

阿里云 云数据库 PolarDB

PolarDB MySQL数据库

文档版本：20200709

法律声明

阿里云提醒您在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. 阿里云文档中所有内容，包括但不限于图片、架构设计、页面布局、文字描述，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
	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。	 禁止： 重置操作将丢失用户配置数据。
	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。
	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。
	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。	执行cd /d C:/window命令，进入Windows系统文件夹。
斜体	表示参数、变量。	bae log list --instanceid Instance_ID
[]或者[a b]	表示可选项，至多选择一个。	ipconfig [-all -t]
{ }或者[a b]	表示必选项，至多选择一个。	switch {active stand}

目录

法律声明	I
通用约定	I
1 概述	1
2 功能概览	2
3 PolarDB MySQL快速入门	5
4 常见问题FAQ	6
5 数据迁移或同步	16
5.1 数据迁移/同步方案概览	16
5.2 【注意事项】MySQL 5.7迁移到PolarDB MySQL 8.0	17
5.3 从RDS迁移至PolarDB	18
5.3.1 从RDS迁移到PolarDB后可申请无责退款	18
5.3.2 一键升级RDS MySQL至PolarDB MySQL	18
5.3.3 一键克隆RDS MySQL至PolarDB MySQL	27
5.3.4 从RDS MySQL迁移至PolarDB MySQL	29
5.4 PolarDB间的数据迁移	35
5.4.1 PolarDB MySQL集群间的数据迁移	36
5.5 从其他数据库迁移至PolarDB	41
5.5.1 从自建MySQL迁移至PolarDB MySQL	42
5.5.2 从本地MySQL迁移至PolarDB MySQL	47
5.5.3 从Amazon Aurora MySQL迁移至PolarDB MySQL	54
5.6 从PolarDB迁移至其他数据库	62
5.6.1 从PolarDB MySQL迁移至RDS MySQL	62
5.7 PolarDB间的数据同步	67
5.7.1 PolarDB MySQL集群间的单向同步	68
5.7.2 PolarDB MySQL集群间的双向同步	74
5.8 PolarDB与其他数据库的数据同步	82
5.8.1 从RDS MySQL同步至PolarDB MySQL	82
5.8.2 从PolarDB MySQL同步至RDS MySQL	88
5.8.3 从PolarDB MySQL同步至AnalyticDB for MySQL	94
5.8.4 从PolarDB MySQL同步至云原生数据仓库AnalyticDB PostgreSQL	101
5.8.5 从PolarDB MySQL同步至Datahub	107
5.8.6 从PolarDB MySQL同步到Kafka	113
5.8.7 从ECS上的自建MySQL同步至PolarDB MySQL	118
5.8.8 从通过专线、VPN网关或智能接入网关接入的自建MySQL同步至PolarDB MySQL	125
6 购买集群或存储包	133
6.1 创建PolarDB MySQL数据库集群	133
6.2 使用存储包	138
6.3 按量付费转包年包月	142

6.4 包年包月转按量付费.....	143
6.5 手动续费集群.....	144
6.6 自动续费集群.....	147
6.7 释放集群.....	153
7 集群访问.....	155
7.1 查看连接地址.....	155
7.2 连接数据库集群.....	157
7.3 集群地址.....	162
7.3.1 新增自定义集群地址.....	162
7.3.2 修改和释放集群地址.....	166
7.3.3 PolarProxy功能介绍.....	170
7.3.4 读写分离.....	174
7.3.5 PolarDB一致性级别.....	179
7.4 私有域名.....	184
8 账号和数据库.....	188
8.1 账号概述.....	188
8.2 注册和登录阿里云账号.....	189
8.3 创建和使用子账号.....	191
8.4 创建数据库账号.....	194
8.5 管理数据库账号.....	198
8.6 数据库.....	201
9 部署架构.....	204
9.1 主备切换.....	204
9.2 多可用区部署和更换主可用区.....	206
9.3 全球数据库.....	209
9.3.1 新建全球数据库和添加只读集群.....	210
9.3.2 管理全球数据库.....	214
10 弹性升降级.....	216
10.1 变更配置.....	216
10.2 临时升配.....	219
10.3 增加或删除节点.....	221
11 数据安全/加密.....	226
11.1 设置集群白名单.....	226
11.2 设置SSL加密.....	230
11.3 设置透明数据加密TDE.....	239
12 备份与恢复.....	243
12.1 备份数据.....	243
12.2 集群回收站.....	251
12.3 恢复数据.....	253
12.4 克隆集群.....	257
12.5 备份功能FAQ.....	258
13 诊断与优化.....	259
13.1 SQL洞察.....	259

13.2 性能监控与报警.....	264
13.3 慢SQL.....	269
13.4 一键诊断.....	272
13.5 自治中心.....	273
13.6 会话管理.....	275
13.7 实时性能.....	276
13.8 空间分析.....	277
13.9 锁分析.....	278
13.10 诊断报告.....	280
13.11 性能洞察.....	281
14 其他操作.....	284
14.1 开启Binlog.....	284
14.2 设置集群参数.....	286
14.3 设置可维护窗口.....	288
14.4 重启节点.....	289
14.5 待处理事件.....	290
14.6 标签.....	292
14.6.1 绑定标签.....	292
14.6.2 根据标签筛选集群.....	294
14.6.3 查看集群绑定的标签.....	295
14.6.4 解绑标签.....	295
15 内核功能.....	297
15.1 PolarDB MySQL内核兼容性说明.....	297
15.2 小版本升级.....	297
15.3 开启Binlog.....	298
15.4 Thread Pool.....	301
15.5 并行查询.....	306
15.5.1 并行查询.....	306
15.5.2 并行查询使用示例.....	317
15.5.3 并行查询使用限制与串行执行结果兼容问题.....	330

1 概述

PolarDB是阿里云自研的下一代关系型云数据库，兼容MySQL、PostgreSQL、Oracle引擎，存储容量最高可达100TB，单库最多可扩展到16个节点，适用于企业多样化的数据库应用场景。

PolarDB采用存储和计算分离的架构，所有计算节点共享一份数据，提供分钟级的配置升降级、秒级的故障恢复、全局数据一致性和免费的数据备份容灾服务。PolarDB既融合了商业数据库稳定可靠、高性能、可扩展的特征，又具有开源云数据库简单开放、自我迭代的优势。

PolarDB兼容Oracle数据库提供公共云和专有云形态，其中专有云形态支持CentOS、UOS、麒麟等操作系统，支持X86架构CPU以及ARM架构CPU（飞腾等）。

基本概念

- 集群

一个集群包含一个主节点以及最多15个只读节点（最少一个，用于提供Active-Active高可用）。集群ID以pc开头（代表PolarDB cluster）。

- 节点

一个独立占用物理内存的数据库服务进程。节点ID以pi开头（代表PolarDB instance）。

- 数据库

在节点下创建的逻辑单元，一个节点可以创建多个数据库，数据库在节点内的命名唯一。

- 地域和可用区

地域是指物理的数据中心。可用区是指在同一地域内，拥有独立电力和网络的物理区域。更多信息请参考[阿里云全球基础设施](#)。

控制台

阿里云提供了简单易用的Web控制台，方便您操作阿里云的各种产品和服务，包括云数据库PolarDB。在控制台上，您可以创建、连接和配置PolarDB数据库。

关于控制台的界面介绍，请参考[阿里云管理控制台](#)。

PolarDB控制台地址：[PolarDB控制台](#)。

2 功能概览

本文汇总了PolarDB MySQL相关功能供您参考。

功能	链接
数据迁移	数据迁移/同步方案概览
集群管理	创建PolarDB MySQL数据库集群
	使用存储包
	按量付费转包年包月
	手动续费集群
	自动续费集群
	释放集群
	设置集群参数
	小版本升级
	设置集群白名单
	克隆集群
	绑定集群标签
	根据标签筛选集群
	查看集群绑定的标签
	解绑集群标签
集群访问	查看连接地址
	连接数据库集群
	私有域名
	新增自定义集群地址
	修改和释放集群地址
	PolarProxy功能介绍
	PolarDB一致性级别
	读写分离
弹性升降级	变更配置
	临时升配
	增加或删除节点

功能	链接
账号	注册和登录阿里云账号
	创建和使用子账号
	创建数据库账号
	管理数据库账号
数据库	数据库
部署架构	多可用区部署和更换主可用区
	主备切换
全球数据库	新建全球数据库和添加只读集群
	管理全球数据库
数据安全	设置透明数据加密TDE
	设置SSL加密
备份与恢复	备份数据
	恢复数据
	集群回收站
诊断与优化	SQL洞察
	性能监控与报警
	慢SQL
	一键诊断
	自治中心
	会话管理
	实时性能
	空间分析
	锁分析
	诊断报告
	性能洞察
内核功能	开启Binlog
	Thread Pool
	并行查询
其他功能	设置可维护窗口
	待处理事件

功能	链接
	重启节点

3 PolarDB MySQL快速入门

快速入门旨在介绍如何创建PolarDB MySQL集群、进行基本设置以及连接数据库集群，使您能够了解从购买PolarDB到开始使用的流程。

使用流程

通常，从购买PolarDB（创建新集群）到可以开始使用，您需要完成如下操作。

1. [创建PolarDB MySQL数据库集群](#)
2. [设置集群白名单](#)
3. [创建数据库账号](#)
4. [查看连接地址](#)
5. [连接数据库集群](#)

4 常见问题FAQ

本文介绍PolarDB MySQL的常见问题和解答。

基本问题

- Q: 什么是PolarDB?

A: PolarDB是一个关系型数据库云服务, 目前已在全球十多个地域 (Region) 的数据中心部署, 向用户提供开箱即用的在线数据库服务。PolarDB目前支持3种独立的引擎, 分别可以100%兼容MySQL、100%兼容PostgreSQL、高度兼容Oracle语法, 存储容量最高可达100TB, 用户可以按需购买、按量计费, 最低每小时只需要付费1.3元即可体验完整的产品功能。详情请参见[#unique_64](#)。

- Q: PolarDB是什么时候发布? 什么时候开始商用?

A: 2017年9月发布公测, 2018年3月开始商用。

- Q: 集群和节点分别指的是什么?

PolarDB MySQL采用多节点集群的架构, 集群中有一个Writer节点 (主节点) 和多个Reader节点 (只读节点)。单个PolarDB MySQL集群支持跨可用区, 但不能跨地域, 面向集群进行管理和计费。详情请参见[#unique_65](#)。

- Q: 支持哪些编程语言?

A: PolarDB MySQL支持Java、Python、PHP、Golang、C、C++、.NET、Node.js等编程语言。只要支持原生MySQL的编程语言都可以直接使用PolarDB MySQL, 详情请参见[MySQL官网](#)。

- Q: 支持哪些存储引擎?

A: PolarDB MySQL全部表均使用InnoDB存储引擎。创建表的时候, PolarDB MySQL会自动将非InnoDB引擎 (如 MyISAM、Memory、CSV 等) 转换为InnoDB 引擎, 因此即使迁移之前的数据表不是InnoDB, 也仍然能够正常迁移至PolarDB MySQL。

- Q: 是否支持自建Slave实例, 是否有推荐的实现方式?

A: 支持。启用Binlog后可以将PolarDB MySQL同步到其他MySQL库, 构成Master-Slave架构。为方便后续维护, 建议您使用数据传输服务DTS (Data Transmission Service), 关于如何使用DTS实现同步, 请参见[从PolarDB MySQL同步至RDS MySQL](#)。

兼容性

- Q: 是否兼容社区版MySQL?

A: PolarDB MySQL可以100% 兼容社区版MySQL。

- Q: 支持哪些事务隔离级别?

A: PolarDB MySQL支持READ_UNCOMMITTED、READ_COMMITTED（默认）、REPEATABLE_READ这三种隔离级别，不支持SERIALIZABLE隔离级别。

- Q: SHOW PROCESSLIST与社区版MySQL是否存在差异?

A: 如果是通过主地址查询，两者没有区别。但如果是通过集群地址查询，略有差异，此时会出现有多条相同Thread ID的记录，分别对应PolarDB MySQL集群中的每一个节点。

- Q: 锁机制和社区版MySQL是否存在差异?

A: PolarDB MySQL会将DDL中涉及到的Exclusive MDL锁同步到读节点上（同样通过redo日志），并且使读节点持有MDL锁直到DDL操作结束，来阻止读节点上其它用户线程在DDL执行过程中访问表数据。与社区版MySQL不同，PolarDB MySQL的主节点和读节点是基于共享存储的，这会导致主节点在做DDL的时候，读节点可能会查询到DDL过程中的中间数据而出现错误。

- Q: Binlog格式和MySQL原生格式是否存在差异?

A: 没有差异。

- Q: 是否支持performance schema和sys schema?

A: 支持。

- Q: 表统计信息收集和社区版MySQL是否存在差异?

A: PolarDB MySQL主节点的表统计信息和社区版MySQL一致。为了保证主节点和只读节点执行计划的一致性，主节点每次更新统计信息时，会同步到只读节点。此外，只读节点还可以通过ANALYZE TABLE操作，主动从磁盘加载最新的统计信息。

- Q: PolarDB是否支持XA事务，和官方MySQL是否存在差异?

A: 支持，没有差异。

- Q: PolarDB是否支持全文索引?

A: 支持。



说明:

目前，用户使用全文索引时，只读节点存在一定的索引缓存数据延迟，建议读写全文索引的操作都使用主地址，以读到最新的数据。

- Q: 是否支持Percona工具集?

A: 支持，但是建议您使用online DDL。

- Q: 是否支持gh-ost?

A: 支持，但是建议您使用online DDL。

费用

- Q: PolarDB的费用都包含哪些?

A: 包含存储空间、计算节点、备份（附赠免费额度）、SQL洞察（可选），详情请参见[#unique_67](#)。

- Q: 收费的存储空间都包含哪些内容?

A: 包含数据库表文件、索引文件、undo日志文件、redo日志文件、binlog文件、slowlog文件及少量的系统文件，详情请参见[#unique_67](#)。

- Q: PolarDB的存储包怎么用?

A: 购买的包年包月或按量付费的集群，均可使用存储包抵扣存储费用。例如您有3个存储容量均为40GB的集群（即总容量为120GB），这3个集群可以共享一个100GB的存储包，多出的20GB则按量计费，详情请参见[使用存储包](#)。

集群访问（读写分离）

- Q: 如何实现PolarDB的读写分离?

A: 只需在应用程序中使用集群地址，即可根据配置的读负载节点实现读写分离，详情请参见[自定义集群地址](#)。

访问信息 ①

白名单 ②

白名单列表  [配置](#)

主地址 ②

私网  [修改](#)

公网 [申请](#)

集群地址【推荐】 ② [新增自定义地址](#)

默认地址 () [修改配置](#)

读写模式 可读可写（自动读写分离）

私网  [修改](#)

公网 [申请](#)

[> 节点配置](#)

[> 高级配置](#)

- Q: 多个只读节点间负载不均衡的原因是什么?

A: 只读节点间负载不均衡的原因有只读节点连接数较少、自定义集群地址分配时未包括某个只读节点等。

- Q: 造成主节点负载高或低的原因是什么?

A: 造成主节点（主库）负载高的原因有直连主地址、主库接受读请求、存在大量的事务请求、主从复制延迟高导致请求被路由到主库、只读节点异常导致读请求被路由到主库等。

而主节点负载较低的原因可能是主库开启了不接受读选项。

- Q: 怎么降低主节点的负载?

A: 您可以参考如下建议降低主节点负载:

- 使用集群地址来连接PolarDB集群，详情请参见[修改和释放集群地址](#)。
- 如果由于事务较多导致主节点压力大，您可以通过控制台打开事务拆分功能，把事务中的部分查询路由到只读节点，详情请参见[高级选项-事务拆分](#)。
- 如果由于复制延迟导致请求被路由到主库，您可以考虑降低一致性等级（如使用最终一致性），详情请参见[一致性级别](#)。
- 主库接受读请求，可能也会导致主库负载高，您可以通过控制台关闭主库接受读功能，减少读请求被路由到主库，详情请参见[主库不接受读](#)。

- Q: 为什么读不到刚插入的数据?

A: 该问题可能是由于一致性级别的配置导致的，PolarDB MySQL的集群地址支持如下三种一致性级别：

- 最终一致性：不论是同一会话（连接）或不同会话，最终一致性都不保证读能够马上读到刚插入的数据。
- 会话一致性：一定能够读到同一会话插入之后的读数据。
- 全局一致性：保证同一会话和不同会话都能够读到最新数据。



说明：

一致性等级越高，性能越差，对主库的压力越大，请谨慎选择。对于大多数应用场景会话一致性能够保证业务正常工作，对于少数有强一致性的需求的语句，可以通过Hint `/* FORCE_MASTER */`来实现，详情请参见[PolarDB一致性级别](#)。

- Q: 如何强制SQL到主节点执行?

使用集群地址时, 在SQL语句前加上`/* FORCE_MASTER */`或`/* FORCE_SLAVE */`, 即可强制指定这条SQL的路由方向。详情请参见[Hint语法](#)。

- `/* FORCE_MASTER */`强制请求被路由到主库。该用法可以用于解决少数一致性要求较高的读请求的场景。
- `/* FORCE_SLAVE */`强制请求被路由到从库。该用法可以用于解决少数由于保证正确性要求特殊语法被路由到主库的场景, 比如存储过程的调用、`multistatement`的使用等。

- Q: 是否可以给不同的业务分配不同的地址? 不同地址间是否可以达到隔离的效果?

A: 您可以创建多个自定义地址给不同的业务使用, 若底层节点不同则自定义地址间可同时具备隔离的效果, 不会互相影响。关于如何创建自定义地址, 详情请参见[设置集群地址](#)。

- Q: 如果有多个只读节点, 如何为其中某个只读节点单独创建单节点地址?

A: 仅当集群地址读写模式为只读且集群内拥有三个及以上节点时, 才支持创建单节点地址, 详细操作步骤请参见[设置集群地址](#)。

**警告:**

创建单节点地址后, 当此节点故障时, 该地址可能会有最多1小时的不可用, 请勿用于生产环境。

- Q: 一个集群内最多允许创建多少个单节点地址?

A: 如果您的集群内有3个节点, 则只允许为其中1个只读节点创建单节点地址; 若集群内有4个节点, 则允许为其中2个只读节点创建各自的单节点地址, 以此类推。

- Q: 只用了主地址, 但是发现只读节点也有负载, 是否主地址也支持读写分离?

A: 主地址不支持读写分离, 始终只连接到主节点。只读节点有少量QPS是正常现象, 与主地址无关。

管理与维护

- Q: 如何在线加字段?

A: 支持原生自带的online DDL、`pt-osc`和`gh-ost`等工具, 建议您使用自带的online DDL操作。

- Q: 如何在线加索引?

A: 支持原生自带的online DDL、`pt-osc`和`gh-ost`等工具, 建议您使用自带的online DDL操作。

- Q: 主节点(主)与只读节点(备)是否存在复制延迟?

A: 是, 它们之间存在毫秒级延迟。

- Q: 什么情况下会导致复制延迟增大?

A: 出现如下情况时会致复制延迟增大:

- 主节点写入负载高, 产生了过多的redo日志, 导致只读节点来不及应用。
- 只读节点负载过高, 抢占了过多原本属于应用redo日志的资源。
- I/O出现瓶颈, 导致读写redo日志过慢。

- Q: 存在复制延迟的情况下, 如何保证查询的一致性?

A: 您可以使用集群地址并为其选择合适的一致性级别。目前一致性从高到低分别为全局一致性(强一致性)、会话一致性和最终一致性, 详情请参见[设置集群地址](#)。

- Q: 单节点故障的情况下是否可以保证RPO为0?

A: 可以。

- Q: 升级规格配置(比如从2核8GB升级到4核16GB)后端是怎么实现的? 对业务有什么影响?

A: PolarDB MySQL的代理(Proxy)和数据库节点(Node)均需要升级到最新的配置, 采用多个节点滚动升级的方式尽量减少对业务的影响。目前每次升级大概需要10~15分钟, 对业务的影响时间不超过30秒, 期间可能会产生1~3次连接闪断, 详情请参见[变更配置](#)。

- Q: 添加节点要多久? 是否会影响业务?

A: 每增加一个节点需要5分钟, 对业务无影响。关于如何添加节点, 详情请参见[增加只读节点](#)。



说明:

新增只读节点之后新建的读写分离连接会转发请求到该只读节点。新增只读节点之前建立的读写分离连接不会转发请求到新增的只读节点, 需要断开该连接并重新建立连接, 例如, 重启应用。

- Q: 升级到最新版本(小版本升级)要多久? 是否会影响业务?

A: PolarDB MySQL采用多节点滚动升级的方式尽量减少对业务的影响, 目前每次升级大概需要10~15分钟, 对业务的影响时间不超过30秒, 期间可能会产生1~3次连接闪断, 详情请参见[小版本升级](#)。

备份与恢复

- Q: 数据库恢复的速度如何?

A: 目前, 基于备份集(快照)进行恢复(克隆)的速度是40分钟/TB。如果是恢复到任意时间点, 则需要包含应用redo日志的时间, 这部分的恢复速度大概是20~70秒/GB, 整个恢复时间是这两部分之和。

性能和容量

- Q：为什么PolarDB MySQL相比RDS MySQL性能提升不明显？

A：在您对PolarDB和RDS进行性能对比前，请了解以下注意事项，以便能获得比较准确、合理的性能对比结果。

- 使用相同规格配置的PolarDB和RDS进行性能对比。
- 使用相同版本的PolarDB和RDS进行性能对比。

因为不同版本的实现机制不一样，例如MySQL 8.0针对多核数CPU做优化，单独抽象出来Log_writer、log_flusher、log_checkpoint、log_write_notifier等线程，但在CPU核数较少的情况下性能则不如MySQL 5.6/5.7。不推荐使用PolarDB MySQL 5.6和RDS MySQL 5.7/8.0进行对比，因为MySQL 5.6的优化器比较旧，不如新版本。

- 推荐使用模拟线上压力的场景进行实际性能对比，或者使用sysbench进行对比，这样获得的数据更接近线上实际场景。
- 在对比读性能的时候，不推荐您使用单条SQL进行比较。

因为PolarDB是计算存储分离的架构，所以单条语句有网络延迟的影响，导致读性能不如RDS。但是线上数据库的缓存命中率基本都在99%以上，因此只有第一次的读会调用I/O，因此读取性能会降低，后续数据都在缓存池（Buffer Pool）中，并不需要调用I/O，因此性能是一样的。

- 在对比写性能的时候，同样不推荐您使用单条SQL进行比较，推荐模拟线上环境进行压力测试。

如果要对比RDS性能，请使用PolarDB（主节点+只读节点）和RDS（主实例+半同步的只读实例）进行对比。这是因为PolarDB的架构在写入数据的时候默认采用Quorum机制，即写入数据时默认写入到三副本里面的大多数（在三个副本中的两个或两个以上写入成功，就认为写操作成功了）。PolarDB已经在存储层面做数据冗余，并保证三副本强同步高可靠，使用RDS MySQL的半同步复制（而不是异步复制）进行对比更合理。

PolarDB MySQL与RDS MySQL的性能对比结果，请参见[#unique_69](#)。

- Q：删除数据库后为什么还是占用很多空间？

A：这是由于redo日志文件占用了空间，通常在2GB~11GB左右，最多时会占用11GB，其中包括缓冲池中8个redo日志（8G）、正在写的redo日志（1GB）、提前创建的redo日志（1GB）以及最后一个redo日志（1GB）。

缓冲池内的redo日志文件数量由参数`loose_innodb_polar_log_file_max_reuse`控制，默认值为8。您可以修改这个参数从而减少日志空间占用量，但在压力大的情况下，性能可能会出现周期性的小幅波动。



说明：

调整参数`loose_innodb_polar_log_file_max_reuse`后，缓冲池不会立刻被清空，随着DML被执行，缓冲池内的数据量才会慢慢减少。如果需要立即清空，请[提交工单](#)联系售后服务。

参数配置	<code>log_throttle_queries_not_using_indexes</code> ⓘ	0	否	0	[0-4294967295]
诊断与优化	<code>long_query_time</code> ⓘ	1	否	1	[0.03-10]
集群总览	<code>loose_expire_logs_hours</code> ⓘ	336	否	336	[0-2376]
性能监控	<code>loose_innodb_polar_log_file_max_reuse</code> ⓘ	8	否	8	[0-100]
问题分析	<code>loose_innodb_primary_abort_ddl_wait_replica_timeout</code> ⓘ	3600	否	3600	[1-31536000]
慢SQL					

- Q：表个数上限是多少？表个数到多少时有可能引起性能下降？

A：表个数的上限受文件数量限制，详情请参见[#unique_70](#)。

- Q：IOPS是怎么限制和隔离的？是否会出现多个PolarDB集群的I/O争抢？

A：PolarDB MySQL集群的每个节点根据规格大小设置IOPS，每个节点之间IOPS独立隔离，互不影响。

- Q：只读节点的性能变慢是否会影响主节点？

A：只读节点的负载过高、复制延迟增高时，可能会少量增加主节点的内存消耗。

- Q：打开Binlog之后，对性能有什么影响？

A：不影响SELECT查询性能，但会对INSERT、UPDATE、DELETE等写操作有10~30%的性能影响。一般情况下，数据库内部90%以上都是SELECT查询操作，因此整体来看性能影响在5%以内。

- Q：打开SQL洞察（全量SQL日志审计），对性能有什么影响？

A：无影响。

大表问题

- Q: 对比传统本地盘的数据库, PolarDB MySQL中的大表存储有什么优势?

A: PolarDB MySQL中的一张表, 物理上会被拆分到N台存储服务器上存储, 因此对一张表的I/O会被分摊到多块存储磁盘中, I/O读取的整体吞吐性能(而不是I/O延迟)要远优于集中式的本地盘数据库。

- Q: 大表如何优化?

A: 推荐使用分区表。

- Q: 哪种情况下适合使用分区表?

A: 需要通过裁剪大表来控制查询访问的数据量并且希望该裁剪对业务代码透明(无需修改业务代码)的场景下, 适合使用分区表。例如您可以通过使用分区表来定期清理业务历史数据(如删除最早的一个月份分区并新建下个月份分区, 实现只保留最近6个月份数据的效果)。

稳定性

- Q: 是否可以对高并发下的PHP短连接进行优化?

A: 可以。在集群地址中, 可以通过开启会话级连接池进行优化, 详情请参见[修改集群地址](#)。

- Q: 如何规避个别执行效率低下的SQL拖垮整个数据库?

A: 如果您的PolarDB MySQL是8.0版本, 您可以使用语句并发控制(Statement Concurrency Control)特性来实现针对指定语句的限流, 关于如何使用语句并发控制, 请参见[#unique_71](#)。

- Q: PolarDB是否支持空闲会话超时?

A: 支持。您可以通过修改`wait_timeout`参数来自定义空闲会话的超时时间, 具体操作步骤请参见[设置集群参数](#)。

- Q: 如何发现慢SQL?

A: 您可以通过如下两种方式发现慢SQL:

- 直接在控制台上查询慢SQL, 详情请参见[慢SQL](#)。
- 连接数据库集群后执行`show processlist;`找出执行时间过长的SQL, 关于如何连接数据库集群, 请参见[连接数据库集群](#)。

```
mysql> show processlist;
+-----+-----+-----+-----+-----+-----+-----+-----+
| Id    | User | Host                                | db  | Command | Time | State | Info                                |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 33554490 | acc | 193.50.135.193:19358 | NULL | Query   | 0    | starting | show processlist |
| 33554499 | acc | 193.50.135.193:19358 | NULL | Query   | 250  | User sleep | select sleep(600) |
| 33554499 | acc | 193.50.135.193:19358 | NULL | Sleep   | 253  |         | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set, 13312 warnings (0.00 sec)
```


- Q：如何终止慢SQL？

A：发现慢SQL后，您可以先查看慢SQL的ID，然后执行kill <Id>终止慢SQL。

```
mysql> kill 33554499;  
Query OK, 0 rows affected (0.01 sec)
```

5 数据迁移或同步

5.1 数据迁移/同步方案概览

云数据库PolarDB提供了多种数据迁移同步方案，可满足不同上云、迁云、同步的业务需求，使您可以在不影响业务的情况下平滑将数据库迁移、同步至阿里云云数据库PolarDB上面。

通过使用阿里云[数据传输服务（DTS）](#)，您可以实现PolarDB的结构迁移、全量迁移和实时同步。

数据迁移

使用场景	文档链接
从RDS迁移至PolarDB	一键升级RDS MySQL至PolarDB MySQL（平滑迁移，推荐） 一键克隆RDS MySQL至PolarDB MySQL 从RDS MySQL迁移至PolarDB MySQL
从PolarDB迁移至RDS	从PolarDB MySQL迁移至RDS MySQL
从自建数据库迁移至PolarDB	从自建MySQL迁移至PolarDB MySQL 从本地MySQL迁移至PolarDB MySQL
从第三方云数据库迁移至PolarDB	从Amazon Aurora MySQL迁移至PolarDB MySQL
PolarDB之间的数据迁移	PolarDB MySQL集群间的数据迁移

数据同步

使用场景	文档链接
从RDS同步至PolarDB	从RDS MySQL同步至PolarDB MySQL
从PolarDB间的同步	PolarDB MySQL集群间的单向同步 PolarDB MySQL集群间的双向同步
从自建数据库同步至POLARDB	从ECS上的自建MySQL同步至PolarDB MySQL
从PolarDB同步至RDS	从PolarDB MySQL同步至RDS MySQL
从PolarDB同步至分析型数据库	从PolarDB MySQL同步至AnalyticDB for MySQL 从PolarDB MySQL同步至云原生数据仓库AnalyticDB PostgreSQL
从PolarDB同步至Datahub	从PolarDB MySQL同步至Datahub
从PolarDB同步至Kafka	从PolarDB MySQL同步到Kafka

5.2 【注意事项】MySQL 5.7迁移到PolarDB MySQL 8.0

PolarDB MySQL 8.0完全兼容MySQL 5.7，您可以将MySQL 5.7数据库迁移到PolarDB MySQL 8.0使用，数据不会丢失，但需要注意客户端版本和PolarDB MySQL 8.0的兼容性问题。



说明：

kickout是PolarDB MySQL 8.0的保留关键字。因此，若您已经在MySQL 5.7或开源MySQL 8.0上使用该关键字作为对象名称（如表名、字段名、存储过程名等），在迁移到PolarDB MySQL 8.0前，请您先修改对象名称避免使用该关键字。否则迁移时，将会出现错误码为1064的语法报错。

如何将MySQL 5.7迁移至PolarDB MySQL 8.0请参见如下文档：

- [从RDS MySQL迁移至PolarDB MySQL](#)
- [#unique_90](#)
- [从本地MySQL迁移至PolarDB MySQL](#)
- [#unique_91](#)
- [#unique_92](#)

客户端版本

您需要将MySQL客户端程序升级到如下版本：



说明：

若原MySQL 5.7用户由于各种原因确实不能（或不方便）升级客户端驱动以适配PolarDB MySQL 8.0，可以考虑使用PolarDB MySQL 5.6代替。前提是：

- 没有使用JSON数据类型。
- 切换前进行详细的性能测试。
- Java：MySQL Connector/J 8.0及以上版本。
- ODBC：MySQL Connector/ODBC 8.0及以上版本。
- CPP：MySQL Connector/Cpp 8.0及以上版本。
- .NET：MySQL Connector/NET 8.0及以上版本。
- Nodejs：MySQL Connector/Nodejs 8.0及以上版本。
- Python：MySQL Connector/Python 8.0及以上版本。
- Python：mysql-connector-python 8.0.5及以上版本。
- Golang：go-sql-driver/mysql 1.4.0及以上版本。
- PHP：mysqlnd 7.4及以上版本。
- C/Cpp：libmysqlclient 8.0及以上版本。

已知客户端问题

- 问题现象：MySQL数据库连接异常，`query_cache_size`无法识别。
 - Driver版本：mysql-connector-java:5.1.42
 - 数据库版本：mysql 8.0.13
 - 解决办法：使用mysql-connector-java:5.1.42以上的版本，该版本相关更新日志请参见[Changes in MySQL Connector/J 5.1.43](#)。
- 问题现象：mysql python driver由于COM_STMT_EXECUTE的flag没有设置正确且没有发送`com_stmt_fetch`获取结果集，在MySQL 8.0版本上会导致无法正常获取返回结果，在MySQL 5.6/5.7版本可以正常返回结果。
 - Driver版本：mysql-connector-2.2.9。
 - 数据库版本：mysql 8.0.13。
 - 解决办法：安装8.0的[python driver](#)。

5.3 从RDS迁移至PolarDB

5.3.1 从RDS迁移到PolarDB后可申请无责退款

您完成业务从RDS迁移到PolarDB后，若确定业务已在PolarDB上稳定运行且不再需要RDS时，您可以申请RDS无责退款，避免浪费闲置的RDS资源。

前提条件

- RDS实例的付费类型需为**包年包月（预付费）**，且数据库引擎需为MySQL。
- 目标PolarDB集群的创建方式需为**从RDS迁移**，且该PolarDB集群的包年包月（含按量转包年包月）订单金额需高于源RDS的退款金额。

退款规则

根据源RDS实例的剩余时长按比例退款。

假设源RDS实例的购买时长为一年，订单金额为10000元。使用半年后若您从RDS迁移至PolarDB并申请了RDS无责退款，则您可以获得5000元的退款金额，退款金额将原路返回。

退款申请

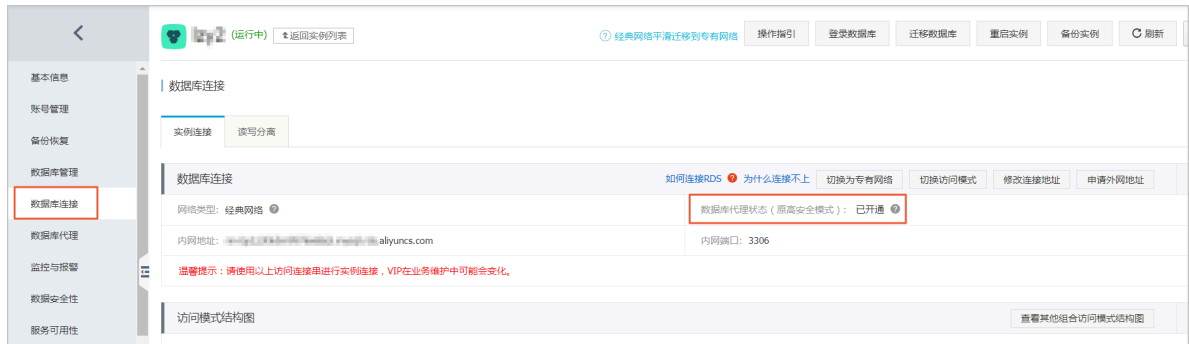
请[提交工单](#)联系售后客服申请RDS无责退款。

5.3.2 一键升级RDS MySQL至PolarDB MySQL

PolarDB支持将RDS MySQL一键升级至PolarDB MySQL。

前提条件

- 源RDS实例版本为RDS MySQL 5.6高可用版。
- 源RDS实例未开启TDE和SSL。
- 源RDS实例的表存储引擎为InnoDB。
- 如果RDS处于高安全模式（数据库代理模式），需要创建有高权限账号（请参见[#unique_95/unique_95_Connect_42_section_wkq_j35_q2b](#)），或者切换到高性能模式（参见[#unique_96](#)），才能进行一键升级。



背景信息

PolarDB是阿里云自研的下一代关系型云数据库，主要优势如下：

- 存储容量高：最高可达100TB。
- 性能高：最高可以提升至MySQL的6倍。
- Serverless存储：存储容量无需提前购买，自动扩缩容，按使用量计费。
- 临时升配：临时升级规格，轻松应对短期的业务高峰。

详情请参见[#unique_97](#)。

一键升级功能可以将RDS MySQL一键升级为PolarDB MySQL，升级后PolarDB集群包含源RDS实例的账号、数据库、IP白名单和必要的参数。

无责退款

您完成业务从RDS迁移到PolarDB后，若确定业务已在PolarDB上稳定运行且不再需要RDS时，您可以申请RDS无责退款，避免浪费闲置的RDS资源，详情请参见[从RDS迁移到PolarDB后可申请无责退款](#)。

一键升级的功能亮点

- 迁移完全免费。
- 迁移过程数据0丢失。
- 支持增量迁移，停机时间小于10分钟。
- 支持回滚，迁移失败可以在10分钟内恢复。
- 支持带地址切换，可以在不修改应用程序的连接配置情况下切换到PolarDB。

迁移流程

1. 参见[从RDS迁移](#)的说明，创建一个与源RDS实例数据相同的PolarDB集群，源RDS实例的增量数据会实时同步到该PolarDB集群。



说明：

需要在7天内修改应用端的数据库地址为PolarDB地址，确认业务正常，以及单击**完成迁移**。单击**完成迁移**会中断RDS和PolarDB之间的数据同步。

2. 单击**迁移切换**。该操作将源RDS实例修改为只读，将PolarDB集群修改为可读可写，PolarDB的增量数据会实时同步到RDS。修改数据库连接地址。具体操作请参见[迁移切换](#)。



说明：

迁移切换后，也可以选择[迁移回滚](#)。

3. [完成迁移](#)。

注意事项

- 迁移只能在相同地域内进行。
- 源RDS实例在迁移时不能修改参数。

从RDS迁移

本操作将创建一个与源RDS实例数据相同的PolarDB集群，源RDS实例的增量数据会实时同步到该PolarDB集群。

1. 登录[PolarDB控制台](#)。
2. 单击**创建新集群**。
3. 选择**包年包月**或**按量付费**页签。
4. 设置以下参数。

参数	说明
地域	源RDS MySQL实例所在地域。 说明： 新建的PolarDB集群也在此地域。
创建方式	选择 从RDS迁移 。即从RDS实例克隆一个PolarDB集群，同时保持数据同步。默认开启新集群的Binlog。
源RDS引擎	源RDS实例的引擎类型，不可变更。
源RDS版本	源RDS实例的版本，不可变更。

参数	说明
源RDS实例	可选的源RDS实例，不包括只读实例。
可用区	可用区是地域中的一个独立物理区域，不同可用区之间没有实质性区别。 您可以选择将PolarDB集群与ECS创建在同一可用区或不同的可用区。
网络类型	PolarDB集群的网络类型，不可变更。
VPC网络 VPC交换机	PolarDB集群所属的VPC和虚拟交换机。请确保PolarDB集群与需要连接的ECS创建于同一个VPC，否则它们无法通过内网互通，无法发挥最佳性能。
兼容性	PolarDB集群的数据库引擎，不可变更。
节点规格	按需选择，建议不低于源RDS实例规格。所有PolarDB节点均为独享型，性能稳定可靠。详情请参见 #unique_67 。
节点个数	无需选择。系统将自动创建一个与主节点规格相同的只读节点。
存储费用	无需选择容量，根据实际数据使用量按小时计费。详情请参见 #unique_67 。

5. 设置**购买时长**（仅针对包年包月集群），然后单击右侧的**立即购买**。

6. 确认订单信息，阅读和勾选**服务协议**，单击**去开通**，完成支付。

7. 进入[PolarDB控制台](#)，查看新建的PolarDB集群的状态。



说明：

- 集群创建后开始从RDS实例同步数据，7 天内需要完成修改数据库连接地址以及[完成迁移](#)操作。超过7天将自动关闭迁移功能。
- 您可以在此步骤选择取消迁移，相关影响请参见[迁移常见问题](#)。

迁移切换

满足以下条件后，您可以进行迁移切换，然后修改应用里的数据库连接地址。

- 已完成[从RDS迁移](#)的操作。
- 复制延迟**小于60秒。

RDS迁移功能				迁移切换	取消迁移
源RDS	rm- 	状态	● 数据同步中	源RDS读写状态	读写
PolarDB读写状态	只读	复制延迟	0		
复制关系	从RDS (rm-) 到PolarDB (pc-) 同步				
复制到截止时间	2020年5月29日 15:11:29 (剩余时长 6天23小时57分36秒)				
	到期后，将关闭迁移功能，源和目标数据库均恢复【读写】状态，脱离复制关系。				

1. 进入[PolarDB控制台](#)。

2. 找到目标集群，单击集群的ID。
3. 在**基本信息**页面单击**迁移切换**，在弹出的对话框中单击**确定**。本操作将源RDS实例修改为只读，将PolarDB集群修改为可读可写，同时会将PolarDB集群的新增数据同步到RDS实例。



说明：

- 数据同步的延迟超过60秒时无法进行迁移切换。
- 切换过程一般小于5分钟。

4. 在**开始切换**对话框中，选择**带地址切换（应用程序不用改连接配置）**或**不带地址切换（应用程序需要改为新的PolarDB连接配置）**。

开始切换

×

切换业务到新数据库。

将交换RDS MySQL实例 rm- 和 该PolarDB集群的读写状态，同时更换复制方向。

☒ 带地址切换（应用程序不用改连接配置） ?

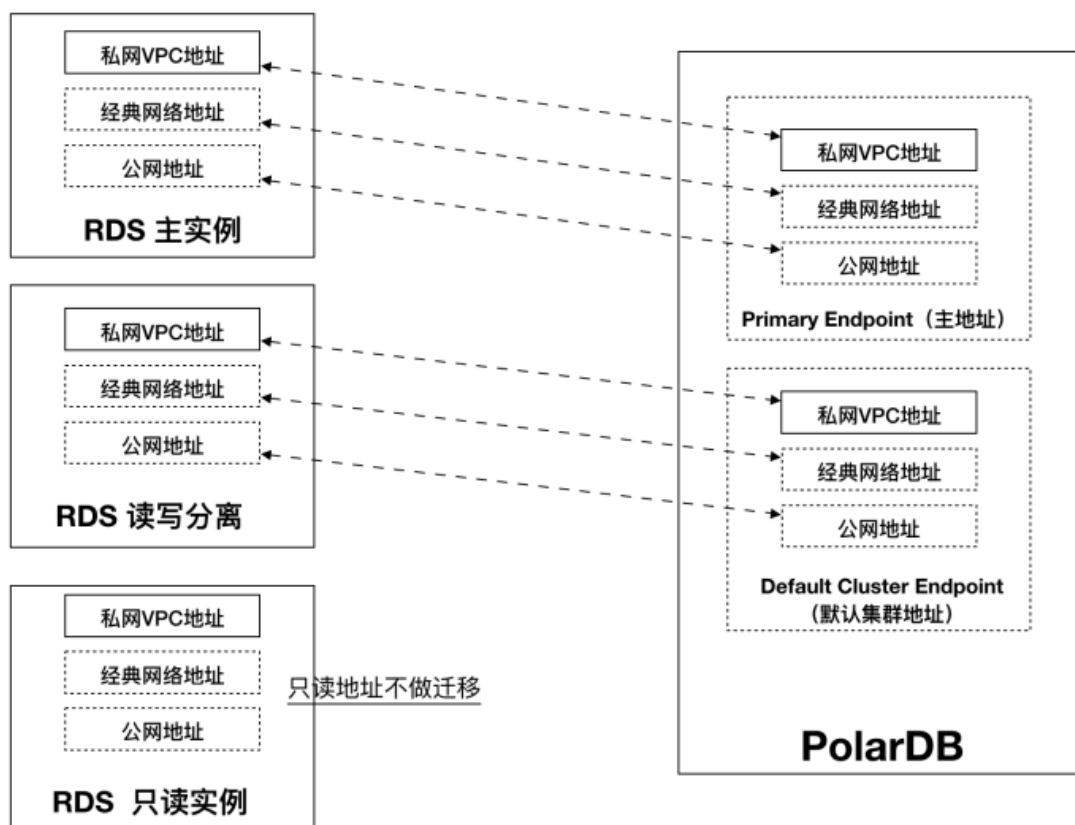
	RDS	POLARDB
主地址	rm- 3...	pc- 4...
读写分离/集群地址	rm- 3...	pc- 4...

☐ 不带地址切换（应用程序需要改为新的PolarDB连接配置） ?

确定

取消

如果您不方便修改应用程序的连接配置，可以选择**带地址切换（应用程序不用改连接配置）**。选择**带地址切换**后，RDS连接地址对应的PolarDB连接地址如下图所示。

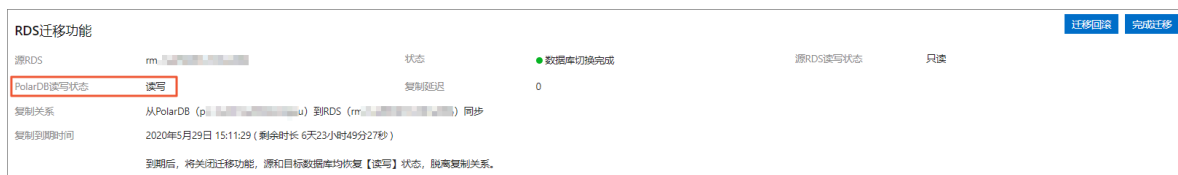


注意：

- 带地址切换只会切换RDS和PolarDB的域名，Vswitch、Vip等配置不会切换。
- 带地址切换不会切换端口，请确保RDS和PolarDB的连接端口一致，修改RDS端口请参见[#unique_103](#)。
- 域名交换后，可能会存在DNS解析缓存问题，在缓存过期时间内可能会连接不上数据库，建议您刷新一下服务器的DNS缓存。
- 切换域名后，如果您需要使用DMS登录PolarDB数据库，必须使用新版本的DMS并且使用集群ID来进行登录，连接串无法登录。

5. 单击**确定**。

6. 刷新页面，当**PolarDB读写状态**显示为**读写**后，尽快修改应用里的数据库连接地址。



说明：

迁移切换完成后，也可以选择[迁移回滚](#)。

完成迁移

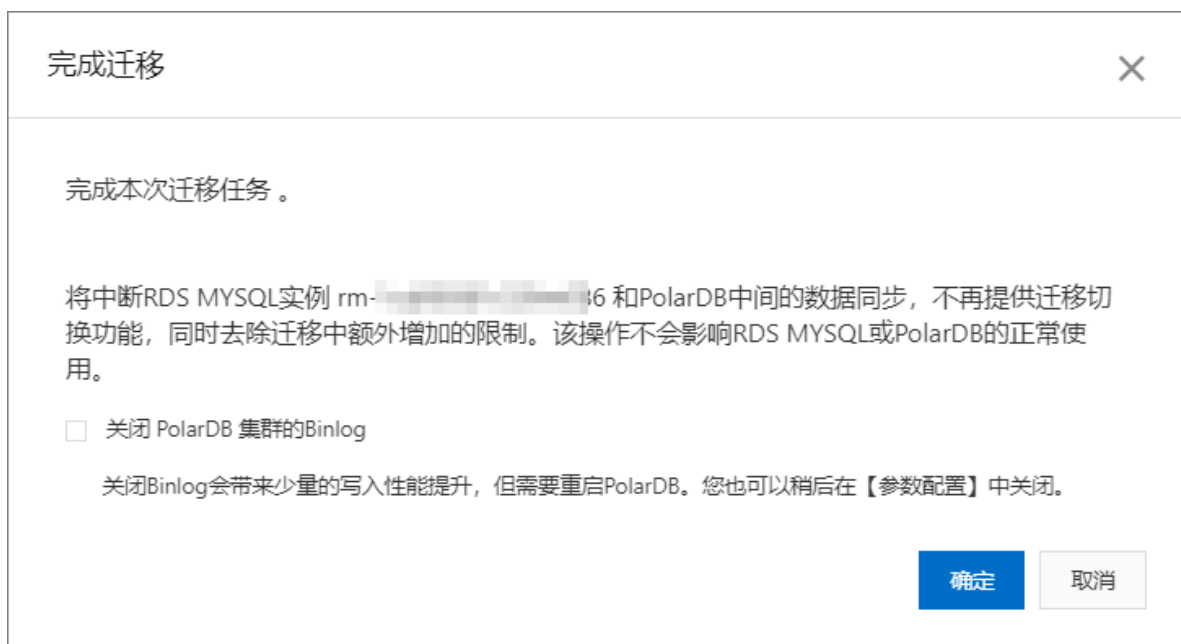
从RDS迁移后，需要在7天内修改数据库连接地址以及单击**完成迁移**。该操作将中断PolarDB集群和RDS实例间的数据同步。



警告：

由于本操作将中断PolarDB集群和RDS实例间的数据同步，不再提供[迁移回滚](#)功能，建议您使用一段时间PolarDB集群，确认正常后再执行本操作。

1. 进入[PolarDB控制台](#)。
2. 找到目标集群，单击集群的ID。
3. 在**基本信息**页面，单击**完成迁移**，在弹出的对话框中单击**确定**。



说明：

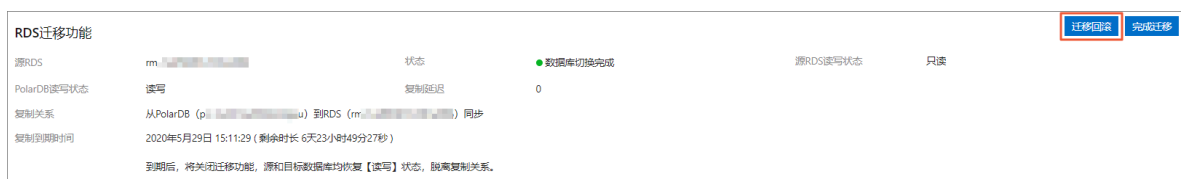
- 单击**确定**后，系统将在约2分钟内中断同步关系，期间**完成迁移**按钮不会消失，请勿重复单击。
- 您可以选择是否关闭PolarDB集群的Binlog。关闭Binlog会带来少量的写入性能提升，但需要重启PolarDB。

4. 如果不再需要源RDS实例，可以释放实例。

迁移回滚

在完成迁移前，如果您发现数据存在异常等问题，可以进行回滚操作，快速恢复至迁移前的状态（RDS实例为可读可写，PolarDB集群为只读，同时会将RDS实例的数据同步到PolarDB集群）。

1. 进入[PolarDB控制台](#)。
2. 找到目标集群，单击集群的ID。
3. 在**基本信息**页面单击**迁移回滚**，在弹出的对话框中单击**确定**。



说明：

单击**确定**后RDS实例为可读可写，PolarDB集群为只读，同时会将RDS实例的数据同步到PolarDB集群。当**源RDS读写状态**显示为**读写**后，请尽快修改应用里的数据库连接地址为RDS连接地址。

迁移常见问题

- 从RDS迁移会影响源RDS实例吗？

答：不会影响源RDS实例的正常运行。

- 平滑迁移对业务有影响吗？

答：平滑迁移能够保证迁移过程不丢失数据，停机时间小于10分钟，如果有需要还可以进行回滚。

- 取消迁移会有什么影响？

答：取消迁移后，源RDS实例可以修改参数；PolarDB集群恢复可读可写，且数据不会释放。手动取消时可以选择是否关闭PolarDB集群的Binlog，自动取消时不会关闭。

相关API

API	描述
#unique_104	创建PolarDB集群。  说明： 一键升级时，参数 CreationOption 取值需要为 MigrationFromRDS 。

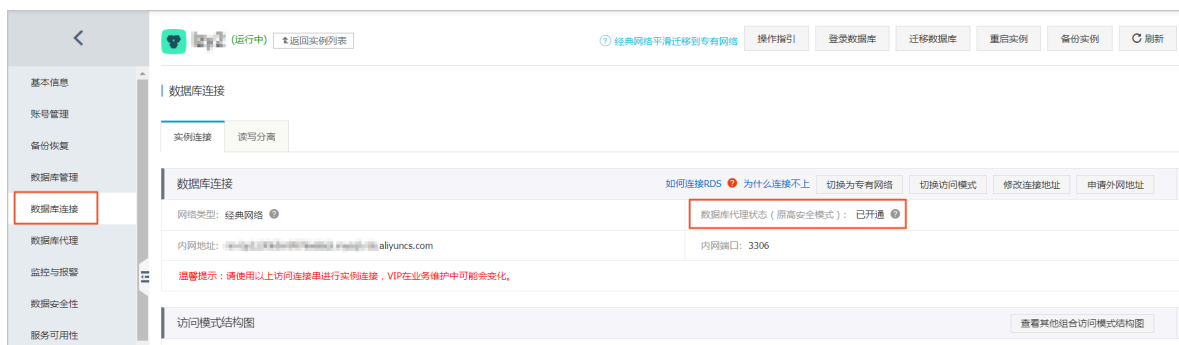
API	描述
#unique_105	查询PolarDB集群的迁移状态。
#unique_106	修改迁移任务，进行任务的切换或回滚。
#unique_107	取消或完成迁移。

