参见[通过系统策略授权子账号管理DTS](#)。

操作步骤

本文以IntelliJ IDEA软件（Community Edition 2018.1.4 Windows版本）为例，介绍如何运行SDK示例代码来消费订阅数据。

1. [下载SDK包（dts-subscribe-3.0.jar）](#)。
2. 将 *dts-subscribe-3.0.jar* 添加到项目依赖中。
 - i. 打开IntelliJ IDEA软件，单击File > New > Project，然后根据提示创建一个Maven项目。
 - ii. 单击File > Project Structure。
 - iii. 单击左侧的Modules，然后在页面右侧单击 > 1 JARs or directories。
 - iv. 根据提示选择下载的文件：*dts-subscribe-3.0.jar*，将其加入到项目依赖中。
3. 根据业务需求选择下述步骤，运行MongoDB或Redis的缓存示例代码。
 - o 运行MongoDB缓存示例代码

- a. 双击打开左侧的 *pom.xml* 文件，将下述配置添加至该文件中。


```
<dependencies>
  <dependency>
    <groupId>com.vividsolutions</groupId>
    <artifactId>jts</artifactId>
    <version>1.13</version>
  </dependency>
  <dependency>
    <groupId>org.mongodb</groupId>
    <artifactId>mongo-java-driver</artifactId>
    <version>3.2.0</version>
  </dependency>
</dependencies>
```

- b. 保存并等待相关依赖包加载完成。

 说明 建议开启IntelliJ IDEA软件的Maven依赖自动更新功能。

- c. 在项目中新建一个名称为 *DTSMysql2Mongo* 的Java Class文件。
- d. [下载MongoDB缓存示例代码](#)。
- e. 将下载后的示例代码中的内容复制并替换至项目中新建的 *DTSMysql2Mongo* 文件中。

f. 修改项目中 *DTSMysql2Mongo* 文件中代码，设置代码中各参数对应的值。

参数	说明
accessKey	分别填写阿里云账号的AccessKey ID和AccessKey Secret信息，详情请参见 准备工作 。
accessSecret	
Subscription_Instance_ID	填写数据订阅实例ID。您可以登录 数据传输控制台 ，单击左侧导航栏的 数据订阅 ，然后找到目标数据订阅实例ID。
mongUrl	填写MongoDB数据库的连接地址及鉴权数据库名，格式为 <code><连接地址>:<服务端口>/<鉴权数据库名></code> ，例如 <code>ds-bp*****-pub.mongodb.rds.aliyuncs.com:3717/admin</code> 。
mongUserName	填写MongoDB数据库的数据库账号。 <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px;"> <p> 说明 该数据库账号需具备目标数据库的读写权限。例如订阅的表所属的库名为testdata，程序获取到增量变更后会将数据写入至MongoDB数据库的testdata库中，因此该数据库账号需具备testdata库的读写权限。</p> </div>
mongUserPassword	填写该数据库账号对应的密码。

g. 在IntelliJ IDEA软件界面的顶部，单击Run > Run运行该程序。

o. 运行Redis缓存示例代码

a. 双击打开左侧的 *pom.xml* 文件，将下述配置添加至该文件中。

```
<dependencies>
  <dependency>
    <groupId>com.vividsolutions</groupId>
    <artifactId>jts</artifactId>
    <version>1.13</version>
  </dependency>
  <dependency>
    <groupId>redis.clients</groupId>
    <artifactId>jedis</artifactId>
    <version>2.7.2</version>
  </dependency>
</dependencies>
```

b. 保存并等待相关依赖包加载完成。

 说明 建议开启IntelliJ IDEA软件的Maven依赖自动更新功能。

c. 在项目中新建一个名称为 *DTSMysql2Redis* 的Java Class文件。

d. [下载Redis缓存示例代码](#)。

e. 将下载后的示例代码中的内容复制并替换至项目中新建的 *DTSMysql2Redis* 文件中。

f. 修改项目中 *DTSMysql2Redis* 文件中代码，设置代码中各参数对应的值。

参数	说明
accessKey	分别填写阿里云账号的AccessKey ID和AccessKey Secret信息，详情请参见 准备工作 。
accessSecret	
Subscription_Instance_ID	填写数据订阅实例ID。您可以登录 数据传输控制台 ，单击左侧导航栏的 数据订阅 ，然后找到目标数据订阅实例ID。
redisPassword	填写Redis数据库的数据库密码。  说明 如果Redis数据库设置了密码，您还需要将代码中的73行由 <code>pool = new JedisPool(config, redisUrl, redisPort, 3000);</code> 修改为 <code>pool = new JedisPool(config, redisUrl, redisPort, 3000, redisPassword)</code> 。
redisPort	填写Redis数据库的服务端口，默认为6379。
redisUrl	填写Redis数据库的连接地址（域名或IP地址）。

g. 在IntelliJ IDEA软件界面的顶部，单击Run > Run运行该程序。

4. 等待程序运行，程序会获取源库的增量数据并将其写入至MongoDB或Redis数据库中。

执行结果

1. 登录订阅的源数据库，在订阅的源表中插入几条数据，可以查看到该程序订阅到了对应的增量数据变更，然后将对应的数据写入到目标数据库。

- o MongoDB缓存示例代码

- o Redis缓存示例代码

2. 登录目标数据库，查询对应的数据，可查看到对应的数据已正常写入。

- o MongoDB数据库

- o Redis数据库