5.3.3 一键克隆RDS MySQL至PolarDB MySQL

PolarDB支持从RDS MySQL一键克隆数据至新的PolarDB MySQL集群。

前提条件

- 源RDS实例版本为RDS MySQL 5.6高可用版。
- 源RDS实例未开启TDE和SSL。
- 源RDS实例的表存储引擎为InnoDB。
- 如果RDS处于高安全模式（数据库代理模式），需要创建有高权限账号（请参见[#unique_95/unique_95_Connect_42_section_wkq_j35_q2b](#)），或者切换到高性能模式（参见[#unique_96](#)），才能进行一键克隆。



背景信息

PolarDB是阿里云自研的下一代关系型云数据库，主要优势如下：

- 存储容量高：最高可达100TB。
- 性能高：最高可以提升至MySQL的6倍。
- Serverless存储：存储容量无需提前购买，自动扩缩容，按使用量计费。
- 临时升配：临时升级规格，轻松应对短期的业务高峰。

详情请参见[#unique_97](#)。

一键克隆功能将会新建一个与源RDS实例的数据相同的PolarDB集群，PolarDB集群包含源RDS实例的账号、数据库、IP白名单和必要的参数。源RDS实例的增量数据不会同步到PolarDB集群。



说明：

如果需要在新建PolarDB集群的同时，使源RDS实例的增量数据实时同步到PolarDB集群，即实现平滑迁移（不停机迁移），请参见[#unique_108](#)。

一键克隆的功能亮点

- 免费
- 克隆过程数据0丢失

一键克隆的操作步骤

1. 登录[PolarDB控制台](#)。
2. 单击**创建新集群**。
3. 选择**包年包月**或**按量付费**页签。
4. 设置以下参数。

参数	说明
地域	源RDS MySQL实例所在地域。  说明： 克隆的PolarDB集群也在此地域。
创建方式	集群的创建方式： • 默认创建：创建一个全新的PolarDB集群。 • 从RDS克隆：基于所选的RDS实例，克隆一个数据完全一样的PolarDB集群。 • 从RDS迁移：先从RDS实例克隆一个PolarDB集群，同时保持同步。默认开启新集群的Binlog。 这里选择 从RDS克隆 。
源RDS引擎	源RDS实例的引擎类型，不可变更。
源RDS版本	源RDS实例的版本，不可变更。
源RDS实例	可选的源RDS实例，不包括只读实例。
主可用区	可用区是地域中的一个独立物理区域，不同可用区之间没有实质性区别。 您可以选择将PolarDB集群与ECS创建在同一可用区或不同的可用区。
网络类型	PolarDB集群的网络类型，不可变更。
VPC网络 VPC交换机	PolarDB集群所属的VPC和虚拟交换机。请确保PolarDB集群与需要连接的ECS创建于同一个VPC，否则它们无法通过内网互通，无法发挥最佳性能。

参数	说明
数据库引擎	PolarDB集群的数据库引擎，不可变更。
节点规格	按需选择，建议不低于源RDS实例规格。所有PolarDB节点均为独享型，性能稳定可靠。详情请参见 #unique_67 。
节点个数	无需选择。系统将自动创建一个与主节点规格相同的只读节点。
存储费用	无需选择容量，根据实际数据使用量按小时计费。详情请参见 #unique_67 。
集群名称	填写集群名称用于区分业务用途。如果留空，系统将自动生成一个集群名称。创建集群后还可以修改。

5. 设置**购买时长**（仅针对包年包月集群），然后单击右侧的**立即购买**。

6. 确认订单信息，阅读和勾选**服务协议**，单击**去开通**。

常见问题

从RDS克隆会影响源RDS实例吗？

答：不会影响源RDS实例的正常运行。

相关API

API	描述
#unique_109	创建PolarDB集群。  说明： 一键克隆时，参数 CreationOption 取值需要为 CloneFromRDS 。

后续步骤

请尽快将应用的数据库连接地址修改为PolarDB的地址，详情请参见[查看连接地址](#)。

5.3.4 从RDS MySQL迁移至PolarDB MySQL

PolarDB是阿里巴巴自主研发的下一代关系型分布式云原生数据库，可完全兼容MySQL，具备简单易用、高性能、高可靠、高可用等优势。通过数据传输服务DTS（Data Transmission Service），可以帮助您将RDS MySQL迁移至PolarDB MySQL。

前提条件

已购买目标PolarDB MySQL集群，详情请参见[创建PolarDB MySQL集群](#)。

注意事项

- DTS在执行全量数据迁移时将占用源库和目标库一定的读写资源，可能会导致数据库的负载上升，在数据库性能较差、规格较低或业务量较大的情况下（例如源库有大量慢SQL、存在无主键表或目标库存在死锁等），可能会加重数据库压力，甚至导致数据库服务不可用。因此您需要在执行数据迁移前评估源库和目标库的性能，同时建议您在业务低峰期执行数据迁移（例如源库和目标库的CPU负载在30%以下）。
- 如果源数据库没有主键或唯一约束，且所有字段没有唯一性，可能会导致目标数据库中出现重复数据。
- 由于全量数据迁移会并发执行INSERT操作，将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空间会比源实例大。
- 对于迁移失败的任务，DTS会触发自动恢复。当您需要将业务切换至目标数据库，请务必先结束或释放该任务，避免该任务被自动恢复后，导致源端数据覆盖目标数据库的数据。

功能限制

- DTS支持结构迁移的对象为表、视图、触发器、存储过程、存储函数。



说明：

在结构迁移时，DTS会将视图、存储过程或函数中的DEFINER转换为INVOKER。

- 不支持迁移源库的user信息。迁移完成后，如果您需要调用目标库的视图、存储过程或函数，则需对调用者授予读写权限。

迁移类型介绍

支持结构迁移、全量数据迁移和增量数据迁移，详细介绍请参见[#unique_110](#)。



说明：

同时使用这三种迁移类型可实现在应用不停服的情况下，平滑地完成数据库迁移。

费用说明

迁移类型	链路配置费用	公网流量费用
结构迁移和全量数据迁移	不收费。	通过公网将数据迁移出阿里云时将收费，详情请参见 #unique_111 。
增量数据迁移	收费，详情请参见 #unique_111 。	

增量数据迁移阶段支持同步的SQL操作

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

数据库账号的权限要求

数据库	权限要求	
RDS MySQL	待迁移对象的读权限	
PolarDB MySQL	迁移对象的读写权限	



说明：

关于数据库账号的创建和授权方法，请参见[创建RDS MySQL数据库账号](#)和[创建PolarDB MySQL数据库账号](#)。

操作步骤

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和目标库连接信息。

1.源库及目标库
2.迁移类型及列表
3.映射名称修改
4.预检查

* 任务名称: RDS_TO_PolarDB

源库信息

* 实例类型: RDS实例
* 实例地区: 华东1 (杭州)
* RDS实例ID: rm-bp... 其他阿里云账号下的RDS实例
* 数据库账号: dtstest
* 数据库密码: ***** 测试连接 测试通过
* 连接方式: ☒ 非加密连接 ☐ SSL安全连接

目标库信息

* 实例类型: PolarDB
* 实例地区: 华东1 (杭州)
* PolarDB实例ID: pc-bp...
* 数据库账号: dtstest
* 数据库密码: ***** 测试连接 测试通过

取消 授权白名单并进入下一步

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RDS实例所属的地域。
	数据库账号	填入连接RDS实例的数据库账号，权限要求请参见 数据库账号的权限要求 。
	数据库密码	填入该数据库账号的密码。  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信息是否正确。如果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 。
目标实例信息	实例类型	选择 PolarDB 。
	实例地区	选择目标PolarDB集群所属的地域。

类别	配置	说明
	PolarDB实例ID	选择目标PolarDB集群ID。
	数据库账号	填入目标PolarDB的数据库账号，权限要求请参见 数据库账号的权限要求 。
	数据库密码	填入该数据库账号的密码。  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信息是否正确。如果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。

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地址自动添加到源RDS MySQL和目标PolarDB MySQL的白名单中，用于保障DTS服务器能够正常连接源和目标集群。

7. 选择迁移类型和迁移对象。

1. 源库及目标库 2. 迁移类型及列表 3. 映射名称修改 4. 预检查

* 迁移类型: ☒ 结构迁移 ☒ 全量数据迁移 ☒ 增量数据迁移

迁移对象

若全局搜索, 请先展开树

dtstestdata

Tables

Views

全选中

已选择对象 (鼠标移到对象行, 点击编辑可修改对象名或过滤条件) 详情点我

dtstestdata (2个对象)

customer

order

全移除

* 映射名称更改: ☒ 不进行库表名称批量更改 ☐ 要进行库表名称批量更改

注意:

1. 数据迁移只会将源库的数据 (结构) 复制一份到目标数据库, 并不会对源数据库数据 (结构) 造成影响。

2. 数据迁移过程中, 不支持DDL操作, 如进行DDL操作可能导致迁移失败

取消 上一步 保存 预检查并启动

配置	说明
迁移类型	- 如果只需要进行全量迁移, 则同时选中结构迁移和全量数据迁移。 - 如果需要进行不停机迁移, 则同时选中结构迁移、全量数据迁移和增量数据迁移。 注意: 如果未选中增量数据迁移, 为保障数据一致性, 全量数据迁移期间请勿在源库中写入新的数据。
迁移对象	在迁移对象框中单击待迁移的对象, 然后单击  图标将其移动至已选择对象框。 注意: - 迁移对象选择的粒度为库、表、列。 - 默认情况下, 迁移对象在目标库中的名称与源库保持一致。如果您需要改变迁移对象在目标库中的名称, 需要使用对象名映射功能, 详情请参见#unique_112。 - 如果使用了对象名映射功能, 可能会导致依赖这个对象的其他对象迁移失败。

8. 单击页面右下角的**预检查并启动**。



注意：

- 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图标，查看失败详情。根据提示修复后，重新进行预检查。

9. 预检查通过后，单击**下一步**。

10. 在弹出的**购买配置确认**对话框，选择**链路规格**并选中**数据传输（按量付费）服务条款**。

11. 单击**购买并启动**，迁移任务正式开始。

- 结构迁移+全量数据迁移

请勿手动结束迁移任务，否则可能会导致数据不完整。您只需等待迁移任务完成即可，迁移任务会自动结束。

- 结构迁移+全量数据迁移+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。



注意：

请选择合适的时间手动结束迁移任务，例如业务低峰期或准备将业务切换至目标集群时。

- 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。
- 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。



12. 将业务切换至PolarDB集群。

5.4 PolarDB间的数据迁移

5.4.1 PolarDB MySQL集群间的数据迁移

PolarDB是阿里巴巴自主研发的下一代关系型分布式云原生数据库，可完全兼容MySQL，具备简单易用、高性能、高可靠、高可用等优势。通过数据传输服务DTS（Data Transmission Service），可以帮助您实现PolarDB MySQL集群间的数据迁移。



说明：

目前PolarDB MySQL集群暂不支持升级MySQL版本至8.0版本，您可以创建一个新的PolarDB MySQL集群（8.0版本），使用本方法迁移原集群的数据至新集群。跨版本迁移时，建议创建一个按量付费的PolarDB集群来测试兼容性，测试完成后可释放该集群。

前提条件

- 已购买源和目标PolarDB MySQL集群，详情请参见[创建PolarDB MySQL集群](#)。
- 为满足增量数据迁移的要求，源PolarDB MySQL集群需要开启Binlog，详情请参见[如何开启Binlog](#)。

注意事项

- DTS在执行全量数据迁移时将占用源库和目标库一定的读写资源，可能会导致数据库的负载上升，在数据库性能较差、规格较低或业务量较大的情况下（例如源库有大量慢SQL、存在无主键表或目标库存在死锁等），可能会加重数据库压力，甚至导致数据库服务不可用。因此您需要在执行数据迁移前评估源库和目标库的性能，同时建议您在业务低峰期执行数据迁移（例如源库和目标库的CPU负载在30%以下）。
- 如果源库没有主键或唯一约束，且所有字段没有唯一性，可能会导致目标库中出现重复数据。
- 对于迁移失败的任务，DTS会触发自动恢复。在将业务切换至目标库前，请务必先停止或释放该任务，避免该任务被自动恢复后目标库的数据被源库数据覆盖。

迁移类型介绍

支持结构迁移、全量数据迁移和增量数据迁移，详细介绍请参见[#unique_110](#)。



说明：

同时使用这三种迁移类型可实现在应用不停服的情况下，平滑地完成数据库迁移。

费用说明

迁移类型	链路配置费用	公网流量费用
结构迁移和全量数据迁移	不收费。	通过公网将数据迁移出阿里云时将收费，详情请参见 #unique_111 。
增量数据迁移	收费，详情请参见 #unique_111 。	

增量数据迁移阶段支持同步的SQL操作

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

数据库账号的权限要求

数据库	权限要求	
源PolarDB MySQL	待迁移对象的读权限	
目标PolarDB MySQL	迁移对象的读写权限	



说明：

关于数据库账号的创建和授权方法，请参见[创建PolarDB数据库账号](#)。

操作步骤

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和目标库连接信息。

1. 源库及目标库
2. 迁移类型及列表
3. 高级配置
4. 预检查

* 任务名称: PolarDB集群间迁移

源库信息

* 实例类型: PolarDB
DTS支持链路类型

* 实例地区: 华东1 (杭州)

* PolarDB实例ID: pc-

* 数据库账号: dtstest

* 数据库密码: *****
测试连接
测试通过

目标库信息

* 实例类型: PolarDB

* 实例地区: 华东1 (杭州)

* PolarDB实例ID: pc-

* 数据库账号: dtstest

* 数据库密码: *****
测试连接
测试通过

取消
授权白名单并进入下一步

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 PolarDB 。
	实例地区	选择源PolarDB集群所属的地域。
	PolarDB实例ID	选择源PolarDB集群ID。
	数据库账号	填入连接源PolarDB集群的数据库账号。
	数据库密码	填入该数据库账号的密码。  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信息是否正确。如果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。
目标库信息	实例类型	选择 PolarDB 。
	实例地区	选择目标PolarDB集群所属的地域。
	PolarDB实例ID	选择目标PolarDB集群ID。

类别	配置	说明
	数据库账号	填入目标PolarDB集群的数据库账号。
	数据库密码	填入该数据库账号的密码。  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 来验证填入的信息是否正确。如果填写正确则提示 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。

6. 完成上述配置后，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地址自动添加到源和目标PolarDB集群的白名单中，用于保障DTS服务器能够正常连接源和目标集群。

7. 选择迁移类型和迁移对象。

1. 源库及目标库 2. 迁移类型及列表 3. 映射名称修改 4. 预检查

* 迁移类型: ☒ 结构迁移 ☒ 全量数据迁移 ☒ 增量数据迁移

迁移对象

若全局搜索, 请先展开树

dttestdata

- Tables
- Views

全选中

已选择对象 (鼠标移到对象行, 点击编辑可修改对象名或过滤条件) 详情点我

dttestdata (2个对象)

- customer
- order

全移除

* 映射名称更改: ☒ 不进行库表名称批量更改 ☐ 要进行库表名称批量更改

注意:

1. 数据迁移只会将源库的数据 (结构) 复制一份到目标数据库, 并不会对源数据库数据 (结构) 造成影响。
2. 数据迁移过程中, 不支持DDL操作, 如进行DDL操作可能导致迁移失败

取消 上一步 保存 预检查并启动

配置	说明
迁移类型	- 如果只需要进行全量迁移, 则同时选中结构迁移和全量数据迁移。 - 如果需要进行不停机迁移, 则同时选中结构迁移、全量数据迁移和增量数据迁移。 注意: 如果未选中增量数据迁移, 为保障数据一致性, 全量数据迁移期间请勿在源库中写入新的数据。
迁移对象	在迁移对象框中单击待迁移的对象, 然后单击  图标将其移动至已选择对象框。 注意: - 迁移对象选择的粒度为库、表、列。 - 默认情况下, 迁移对象在目标库中的名称与源库保持一致。如果您需要改变迁移对象在目标库中的名称, 需要使用对象名映射功能, 详情请参见#unique_112。 - 如果使用了对象名映射功能, 可能会导致依赖这个对象的其他对象迁移失败。

8. 单击页面右下角的**预检查并启动**。



注意：

- 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图标，查看失败详情。根据提示修复后，重新进行预检查。

9. 预检查通过后，单击**下一步**。

10. 在弹出的**购买配置确认**对话框，选择**链路规格**并选中**数据传输（按量付费）服务条款**。

11. 单击**购买并启动**，迁移任务正式开始。

- 结构迁移+全量数据迁移

请勿手动结束迁移任务，否则可能会导致数据不完整。您只需等待迁移任务完成即可，迁移任务会自动结束。

- 结构迁移+全量数据迁移+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。



注意：

请选择合适的时间手动结束迁移任务，例如业务低峰期或准备将业务切换至目标集群时。

- 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。
- 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。



12. 将业务切换至PolarDB集群。

后续步骤

由于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，删除相关的数据库账号。

5.5 从其他数据库迁移至PolarDB

5.5.1 从自建MySQL迁移至PolarDB MySQL

PolarDB是阿里巴巴自主研发的下一代关系型分布式云原生数据库，可完全兼容MySQL，具备简单易用、高性能、高可靠、高可用等优势。通过数据传输服务DTS（Data Transmission Service），可以帮助您将自建MySQL数据库迁移至PolarDB MySQL。

前提条件

- 自建MySQL数据库版本为5.1、5.5、5.6、5.7或8.0版本。
- 已创建目标PolarDB MySQL集群，详情请参见[创建PolarDB MySQL集群](#)。
- 如果您的MySQL数据库部署在本地，那么您需要将DTS服务器的IP地址设置为该数据库远程连接的白名单，允许其访问您的数据库。详情请参见[#unique_115](#)。

注意事项

- DTS在执行全量数据迁移时将占用源库和目标库一定的读写资源，可能会导致数据库的负载上升，在数据库性能较差、规格较低或业务量较大的情况下（例如源库有大量慢SQL、存在无主键表或目标库存在死锁等），可能会加重数据库压力，甚至导致数据库服务不可用。因此您需要在执行数据迁移前评估源库和目标库的性能，同时建议您在业务低峰期执行数据迁移（例如源库和目标库的CPU负载在30%以下）。
- 如果源数据库没有主键或唯一约束，且所有字段没有唯一性，可能会导致目标数据库中出现重复数据。
- 对于数据类型为FLOAT或DOUBLE的列，DTS会通过ROUND(COLUMN,PRECISION)来读取该列的值。如果没有明确定义其精度，DTS对FLOAT的迁移精度为38位，对DOUBLE的迁移精度为308位，请确认迁移精度是否符合业务预期。
- 对于迁移失败的任务，DTS会触发自动恢复。在您将业务切换至目标集群前，请务必先结束或释放该任务，避免该任务被自动恢复后，导致源端数据覆盖目标集群的数据。

迁移类型介绍

支持结构迁移、全量数据迁移和增量数据迁移，详细介绍请参见[#unique_110](#)。



说明：

同时使用这三种迁移类型可实现在应用不停服的情况下，平滑地完成数据库迁移。

费用说明

迁移类型	链路配置费用	公网流量费用
结构迁移和全量数据迁移	不收费。	通过公网将数据迁移出阿里云时将收费，详情请参见 #unique_111 。
增量数据迁移	收费，详情请参见 #unique_111 。	

增量数据迁移阶段支持同步的SQL操作

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

数据库账号的权限要求

数据库	结构/全量迁移	增量迁移
自建MySQL数据库	SELECT权限	REPLICATION CLIENT、REPLICATION SLAVE、SHOW VIEW和SELECT权限
PolarDB集群	读写权限	读写权限

数据库账号创建及授权方法：

- 自建MySQL数据库请参见[#unique_116](#)。
- PolarDB集群请参见[创建数据库账号](#)。

准备工作

[#unique_116](#)

操作步骤

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和目标库连接信息。

1.源库及目标库
2.迁移类型及列表
3.映射名称修改
4.预检查

* 任务名称: polardb

源库信息

* 实例类型: 有公网IP的自建数据库
* 实例地区: 华东1 (杭州) 获取DTS IP段
* 数据库类型: MySQL
* 主机名或IP地址: 12.***.***.***
* 端口: 3306
* 数据库账号: dtstest
* 数据库密码: ***** 测试连接 测试通过

目标库信息

* 实例类型: PolarDB
* 实例地区: 华东1 (杭州)
* PolarDB实例ID: pc-bp-***.***.***.***
* 数据库账号: dtstest
* 数据库密码: ***** 测试连接 测试通过

取消 上云评估 授权白名单并进入下一步

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	根据源库的部署位置进行选择，本文以 有公网IP的自建数据库 为例介绍配置流程。  说明： 当自建数据库为其他实例类型时，您还需要执行相应的准备工作，详情请参见#unique_117。
	实例地区	当实例类型选择为 有公网IP的自建数据库 时， 实例地区 无需设置。  说明： 如果您的自建MySQL数据库具有白名单安全设置，您需要在实例地区配置项后，单击获取DTS IP段来获取到DTS服务器的IP地址，并将获取到的IP地址加入自建MySQL数据库的白名单安全设置中。
	数据库类型	选择 MySQL
	主机名或IP地址	填入自建MySQL数据库的访问地址，本案例中填入公网地址。

类别	配置	说明
	端口	填入自建MySQL数据库的服务端口，默认为 3306 。
	数据库账号	填入自建MySQL的数据库账号，权限要求请参见 数据库账号的权限要求 。
	数据库密码	填入该账号对应的密码。  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信息是否正确。如果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。
目标实例信息	实例类型	选择 PolarDB 。
	实例地区	选择目标PolarDB集群所属的地域。
	PolarDB实例ID	选择目标PolarDB集群ID。
	数据库账号	填入目标PolarDB集群的数据库账号，权限要求请参见 数据库账号的权限要求 。
	数据库密码	填入该账号对应的密码。  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信息是否正确。如果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。

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地址自动添加到目标PolarDB MySQL的白名单中，用于保障DTS服务器能够正常连接目标集群。

7. 选择迁移类型和迁移对象。

1. 源库及目标库 2. 迁移类型及列表 3. 映射名称修改 4. 预检查

* 迁移类型: ☒ 结构迁移 ☒ 全量数据迁移 ☒ 增量数据迁移

迁移对象

若全局搜索, 请先展开树

dttestdata

Tables

Views

已选择对象 (鼠标移到对象行, 点击编辑可修改对象名或过滤条件) 详情点我

dttestdata (2个对象)

customer

order

* 映射名称更改: ☒ 不进行库表名称批量更改 ☐ 要进行库表名称批量更改

注意:

1. 数据迁移只会将源库的数据 (结构) 复制一份到目标数据库, 并不会对源数据库数据 (结构) 造成影响。

2. 数据迁移过程中, 不支持DDL操作, 如进行DDL操作可能导致迁移失败

取消 上一步 保存 预检查并启动

配置	说明
迁移类型	- 如果只需要进行全量迁移, 则同时选中结构迁移和全量数据迁移。 - 如果需要进行不停机迁移, 则同时选中结构迁移、全量数据迁移和增量数据迁移。 注意: 如果未选中增量数据迁移, 为保障数据一致性, 全量数据迁移期间请勿在源库中写入新的数据。
迁移对象	在迁移对象框中单击待迁移的对象, 然后单击  图标将其移动至已选择对象框。 注意: - 迁移对象选择的粒度为库、表、列。 - 默认情况下, 迁移对象在目标库中的名称与源库保持一致。如果您需要改变迁移对象在目标库中的名称, 需要使用对象名映射功能, 详情请参见#unique_112。 - 如果使用了对象名映射功能, 可能会导致依赖这个对象的其他对象迁移失败。

8. 单击页面右下角的**预检查并启动**。



注意：

- 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图标，查看失败详情。根据提示修复后，重新进行预检查。

9. 预检查通过后，单击**下一步**。

10. 在弹出的**购买配置确认**对话框，选择**链路规格**并选中**数据传输（按量付费）服务条款**。

11. 单击**购买并启动**，迁移任务正式开始。

- 结构迁移+全量数据迁移

请勿手动结束迁移任务，否则可能会导致数据不完整。您只需等待迁移任务完成即可，迁移任务会自动结束。

- 结构迁移+全量数据迁移+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。



注意：

请选择合适的时间手动结束迁移任务，例如业务低峰期或准备将业务切换至目标集群时。

- 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。
- 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。



12. 将业务切换至PolarDB集群。

5.5.2 从本地MySQL迁移至PolarDB MySQL

本文介绍使用DTS进行本地MySQL迁移至PolarDB MySQL的任务配置流程。

前提条件

- 已经**创建阿里云PolarDB集群**。
- 已经**创建拥有读写权限的账号**。

- 已经开通本地MySQL的远程访问权限。开通命令为 **grant all privileges on *.* to <username>@'<ipaddress>' identified by "<password>";**

**说明：**

- <username>：本地MySQL数据库的用户名。
- <ipaddress>：允许登录数据库的ip地址，localhost 表示只能本地登录数据库，%表示任何ip地址都能登录数据库。
- <password>：数据库的用户名对应的密码。

背景信息

使用阿里云[数据传输服务（DTS）](#)，您可以实现本地MySQL到PolarDB MySQL集群的数据迁移。通过DTS增量迁移的存储引擎，可以实现在本地应用不停服的情况下，将数据迁移到目标PolarDB MySQL集群。

支持同步的SQL操作

对于**本地 MySQL > PolarDB** 数据迁移，DTS支持同步的SQL操作如下：

INSERT、UPDATE、DELETE、REPLACE

ALTER TABLE、ALTER VIEW、ALTER FUNCTION、ALTER PROCEDURE

CREATE DATABASE、CREATE SCHEMA、CREATE INDEX、CREATE TABLE、CREATE PROCEDURE、CREATE

FUNCTION、CREATE TRIGGER、CREATE VIEW、CREATE EVENT

DROP FUNCTION、DROP EVENT、DROP INDEX、DROP PROCEDURE、DROP TABLE、DROP TRIGGER、DROP

VIEW

RENAME TABLE、TRUNCATE TABLE

注意事项

- 执行迁移任务前建议提前做好数据备份。
- 对于七天之内的异常任务，DTS会尝试自动恢复，可能会导致迁移任务的源端数据库数据覆盖目标实例数据库中写入的业务数据，迁移任务结束后务必将DTS访问目标实例账号的写权限用revoke命令回收掉。

迁移限制

- 迁移仅支持 MySQL 5.6版本。

- 结构迁移不支持 event 的迁移。
- 对于MySQL的浮点型float/double，DTS通过round(column,precision)来读取该列的值，若列类型没有明确定义其精度，对于float，精度为38位，对于double类型，精度为308，请先确认DTS的迁移精度是否符合业务预期。
- 如果使用了对象名映射功能，依赖这个对象的其他对象可能迁移失败。
- 当选择增量迁移时，源端的MySQL实例需要开启binlog。
- 当选择增量迁移时，源库的binlog_format要为row。
- 当选择增量迁移且源MySQL为5.6版本时，它的binlog_row_image必须为full。
- 当选择增量迁移时，增量迁移过程中如果源MySQL实例出现因实例跨机迁移或跨机重建等导致的binlog文件ID乱序，可能导致增量迁移数据丢失。

迁移权限要求

当使用DTS进行**本地MySQL > PolarDB**迁移时，不同的迁移类型，对源端和目标端MySQL实例的迁移账号权限要求如下：

迁移类型	结构迁移	全量数据迁移	增量数据迁移
本地MySQL实例	select	select	super select replication slave replication client
目标端PolarDB集群	读写权限	读写权限	读写权限

迁移流程

DTS在进行**本地MySQL > POLARDB**数据迁移时，为了解决对象间的依赖关系，提高迁移成功率。

结构对象及数据的迁移顺序如下：

1. 结构对象：表、视图的迁移。
2. 全量数据迁移。
3. 结构对象：存储过程、函数、触发器、外键的迁移。
4. 增量数据迁移。



说明：

如果没有选择增量数据迁移，那么当全量数据迁移完成后，任务列表中的迁移进度为：结构迁移100%，全量迁移100%，迁移状态为**迁移中**。此时迁移任务正在进行步骤3中的对象的迁移。请勿手动结束任务，否则会造成迁移数据丢失。

操作步骤

1. 登录DTS控制台。
2. 在左侧菜单栏中，单击**数据迁移**。
3. 单击右上角**创建迁移任务**。
4. （可选）填写任务名称。

DTS 为每个任务自动生成一个任务名称，任务名称没有唯一性要求。您可以根据需要修改任务名称，建议为任务配置具有业务意义的名称，便于后续的任务识别。

5. 填写源库和目标库信息，具体参数配置说明如下：

* 任务名称：

源库信息

* 实例类型：

* 实例地区： [获取DTS IP段](#)

* 数据库类型：

* 主机名或IP地址：

* 端口：

* 数据库账号：

* 数据库密码：

测试连接 测试通过

目标库信息

* 实例类型：

* 实例地区：

* POLARDB实例ID：

* 数据库账号：

* 数据库密码：

测试连接 测试通过

库类别	参数	说明
源库信息	实例类型	源数据库实例类型，选择有公网IP的自建数据库。
	实例地区	选择源数据库所在的地区。
	数据库类型	源数据库类型，选择MySQL。
	主机名或IP地址	源数据库的公网IP地址。

库类别	参数	说明
	端口	源数据库的端口。
	数据库账号	源数据库具有读写权限的账号。
	数据库密码	源数据库账号对应的密码。
目标库信息	实例类型	选择POLARDB。
	实例地区	目标实例所在的地区。
	POLARDB实例ID	对应地区下的实例ID，选择要迁移到的目标实例的ID。
	数据库账号	目标实例的拥有读写权限的账号。
	数据库密码	目标实例的对应账号的密码。

6. 填写完成后单击**测试连接**，确定源库和目标库都测试通过。

7. 单击右下角**授权白名单并进入下一步**。

8. 选择迁移类型和迁移对象。

创建迁移任务 返回上级

1. 源库及目标库 2. 迁移类型及对象 3. 预检查

迁移类型：☒ 结构迁移 ☒ 全量数据迁移 ☐ 增量数据迁移

全量数据迁移过程中，如果源库有数据更新，这部分更新数据不会被迁移到目标实例，为保证迁移数据的一致性，建议选择结构迁移+全量数据迁移+增量数据迁移。

迁移对象

已选择对象 (鼠标移到对象行, 点击编辑可修改对象名称或过滤条件) 详情去我

test2

test1

全选中 全等除

取消 上一步 保存 预检查并启动

注意：
1. 数据迁移只会迁移库的数据（结构）复制一份到目标数据库，并不会对源数据库数据（结构）造成影响。
2. 数据迁移过程中，不支持DDL操作，如进行DDL操作可能导致迁移失败。

• 迁移类型：

- 结构迁移

DTS 会将迁移对象的结构定义迁移到目标集群。目前 DTS 支持结构迁移的对象有：表、视图、触发器、存储过程、存储函数。

- 全量数据迁移

DTS 会将源数据库迁移对象的存量数据全部迁移到目标集群。由于全量迁移并发Insert导致目标实例的表存在碎片，全量迁移完成后目标实例的表空间会比源实例大。

如果用户只进行全量数据迁移，那么迁移过程中本地 MySQL 实例新增的业务写入不会被同步到目标 RDS for MySQL 实例。

- 增量数据迁移

DTS 会将迁移过程中源实例的数据变更同步到目标集群。如果迁移期间进行了 DDL 操作，那么这些结构变更不会迁移到目标集群。

如果只需要进行全量迁移，建议迁移类型选择：结构迁移 + 全量数据迁移。

如果需要进行不停机迁移，建议迁移类型选择：结构迁移 + 全量数据迁移 + 增量数据迁移。



说明：

结构迁移和全量数据迁移均不收费，增量迁移需要收费。

- **迁移对象：**选择要迁移的对象，单击向右的箭头，将选中的对象添加到右侧。

迁移对象的选择粒度细化为：库、表、列三个粒度。默认情况下，对象迁移到目标PolarDB集群后，对象名跟源RDS实例一致。如果您迁移的对象在源实例跟目标集群上名称不同，那么需要使用DTS提供的对象名映射功能，详细使用方式可以参考[库表列映射](#)。



说明：

- 暂时不支持对系统表的迁移。
- 目标集群中不能有和迁移对象同名的对象。将鼠标移至右侧框中的对象，单击**编辑**，即可修改迁移后的对象名。

9. 单击右下角的预检查并启动**，成功后单击**下一步**。**

预检查

×

预检查通过100%

检测项	检测内容	检测结果
源库连接性检查	检查数据传输服务器是否能连通源数据库	成功
源库连接性检查	检查数据传输服务器是否能连通源数据库	成功
源库权限检查	检查源数据库的账号权限是否满足迁移要求	成功
目的库连接性检查	检查数据传输服务器是否能连通目的数据库	成功
目的库连接性检查	检查数据传输服务器是否能连通目的数据库	成功
目的库权限检查	检查目的数据库的账号权限是否满足迁移要求	成功
存储引擎检查	检查迁移表是否有不支持的存储引擎	成功

下一步



说明：

如果预检查失败，可以单击具体检查项后的检查结果，查看具体的失败详情，并根据失败原因修复后，重新进行预检查。

10. 确认DTS购买信息，阅读并勾选服务条款，单击立即购买并启动。

如果选择了增量迁移，那么进入增量迁移阶段后，源库的更新写入都会被DTS同步到目标PolarDB集群。迁移任务不会自动结束。如果您只是为了迁移，那么建议在增量迁移无延迟的状态时，源库停写几分钟，等待增量迁移再次进入无延迟状态后，停止掉迁移任务，直接将业务切换到目标PolarDB集群上即可。

11. 单击目标地域，查看迁移状态。迁移完成时，状态为已完成。



至此，完成本地MySQL数据库到阿里云PolarDB的数据迁移任务。

5.5.3 从Amazon Aurora MySQL迁移至PolarDB MySQL

本文介绍如何使用数据传输服务DTS（Data Transmission Service），将Amazon Aurora MySQL迁移至阿里云PolarDB MySQL。DTS支持结构迁移、全量数据迁移以及增量数据迁移，同时使用这三种迁移类型可以实现在自建应用不停服的情况下，平滑地完成数据库迁移。

前提条件

- 为保障DTS可以通过公网连接至Amazon Aurora MySQL，Amazon Aurora MySQL的网络与安全配置中须将公开可用性功能设置为**是**。
- 已创建阿里云PolarDB MySQL集群，详情请参见[PolarDB MySQL集群](#)。
- 阿里云PolarDB MySQL的存储空间须大于Amazon Aurora MySQL已使用的存储空间。

注意事项

- DTS在执行全量数据迁移时将占用源库和目标库一定的读写资源，可能会导致数据库的负载上升，在数据库性能较差、规格较低或业务量较大的情况下（例如源库有大量慢SQL、存在无主键表或目标库存在死锁等），可能会加重数据库压力，甚至导致数据库服务不可用。因此您需要在执行数据迁移前评估源库和目标库的性能，同时建议您在业务低峰期执行数据迁移（例如源库和目标库的CPU负载在30%以下）。
- 如果源数据库没有主键或唯一约束，且所有字段没有唯一性，可能会导致目标数据库中出现重复数据。
- 对于数据类型为FLOAT或DOUBLE的列，DTS会通过ROUND(COLUMN,PRECISION)来读取该列的值。如果没有明确定义其精度，DTS对FLOAT的迁移精度为38位，对DOUBLE的迁移精度为308位，请确认迁移精度是否符合业务预期。

- 如果待迁移数据库名称不符合阿里云PolarDB MySQL的定义规范，您需要在配置迁移任务之前在阿里云PolarDB MySQL中创建数据库。

**说明：**

关于阿里云PolarDB MySQL的定义规范和创建数据库的操作方法，请参见[创建数据库](#)。

- 对于迁移失败的任务，DTS会触发自动恢复。在您将业务切换至目标实例前，请务必先结束或释放该任务，避免该任务被自动恢复后，导致源端数据覆盖目标实例的数据。

费用说明

迁移类型	链路配置费用	公网流量费用
结构迁移和全量数据迁移	不收费。	通过公网将数据迁移出阿里云时将收费，详情请参见 #unique_111 。
增量数据迁移	收费，详情请参见 #unique_111 。	

迁移类型说明

- 结构迁移

DTS将待迁移对象的结构定义迁移到阿里云PolarDB MySQL，目前DTS支持结构迁移的对象为表、视图、触发器、存储过程、存储函数，不支持event的结构迁移。

**说明：**

- 在结构迁移时，DTS会将视图、存储过程和函数中的DEFINER转换为INVOKER。
- 由于DTS不迁移user信息，因此在调用目标库的视图、存储过程和函数时需要对调用者授予读写权限。

- 全量数据迁移

DTS会将Amazon Aurora MySQL中待迁移对象的存量数据，全部迁移到阿里云PolarDB MySQL中。

**说明：**

由于全量数据迁移会并发INSERT导致目标实例的表存在碎片，全量迁移完成后目标实例的表空间会比源实例大。

- 增量数据迁移

在全量迁移的基础上，DTS会读取Amazon Aurora MySQL的binlog信息，将Amazon Aurora MySQL的增量更新数据同步到阿里云PolarDB MySQL中。通过增量数据迁移可以实现在应用不停服的情况下，平滑地完成MySQL数据库的迁移。

数据库账号的权限要求

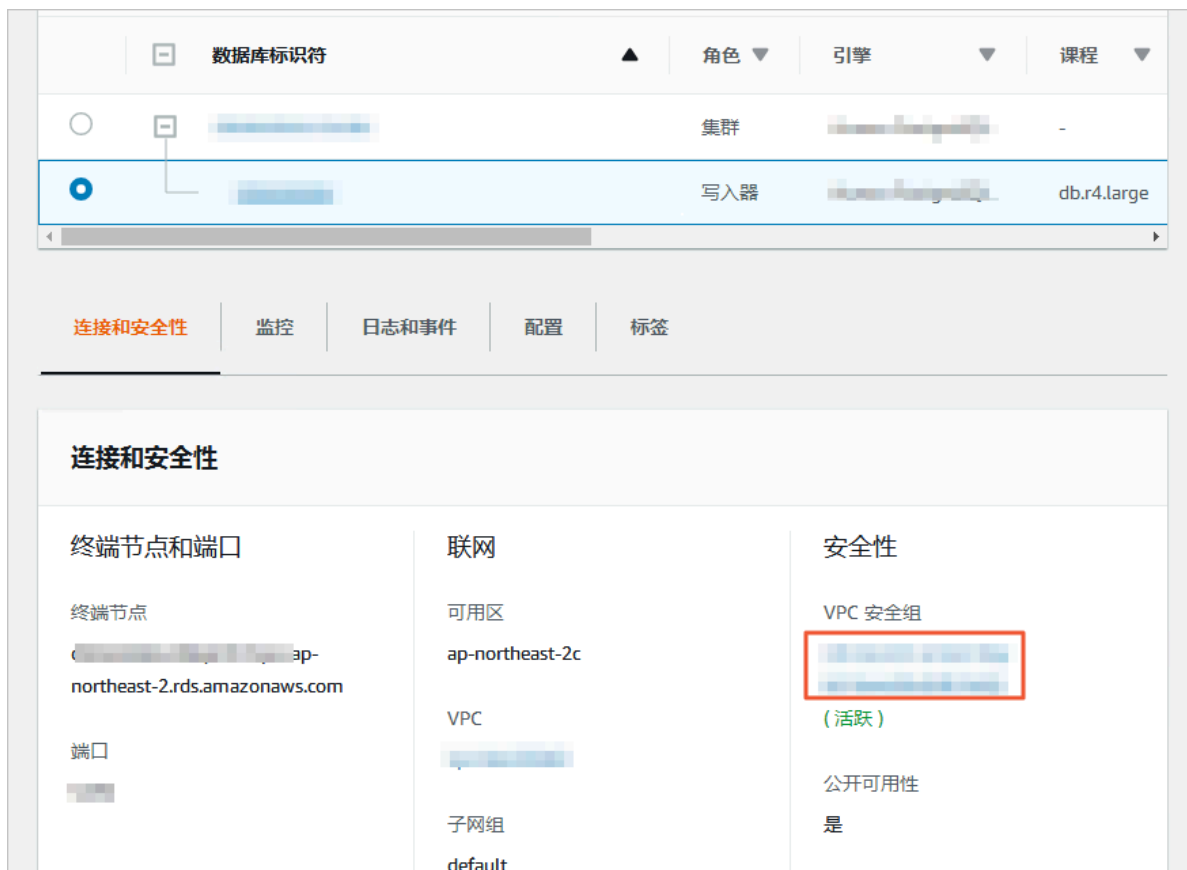
数据库	结构迁移	全量迁移	增量迁移
Amazon Aurora MySQL	迁移对象的SELECT权限	迁移对象的SELECT权限	REPLICATION SLAVE、REPLICATION CLIENT、SHOW VIEW和对迁移对象执行SELECT操作的权限
阿里云PolarDB MySQL	迁移对象的读写权限	迁移对象的读写权限	迁移对象的读写权限

数据库账号创建及授权方法：

- Amazon Aurora MySQL请参见[#unique_116](#)中创建账号的部分。
- 阿里云PolarDB MySQL请参见[创建账号](#)。

迁移前准备工作

1. 登录Amazon Aurora控制台。
2. 进入Amazon Aurora MySQL的**基本信息**页面。
3. 选择角色为**写入器**的节点。
4. 在**连接和安全性**区域框，单击对应的VPC安全组名称。



5. 登录Amazon Aurora MySQL数据库，设置binlog日志保存时间。如果不需要增量数据迁移，可跳过本步骤。

```
call mysql.rds_set_configuration('binlog retention hours', 24);
```

**说明：**

- 上述命令将binlog日志的保存设置为24小时，最大可设置为168个小时，即7天。
- Amazon Aurora MySQL的binlog日志需处于开启状态，且binlog_format需设置为row；当MySQL为5.6及以上版本时，binlog_row_image需设置为full。

操作步骤

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。

1.源库及目标库
2.迁移类型及列表
3.映射名称修改
4.预检查

* 任务名称: polardb

源库信息

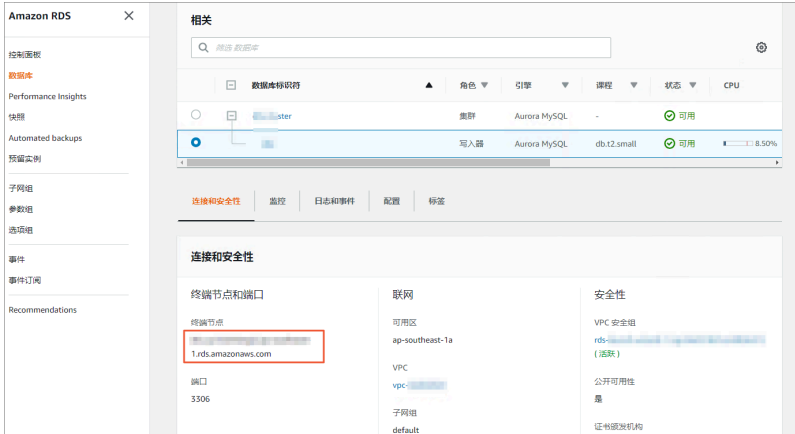
* 实例类型: 有公网IP的自建数据库
* 实例地区: 华东1 (杭州) 获取DTS IP段
* 数据库类型: MySQL
* 主机名或IP地址: 12...
* 端口: 3306
* 数据库账号: dtstest
* 数据库密码: ***** 测试连接 测试通过

目标库信息

* 实例类型: PolarDB
* 实例地区: 华东1 (杭州)
* PolarDB实例ID: pc-bp...
* 数据库账号: dtstest
* 数据库密码: ***** 测试连接 测试通过

取消 上云评估 授权白名单并进入下一步

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 有公网IP的自建数据库 。
	实例地区	当实例类型选择为 有公网IP的自建数据库 时， 实例地区 无需设置。
	数据库类型	选择 MySQL 。

类别	配置	说明
	主机名或IP地址	填入Amazon Aurora MySQL的访问地址。  说明： 您可以在Amazon Aurora MySQL的基本信息页面，获取数据库的访问地址。 
	端口	填入Amazon Aurora MySQL的服务端口，默认为 3306 。
	数据库账号	填入Amazon Aurora MySQL的数据库账号，权限要求请参见 数据库账号的权限要求 。
	数据库密码	填入该数据库账号的密码。  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信息是否正确。如果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。
目标库信息	实例类型	选择 PolarDB 。
	实例地区	选择阿里云PolarDB MySQL实例所属地域。
	PolarDB实例ID	选择阿里云PolarDB MySQL实例ID。
	数据库账号	填入阿里云PolarDB MySQL的数据库账号，权限要求请参见 数据库账号的权限要求 。
	数据库密码	填入该数据库账号的密码。  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信息是否正确。如果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。

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地址自动添加到阿里云PolarDB MySQL的白名单中，用于保障DTS服务器能够正常连接阿里云PolarDB MySQL。

7. 选择迁移对象及迁移类型。

1. 源库及目标库

2. 迁移类型及列表

3. 映射名称修改

4. 预检查

迁移类型：

☒ 结构迁移
 ☒ 全量数据迁移
 ☒ 增量数据迁移

数据迁移适合于短期的数据迁移场景，主要应用于上云迁移、数据库扩容拆分及阿里云数据库之间的数据迁移。

如果需要长期进行数据实时同步，请使用数据同步功能。

迁移对象

若全局搜索，请先展开树

data123

mysqltest

Tables

customer2

Views

Functions

Procedures

sys

已选择对象 (鼠标移到对象行, 点击编辑可修改对象名或过滤条件) 详情

mysqltest (2个对象)

customer

vipinfo

全选中

全移除

映射名称更改：

☒ 不进行库表名称批量更改
 ☐ 要进行库表名称批量更改

注意：

1. 数据迁移只会将源库的数据（结构）复制一份到目标数据库，并不会对源数据库数据（结构）造成影响。

2. 数据迁移过程中，不支持DDL操作，如进行DDL操作可能导致迁移失败

取消

上一步

保存

预检查并启动

配置	说明
迁移类型	- 如果只需要进行全量迁移，则同时勾选结构迁移和全量数据迁移。 - 如果需要进行不停机迁移，则同时勾选结构迁移、全量数据迁移和增量数据迁移。 说明： 如果未选择增量数据迁移，为保障数据一致性，数据迁移期间请勿在Amazon Aurora MySQL中写入新的数据。

60

文档版本：20200709

配置	说明
迁移对象	在迁移对象框中单击待迁移的对象，然后单击将其移动至已选择对象框。  说明： - 迁移对象选择的粒度为库、表、列。 - 默认情况下，迁移对象在目标库中的名称与源库保持一致。如果您需要改变迁移对象在目标库中的名称，需要使用对象名映射功能，详情请参见#unique_112。 - 如果使用了对象名映射功能，可能会导致依赖这个对象的其他对象迁移失败。

8. 单击页面右下角的**预检查并启动**。



注意：

- 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图标，查看失败详情。根据提示修复后，重新进行预检查。

9. 预检查通过后，单击**下一步**。

10. 在弹出的**购买配置确认**对话框，选择**链路规格**并选中**数据传输（按量付费）服务条款**。

11. 单击**购买并启动**，迁移任务正式开始。

- 结构迁移+全量数据迁移

请勿手动结束迁移任务，否则可能会导致数据不完整。您只需等待迁移任务完成即可，迁移任务会自动结束。

- 结构迁移+全量数据迁移+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。



注意：

请选择合适的时间手动结束迁移任务，例如业务低峰期或准备将业务切换至目标集群时。

- 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。



12. 将业务切换至阿里云PolarDB MySQL。

5.6 从PolarDB迁移至其他数据库

5.6.1 从PolarDB MySQL迁移至RDS MySQL

阿里云关系型数据库RDS（Relational Database Service）是一种稳定可靠、可弹性伸缩的在线数据库服务，提供容灾、备份、恢复、迁移等方面的全套解决方案。通过数据传输服务DTS（Data Transmission Service），您可以将PolarDB MySQL迁移至RDS MySQL。

前提条件

- 源PolarDB MySQL集群需开启Binlog，详情请参见[如何开启Binlog](#)。
- 已购买目标RDS MySQL实例，详情请参见[创建RDS MySQL实例](#)。

注意事项

- 如果源数据库没有主键或唯一约束，且所有字段没有唯一性，可能会导致目标数据库中出现重复数据。
- 由于全量数据迁移会并发执行INSERT操作，将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空间会比源实例大。
- 对于迁移失败的任务，DTS会触发自动恢复。当您需要将业务切换至目标数据库，请务必先结束或释放该任务，避免该任务被自动恢复后，导致源端数据覆盖目标数据库的数据。

功能限制

- DTS支持结构迁移的对象为表、视图、触发器、存储过程、存储函数。



说明：

在结构迁移时，DTS会将视图、存储过程或函数中的DEFINER转换为INVOKER。

- 不支持迁移源库的user信息。迁移完成后，如果您需要调用目标库的视图、存储过程或函数，则需对调用者授予读写权限。

迁移类型介绍

支持结构迁移、全量数据迁移和增量数据迁移，详细介绍请参见[#unique_110](#)。



说明：

同时使用这三种迁移类型可实现在应用不停服的情况下，平滑地完成数据库迁移。

费用说明

迁移类型	链路配置费用	公网流量费用
结构迁移和全量数据迁移	不收费。	通过公网将数据迁移出阿里云时将收费，详情请参见 #unique_111 。
增量数据迁移	收费，详情请参见 #unique_111 。	

增量数据迁移阶段支持同步的SQL操作

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

数据库账号的权限要求

数据库	权限要求	
PolarDB MySQL	待迁移对象的读权限	
RDS MySQL	迁移对象的读写权限	



说明：

关于数据库账号的创建和授权方法，请参见[创建PolarDB MySQL数据库账号](#)和[创建RDS MySQL数据库账号](#)。

操作步骤

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，选择迁移的目标实例所属地域。



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和目标库连接信息。

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 PolarDB 。
	实例地区	选择源PolarDB集群所属的地域。
	PolarDB实例ID	选择源PolarDB集群ID。
	数据库账号	填入PolarDB集群的数据库账号，权限要求请参见 数据库账号的权限要求 。

类别	配置	说明
	数据库密码	填入该数据库账号的密码。  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 来验证填入的信息是否正确。如果填写正确则提示 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。
目标库信息	实例类型	选择 RDS实例 。
	实例地区	当RDS实例所属的地域。
	数据库账号	填入连接RDS实例的数据库账号，权限要求请参见 数据库账号的权限要求 。
	数据库密码	填入该数据库账号的密码。  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 来验证填入的信息是否正确。如果填写正确则提示 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。
	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地址自动添加到源PolarDB MySQL和目标RDS MySQL的白名单中，用于保障DTS服务器能够正常连接源集群和目标实例。

7. 选择迁移类型和迁移对象。

1. 源库及目标库
2. 迁移类型及列表
3. 映射名称修改
4. 预检查

* 迁移类型: ☒ 结构迁移 ☒ 全量数据迁移 ☒ 增量数据迁移

迁移对象

若全局搜索, 请先展开树

dttestdata

Tables

Views

全选中

已选择对象 (鼠标移到对象行, 点击编辑可修改对象名或过滤条件) 详情点我

dttestdata (2个对象)

customer

order

全移除

>
<

* 映射名称更改: ☒ 不进行库表名称批量更改 ☐ 要进行库表名称批量更改

注意:

1. 数据迁移只会将源库的数据 (结构) 复制一份到目标数据库, 并不会对源数据库数据 (结构) 造成影响。
2. 数据迁移过程中, 不支持DDL操作, 如进行DDL操作可能导致迁移失败

取消
上一步
保存
预检查并启动

配置	说明
迁移类型	- 如果只需要进行全量迁移, 则同时选中结构迁移和全量数据迁移。 - 如果需要进行不停机迁移, 则同时选中结构迁移、全量数据迁移和增量数据迁移。  说明: 如果未选中增量数据迁移, 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。
迁移对象	在迁移对象框中单击待迁移的对象, 然后单击  图标将其移动至已选择对象框。  说明: - 迁移对象选择的粒度为库、表、列。 - 默认情况下, 迁移对象在目标库中的名称与源库保持一致。如果您需要改变迁移对象在目标库中的名称, 需要使用对象名映射功能, 详情请参见#unique_112。 - 如果使用了对象名映射功能, 可能会导致依赖这个对象的其他对象迁移失败。

8. 单击页面右下角的**预检查并启动**。



注意：

- 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图标，查看失败详情。根据提示修复后，重新进行预检查。

9. 预检查通过后，单击**下一步**。

10. 在弹出的**购买配置确认**对话框，选择**链路规格**并选中**数据传输（按量付费）服务条款**。

11. 单击**购买并启动**，迁移任务正式开始。

- 结构迁移+全量数据迁移

请勿手动结束迁移任务，否则可能会导致数据不完整。您只需等待迁移任务完成即可，迁移任务会自动结束。

- 结构迁移+全量数据迁移+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。



注意：

请选择合适的时间手动结束迁移任务，例如业务低峰期或准备将业务切换至目标集群时。

- 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。
- 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。



12. 将业务切换至RDS实例。

5.7 PolarDB间的数据同步

5.7.1 PolarDB MySQL集群间的单向同步

PolarDB是阿里巴巴自主研发的下一代关系型分布式云原生数据库，可完全兼容MySQL，具备简单易用、高性能、高可靠、高可用等优势。通过数据传输服务DTS（Data Transmission Service），您可以实现PolarDB MySQL集群间的单向数据同步。

前提条件

- 已购买源和目标PolarDB MySQL集群，详情请参见[创建PolarDB MySQL集群](#)。
- 源PolarDB MySQL集群已开启Binlog，详情请参见[如何开启Binlog](#)。

注意事项

- 全量初始化过程中，并发INSERT会导致目标集群的表碎片，全量初始化完成后，目标集群的表空间比源库的表空间大。
- 如果数据同步的源库没有主键或唯一约束，且记录的全字段没有唯一性，可能会出现重复数据。

支持同步的SQL操作

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

支持的同步架构

- 一对一单向同步
- 一对多单向同步
- 级联单向同步
- 多对一单向同步

关于各类同步架构的介绍及注意事项，请参见[#unique_122](#)。

功能限制

- 不兼容触发器

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。相关解决方案请参见[#unique_123](#)。

- RENAME TABLE限制

RENAME TABLE操作可能导致同步数据不一致。例如同步对象只包含某个表，如果同步过程中源实例对该表执行了重命名操作，那么该表的数据将不会同步到目标库。为避免该问题，您可以在数据同步配置时将该表所属的整个数据库作为同步对象。

操作步骤

1. 购买数据同步作业，详情请参见[#unique_124](#)。



说明：

购买时，选择源实例和目标实例均为**POLARDB**，并选择同步拓扑为**单向同步**。

2. 登录[数据传输控制台](#)。
3. 在左侧导航栏，单击**数据同步**。
4. 在**同步作业列表**页面顶部，选择同步的目标实例所属地域。



5. 定位至已购买的数据同步实例，单击**配置同步链路**。

6. 配置同步通道的源实例及目标实例信息。

1. 选择同步通道的源及目标实例
2. 选择同步对象
3. 高级设置
4. 预检查

同步作业名称: POLARDB_TO_POLARDB

源实例信息

实例类型: PolarDB实例

实例地区: 华东1 (杭州)

* PolarDB实例ID: pc-bp[REDACTED]

* 数据库账号: dtstest

* 数据库密码: [REDACTED]

目标实例信息

实例类型: PolarDB

实例地区: 华东1 (杭州)

* PolarDB实例ID: pc-bp[REDACTED]

* 数据库账号: dtstest

* 数据库密码: [REDACTED]

取消
授权白名单并进入下一步

类别	配置	说明
无	同步作业名称	DTS会自动生成一个同步作业名称，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。
源实例信息	实例类型	固定为 PolarDB实例 ，不可变更。
	实例地区	购买数据同步实例时选择的源实例地域信息，不可变更。
	PolarDB实例ID	选择源PolarDB集群ID。
	数据库账号	填入连接PolarDB集群的数据库账号。  说明： 该账号需具备REPLICATION CLIENT、REPLICATION SLAVE、SHOW VIEW和所有同步对象的SELECT权限。
	数据库密码	填入数据库账号对应的密码。
目标实例信息	实例类型	固定为 PolarDB ，不可变更。
	实例地区	购买数据同步实例时选择的目标实例地域信息，不可变更。
	PolarDB实例ID	选择目标PolarDB集群ID。
	数据库账号	填入连接PolarDB集群的数据库账号。  说明： 用于数据同步的数据库账号需具备目标同步对象的ALL权限。

类别	配置	说明
	数据库密码	填入数据库账号对应的密码。

7. 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地址自动添加到源和目标PolarDB集群的白名单中，用于保障DTS服务器能够正常连接源和目标集群。

8. 配置目标已存在表的处理模式和同步对象。

1. 选择同步通道的源及目标实例

2. 选择同步对象

3. 高级设置

4. 预检查

同步架构: 单向同步
目标已存在表的处理模式: ☒ 预检查并报错拦截 ☐ 忽略报错并继续执行

源库对象

若全局搜索, 请先展开树

dtstestdata

Tables

vip

polardbdatabase

全选

已选择对象 (鼠标移到对象行, 点击编辑可修改对象名或过滤条件) 详情点我

dtstestdata (1个对象)

order

全选

映射名称更改:

☒ 不进行库表名称批量更改
☐ 要进行库表名称批量更改

取消

上一步

下一步

配置项目	配置说明
目标已存在表的处理模式	预检查并报错拦截：检查目标数据库中是否有同名的表。如果目标数据库中沒有同名的表，则通过该检查项目；如果目标数据库中有同名的表，则在预检查阶段提示错误，数据同步作业不会被启动。  说明： 如果目标库中同名的表不方便删除或重命名，您可以#unique_125来避免表名冲突。 忽略报错并继续执行：跳过目标数据库中是否有同名表的检查项。  警告： 选择为忽略报错并继续执行，可能导致数据不一致，给业务带来风险，例如： - 表结构一致的情况下，如果在目标库遇到与源库主键的值相同的记录，在初始化阶段会保留目标库中的该条记录；在增量同步阶段则会覆盖目标库的该条记录。 - 表结构不一致的情况下，可能会导致无法初始化数据、只能同步部分列的数据或同步失败。

配置项目	配置说明
选择同步对象	在源库对象框中单击待同步的对象，然后单击图标将其移动至已选择对象框。 同步对象的选择粒度为库、表。  说明： - 如果选择整个库作为同步对象，那么该库中所有对象的结构变更操作都会同步至目标库。 - 默认情况下，同步对象的名称保持不变。如果您需要改变同步对象在目标集群中的名称，请使用对象名映射功能，详情请参见#unique_125。

9. 上述配置完成后，单击页面右下角的**下一步**。

10. 配置同步初始化的高级配置信息。

创建同步作业 返回数据同步列表

1. 选择同步通道的源及目标实例
2. 选择同步对象
3. 高级设置
4. 预检查

同步初始化：
☒ 结构初始化
☒ 全量数据初始化

取消 上一步 保存 预检查并启动



说明：

同步初始化类型细分为：结构初始化，全量数据初始化。选中**结构初始化**和**全量数据初始化**后，DTS会在增量数据同步之前，将源数据库中待同步对象的结构和存量数据，同步到目标数据库。

11. 上述配置完成后，单击页面右下角的**预检查并启动**。



说明：

- 在数据同步作业正式启动之前，会先进行预检查。只有预检查通过后，才能成功启动数据同步作业。
- 如果预检查失败，单击具体检查项后的图标，查看失败详情。根据提示修复后，重新进行预检查。

12. 在**预检查**对话框中显示**预检查通过**后，关闭**预检查**对话框，同步作业将正式开始。

13.等待同步作业的链路初始化完成，直至处于**同步中**状态。

您可以在**数据同步**页面，查看数据同步作业的状态。

同步作业名称

搜索

排序：默认排序

状态：全部

☐	实例ID/作业名称	状态	同步概况	付费方式	同步架构(全部)	操作
☐	hangzhou-hangzhou-small	同步中	延时：565 毫秒 速度：0TPS(0.00MB/s)	按量付费	单向同步	暂停同步 转包年包月 升级更多
☐	暂停同步 释放同步	共有1页， 每页显示：20条				

5.7.2 PolarDB MySQL集群间的双向同步

数据传输服务DTS（Data Transmission Service）支持两个PolarDB MySQL集群间的双向数据同步，适用于异地多活、异地容灾等多种应用场景。本文介绍双向数据同步的配置步骤。

前提条件

- 已购买源和目标PolarDB MySQL集群（简称PolarDB集群），详情请参见[创建PolarDB MySQL集群](#)。
- 源和目标PolarDB集群已开启Binlog，详情请参见[如何开启Binlog](#)。

注意事项

- DTS在执行全量数据初始化时将占用源库和目标库一定的读写资源，可能会导致数据库的负载上升，在数据库性能较差、规格较低或业务量较大的情况下（例如源库有大量慢SQL、存在无主键表或目标库存在死锁等），可能会加重数据库压力，甚至导致数据库服务不可用。因此您需要在执行数据同步前评估源库和目标库的性能，同时建议您在业务低峰期执行数据同步（例如源库和目标库的CPU负载在30%以下）。
- 全量初始化过程中，并发INSERT会导致目标集群的表碎片，全量初始化完成后，目标集群的表空间比源集群的表空间大。
- 如果数据同步的源集群没有主键或唯一约束，且记录的全字段没有唯一性，可能会出现重复数据。

功能限制

- 目前DTS仅支持一对一双向同步，暂不支持多个PolarDB集群间的双向同步。
- 如果同步对象为单个或多个表（非整库），那么在数据同步时，请勿对源库的同步对象使用gh-ost或pt-online-schema-change等类似工具执行在线DDL变更，否则会导致同步失败。



注意：

为避免同步失败，数据同步期间您可以使用数据管理DMS（Data Management Service）来执行在线DDL变更，详情请参见[不锁表结构变更](#)。

- 不兼容触发器

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。相关解决方案请参见[#unique_123](#)。

- RENAME TABLE限制

RENAME TABLE操作可能导致同步数据不一致。例如同步对象只包含某个表，如果同步过程中源实例对该表执行了重命名操作，那么该表的数据将不会同步到目标库。为避免该问题，您可以在数据同步配置时将该表所属的整个数据库作为同步对象。

- DDL语法同步方向限制

为保障双向同步链路的稳定性，对于同一张表的DDL更新只能在其中一个同步方向进行同步。即一旦某个同步方向配置了DDL同步，则在反方向上不支持DDL同步，只进行DML同步。

支持同步的SQL操作

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

支持的冲突检测

为保障同步数据的一致性，您需要确保同一个主键、业务主键或唯一键的记录只在一个PolarDB集群上执行更新。如果发生了误操作，在两个PolarDB集群上均执行更新，那么将出现同步冲突。

DTS通过冲突检测和修复最大程度地维护双向同步实例的稳定性。目前DTS支持进行检测的冲突类型包括：

- INSERT导致的唯一性冲突

同步INSERT语句时违背了唯一性约束，例如双向同步的两个节点同时或者在极为接近的时间新增了某个主键值相同的记录，那么同步到对端时，会因为已经存在相同主键值的记录，导致INSERT操作的同步失败。

- UPDATE更新的记录不完全匹配
 - UPDATE要更新的记录在同步目标集群中不存在时，DTS会自动转化为INSERT，此时可能会出现唯一键的唯一性冲突。
 - UPDATE要更新的记录出现主键或唯一键冲突。

- DELETE对应的记录不存在

DELETE要删除的记录在同步的目标集群中不存在。出现这种冲突时，不论配置何种冲突修复策略，DTS都会自动忽略DELETE操作。



注意：

- 由于数据同步两端的系统时间可能存在差异、同步存在延时等多种因素，DTS无法完全保证冲突检测机制能够完全防止数据的冲突。在使用双向同步时，您需要在业务层面配合进行相应的改造，保证同一个主键、业务主键或唯一键的记录只在双向同步的某个节点进行更新。
- 对于上述数据同步的冲突，DTS提供了修复策略，您可以在配置双向同步时选择。

操作步骤

1. 购买数据同步作业，详情请参见[#unique_124](#)。



注意：

购买时，选择源实例和目标实例均为**PolarDB**，并选择同步拓扑为**双向同步**。

2. 登录[数据传输控制台](#)。
3. 在左侧导航栏，单击**数据同步**。
4. 在**同步作业列表**页面顶部，选择同步的目标实例所属地域。



5. 配置正向同步作业。
 - a) 定位至已购买的数据同步实例，单击该实例下第一个同步作业**操作列**的**配置同步链路**。



注意：

一个双向数据同步实例会包含两个同步作业，需要分别进行配置。在配置反向同步作业时，定位至第二个同步作业，单击操作列的**配置同步链路**。

同步作业名称

搜索

排序：默认排序

状态：全部

☐	实例ID/作业名称	状态	同步概况	付费方式	同步架构(全部)	操作
☐		--	--	按量付费	双向同步	转包年包月 升级 查看同步作业 更多

☐	作业名称	状态	同步概况	源实例/目标实例	操作
☐		未配置		未配置 未配置	配置同步链路
☐		未配置		未配置 未配置	配置同步链路

b) 配置同步作业的源集群及目标集群信息。

1.选择同步通道的源及目标实例

2.选择同步对象

3.高级设置

4.预检查

同步作业名称: PolarDB MySQL_Forward

源实例信息

实例类型: PolarDB实例

实例地区: 华东1 (杭州)

PolarDB实例ID: pc-bp

数据库账号: dtstest

数据库密码:

目标实例信息

实例类型: PolarDB实例

实例地区: 华东1 (杭州)

PolarDB实例ID: pc-bp

数据库账号: dtstest

数据库密码:

取消

授权白名单并进入下一步

类别	配置	说明
无	同步作业名称	DTS会自动生成一个同步作业名称，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。
源实例信息	实例类型	固定为 PolarDB实例 ，不可变更。
	实例地区	购买数据同步实例时选择的源实例地域信息，不可变更。
	PolarDB实例ID	选择源PolarDB集群ID。 注意： 稍后配置反向同步作业时，此处需选择为正向同步作业里的目标PolarDB集群ID。

类别	配置	说明
	数据库账号	填入连接PolarDB集群的数据库账号，该账号需具备待同步对象读写权限。
	数据库密码	填入数据库账号的密码。
目标实例信息	实例类型	固定为 PolarDB实例 ，不可变更。
	实例地区	购买数据同步实例时选择的目标实例地域信息，不可变更。
	PolarDB实例ID	选择目标PolarDB集群ID。  注意： 稍后配置反向同步作业时，此处需选择为正向同步作业里的源PolarDB集群ID。
	数据库账号	填入连接PolarDB集群的数据库账号，该账号需具备目标同步对象读写权限。

类别	配置	说明
	数据库密码	填入数据库账号的密码。

c) 单击页面右下角的**授权白名单**并进入下一步。

DTS会将DTS服务器的IP地址会自动添加到源和目标PolarDB集群的白名单中，用于保障DTS服务器能够正常连接源和目标集群。

d) 配置同步策略及对象信息。

类别	配置	说明
同步策略配置	是否过滤DDL	- 选择为是：不同步DDL操作。 - 选择为否：同步DDL操作。 注意： 一旦该同步方向选择同步DDL操作，那么同一张表在另一个同步方向则不支持同步DDL操作。
	DML同步类型	选择需要同步的DML类型，默认为 Insert 、 Update 、 Delete ，您可以根据业务需求调整。

类别	配置	说明
	冲突修复策略	选择同步冲突的修复策略，默认为TaskFailed，您可以根据业务情况选择合适的冲突修复策略。 TaskFailed（遇到冲突，任务报错退出） 默认的冲突修复策略。当数据同步遇到上述冲突类型时，同步任务直接报错并退出，同步任务进入失败状态，需要用户介入修复任务。 Ignore（遇到冲突，直接使用目标实例中的冲突记录） 当数据同步遇到上述的冲突类型时，直接跳过当前同步语句，继续往下执行，选择使用目标库中的冲突记录。 Overwrite（遇到冲突，直接覆盖目标实例中的冲突记录） 当数据同步遇到上述的冲突类型时，直接覆盖目标库中的冲突记录。
	目标已存在表的处理模式	预检查并报错拦截：检查目标库中是否有同名的表。如果目标库中没有同名的表，则通过该检查项目；如果目标库中有同名的表，则在预检查阶段提示错误，数据同步作业不会被启动。  注意： 如果目标库中同名的表不方便删除或重命名，您可以#unique_125来避免表名冲突。 忽略报错并继续执行：跳过目标库中是否有同名表的检查项。  警告： 选择为忽略报错并继续执行，可能导致数据不一致，给业务带来风险，例如： - 表结构一致的情况下，如果在目标库遇到与源库主键的值相同的记录，在初始化阶段会保留目标库中的该条记录；在增量同步阶段则会覆盖目标库的该条记录。 - 表结构不一致的情况下，可能会导致无法初始化数据、只能同步部分列的数据或同步失败。

类别	配置	说明
选择同步对象	无	在源库对象框中单击待同步的对象（选择粒度为库或表），然后单击将其移动到已选择对象框。  注意： - 如果选择整个库作为同步对象，该库中所有对象的结构变更操作都会同步至目标库。 - 默认情况下，同步对象的名称保持不变。如果您需要改变同步对象在目标库中的名称，需要使用对象名映射功能，详情请参见#unique_125。

e) 上述配置完成后，单击页面右下角的**下一步**。

f) 配置同步初始化的高级配置信息。

1.选择同步通道的源及目标实例
2.选择同步对象
3.高级设置
4.预检查

同步初始化: ☒ 结构初始化 ☒ 全量数据初始化 注：不支持trigger的同步，详情请[参考文档](#) (如果同步对象中部分表，在反向同步实例中已经进行了初始化，那么这部分表不会进行初始化)

取消 上一步 保存 **预检查并启动**



注意：

- 此步骤会将源集群中待同步对象的结构及数据同步至目标集群，作为后续增量同步数据的基线数据。同步初始化类型细分为：**结构初始化**、**全量数据初始化**。默认情况下，需要同时选择**结构初始化**和**全量数据初始化**。
- 在配置反向同步作业时，如果同步对象已经初始化至目标集群，则直接同步增量数据。

g) 单击页面右下角的**预检查并启动**。



说明：

- 在数据同步作业正式启动之前，会先进行预检查。只有预检查通过后，才能成功启动数据同步作业。
- 如果预检查失败，单击具体检查项后的图标，查看失败详情。根据提示修复后，重新进行预检查。

h) 在**预检查**对话框中显示**预检查通过**后，关闭**预检查**对话框，正向同步作业将正式开始。

6. 等待正向同步作业完成初始化，直至处于**同步中**状态。

您可以在 **数据同步** 页面，查看数据同步作业的状态。

7. 配置反向同步作业。

a) 定位至第二个同步作业，单击**配置同步链路**。

☐	实例ID/作业名称	状态	同步概况	付费方式	同步架构(全部) ▾	操作
☐		--	--	按量付费	双向同步	转包年包月 升级更多 查看同步作业 ▾
☐	作业名称	状态	同步概况	源实例/目标实例		操作
☐	PolarDB MySQL_Forward	同步中	延时: 0 毫秒 速度: 0TPS(0.00MB/s)	pc-bp- t5b pc-bp- 3hf		暂停同步 更多
☐	hangzhou-hangzhou-medium	未配置		pc-bp- 3hf pc-bp- 5b		配置同步链路

b) 重复**步骤5**中的ii到viii配置步骤，完成反向同步作业的配置。

预期结果

等待一段时间后，两个同步作业的链路状态均会处于**同步中**。

同步作业ID		搜索	排序: 默认排序	状态: 全部	
☐ 实例ID/作业名称	状态	同步概况	付费方式	同步架构(全部)	操作
☐	--	--	按量付费	双向同步	转包年包月 查看同步作业 升级更多
☐ 作业名称	状态	同步概况	源实例/目标实例		操作
☐ PolarDB MySQL_Forward	同步中	延时: 0 毫秒 速度: 0TPS(0.00MB/s)	pc-bp- t5b pc-bp- 3hf		暂停同步 更多
☐ PolarDB MySQL_Reverse	同步中	延时: 0 毫秒 速度: 0TPS(0.00MB/s)	pc-bp- 3hf pc-bp- t5b		暂停同步 更多
☐ 暂停同步 释放同步					
共有1条, 每页显示: 20条					<< < 1 > >>

5.8 PolarDB与其他数据库的数据同步

5.8.1 从RDS MySQL同步至PolarDB MySQL

PolarDB是阿里巴巴自主研发的下一代关系型分布式云原生数据库，可完全兼容MySQL，具备简单易用、高性能、高可靠、高可用等优势。通过数据传输服务DTS（Data Transmission Service），您可以将RDS MySQL同步至PolarDB MySQL。

前提条件

已创建PolarDB MySQL集群，详情请参见[创建PolarDB MySQL集群](#)。

注意事项

- 全量初始化过程中，并发INSERT会导致目标集群的表碎片，全量初始化完成后，目标集群的表空间比源库的表空间大。
- 如果数据同步的源库没有主键或唯一约束，且记录的全字段没有唯一性，可能会出现重复数据。

支持同步的SQL操作

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

支持的同步架构

- 一对一单向同步
- 一对多单向同步
- 级联单向同步
- 多对一单向同步

关于各类同步架构的介绍及注意事项，请参见[#unique_122](#)。

功能限制

- 不兼容触发器

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。相关解决方案请参见[#unique_123](#)。

- RENAME TABLE限制

RENAME TABLE操作可能导致同步数据不一致。例如同步对象只包含某个表，如果同步过程中源实例对该表执行了重命名操作，那么该表的数据将不会同步到目标库。为避免该问题，您可以在数据同步配置时将该表所属的整个数据库作为同步对象。

操作步骤

1. 购买数据同步作业，详情请参见[#unique_124](#)。



说明：

购买时，选择源实例为**MySQL**、目标实例为**POLARDB**，并选择同步拓扑为**单向同步**。

2. 登录[数据传输控制台](#)。
3. 在左侧导航栏，单击**数据同步**。

4. 在同步作业列表页面顶部，选择同步的目标实例所在地域。

5. 定位至已购买的数据同步实例，单击配置同步链路。

6. 配置同步通道的源实例及目标实例信息。

类别	配置	说明
无	同步作业名称	DTS会自动生成一个同步作业名称，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。
源实例信息	实例类型	选择 RDS实例 。
	实例地区	购买数据同步实例时选择的源实例地域信息，不可变更。
	数据库账号	填入连接RDS实例的数据库账号。  说明： - 该账号需具备REPLICATION CLIENT、REPLICATION SLAVE、SHOW VIEW和所有同步对象的SELECT权限。 - 当源RDS实例的数据库类型为MySQL 5.5或MySQL 5.6时，无需配置数据库账号和数据库密码。

类别	配置	说明
	数据库密码	填入数据库账号的密码。
	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 。
目标实例信息	实例类型	固定为 PolarDB ，不可变更。
	实例地区	购买数据同步实例时选择的目标实例地域信息，不可变更。
	PolarDB实例ID	选择目标PolarDB集群ID。
	数据库账号	填入PolarDB集群的数据库账号。  说明： 用于数据同步的数据库账号需具备目标同步对象的ALL权限。
	数据库密码	填入数据库账号的密码。

7. 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地址自动添加到源RDS实例和目标PolarDB集群的白名单中，用于保障DTS服务器能够正常连接源实例和目标集群。

8. 配置目标已存在表的处理模式和同步对象。

1. 选择同步通道的源及目标实例

2. 选择同步对象

3. 高级设置

4. 预检查

同步架构：单向同步

目标已存在表的处理模式：☒ 预检查并报错拦截 ☐ 忽略报错并继续执行

源库对象

若全局搜索，请先展开树

dtstestdata

Tables

vip

polardbdatabase

已选择对象 (鼠标移到对象行, 点击编辑可修改对象名或过滤条件) 详情点我

dtstestdata (1个对象)

order

全选

全选

映射名称更改：

☒ 不进行库表名称批量更改
 ☐ 要进行库表名称批量更改

取消

上一步

下一步

配置项目	配置说明
目标已存在表的处理模式	预检查并报错拦截：检查目标数据库中是否有同名的表。如果目标数据库中不存在同名的表，则通过该检查项目；如果目标数据库中有同名的表，则在预检查阶段提示错误，数据同步作业不会被启动。  说明： 如果目标库中同名的表不方便删除或重命名，您可以#unique_125来避免表名冲突。 忽略报错并继续执行：跳过目标数据库中是否有同名表的检查项。  警告： 选择为忽略报错并继续执行，可能导致数据不一致，给业务带来风险，例如： - 表结构一致的情况下，如果在目标库遇到与源库主键的值相同的记录，在初始化阶段会保留目标库中的该条记录；在增量同步阶段则会覆盖目标库的该条记录。 - 表结构不一致的情况下，可能会导致无法初始化数据、只能同步部分列的数据或同步失败。

86

文档版本：20200709

配置项目	配置说明
选择同步对象	在源库对象框中单击待同步的对象，然后单击图标将其移动至已选择对象框。 同步对象的选择粒度为库、表。  说明： - 如果选择整个库作为同步对象，那么该库中所有对象的结构变更操作都会同步至目标库。 - 默认情况下，同步对象的名称保持不变。如果您需要改变同步对象在目标集群中的名称，请使用对象名映射功能，详情请参见#unique_125。

9. 上述配置完成后，单击页面右下角的**下一步**。

10. 配置同步初始化的高级配置信息。

创建同步作业 返回数据同步列表

1. 选择同步通道的源及目标实例
2. 选择同步对象
3. 高级设置
4. 预检查

同步初始化：
☒ 结构初始化
☒ 全量数据初始化

取消 上一步 保存 预检查并启动



说明：

同步初始化类型细分为：结构初始化，全量数据初始化。选中**结构初始化**和**全量数据初始化**后，DTS会在增量数据同步之前，将源数据库中待同步对象的结构和存量数据，同步到目标数据库。

11. 上述配置完成后，单击页面右下角的**预检查并启动**。



说明：

- 在数据同步作业正式启动之前，会先进行预检查。只有预检查通过后，才能成功启动数据同步作业。
- 如果预检查失败，单击具体检查项后的图标，查看失败详情。根据提示修复后，重新进行预检查。

12. 在**预检查**对话框中显示**预检查通过**后，关闭**预检查**对话框，同步作业将正式开始。

13.等待同步作业的链路初始化完成，直至处于**同步中**状态。

您可以在**数据同步**页面，查看数据同步作业的状态。

同步作业名称

搜索

排序：默认排序

状态：全部

☐ 实例ID/作业名称	状态	同步概况	付费方式	同步架构(全部)	操作
☐ hangzhou-hangzhou-small	同步中	延时：565 毫秒 速度：0TPS(0.00MB/s)	按量付费	单向同步	暂停同步 转包年包月 升级更多
☐	暂停同步 释放同步	共有1条，每页显示：20条			

5.8.2 从PolarDB MySQL同步至RDS MySQL

本文介绍如何使用数据传输服务DTS（Data Transmission Service），将PolarDB MySQL同步至RDS MySQL，实现增量数据的实时同步。

前提条件

- 已购买PolarDB MySQL集群，详情请参见[创建PolarDB MySQL集群](#)。
- PolarDB MySQL集群已开启Binlog，详情请参见[如何开启Binlog](#)。

注意事项

- 全量初始化过程中，并发INSERT会导致目标集群的表碎片，全量初始化完成后，目标集群的表空间比源库的表空间大。
- 如果数据同步的源库没有主键或唯一约束，且记录的全字段没有唯一性，可能会出现重复数据。

支持同步的SQL操作

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

支持的同步架构

- 一对一单向同步
- 一对多单向同步
- 级联单向同步
- 多对一单向同步

关于各类同步架构的介绍及注意事项，请参见[#unique_122](#)。

功能限制

- 不兼容触发器

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。相关解决方案请参见[#unique_123](#)。

- RENAME TABLE限制

RENAME TABLE操作可能导致同步数据不一致。例如同步对象只包含表A，如果同步过程中源实例将表A重命名为表B，那么表B将不会被同步到目标库。为避免该问题，您可以在数据同步配置时，选择同步表A和表B所在的整个数据库作为同步对象。

操作步骤

1. 购买数据同步作业，详情请参见[#unique_124](#)。



说明：

购买时，选择源实例为**POLARDB**，选择目标实例为**MySQL**，并选择同步拓扑为**单向同步**。

2. 登录[数据传输控制台](#)。
3. 在左侧导航栏，单击**数据同步**。
4. 在**同步作业列表**页面顶部，选择同步的目标实例所属地域。



5. 定位至已购买的数据同步实例，单击**配置同步链路**。

6. 配置同步通道的源实例及目标实例信息。

1.选择同步通道的源及目标实例
2.选择同步对象
3.高级设置
4.预检查

同步作业名称: POLARDB源

源实例信息

实例类型: PolarDB实例

实例地区: 华东1 (杭州)

* PolarDB实例ID: pc-
* 数据库账号: dtstest
* 数据库密码: *****

目标实例信息

实例类型: RDS实例

实例地区: 华东1 (杭州)

* 实例ID: rm-
* 数据库账号: dtstest
* 数据库密码: *****

* 连接方式: ☒ 非加密连接 ☐ SSL安全连接

取消 授权白名单并进入下一步

类别	配置	说明
无	同步作业名称	DTS会自动生成一个同步作业名称，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。
源实例信息	实例类型	固定为 PolarDB实例 ，不可变更。
	实例地区	购买数据同步实例时选择的源实例地域信息，不可变更。
	PolarDB实例ID	选择源PolarDB集群ID。
	数据库账号	填入源PolarDB集群的数据库账号。
	数据库密码	填入数据库账号对应的密码。
目标实例信息	实例类型	选择为 RDS实例 。
	实例地区	购买数据同步实例时选择的目标实例地域信息，不可变更。
	数据库账号	填入目标RDS实例的数据库账号。
	数据库密码	填入数据库账号对应的密码。
		 说明： 当目标RDS实例的数据库类型为 MySQL 5.5 或 MySQL 5.6 时，无需配置 数据库账号 和 数据库密码 。

类别	配置	说明
	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 。

7. 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地址自动添加到源PolarDB集群和目标RDS实例的白名单中，用于保障DTS服务器能够正常连接源集群和目标实例。

8. 配置目标已存在表的处理模式和同步对象。

1. 选择同步通道的源及目标实例

2. 选择同步对象

3. 高级设置

4. 预检查

同步架构: 单向同步

目标已存在表的处理模式: ☒ 预检查并报错拦截 ☐ 忽略报错并继续执行

源库对象

若全局搜索, 请先展开树

dtstestdata

Tables

vip

polardbdatabase

已选择对象 (鼠标移到对象行, 点击编辑可修改对象名或过滤条件) 详情点我

dtstestdata (1个对象)

order

全选

全选

映射名称更改:

☒ 不进行库表名称批量更改
 ☐ 要进行库表名称批量更改

取消

上一步

下一步

配置项目	配置说明
目标已存在表的处理模式	预检查并报错拦截：检查目标数据库中是否有同名的表。如果目标数据库中不存在同名的表，则通过该检查项目；如果目标数据库中有同名的表，则在预检查阶段提示错误，数据同步作业不会被启动。  说明： 如果目标库中同名的表不方便删除或重命名，您可以#unique_125来避免表名冲突。 忽略报错并继续执行：跳过目标数据库中是否有同名表的检查项。  警告： 选择为忽略报错并继续执行，可能导致数据不一致，给业务带来风险，例如： - 表结构一致的情况下，如果在目标库遇到与源库主键的值相同的记录，在初始化阶段会保留目标库中的该条记录；在增量同步阶段则会覆盖目标库的该条记录。 - 表结构不一致的情况下，可能会导致无法初始化数据、只能同步部分列的数据或同步失败。

92

文档版本：20200709

配置项目	配置说明
选择同步对象	在源库对象框中单击待同步的对象，然后单击图标将其移动至已选择对象框。 同步对象的选择粒度为库、表。  说明： - 如果选择整个库作为同步对象，那么该库中所有对象的结构变更操作都会同步至目标库。 - 默认情况下，同步对象的名称保持不变。如果您需要改变同步对象在目标集群中的名称，请使用对象名映射功能，详情请参见#unique_125。

9. 上述配置完成后，单击页面右下角的**下一步**。

10. 配置同步初始化的高级配置信息。

创建同步作业 返回数据同步列表

1. 选择同步通道的源及目标实例
2. 选择同步对象
3. 高级设置
4. 预检查

同步初始化：
☒ 结构初始化
☒ 全量数据初始化

取消 上一步 保存 预检查并启动



说明：

同步初始化类型细分为：结构初始化，全量数据初始化。选中**结构初始化**和**全量数据初始化**后，DTS会在增量数据同步之前，将源数据库中待同步对象的结构和存量数据，同步到目标数据库。

11. 上述配置完成后，单击页面右下角的**预检查并启动**。



说明：

- 在数据同步作业正式启动之前，会先进行预检查。只有预检查通过后，才能成功启动数据同步作业。
- 如果预检查失败，单击具体检查项后的图标，查看失败详情。根据提示修复后，重新进行预检查。

12. 在**预检查**对话框中显示**预检查通过**后，关闭**预检查**对话框，同步作业将正式开始。

13.等待同步作业的链路初始化完成，直至处于**同步中**状态。

您可以在**数据同步**页面，查看数据同步作业的状态。

同步作业名称		搜索	排序：默认排序	状态：全部	
☐ 实例ID/作业名称	状态	同步概况	付费方式	同步架构(全部)	操作
☐ hangzhou-hangzhou-small	同步中	延时：565 毫秒 速度：0TPS(0.00MB/s)	按量付费	单向同步	暂停同步 转包年包月 升级更多
☐ 暂停同步 释放同步					
共有1条，每页显示：20条					

5.8.3 从PolarDB MySQL同步至AnalyticDB for MySQL

分析型数据库MySQL版（AnalyticDB for MySQL），是阿里巴巴自主研发的海量数据实时高并发在线分析（Realtime OLAP）云计算服务，可以对千亿级数据进行毫秒级的即时多维分析透视和业务探索。通过数据传输服务DTS（Data Transmission Service），您可以将PolarDB MySQL同步到AnalyticDB for MySQL，帮助您快速构建企业内部BI、交互查询、实时报表等系统。

前提条件

- 已创建目标AnalyticDB for MySQL集群，详情请参见[创建AnalyticDB for MySQL（2.0）](#)或[创建AnalyticDB for MySQL（3.0）](#)。



说明：

不支持将华北1（青岛）、美国东部1（弗吉尼亚）、英国（伦敦）地域的AnalyticDB MySQL（2.0）集群作为同步的目标集群；不支持将美国西部1（硅谷）地域的AnalyticDB for MySQL（3.0）集群作为同步的目标集群。

- 确保目标AnalyticDB for MySQL具备充足的存储空间。
- PolarDB MySQL已开启Binlog，详情请参见[如何开启Binlog](#)。
- 如果同步的目标为AnalyticDB for MySQL（2.0），源库中待同步的对象不能包含AnalyticDB for MySQL（2.0）[保留的库名和列名](#)，否则将造成数据同步失败或DDL操作同步失败。

注意事项

- 请勿在数据同步时，对源库的同步对象使用gh-ost或pt-online-schema-change等类似工具执行在线DDL变更，否则会导致同步失败。



说明：

如果同步的目标为AnalyticDB for MySQL（3.0），您可以使用[DMS企业版](#)提供的相关功能来执行在线DDL变更，详情请参见[不锁表结构变更](#)。

- 由于AnalyticDB for MySQL（3.0）本身的使用限制，当AnalyticDB for MySQL（3.0）集群中的节点磁盘空间使用量超过80%，该集群将被锁定。请提前根据待同步的对象预估所需空间，确保目标集群具备充足的存储空间。

- 暂不支持同步前缀索引，如果源库存在前缀索引可能导致数据迁移失败。

术语/概念对应关系

PolarDB MySQL	AnalyticDB for MySQL
数据库	- AnalyticDB for MySQL (2.0)：表组 - AnalyticDB for MySQL (3.0)：数据库
表	- AnalyticDB for MySQL (2.0)：表 - AnalyticDB for MySQL (3.0)：表



说明：

关于AnalyticDB for MySQL中表组和表的相关介绍，请参见[#unique_127](#)。

支持同步的SQL操作

目标数据库版本	支持的SQL操作
AnalyticDB for MySQL 2.0	- DDL操作：ADD COLUMN - DML操作：INSERT、UPDATE、DELETE
AnalyticDB for MySQL 3.0	- DDL操作：CREATE TABLE、DROP TABLE、RENAME TABLE、TRUNCATE TABLE、ADD COLUMN、DROP COLUMN - DML操作：INSERT、UPDATE、DELETE  说明： 如果在数据同步的过程中变更了源表的字段类型，同步作业将报错并中断。您可以提交工单处理或手动修复，详情请参见#unique_128/unique_128_Connect_42_section_o6l_gcd_cqe。

数据库账号的权限要求

数据库	所需权限
PolarDB MySQL	待同步对象的读权限。
AnalyticDB for MySQL (2.0)	无需填写数据库账号信息，DTS会自动创建账号并授权。
AnalyticDB for MySQL (3.0)	读写权限。

数据库账号的创建和授权方法，请参见[创建PolarDB MySQL数据库账号](#)或[创建AnalyticDB for MySQL数据库账号](#)。

数据类型映射关系

详情请参见[#unique_129](#)。

操作步骤

1. [#unique_124/unique_124_Connect_42_section_39h_fto_gdl](#)。



说明：

购买时，选择源实例为**POLARDB**、目标实例为**分析型数据库AnalyticDB**，并选择同步拓扑为**单向同步**。

2. 登录[数据传输控制台](#)。
3. 在左侧导航栏，单击**数据同步**。
4. 在**同步作业列表**页面顶部，选择同步的目标实例所属地域。



5. 定位至已购买的数据同步实例，单击**配置同步链路**。

6. 配置同步通道的源实例及目标实例信息。

1. 选择同步通道的源及目标实例
2. ADS账号授权
3. 选择同步对象
4. 预检查

同步作业名称: POLARDB_TO_AnalyticDB for MySQL

源实例信息

实例类型: PolarDB实例

实例地区: 华东1 (杭州)

* PolarDB实例ID: pc-bp

* 数据库账号: dtstest

* 数据库密码: *****

目标实例信息

实例类型: ADS

实例地区: 华东1 (杭州)

* 版本: 2.0 3.0

* 数据库: am-bp

* 数据库账号: dtstest

* 数据库密码: *****

取消 授权白名单并进入下一步

类别	配置	说明
无	同步作业名称	DTS会自动生成一个同步作业名称，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。
源实例信息	实例类型	固定为 PolarDB实例 ，不可变更。
	实例地区	购买数据同步实例时选择的源实例地域信息，不可变更。
	PolarDB实例ID	选择源PolarDB集群ID。
	数据库账号	填入PolarDB集群的数据库账号，权限要求请参见 数据库账号的权限要求 。
	数据库密码	填入数据库账号对应的密码。
目标实例信息	实例类型	固定为 ADS ，不可变更。
	实例地区	购买数据同步实例时选择的目标实例地域信息，不可变更。
	版本	根据目标AnalyticDB MySQL集群的版本，选择 2.0 或 3.0 。  说明： - 选择为2.0后，DTS将自动创建数据库账号并进行授权，您无需配置数据库账号和数据库密码。 - 选择为3.0后，您还需要配置数据库账号和数据库密码。
	数据库	选择目标AnalyticDB for MySQL的集群ID。

类别	配置	说明
	数据库账号	填入AnalyticDB for MySQL的数据库账号，权限要求请参见 数据库账号的权限要求 。
	数据库密码	填入数据库账号对应的密码。

7. 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PolarDB MySQL和AnalyticDB for MySQL的白名单中，用于保障DTS服务器能够正常连接源和目标集群。

8. 配置同步策略及对象信息。

1.选择同步通道的源及目标实例
2.ADS账号授权
3.选择同步对象
4.预检查

同步初始化: ☒ 结构初始化 ☒ 全量数据初始化
目标已存在表的处理模式: ☒ 预检查并报错拦截 ☐ 忽略报错并继续执行
多表归并: ☐ 是 ☒ 否
同步操作类型: ☒ Insert ☒ Update ☒ Delete ☒ Alter Table ☒ Truncate Table
☒ Create Table ☒ Drop Table ☒ Create Database ☒ Drop Database

源库对象
若全局搜索, 请先展开树

dtstestdata
Tables
name
vip
dtstestdata617
dtstestdatanew

已选择对象 (鼠标移到对象行, 点击编辑可修改对象名或过滤条件) 详情点我

dtstestdata (1个对象)
order

映射名称更改:
☒ 不进行库表名称批量更改
☐ 要进行库表名称批量更改

取消

上一步

下一步

配置	说明
同步初始化	默认情况下，您需要同时选中 结构初始化 和 全量数据初始化 。预检查完成后，DTS会将源实例中待同步对象的结构及数据在目标集群中初始化，作为后续增量同步数据的基线数据。

配置	说明
目标已存在表的处理模式	预检查并报错拦截：检查目标数据库中是否有同名的表。如果目标数据库中没有同名的表，则通过该检查项目；如果目标数据库中有同名的表，则在预检查阶段提示错误，数据同步作业不会被启动。  说明： 如果目标库中同名的表不方便删除或重命名，您可以更改该表在目标库中的名称，详情请参见#unique_125。 忽略报错并继续执行：跳过目标数据库中是否有同名表的检查项。  警告： 选择为忽略报错并继续执行，可能导致数据不一致，给业务带来风险，例如： - 表结构一致的情况下，在目标库遇到与源库主键的值相同的记录，则会保留目标集群中的该条记录，即源库中的该条记录不会同步至目标数据库中。 - 表结构不一致的情况下，可能会导致无法初始化数据、只能同步部分列的数据或同步失败。
多表归并	- 选择为是：DTS将在每个表中增加__dts_data_source列来存储数据来源，且不再支持DDL同步。 - 选择为否：默认选项，支持DDL同步。  说明： 多表归并功能基于任务级别，即不支持基于表级别执行多表归并。如果需要让部分表执行多表归并，另一部分不执行多表归并，您可以创建两个数据同步作业。
同步操作类型	根据业务选中需要同步的操作类型，默认情况下都处于选中状态。  说明： 目前仅支持INSERT、UPDATE、DELETE、ADD COLUMN。

配置	说明
选择同步对象	在源库对象框中单击待迁移的对象，然后单击图标将其移动至已选择对象框。 同步对象的选择粒度为库、表。  说明： - 如果选择整个库作为同步对象，那么该库中所有对象的结构变更操作会同步至目标库。 - 如果选择某个表作为同步对象，那么只有这个表的ADD COLUMN操作会同步至目标库。 - 默认情况下，同步对象的名称保持不变。如果您需要同步对象在目标集群上名称不同，请使用对象名映射功能，详情请参见#unique_125。

9. 上述配置完成后，单击页面右下角的**下一步**。

10. 设置待同步的表在目标库中类型。

1. 选择同步通道的源及目标实例		2. ADS账号授权		3. 选择同步对象		4. 预检查	
ADB表组	ADB表名	类型(全部)	主键列	分布列	定义状态(全部)		
dtstestdata	order	分区表	orderid	orderid	已定义		
全部设置为分区表 全部设置为维度表 请输入ADB表名 搜索				共有1条，每页显示：20条		取消 上一步 保存 预检查并启动	



说明：

选择了**结构初始化**后，您需要定义待同步的表在AnalyticDB for MySQL中的**类型**、**主键列**、**分区列**等信息，详情请参见[ADB 2.0 SQL手册](#)和[ADB 3.0 SQL手册](#)。

11. 上述配置完成后，单击页面右下角的**预检查并启动**。



说明：

- 在数据同步作业正式启动之前，会先进行预检查。只有预检查通过后，才能成功启动数据同步作业。
- 如果预检查失败，单击具体检查项后的图标，查看失败详情。根据提示修复后，重新进行预检查。

12. 在**预检查**对话框中显示**预检查通过**后，关闭**预检查**对话框，同步作业将正式开始。

13.等待同步作业的链路初始化完成，直至处于**同步中**状态。

您可以在**数据同步**页面，查看数据同步作业的状态。

同步作业名称		搜索	排序：默认排序	状态：全部	
☐	实例ID/作业名称	状态	同步概况	付费方式	同步架构(全部) 操作
☐	hangzhou-hangzhou-small	同步中	延时：565 毫秒 速度：0TPS(0.00MB/s)	按量付费	单向同步 暂停同步 转包年包月 升级更多
☐	暂停同步 释放同步				
共有1条，每页显示：20条					

5.8.4 从PolarDB MySQL同步至云原生数据仓库AnalyticDB PostgreSQL

云原生数据仓库AnalyticDB PostgreSQL版（原分析型数据库PostgreSQL版）为您提供简单、快速、经济高效的PB级云端数据仓库解决方案。通过数据传输服务DTS（Data Transmission Service），您可以将PolarDB MySQL同步至云原生数据仓库AnalyticDB PostgreSQL，帮助您快速实现对海量数据的即席查询分析、ETL处理和可视化探索。

前提条件

- PolarDB MySQL集群已开启Binlog，详情请参见[如何开启Binlog](#)。
- PolarDB MySQL集群中待同步的数据表必须具备主键。
- 已创建目标云原生数据仓库AnalyticDB PostgreSQL实例，详情请参见[创建云原生数据仓库AnalyticDB PostgreSQL实例](#)。

注意事项

- DTS在执行全量数据初始化时将占用源库和目标库一定的读写资源，可能会导致数据库的负载上升，在数据库性能较差、规格较低或业务量较大的情况下（例如源库有大量慢SQL、存在无主键表或目标库存在死锁等），可能会加重数据库压力，甚至导致数据库服务不可用。因此您需要在执行数据同步前评估源库和目标库的性能，同时建议您在业务低峰期执行数据同步（例如源库和目标库的CPU负载在30%以下）。
- 全量初始化过程中，并发INSERT会导致目标实例的表碎片，全量初始化完成后，目标实例的表空间比源集群的表空间大。

同步限制

- 同步对象仅支持数据表。
- 不支持BIT、VARBIT、GEOMETRY、ARRAY、UUID、TSQUERY、TSVECTOR、TXID_SNAPSHOT类型的数据同步。
- 在数据同步时，请勿对源库的同步对象使用gh-ost或pt-online-schema-change等类似工具执行在线DDL变更，否则会导致同步失败。

支持同步的SQL操作

- DML操作：INSERT、UPDATE、DELETE。
- DDL操作：ADD COLUMN、RENAME COLUMN。



说明：

不支持CREATE TABLE操作，如果您需要将新增的表作为同步对象，则需要执行[#unique_130](#)操作。

支持的同步架构

- 1对1单向同步。
- 1对多单向同步。
- 多对1单向同步。

术语对应关系

PolarDB MySQL	云原生数据仓库AnalyticDB PostgreSQL
Database	Schema
Table	Table

操作步骤

1. 购买数据同步作业，详情请参见[#unique_124](#)。



说明：

购买时，选择源实例为**POLARDB**、目标实例为**AnalyticDB for PostgreSQL**，并选择同步拓扑为**单向同步**。

2. 登录[数据传输控制台](#)。
3. 在左侧导航栏，单击**数据同步**。
4. 在**同步作业列表**页面顶部，选择同步的目标实例所属地域。



5. 定位至已购买的数据同步实例，单击**配置同步链路**。

6. 配置同步通道的源实例及目标实例信息。

1. 选择同步通道的源及目标实例
2. 选择同步对象
3. 预检查

同步作业名称: PolarDB MySQL_To_ADB for PG

源实例信息

实例类型: PolarDB实例
实例地区: 华东1 (杭州)
* PolarDB实例ID: pc-
* 数据库账号: dtstest
* 数据库密码:

目标实例信息

实例类型: AnalyticDB for PostgreSQL
实例地区: 华东1 (杭州)
* 实例ID: gp-
* 数据库名称: dtstestdata
* 数据库账号: dtstest
* 数据库密码:

取消 授权白名单并进入下一步

类别	配置	说明
无	同步作业名称	DTS会自动生成一个同步作业名称，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。
源实例信息	实例类型	固定为 PolarDB实例 。
	实例地区	购买数据同步实例时选择的源PolarDB集群的地域信息，不可变更。
	PolarDB实例ID	选择PolarDB集群ID。
	数据库账号	填入PolarDB集群的数据库账号。  说明： 该账号需具备待同步对象的读权限。
	数据库密码	填入该数据库账号的密码。
目标实例信息	实例类型	固定为 AnalyticDB for PostgreSQL ，无需设置。
	实例地区	购买数据同步实例时选择的目标实例地域信息，不可变更。
	实例ID	选择云原生数据仓库AnalyticDB PostgreSQL实例ID。

类别	配置	说明
	数据库名称	填入云原生数据仓库AnalyticDB PostgreSQL实例中，待同步的目标表所属的数据库名称。
	数据库账号	填入云原生数据仓库AnalyticDB PostgreSQL实例的数据库账号。  说明： 该账号需具备目标同步对象的ALL权限。
	数据库密码	填入数据库账号的密码。

7. 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PolarDB MySQL和云原生数据仓库AnalyticDB PostgreSQL的白名单中，用于保障DTS服务器能够正常连接源集群和目标实例。

8. 配置同步策略及同步对象。

1.选择同步通道的源及目标实例
2.选择同步对象
3.预检查

同步初始化: ☒ 结构初始化 ☒ 全量数据初始化
目标已存在表的处理模式: ☐ 清空目标表数据 ☐ 忽略报错并继续执行
同步操作类型: ☒ Insert ☒ Update ☒ Delete ☐ Alter Table

源库对象

若全局搜索, 请先展开树

public

testdata

Tables

全选

已选择对象 (鼠标移到对象行, 点击编辑可修改对象名或过滤条件) 详情点我

testdata (1个对象)

customer

全选

>
<

*映射名称更改: ☐ 不进行库表名称批量更改 ☐ 要进行库表名称批量更改

取消 上一步 预检查并启动

类别	配置	说明
同步策略配置	同步初始化	默认情况下, 您需要同时选中 结构初始化 和 全量数据初始化 。预检查完成后, DTS会将源实例中待同步对象的结构及数据在目标实例中初始化, 作为后续增量同步数据的基线数据。
	目标已存在表的处理模式	清空目标表的数据 在预检查阶段跳过同名对象存在性检查的检查项目。全量初始化之前将目标表的数据清空。适用于完成同步任务测试后的正式同步场景。 忽略报错并继续执行 在预检查阶段跳过同名对象存在性检查的检查项目。全量初始化时直接追加数据。适用于多张表同步到一张表的汇总同步场景。

类别	配置	说明
	同步操作类型	根据业务需求选择需要同步的操作类型： • Insert • Update • Delete • AlterTable
选择同步对象	无	在源库对象框中单击待同步的表，然后单击  图标将其移动至已选择对象框。  说明： • 同步对象的选择粒度为表。 • 如果需要目标表中的列名称与源表不同，需要使用DTS的字段映射功能，详情请参见#unique_125。

9. 设置待同步的表在中的主键列和分布列信息。

1. 选择同步通道的源及目标实例
2. 选择同步对象
3. 预检查

Schema	Table	主键列	分布列	定义状态(全部)
mysqltest	customer	id	id	已定义

请输入表名 搜索

共有1条， 每页显示：20条

« < 1 > »

取消 上一步 保存 预检查并启动



说明：

当您在上一步中选择了**结构初始化**才会出现该页面。关于主键列和分布列的详细说明，请参见[表的约束定义](#)和[表分布键定义](#)。

10. 上述配置完成后，单击页面右下角的**预检查并启动**。



说明：

- 在数据同步作业正式启动之前，会先进行预检查。只有预检查通过后，才能成功启动数据同步作业。
- 如果预检查失败，单击具体检查项后的  图标，查看失败详情。根据提示修复后，重新进行预检查。

11. 在**预检查**对话框中显示**预检查通过**后，关闭**预检查**对话框，同步作业将正式开始。

12.等待同步作业的链路初始化完成，直至处于**同步中**状态。

您可以在 **数据同步** 页面，查看数据同步作业的状态。

同步作业名称

搜索

排序：

默认排序

状态：

全部

☐ 实例ID/作业名称	状态	同步概况	付费方式	同步架构(全部)	操作
☐ hangzhou-hangzhou-small	同步中	延时：565 毫秒 速度：0TPS(0.00MB/s)	按量付费	单向同步	暂停同步 转包年包月 升级更多
☐ 暂停同步 释放同步					

共有1条，每页显示：20条

<<

<

1

>

>>

5.8.5 从PolarDB MySQL同步至Datahub

阿里云流式数据服务DataHub是流式数据（Streaming Data）的处理平台，提供对流式数据的发布、订阅和分发功能，让您可以轻松构建基于流式数据的分析和应用。通过数据传输服务（Data Transmission Service，简称DTS），您可以将PolarDB MySQL同步至DataHub，帮助您快速实现使用流计算等大数据产品对数据实时分析。

前提条件

- DataHub实例的地域为华东1、华东2、华北2或华南1。
- DataHub实例中，已创建用作接收同步数据的项目（Project），详情请参见[创建项目](#)。
- PolarDB MySQL集群已开启Binlog，详情请参见[如何开启Binlog](#)。
- PolarDB MySQL中待同步的表需具备主键或唯一约束。

功能限制

- 不支持全量数据初始化，即DTS不会将源PolarDB集群中同步对象的存量数据同步至目标DataHub实例。
- 仅支持表级别的数据同步。
- 不支持新增列的数据同步，即源数据表新增了某个列，该列的数据不会同步至目标DataHub实例。
- 数据同步的过程中，请勿对源库中待同步的表执行DDL变更，否则会导致同步失败。

支持同步的SQL操作

INSERT、UPDATE、DELETE。

操作步骤

1. 购买数据同步作业，详情请参见[#unique_124](#)。



说明：

购买时，选择源实例为**POLARDB**、目标实例为**DataHub**，并选择同步拓扑为**单向同步**。

2. 登录[数据传输控制台](#)。

3. 在左侧导航栏，单击**数据同步**。

4. 在**同步作业列表**页面顶部，选择同步的目标实例所属地域。

5. 定位至已购买的数据同步实例，单击**配置同步链路**。

6. 配置同步作业的源实例及目标实例信息。

类别	配置	说明
无	同步作业名称	DTS会自动生成一个同步作业名称，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。
源实例信息	实例类型	固定为 PolarDB实例 ，不可变更。
	实例地区	购买数据同步实例时选择的源实例地域信息，不可变更。
	PolarDB实例ID	选择源PolarDB集群ID。
	数据库账号	填入PolarDB集群的数据库账号。
	数据库密码	填入数据库账号对应的密码。
目标实例信息	实例类型	固定为 DataHub ，不可变更。
	实例地区	购买数据同步实例时选择的目标实例地域信息，不可变更。
	Project	选择DataHub实例的 Project 。

7. 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PolarDB集群的白名单中，用于保障DTS服务器能够正常连接PolarDB集群。

8. 配置同步策略和同步对象。

1.选择同步通道的源及目标实例

2.选择同步对象

3.预检查

同步初始化：☒ 结构初始化

源库对象

若全局搜索，请先展开树

sys

dtstestdata

Tables

全选

已选择对象 (鼠标移到对象行,点击编辑可修改对象名或过滤条件) 详情点我

dtstestdata (2个对象)

customer

order

全选

>

<

*映射名称更改：

☐ 不进行库表名称批量更改

☐ 要进行库表名称批量更改

*是否启用新的附加列规则

☐ 是

☐ 否

取消

上一步

预检查并启动

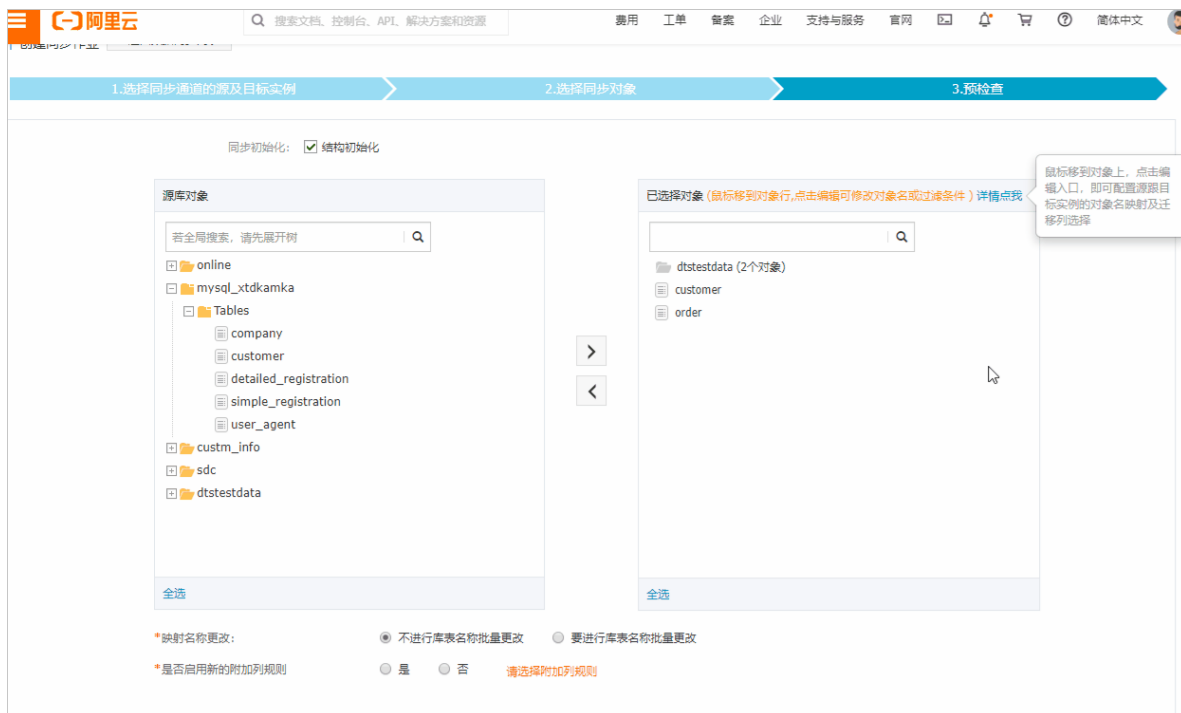
配置	说明
同步初始化	勾选结构初始化。 说明： 勾选结构初始化后，在数据同步作业的初始化阶段，DTS会将同步对象的结构信息（例如表结构）同步至目标DataHub实例。

文档版本：20200709

109

配置	说明
选择同步对象	在源库对象框中单击待迁移的对象，然后单击将其移动至已选择对象框。 说明： - 同步对象的选择粒度为表。 - 默认情况下，同步对象的名称保持不变。如果您需要改变同步对象在目标实例中的名称，需要使用对象名映射功能，详情请参见#unique_125。
选择附加列规则	DTS在将数据同步到DataHub时，会在同步的目标Topic中添加一些附加列。如果附加列和目标Topic中已有的列出现名称冲突将会导致数据同步失败。您需要根据业务需求选择是否启用新的附加列规则为是或否。 警告： 在选择附加列规则前，您需要评估附加列和目标Topic中已有的列是否会出现名称冲突。关于附加列的规则和定义说明，请参见附加列名称和定义说明。

9. （可选）将鼠标指针放置在**已选择对象**框中待同步的Topic名上，单击对象后出现的**编辑**，然后在弹出的对话框中设置Shardkey（即用于分区的key）。



10. 上述配置完成后，单击页面右下角的**预检查并启动**。



说明：

- 在数据同步作业正式启动之前，会先进行预检查。只有预检查通过后，才能成功启动数据同步作业。
- 如果预检查失败，单击具体检查项后的图标，查看失败详情。根据提示修复后，重新进行预检查。

11. 在**预检查**对话框中显示**预检查通过**后，关闭**预检查**对话框，同步作业将正式开始。

12. 等待同步作业的链路初始化完成，直至处于**同步中**状态。

您可以在**数据同步**页面，查看数据同步作业的状态。

同步作业名称	状态	同步概况	付费方式	同步架构(全部)	操作
☐ hangzhou-hangzhou-small	同步中	延时：565 毫秒 速度：0TPS(0.00MB/s)	按量付费	单向同步	暂停同步 转包年包月 升级更多
☐ 暂停同步	☐ 释放同步	共有1条，每页显示：20条			

Topic结构定义说明

DTS在将数据变更同步至DataHub实例的Topic时，目标Topic中除了存储变更数据外，还会新增一些附加列用于存储元信息，示例如下。



说明：

本案例中的业务字段为id、name、address，由于在配置数据同步时选用的是旧版附加列规则，DTS会为业务字段添加dts_的前缀。

dts_id	dts_name	dts_address	dts_record_id	dts_operation_flag	dts_instance_id	dts_db_name	dts_table_name	dts_utc_timestamp	dts_before_flag	dts_after_flag
10006	曲靖市	1574832130000000000	U		dtstestdata	customer	1574832130	Y	N	
10006	杭州市	1574832130000000000	U		dtstestdata	customer	1574832130	N	Y	
10009	马鞍山市	1574832919000000000	D		dtstestdata	customer	1574832919	Y	N	
10112	北京市	1574832919000000000	I		dtstestdata	customer	1574832919	N	Y	

结构定义说明：

旧版附加列名称	新版附加列名称	说明
dt _s _record_id	new_dts_sync_dts_record_id	增量日志的记录id，为该日志唯一标识。  说明： - id的值唯一且递增。 - 如果增量日志的操作类型为UPDATE，那么增量更新会被拆分成两条记录，一条为DELETE，一条为INSERT，并且这两条记录的dt_s_record_id的值相同。
dt _s _operation_flag	new_dts_sync_dts_operation_flag	操作类型，取值： - I：INSERT操作。 - D：DELETE操作。 - U：UPDATE操作。
dt _s _instance_id	new_dts_sync_dts_instance_id	数据库的server id。
dt _s _db_name	new_dts_sync_dts_db_name	数据库名称。
dt _s _table_name	new_dts_sync_dts_table_name	表名。
dt _s _utc_timestamp	new_dts_sync_dts_utc_timestamp	操作时间戳，即binlog的时间戳（UTC时间）。
dt _s _before_flag	new_dts_sync_dts_before_flag	所有列的值是否更新前的值，取值：Y或N。
dt _s _after_flag	new_dts_sync_dts_after_flag	所有列的值是否更新后的值，取值：Y或N。

关于dt_s_before_flag和dt_s_after_flag的补充说明

对于不同的操作类型，增量日志中的dt_s_before_flag和dt_s_after_flag定义如下：

- INSERT

当操作类型为INSERT时，所有列的值为新插入的值，即为更新后的值，所以dt_s_before_flag取值为N，dt_s_after_flag取值为Y，示例如下。

dt _s _id	dt _s _name	dt _s _address	dt _s _record_id	dt _s _operation_flag	dt _s _instance_id	dt _s _db_name	dt _s _table_name	dt _s _utc_timestamp	dt _s _before_flag	dt _s _after_flag
10112		北京市	1574832919000000000	I		dts_testdata	customer	1574832919	N	Y

- UPDATE

当操作类型为UPDATE时，DTS会将UPDATE操作拆为两条增量日志。这两条增量日志的 dts_record_id、dts_operation_flag及dts_utc_timestamp对应的值相同。

第一条增量日志记录了更新前的值，所以dts_before_flag取值为Y，dts_after_flag取值为N。第二条增量日志记录了更新后的值，所以 dts_before_flag取值为N，dts_after_flag取值为Y，示例如下。

dts_id	dts_name	dts_address	dts_record_id	dts_operation_flag	dts_instance_id	dts_db_name	dts_table_name	dts_utc_timestamp	dts_before_flag	dts_after_flag
10006		曲靖市	1574832130000000000	U		dtstestdata	customer	1574832130	Y	N
10006		杭州市	1574832130000000000	U		dtstestdata	customer	1574832130	N	Y

- DELETE

当操作类型为DELETE时，增量日志中所有的列值为被删除的值，即列值不变，所以dts_before_flag取值为Y，dts_after_flag取值为N，示例如下。

dts_id	dts_name	dts_address	dts_record_id	dts_operation_flag	dts_instance_id	dts_db_name	dts_table_name	dts_utc_timestamp	dts_before_flag	dts_after_flag
10009		马鞍山市	1574832919000000000	D		dtstestdata	customer	1574832919	Y	N

后续操作

配置完数据同步作业后，您可以对同步到DataHub实例中的数据执行计算分析。更多详情，请参见[阿里云实时计算](#)。

5.8.6 从PolarDB MySQL同步到Kafka

Kafka是应用较为广泛的分布式、高吞吐量、高可扩展性消息队列服务，普遍用于日志收集、监控数据聚合、流式数据处理、在线和离线分析等大数据领域，是大数据生态中不可或缺的产品之一。通过数据传输服务DTS（Data Transmission Service），您可以将PolarDB MySQL同步至自建Kafka集群，扩展消息处理能力。

前提条件

- Kafka集群的版本为0.10或1.0版本。
- Kafka集群已创建用于存储待同步数据的Topic。
- PolarDB MySQL已开启Binlog，详情请参见[如何开启Binlog](#)。

注意事项

如果源数据库没有主键或唯一约束，且所有字段没有唯一性，可能会导致目标数据库中出现重复数据。

功能限制

- 仅支持表粒度的数据同步。

- 不支持自动调整同步对象。

**说明：**

如果在同步的过程中，对源库中待同步的表执行了重命名操作，且重命名后的名称不在同步对象中，那么该表将不再被同步到目标Kafka集群中。如果该表还需要同步，那么您需要[#unique_130](#)。

消息格式

同步到Kafka集群中的数据以avro格式存储，schema定义详情请参见[DTS avro schema定义](#)。

在数据同步到Kafka集群后，您需要根据avro schema定义进行数据解析。

操作步骤

1. 购买数据同步作业，详情请参见[#unique_124](#)。

**说明：**

购买时，选择源实例为**POLARDB**、目标实例为**Kafka**，并选择同步拓扑为**单向同步**。

2. 登录[数据传输控制台](#)。
3. 在左侧导航栏，单击**数据同步**。
4. 在**同步作业列表**页面顶部，选择同步的目标实例所属地域。



5. 定位至已购买的数据同步实例，单击**配置同步链路**。

6. 配置同步通道的源实例及目标实例信息。

同步作业名称: PolarDB_TO_Kafka

源实例信息

实例类型: PolarDB实例
实例地区: 华东1 (杭州)
* PolarDB实例ID: pc-bp-
* 数据库账号: dtstest
* 数据库密码:

目标实例信息

实例类型: ECS上的自建数据库
实例地区: 华东1 (杭州)
* ECS实例ID: i-bp-
数据库类型: Kafka
* 端口: 9092
数据库账号: dtstest 非必填项
数据库密码: 非必填项
* Topic: customer 获取Topic列表
请先点击右侧按钮, 获取Topic列表后选择具体的Topic
* Kafka版本: 0.10
* 连接方式: ☒ 非加密连接 ☐ SCRAM-SHA-256

取消 授权白名单并进入下一步

类别	配置	说明
无	同步作业名称	DTS会自动生成一个同步作业名称, 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。
源实例信息	实例类型	固定为 PolarDB实例 , 不可变更。
	实例地区	购买数据同步实例时选择的源实例地域信息, 不可变更。
	PolarDB实例ID	选择PolarDB集群ID。
	数据库账号	填入PolarDB集群的数据库账号, 需要具备待同步数据库的读权限。
	数据库密码	填入该数据库账号的密码。
目标实例信息	实例类型	- Kafka集群部署在ECS上时, 选择ECS上的自建数据库。 - Kafka集群部署在本地服务器时, 选择通过专线/VPN网关/智能接入网关接入的自建数据库。  说明: 选择通过专线/VPN网关/智能接入网关接入的自建数据库时, 您需要配置VPC ID并填写IP地址和端口信息。
	实例地区	购买数据同步实例时选择的目标实例地域信息, 不可变更。

类别	配置	说明
	ECS实例ID	选择部署了Kafka集群的ECS实例ID。
	数据库类型	选择为 Kafka 。
	端口	Kafka集群对外提供服务的端口，默认为9092。
	数据库账号	填入Kafka集群的用户名，如Kafka集群未开启验证可不填写。
	数据库密码	填入Kafka集群用户名对应的密码，如Kafka集群未开启验证可不填写。
	Topic	单击右侧的 获取Topic列表 ，然后在下拉框中选择具体的Topic。
	Kafka版本	根据目标Kafka集群版本，选择对应的版本信息。
	连接方式	根据业务及安全需求，选择 非加密连接 或 SCRAM-SHA-256 。

7. 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地址自动添加到源PolarDB集群的白名单和目标ECS实例的内网入方向安全组规则中，用于保障DTS服务器能够正常连接源和目标实例。

8. 配置目标已存在表的处理模式和同步对象。

1.选择同步通道的源及目标实例

2.选择同步对象

3.高级设置

4.预检查

同步架构: 单向同步

源库对象

若全局搜索, 请先展开树

dtstest

dtstestdata

Tables

order

全选

已选择对象 (鼠标移到对象行, 点击编辑可修改对象名或过滤条件) 详情点我

polardb_test 源库名:dtstestdata (1个对象)

polardb_test 源表名:customer

全选

>

<

映射名称更改:

☒ 不进行库表名称批量更改

☐ 要进行库表名称批量更改

取消

上一步

下一步

配置	说明
选择同步对象	在源库对象框中单击待同步的对象, 然后单击图标将其移动至已选择对象框。  说明: - 仅支持表粒度的数据同步。 - 默认情况下, 同步对象的名称保持不变。如果您需要同步对象在目标实例上名称不同, 那么需要使用DTS提供的对象名映射功能, 详情请参见#unique_125。

9. 上述配置完成后, 单击页面右下角的**下一步**。

注意事项

- DTS在执行全量数据初始化时将占用源库和目标库一定的读写资源，可能会导致数据库的负载上升，在数据库性能较差、规格较低或业务量较大的情况下（例如源库有大量慢SQL、存在无主键表或目标库存在死锁等），可能会加重数据库压力，甚至导致数据库服务不可用。因此您需要在执行数据同步前评估源库和目标库的性能，同时建议您在业务低峰期执行数据同步（例如源库和目标库的CPU负载在30%以下）。
- 如果同步对象为单个或多个表（非整库），那么在数据同步时，请勿对源库的同步对象使用gh-ost或pt-online-schema-change等类似工具执行在线DDL变更，否则会导致同步失败。



注意：

为避免同步失败，数据同步期间您可以使用数据管理DMS（Data Management Service）来执行在线DDL变更，详情请参见[不锁表结构变更](#)。

- 全量初始化过程中，并发INSERT会导致目标集群的表碎片，全量初始化完成后，目标集群的表空间比源库的表空间大。
- 如果数据同步的源库没有主键或唯一约束，且记录的全字段没有唯一性，可能会出现重复数据。

支持同步的SQL操作

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

功能限制

- 不兼容触发器

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。相关解决方案请参见[#unique_123](#)。

- RENAME TABLE限制

RENAME TABLE操作可能导致同步数据不一致。例如同步对象只包含某个表，如果同步过程中源实例对该表执行了重命名操作，那么该表的数据将不会同步到目标库。为避免该问题，您可以在数据同步配置时将该表所属的整个数据库作为同步对象。

准备工作

#unique_116



说明：

用于数据同步的数据库账号需具备待同步对象的REPLICATION CLIENT、REPLICATION SLAVE、SHOW VIEW和所有同步对象的SELECT权限。

支持的同步架构

- 一对一单向同步
- 一对多单向同步
- 级联单向同步
- 多对一单向同步

关于各类同步架构的介绍及注意事项，请参见[#unique_122](#)。

操作步骤

1. 购买数据同步作业，详情请参见[#unique_124](#)。



说明：

购买时，选择源实例为**MySQL**、目标实例为**POLARDB**，并选择同步拓扑为**单向同步**。

2. 登录[数据传输控制台](#)。
3. 在左侧导航栏，单击**数据同步**。
4. 在**同步作业列表**页面顶部，选择同步的目标实例所属地域。



5. 定位至已购买的数据同步实例，单击**配置同步链路**。

6. 配置同步通道的源实例及目标实例信息。

1. 选择同步通道的源及目标实例
2. 选择同步对象
3. 高级设置
4. 预检查

同步作业名称: MySQL_TO_POLARDB

源实例信息

实例类型: ECS上的自建数据库

实例地区: 华东1 (杭州)

* ECS实例ID: i-bp

数据库类型: MySQL

* 端口: 3306

* 数据库账号: dtstest

* 数据库密码: *****

目标实例信息

实例类型: PolarDB

实例地区: 华东1 (杭州)

* PolarDB实例ID: pc-bp

* 数据库账号: dtstest

* 数据库密码: *****

取消 按默认名单并进入下一步

类别	配置	说明
无	同步作业名称	DTS会自动生成一个同步作业名称，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。
源实例信息	实例类型	选择 ECS上的自建数据库 。
	实例地区	购买数据同步实例时选择的源实例地域信息，不可变更。
	ECS实例ID	选择自建MySQL数据库所属的ECS实例ID。
	数据库类型	固定为 MySQL ，不可变更。
	端口	填入自建MySQL的数据库服务端口。
	数据库账号	填入连接自建MySQL的数据库账号。  说明： 用于数据同步的数据库账号需具备REPLICATION CLIENT、REPLICATION SLAVE、SHOW VIEW和所有同步对象的SELECT权限。
	数据库密码	填入数据库账号的密码。
目标实例信息	实例类型	固定为 PolarDB ，不可变更。
	实例地区	购买数据同步实例时选择的目标实例地域信息，不可变更。

类别	配置	说明
	PolarDB实例ID	选择目标PolarDB集群ID。
	数据库账号	填入连接PolarDB集群的数据库账号。  说明： 用于数据同步的数据库账号需具备目标同步对象的ALL权限。
	数据库密码	填入数据库账号的密码。

7. 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ECS实例的内网入方向安全组规则和目标PolarDB集群的白名单中，用于保障DTS服务器能够正常连接源实例和目标集群。

8. 配置目标已存在表的处理模式和同步对象。

1. 选择同步通道的源及目标实例

2. 选择同步对象

3. 高级设置

4. 预检查

同步架构: 单向同步
目标已存在表的处理模式: ☒ 预检查并报错拦截 ☐ 忽略报错并继续执行

源库对象

若全局搜索, 请先展开树

dtstestdata

Tables

vip

polardbdatabase

全选

已选择对象 (鼠标移到对象行, 点击编辑可修改对象名或过滤条件) 详情点我

dtstestdata (1个对象)

order

全选

映射名称更改:

☒ 不进行库表名称批量更改
☐ 要进行库表名称批量更改

取消

上一步

下一步

配置项目	配置说明
目标已存在表的处理模式	预检查并报错拦截：检查目标数据库中是否有同名的表。如果目标数据库中沒有同名的表，则通过该检查项目；如果目标数据库中有同名的表，则在预检查阶段提示错误，数据同步作业不会被启动。  说明： 如果目标库中同名的表不方便删除或重命名，您可以#unique_125来避免表名冲突。 忽略报错并继续执行：跳过目标数据库中是否有同名表的检查项。  警告： 选择为忽略报错并继续执行，可能导致数据不一致，给业务带来风险，例如： - 表结构一致的情况下，如果在目标库遇到与源库主键的值相同的记录，在初始化阶段会保留目标库中的该条记录；在增量同步阶段则会覆盖目标库的该条记录。 - 表结构不一致的情况下，可能会导致无法初始化数据、只能同步部分列的数据或同步失败。

配置项目	配置说明
选择同步对象	在源库对象框中单击待同步的对象，然后单击图标将其移动至已选择对象框。 同步对象的选择粒度为库、表。  说明： - 如果选择整个库作为同步对象，那么该库中所有对象的结构变更操作都会同步至目标库。 - 默认情况下，同步对象的名称保持不变。如果您需要改变同步对象在目标集群中的名称，请使用对象名映射功能，详情请参见#unique_125。

9. 上述配置完成后，单击页面右下角的**下一步**。

10. 配置同步初始化的高级配置信息。

创建同步作业 返回数据同步列表

1. 选择同步通道的源及目标实例
2. 选择同步对象
3. 高级设置
4. 预检查

同步初始化：
☒ 结构初始化
☒ 全量数据初始化

取消 上一步 保存 **预检查并启动**



说明：

同步初始化类型细分为：结构初始化，全量数据初始化。选中**结构初始化**和**全量数据初始化**后，DTS会在增量数据同步之前，将源数据库中待同步对象的结构和存量数据，同步到目标数据库。

11. 上述配置完成后，单击页面右下角的**预检查并启动**。



说明：

- 在数据同步作业正式启动之前，会先进行预检查。只有预检查通过后，才能成功启动数据同步作业。
- 如果预检查失败，单击具体检查项后的图标，查看失败详情。根据提示修复后，重新进行预检查。

12. 在**预检查**对话框中显示**预检查通过**后，关闭**预检查**对话框，同步作业将正式开始。

13.等待同步作业的链路初始化完成，直至处于**同步中**状态。

您可以在**数据同步**页面，查看数据同步作业的状态。

同步作业名称

搜索

排序：默认排序

状态：全部

☐	实例ID/作业名称	状态	同步概况	付费方式	同步架构(全部)	操作
☐	hangzhou-hangzhou-small	同步中	延时：565 毫秒 速度：0TPS(0.00MB/s)	按量付费	单向同步	暂停同步 转包年包月 升级更多
☐	暂停同步 释放同步	共有1条， 每页显示：20条 << < 1 > >>				

5.8.8 从通过专线、VPN网关或智能接入网关接入的自建MySQL同步至PolarDB MySQL

PolarDB是阿里巴巴自主研发的下一代关系型分布式云原生数据库，可完全兼容MySQL，具备简单易用、高性能、高可靠、高可用等优势。通过数据传输服务DTS（Data Transmission Service），您可以将自建的MySQL数据库同步至PolarDB MySQL，本文以通过专线、VPN网关或智能接入网关接入的自建MySQL为例介绍配置流程。

前提条件

- 自建MySQL数据库版本为5.1、5.5、5.6、5.7或8.0版本。
- 已经将自建MySQL数据库通过专线、VPN网关或智能接入网关接入至阿里云专有网络，详情请参见[#unique_133](#)。



说明：

接入至阿里云专有网络后，还需要允许DTS服务器的IP地址访问自建数据库所属的网络，详情请参见[#unique_134](#)。

- 已创建PolarDB MySQL集群，详情请参见[创建PolarDB MySQL集群](#)。



说明：

PolarDB MySQL集群的存储空间须大于自建MySQL数据库占用的存储空间。

注意事项

- DTS在执行全量数据初始化时将占用源库和目标库一定的读写资源，可能会导致数据库的负载上升，在数据库性能较差、规格较低或业务量较大的情况下（例如源库有大量慢SQL、存在无主键表或目标库存在死锁等），可能会加重数据库压力，甚至导致数据库服务不可用。因此您需要在执行数据同步前评估源库和目标库的性能，同时建议您在业务低峰期执行数据同步（例如源库和目标库的CPU负载在30%以下）。

- 如果同步对象为单个或多个表（非整库），那么在数据同步时，请勿对源库的同步对象使用gh-ost或pt-online-schema-change等类似工具执行在线DDL变更，否则会导致同步失败。

**注意：**

为避免同步失败，数据同步期间您可以使用数据管理DMS（Data Management Service）来执行在线DDL变更，详情请参见[不锁表结构变更](#)。

- 全量初始化过程中，并发INSERT会导致目标集群的表碎片，全量初始化完成后，目标集群的表空间比源库的表空间大。
- 如果数据同步的源集群没有主键或唯一约束，且记录的全字段没有唯一性，可能会出现重复数据。

支持同步的SQL操作

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

支持的同步架构

- 一对一单向同步
- 一对多单向同步
- 级联单向同步
- 多对一单向同步

关于各类同步架构的介绍及注意事项，请参见[#unique_122](#)。

功能限制

- 不兼容触发器

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。相关解决方案请参见[#unique_123](#)。

- RENAME TABLE限制

RENAME TABLE操作可能导致同步数据不一致。例如同步对象只包含某个表，如果同步过程中源实例对该表执行了重命名操作，那么该表的数据将不会同步到目标库。为避免该问题，您可以在数据同步配置时将该表所属的整个数据库作为同步对象。

准备工作

#unique_116



说明：

用于数据同步的数据库账号需具备REPLICATION CLIENT、REPLICATION SLAVE、SHOW VIEW和所有同步对象的SELECT权限。

操作步骤

1. 购买数据同步作业，详情请参见[#unique_124](#)。



说明：

购买时，选择源实例为**MySQL**、目标实例为**POLARDB**，并选择同步拓扑为**单向同步**。

2. 登录[数据传输控制台](#)。
3. 在左侧导航栏，单击**数据同步**。
4. 在**同步作业列表**页面顶部，选择同步的目标实例所属地域。



5. 定位至已购买的数据同步实例，单击**配置同步链路**。

6. 配置同步通道的源实例及目标实例信息。

1.选择同步通道的源及目标实例
2.选择同步对象
3.高级设置
4.预检查

同步作业名称: MySQL_TO_POLARDB

源实例信息

实例类型: 通过专线/VPN网关/智能网关接入的自建数据库

实例地区: 华东1 (杭州)

* 对端专有网络: vpc-

数据库类型: MySQL

* IP地址: 172.16

* 端口: 3306

* 数据库账号: dtstest

* 数据库密码: *****

目标实例信息

实例类型: PolarDB

实例地区: 华东1 (杭州)

* PolarDB实例ID: pc-bp

* 数据库账号: dtstest

* 数据库密码: *****

取消 授权白名单并进入下一步

类别	配置	说明
无	同步作业名称	DTS会自动生成一个同步作业名称，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。
源实例信息	实例类型	选择 通过专线/VPN网关/智能接入网关接入的自建数据库 。
	实例地区	购买数据同步实例时选择的源实例地域信息，不可变更。
	对端专有网络	选择自建数据库接入的VPC ID。
	数据库类型	固定为 MySQL ，不可变更。
	IP地址	填入自建MySQL数据库的服务器IP地址。
	端口	填入自建MySQL的数据库服务端口。
	数据库账号	填入为自建MySQL创建的数据库账号，详情请参见 准备工作 。
	数据库密码	填入该数据库账号的密码。
目标实例信息	实例类型	固定为 PolarDB ，不可变更。
	实例地区	购买数据同步实例时选择的目标实例地域信息，不可变更。
	PolarDB实例ID	选择目标PolarDB集群ID。

类别	配置	说明
	数据库账号	填入PolarDB集群的数据库账号。  说明： 该数据库账号需具备目标同步对象的ALL权限。
	数据库密码	填入该数据库账号的密码。

7. 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地址自动添加到目标PolarDB集群的白名单中，用于保障DTS服务器能够正常连接目标集群。

8. 配置目标已存在表的处理模式和同步对象。

1. 选择同步通道的源及目标实例

2. 选择同步对象

3. 高级设置

4. 预检查

同步架构: 单向同步
目标已存在表的处理模式: ☒ 预检查并报错拦截 ☐ 忽略报错并继续执行

源库对象

若全局搜索, 请先展开树

dtstestdata

Tables

vip

polardbdatabase

全选

已选择对象 (鼠标移到对象行, 点击编辑可修改对象名或过滤条件) 详情点我

dtstestdata (1个对象)

order

全选

映射名称更改:

☒ 不进行库表名称批量更改
☐ 要进行库表名称批量更改

取消

上一步

下一步

配置项目	配置说明
目标已存在表的处理模式	预检查并报错拦截：检查目标数据库中是否有同名的表。如果目标数据库中沒有同名的表，则通过该检查项目；如果目标数据库中有同名的表，则在预检查阶段提示错误，数据同步作业不会被启动。  说明： 如果目标库中同名的表不方便删除或重命名，您可以#unique_125来避免表名冲突。 忽略报错并继续执行：跳过目标数据库中是否有同名表的检查项。  警告： 选择为忽略报错并继续执行，可能导致数据不一致，给业务带来风险，例如： - 表结构一致的情况下，如果在目标库遇到与源库主键的值相同的记录，在初始化阶段会保留目标库中的该条记录；在增量同步阶段则会覆盖目标库的该条记录。 - 表结构不一致的情况下，可能会导致无法初始化数据、只能同步部分列的数据或同步失败。

配置项目	配置说明
选择同步对象	在源库对象框中单击待同步的对象，然后单击图标将其移动至已选择对象框。 同步对象的选择粒度为库、表。  说明： - 如果选择整个库作为同步对象，那么该库中所有对象的结构变更操作都会同步至目标库。 - 默认情况下，同步对象的名称保持不变。如果您需要改变同步对象在目标集群中的名称，请使用对象名映射功能，详情请参见#unique_125。

9. 上述配置完成后，单击页面右下角的**下一步**。

10. 配置同步初始化的高级配置信息。

创建同步作业 返回数据同步列表

1. 选择同步通道的源及目标实例
2. 选择同步对象
3. 高级设置
4. 预检查

同步初始化：
☒ 结构初始化
☒ 全量数据初始化

取消
上一步
保存
预检查并启动



说明：

同步初始化类型细分为：结构初始化，全量数据初始化。选中**结构初始化**和**全量数据初始化**后，DTS会在增量数据同步之前，将源数据库中待同步对象的结构和存量数据，同步到目标数据库。

11. 上述配置完成后，单击页面右下角的**预检查并启动**。



说明：

- 在数据同步作业正式启动之前，会先进行预检查。只有预检查通过后，才能成功启动数据同步作业。
- 如果预检查失败，单击具体检查项后的图标，查看失败详情。根据提示修复后，重新进行预检查。

12. 在**预检查**对话框中显示**预检查通过**后，关闭**预检查**对话框，同步作业将正式开始。

13.等待同步作业的链路初始化完成，直至处于**同步中**状态。

您可以在**数据同步**页面，查看数据同步作业的状态。

同步作业名称

搜索

排序：默认排序

状态：全部

☐	实例ID/作业名称	状态	同步概况	付费方式	同步架构(全部)	操作
☐	hangzhou-hangzhou-small	同步中	延时：565 毫秒 速度：0TPS(0.00MB/s)	按量付费	单向同步	暂停同步 转包年包月 升级更多
☐	暂停同步 释放同步					共有1页， 每页显示：20条 << < 1 > >>

6 购买集群或存储包

6.1 创建PolarDB MySQL数据库集群

本文介绍如何通过PolarDB管理控制台创建PolarDB MySQL数据库集群。

前提条件

已注册并登录阿里云账号，详细操作步骤请参见[注册和登录阿里云账号](#)。

背景信息

一个集群包含一个主节点以及最多15个只读节点，最少一个只读节点，用于提供双主（Active-Active）高可用。节点是虚拟化的数据库服务器，节点中可以创建和管理多个数据库。



说明：

- PolarDB支持专有网络VPC（Virtual Private Cloud）和经典网络。VPC是阿里云上一种隔离的网络环境，安全性比传统的经典网络更高。
- PolarDB与其他阿里云产品通过内网互通时才能发挥PolarDB的最佳性能，因此，建议将PolarDB与云服务器ECS配合使用，且与ECS创建于同一个VPC，否则PolarDB无法发挥最佳性能。如果您ECS的网络类型为经典网络，需将ECS从经典网络迁移到VPC，具体请参见[ECS实例迁移](#)。

优惠活动

首购折扣价：首次购买PolarDB享受折扣价。详情请参见[优惠活动](#)。

操作步骤

1. 登录[PolarDB控制台](#)。
2. 单击页面左上角**创建新集群**。
3. 选择**包年包月**或**按量付费**。



说明：

- **包年包月**：在创建集群时支付计算节点（一个主节点和一个只读节点）的费用，而存储空间会根据实际数据量按小时计费，并从账户中按小时扣除。如果您要长期使用该集群，**包年包月**方式更为经济，而且购买时长越长，折扣越多。

- **按量付费（按小时付费）**：无需预先支付费用，计算节点和存储空间（根据实际数据量）均按小时计费，并从账户中按小时扣除。如果您只需短期使用该集群，可以选择**按量付费**，用完即可释放，节省费用。

4. 设置如下参数。

控制台区域	参数	说明
基本配置	地域	集群所在的地理位置。购买后无法更换地域。  说明： 请确保PolarDB与需要连接的ECS创建于同一个地域，否则它们无法通过内网互通，只能通过外网互通，无法发挥最佳性能。

控制台区域	参数	说明
	创建方式	创建PolarDB集群的方式。 • 创建主集群：创建一个全新的PolarDB。 • 创建只读集群：在一个全球数据库网络（GDN）中创建只读集群。 • 从RDS迁移：先从RDS克隆数据，然后保持增量同步，主要用于迁移。在正式迁移切换前PolarDB为只读，且默认开启Binlog。详情请参见一键升级RDS MySQL至PolarDB MySQL。 - 源RDS引擎：源RDS实例的引擎类型，不可变更。 - 源RDS版本：源RDS实例的版本，不可变更。 - 源RDS实例：可选的源RDS实例，不包括只读实例。  说明： 可选的源实例列表，不包含只读实例。暂不支持已开通TDE/SSL功能以及包含非InnoDB引擎的RDS实例。从备份文件恢复到PolarDB，不影响源实例的正常运行。 • 从RDS克隆：基于所选的RDS实例，快速克隆一个数据完全一样的PolarDB集群。详情请参见一键克隆RDS MySQL至PolarDB MySQL。 - 源RDS引擎：源RDS实例的引擎类型，不可变更。 - 源RDS版本：源RDS实例的版本，不可变更。 - 源RDS实例：可选的源RDS实例，不包括只读实例。  说明： 可选的源实例列表，不包含只读实例。暂不支持已开通TDE/SSL功能以及包含非InnoDB引擎的RDS实例。从备份文件恢复到PolarDB，不影响源实例的正常运行。 • 从回收站恢复：您可以通过从回收站中恢复已删除集群的备份来创建新集群。 - 原版本：已删除集群的版本。 - 已删除集群：已删除的集群名称。 - 历史备份：选择想要恢复的备份。
	主可用区	集群的主可用区。 • 可用区是地域中的一个独立物理区域，不同可用区之间没有实质性区别。 • 您可以选择将PolarDB与ECS创建在同一可用区或不同的可用区。 • 您只需要选择主可用区，系统会自动选择备可用区。

控制台区域	参数	说明
	网络类型	固定为VPC专有网络，无需选择。如需使用经典网络请先任选一个VPC专有网络，待创建完成后再进行配置，详情请参见 私网地址和公网地址 。
	VPC网络 VPC交换机	请确保PolarDB与需要连接的ECS创建于同一个VPC，否则它们无法通过内网互通，无法发挥最佳性能。 - 如果您已创建符合您网络规划的VPC，直接选择该VPC。例如，如果您已创建ECS，且该ECS所在的VPC符合您的规划，那么选择该VPC。 - 如果您未创建符合您网络规划的VPC，您可以使用默认VPC和交换机： - 默认VPC： - 在您选择的地域中是唯一的。 - 网段掩码是16位，如172.31.0.0/16，最多可提供65536个私网IP地址。 - 不占用阿里云为您分配的VPC配额。 - 默认交换机： - 在您选择的可用区中是唯一的。 - 网段掩码是20位，如172.16.0.0/20，最多可提供4096个私网IP地址。 - 不占用VPC中可创建交换机的配额。 - 如果以上默认VPC和交换机无法满足您的要求，您可以自行创建VPC和交换机。
资源组	资源组	您可以选择 全部 或 自营 里的资源组。
实例配置	兼容性	- 兼容MySQL 8.0（完全兼容）。原生支持并行查询，特定场景下（TPC-H测试）性能提升十倍，详情请参见并行查询。 - 兼容MySQL 5.6（完全兼容）。 - 兼容PostgreSQL 11（完全兼容）。 - 兼容Oracle（高度兼容）。
	系列	默认为 普通版 。
	节点规格	按需选择。所有PolarDB节点均为独享型，性能稳定可靠。关于各规格的具体信息，请参见 #unique_67 。
	节点个数	- 无需选择。系统将自动创建一个与主节点规格相同的只读节点。 - 如果主节点故障，系统会自动将只读节点切换为新的主节点，并重新生成一个只读节点。 - 关于只读节点的更多信息，请参见#unique_138。

控制台区域	参数	说明
	存储费用	无需选择。系统会根据实际数据使用量按小时计费，详情请参见产品价格。  说明： 创建集群时无需选择存储容量，存储容量随数据量的增减而自动弹性伸缩。
	开启TDE	选择是否开启TDE加密。启用TDE加密后，PolarDB将对集群数据文件进行加密，对于业务访问透明，会有5%~10%的性能损失。  说明： - 仅PolarDB PostgreSQL和PolarDB兼容Oracle数据库支持开启TDE加密。 - TDE功能开启后不可关闭。

5. 设置**购买时长**（仅针对预付费集群）和**集群数量**后，单击右侧的**立即购买**。



说明：

最多可以一次性创建50个集群，适用于游戏批量开服等业务场景。

6. 在**确认订单**页面，确认订单信息，阅读和勾选服务协议，单击**去支付**。

7. 在**支付**页面，确认未支付订单信息和支付方式，单击**订购**。

8. 支付成功后，需要10~15分钟创建集群，之后您就可以在集群列表中看到新创建的集群。



说明：

- 当集群中的节点状态为**创建中**时，整个集群可能仍未创建完成，此时集群不可用。只有当集群状态为**运行中**时，集群才可以正常使用。
- 请确认已选中正确的地域，否则无法看到您创建的集群。
- 当您的数据量较大时，推荐您购买PolarDB**存储包**，相比按小时付费，预付费购买存储包有折扣，购买的容量越大，折扣力度就越大。

下一步

[设置集群白名单](#)

相关API

API	描述
#unique_109	创建数据库集群。

API	描述
#unique_140	查看集群列表。
#unique_141	查看指定PolarDB集群的详细属性。
#unique_142	查询PolarDB包年包月集群自动续费状态。
#unique_143	设置PolarDB包年包月集群自动续费状态。

6.2 使用存储包

为了更好地帮助您降低存储成本，PolarDB推出了预付费形式的存储包。

背景信息

PolarDB的存储空间无需手动配置，根据数据量自动伸缩，您只需为实际使用的数据量按量付费。当您的数据量较大时，推荐您使用PolarDB存储包，相比按量付费，预付费购买存储包更加优惠，购买的容量越大，优惠力度就越大。

存储空间费用

PolarDB存储空间费用请参见[#unique_67/unique_67_Connect_42_section_u9d_n9d_3jt](#)。

存储包和按量付费的价格对比

下表为按月计费存储包和按量付费的价格对比。使用容量越大的存储包，价格越优惠。

容量（GB）	中国内地		中国香港及海外	
	按量付费（元/月）	存储包（元/月）	按量付费（元/月）	存储包（元/月）
50	175	175	195	195
100	350	350	390	390
200	700	700	780	780
300	1,050	1,050	1,170	1,170
500	1,750	1,750	1,950	1,950
1,000	3,500	3,150	3,900	3,510
2,000	7,000	6,300	7,800	7,020
3,000	10,500	7,800	11,700	8,600
5,000	17,500	13,000	19,500	14,400
10,000	35,000	21,000	39,000	23,400
20,000	70,000	42,000	78,000	46,800

容量（GB）	中国内地		中国香港及海外	
	按量付费（元/月）	存储包（元/月）	按量付费（元/月）	存储包（元/月）
30,000	105,000	63,000	117,000	70,200
50,000	175,000	96,000	195,000	106,900
100,000	350,000	192,000	390,000	213,900

注意事项

- 每种类型资源包只允许购买一个，存储包容量不够时，可以[升级存储包](#)。
- 存储包暂不支持降级。
- 存储包由**资源包类型**规定的地域内的所有集群共享使用。
- 超出存储包容量的部分以按量付费的方式收取费用。例如您有三个存储容量均为400GB的PolarDB集群，这三个集群可以共享1个1000GB的存储包，超出的200GB则按量付费，您可以在[费用中心查看存储包抵扣量](#)。



说明：

更多关于存储包的说明，请参见[常见问题](#)。

购买存储包

1. 登录[PolarDB控制台](#)。
2. 在控制台左上角，选择集群所在地域。
3. 单击页面左上角。

4. 单击**存储包**页签，设置如下参数。

基本配置

存储包类型

中国内地通用

中国香港及海外通用

存储包可以被所选地域内的所有 PolarDB 集群共享，存储计费时优先抵用存储包中容量。
温馨提示：海外通用当前仅支持新加坡、马来西亚、印度尼西亚、印度孟买、美国硅谷、美国弗吉尼亚、日本东京和德国法兰克福地域。

存储包规格

50 GB	100 GB	200 GB	300 GB	500 GB
1,000 GB	2,000 GB	3,000 GB	5,000 GB	10,000 GB
20,000 GB	30,000 GB	50,000 GB	100,000 GB	

购买量

购买时长

1个月

2

3

4

5

6

7

8

9

1年

2年

3年

5年

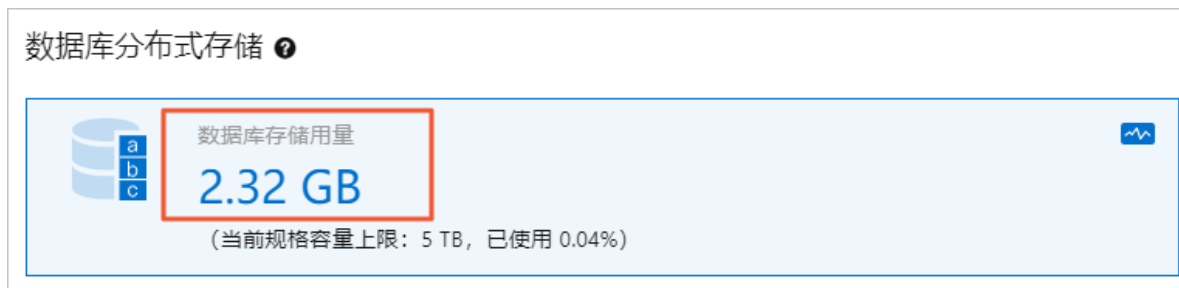
参数	说明
资源包类型	中国内地通用：购买后可用于中国内地地域的PolarDB集群。 中国香港及海外通用：购买后可用于中国香港及海外地域的PolarDB集群。
存储包规格	存储包的容量大小。
购买时长	购买存储包的时长。

5. 单击**立即购买**。

6. 勾选服务协议，单击**去支付**完成支付。

查看数据库存储用量

1. 登录[PolarDB控制台](#)。
2. 在控制台左上角，选择集群所在地域。
3. 找到目标集群，单击集群ID。
4. 在页面的**数据库分布式存储**区域，查看**数据库存储用量**。



查看存储包抵扣量



说明：

仅支持查看当前有效资源包及失效时间未超一年的资源包。

1. 登录[费用中心](#)。
2. 在左侧导航栏中，单击[资源管理](#) > [资源包](#)。
3. 在[资源包管理](#)页面的[资源包总览](#)页签，找到目标资源包，单击操作栏中的[明细](#)。

资源包管理

资源包总览

使用明细

仅支持查询当前有效资源包及失效时间未超一年的资源包

生效时间:

请选择生效时间

资源包实例ID:

请输入资源包ID进行检索

搜索

设置额度预警

导出记录

定制列

上一步

下一步

产品名称 ▾	资源包名称 ▾	资源包ID	状态 ▾	总量	剩余量	生效时间 ↕	失效时间 ↕	抵扣类型 ①	操作
云数据库 PolarDB	POLARDB大陆通用	POLARDB-实例ID	有效	50GB	50GB	2020-03-19 16:49:02	2020-04-20 00:00:00	总量恒定型	统计 明细 续费 升级
云数据库 Redis	Redis高可用版	实例ID	有效	100GB	100GB	2020-03-19 16:49:02	2020-04-20 00:00:00	总量恒定型	统计 明细 续费 升级
云数据库 MySQL	MySQL高可用版	实例ID	有效	100GB	100GB	2020-03-19 16:49:02	2020-04-20 00:00:00	总量恒定型	统计 明细 续费 升级
云数据库 PostgreSQL	PostgreSQL高可用版	实例ID	有效	100GB	100GB	2020-03-19 16:49:02	2020-04-20 00:00:00	总量恒定型	统计 明细 续费 升级

共有 4 条，每页显示:

20

< 1 >

跳转至:

跳转

- 4. 在使用明细页签，查看资源包使用情况。**

资源包管理

资源包总览

使用明细

仅支持查询当前有效资源包及失效时间未超一年的资源包

抵扣时间：

2020-03-01 ~ 2020-03-19

资源包实例ID：

请输入资源包ID进行搜索

抵扣实例ID：

请输入抵扣实例ID进行搜索

搜索

导出记录

定制列

导出

产品名称 ▾	资源包名称 ▾	资源包ID	抵扣时间 ⌚	消费时间 ⌚	使用类型 ▾	使用前剩余量	抵扣量	使用后剩余量	抵扣产品 ▾	抵扣实例ID
云数据库 PolarDB	POLARDB大陆通用	POLARDB-	2020-03-19 16:49:03	2020-03-19 16:49:03	购买	50GB	-	50GB	polardb	-

共有 1 条，每页显示：20 < 1 > 跳转至：

续费或升级存储包

1. 登录[费用中心](#)。
2. 在左侧导航栏中，单击[资源管理](#) > [资源包](#)。
3. 在[资源包管理](#)页面的[资源包总览](#)页签，找到目标资源包，单击操作栏中的[续费或升级](#)。

[illegible]

4. 您可以按以下方法进行续费或升级操作：

- **续费**

- a. 单击**操作列**的**续费**。

- b. 选择**续费时长**，勾选服务协议，单击**去支付**完成支付。

- **升级**

- a. 单击**操作列**的**升级**。

- b. 选择**存储包规格**，勾选服务协议，单击**去支付**完成支付。

常见问题

- 存储包是否跟集群绑定售卖？

答：不绑定。您需要单独购买存储包，购买后会自动抵扣相应地域内的集群存储空间。

- 存储包是否可以被多个集群共享？

答：可以。存储包由**资源包类型**（中国内地或中国香港及海外）规定的地域内的所有集群共享使用。

- 存储包是否可以被不同引擎的集群共享？

答：可以，存储包可以同时用于PolarDB MySQL/PolarDB-P/PolarDB-O集群。

- 存储包容量不够用或用完了怎么办？可以再买一个吗？

答：每种类型的存储包只能购买一个，您可以通过升级存储包的方式解决，具体请参见[续费或升级存储包](#)。

- 当前数据量超出存储包容量的部分如何计费？

答：超出存储包容量的部分以按量付费的方式收取费用，具体请参见[#unique_67/unique_67_Connect_42_section_u9d_n9d_3jt](#)。

- 存储包支持3TB和5TB，业务只需要4TB，如何选购？

答：您可以先购买3TB，存储量接近5TB时再升级为5TB。

- 存储包要到期了怎么续费？

答：您可以在用户中心对即将到期的存储包进行续费，具体请参见[续费或升级存储包](#)。

6.3 按量付费转包年包月

您可以根据需求将后付费（按量付费）的集群转变为预付费（包年包月）的计费方式。

前提条件

- 集群状态为**运行中**。

- 集群没有未完成的按量付费转包年包月的订单。如果有，需要先在[订单管理](#)页面支付或作废该订单。

背景信息

按量付费转包年包月操作对集群的运行不会有任何影响。



说明：

历史规格不支持直接转包年包月，请先[变更配置](#)，然后再转包年包月。

操作步骤

1. 登录[PolarDB控制台](#)。
2. 在控制台左上角，选择集群所在地域。
3. 找到目标集群，在操作列中选择... > 转包年包月。

云数据库POLARDB... 集群列表							
集群列表							
集群名称	集群ID	状态	实例数据库	节点数	主节点配置	已使用数据	付费类型
pc-xxxxxx	pc-xxxxxx	运行中	MySQL 5.6	3	4核 32GB	2.76 GB	按小时付费 2018年12月10日 16:56:41 创建
pc-xxxxxx	pc-xxxxxx	运行中	MySQL 5.6	2	4核 16GB	2.77 GB	按小时付费 2018年12月7日 15:11:12 创建
pc-xxxxxx	pc-xxxxxx	运行中	MySQL 5.6	3	2核 4GB	2.78 GB	包年包月 2018年12月21日 00:00:00 到期
pc-xxxxxx	pc-xxxxxx	运行中	MySQL 5.6	3	4核 32GB	2.80 GB	按小时付费 2018年11月1日 11:49:52 创建

4. 选择购买时长，勾选云数据库PolarDB服务协议，单击去开通，根据提示完成支付。



说明：

- 本操作会产生一个新购订单，只有完成了订单的支付，计费方式的变更才能生效。
- 若未支付或未成功支付，您的[订单管理](#)页面将会出现未完成订单，导致您无法新购集群或再次执行转包年包月。此时需支付或作废该订单。

6.4 包年包月转按量付费

您可以根据业务需求，将PolarDB集群的计费方式由预付费（包年包月）转变为后付费（按量付费）。

前提条件

- 集群状态为**运行中**。
- 集群的计费方式为包年包月。
- 集群不处于临时升配的状态。
- 请确保账户余额充足，避免集群转按量付费后欠费导致停机。

费用

包年包月实例转为按量付费后，会将包年包月的余款按之前的支付方式原路退回给您。

余款 = (剩余天数/总天数) × 订单现金金额 × (1 - 15%)



说明：

- 订单现金金额指实际支付的现金金额，不包括代金券或优惠券。
- 预付费转按量付费会收取实际剩余价值金额15%的手续费。

影响

本操作对实例的运行不会有任何影响。



说明：

转换前请慎重考虑，如果您长期使用实例，建议维持包年包月的计费方式，价格比按量付费更实惠，且购买时长越长，折扣越多。

操作步骤

1. 登录[PolarDB控制台](#)。
2. 在控制台左上角，选择集群所在地域。
3. 找到目标集群，单击操作栏的  > **转按量付费**。



4. 在**确认订单**页面，确认集群信息后勾选服务协议，单击**去支付**。

6.5 手动续费集群

本文介绍如何通过PolarDB控制台或续费管理控制台进行手动续费。

背景信息

您可以通过PolarDB控制台或续费管理控制台进行手动续费。通过续费管理控制台还可以为多个实例进行批量续费。

**说明：**

按小时付费集群没有到期时间，不涉及续费操作。

方法一：PolarDB控制台续费

1. 登录PolarDB控制台。
2. 在控制台左上角，选择集群所在地域。
3. 找到目标集群，在右侧选择... > 续费。

云数据库POLAR...

集群列表

集群列表

创建新集群

集群ID 请输入

集群名称	状态	兼容数据库	节点数	主节点配置	已使用数据	付费类型	操作
pc-xxxxxx	运行中	MySQL 5.6	3	2核 4GB	2.74 GB	包年包月 2019年1月20日 00:00:00 到期	升级 增节点 ...
pc-xxxxxx	创建中	MySQL 5.6	2	2核 4GB	-	按小时付费 2019年12月10日 16:58:14 创建	降配 减节点
pc-xxxxxx	变更配置中	MySQL 5.6	2	4核 16GB	2.75 GB	按小时付费 2018年12月5日 18:59:16 创建	你喜爱的新集群 续费

4. 选择续费时长并勾选服务协议，单击去支付完成支付即可。

方法二：续费管理控制台续费

1. 登录PolarDB控制台。
2. 在控制台右上方，选择费用 > 续费管理。

☰

阿里云

华东1 (杭州)

🔍 搜索

费用 工单 备案

云数据库POLAR...

集群列表

集群列表

创建新集群

集群ID 请输入

待处理事件

历史事件

集群名称	状态	兼容数据库	节点数	主节点配置	已使用数据	续费管理
	运行中	Oracle	2	2核 8GB	8.11 GB	
	运行中	PostgreSQL 11	2	2核 8GB	5.90 GB	按小时付费 2019年5月24日 09:21:44 创建

进入费用中心

续费管理

消费记录

发票

订单

充值

3. 在控制台左上角单击体验新版，切换到新版控制台。

阿里云 续费管理 体验新版-> 云服务器 云数据库RDS版	云服务器 全部时间 全部地域
---	---

4. 通过搜索过滤功能在**手动续费**页签中找到目标集群，您可以单个续费或批量续费：

- 单个续费

a. 在目标集群右侧，单击**续费**。



说明：

- 示例为新版续费管理控制台操作步骤，如果您使用旧版控制台，需要在左侧导航栏中找到**云数据库PolarDB**，然后进行续费操作。

- 如果目标集群在**自动续费或到期不续费**页签中，您可以单击**恢复手动续费**，在弹出的对话框中单击**确定**即可恢复为手动续费。

b. 选择续费时长并勾选服务协议，单击**去支付**完成支付即可。

批量续费

a. 勾选目标集群，单击下方**批量续费**。

b. 选择每个集群的**续费时长**，单击**去支付**完成支付即可。

自动续费

开通自动续费可以免去您定期手动续费的烦恼，且不会因为忘记续费而导致业务中断。详情请参见[#unique_145](#)。

6.6 自动续费集群

本文为您介绍如何为集群开通自动续费，免去您手动续费的烦恼。

背景信息

包年包月集群有到期时间，如果到期未续费，会导致业务中断甚至数据丢失。开通自动续费可以免去您定期手动续费的烦恼，且不会因为忘记续费而导致业务中断。



说明：

按小时付费集群没有到期时间，不涉及续费操作。

注意事项

- 自动续费将于集群到期前9天开始扣款，支持现金及代金券扣款，请保持账户余额充足。
- 若您在自动扣款日期前进行了手动续费，则系统将在下一次到期前进行自动续费。
- 自动续费功能于次日生效。若您的集群将于次日到期，为避免业务中断，请手动进行续费，详细步骤请参见[手动续费集群](#)。

购买集群时开通续费



说明：

开通自动续费后，系统将根据您的**购买时长**为周期进行自动续费。例如，如果您购买了3个月的集群并勾选了自动续费，则每次自动续费时会缴纳3个月的费用。

在创建新集群时，可以勾选**自动续费**。

购买集群后开通自动续费



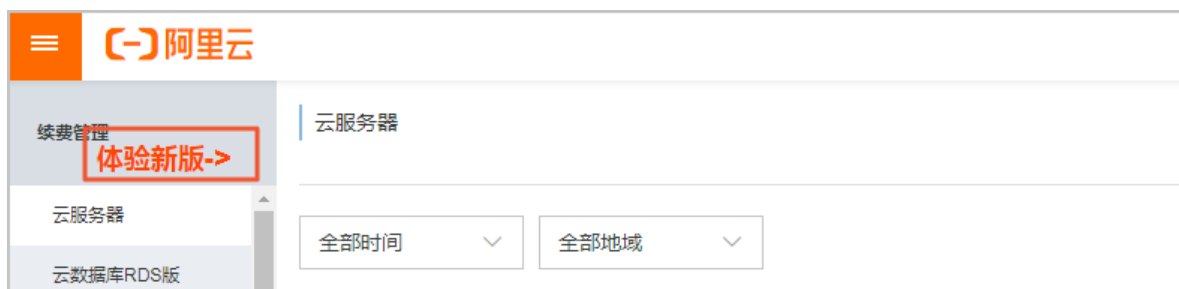
说明：

开通自动续费后，系统将根据您选择的续费周期进行自动续费。例如，如果您选择了3个月的续费周期，则每次自动续费时会缴纳3个月的费用。

1. 登录[POLARDB管理控制台](#)。
2. 在控制台右上方，选择**费用 > 续费管理**。



3. 在控制台左上角单击**体验新版**，切换到新版控制台。



4. 通过搜索过滤功能在**手动续费**或**到期不续费**页签中找到目标集群，您可以单个开通或批量开通：

- 单个开通

a. 单击右侧**开通自动续费**。



说明：

示例为新版续费管理控制台操作步骤，如果您使用旧版控制台，需要在左侧导航栏中找到**云数据库POLARDB**，然后开通自动续费。

- b. 在弹出的对话框中，选择**自动续费周期**，单击**开通自动续费**。



- 批量开通

- a. 勾选目标集群，单击下方**开通自动续费**。



- b. 在弹出的对话框中，选择**自动续费周期**，单击**开通自动续费**。



修改自动续费周期

1. 登录POLARDB管理控制台。
2. 在控制台右上方，选择**费用 > 续费管理**。



3. 在控制台左上角单击**体验新版**，切换到新版控制台。

4. 通过搜索过滤功能在自动续费页签中找到目标集群，单击右侧修改自动续费。



说明：

示例为新版续费管理控制台操作步骤，如果您使用旧版控制台，需要在左侧导航栏中找到云数据库POLARDB，然后修改自动续费。

5. 在弹出的对话框中，修改自动续费周期后，单击确定。

关闭自动续费

1. 登录POLARDB管理控制台。
2. 在控制台右上方，选择费用 > 续费管理。



3. 在控制台左上角单击体验新版，切换到新版控制台。

4. 通过搜索过滤功能在自动续费页签中找到目标集群，单击右侧恢复手动续费。

实例名称 请输入实例名称进行搜索 搜索 云服务商ECS的统一到期日设置

时间: 全部 7天内到期 15天内到期 30天内到期 2个月内到期 3个月内到期

产品: 全部 云数据库RDS版 9 云解析 DNS 3 云数据库MongoDB版 3 数据传输服务DTS 3 云数据库MongoDB集群版 2 云数据库POLARDB 2

其他产品

地域: 全部地域

手动续费 自动续费 2 到期不续费 1

产品	实例ID/实例名称	地域	倒计时	付费方式	开始/结束时间	续费周期	操作
云数据库POLARDB	pc-xxxxxx	华东1 (杭州)	28天	包年包月	2019-04-28 13:56:00 2019-06-29 00:00:00	1个月	续费 修改自动续费 不续费 恢复手动续费
云数据库POLARDB	pc-xxxxxx	华东1 (杭州)	30天	包年包月	2019-05-31 15:29:00 2019-07-01 00:00:00	1个月	续费 修改自动续费 不续费 恢复手动续费

选中 0 项 批显续费 修改自动续费 恢复为手动续费 设置为不续费 导出续费清单 共有 2 条, 每页显示: 20 1 跳转至: 跳转



说明:

示例为新版续费管理控制台操作步骤，如果您使用旧版控制台，需要在左侧导航栏中找到云数据库POLARDB，然后关闭自动续费。

5. 在弹出的对话框中，单击确定。

相关API

API	描述
#unique_104	创建数据库集群。 说明: 创建集群时开通自动续费。
#unique_146	设置包年包月集群自动续费状态。 说明: 创建集群后开通自动续费。
#unique_147	查询包年包月集群自动续费状态。

6.7 释放集群

本文介绍如何手动释放整个集群。

背景信息

您可以根据业务需求手动释放后付费（按小时付费）的集群。

注意事项

- 预付费（包年包月）集群不支持手动释放，集群到期后会自动被释放。
- 只有在运行状态下的集群才能被手动释放。
- 集群被释放后，数据将无法找回，请谨慎操作。
- 本功能用于释放整个集群，包括集群中的所有节点。如要释放单个只读节点，请参见[增加或删除节点](#)。
- 按小时付费的集群可以直接转为包年包月，具体请参见[按量付费转包年包月](#)。

操作步骤

1. 登录[PolarDB控制台](#)。
2. 在控制台左上角，选择集群所在地域。
3. 找到目标集群，单击操作列的... > 释放。

集群ID/名称	状态	兼容性	节点数	主节点配置	已使用数据	付费类型	标签	操作
pr-xxxxxx-z	运行中	100% 兼容 MySQL 5.6	2	4核 16GB	2.31 GB	按小时付费 2020年4月28日 17:01:58 创建		升降配 增删节点
pr-xxxxxx-5	运行中	100% 兼容 MySQL 5.6	2	4核 16GB	2.31 GB	按小时付费 2020年4月28日 17:01:57 创建		升降配 增删节点
pr-xxxxxx	运行中	100% 兼容 MySQL 5.6	2	4核 16GB	8.90 GB	按小时付费 2020年4月28日 17:01:57 创建		升降配 增删节点
pr-xxxxxx	运行中	100% 兼容 PostgreSQL 11	2	2核 8GB	8.82 GB	按小时付费 2020年4月27日 15:58:32 创建		升降配 增删节点

克隆数据到新集群
恢复数据到新集群
转包年包月
释放

4. 在弹出的提示框中，单击**确认**。

相关API

API	描述
#unique_140	查看集群列表。
#unique_148	删除数据库集群。

7 集群访问

7.1 查看连接地址

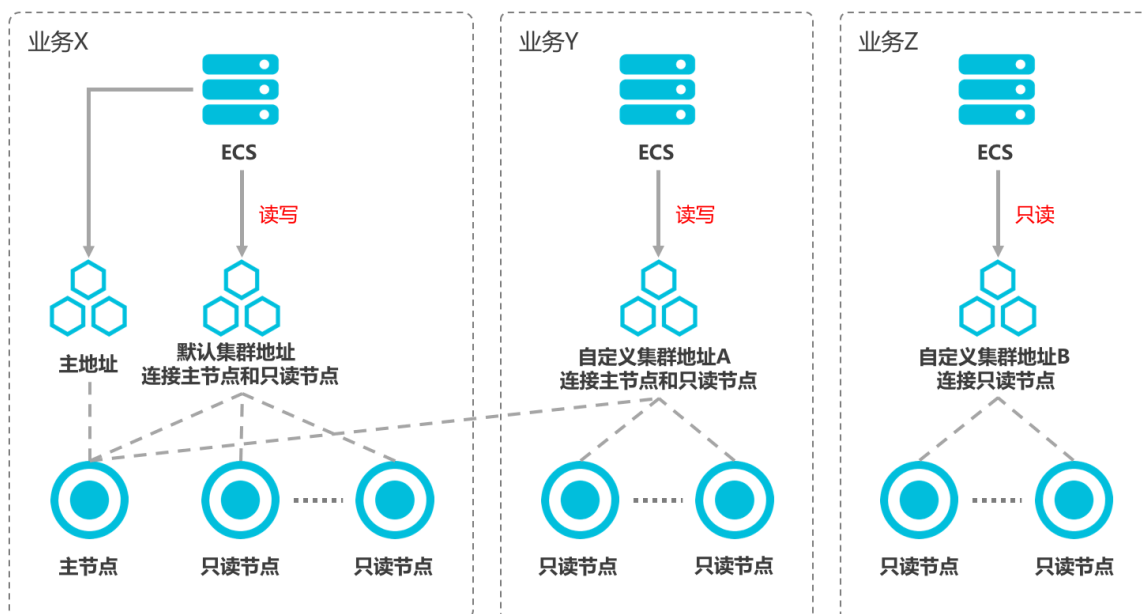
在连接PolarDB集群时，您需要填写PolarDB集群的连接地址。PolarDB提供了集群地址和主地址，本文将介绍如何在控制台查看PolarDB集群地址和主地址。

查看方法

1. 登录[PolarDB控制台](#)。
2. 在控制台左上角，选择集群所在地域。
3. 找到目标集群，单击集群ID。
4. 在**基本信息**页面的**访问信息**区域，查看连接地址。

集群地址和主地址

地址类型	地址说明	支持的网 络类型	查看方式
集群地址（推荐）	应用程序只需连接一个集群地址，即可连接到多个节点。带有读写分离功能，写请求会自动发往主节点，读请求会自动根据各节点的负载发往主节点或只读节点。  说明： PolarDB包含一个默认的集群地址，您还可以根据业务需求创建多个自定义的集群地址，自定义集群地址可以连接到指定的节点，以及设置读写模式等。详情请参见修改和释放集群地址。	私网和公网	1. 找到目标集群，单击集群的ID。 2. 在访问信息区域查看。
主地址	总是连接到主节点，支持读和写操作。当主节点发生故障时，主访问地址会自动切换到新的主节点。	私网和公网	



私网地址和公网地址

地址类型	说明	使用场景
私网地址	- 通过私网地址访问可以发挥PolarDB的最佳性能。 - 私网地址无法被释放。	例如： - ECS与数据库集群位于同一VPC，那么ECS可以通过私网地址访问数据库集群。 - 使用DMS通过VPC访问数据库集群
公网地址	- 需手动申请公网的连接地址，也可以释放公网的连接地址。 - 公网即因特网。通过公网访问将无法实现PolarDB最佳性能。	例如：通过公网访问数据库集群进行维护操作。

下一步

[连接数据库集群](#)

常见问题

Q：如果应用程序使用PolarDB公网地址连接数据库，产生的流量需要收费吗？

A：使用PolarDB公网地址连接数据库产生的流量阿里云不会收费。

相关API

API	描述
#unique_150	查询集群的地址信息。
#unique_151	创建集群的公网地址。
#unique_152	修改集群默认访问地址。
#unique_153	释放集群地址。

7.2 连接数据库集群

本文将分别介绍如何使用DMS（Data Management Service）和通用MySQL客户端连接PolarDB MySQL集群。

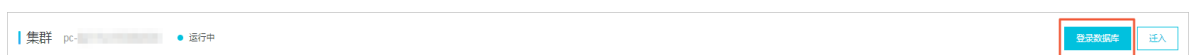
前提条件

已创建数据库集群的高权限账号或普通账号。具体操作请参见[创建数据库账号](#)。

使用DMS连接集群

DMS是阿里云提供的图形化的数据管理工具，它是一种集数据管理、结构管理、访问安全、BI图表、数据趋势、数据轨迹、性能与优化和服务器管理于一体的数据管理服务。支持对关系型数据库（MySQL、SQL Server、PostgreSQL等）和NoSQL数据库（MongoDB、Redis等）的管理，同时还支持Linux服务器管理。

1. 找到目标集群，单击集群ID，进入基本信息页面。
2. 单击页面右上角的[登录数据库](#)。



3. 在数据库登录页面，输入主地址和端口号（以英文冒号隔开），以及高权限账号或普通账号的用户名和密码，然后单击**登录**。



The image shows the RDS database login interface. At the top left is a green icon of a database cylinder. To its right is the title 'RDS数据库登录' in large black font, followed by a blue link '独立单元' with a dropdown arrow. Below the title are three input fields: the first contains 'pc-...mysql.polardb.rds.aliyuncs.com:3306', the second is empty, and the third contains masked characters. Below these fields is a checkbox labeled '记住密码'. At the bottom is a large blue button labeled '登录'. Below the button, there is a link '关于 DMS (Data Management Service)' and a copyright notice 'Copyright © DMS All Rights Reserved (Alibaba 数据管理产品)'.

**说明：**

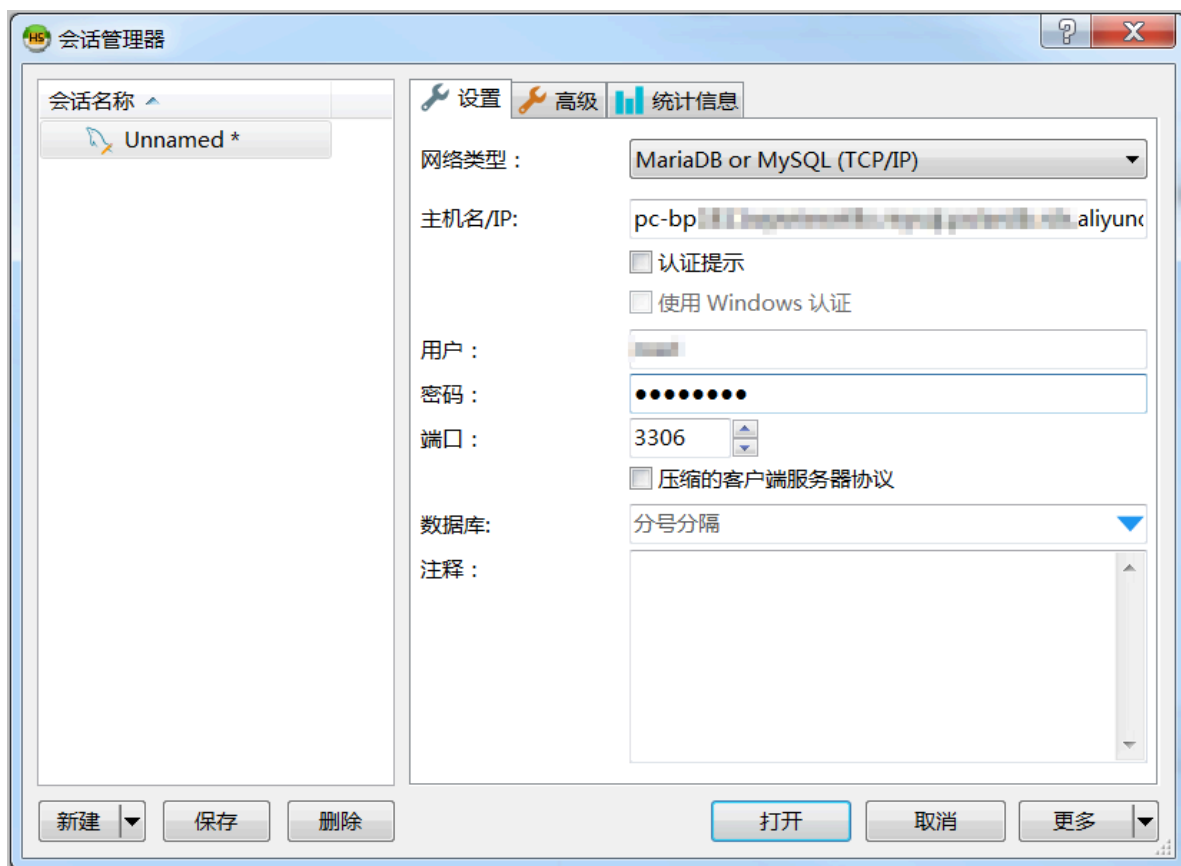
DMS登录仅支持主地址，不支持集群地址。关于如何查看连接地址，请参见[查看连接地址](#)。

使用客户端连接集群

您可以使用任何通用的客户端连接PolarDB集群。本文以[HeidiSQL](#)为例。

1. 启动HeidiSQL客户端。

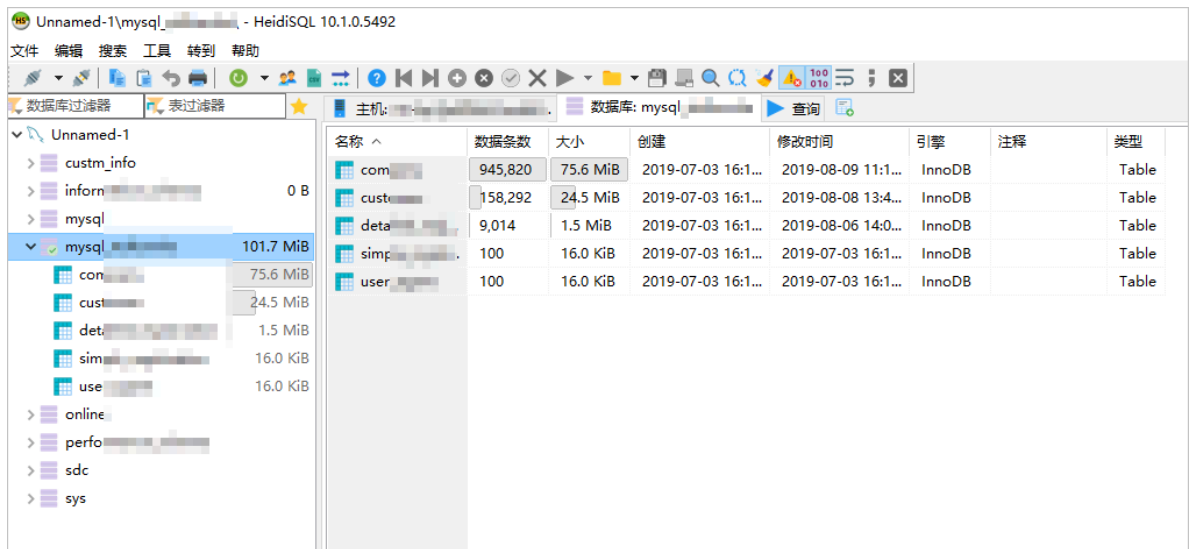
2. 在左下角单击**新建**。



3. 输入要连接的PolarDB集群信息，参数说明如下：

参数	说明
网络类型	连接数据库的形式。选择MariaDB or MySQL（TCP/IP）。
主机名/IP地址	输入PolarDB集群的私网地址或公网地址。 - 若您的客户端部署在ECS实例上，且ECS实例与要访问的PolarDB集群的地域、网络类型相同，请使用私网地址。例如ECS实例和PolarDB集群都是华东1的专有网络实例，使用私网地址连接能提供安全高效的访问。 - 其它情况只能使用公网地址。 查看PolarDB集群的私网、公网地址及端口信息的步骤如下： - 登录PolarDB控制台。 - 在页面左上角，选择集群所在地域。 - 找到目标集群，单击集群ID。 - 在基本信息页中，即可查看私网、公网地址及其端口信息。
用户	要访问PolarDB集群的账号名称。
密码	以上账号的密码。
端口	输入PolarDB集群的私网地址或公网地址对应的端口号。

4. 单击打开。若连接信息无误，即会成功连接集群。



使用命令行连接集群

如果您的服务器安装了MySQL，可以通过命令行连接PolarDB MySQL数据库集群，连接方式如下：

```
mysql -h<连接地址> -P<端口> -u<用户名> -p<密码> -D<数据库>
```

参数	说明	示例
-h	PolarDB集群的私网地址或公网地址。连接地址请参见 查看连接地址 。	pc-bpxxxxxxxxxxxxxxxx.mysql. polardb.rds.aliyuncs.com
-P	PolarDB集群的端口号。 - 若使用私网连接，需输入PolarDB集群的私网端口。 - 若使用公网连接，需输入PolarDB集群的公网端口。  说明： - 默认端口为3306。 - 如果端口号为默认端口，该参数可以不填。	3306
-u	要访问PolarDB集群的账号名称。	root

参数	说明	示例
-p	以上账号的密码。  说明： 该参数非必填参数。 - 如果不填写该参数，后续操作中会重新要求输入密码。 - 如果填写该参数，-p与数据库密码之间不能有空格。	password233
-D	需要登录的数据库名称。  说明： 该参数非必填参数。 可以不输入-D仅输入数据库名称。	mysql

解决连接失败问题

- IP白名单设置有误。
 - 由于白名单设置中只有默认地址127.0.0.1。127.0.0.1表示禁止任何IP地址访问PolarDB集群。因此需在白名单中添加对端的IP地址，具体操作请参见[设置集群白名单](#)。
 - 白名单设置成了0.0.0.0，正确格式为0.0.0.0/0。



说明：

0.0.0.0/0表示允许任何设备访问PolarDB集群，请谨慎使用。

- 您在白名单中添加的设备公网IP地址可能并非设备真正的出口IP地址。例如，公网IP地址不固定，可能会变动。或者IP地址查询工具、网站查询的公网IP地址不准确。
- 私网、公网地址使用错误。

使用私网地址从公网进行连接，或者使用公网地址从私网进行连接，都会导致连接失败。

请确认您使用的地址类型。如果您需要从私网连接PolarDB集群，请使用PolarDB集群的私网地址；如果您需要从公网连接PolarDB集群，请使用PolarDB集群的公网地址。
- 网络类型不同，ECS实例采用经典网络，PolarDB集群采用专有网络（VPC）。
 - 解决办法一（推荐）：将ECS实例从经典网络迁移到VPC，具体操作请参见[单ECS迁移示例](#)。



说明：

切换后，两者必须处于同一个VPC，才能内网互通。

- 解决办法二：使用ClassicLink功能，使经典网络的ECS实例可以和VPC中的云资源通过内网互通。
- 解决办法三：ECS实例使用PolarDB集群的公网连接地址连接PolarDB集群，即通过公网连接PolarDB集群。这种方式的性能、安全性、稳定性较差。

7.3 集群地址

7.3.1 新增自定义集群地址

您可以在PolarDB MySQL集群上新增自定义集群地址，通过设置集群地址的读写模式、一致性级别及关联的只读节点等，来满足不同的业务场景，增强业务的灵活性。本文将介绍如何为PolarDB MySQL新增自定义集群地址。

前提条件

对于2019年4月29日起创建的集群，您可以直接新增自定义集群地址。若您需要为2019年4月29日之前创建的集群新增自定义地址，请通过[提交工单](#)联系售后服务。

操作步骤

1. 登录[PolarDB控制台](#)。
2. 在控制台左上角，选择集群所在地域。
3. 单击目标集群ID，进入集群**基本信息**详情页。
4. 在**链接地址**区域，单击**创建自定义地址**。



5. 在**创建自定义地址**对话框内，设置如下参数。

参数	说明
读写模式	本地址的读写模式，可选模式为只读和可读可写（自动读写分离）。  说明： 创建自定义地址后还可以修改读写模式。修改读写模式后，只对新建连接生效，已有的连接保持原来的模式。
读负载节点	在左侧选择想要加入本地址用于处理读请求的节点，可选节点包括主节点和所有只读节点。  说明： - 读写模式为可读可写（自动读写分离）时，至少要选择2个节点。不论是否选中主节点，写请求都只会发往主节点。 - 读写模式为只读时，支持创建单节点地址，详情请参见常见问题。
新节点自动加入	新增的节点是否要自动添加到该地址中。
负载均衡策略	读写分离时，在多个节点间用于处理读请求的调度策略，默认为 基于负载的自动调度 ，且不可更改。
一致性级别	- 读写模式为可读可写（自动读写分离）时，可选一致性级别有最终一致性、会话一致性（推荐）和全局一致性，详情请参见PolarDB一致性级别。 - 读写模式为只读时，默认一致性级别为最终一致性且不可更改。
主库不接受读	开启之后，查询SQL将仅发送到只读节点，来降低主节点的负载，确保主节点稳定。  说明： 仅读写模式为可读可写（自动读写分离）时支持该配置。
事务拆分	开启或关闭事务拆分，详情请参见高级选项-事务拆分。  说明： 仅读写模式为可读可写（自动读写分离）时支持该配置。

参数	说明
连接池	• 关闭（默认选项） • 开启会话级连接池 会话级连接池适用于PHP短连接场景。 PHP短连接优化主要用于减少短连接业务频繁建立新连接带来的负载。当某客户端连接断开时，系统会判断当前连接是否为闲置连接，如果是闲置连接，系统将会代理该连接并在连接池中保留一小段时间。当客户端重新发起连接请求时，若该闲置连接仍被保留在连接池中，则可直接使用已保留的闲置连接（命中的条件包括user、clientip和 dbname等），从而减少与数据库的建连开销。如果连接池内没有可用的闲置连接，则走正常连接流程，与数据库重新建立新连接。  说明： 该优化并不能减少数据库的并发连接数，而是通过降低应用与数据库建连的速率来减少MySQL主线程的开销，更好的处理业务请求。但连接池里的闲置连接会短暂占用您的连接数。 • 开启事务级连接池 事务级连接池适用于总连接数比较多（如连接数上万）的场景。 事务级连接池主要用于减少业务的连接数和短连接场景下频繁新建连接带来的负载。客户端可以与代理建立大量连接，但代理到数据库只创建少量连接。当客户端发送连接请求时，代理将从连接池中选取符合条件（即系统变量一致）的连接发送到数据库，并在当前事务结束后将该连接放回连接池。 使用限制： - 当您执行以下行为时，连接将被锁定直至连接结束（即该连接不会再被放到连接池里供其它用户连接使用）。 ■ 执行prepare语句或命令。 ■ 创建临时表。 ■ 修改用户变量。 ■ 大报文（例如16M以上）。 ■ 使用lock table。 ■ 多语句。 ■ 存储过程调用。 - 不支持FOUND_ROWS、ROW_COUNT和LAST_INSERT_ID函数的调用，即使这些函数调用成功，但无法保证结果是否正确。 - 对于设置了wait_timeout的连接，wait_timeout在客户端的表现可能不会生效，因为每次请求都会从连接池中获取连接，当wait_timeout超时后，只有连接池中的后端连接会断开，而后端连接断开并不会导致客户端连接断开。

参数	说明
并行查询	开启或关闭并行查询，详情请参见 并行查询 。  说明： 仅当兼容性为MySQL 8.0的PolarDB MySQL集群在读写模式为只读时支持该配置。

6. 单击**确定**。

常见问题

- Q：如何创建单个节点的独立地址？

A：仅当集群地址读写模式为**只读**且集群内拥有三个及以上节点时，才支持创建单节点地址。单节点地址的读负载节点只能是只读节点，详细操作步骤请参见[操作步骤](#)。



警告：

创建单节点地址后，当此节点故障时，该地址可能会有最多1小时的不可用，请勿用于生产环境。

- Q：一个集群最多允许创建多少个单节点地址？

A：如果您的集群内有3个节点，则只允许为其中1个只读节点创建单节点地址；若集群内有4个节点，则允许为其中2个只读节点创建各自的单节点地址，以此类推。

- Q：当出现系统故障需要进行主备切换时，单节点地址下对应的只读节点能否切换为新主节点？

A：主备切换时，单节点地址下对应的只读节点不会被自动切换为新主节点。但您仍可以通过手动切换将其设置为新主节点，详细操作请参见[主备切换](#)。

- Q：一个集群最多可拥有多少个集群地址？

A：一个集群最多可拥有4个集群地址，其中1个为默认地址，另外3个为自定义地址。

- Q：可以修改集群地址吗？

A：默认集群地址和自定义地址都支持修改配置，详情请参见[修改集群地址](#)。

- Q：可以释放集群地址吗？

A：仅自定义集群地址可以被释放，默认集群地址无法被释放，详情请参见[释放自定义集群地址](#)。

相关API

API	描述
#unique_155	创建自定义集群地址。
#unique_150	查询集群地址。

7.3.2 修改和释放集群地址

PolarDB集群支持修改和释放集群地址，本文以PolarDB MySQL集群为例介绍如何修改和释放集群地址。

修改集群地址

1. 登录 [PolarDB控制台](#)。
2. 在控制台左上角，选择集群所在地域。
3. 单击目标集群ID，进入集群**基本信息**页面。
4. 在**链接地址**区域，找到目标集群（本文以默认集群地址为例），单击目标集群名称右侧的 图标后，再单击**编辑配置**。
5. 在弹出的对话框内，设置如下参数。

参数	说明
读写模式	本地地址的读写模式，分为 只读 和 可读可写（自动读写分离） 。  说明： 创建自定义地址后还可以修改读写模式。修改读写模式后，只对新建连接生效，已有的连接保持原来的模式。
读负载节点	在左侧选择想要加入本地地址用于处理读请求的节点。可选节点包括主节点和所有只读节点。  说明： • 读写模式为只读时，支持创建单节点地址。若创建单节点地址，当此节点故障时，该地址可能会有最多1小时的不可用，请勿用于生产环境。因此推荐至少选择2个节点，提升可用性。 • 读写模式为可读可写（自动读写分离）时，至少要选择2个节点。不论是否选中主节点，写请求都只会发往主节点。
新节点自动加入	新增的节点是否要自动添加到该地址中。
负载均衡策略	读写分离时，在多个节点间用于处理读请求的调度策略，默认为 基于负载的自动调度 ，无需选择。
主库不接受读	在确保一致性的前提下，将查询SQL发送到只读节点，来降低主节点的负载，确保主节点稳定。  说明： 仅可读可写（自动读写分离）模式下支持该配置。

参数	说明
一致性级别	读写模式为可读可写（自动读写分离）时，一致性级别可选择为最终一致性、会话一致性（推荐）或全局一致性，详情请参见PolarDB一致性级别。  说明： - 读写模式为只读时，默认选择最终一致性且无法更改。 - 一致性级别修改后对所有连接立即生效。
事务拆分	开启或关闭事务拆分，详情请参见高级选项-事务拆分。  说明： 仅可读可写（自动读写分离）模式下支持该配置。

参数	说明
连接池	- 关闭（默认选项） - 开启会话级连接池 会话级连接池适用于PHP短连接场景。 PHP短连接优化主要用于减少短连接业务频繁建立新连接带来的负载。当某客户端连接断开时，系统会判断当前连接是否为闲置连接，如果是闲置连接，系统将会代理该连接并在连接池中保留一小段时间。当客户端重新发起连接请求时，若该闲置连接仍被保留在连接池中，则可直接使用已保留的闲置连接（命中的条件包括user、clientip和 dbname等），从而减少与数据库的建连开销。如果连接池内没有可用的闲置连接，则走正常连接流程，与数据库重新建立新连接。  说明： 该优化并不能减少数据库的并发连接数，而是通过降低应用与数据库建连的速率来减少MySQL主线程的开销，更好的处理业务请求。但连接池里的闲置连接会短暂占用您的连接数。 - 开启事务级连接池 事务级连接池适用于总连接数比较多（如连接数上万）的场景。 事务级连接池主要用于减少业务的连接数和短连接场景下频繁新建连接带来的负载。客户端可以与代理建立大量连接，但代理到数据库只创建少量连接。当客户端发送连接请求时，代理将从连接池中选取符合条件（即系统变量一致）的连接发送到数据库，并在当前事务结束后将该连接放回连接池。 使用限制： - 当您执行以下行为时，连接将被锁定直至连接结束（即该连接不会再被放到连接池里供其它用户连接使用）。 ■ 执行prepare语句或命令。 ■ 创建临时表。 ■ 修改用户变量。 ■ 大报文（例如16M以上）。 ■ 使用lock table。 ■ 多语句。 ■ 存储过程调用。 - 不支持FOUND_ROWS、ROW_COUNT和LAST_INSERT_ID函数的调用，即使这些函数调用成功，但无法保证结果是否正确。 - 对于设置了wait_timeout的连接，wait_timeout在客户端的表现可能不会生效，因为每次请求都会从连接池中获取连接，当wait_timeout超时后，只有连接池中的后端连接会断开，而后端连接断开并不会导致客户端连接断开。

6. 单击**确定**。

释放自定义集群地址



说明：

仅自定义集群地址可以被释放，默认集群地址无法被释放。

1. 登录[PolarDB控制台](#)。
2. 在控制台左上角，选择集群所在地域。
3. 单击目标集群ID，进入集群**基本信息**页面。
4. 在**链接地址**区域，找到目标集群，单击目标集群名称右侧的图标后，再单击**释放**。
5. 在弹出的对话框中，单击**确定**。

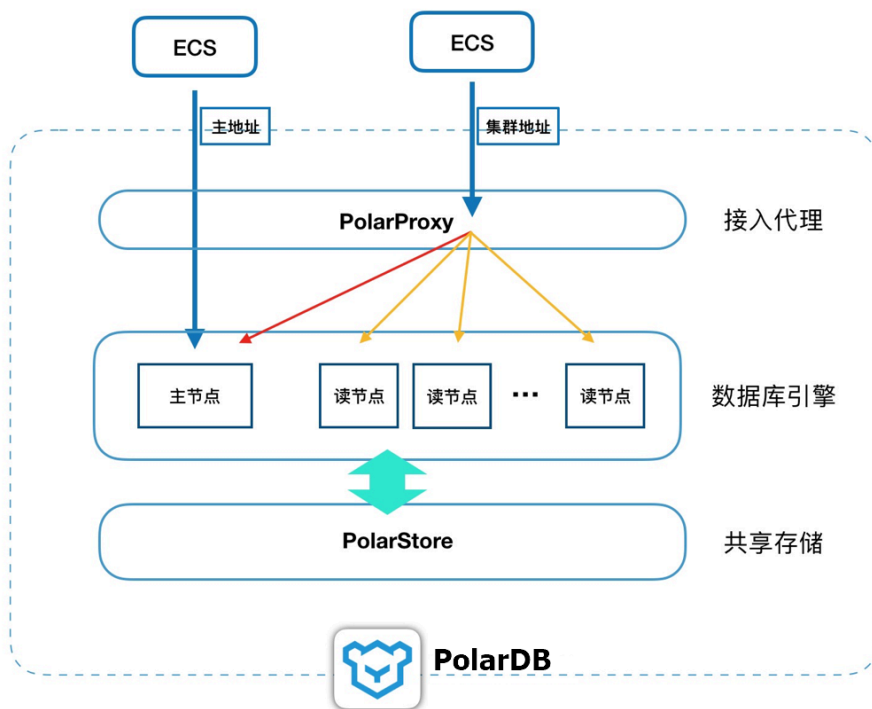
相关API

API	描述
#unique_150	查询集群地址。
#unique_156	修改集群地址。
#unique_157	释放自定义集群地址。

7.3.3 PolarProxy功能介绍

本文为您介绍PolarProxy（代理）的一致性级别、事务拆分、主库不接受读、连接池、Hint以及并行查询等功能。

PolarDB架构及PolarProxy介绍



PolarDB是一个由多个节点构成的数据库集群，一个主节点，多个读节点。对外默认提供两个地址，分别为主地址和集群地址。其中，集群地址功能由PolarProxy提供，集群地址分为读写集群地址和只读集群地址，读写集群地址支持读写分离，只读集群地址支持按连接数负载。读写分离请参见[读写分离](#)，更多关于PolarProxy的功能介绍请参见下文。

一致性级别

由于MySQL使用异步复制的方式进行主从同步，该模式在读写分离模式下，可能出现一个写请求之后的读请求无法获得最新的写结果，导致不一致。PolarDB提供了三种一致性级别供您选择取值范围如下：

- 最终一致性

PolarDB采用异步物理复制方式在主库和只读库间做数据同步，在主库更新后，相关的更新会同步到只读库，具体的延迟时间与写入压力有关，延迟一般在毫秒（ms）级别，通过异步复制的方式实现主库和只读库之间的最终数据一致。该级别无法保证业务通过读写分离地址获得一致性的数据。

- 会话一致性（默认）

为解决最终一致性会出现的查询不一致，PolarDB利用自身物理复制速度快的优点，内部数据库代理会保证将查询发给已经更新了数据的只读节点，以满足会话一致性需求。

**说明：**

会话一致性只能满足一个会话内的请求是一致的，不能保证不同会话间的一致性，当所有只读节点都无法满足一致性要求时，会把该读请求路由到主库，所以可能引起主库读压力大。

- 全局一致性

PolarDB的读写分离模块提供了会话一致的功能，保证同一个会话内的请求能保证因果一致，但是部分应用场景，除了会话内部有逻辑上的因果依赖关系，会话之间也有依赖关系，例如短连接场景或者连接池，这时需要全局一致的功能。全局一致行是通过跟踪复制位点来实现的，在主从延迟较高时，可能会导致更多的请求路由到主库，造成主库压力大，同时业务延迟也可能增加。

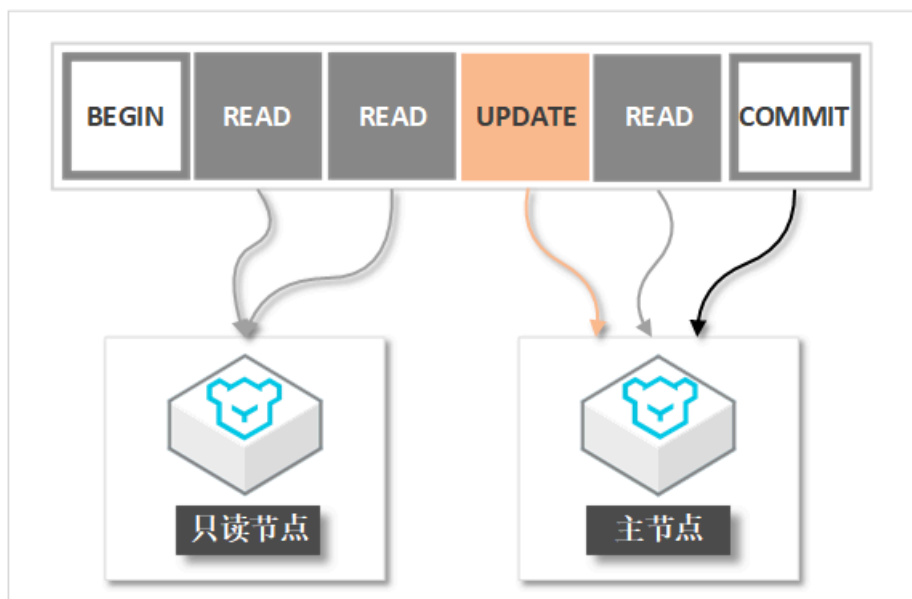
**说明：**

更多关于一致性级别的内容请参见[PolarDB一致性级别](#)。

事务拆分

在默认的Read Committed隔离级别下，当数据库接收到开启事务的语句（例如begin或set autocommit=0）时，不会立即开启事务，而是在发生写操作时才正式开启事务。

通常情况数据库代理会将事务内的所有请求都发送到主节点，但由于某些框架实现的原因会将所有请求封装到事务中，导致主节点负载过大。此时您可以开启事务拆分功能，开启后数据库代理会识别当前事务的真正状态，将正式开启事务前的读请求通过负载均衡模块分流至只读节点。



**说明：**

- 读写分离模式下，事务拆分功能默认开启。
- 某些业务对一致有较高要求，开启事务拆分后将不满足一致，因此在事务拆分前请充分评估事务拆分功能是否适用于您的业务。
- 更改事务拆分设置后，仅对新建连接生效。

主库不接受读

当主库配置为不接受读时，普通的读请求将不再路由到主库。而事务内，一致性要求的读还是会被路由到主库，以保证业务的需求，另外当所有只读节点出现故障后读请求也会路由到主库。如果业务对一致的要求较低，可以通过设置**一致性级别为最终一致性**来减少读请求到主库，也可以通过**事务拆分**功能来减少真正事务前的读请求路由到主库。广播的请求（例如set, prepare）还是会路由到主库上。

**说明：**

- 读写分离模式下，主库不接受读默认开启。
- 更改主库不接受读后立即生效。

连接池

PolarDB提供了会话级连接池和事务级连接池。

- 会话级连接池

会话级连接池适用于PHP短连接场景。

PHP短连接优化用于减少短连接业务频繁建立新连接导致MySQL负载高。当您的连接断开时，系统会判断当前的连接是否是一个闲置的连接，如果是闲置连接，系统将会代理该连接并保留在连接池中一小段时间，如果这时新的连接建立的话就会直接从连接池里获得连接（命中的条件包括user, clientip, dbname等），从而减少与数据库的建连开销；如果没有可用的连接，则走正常连接流程，重新与数据库建立一个新的连接。

**说明：**

该优化并不能减少数据库的并发连接数，只是减少应用与数据库建连的速率，从而减少MySQL主线程的开销，能够更好的处理业务请求。但是连接池里空闲的连接会短暂占您的连接数。

- 事务级连接池

事务级连接池适用于总连接数比较多的场景，例如连接数上万的情况。

事务级连接池主要用于减少业务的连接数，以及减少短连接场景下频繁新建连接带来的负载。客户端可以与代理建立大量连接，而同时代理到数据库只创建少量连接。当客户端发送连接请求时，代理将从连接池中选取符合条件的（现阶段为系统变量一致）连接进行请求发送到数据库，并在当前事务结束后将连接放入连接池。

使用限制如下：

- 当执行以下行为时，锁定连接，直至连接结束，即该连接不会再被放到连接池里供其它用户连接使用。
 - 执行prepare语句或命令
 - 创建临时表
 - 修改用户变量
 - 大报文（例如16M以上）
 - 使用lock table
 - 多语句
 - 存储过程调用
- 不支持FOUND_ROWS、ROW_COUNT和LAST_INSERT_ID函数的调用，这些函数可以调用成功，但是结果无法保证正确。
- 对于设置了wait_timeout的连接，wait_timeout在客户端的表现可能不会生效，因为每次请求都会从连接池中获取连接，当wait_timeout超时时，只有连接池中的后端连接会断开，而后端连接断开并不会导致客户端连接断开。
- 除了sql_mode、character_set_server、collation_server、time_zone这四个变量以外，如果业务依赖其他session级别的系统变量，那么需要客户端在建连之后显式进行set语句执行，否则连接池可能会复用系统变量已经被更改过的连接。
- 由于连接可能会被复用，所以使用select connection_id()查询当前连接的thread id可能会变化。
- 由于连接可能会被复用，所以show processlist中，显示的IP地址和端口可能和客户端实际的IP地址和端口不一致。
- 数据库代理会将所有节点上的show processlist结果合并并返回，而在事务级连接池开启后，前端连接和后端连接的thread id无法对应。这导致kill命令会可能返回一个错误，但是实际上kill命令已经正常执行成功，可再通过show processlist确定相应的连接是否断开。



说明：

- 更改连接池设置后，仅对新建连接生效。
- 当前连接池功能不支持同一个账号对不同ip有不同的权限。如果您为同一个账号的不同ip设置了不同的库或者表权限，开通连接池可能会导致权限错误问题。例如user@192.xx.xx.1设置了database_a的权限，而user@192.xx.xx.2没有database_a的权限，可能会导致连接复用时长权限出错。

Hint语法

在SQL语句前加上`/*FORCE_MASTER*/`或`/*FORCE_SLAVE*/`可以强制指定这条SQL的路由方向。

例如`select * from test`默认会路由到只读节点，改为`/*FORCE_MASTER*/ select * from test`就会路由到主节点。



说明：

- Hint的路由优化级最高，不受一致性级别和事务拆分的约束，使用前请进行评估。
- Hint语句里不要有改变环境变量的语句，例如`/*FORCE_SLAVE*/ set names utf8;`等，这类语句可能导致后续的业务出错。

7.3.4 读写分离

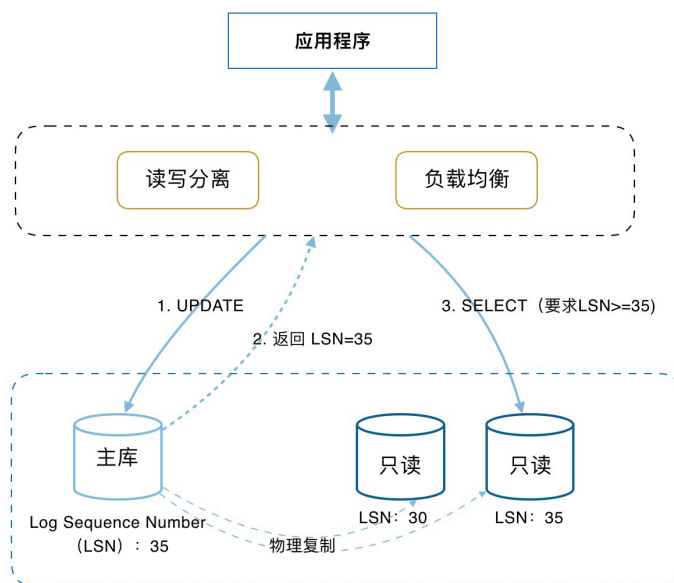
PolarDB MySQL集群自带读写分离功能。应用程序只需连接一个集群地址，写请求会自动发往主节点，读请求会自动根据各节点的负载（当前未完成的请求数）发往主节点或只读节点。

功能优势

- 读一致性

在PolarDB的链路中间层进行读写分离时，中间层会追踪（track）各个节点已经应用（apply）的重做日志（redolog）位点，即日志序号（LSN）；同时，每次数据更新时将此次更新的位点记录为Session LSN。当PolarDB收到新的请求时，将Session LSN与各个节点的LSN进行比较，仅将请求发往LSN早于Session LSN的节点，从而保证会话的一致性。表面上看该方案可能导致主节点压力大，但是因为PolarDB是物理复制，速度极快，在上述场景中，当更新完成后，返回客户端结果时复制就同步在进行，而当下一个读请求到来时，主节点和只读节点之间的数据复

制极有可能已经完成，然后大多数应用场景都是读多写少，所以经验证在该机制下既保证了会话一致性，也保证了读写分离负载均衡的效果。



- 原生支持读写分离，提升性能

如果您在云上通过自己搭建代理层实现读写分离，在数据到达数据库之前需要经历多个组件的语句解析和转发，对响应延迟有较大的影响。而PolarDB读写分离在已有的高安全链路中直接内置，没有任何额外的组件来消耗时间，能够有效降低延迟，提升处理速度。

- 维护方便

在传统模式下，您需要在应用程序中配置主节点和每个只读节点的连接地址，并且对业务逻辑进行拆分，才能实现将写请求发往主节点而将读请求发往各个节点。

PolarDB提供集群地址，应用程序连接该地址后即可对主节点和只读节点进行读写操作，读写请求会被自动转发，转发逻辑完全对使用者透明，可降低维护成本。

同时，您只需添加只读节点的个数，即可不断扩展系统的处理能力，应用程序无需做任何修改。

- 节点健康检查，提升数据库系统的可用性

读写分离模块自动对集群内的所有节点进行健康检查，当发现某个节点宕机或者延迟超过阈值后，将不再分配读请求给该节点，读写请求在剩余的健康节点间进行分配。以此确保单个只读节点发生故障时，不会影响应用的正常访问。当节点被修复后，PolarDB会自动将该节点纳回请求分配体系内。

- 免费使用，降低资源及维护成本

免费提供读写分离功能，无需支付任何额外费用。

转发逻辑

- 可读可写模式转发逻辑：
 - 只发往主节点
 - 所有DML操作（INSERT、UPDATE、DELETE、SELECT FOR UPDATE）。
 - 所有DDL操作（建表/库、删表/库、变更表结构、权限等）。
 - 所有事务中的请求。
 - 用户自定义函数。
 - 存储过程。
 - EXECUTE语句。
 - [Multi Statements](#)。
 - 使用到临时表的请求。
 - SELECT last_insert_id()。
 - 所有对用户变量的查询和更改。
 - SHOW PROCESSLIST。
 - KILL（SQL语句中的KILL，非命令KILL）。
 - 发往只读节点或主节点
 - 非事务中的读请求。
 - COM_STMT_EXECUTE命令。
 - 总是发往所有节点
 - 所有系统变量的更改。
 - USE命令。
 - COM_STMT_PREPARE命令。
 - COM_CHANGE_USER/COM_QUIT/COM_SET_OPTION等命令。
- 只读模式转发逻辑：
 - 不允许执行DDL、DML操作。
 - 所有请求按照负载均衡的方式转发到各只读节点。
 - 所有请求不会转发到可读可写节点。

使用限制

- 若执行了[Multi Statements](#)或存储过程，当前连接的后续请求会全部路由到主节点，需断开当前连接并重新连接才能恢复读写分离。

- 集群地址在只读模式下，不支持设置环境变量，例如 `set @endtime=now(); select * from tab where dt < @endtime`，可能导致查询无法获得正确的环境变量。
- 使用视图时无法保证会话一致性，例如 `CREATE VIEW tab2 AS SELECT * FROM tab1; INSERT INTO tab1(key, value) (1, 1); SELECT * FROM tab2 where key=1`；可能会因为延迟而查不到结果。

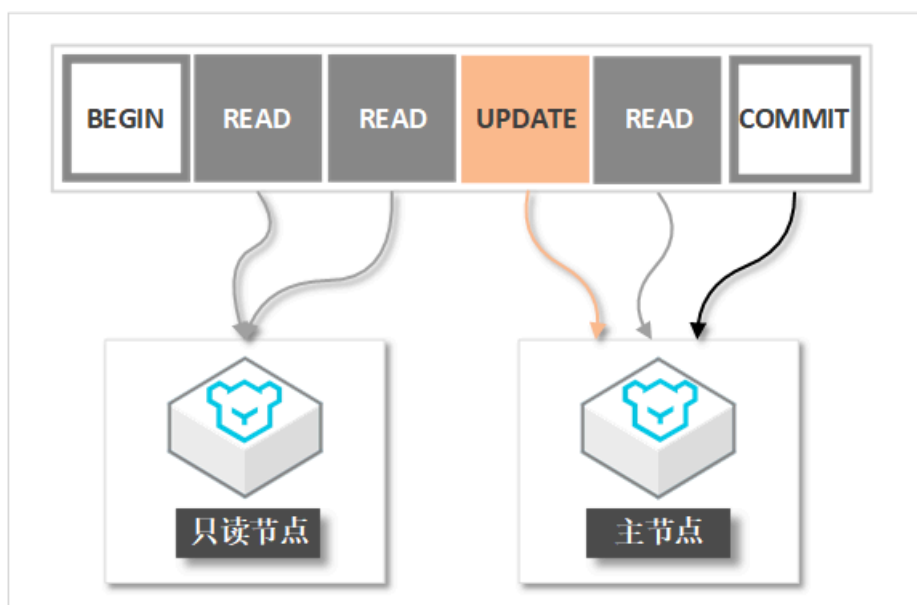
创建或修改集群地址

创建或修改集群地址具体操作方式请参见[修改和释放集群地址](#)。

高级选项-事务拆分

在默认的Read Committed隔离级别下，当PolarDB接收到开启事务的语句（例如begin或set autocommit=0）时，不会立即开启事务，而是在发生写操作时才正式开启事务。

默认情况下，PolarDB会将事务内的所有请求都发送到主节点以保障事务的正确性，但是某些框架会将所有请求封装到事务中，导致主节点负载过大。此时您可以开启事务拆分功能，开启后PolarDB会识别当前事务的状态，将正式开启事务前的读请求通过负载均衡模块分流至只读节点。



说明：

某些业务对全局一致有要求，开启事务拆分后将不满足全局一致，因此在事务拆分前请充分评估事务拆分功能是否适用于您的业务。

1. 单击修改配置。

2. 在弹出的对话框，选择**事务拆分为开启**。

编辑地址配置 如何设置集群地址?

读写模式

☐ 只读 ☒ 可读可写 (自动读写分离)

节点配置

读负载均衡

可选节点

无节点

☐ 0 项

已选节点

☐ 只读节点
☐ 主节点:

☐ 2 项

节点的选择不影响读写模式。不论是否选中主节点，写操作都会发向主节点。

新节点自动加入

☒ 开启 ☐ 关闭

高级配置

负载均衡策略

基于负载的自动调度

一致性级别

会话一致性 (推荐)

主库不接受读

☐ 开启 ☒ 关闭

事务拆分

☒ 开启 ☐ 关闭

连接池

关闭

确定

取消



说明:

开启事务拆分后将新的连接生效，现有的连接需重新连接才生效。

3. 单击**确定**。

高级选项-一致性级别

详情请参见[PolarDB一致性级别](#)。

FAQ

1. 为什么刚插入的语句，立即查的时候查不到？

答：读写分离的架构下，主节点和只读节点之间复制会有延迟，但PolarDB支持会话一致性，即同一个会话内保证能读到之前的更新。

2. 为什么只读节点没有压力？

答：默认情况下事务中的请求都会路由到主节点，若是用sysbench做压测，0.5版本的sysbench可以加上--oltp-skip-trx=on，1.0版本的sysbench可以加上--skip-trx=on去掉事务，若业务上因为事务较多导致只读库负载过低，可以提交工单开启读写分离下的分布式事务。

3. 为什么某个节点请求数比别的节点多？

答：当前是根据负载来分发请求的，负载小的节点接收的请求数会更多。

4. 是否支持0毫秒延迟的读取？

答：PolarDB集群的主节点和只读节点在正常负载情况下，具有毫秒级的延迟，读写分离连接地址暂时不支持在数据写入后0毫秒的读取。如果要求0毫秒延迟的读取，可使用主地址（动态指向PolarDB主节点）将读写请求发给主节点。

5. 新增的只读节点会自动加入到读写分离吗？

新增只读节点之后新建的读写分离连接会转发请求到该只读节点。新增只读节点之前建立的读写分离连接不会转发请求到新增的只读节点，需要断开该连接并重新建立连接，例如，重启应用。

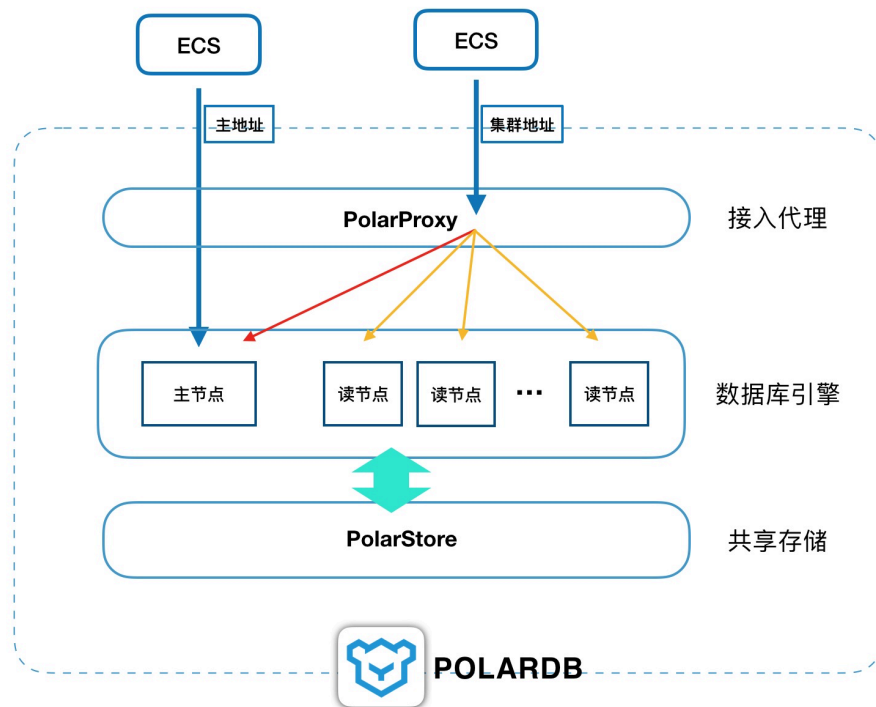
相关API

API	描述
#unique_151	创建PolarDB集群的公网地址。
#unique_155	创建PolarDB自定义集群地址。
#unique_150	查询PolarDB集群的地址信息。
#unique_156	修改PolarDB集群地址属性。
#unique_152	修改PolarDB集群默认访问地址前缀。
#unique_153	释放PolarDB集群地址（除了自定义集群地址的私网地址）。
#unique_157	释放PolarDB自定义集群地址。

7.3.5 PolarDB一致性级别

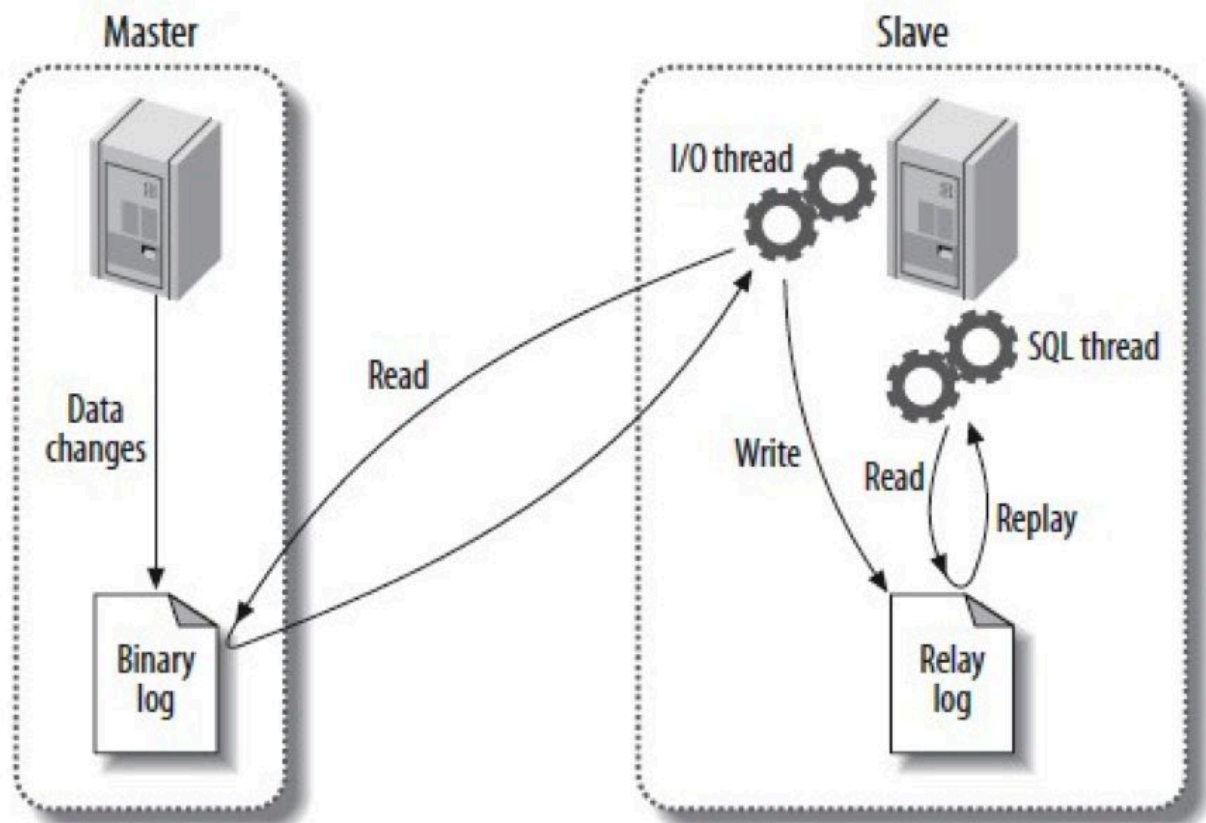
PolarDB架构

PolarDB是一个由多个节点构成的数据库集群，一个主节点，多个读节点。对外默认提供两个地址，一个是集群地址，一个是主地址，推荐使用集群地址，因为它具备读写分离功能可以把所有节点的资源整合到一起对外提供服务。



MySQL读写分离解决和引入

用过MySQL的都知道，MySQL的主从复制简单易用，非常流行，通过把主库的Binlog异步地传输到备库并实时应用，一方面可以实现高可用，另一方面备库也可以提供查询，来减轻对主库的压力。

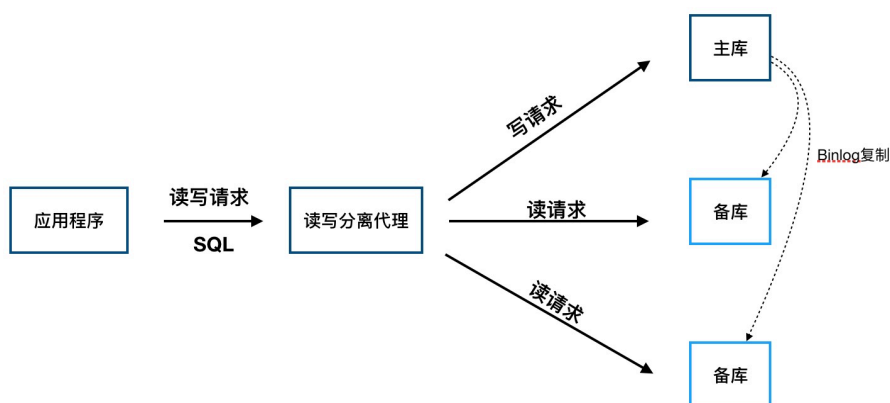


虽然备库可以提供查询，但存在两个问题：

一是主库和备库一般提供两个不同的访问地址，应用程序端需要选择使用哪一个，对应用有侵入。

二是MySQL的复制是异步的，即使是半同步也没办法做到100%强同步，因此备库的数据并不是最新的，有延迟，无法保证查询的一致性。

为了解决第一个问题，我们引入了读写分离代理。一般的实现是，代理会伪造成MySQL与应用程序建立好连接，解析发送进来的每一条SQL，如果是UPDATE、DELETE、INSERT、CREATE等写操作则直接发往主库，如果是SELECT则发送到备库。



但是第二个问题，延迟导致的查询不一致，还是没有解决，使用时，就不可避免地会遇到备库SELECT查询数据不一致的现象（因为主备有延迟）。MySQL负载低的时候延迟可以控制在5秒内，但当负载很高时，尤其是对大表做DDL（比如加字段）或者大批量插入的时候，延迟会非常严重。

PolarDB最终一致性、会话读一致性以及全局一致性

- **最终一致性**：PolarDB采用异步物理复制方式在主库和只读库间做数据同步，在主库更新后，相关的更新会apply到只读库，具体的延迟时间与写入压力有关，一般在毫秒级别，通过异步复制的方式实现主库和只读库之间的最终数据一致。
- **会话读一致性**：为了解决**最终一致性**会出现的查询不一致，PolarDB利用自身物理复制速度快的优点，将查询发给已经更新了数据的只读节点，详细原理请参见[实现原理](#)。
- **全局一致性**：部分应用场景，除了会话内部有逻辑上的因果依赖关系，会话之间也有依赖关系，**会话读一致性**无法满足需求，就需要使用**全局一致性**。

PolarDB读写分离的会话读一致性

PolarDB是读写分离的架构，传统的读写分离都只提供最终一致性的保证，主从复制延迟会导致从不同节点查询到的结果不同，比如一个会话内连续执行以下查询：

```
INSERT INTO t1(id, price) VALUES(111, 96);
UPDATE t1 SET price = 100 WHERE id=111;
```

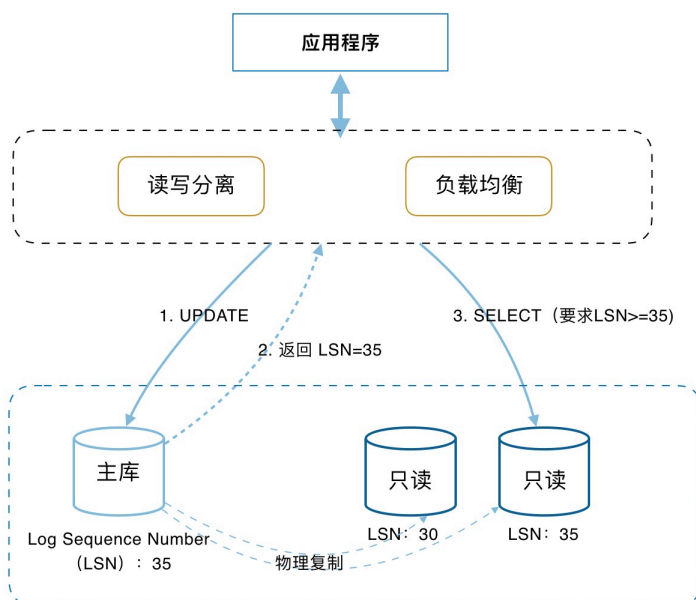


```
SELECT price FROM t1;
```

在读写分离的下，最后一个查询的结果是不确定的，因为读会发到只读库，在执行SELECT时之前的更新是否同步到了只读库时不确定的，因此结果也是不确定的；因为有问题，所以就要求应用程序去适应最终一致性，而一般的解决方法是：将业务做拆分，有高一致性要求的请求直连到主库，可以接受最终一致性的部分走读写分离；显然这样会增加应用开发的负担，还会增大主库的压力，影响读写分离的效果。

为了解决这个问题，在PolarDB中我们提供了会话一致性或者说因果一致性的保证，会话一致性即保证同一个会话内，后面的请求一定能够看到此前更新所产生版本的数据或者比这个版本更新的数据，保证单调性，就很好的解决了上面这个例子的问题。

实现原理

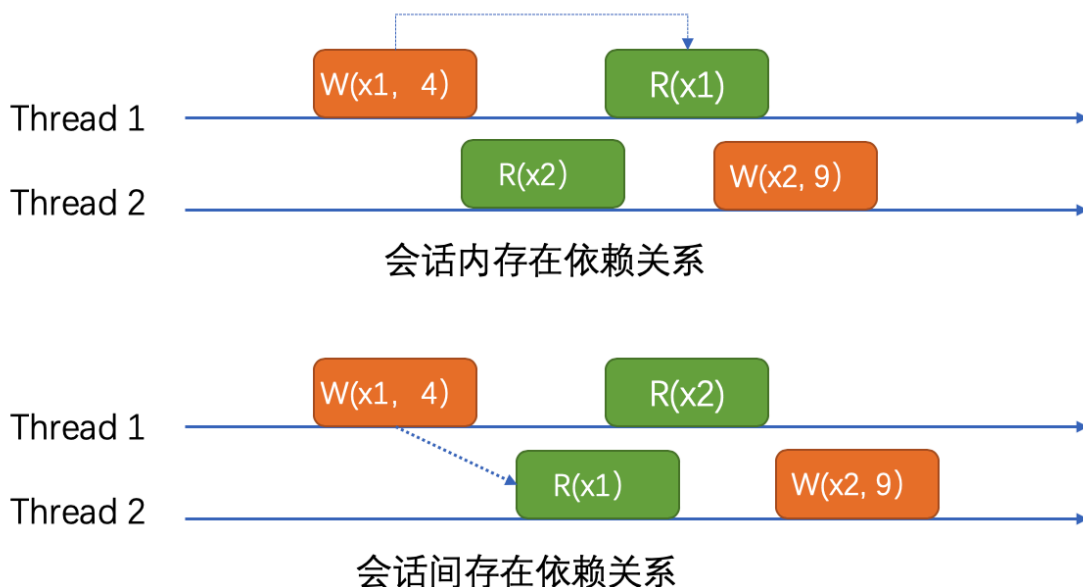


在PolarDB的链路中间层做读写分离的同时，中间层会跟踪各个节点已经应用了的redo日志位点即LSN，同时每次更新时会记录此次更新的位点为Session LSN，当有新请求到来时我们会比较Session LSN和当前各个节点的LSN，仅将请求发往LSN大于或等于Session LSN的节点，从而保证了会话一致性；表面上看该方案可能导致主库压力大，但是因为PolarDB是物理复制，速度极快，在上述场景中，当更新完成后，返回客户端结果时复制就同步在进行，而当下一个读请求到来时主从极有可能已经完成，然后大多数应用场景都是读多写少，所以经验证在该机制下即保证了会话一致性，也保证了读写分离负载均衡的效果。

PolarDB全局一致性

PolarDB当前的架构是一写多读的分布式架构，该架构下只读节点和读写节点之间存在延迟，会导致一些一致性问题的出现，例如刚执行完的更新，立即查询查不到。为此PolarDB的读写分离模块实现

了会话一致的功能，保证同一个会话内的请求能保证因果一致，但是有一些应用场景，除了会话内部有逻辑上的因果依赖关系，会话之间也有依赖关系，那么会话一致就不够用了，因此就需要本文提供的全局一致的功能。



例如使用连接池的场景下，同一个线程的请求有可能通过不同连接发出去，在数据库侧看来这些请求属于不同会话，但是业务逻辑上这些请求有前后依赖关系，那么就需要全局一致来保证正确性。



说明：

全局一致行是通过跟踪复制位点来实现的，在主从延迟较高时，可能会导致更多的请求路由到主库，造成主库压力大，同时业务延迟也可能增加，此功能适用于读多写少的场景。

一致性级别选择最佳实践

PolarDB一致性级别越高，集群性能越低，推荐使用会话一致性，该级别对性能影响很小而且能满足绝大多数应用场景的需求。

对于有不同会话间一致性需求的可以选择以下方案：

- 使用Hint将特定查询强制发到主库。

```
eg: /*FORCE_MASTER*/ select * from user;
```

- 选择全局一致性。

7.4 私有域名

如果您之前是通过域名连接到数据库，当数据库迁移上云后，想要保留原来的数据库域名，可以通过私有域名功能进行绑定。

应用场景

PolarDB的每一个内网地址，均可以绑定一个私有域名，私有域名仅在当前地域内指定的VPC中生效。私有域名优先级比全球生效的域名优先级高，会优先解析。

例如，原来的数据库的域名是abc.test.com，把数据库迁移到迁移至PolarDB集群，PolarDB集群的连接地址是a.polardb.aliyuncs.com，为了保留原来的域名不变，可以创建一个私有域名将abc.test.com绑定（CNAME）到a.polardb.aliyuncs.com。绑定成功后在指定VPC中访问abc.test.com就可以直接访问到PolarDB集群，如下图所示。



费用说明

PolarDB私有域名功能是通过PrivateZone管理的私有域名映射到PolarDB的私网地址来实现的，PrivateZone会收取少量费用，具体请参见[产品定价](#)。

绑定私有域名

1. 登录[PolarDB控制台](#)。
2. 在页面左上角，选择集群所在的地域。
3. 找到目标集群，单击目标集群ID。

4. 在**基本信息**页面的**访问信息**区域，单击目标私网地址右侧的**绑定私有域名**。

访问信息 ?

主地址 ?

私网

pc-ysql.polardb.rds.aliyuncs.com:3306 编辑 绑定私有域名

公网

申请

私网（经典网络）

申请

集群地址【推荐】 ? 创建自定义地址

默认地址 (pe0) 编辑配置

读写模式

可读可写（自动读写分离）

私网

pc-v1b.rds.aliyuncs.com:3306 编辑 绑定私有域名

公网

申请

私网（经典网络）

申请

> 节点配置

> 高级配置

文档版本：20200709

185

5. 在绑定私有域名对话框中，输入私有域名前缀和私有域名后缀。

绑定私有域名

对于PolarDB的每一个私网地址，均可以绑定一个用户的私有域名，该域名仅在当前Region内指定的VPC中生效。实际上是通过PrivateZone管理的私有域名，通过CNAME 映射到 PolarDB 自带的私网地址上，该功能PrivateZone可能会收取少量费用，[定价文档](#)。

执行此操作时，将会自动创建一个服务关联角色AliyunServiceRoleForPolarDB，来完成PrivateZone服务的相关操作，[服务关联角色说明文档](#)。

私有域名：

1

example.com

2

私有域名格式为"<前缀>.<后缀>"，后缀是PrivateZone的名称，可下拉选择已有的后缀或手动输入新的后缀。

确定

取消

私有域名格式为<前缀>.<后缀>，详细信息请参见下表。

序号	配置	说明
1	私有域名前缀	私有域名前缀由小写字母、数字、中划线（-）中的至少一种组成；以字母开头，以数字或字母结尾；6-30个字符。
2	私有域名后缀（Zone）	您可以下拉选择已有Zone或手动输入新的Zone，更多关于Zone的信息请参见PrivateZone。 说明： - 若PolarDB所在的VPC不在配置的Zone中，系统将会自动绑定VPC和Zone。 - 您可以在PrivateZone控制台查看和管理Zone。

说明：

绑定私有域名时，系统将会自动创建一个服务关联角色（AliyunServiceRoleForPolarDB），详细信息请参见[#unique_159](#)。

6. 单击确定。

186

文档版本：20200709

7. 在**绑定私有域名**对话框中，再次确认域名信息无误后，单击**确定**即可。

绑定私有域名

再次确认

绑定后，域名 (h*****bc) 在您指定的VPC中将被优先映射到 PolarDB的地址 (a*****b.rds.aliyuncs.com) 上，立即生效。

确定

取消



说明：

- 私有域名绑定成功后，会在PrivateZone控制台自动生成对应的记录。
- 私有域名在您指定的VPC中将被优先映射到PolarDB的私网地址上，绑定后立即生效。
- 如果您在PrivateZone控制台对Zone和PolarDB所在的VPC进行了解绑，会导致对应的私有域名失效。PrivateZone绑定或解绑VPC具体信息请参见[PrivateZone文档](#)。

相关API

API	描述
#unique_152	该接口用于修改PolarDB集群的连接地址，包括主地址、默认集群地址、自定义集群地址和私有域名。

8 账号和数据库

8.1 账号概述

本文介绍阿里云控制台账号和PolarDB数据库集群账号的相关内容。

控制台账号

您可以使用以下账号登录控制台：

- 阿里云账号：该账号是阿里云资源的归属和计费主体。购买阿里云产品之前，您需要先注册阿里云账号。
- 子账号（可选）：如果其他人需要使用您账号下的资源，您可以使用RAM控制台创建和管理子账号。子账号本身不拥有资源，且以主账号作为计费主体。

数据库集群账号

您可以使用以下账号登录数据库集群。更多信息请参见[创建数据库账号](#)。

账号类型	说明
高权限账号	• 只能通过控制台创建和管理。 • 一个集群只能有一个高权限账号，可以管理所有普通账号和数据库。 • 开放了更多权限，可满足个性化和精细化的权限管理需求，例如可按用户分配不同表的查询权限。 • 拥有集群中所有数据库的所有权限。 • 可以断开任意账号的连接。
普通账号	• 可以通过控制台或者SQL语句创建和管理。 • 一个集群可以创建多个普通账号，具体的数量与数据库内核有关。 • 需要手动给普通账号授予特定数据库的权限。 • 普通账号不能创建和管理其他账号，也不能断开其他账号的连接。

相关API

API	描述
#unique_162	创建账号。
#unique_163	查看账号列表。
#unique_164	修改账号备注。
#unique_165	修改账号密码。

API	描述
#unique_166	账号授权。
#unique_167	撤销账号权限。
#unique_168	重置账号权限。
#unique_169	删除账号。

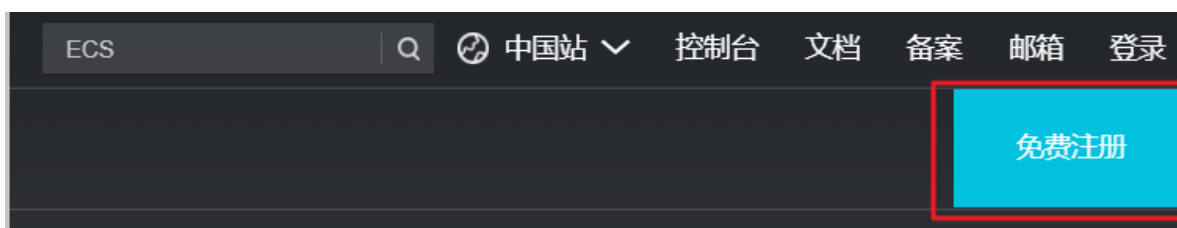
8.2 注册和登录阿里云账号

本文介绍如何注册和登录阿里云账号。

注册阿里云账号

您可以通过两种方式注册阿里云账号：

- 进入[阿里云官网](#)，单击右上角的**免费注册**。



- 直接访问[注册页面](#)。

登录阿里云账号

阿里云账号的登录入口与子账号不同。

- 阿里云账号的[登录入口](#)。

密码登录

扫码登录

扫码登录

淘宝及1688会员可直接使用会员名登录

邮箱/会员名/8位ID

登录密码

登录

[忘记密码](#) [忘记会员名](#) [免费注册](#)

其他方式登录：  

- 子账号的[登录入口](#)。

[使用主帐号登录](#)

子用户登录

登录用户名称

RAM用户新版登录格式：<子用户名称>@<企业别名> 例如：
username@company-alias

下一步

8.3 创建和使用子账号

如果只有您本人使用PolarDB，那么使用阿里云账号即可。如果需要让其他人使用您账号下的资源，请创建子账号。

创建RAM子账号

1. 您可以使用阿里云账号或有RAM权限的子账号来创建子账号，首先需要登录RAM控制台。具体操作如下：
 - 如果使用阿里云账号，请使用[阿里云账号登录](#)。
 - 如果使用子账号，请使用[RAM用户登录](#)。



说明：

子账号登录的格式为子账号名@公司别名。

2. 在左侧导航栏的**人员管理**菜单下，单击**用户**。

3. 单击新建用户。



说明：

单击**添加用户**，可一次性创建多个RAM用户。

4. 输入登录名称和显示名称。

5. 在访问方式区域下，选择控制台密码登录。

6. 控制台密码选择自动生成默认密码或自定义登录密码。

7. 要求重置密码选择用户在下次登录时必须重置密码或无需重置。

8. 多因素认证选择不要求。

9. 单击确认。

在授权页面下为RAM用户授权

1. 在左侧导航栏的**权限管理**菜单下，单击**授权**。

2. 单击**新增授权**。

3. 在**被授权主体**区域下，输入RAM用户名称后，单击需要授权的RAM用户。

4. 在左侧**权限策略名称**列表下，单击需要授予RAM用户的权限策略。

可选策略如下所示：

权限策略名称	说明
AliyunPolarDBReadOnlyAccess	只读访问云数据库PolarDB的权限。
AliyunPolarDBFullAccess	管理云数据库PolarDB的权限。



说明：

在右侧区域框，选择某条策略并单击**×**，可撤销该策略。

5. 单击**确定**。

6. 单击**完成**。

在用户页面下为RAM用户授权

1. 在左侧导航栏的**人员管理**菜单下，单击**用户**。

2. 在**用户登录名称/显示名称**列表下，找到目标RAM用户。

3. 单击**添加权限**，被授权主体会自动填入。

4. 在左侧**权限策略名称**列表下，单击需要授予RAM用户的权限策略。

可选策略如下所示：

权限策略名称	说明
AliyunPolardbReadOnlyAccess	只读访问云数据库PolarDB的权限。
AliyunPolardbFullAccess	管理云数据库PolarDB的权限。



说明：

在右侧区域框，选择某条策略并单击×，可撤销该策略。

5. 单击**确定**。

6. 单击**完成**。

登录子账号

前提条件：您已完成上述账号授权步骤。

您可以通过两种地址登录子账号：

- 通用登录地址：[RAM用户登录](#)

如果通过此地址登录，您需手动输入子账号名以及公司别名。格式为子账号名@公司别名。

- 专用登录地址：如果您可以登录[RAM控制台](#)，可以在RAM控制台查看到您公司的子账号登录地址。



如果通过此地址登录，系统将自动为您填写公司别名，您只需输入子账号名。

更多操作

您还可以对子账号进行更多的操作，如把子账号添加到用户组、为子账号分配角色、为用户组或角色授权等。详情请参见[RAM用户指南](#)。

8.4 创建数据库账号

本文为您介绍如何创建数据库账号，以及高权限账号与普通账号的区别。

背景信息

PolarDB MySQL支持两种数据库账号：高权限账号和普通账号。您可以在控制台管理所有账号。



说明：

出于安全原因，PolarDB不提供root账号。

账号类型	说明
高权限账号	- 只能通过控制台创建和管理。 - 一个集群只能有一个高权限账号，可以管理所有普通账号和数据库。 - 开放了更多权限，可满足个性化和精细化的权限管理需求，例如可按用户分配不同表的查询权限。 - 拥有集群中所有数据库的所有权限。 - 可以断开任意账号的连接。
普通账号	- 可以通过控制台或者SQL语句创建和管理。 - 一个集群可以创建多个普通账号，具体的数量与数据库内核有关。 - 需要手动给普通账号授予特定数据库的权限。 - 普通账号不能创建和管理其他账号，也不能断开其他账号的连接。

创建高权限账号

1. 登录[PolarDB控制台](#)。
2. 在控制台左上角，选择集群所在地域。
3. 单击目标集群ID。
4. 在左侧导航栏中，选择**账号管理**。
5. 单击**创建账号**。
6. 设置以下参数：

参数	说明
账号名	填写账号名称。要求如下： - 以小写字母开头，以字母或数字结尾； - 由小写字母、数字或下划线组成； - 长度为2~16个字符。 - 不能使用某些预留的用户名，如root、admin。
账号类型	此处选择 高权限账号 。  说明： 如果已经创建过高权限账号，则无法再选择高权限账号，因为每个集群只能有一个高权限账号。

参数	说明
密码	设置账号的密码。要求如下： - 由大写字母、小写字母、数字或特殊字符组成，至少包含其中三类； - 长度为8~32个字符； - 特殊字符为： ! @ # \$ % ^ & * () - +
确认密码	再次输入密码。
备注	备注该账号的相关信息，便于后续账号管理。要求如下： - 不能以http://或https://开头； - 必须以大小写字母或中文开头； - 可以包含大小写字母、中文、数字、下划线（_）或连字符（-）； - 长度为2~256个字符。

7. 单击**确定**。

创建普通账号

1. 登录[PolarDB控制台](#)。
2. 在控制台左上角，选择集群所在地域。
3. 单击目标集群ID。
4. 在左侧导航栏中，选择**账号管理**。
5. 单击**创建账号**。

6. 设置以下参数：

参数	说明
账号名	填写账号名称。要求如下： - 以小写字母开头，以字母或数字结尾； - 由小写字母、数字或下划线组成； - 长度为2~16个字符； - 不能使用某些预留的用户名，如root、admin。
账号类型	此处选择 普通账号 。
授权数据库	为该账号授予一个或多个数据库的权限。本参数也可以留空，在创建账号后再给账号授权。 - 从左侧选中一个或多个数据库，单击向右的箭头添加到右侧。 - 在右侧框中，为数据库选择读写、只读或仅DML权限。
密码	设置账号密码。要求如下： - 由大写字母、小写字母、数字或特殊字符组成，至少包含其中三类。 - 长度为8~32个字符。 - 特殊字符为!@#\$%^&*()_+-=。
确认密码	再次输入密码。
备注	备注该账号的相关信息，便于后续账号管理。要求如下： - 不能以http://或https://开头。 - 必须以大小写字母或中文开头。 - 可以包含大小写字母、中文、数字、下划线（_）或连字符（-）。 - 长度为2~256个字符。

7. 单击**确定**。

重置高权限账号的权限

如果高权限账号自身出现问题，例如权限被意外回收（REVOKE），您可以通过输入高权限账号的密码来重置高权限账号的权限，使其恢复正常。

1. 登录[PolarDB控制台](#)。
2. 在控制台左上角，选择集群所在地域。
3. 单击目标集群ID。
4. 在左侧导航栏中，选择**账号管理**。
5. 单击**高权限账号**右侧的**重置权限**。
6. 在弹出的对话框中，输入高权限账号的密码，即可重置该账号的权限。

下一步

[查看连接地址](#)

相关API

API	描述
#unique_162	创建账号。
#unique_163	查看账号列表。
#unique_164	修改账号备注。
#unique_165	修改账号密码。
#unique_166	账号授权。
#unique_167	撤销账号权限。
#unique_168	重置账号权限。

8.5 管理数据库账号

本文为您介绍如何管理数据库账号，包括重置高权限账号权限、修改普通账号权限、修改账号密码以及删除账号等。

背景信息

PolarDB MySQL支持两种数据库账号：高权限账号和普通账号。您可以在控制台管理所有账号和数据库。

注意事项

- 出于安全原因，PolarDB不提供root账号。
- 当使用UPDATE或INSERT语句直接修改MySQL库下的权限表来修改数据库账号密码或权限时，会导致只读节点无法同步。因此建议您通过如下方式进行修改：
 - 直接在PolarDB控制台上修改账号密码或权限，详情请参见[修改账号密码](#)。
 - 使用高权限账号通过命令行修改账号密码或权限，详情请参见[通过命令行修改账号密码和权限](#)。

创建数据库账号

具体操作请参见[创建数据库账号](#)。

修改账号密码

- 登录[PolarDB控制台](#)。
- 单击目标集群ID。

3. 在左侧导航栏中，单击**配置与管理 > 账号管理**。

4. 找到目标账号，单击操作栏中的**修改密码**。



5. 在弹出的对话框中，输入新密码和确认新密码，单击**确定**。

重置高权限账号权限

如果高权限账号自身出现问题，例如权限被意外回收（REVOKE），您可以通过输入高权限账号的密码来重置高权限账号的权限，使其恢复正常。

1. 登录**PolarDB控制台**。

2. 单击目标集群ID。

3. 在左侧导航栏中，单击**配置与管理 > 账号管理**。

4. 找到目标账号，单击操作栏中的**重置权限**。

5. 在弹出的对话框中，输入高权限账号的密码，单击**确定**，即可重置该账号的权限。

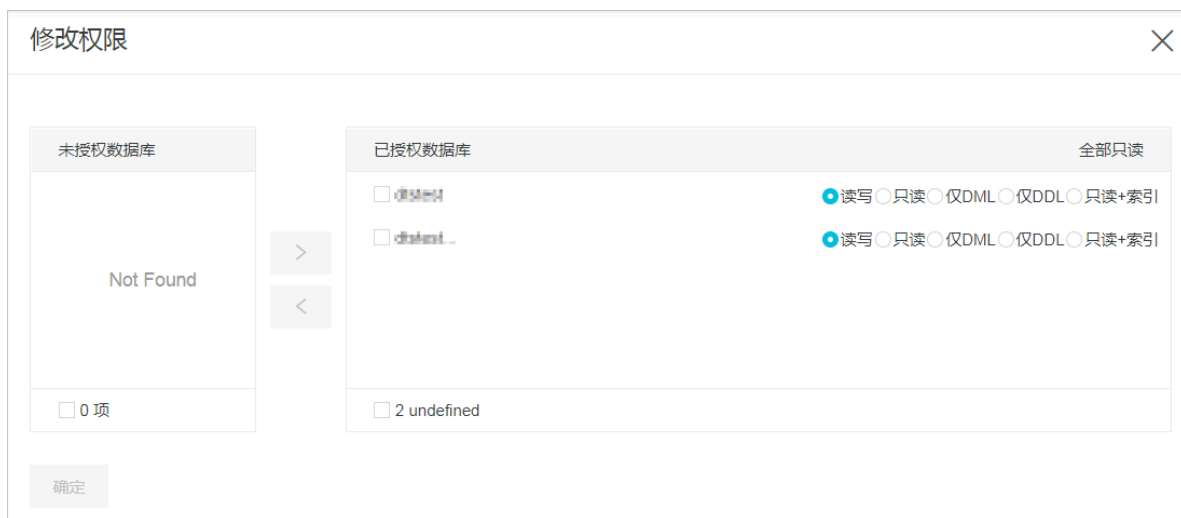
修改普通账号权限

1. 登录**PolarDB控制台**。

2. 单击目标集群ID。

3. 在左侧导航栏中，单击**配置与管理 > 账号管理**。

4. 找到目标账号，单击操作栏中的**修改权限**。



5. 在弹出的对话框中，修改授权数据库以及修改已授权数据库的权限，单击**确定**。

删除账号

1. 登录[PolarDB控制台](#)。
2. 单击目标集群ID。
3. 在左侧导航栏中，单击**配置与管理 > 账号管理**。
4. 找到目标账号，单击操作栏中的**删除**。
5. 在弹出的对话框中，单击**确定**。

通过命令行修改账号密码和权限

- 您可以使用高权限账号通过如下命令行修改账号密码：

- PolarDB MySQL 8.0:

```
ALTER USER {{username}} IDENTIFIED BY '{{password}}'
```

参数	说明
username	待修改密码的账号。
password	账号的密码。

- PolarDB MySQL 5.6:

```
SET PASSWORD FOR 'username'@'host' = PASSWORD('password')
```

参数	说明
username	待修改密码的账号。
host	允许该账号登录的主机，如果允许该账号从任意主机登录，则使用百分号（%）。
password	账号的密码。

- 您可以使用高权限账号通过如下命令行修改权限。

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

参数	说明
privileges	授予该账号的操作权限，如SELECT、INSERT、UPDATE等，如果要授予该账号所有权限，则使用ALL。
databasename	数据库名。如果要授予该账号具备所有数据库的操作权限，则使用星号（*）。

参数	说明
tablename	表名。如果要授予该账号具备所有表的操作权限，则使用星号（*）。
username	待授权的账号。
host	允许该账号登录的主机，如果允许该账号从任意主机登录，则使用百分号（%）。
WITH GRANT OPTION	授予该账号使用GRANT命令的权限，该参数为可选。

相关API

API	描述
#unique_162	创建账号
#unique_163	查看账号列表
#unique_164	修改账号备注
#unique_165	修改账号密码
#unique_166	账号授权
#unique_167	撤销账号权限
#unique_168	重置账号权限
#unique_169	删除账号

8.6 数据库

您可以在控制台创建和管理所有的数据库。

创建数据库

1. 登录[PolarDB控制台](#)。
2. 在控制台左上角，选择集群所在地域。
3. 单击目标集群的ID。
4. 在左侧导航栏中，选择**配置与管理 > 数据库管理**。
5. 单击**创建数据库**。

6. 设置以下参数。

The screenshot shows the '创建数据库' (Create Database) dialog in the PolarDB console. On the left, a sidebar shows a list of existing databases: 'gc', 'lzy', 'test01', and 'web', all in '运行中' (Running) status with 'utf8' character set. The main dialog has the following fields:

- * 数据库(DB)名称:** 'test1' (5/64 characters). A note specifies: '由小写字母、数字、中划线、下划线组成，字母开头，字母或数字结尾，最长64个字符'.
- * 支持字符集:** Radio buttons for 'utf8mb4' (selected), 'utf8', 'gbk', and 'latin1'. A '请选择' (Please select) dropdown is also present.
- 授权账号:** A dropdown menu with a '创建新账号' (Create new account) link.
- 账号权限:** Radio buttons for '读写' (selected), '只读' (Read-only), and '仅DML' (DML only).
- 备注说明:** A text area for remarks, with a character count of '0/256'.
- 确定** (Confirm) button at the bottom.

参数	说明
数据库 (DB) 名称	- 以字母开头，以字母或数字结尾； - 由小写字母、数字、下划线或中划线组成； - 长度为2~64个字符。 - 数据库名称在实例内必须是唯一的。
支持字符集	选择utf8mb4、utf8、gbk或latin1。 如果需要其他字符集，请在右侧下拉菜单中选择需要的字符集。
授权账号	选中需要授权访问本数据库的账号。本参数可以留空，在创建数据库后再绑定账号。 说明： 此处只会显示普通账号，因为高权限账号拥有所有数据库的所有权限，不需要授权。
账号权限	选择要授予账号的权限： 读写 、 只读 或 仅DML 。
备注说明	用于备注该数据库的相关信息，便于后续数据库管理。要求如下： - 不能以http://或https://开头； - 必须以大小写字母或中文开头； - 可以包含大小写字母、中文、数字、下划线"_"或连字符"-"; - 长度为2~256个字符。

7. 单击**确定**。

删除数据库

1. 登录**PolarDB控制台**。

2. 在控制台左上角，选择集群所在地域。
3. 单击目标集群的ID。
4. 在左侧导航栏中，选择**配置与管理 > 数据库管理**。
5. 找到目标数据库，单击**操作**列中的**删除**。
6. 在弹出的对话框中，单击**确定**。

相关API

API	描述
#unique_170	创建数据库。
#unique_171	查看数据库列表信息。
#unique_172	修改数据库描述。
#unique_173	删除数据库。

9 部署架构

9.1 主备切换

POLARDB是一个多节点的集群，其中一个节点是主节点（Master），其他节点为只读节点。除了因系统故障自动进行主备切换外，您也可以手动进行主备切换，指定一个只读节点为新的主节点，适用于高可用演练，或者需要指定某个节点为主节点的场景。

手动切换

1. 登录[POLARDB控制台](#)。
2. 在控制台左上角，选择集群所在地域。
3. 单击目标集群ID。
4. 在**基本信息**页面的**节点信息**区域，单击**主备切换**。

The screenshot shows the PolarDB console interface. On the left is a navigation menu with options like 'Basic Information', 'Configuration and Management', 'Diagnosis and Optimization', and 'Logs and Audit'. The main content area is titled 'Basic Information' and contains sections for 'Main Address', 'Cluster Address', 'Read/Write Mode', and 'Node Information'. The 'Node Information' section is expanded, displaying a table with two nodes. The 'Master Switch' button is highlighted in the top right corner of the node information section.

节点名称	可用区	状态	当前角色	规格	最大IOPS	优先级	操作
pi-bp	杭州 可用区I	运行中	主节点	4核 16GB	32000	1	重启
pi-bp	杭州 可用区I	运行中	只读节点	4核 16GB	32000	1	重启

5. 在弹出的对话框中，选择**新主节点**，单击**确认**。

主备切换

提升一个只读节点为新的主节点。

新主节点:

请选择

切换时可能会出现30秒左右的闪断，请确保应用具备重连机制。

确认

取消

**说明：**

如果您没有选择**新主节点**，系统将自动选举优先级最高的只读节点作为新主节点进行切换。切换时可能会出现30秒左右的闪断，请确保应用具备重连机制。

自动切换

POLARDB采用双活（Active-Active）的高可用集群架构，可读写的主节点和只读节点之间自动进行故障切换（Failover），系统自动选举新的主节点。

POLARDB每个节点都有一个故障切换（Failover）优先级，决定了故障切换时被选举为主节点的概率高低。当多个节点的优先级相同时，则有相同的概率被选举为主节点。

自动选举主节点按以下步骤进行：

1. 系统找出当前可以被选举的所有只读节点。
2. 选择优先级最高的一个或多个只读节点。
3. 如果切换第一个节点失败（例如，网络原因、复制状态异常等），则会尝试切换下一个，直至成功。

相关API

API	描述
#unique_175	手动对POLARDB集群进行主备切换，可以指定一个只读节点为新的主节点。

9.2 多可用区部署和更换主可用区

PolarDB MySQL支持创建多可用区的集群。相比单可用区集群，多可用区集群具备更高的容灾能力，可以抵御机房级别的故障。本文将为您介绍如何实施多可用区部署以及如何更换主可用区。



说明：

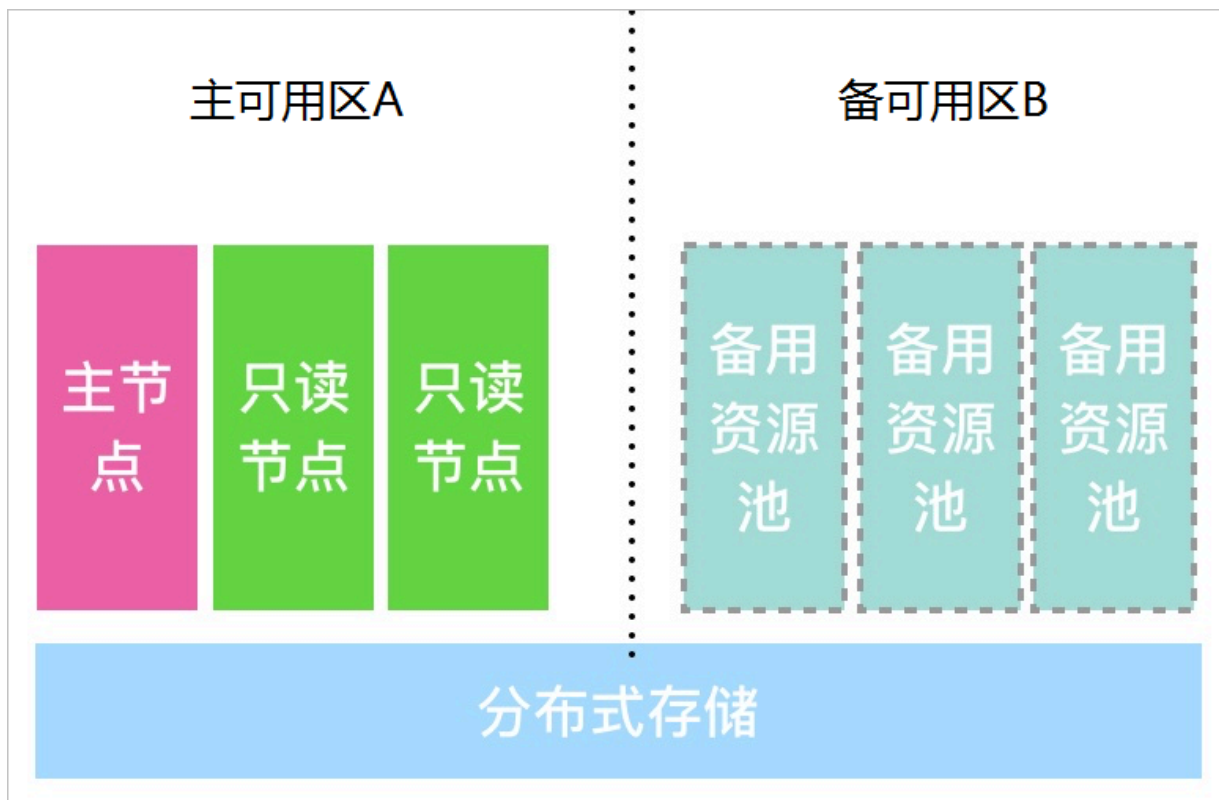
- PolarDB PostgreSQL请参见[#unique_176](#)。
- PolarDB兼容Oracle数据库请参见[#unique_177](#)。

前提条件

- 可用区数量为两个及以上的地域。
- 目标可用区拥有足够计算资源。

多可用区架构

使用多可用区集群时，数据分布在多个可用区内。计算节点暂时要求位于主可用区，PolarDB会在备可用区预留足够的资源用于主可用区故障时进行故障切换。多可用区架构如下。



费用

多可用区功能不需要支付额外费用。

**说明：**

单可用区集群也会免费升级至多可用区集群。

如何实现多可用区架构

当满足前提条件时，[新建集群](#)会默认为多可用区集群。

存量的单可用区集群也会升级至多可用区集群，该升级通过在线迁移数据的方式自动完成，对您的业务无任何影响。

包年包月

按小时付费

① 包年包月集群中也存在按小时付费的计费项，共两个：存储空间、SQL洞察（可选）。

地域

华东1（杭州）

华东2（上海）

华北1（青岛）

华北2（北京）

华北5（呼和浩特）

华南1（深圳）

香港

华北3（张家口）

所在的地理位置，购买后无法更换。
请确保POLARDB与需要连接的ECS位于同一地域，否则它们内网无法直接互通。

创建方式

默认创建

从RDS克隆

从RDS迁移

创建一个全新的POLARDB

主可用区

华东1可用区G

华东1可用区I

您可以选择将POLARDB与ECS创建在同一可用区或不同的可用区。
在有二个及以上可用区的地域，POLARDB会自动复制数据到备可用区，用于灾难恢复。

网络类型

专有网络

请确保与需要连接的ECS在同一个VPC网络！如需使用其他VPC网络或者VPC交换机，请到[控制台](#)创建。

查看集群所属可用区

1. 登录[PolarDB控制台](#)。
2. 在控制台左上角，选择集群所在地域。
3. 单击目标集群ID。
4. 在**基本信息**页面，查看数据分布的可用区。

基本信息

配置与管理

账号管理

数据库管理

备份恢复

参数配置

基本信息

集群ID

pc-

编辑

地域

华东1（杭州）

兼容数据库

MySQL 5.6

VPC

vpc-

可维护窗口

02:00-03:00 [修改](#)

集群名称

数据分布的可用区

杭州可用区G（主），杭州可用区I

编辑

状态

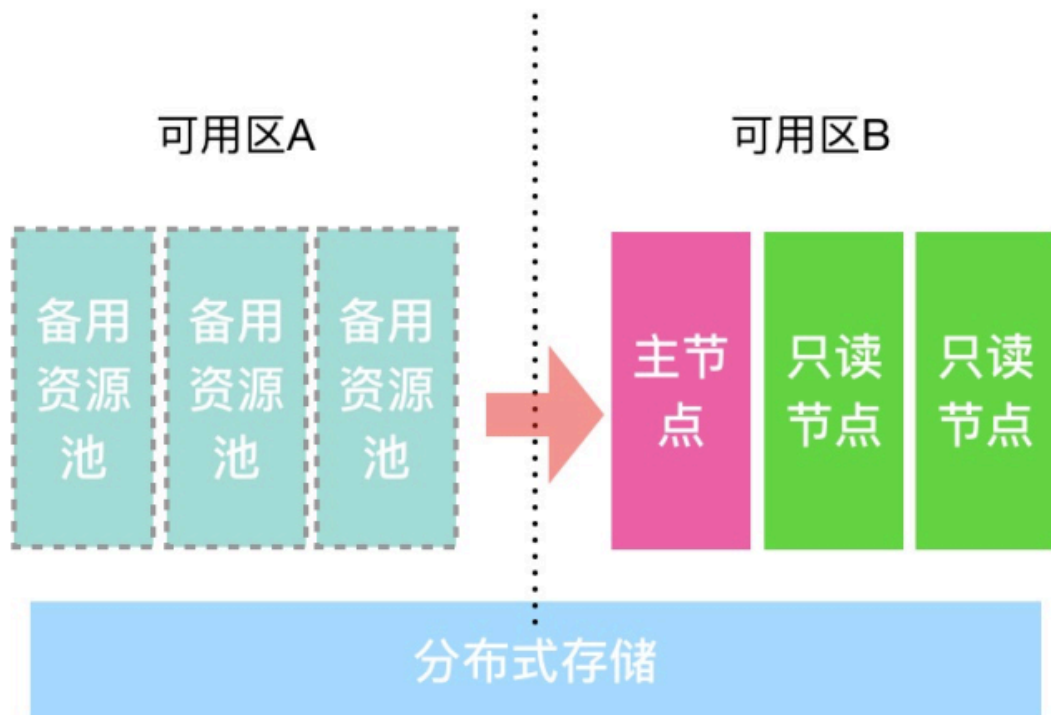
运行中

交换机

vsw-

更换主可用区

PolarDB支持更换主可用区，您可以通过该功能将数据库集群计算节点迁移到其他可用区，适用于灾难恢复或者让ECS就近访问的场景。



1. 登录[PolarDB控制台](#)。
2. 在控制台左上角，选择集群所在地域。
3. 单击目标集群ID。
4. 在**基本信息**页面，单击**迁移可用区**。



5. 在**更换主可用区**对话框中，选择**目标可用区**和**目标交换机**。

更换主可用区

* 目标可用区

请选择

* 目标交换机

请选择

数据库节点

☒ 全部

生效时间

☒ 立即生效

更换主可用区将迁移全部数据库节点到新的可用区，连接地址不变，但可能会使用新可用区的IP。该操作可能会影响数据库的可用性，参见[文档](#)

确定取消



说明：

- 如果目标可用区是备可用区，则不需要迁移数据。系统只需要切换数据库计算节点，因此可以达到比较快的跨机房切换效果（平均耗时5分钟/节点），该操作常用于容灾演练。
- 如果目标可用区不在备可用区，则需要迁移数据。系统执行迁移时间长短跟数据容量有关，可能需要几个小时，请谨慎操作。该操作一般用于调整应用和数据库的可用区分布，达到就近访问数据库的目的。

6. 单击**确认**。



注意：

更换主可用区后，数据库连接地址（集群访问地址和主访问地址）不变，但使用的虚拟交换机vSwitch和IP地址可能会发生变化。该操作可能会影响数据库的可用性，请谨慎操作。

9.3 全球数据库

9.3.1 新建全球数据库和添加只读集群

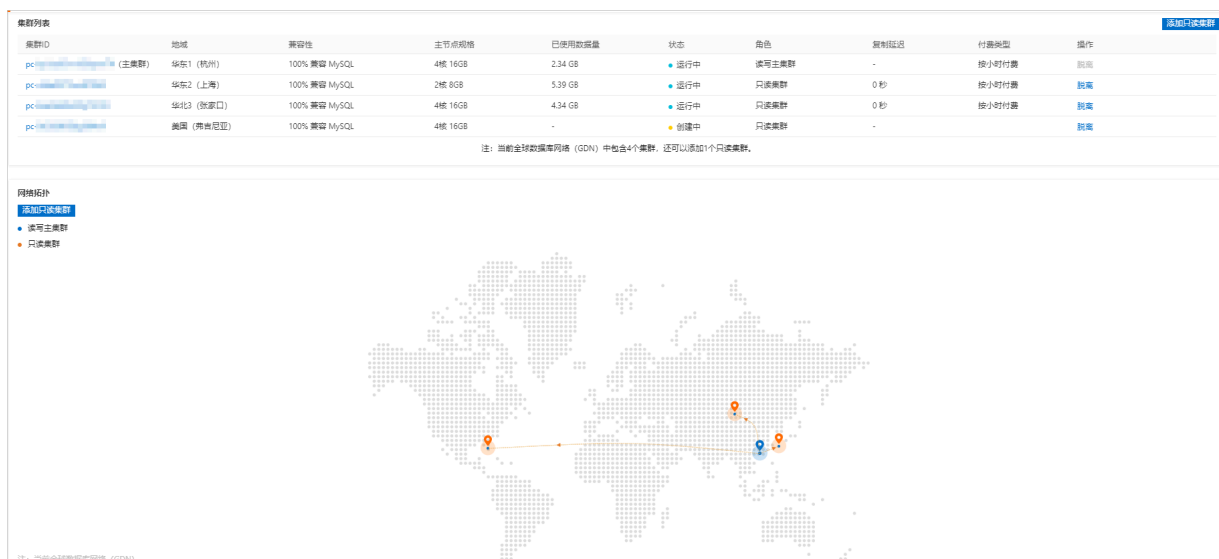
全球数据库（Global Database Network，简称GDN）是由分布在全球不同地域的多个PolarDB集群组成的网络，网络中所有集群的数据保持同步。当您的业务部署在多个地域时，实现应用访问数据库的低延迟和高稳定性。本文介绍如何新建全球数据库以及添加只读集群。

前提条件

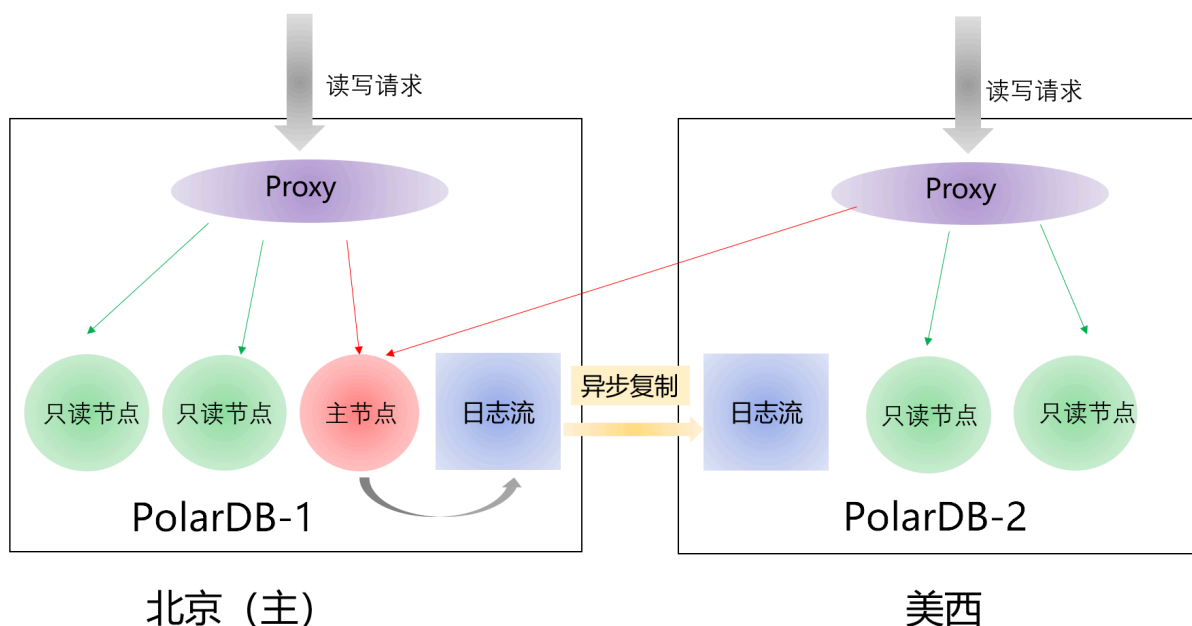
- 已通过PolarDB全球数据库公测申请，公测申请地址请参见[PolarDB全球数据库GDN公测申请](#)。
- PolarDB集群需为PolarDB MySQL 8.0版本。
- 已创建主PolarDB集群，创建PolarDB集群请参见[创建PolarDB MySQL数据库集群](#)。
- 主集群的内核小版本需要为最新版本，升级版本请参见[小版本升级](#)。

全球数据库支持全局读写分离

全球化公司的业务会部署在多个地域，一般情况下，数据库在中心地域。如果全球的应用都跨地域来访问中心的数据库，网络延迟会导致性能低下，带来不良体验。全球数据库内的主集群（Primary Cluster）与只读集群（Read-only Cluster）间使用了异步复制机制，所有集群间的数据都能保持同步且网络延迟被控制在2秒之内。这将大大降低非中心区域应用访问的读取延迟。



此外，除了降低非中心区域应用访问的读取延迟，GDN还提供了全局读写分离功能。借助GDN的全局读写分离功能，您可以在多达五个非中心区域中各创建一个只读集群，当应用需要访问数据库时，读请求将被直接发送到本地只读集群，而写请求则自动转发到中心地域的主集群。您无需在应用程序上修改代码拆分读写请求，即可轻松构建全球部署的应用架构！



注意事项

- 一个GDN中包含一个主集群和最多四个只读集群。
- 主集群和只读集群不做互相约束，各自有独立的配置，可以有不同的规格、不同的白名单、不同的参数值等。
- 一个集群只能属于一个GDN。
- 一个GDN最多可覆盖5个地域，多个集群可以存在于同一个地域、同一个可用区。
- 只读集群只能新建，不能将已经创建好的集群添加为只读集群。

费用

GDN功能在公测期间免费，商业化后可能会收费。新建集群的费用请参见[#unique_67](#)。

性能测试结果

PolarDB全球数据库可以帮助业务跨地域复制数据，创建异地只读数据库并保持数据同步，不会影响现有主数据库的稳定性和性能，通过物理日志、并行处理等技术降低异地复制的延迟。

测试集群规格和拓扑	Sysbench压力测试	峰值QPS/TPS	美西集群只读节点延迟
GDN张家口-美西地域 PolarDB MySQL 16核128GB	OLTP_INSERT插入负载	82655/82655	<1秒
	OLTP_WRITE_ONLY只读负载	157953/26325	<1秒
	OLTP_READ_WRITE读写负载	136758/6837	<1秒

新建全球数据库

1. 登录PolarDB控制台。
2. 单击左侧导航栏中的**全球数据库网络（GDN）**。
3. 在**全球数据库网络（GDN）**页面，单击**创建全球数据库网络**。

全球数据库（Global Database Network，简称GDN）

创建全球数据库网络					
全球数据库网络ID/名称	状态	兼容数据库	包含的集群	创建时间	操作
gdn- test	运行中	100% 兼容 MySQL 8.0	pc- test	2020年4月24日 09:36:24	添加只读集群 删除

4. 在**创建全球数据库网络（GDN）**对话框中，配置以下信息。

配置	说明
名称	GDN的名称，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。
主地域	选择主集群所在的地域。  说明： 请先确认要选择哪个集群作为主集群，并在此选择该集群所在的地域。
主集群	选择主集群。  说明： 仅支持PolarDB MySQL 8.0版本。

5. 完成配置后，单击**确定**。

创建只读集群

1. 登录PolarDB控制台。
2. 单击左侧导航栏中的**全球数据库网络（GDN）**。
3. 找到目标全球数据库ID，单击**操作**栏中的**添加只读集群**。

创建全球数据库网络					
全球数据库网络ID/名称	状态	兼容数据库	包含的集群	创建时间	操作
gdn- test	运行中	100% 兼容 MySQL 8.0	pc- test	2020年4月24日 09:36:24	添加只读集群 删除

 **说明：**

GDN不支持添加现有集群为只读集群。

4. 在弹出的云数据库PolarDB售卖页，选择**包年包月**或**按量付费**。

- **包年包月**：在创建集群时支付计算节点（一个主节点和一个只读节点）的费用，而存储空间会根据实际数据量按小时计费，并从账户中按小时扣除。如果您要长期使用该集群，**包年包月**方式更为经济，而且购买时长越长，折扣越多。
- **按量付费（按小时付费）**：无需预先支付费用，计算节点和存储空间（根据实际数据量）均按小时计费，并从账户中按小时扣除。如果您只需短期使用该集群，可以选择**按量付费**，用完即可释放，节省费用。

5. 设置如下参数。

类别	配置	说明
基本信息	地域	选择要创建只读集群的地域，创建后无法更改。  说明： 请确保PolarDB与需要连接的ECS创建于同一个地域，否则它们无法通过内网互通，只能通过外网互通，无法发挥最佳性能。
	创建方式	选择 创建只读集群 。
	全球数据库网络	选择需要加入的GDN。  说明： 默认为创建只读集群前选择的GDN。
	主可用区	集群的主可用区。 • 可用区是地域中的一个独立物理区域，不同可用区之间没有实质性区别。 • 您可以选择将PolarDB与ECS创建在同一可用区或不同的可用区。 • 您只需要选择主可用区，系统会自动选择备可用区。
	网络类型	仅支持 专有网络 。
	VPC网络	选择需要加入的VPC网络。请确保PolarDB与需要连接的ECS创建于同一个VPC，否则它们无法通过内网互通，无法发挥最佳性能。
	VPC交换机	选择需要加入的VPC交换机。
实例配置	兼容性	仅支持 MySQL 8.0 。
	系列	您可以选择实例系列为 标准版 。

类别	配置	说明
	节点规格	按需选择。所有PolarDB节点均为独享型，性能稳定可靠。关于各规格的具体信息，请参见#unique_67。  说明： 只读集群规格无需和主集群规格一致。
	节点个数	- 无需选择。集群默认包含一个主节点和一个只读节点。只读节点与主节点规格相同。 - 如果主节点故障，系统会自动将只读节点切换为新的主节点，并重新生成一个只读节点。 - 关于只读节点的更多信息，请参见#unique_138。
	存储费用	无需选择。系统会根据实际数据使用量按小时计费，详情请参见#unique_67。
购买量	购买时长	选择PolarDB只读集群的购买时长。  说明： 该配置仅包年包月情况下可见。
	购买数量	选择PolarDB只读集群的购买数量。  说明： 一个GDN中最多支持4个只读集群。

6. 单击右侧**立即购买**。

7. 在**确认订单**页面，确认订单信息，阅读和勾选服务协议，单击**去支付**。

8. 在**支付**页面，确认未支付订单信息和支付方式，单击**订购**。

常见问题

Q：一个阿里云账号可以创建多少个GDN？

A：可以创建GDN的数量没有限制。

9.3.2 管理全球数据库

全球数据库（Global Database Network，简称GDN）是由分布在全球不同地域的多个PolarDB集群组成的网络，网络中所有集群的数据保持同步。当您的业务部署在多个地域时，实现应用访问数据库的低延迟和高稳定性。本文介绍如何移除只读集群以及释放全球数据库。

移除只读集群

1. 登录[PolarDB控制台](#)。

2. 单击左侧导航栏中的。
3. 找到目标GDN，单击目标。

全球数据库（Global Database Network，简称GDN）						
创建全球数据库网络						
全球数据库网络ID/名称	状态	兼容数据库	包含的集群	创建时间	操作	
gdn- GDN测试	运行中	100% 兼容 MySQL 8.0		2020年3月23日 13:46:54	添加只读集群 删除	

4. 在集群列表区域，单击操作栏的脱离。



说明：

- 整个脱离过程耗时5分钟左右。
- 脱离过程中，GDN以及只读集群的所有连接地址的连接均不会中断，不影响数据库的正常访问。
- 只有只读集群支持脱离GDN，主集群无法脱离。
- 脱离后只读集群将中断与主集群的数据同步，同时设置该集群为读写。
- 脱离后该集群不能再作为只读集群加入GDN，请谨慎操作。

集群列表										添加只读集群
集群ID	地域	兼容性	主节点规格	已使用数据量	状态	角色	复制延迟	付费类型	操作	
pc- (主集群)	华东1（杭州）	100% 兼容 MySQL	4核 16GB	2.34 GB	运行中	读写主集群	-	按小时付费	脱离	
pc- (只读集群)	华东2（上海）	100% 兼容 MySQL	2核 8GB	5.38 GB	运行中	只读集群	1 秒	按小时付费	脱离	

5. 在从全球数据库网络（GDN）中脱离对话框，单击确认。

释放全球数据库

1. 登录PolarDB控制台。
2. 单击左侧导航栏中的。
3. 找到目标GDN，单击操作栏的删除。



注意：

- 当全球数据库中只有一个主集群时，才可以释放全球数据库。
- GDN释放后将无法恢复，请谨慎处理。
- 连接GDN连接地址的应用程序将无法访问数据库，请及时修改。

4. 在删除全球数据库网络对话框，单击确定。

10 弹性升降级

10.1 变更配置

本文介绍如何升级或降级集群的规格，新规格会立即开始生效（每个节点需要5到10分钟）。您可以根据业务需求变更集群的配置。

前提条件

集群没有正在进行的配置变更时，才可以变更集群规格。

背景信息

PolarDB支持在线扩容，变更配置过程中无需对数据库加锁。

您可以根据业务需求变更集群的配置。PolarDB支持三维扩展能力：

- 计算能力纵向扩展：集群规格升降配。本文介绍详细信息。
- 计算能力横向扩展：增加或减少只读节点。具体操作说明，请参见[增加或删除节点](#)。
- 存储空间横向扩展：PolarDB采用Serverless架构，无需手动设置容量或扩缩容，容量随用户数据量的变化而自动在线调整。

变更配置的费用说明

详情请参见[#unique_180](#)。

注意事项

- 您只能对整个集群进行规格升降级，无法对集群中的单个节点进行规格升降级。
- 集群规格的升降级不会对集群中已有数据造成任何影响。
- 在集群规格变更期间，PolarDB服务会出现几秒钟的闪断且部分操作不能执行的状况，建议您在业务低谷期执行变更。闪断后需在应用端重新连接。
- PolarDB集群变更配置期间，只读请求相比读写请求的滞后时间比正常运行状态的滞后时间可能更长。

操作步骤

1. 登录[PolarDB控制台](#)。
2. 在控制台左上角，选择集群所在地域。

3. 进入升降配向导页面。您可以按照如下两种方式操作：

- 单击目标集群操作栏中的**升降配**。

集群列表

集群ID/名称	状态	兼容性	节点数	主节点配置	已使用数据	付费类型	标签	操作
pc-xxxxxx	运行中	100% 兼容 MySQL 5.6	2	4核 16GB	2.58 GB	按小时付费 2020年6月24日 14:42:29 创建		升降配 新增节点

- 单击目标集群ID，在**基本信息**页面下方，单击**升降配**。

基本信息

配置与管理

账号管理

数据库管理

备份恢复

参数配置

诊断与优化

集群总览

性能监控

问题分析

慢 SQL

日志与审计

SQL 洞察

集群地址【推荐】 ① 新增自定义地址

- 默认地址 (pe-xxxxxx) 修改配置

读写模式 可读可写 (自动读写分离)

私网 pc-xxxxxx 修改

公网 申请

> 节点配置

> 高级配置

节点信息

新增节点 **升降配**

节点名称	可用区	状态	当前角色	规格	最大IOPS	操作
pi-xxxxxx	杭州 可用区G	运行中	主节点	2核 4GB	8000	重启
pi-xxxxxx	杭州 可用区G	运行中	只读节点	2核 4GB	8000	重启

4. 勾选升级配置或降级配置，单击确定。

升降配向导（包年包月）

您当前的付费方式为 **包年包月**，支持以下配置变更方案：

☒ 升级配置

支持您在当前生命周期内立即升级POLARDB的规格配置，预计10分钟生效，过程中每个连接地址都会有不超过30秒的连接闪断，请确保应用具备重连机制。参考文档：[变更配置](#)

☐ 临时升级配置

支持您临时升级POLARDB的规格配置，应对短时间（一般小于7天）的业务高峰期。临时升级只需要支付升级期间的费用，升级前支付（预付费）。临时升级期间暂不支持添加节点，请先扩容节点，再操作临时升级。临时升级期间也不支持普通升降级或增删节点，因此建议您尽量一次性升级到最高的配置，避免重复升级。临时升级和到期还原时，会引起连接闪断，请确保应用具备重连机制。详细功能和计费介绍，请参考文档：[临时升配](#)

☐ 降级配置

支持您在当前生命周期内立即降低POLARDB的规格配置，预计10分钟生效，过程中每个连接地址都会有不超过30秒的连接闪断，请确保应用具备重连机制。参考文档：[变更配置](#)、[降配退费规则](#)

确定

取消



说明：

仅包年包月集群支持**临时升级配置**，详情请参见[临时升配](#)。

5. 选择所需的规格。



说明：

同一集群中，所有节点的规格总是保持一致。

6. 勾选服务协议，单击去支付并完成支付。



说明：

规格变更预计需要10分钟生效。

相关API

API	描述
#unique_181	变更PolarDB集群节点规格。

10.2 临时升配

PolarDB的包年包月集群支持临时升配，可以帮助您轻松应对短时间的业务高峰期。

前提条件

- 集群为包年包月集群。
- 集群没有尚未生效的续费变配订单。
- 集群没有尚未生效的临时升配订单。

背景介绍

临时升配是指临时升级规格，提升整体性能。到达指定的还原时间后，集群的规格会自动还原到临时升配前的状态。



说明：

不支持临时降级，如需降级请参见[变更配置](#)。

注意事项

- 还原过程可能会出现闪断，请确保应用程序具备重连机制。
- 还原时间不能晚于集群到期时间的前1天。例如集群1月10日到期，则临时升配的还原时间最多为1月9日。
- 临时升配的最短时间为1小时，由于设置还原时间后无法修改，建议升配时间最长不超过14天。
- 临时升配期间不支持普通的[变更配置](#)。
- 临时升配后如果性能不够，在还原时间到达之前最多可以再进行1次升配，此次设置的**还原时间**不能早于第1次。

计费

临时升配的价格是新老配置差价的1.5倍。计算公式如下：

临时升配N天，费用 = (新规格包月价格 - 老规格包月价格) / 30 x 1.5 x N。

操作步骤

1. 登录[PolarDB控制台](#)。
2. 在控制台左上角，选择集群所在地域。
3. 在[集群列表页](#)，找到目标集群。

4. 您可以选择如下两种方式中的任意一种进入**升降配向导（包年包月）**页面：

- 单击目标集群操作栏中的**升降配**。



- a. 单击目标集群ID，进入目标集群**基本信息**页。
- b. 在**数据库节点**区域，单击**升降配置**。



5. 在**升降配向导（包年包月）**页面，勾选**临时升级配置**，单击**确定**。



说明：

仅包年包月集群支持**临时升级配置**。

6. 在弹出的对话框中，设置如下参数。

配置变更

可选择在一定时间内提升实例的配置，并在到期后自动恢复

节点

pi-xxxxxx

8核 32GB (独享)

所有 PolarDB 节点规格均为独享型配置，性能稳定可靠，详见 [规格与定价](#)。

pi-xxxxxx

8核 32GB (独享)

所有 PolarDB 节点规格均为独享型配置，性能稳定可靠，详见 [规格与定价](#)。

[撤销所有规格变更](#)

短时升级

还原时间

2020-06-28 17 时

重要提醒：还原时数据库连接可能会闪断，请选择在业务低谷期还原配置，避免对业务产生影响。最短升级间隔为1小时，最长建议不超过14天。参考文档

参数	说明
节点	为当前节点选择升级后的目标节点规格。

参数	说明
还原时间	选择短时升配的到期还原时间。  说明： - 临时升配后如果性能不够，在还原时间到达之前最多可以再进行一次升配，此次设置的还原时间不能早于第1次。 - 临时升配的最短时间为1小时，由于设置还原时间后无法修改，建议升配时间为14天以内。 - 还原时间不能晚于集群到期时间的前一天。

7. 勾选服务协议，单击**去支付**完成支付。

8. 在**支付**页面，确认待支付订单，单击**订购**。

10.3 增加或删除节点

创建PolarDB集群后，您可以手动增加或删除只读节点。

背景信息

集群最多包含15个只读节点，最少一个只读节点（用于保障集群的高可用）。同一集群中，所有节点的规格总是保持一致。

节点费用

增加节点时的计费方式如下：

- 如果集群为包年包月（预付费），则增加的节点也是包年包月。
- 如果集群为按小时付费（后付费），则增加的节点也是按小时付费。



说明：

- 包年包月和按小时付费的只读节点都可以随时释放，释放后会**退款或停止计费**。
- 增加节点仅收取节点规格的费用（详情请参见[#unique_67](#)），存储费用仍然按实际使用量收费，与节点数量无关。

节点数量对性能的影响

详情请参见[性能测试（OLTP）](#)。

注意事项

- 仅当集群没有正在进行的配置变更时，才可以增加或删除只读节点。
- 为避免操作失误，每次操作只能增加或删除一个只读节点，增加或删除多个只读节点请多次操作。

- 增加或删除节点需要5分钟左右生效。

增加只读节点



说明：

新增只读节点之后新建的读写分离连接会转发请求到该只读节点。新增只读节点之前建立的读写分离连接不会转发请求到新增的只读节点，需要断开该连接并重新建立连接，例如，重启应用。

1. 登录PolarDB控制台。
2. 在控制台左上角，选择集群所在地域。
3. 进入**增删节点向导**页面。您可以按照如下两种方式操作：
 - 找到目标集群，在**操作**列单击**增删节点**。

集群列表							
创建新集群		集群ID	请输入	Q			
				刷新			
集群名称	状态	兼容数据库	节点数	主节点配置	已使用数据	付费类型	操作
pc-xxxxxx	运行中	MySQL 5.6	2	2核 4GB	2.83 GB	包年包月 2019年8月15日 00:00:00 到期	升降配 增删节点

- 找到目标集群，单击集群ID，在基本信息页面下方单击**增删节点**。

集群地址【推荐】 ① 新增自定义地址							
默认地址 (pe-xxxxxx) 修改配置							
读写模式 可读可写 (自动读写分离)							
私网 pc-xxxxxx 修改							
公网 申请							
> 节点配置							
> 高级配置							
节点信息							
增删节点 升降配							
节点名称	可用区	状态	当前角色	规格	最大IOPS	操作	
pi-xxxxxx	杭州 可用区G	运行中	主节点	2核 4GB	8000	重启	
pi-xxxxxx	杭州 可用区G	运行中	只读节点	2核 4GB	8000	重启	

4. 勾选增加节点并单击确定。

增删节点向导

您当前的付费方式为 **包年包月**, 支持以下配置变更方案:

☒ 增加节点

支持您在当前生命周期内立即增加POLARDB的数据库计算节点, 增加一个节点大概需要5分钟, 整个过程对数据库无任何影响。使用默认集群地址可自动识别新节点, 自动分流到新节点, 达到负载均衡, 不需要修改应用配置。参考文档: [增加节点](#), [包年包月集群增加节点的计费规则](#)

☐ 删除节点

支持您在当前生命周期内立即删除POLARDB的数据库计算节点, 该节点上的连接会发生闪断, 其他节点不受影响。使用集群地址可自动屏蔽失效节点, 不需要修改应用配置。参考文档: [删除节点](#), [包年包月集群删除节点的退费规则](#)

确定

取消

5. 单击 增加一个只读节点, 勾选服务协议, 单击去支付并完成支付。

删除只读节点

1. 登录[PolarDB控制台](#)。
2. 在控制台左上角, 选择集群所在地域。

3. 进入增删节点向导页面。您可以按照如下两种方式操作：

- 找到目标集群，在操作列单击**增删节点**。

集群列表							POLARDB简介
创建新集群	集群ID	请输入	Q	刷新	下载		
集群名称	状态	兼容数据库	节点数	主节点配置	已使用数据	付费类型	操作
pc-xxxxxx	运行中	MySQL 5.6	2	2核 4GB	2.83 GB	包年包月 2019年8月15日 00:00:00 到期	升降配 增删节点

- 找到目标集群，单击集群ID，在基本信息页面下方单击**增删节点**。

基本信息	集群地址【推荐】 ① 新增自定义地址
配置与管理	- 默认地址 (pe-xxxxxx) 修改配置 - 读写模式 可读写 (自动读写分离) - 私网 pc-xxxxxx 修改 - 公网 申请 > 节点配置 > 高级配置
账号管理	
数据库管理	
备份恢复	
参数配置	
诊断与优化	
集群总览	
性能监控	
问题分析	
慢 SQL	
日志与审计	
SQL 洞察	

节点信息						
增删节点	升降配					
节点名称	可用区	状态	当前角色	规格	最大IOPS	操作
pi-xxxxxx	杭州 可用区G	运行中	主节点	2核 4GB	8000	重启
pi-xxxxxx	杭州 可用区G	运行中	只读节点	2核 4GB	8000	重启

4. 勾选删除节点并单击确定。

增删节点向导

×

您当前的付费方式为 **包年包月**，支持以下配置变更方案：

☐ 增加节点

支持您在当前生命周期内立即增加POLARDB的数据库计算节点，增加一个节点大概需要5分钟，整个过程对数据库无任何影响。使用默认集群地址可自动识别新节点，自动分流到新节点，达到负载均衡，不需要修改应用配置。参考文档：[增加节点](#)，[包年包月集群增加节点的计费规则](#)

☒ 删除节点

支持您在当前生命周期内立即删除POLARDB的数据库计算节点，该节点上的连接会发生闪断，其他节点不受影响。使用集群地址可自动屏蔽失效节点，不需要修改应用配置。参考文档：[删除节点](#)，[包年包月集群删除节点的退费规则](#)

确定

取消

5. 单击想要删除的节点后面的 ，并在弹出对话框中单击**确定**。

**说明：**

集群中必须保留至少一个只读节点，以保障集群的高可用。

6. 勾选服务协议，单击**确认**。

相关API

API	描述
#unique_183	增加PolarDB集群节点
#unique_181	变更PolarDB集群节点规格
#unique_184	重启PolarDB集群节点
#unique_185	删除PolarDB集群节点

11 数据安全/加密

11.1 设置集群白名单

创建PolarDB MySQL数据库集群后，您需要设置PolarDB集群的IP白名单，并创建集群的初始账号，才能连接和使用该集群。

背景信息

设置集群白名单包括两种操作：

- 设置IP白名单

添加IP地址，允许这些IP地址访问该集群。默认的IP白名单只包含默认IP地址127.0.0.1，表示任何设备均无法访问该集群。只有已添加到IP白名单中的IP地址才可以访问该集群。

- 设置安全组

ECS安全组是一种虚拟防火墙，用于控制安全组中的ECS实例的出入流量。在PolarDB集群白名单中添加ECS安全组后，该安全组中的ECS实例就可以访问PolarDB集群。

关于ECS安全组的更多信息，请参见[创建安全组](#)。

注意事项

- 默认情况下，IP白名单只包含IP地址127.0.0.1，表示任何IP地址均无法访问该数据库集群。
- 若将IP白名单设置为%或者0.0.0.0/0，表示允许任何IP地址访问数据库集群。该设置将极大降低数据库的安全性，如非必要请勿使用。
- PolarDB暂不支持自动获取VPC中的ECS内网IP以供您选择，请手动填写需要访问PolarDB的ECS内网IP。
- 您可以同时设置IP白名单和ECS安全组。IP白名单中的IP地址和安全组中的ECS实例都可以访问该PolarDB集群。
- 目前仅PolarDB MySQL支持设置安全组，最多可以添加3个安全组。
- 目前支持创建50个IP白名单，所有IP白名单累积支持添加1000个IP地址或地址段。

设置白名单

1. 登录[PolarDB控制台](#)。
2. 在页面左上角，选择实例所在地域。
3. 单击目标集群ID，进入[基本信息](#)页面。
4. 单击[配置与管理 > 集群白名单](#)。

5. 在**集群白名单**页面，可以**配置**已有白名单或**新增IP白名单分组**。

新增IP白名单		选择安全组	
类型	名称	内容	操作
IP列表	default	127.0.0.1	配置 删除
IP列表		10.0.0.1...	配置 删除
IP列表		11.0.0.2...	配置 删除
安全组		sg-bp1rs	配置 删除
安全组		sg-bp1h	配置 删除

- 单击**操作**栏中的**配置**，配置该IP白名单。
- 单击**新增IP白名单分组**，可以新增IP白名单。

6. 填写白名单，单击**提交**。

- 如果您的ECS服务器需要访问PolarDB，可在**ECS实例详情**页面**配置信息**区域，查看ECS服务器的IP地址，然后填写到白名单中。



说明：

如果ECS与PolarDB位于同一地域（例如，华东1），填写ECS的私网IP地址；如果ECS与PolarDB位于不同的地域，填写ECS的公网IP地址，或者将ECS迁移到PolarDB所在地域后填写ECS私网IP地址。

- 如果您本地的服务器、电脑或其它云服务器需要访问PolarDB，请将其IP地址添加到白名单中。

设置安全组

1. 登录**PolarDB控制台**。
2. 在页面左上角，选择实例所在地域。
3. 单击目标集群ID，进入**基本信息**页面。
4. 单击**配置与管理 > 集群白名单**。

5. 在集群白名单页面，可以配置已有安全组或选择安全组。

新增IP白名单		选择安全组		
类型	名称	内容	操作	
IP列表	default	127.0.0.1	配置	删除
IP列表	sg-bp1xxxxxxxxxxxx	10.0.0.0/24	配置	删除
IP列表	sg-bp1xxxxxxxxxxxx	11.0.0.0/24	配置	删除
安全组	sg-bp1xxxxxxxxxxxx	sg-bp1xxxxxxxxxxxx3rs	配置	删除
安全组	sg-bp1xxxxxxxxxxxx	sg-bp1xxxxxxxxxxxx1h	配置	删除

- 单击操作栏中的**配置**，配置该安全组。
- 单击**选择安全组**，可以新增安全组。

6. 选择安全组，单击提交。

关于如何创建ECS安全组，请参见[创建安全组](#)。

下一步

设置白名单以及创建数据库账号后，您就可以连接数据库集群，对数据库进行操作。

- [创建数据库账号](#)
- [连接数据库集群](#)

常见问题

1. 如何指定服务器只能访问集群中的某个节点？

答：使用[自定义集群地址](#)访问集群，就只能访问集群中的特定节点。

2. IP白名单最多支持填写多少个IP地址？

答：IP白名单最多可以填写1000个IP地址或地址组。安全组关联的IP不受此限制。

3. 已添加ECS的IP地址到IP白名单中，但是还是无法访问。

答：

- 确认IP白名单是否正确。如果是通过内网地址访问，需添加ECS的私网IP地址。如果是通过公网地址进行访问，需添加ECS的公网IP地址。
- 确认网络类型是否一致。如果ECS实例的网络类型是经典网络，可参见[经典网络迁移到专有网络方案](#)将ECS实例迁移至PolarDB所在的专有网络。



说明：

如果该ECS 还要访问其他经典网络内网资源，请勿操作，因为迁移后会无法访问经典网络。

或者通过[Classlink](#)打通经典网络到专有网络的网络。

- c. 确认是否位于同一个VPC。如果不是，需要重新购买一个PolarDB，或者通过[云企业网](#)来打通两个VPC网络实现访问。

4. 什么原因导致公网连接失败？

答：

- a. 如果是ECS通过公网地址进行访问，请确认添加的是ECS的公网IP地址，而不是私网IP地址。
- b. IP白名单设置为0.0.0.0/0，然后尝试访问，如果能成功访问，表示IP白名单中之前填写的公网地址错误。请参见[查看连接地址](#)确定真正的公网地址。

5. 如何实现内网连接？

答：ECS和PolarDB内网访问需要满足以下条件：

- 相同地域。
- 相同网络类型，如果是VPC，需要在相同VPC下。
- ECS内网IP在PolarDB集群白名单中。

6. 如何限制某个用户只能从特定的IP地址访问PolarDB？

答：可以创建高权限账号，然后使用高权限账号对普通账号限定访问IP。

```
1  
2  
3 CREATE USER 'alitest'@'192.168.1.101' ;  
4  
5  
6 select * from mysql.user where user='alitest';|
```

相关API

API	描述
#unique_187	查看允许访问数据库集群的IP名单。
#unique_188	修改允许访问数据库集群的IP名单。

11.2 设置SSL加密

为了提高链路安全性，您可以启用SSL（Secure Sockets Layer）加密，并安装SSL CA证书到需要的应用服务。SSL在传输层对网络连接进行加密，能提升通信数据的安全性和完整性，但会同时增加网络连接响应时间。

注意事项

- SSL的证书有效期为1年，请及时[更新证书有效期](#)并重新下载和配置CA证书，否则使用加密连接的客户端程序将无法连接。
- 由于SSL加密的固有缺陷，启用SSL加密会显著增加CPU使用率，建议您仅在外网链路有加密需求的时候启用SSL加密。内网链路相对较安全，一般无需对链路加密。
- 开启SSL加密的PolarDB连接地址需要少于64个字符，修改连接地址请参见[修改和释放集群地址](#)。
- 关闭SSL加密会重启集群，请谨慎操作。

开启SSL加密和下载证书

1. 登录[PolarDB管理控制台](#)。
2. 在页面左上角，选择集群所在地域。



3. 找到目标集群，单击集群ID。
4. 在左侧菜单栏中单击[配置与管理 > 安全管理](#)。

5. 在**SSL配置**页签，单击**SSL状态**右侧滑块，开启SSL加密。



6. 在**设置SSL**对话框中，选择需要开通SSL加密的链路，单击**确定**。



说明：

您可以根据需要，选择加密内网链路或者外网链路，但只可以加密一条链路。

设置SSL

选择受保护地址

☒ (公网)

☐ b.rds.aliyuncs.com (私网)

1. 修改受保护地址后，证书会自动更新，同时重启您的数据库实例。

2. 全长超过64个字符的连接地址不能启用SSL。

确定

取消

7. SSL状态变为**已开通**后，单击**下载证书**。

主地址

SSL状态

已开通

更新有效期

SSL受保护的地址

ardb.rds.aliyuncs.com

SSL证书有效期

2021年5月20日 14:05:07

SSL证书有效性

有效

设置SSL

下载证书

下载的文件为压缩包，包含如下三个文件：

- p7b文件：用于Windows系统中导入CA证书。
- pem文件：用于其他系统或应用中导入CA证书。
- jks文件：Java中的truststore证书存储文件，密码统一为apsaradb，用于Java程序中导入CA证书链。



说明：

在Java中使用JKS证书文件时，jdk7和jdk8需要修改默认的jdk安全配置，在连接PolarDB数据库的服务器的jre/lib/security/java.security文件中，修改如下两项配置：

```
jdk.tls.disabledAlgorithms=SSLv3, RC4, DH keySize < 224
```

```
jdk.certpath.disabledAlgorithms=MD2, RSA keySize < 1024
```

若不修改jdk安全配置，会报如下错误。其它类似报错，一般也都由Java安全配置导致。

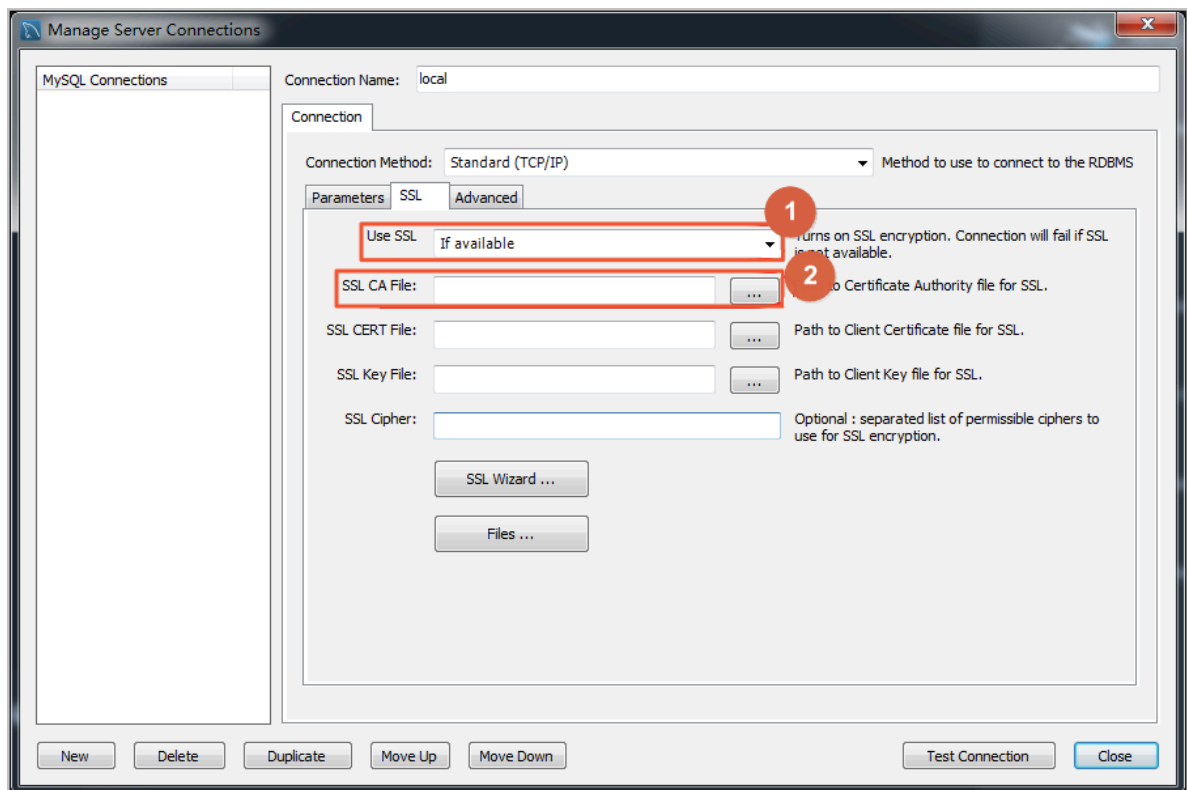
```
javax.net.ssl.SSLHandshakeException: DHPublicKey does not comply to algorithm constraints
```

配置SSL CA证书

开通SSL加密后，应用或者客户端连接PolarDB时需要配置SSL CA证书。本文以MySQL Workbench和Navicat为例，介绍SSL CA证书安装方法。其它应用或者客户端请参见对应产品的使用说明。

MySQL Workbench配置方法

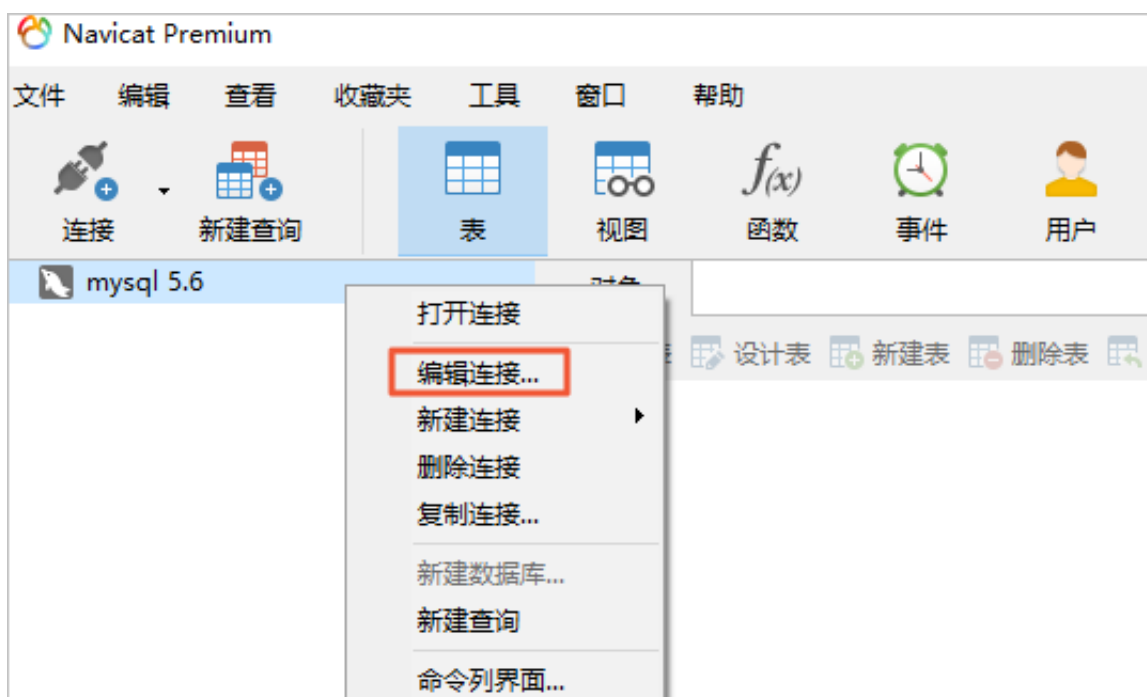
1. 打开MySQL Workbench。
2. 选择**Database > Manage Connections**。
3. 启用**Use SSL**，并导入SSL CA证书，如下图所示。



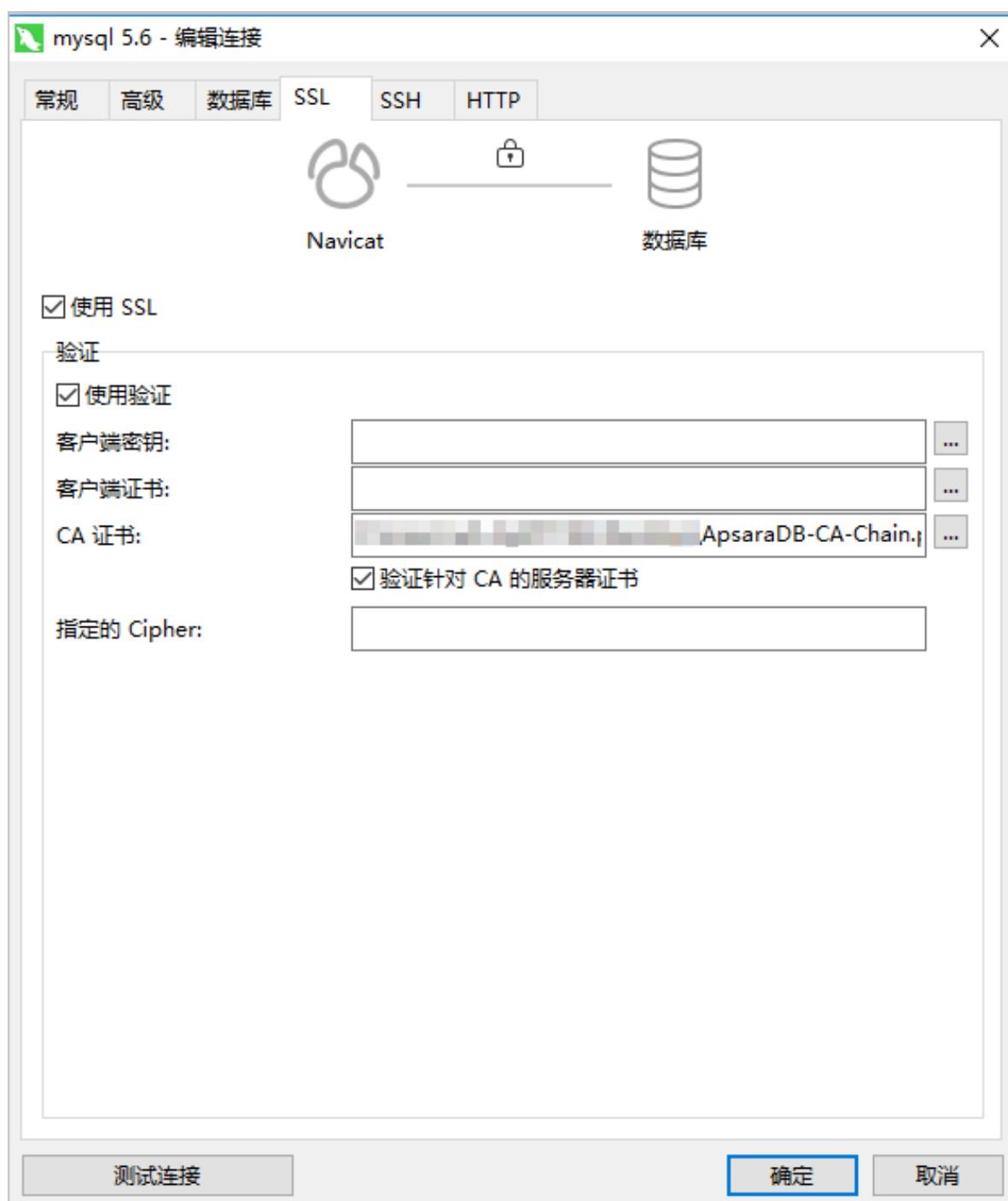
Navicat配置方法

1. 打开Navicat。

2. 在目标数据库上单击鼠标右键，选择**编辑连接**。



3. 选择SSL页签，选择.pem格式CA证书的路径。参照下图进行设置。



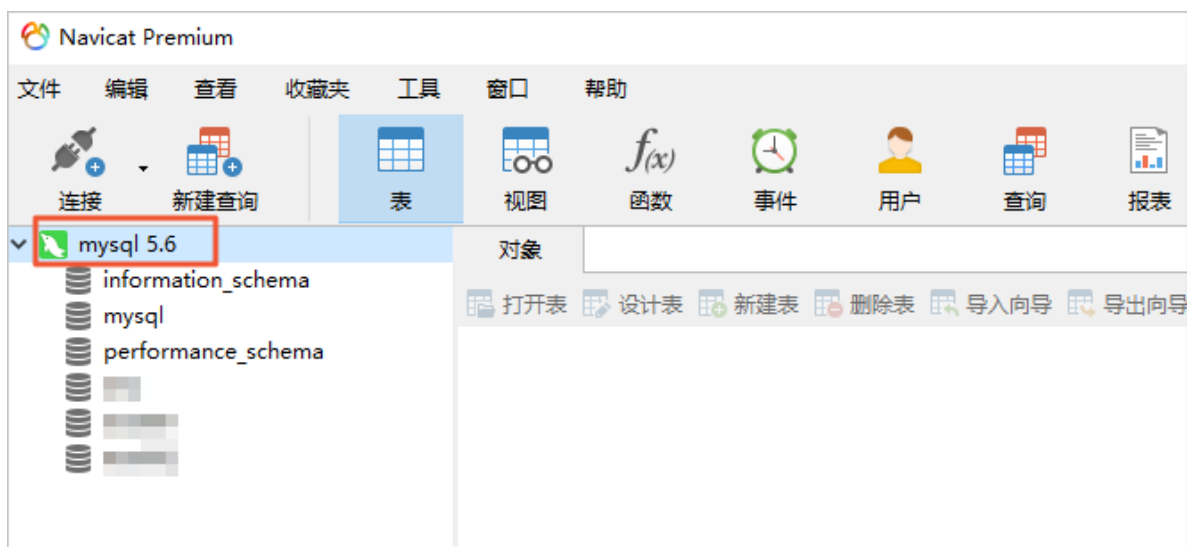
4. 单击**确定**。



说明：

如果报connection is being used错误，是由于之前的会话未断开，请关闭Navicat重新打开。

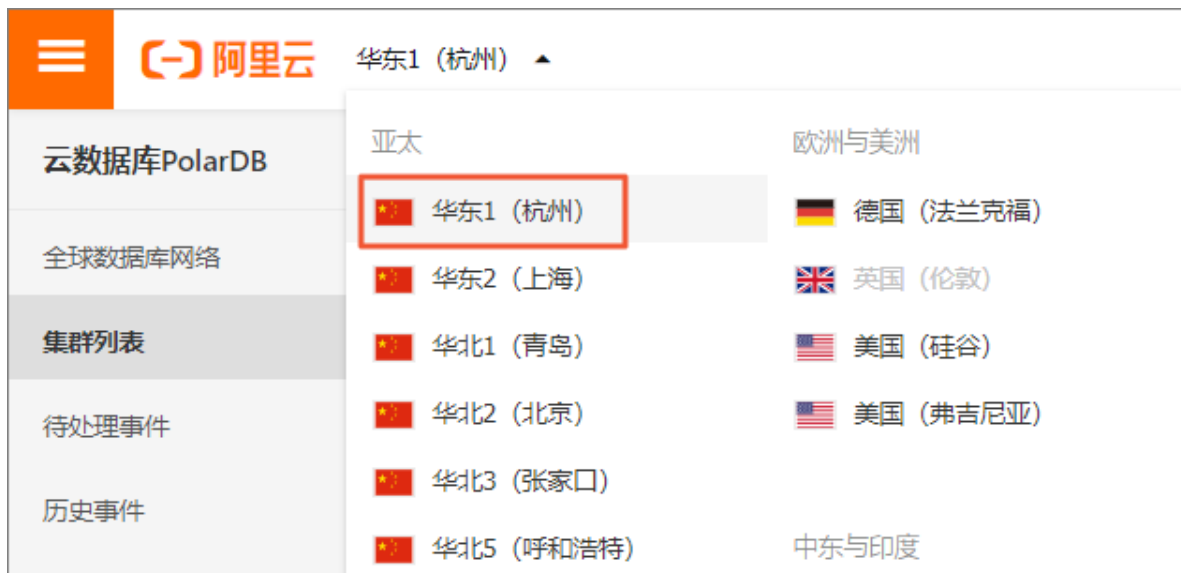
5. 双击目标数据库测试能否正常连接。



更新证书有效期

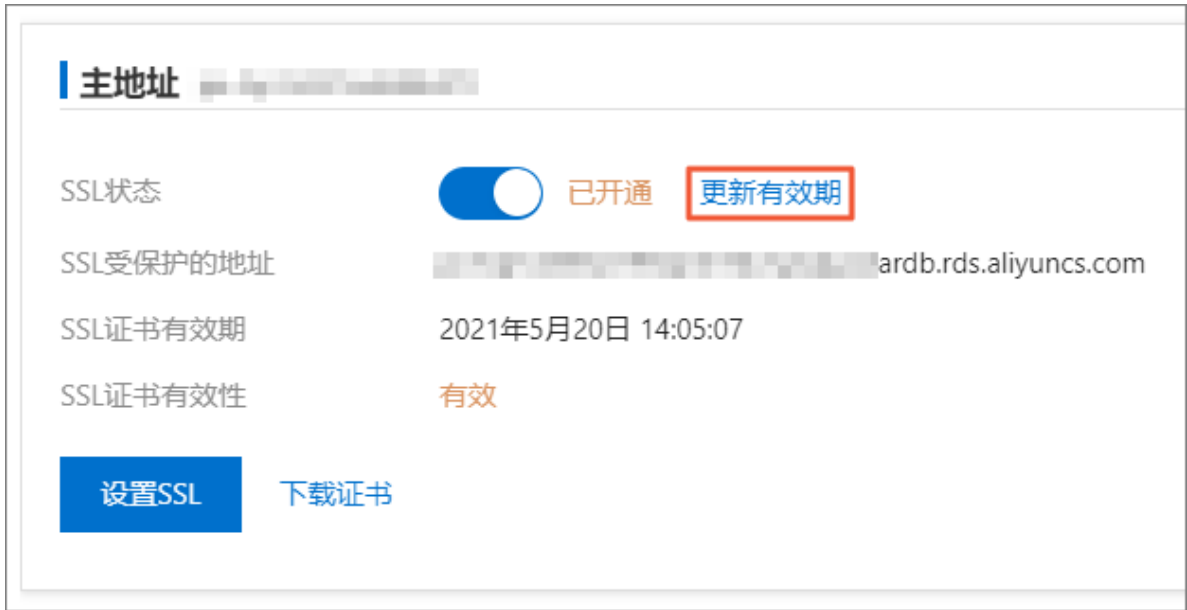
如果您修改了SSL连接地址或证书有效期即将到期，您需要更新证书有效期，以下内容将为您介绍如何更新证书有效期。

1. 登录[PolarDB管理控制台](#)。
2. 在页面左上角，选择集群所在地域。



3. 找到目标集群，单击集群ID。
4. 在左侧菜单栏中单击**配置与管理 > 安全管理**。

5. 在SSL配置页签中单击**更新有效期**。



6. 在**设置SSL**对话框单击**确定**。



说明：

更新有效期操作将会重启集群，重启前请做好业务安排，谨慎操作。

7. 更新有效期后，重新下载和配置证书。



说明：

- 下载证书请参见[开启SSL加密和下载证书](#)第七步。
- 配置证书请参见[配置SSL CA证书](#)。

关闭SSL加密

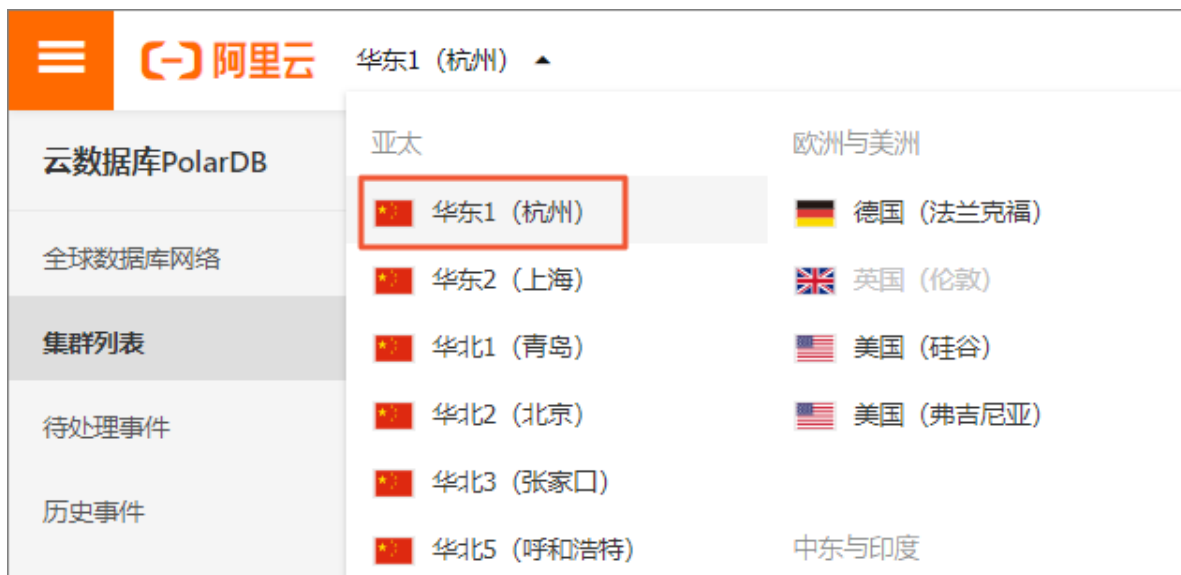


说明：

- 关闭SSL加密会重启集群，建议您在业务低峰期操作。
- SSL加密关闭后，数据库访问性能会有一定程度提升，但安全性上有削弱，故非安全环境下不建议关闭SSL加密。

1. 登录[PolarDB管理控制台](#)。

2. 在页面左上角，选择集群所在地域。



3. 找到目标集群，单击集群ID。
4. 在左侧菜单栏中单击**配置与管理 > 安全管理**。
5. 在**SSL配置**页签中单击**SSL状态**右侧滑块，关闭SSL加密。



6. 在**设置SSL**对话框，单击**确定**。

常见问题

Q：SSL证书到期后不更新会有什么影响？会影响实例运行或数据安全吗？

A：SSL证书到期后不更新，会导致使用加密连接的客户端程序无法正常连接实例，不会影响实例运行或数据安全。

相关API

API	描述
#unique_189	调用DescribeDBClusterSSL接口查询PolarDB集群SSL设置。
#unique_190	调用ModifyDBClusterSSL接口设置PolarDB集群SSL加密的开通、关闭或更新CA证书。

11.3 设置透明数据加密TDE

透明数据加密TDE（Transparent Data Encryption）可对数据文件执行实时I/O加密和解密，数据在写入磁盘之前进行加密，从磁盘读入内存时进行解密。TDE不会增加数据文件的大小，开发人员无需更改任何应用程序，即可使用TDE功能。

前提条件

- 集群版本为PolarDB MySQL 8.0。
- 已开通KMS。如果您未开通KMS，可在开通TDE过程中根据引导开通KMS。更多关于KMS相关信息请参见[密钥管理服务](#)。

背景信息

加密使用的密钥由密钥管理服务（KMS）产生和管理，PolarDB不提供加密所需的密钥和证书。部分可用区不仅可以使使用阿里云自动生成的密钥，也可以使用自带的密钥材料生成数据密钥，然后授权PolarDB使用。

注意事项

- TDE开通后无法关闭。
- 开通TDE后，如果是IO bound场景，可能会对数据库性能产生一定影响。
- 启用TDE加密会导致PolarDB集群重启，请谨慎操作。
- 已加入全球数据库网络的集群无法开启TDE功能，已开启TDE功能的集群也无法加入全球数据库网络。
- 使用已有自定义密钥时，需要注意：
 - 禁用密钥、设置密钥删除计划或者删除密钥材料都会造成密钥不可用。
 - 撤销授权关系后，重启PolarDB集群会导致PolarDB集群不可用。
 - 需要使用主账号或者具有AliyunSTSAssumeRoleAccess权限的账号。

使用由阿里云自动生成的密钥

1. 登录[PolarDB控制台](#)。

2. 在页面左上角，选择集群所在地域。
3. 找到目标集群，单击集群ID。
4. 在左侧导航栏单击**配置与管理** > **安全管理**。
5. 在**TDE配置**页签，单击**TDE状态**右侧滑块。



6. 在**设置TDE**对话框中，选择**使用由阿里云自动生成的密钥**。



7. 单击**确定**，开通TDE。

使用已有自定义密钥

1. 登录PolarDB控制台。
2. 在页面左上角，选择集群所在地域。
3. 找到目标集群，单击集群ID。

4. 在TDE配置页签，单击TDE状态右侧滑块。

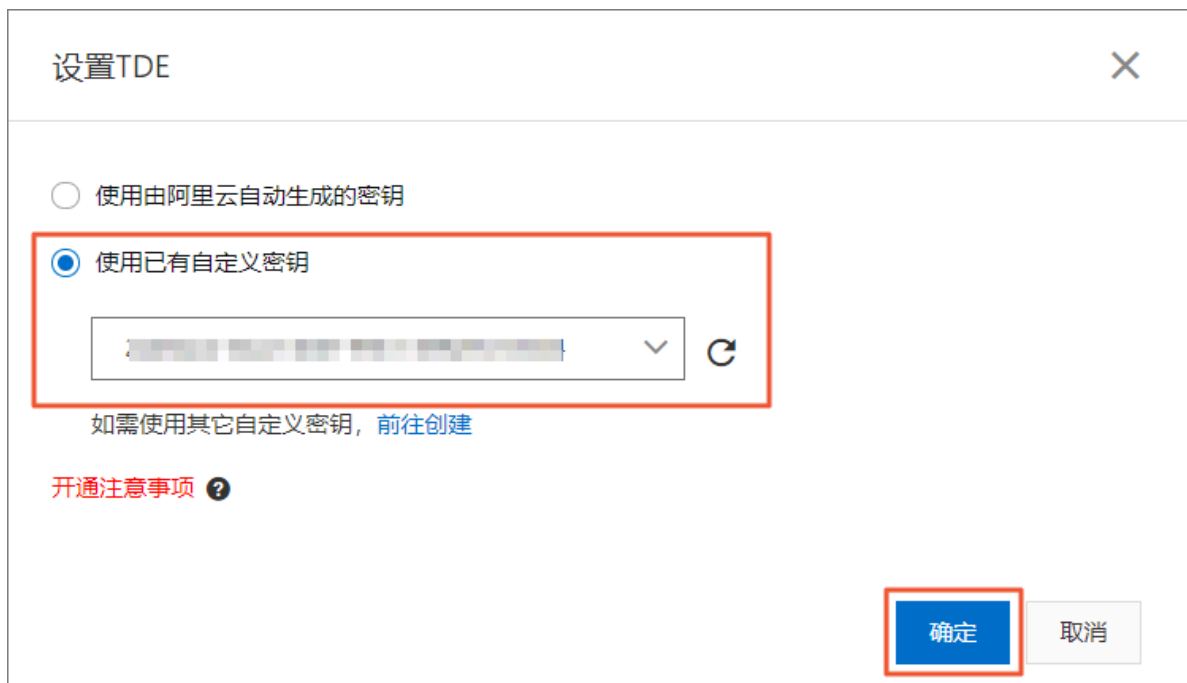


5. 在设置TDE对话框中，选择使用已有自定义密钥。



说明：

如果没有自定义密钥，需要单击[前往创建](#)，在密钥管理服务控制台创建密钥并导入自带的密钥材料。详情请参见[管理密钥](#)。



6. 单击确定，开通TDE。

加密操作

登录数据库，执行如下命令，对要加密的表进行加密。

```
alter table <tablename> encryption= 'Y';
```

解密操作

如果您要对TDE加密的表解密，请执行如下命令。

```
alter table <tablename> encryption= 'N';
```

常见问题

- Q：开启TDE后，常用数据库工具（Navicat等）还能正常使用吗？
A：可以正常使用。
- Q：加密后查看数据为什么还是明文的？
A：查询数据时会解密并读取到内存，所以是明文显示。开启TDE后存储的数据是加密的。

12 备份与恢复

12.1 备份数据

可靠的备份功能可以有效防止数据丢失，PolarDB MySQL支持周期性的自动备份以及即时生效的手动备份。在删除PolarDB MySQL集群时，您还可以选择保留备份数据。

特色功能

PolarDB支持长期保留备份文件，并且可以设置删除实例后继续保留备份文件，避免了误操作导致的数据丢失。

一级备份采用了ROW（Redirect-on-Write）快照的方式。每次保存快照并没有真正复制数据，当数据块有修改时系统会将其中一个历史版本的数据块保留给快照，同时生成新的数据块被原数据引用（Redirect）。因此无论数据库容量多少，都可以做到秒级备份。

日志备份通过实时并行上传Redo日志到OSS来达到备份的目的。通过一个完整的数据全量备份（快照）以及后续一段时间的Redo日志，就可以将PolarDB集群恢复到任意时间点（Point-In-Time Recovery，简称：PITR）。

PolarDB集群备份和恢复功能均采用多线程并行处理来提高效率。目前，一级备份（快照）进行恢复或克隆的速度为40分钟/TB；如果您需要**按时间点恢复**，则需要包含应用Redo日志的时间，应用Redo日志速度为20秒/GB至70秒/GB，整个恢复时间是快照恢复时间以及应用Redo日志时间之和。



警告：

如果您需要关闭备份功能，可以[提交工单](#)联系售后客服进行关闭，但您须自行承担关闭备份所带来的风险（如误操作导致的数据丢失等）。

产品定价

PolarDB备份和恢复功能均免费使用，但备份文件需要占用一定的存储空间，PolarDB会根据备份文件（数据和日志）的存储容量和保存时长收取一定费用。



说明：

PolarDB备份功能于2020年5月11日商业化，为了您可以更方便地了解备份功能以及备份文件的存储空间情况，在接下来的一周时间，PolarDB备份不会产生任何费用。详细信息请参见[#unique_192](#)。

表 12-1: 价格表

地域	一级备份	二级备份	日志备份
中国内地	0.003元/GB/小时	0.00021元/GB/小时	0.00021元/GB/小时
中国香港及海外	0.0042元/GB/小时	0.000294元/GB/小时	0.000294元/GB/小时

计费方式

备份类型	免费额度	计费方式
一级备份	数据库存储用量 x 50% 您可以在控制台基本信息页面查看数据库存储用量。	每小时费用 = (一级备份总大小 - 免费额度) x 每小时价格 - 一级备份小于免费额度时，一级备份不收取任何费用。 - 每小时单价请参见表 12-1: 价格表。 一级备份（快照）总大小如下图中①所示，而非②。  示例：一级备份（快照）总大小为700GB，数据库存储用量为1000GB，那么每小时费用为0.6元。 计算公式：[700GB - (1000GB x 50%)] x 0.003元 = 0.6元
二级备份	无	每小时费用 = 二级备份总大小 x 每小时价格 示例：二级备份总大小为1000GB，那么每小时费用为0.21元。 计算公式：1000GB x 0.00021元 = 0.21元
日志备份	100GB	每小时费用 = (日志备份总大小 - 100GB) x 每小时价格 示例：日志备份总大小为1000GB，那么每小时费用为0.189元。 计算公式：(1000GB - 100GB) x 0.00021元 = 0.189元

备份类型

备份类型	说明
一级备份（数据备份）	直接存储在分布式存储集群上，备份和恢复速度快，但保存成本高。 最短保留时间为7天，最长保留时间为14天。 一级备份（快照）总大小如下图所示。 
二级备份（数据备份）	指一级备份压缩后保存在其它离线存储介质上的备份数据，使用二级备份恢复数据的速度较慢，但其保存成本较低。 最短保留时间为30天，最长保留时间为7300天，也可以开启删除集群前永久保留功能。  说明： - 二级备份功能默认关闭。 - 开启二级备份后，一级备份超出您设置的保存时间后将自动转存为二级备份，转存速度在150MB/s左右。 - 如果转存二级备份较久，到下次转存二级备份时仍没有完成，将跳过下次转存。例如每天凌晨1点自动备份1次，一级备份A在凌晨1点过期开始转存为二级备份，由于备份文件较大转存时间较长，第二天凌晨1点该转存任务仍未完成，一级备份B在第二天凌晨1点到期后将直接删除而不会转存为二级备份。
日志备份	日志备份是把数据库的Redo日志保存下来，用于按时间点恢复数据，保证最近一段时间的数据安全性，避免误操作导致的数据丢失。 日志备份最短保留时间为7天，最长保留时间为7300天，您也可以通过开启删除集群前永久保留功能永久保存。

备份方式

备份方式	说明
系统备份（自动）	- 自动备份默认为每天1次，您可以设置自动执行备份的时间段和周期。具体请参见设置自动备份。 - 备份文件不可删除。  说明： 出于安全考虑，自动备份的频率为每周至少2次。
主动备份	- 您可以随时发起主动备份。每个集群最多可以有3个主动创建的备份。具体请参见手动创建备份。 - 备份文件可删除。

设置自动备份

1. 登录[PolarDB控制台](#)。
2. 在控制台左上角，选择集群所在地域。
3. 找到目标集群，单击集群ID。
4. 在左侧导航栏中，选择**配置与管理 > 备份恢复**。
5. 单击**备份设置**。



6. 在弹出的窗口中，设置如下参数。

备份设置

数据备份方式: 快照备份

* 数据备份周期: ☒ 星期一 ☒ 星期二 ☒ 星期三 ☒ 星期四
☒ 星期五 ☒ 星期六 ☒ 星期日

* 备份开始时间: 15:00 - 16:00

一级备份保留 ? 天 * 若要保留更久，请打开二级备份

二级备份开关 ? ☐ 关闭 ☒ 开启

二级备份保留: 天 ☐ 删除集群前永久保留

日志备份保留 ? 天 ☐ 删除集群前永久保留

删除集群时 ? ☐ 永久保留全部备份
☐ 永久保留最后一个备份 (删除前自动备份)
☒ 立即删除该集群的所有备份

参数	说明
数据备份方式	默认为快照备份。

参数	说明
数据备份周期	设置数据自动备份的周期。  说明： 出于安全考虑，自动备份的频率为每周至少2次。
备份开始时间	设置自动备份开始的时间。
一级备份保留	设置一级备份保留时间。  说明： 一级备份最短保留时间为7天，最长保留时间为14天。
二级备份开关	开启或关闭二级备份。  说明： 二级备份默认为关闭状态。
二级备份保留	设置二级备份的保留时间。  说明： - 二级备份最短保留时间为30天，最长保留天数为7300天。 - 如果您需要永久保存二级备份，可以选中删除集群前永久保留，选中后将无法设置保留天数。
日志备份保留	设置日志备份的保留时间。  说明： - 日志备份最短保留时间为7天，最长保留天数为7300天。 - 如果您需要永久保存日志备份，可以选中删除集群前永久保留，选中后将无法设置保留天数。

参数	说明
删除集群时	设置删除集群时的备份保留策略。 永久保留全部备份：删除集群时保留所有备份。 永久保留最后一个备份（删除前自动备份）：删除集群时保留最后一个备份。 立即删除该集群的所有备份：删除集群时不保留任何备份。  说明： - 如果您选择了永久保留全部备份或永久保留最后一个备份（删除前自动备份）策略，删除PolarDB集群时，系统会主动发起1次备份，为您保存删除前的所有数据。 - 删除集群后，一级备份将自动转为二级备份，您可以在集群回收站中查看所有保存的备份，更多内容请参见集群回收站。

7. 完成备份设置后，单击**确定**。

手动创建备份

1. 登录[PolarDB控制台](#)。
2. 在控制台左上角，选择集群所在地域。
3. 找到目标集群，单击集群ID。
4. 在左侧导航栏中，选择**配置与管理 > 备份恢复**。
5. 在**备份列表**页签，单击**创建备份**。



6. 在弹出的**创建备份**对话框中，单击**确定**。



说明：

每个集群最多可以有3个手动创建的备份。

恢复数据

请参见[恢复数据](#)。

常见问题

- Q：一级备份大小是否为所有快照备份之和？

A：一级备份大小不是所有快照备份之和，即并非下图中的②，而是①所示的大小。

备份列表

日志列表

【Binlog备份】数据库备份DBS可满足异地备份、长期归档、备份可读等需求。[查看更多介绍](#)

创建备份

按时间点恢复

备份SQL文件及Binlog

备份设置

2020年3月12日

-

2020年5月12日

📅

一级备份(快照)总大小  1.30 GB

2

备份集ID	备份开始时间	备份结束时间	状态	一致性快照时间点 	备份方法	备份类型	大小 	存储位置
	2020年5月12日 06:56:32	2020年5月12日 06:56:37	备份完成	2020年5月12日 06:56:34	快照备份	全量备份	5.74 GB	一级备份
	2020年5月11日 06:56:24	2020年5月11日 06:56:40	备份完成	2020年5月11日 06:56:27	快照备份	全量备份	5.64 GB	一级备份
	2020年5月10日 06:56:25	2020年5月10日 06:56:30	备份完成	2020年5月10日 06:56:27	快照备份	全量备份	5.54 GB	一级备份

- Q：为什么一级备份的总大小比单个备份还要小？

A：PolarDB的一级备份有两个容量数据，一个是每个备份的逻辑大小，一个是全部备份的物理大小。PolarDB的一级备份采用快照链的机制，相同的数据块只会记录一份，因此总物理大小要小于逻辑大小，有时候甚至会小于单个备份逻辑大小。

- Q：PolarDB备份有哪些费用？

A：一级备份、二级备份以及日志备份的存储空间费用。其中一级备份和日志备份默认开启，并赠送一定的免费空间。二级备份默认关闭。

- Q：一级备份的费用怎么算？

A：每小时费用 = [一级备份总大小 - (数据库存储用量 × 50%)] × 每小时价格。例如，PolarDB数据库的一级备份总大小为700G，数据库存储用量为1000GB，那么每小时费用为 (700G-500G) * 0.003元/GB = 0.6元。

- Q：存储包能否抵扣备份空间的费用？

A：目前存储包只能抵扣数据存储占用的空间，不能抵扣备份占用的空间。

相关API

API	描述
#unique_195	创建PolarDB集群全量快照备份。
#unique_196	查询PolarDB集群备份信息。
#unique_197	删除PolarDB集群备份。
#unique_198	查询PolarDB集群自动备份策略。

API	描述
#unique_199	修改PolarDB集群自动备份策略。

12.2 集群回收站

集群回收站用于保存已释放的PolarDB集群，您可以将回收站中的集群恢复到新集群或者删除某个备份集。本文以PolarDB MySQL集群为例为您介绍如何使用集群回收站。

注意事项

- 集群回收站中每个集群至少需要存在一个备份集，如果删除了某个集群下所有的备份集，该集群将无法恢复。
- 释放集群后，集群回收站中所有一级备份都会依次异步转存为二级备份，速度为150MB/s左右。更多关于备份的介绍请参见[备份数据](#)。

产品定价



说明：

PolarDB备份功能于2020年5月11日商业化，为了让您熟悉集群回收站功能以及资费情况，在接下来的一周时间，集群回收站不会产生任何费用。

集群回收站中一级备份不收取任何费用，转存为二级备份后才会进行收费。

地域	费用（元/GB/小时）
中国内地	0.00021元
中国香港及海外	0.000294元

恢复数据到新集群

1. 登录[PolarDB控制台](#)。
2. 在控制台左上角，选择集群所在地域。
3. 在左侧导航栏中单击[集群回收站](#)。
4. 找到目标集群，单击右侧操作栏中的[恢复数据到新集群](#)。

集群回收站							
集群ID	▼	请输入	Q				
集群ID/名称	地域	主节点配置	兼容性	创建时间	删除时间	状态	操作
+ polar-cluster-xxxxxx	华南1（深圳）	4核 16GB	100% 兼容 MySQL 5.6	2020年4月28日 17:01:58	2020年5月8日 15:20:18	● 已释放	恢复数据到新集群
+ polar-cluster-xxxxxx	华南1（深圳）	4核 16GB	兼容ORACLE语法	2020年4月26日 13:52:00	2020年4月28日 16:10:17	● 已释放	恢复数据到新集群

5. 选择包年包月或按量付费。

- **包年包月**：在创建集群时支付计算节点（一个主节点和一个只读节点）的费用，而存储空间会根据实际数据量按小时计费，并从账户中按小时扣除。如果您要长期使用该集群，**包年包月**方式更为经济，而且购买时长越长，折扣越多。
- **按量付费（按小时付费）**：无需预先支付费用，计算节点和存储空间（根据实际数据量）均按小时计费，并从账户中按小时扣除。如果您只需短期使用该集群，可以选择**按量付费**，用完即可释放，节省费用。

6. 在售卖页面，确认**已删除集群**信息无误后，选择需要恢复的**历史备份**。



说明：

- **历史备份**显示的备份时间为UTC时间，备份列表中显示的备份时间为当前系统时间，请确保所选择的历史备份无误。例如，当前备份列表显示的备份时间为2020年5月8日 10:00:00（北京时间），对应的**历史备份**为2020-05-08T02:00:00Z。
- 更多售卖页相关信息请参见[创建PolarDB MySQL集群](#)。

创建方式	默认创建	从 RDS 克隆	从 RDS 迁移	创建只读集群	从回收站恢复
从回收站恢复已经删除的数据库					
原版本	MySQL 5.6	MySQL 8.0	PostgreSQL 11	兼容Oracle语法	
已删除集群的版本					
已删除集群	选择要恢复的备份				
历史备份	2020-05-08T07:20:23Z				

7. 单击页面右侧的**立即购买**。

8. 在**确认订单**页面，确认订单信息，阅读和勾选**服务协议**，单击**去支付**。



说明：

恢复到新集群所需的时间和您的备份大小有关，备份越大恢复到新集群所需的时间越久。

删除备份

1. 登录[PolarDB控制台](#)。
2. 在控制台左上角，选择集群所在地域。
3. 在左侧导航栏中单击**集群回收站**。

4. 找到目标集群，单击左侧的  图标，展开备份集列表。

5. 找到需要删除的备份集，单击操作栏的删除备份。

集群ID/名称	地域	主节点配置	兼容性	创建时间	删除时间	状态	操作				
— 集群ID: [模糊]	华南1 (深圳)	4核 16GB	100% 兼容 MySQL 5.6	2020年4月28日 17:01:58	2020年5月8日 15:20:18	● 已释放	恢复数据到新集群				
2020年3月8日 - 2020年5月8日 											
备份集ID	备份开始时间	备份结束时间	状态	一致性快照时间点 	大小 	存储位置	是否有效	备份方法	备份类型	备份策略	操作
[模糊]	2020年5月8日 15:20:23	2020年5月8日 15:20:33	备份完成	2020年5月8日 15:20:25	2.30 GB	一级备份	有效	快照备份	全量备份	主动备份	删除备份
[模糊]	2020年5月8日 01:04:22	2020年5月8日 01:04:32	备份完成	2020年5月8日 01:04:25	2.30 GB	一级备份	有效	快照备份	全量备份	系统备份	删除备份
[模糊]	2020年5月7日 01:04:17	2020年5月7日 01:04:27	备份完成	2020年5月7日 01:04:20	2.30 GB	一级备份	有效	快照备份	全量备份	系统备份	删除备份
[模糊]	2020年5月6日 01:12:16	2020年5月6日 01:12:26	备份完成	2020年5月6日 01:12:20	2.30 GB	一级备份	有效	快照备份	全量备份	系统备份	删除备份
[模糊]	2020年5月5日 01:12:23	2020年5月5日 01:12:33	备份完成	2020年5月5日 01:12:25	2.30 GB	一级备份	有效	快照备份	全量备份	系统备份	删除备份

6. 在删除备份对话框中，单击确定。



警告：

如果删除了集群回收站中某个集群下所有的备份集，该集群将无法恢复，请谨慎处理。

相关文档

[释放集群](#)

12.3 恢复数据

本文为您介绍恢复PolarDB MySQL数据的流程以及不同恢复方式的操作步骤。

恢复数据的流程

恢复PolarDB MySQL数据的流程如下：

1. 将历史数据恢复到新集群中。您可以选择以下其中一种恢复方式：

- [按时间点恢复（恢复到任意时间点）](#)
- [按备份集（快照）恢复](#)

2. [登录到集群](#)，验证数据的准确性。

3. [将需要的数据迁移到原集群](#)。



说明：

恢复后的集群包含原集群的数据和账号信息，不包含原集群的参数设置。

按时间点恢复数据（恢复到任意时间点）

您可以将过去7天内任意一个时间点的数据恢复到新集群。

1. 登录[PolarDB控制台](#)。
2. 在控制台左上角，选择集群所在地域。
3. 单击目标集群ID。
4. 在左侧导航栏中，选择配置与管理 > 备份恢复。
5. 单击按时间点恢复，在弹出的对话框中单击确认。

<

集群 pc-bp10je14cx6j298sw ● 运行中

登录数据库 迁入

基本信息

配置与管理

账号管理

数据库管理

备份恢复

参数配置

诊断与优化

创建备份

按时间点恢复

备份设置

2018年10月20日 - 2018年12月20日

备份开始/结束时间	备份方法	备份类型	备份策略	状态	操作
2018-12-20 13:56:24	快照备份	全量备份	手动备份	0%	恢复备份 删除备份
2018-12-20 04:47:03 - 2018-12-20 04:47:23	快照备份	全量备份	系统备份	备份完成	恢复备份 删除备份
2018-12-19 04:47:03 - 2018-12-19 04:47:28	快照备份	全量备份	系统备份	备份完成	恢复备份 删除备份
2018-12-18 04:47:06 - 2018-12-18 04:47:26	快照备份	全量备份	系统备份	备份完成	恢复备份 删除备份

6. 在克隆实例页面中，选择新集群的计费方式：

- **预付费：**在创建集群时需要支付计算集群（一个主节点和一个只读节点）的费用，而存储空间会根据实际数据量按小时计费，并从账户中按小时扣除。如果您要长期使用该集群，**预付费**方式更加划算，而且购买时长越长，折扣越多。
- **按小时付费：**无需预先支付费用，计算集群和存储空间（实际数据量）均按小时计费，并从账户中按小时扣除。如果您只需短期使用该集群，可以选择**按量付费**，用完即可释放，节省费用。

7. 设置以下参数：

参数	说明
克隆源类型	选择备份时间点。
备份时间点	可以设置为过去7天内的任意时间点。
地域	无需修改，与原集群相同。
可用区	无需修改。
网络类型	无需修改。
VPC网络和VPC交换机	建议保持不变，即原集群所在的VPC网络和交换机。
数据库引擎	无需修改。
节点规格	不同规格有不同的最大存储容量和性能，具体请参见 #unique_67 。

参数	说明
节点个数	无需修改。系统将自动创建一个与主节点规格相同的只读节点。
集群名称	如果留空，系统将为您自动生成一个集群名称。创建集群后还可以修改集群名称。
购买时长	预付费集群需要填写此参数。
集群数量	默认为1，无法修改。

8. 阅读并勾选《云数据库 PolarDB服务协议》，单击**去开通**，然后完成支付。

按备份集（快照）恢复

1. 登录[PolarDB控制台](#)。
2. 在控制台左上角，选择集群所在地域。
3. 单击目标集群ID。
4. 在左侧导航栏中，选择**配置与管理 > 备份恢复**。
5. 找到目标备份集（快照），单击**恢复备份**，在弹出的对话框中单击**确定**。
6. 在**克隆实例**页面中，选择新集群的计费方式：
 - **预付费**：在创建集群时需要支付计算集群（一个主节点和一个只读节点）的费用，而存储空间会根据实际数据量按小时计费，并从账户中按小时扣除。如果您要长期使用该集群，**预付费**方式更加划算，而且购买时长越长，折扣越多。
 - **按小时付费**：无需预先支付费用，计算集群和存储空间（实际数据量）均按小时计费，并从账户中按小时扣除。如果您只需短期使用该集群，可以选择**按量付费**，用完即可释放，节省费用。
7. 设置以下参数：

参数	说明
克隆源类型	选择 备份集 。
克隆源备份集	请确认是否为您要恢复的备份集。
地域	无需修改，与原集群相同。
可用区	无需修改。
网络类型	无需修改。
VPC网络和VPC交换机	建议保持不变，即原集群所在的VPC网络和交换机。
数据库引擎	无需修改。

参数	说明
节点规格	不同规格有不同的最大存储容量和性能，具体请参见 #unique_67 。
节点个数	无需修改。系统将自动创建一个与主节点规格相同的只读节点。
集群名称	如果留空，系统将为您自动生成一个集群名称。创建集群后还可以修改集群名称。
购买时长	预付费集群需要填写此参数。
集群数量	默认为1，无法修改。

8. 阅读并勾选《云数据库 PolarDB服务协议》，单击去开通，然后完成支付。

常见问题

1. 按时间点恢复的功能是否依赖Binlog？Binlog保存几天，就可以恢复这几天内任意时间点的数据吗？

答：按时间点恢复的功能不依赖Binlog。集群可以恢复到过去7天内的任意时间点，是基于Redo日志的，与Binlog无关。

2. 数据恢复是否基于“全量备份+Binlog”？

答：数据恢复是基于“全量快照备份+Redo日志”。

Redo日志量大小与数据库的写负载有关，写入/更新频繁的数据库会产生较多的Redo日志。系统会定期上传Redo日志，随后清理本地Redo日志，清理前本地Redo日志会临时占用集群的存储空间，收取一定的费用，上传后就不再收取这部分费用了。

相关主题

[备份数据](#)

相关API

API	描述
#unique_109	创建PolarDB集群。  说明： 克隆集群时，参数CreationOption取值需要为CloneFromPolarDB。

12.4 克隆集群

您可以根据已有的PolarDB集群的数据（包括账号信息，但不包括集群参数配置信息），克隆出相同的PolarDB集群。

注意事项

被克隆的是执行克隆动作时的数据。克隆开始后，新写入的数据不会被克隆。

操作步骤

1. 登录[PolarDB控制台](#)。
2. 在控制台左上角，选择集群所在地域。
3. 找到要克隆的集群，单击操作列的... > 恢复到新集群。
4. 在配置页面设置以下参数：

参数	说明
克隆源类型	这里选择 本集群 。
地域	指集群所在的地理位置。克隆集群的地域和原集群相同，不支持修改。
主可用区	- 可用区是地域中的一个独立物理区域，不同可用区之间没有实质性区别。 - 您可以选择将PolarDB与ECS创建在同一可用区或不同的可用区。
网络类型	- 无需选择。 - 仅支持专有网络VPC（Virtual Private Cloud）。VPC是一种隔离的网络环境，安全性和性能均高于传统的经典网络。
VPC网络 VPC交换机	从下拉菜单中选择VPC和交换机，或者 创建新的VPC和交换机 。  说明： 请确保PolarDB与需要连接的ECS创建于同一个VPC，否则它们无法通过内网互通，无法发挥最佳性能。
数据库引擎	无需选择。
节点规格	按需选择。不同规格有不同的最大存储容量和性能，具体请参见 #unique_67 。
节点个数	无需选择。系统将自动创建一个与主节点规格相同的只读节点。
集群名称	- 可选。 - 如果留空，系统将为您自动生成一个集群名称。创建集群后还可以修改集群名称。
购买时长	预付费集群需要填写此参数。
购买数量	默认为1，无法修改。

5. 阅读并勾选《云数据库 PolarDB服务协议》，单击去开通。

12.5 备份功能FAQ

本文介绍PolarDB MySQL备份功能的常见问题。

- Q：一级备份大小是否为所有快照备份之和？

A：一级备份大小不是所有快照备份之和，即并非下图中的②，而是①所示的大小。

备份列表

日志列表

【Binlog备份】数据库备份DBS可满足异地备份、长期归档、备份可读等需求。[查看更多介绍](#)

创建备份

按时间点恢复

备份SQL文件及Binlog

备份设置

2020年3月12日

-

2020年5月12日

一级备份(快照)总大小 1.30 GB

备份集ID	备份开始时间	备份结束时间	状态	一致性快照时间点	备份方法	备份类型	大小	存储位置
	2020年5月12日 06:56:32	2020年5月12日 06:56:37	备份完成	2020年5月12日 06:56:34	快照备份	全量备份	5.74 GB	一级备份
	2020年5月11日 06:56:24	2020年5月11日 06:56:40	备份完成	2020年5月11日 06:56:27	快照备份	全量备份	5.64 GB	一级备份
	2020年5月10日 06:56:25	2020年5月10日 06:56:30	备份完成	2020年5月10日 06:56:27	快照备份	全量备份	5.54 GB	一级备份

- Q：为什么一级备份的总大小比单个备份还要小？

A：PolarDB的一级备份有两个容量数据，一个是每个备份的逻辑大小，一个是全部备份的物理大小。PolarDB的一级备份采用快照链的机制，相同的数据块只会记录一份，因此总物理大小要小于逻辑大小，有时候甚至会小于单个备份逻辑大小。

- Q：PolarDB备份有哪些费用？

A：一级备份、二级备份以及日志备份的存储空间费用。其中一级备份和日志备份默认开启，并赠送一定的免费空间。二级备份默认关闭。

- Q：一级备份的费用怎么算？

A：每小时费用 = [一级备份总大小 - (数据库存储用量 × 50%)] × 每小时价格。例如，PolarDB数据库的一级备份总大小为700G，数据库存储用量为1000GB，那么每小时费用为 (700G-500G) * 0.003元/GB = 0.6元。

- Q：存储包能否抵扣备份空间的费用？

A：目前存储包只能抵扣数据存储占用的空间，不能抵扣备份占用的空间。

相关文档

[备份数据](#)

13 诊断与优化

13.1 SQL洞察

SQL洞察功能为您的数据库提供安全审计、性能诊断等增值服务。

费用说明

- 试用版：免费使用，审计日志仅保存一天，即只能查询一天范围内的数据；不支持数据导出等高级功能；不保障数据完整性。
- 30天或以上：详情请参见[#unique_67](#)。

功能说明

- SQL审计日志。

记录对数据库执行的所有操作。通过审计日志记录，您可以对数据库进行故障分析、行为分析、安全审计等操作。

- 增强搜索。

可以按照数据库、用户、客户端IP、线程ID、执行耗时、执行状态等进行多维度检索，并支持导出和下载搜索结果。

SQL洞察

服务设置

搜索

分析

设置查询条件

关键字

可多字段组合查询，字段间以空格分隔

or

▼

时间范围

2018-12-21 12:57:55 - 2018-12-21 13:12:55

自定义

▼

数据库

可组合查询，如：DB1 DB2 DB3

客户端IP

可组合查询，如：IP1 IP2 IP3

执行状态

☐ 成功 ☐ 失败

扫描记录数

-

用户

可组合查询，如：user1 user2 user3

操作类型

☐ SELECT ☐ INSERT ☐ UPDATE ☐ DELETE

更多

线程ID

可组合查询，如：Threadid1 Threadid2 Threadid3

执行耗时

-

关闭高级查询

查询

日志列表

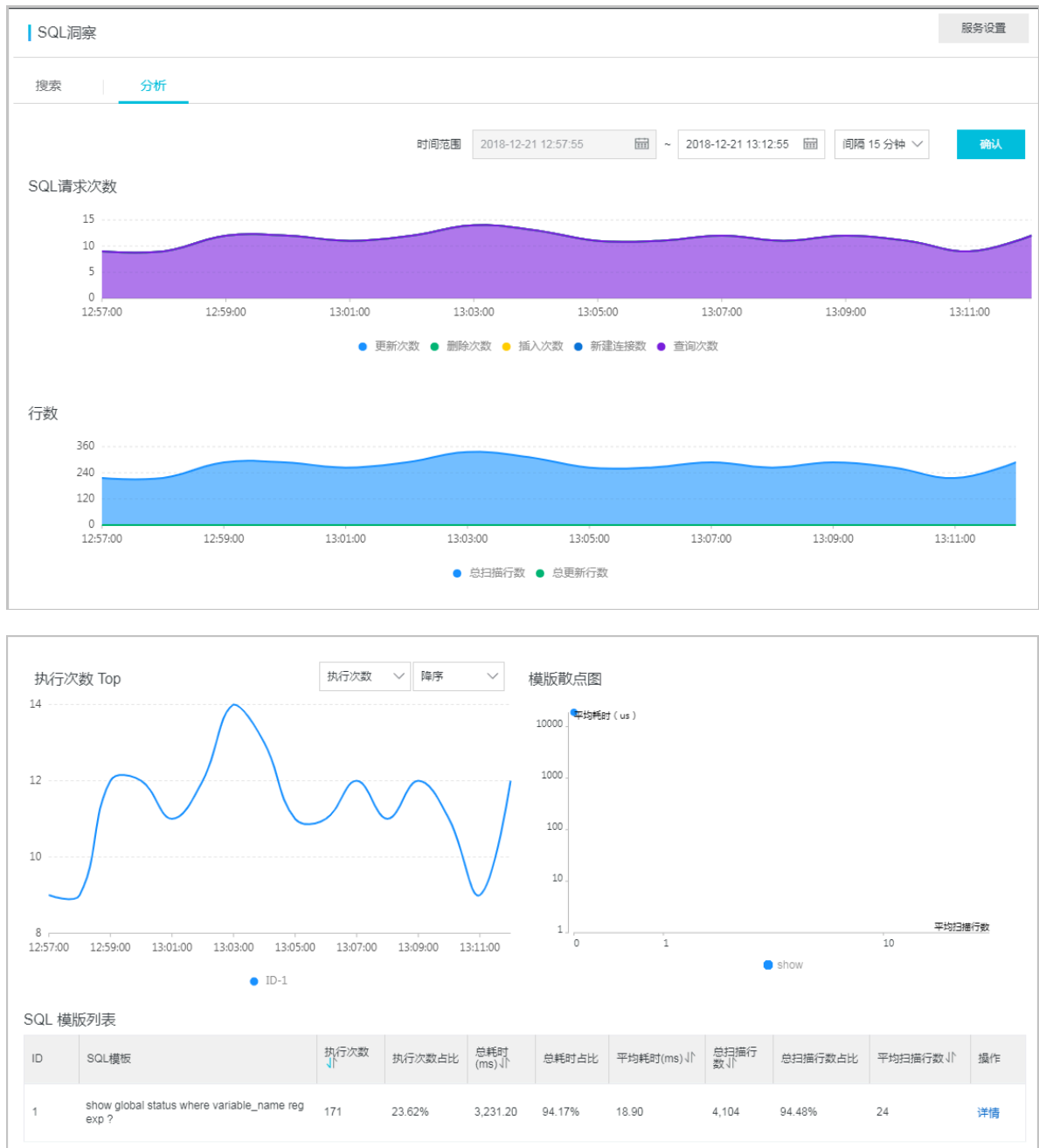
导出

查看导出列表

SQL语句	数据库	线程ID	用户	客户端IP	操作	状态	耗时 (ms) ↓↑	执行时间 ↓↑	更新行 数 ↓↑	扫描行 数 ↓↑

- SQL分析。

可以对指定时间段的SQL日志进行可视化交互式分析，找出异常SQL，定位性能问题。



开通SQL洞察

1. 登录POLARDB控制台。
2. 在控制台左上角，选择集群所在地域。
3. 单击目标集群ID。
4. 在左侧导航栏中，选择日志与审计 > SQL洞察。

5. 单击立即开通。

欢迎体验SQL洞察

通过采集、分析SQL原始日志，帮您洞察数据库的安全和性能风险。

试用版，您只能查询一天范围内的SQL洞察数据，试用版不支持数据导出等高级功能并不保障数据完整性。如您需要数据导出或数据完整性等或调整SQL洞察查询周期等高级功能，欢迎咨询和使用付费版SQL洞察。

立即开通试用版

6. 选择SQL审计日志的保存时长，单击开通服务。

存储时长

☐ 试用版 ☒ 30天 ☐ 6个月 ☐ 1年 ☐ 3年 ☐ 5年

SQL日志保存的时长，超过这个时长的SQL日志将被删除。

试用版，您只能查询一天范围内的SQL洞察数据，试用版不支持数据导出等高级功能并不保障数据完整性。如您需要数据导出或数据完整性等或调整SQL洞察查询周期等高级功能，欢迎咨询和使用付费版SQL洞察。

开通服务取消

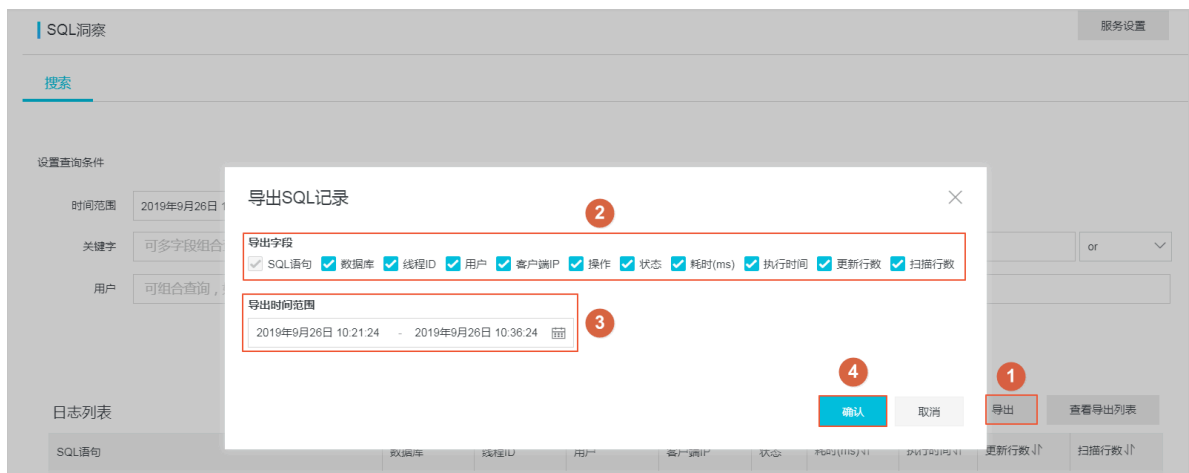
修改SQL日志的存储时长

1. 登录POLARDB控制台。
2. 在控制台左上角，选择集群所在地域。
3. 单击目标集群ID。
4. 在左侧导航栏中，选择日志与审计 > SQL洞察。
5. 单击右上角服务设置。
6. 修改存储时长并单击确认。

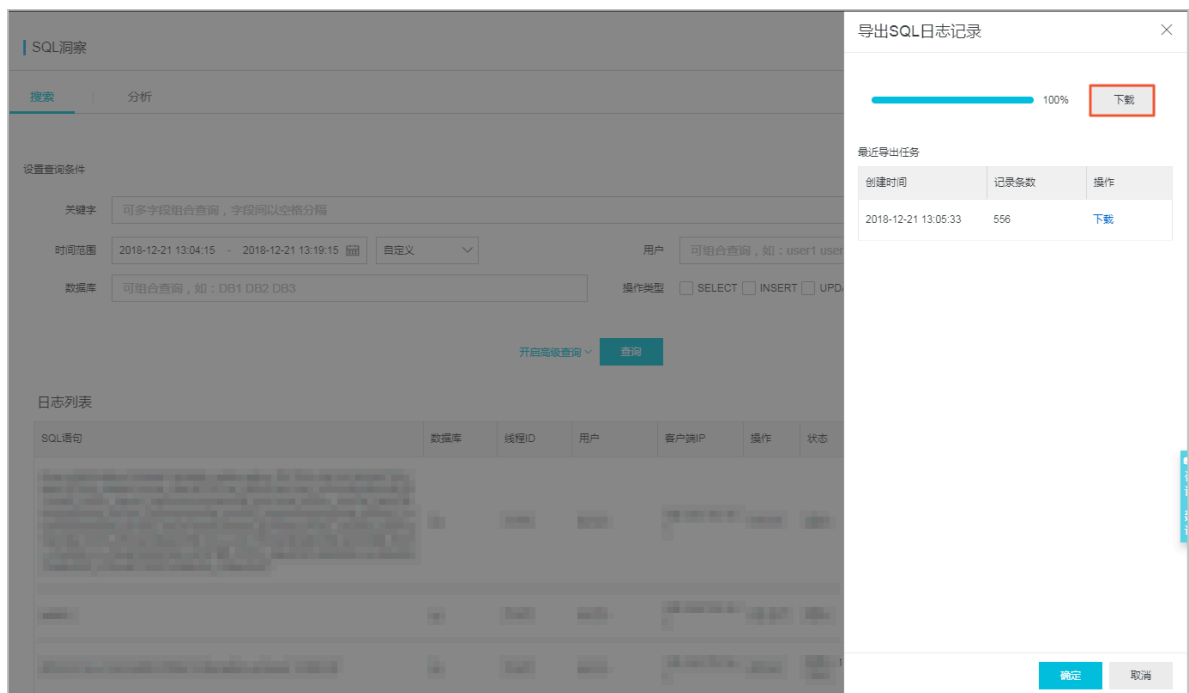
导出SQL记录

1. 登录POLARDB控制台。
2. 在控制台左上角，选择集群所在地域。
3. 单击目标集群ID。
4. 在左侧导航栏中，选择日志与审计 > SQL洞察。
5. 单击右侧导出。

6. 在弹出的对话框中，勾选需要导出的选项，单击**确认**。



7. 导出完成后，在**导出SQL日志记录**中，下载已导出的文件并妥善保存。



关闭SQL洞察



说明：

SQL洞察功能关闭后，SQL审计日志会被清空。请将**SQL审计日志导出**后，再关闭SQL洞察功能。

1. 登录**POLARDB控制台**。
2. 在控制台左上角，选择集群所在地域。
3. 单击目标集群ID。
4. 在左侧导航栏中，选择**日志与审计 > SQL洞察**。

5. 单击**服务设置**。
6. 单击滑块关闭SQL洞察。



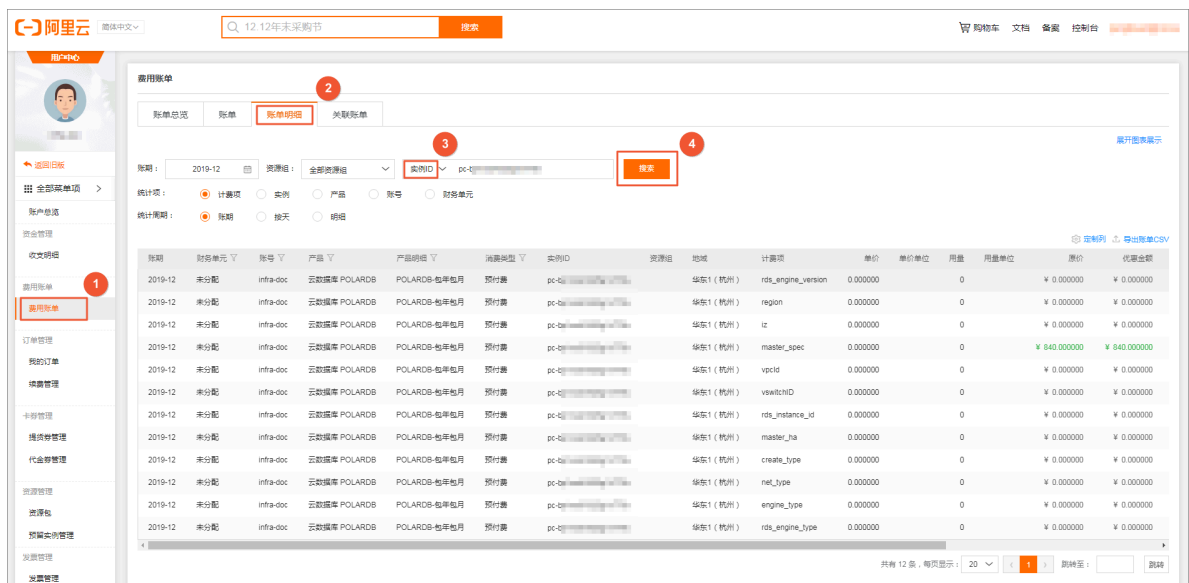
查看审计日志的大小和消费明细

1. 登录[阿里云管理控制台](#)。
2. 在页面右上角，选择**费用 > 用户中心**。
3. 在左侧导航栏中，选择**费用账单 > 费用账单**。
4. 选择**账单明细**页签，设置搜索**实例ID**，并搜索目标实例。



说明：

若要查询超过12个月前的记录，请提交工单。



5. 查看计费项列为**sql_explorer**的费用明细。

13.2 性能监控与报警

为了方便您掌握实例的运行状态，PolarDB控制台提供了丰富的性能监控项。

性能监控

1. 登录[PolarDB控制台](#)。
2. 在控制台左上角，选择集群所在地域。
3. 单击目标集群ID。
4. 在左侧导航栏中，选择**诊断与优化 > 性能监控**。
5. 根据您的需求，可以查看**集群**或**节点**的监控信息。详细说明请参见[监控项说明](#)。
 - 集群性能监控：单击**集群**页签，在右侧设置时间段后单击**确定**。



- 节点性能监控：单击**节点**页签，在右侧选择节点并设置时间段后单击**确定**。



**说明：**

数据库自治服务DAS[实时性能大盘](#)功能现已支持查看PolarDB MySQL，可以帮助您快速识别异常实例，进行进一步的优化分析。

监控项说明

类别	监控项	说明
集群	存储空间	展示本地Binlog、Redolog和其他日志文件的大小，以及数据空间、系统空间和临时空间的使用量。
	QPS	展示各节点的每秒请求数。
	TPS	展示各节点的每秒事务数。
	MPS	展示各节点的每秒数据操作数。
	CPU	展示各节点的CPU使用率。
	内存	展示各节点的内存使用率。
节点	QPS	展示所选择节点的每秒请求数。
	TPS	展示所选择节点的每秒事务数。
	MPS	展示所选择节点的每秒数据操作数。
	CPU	展示所选择节点的CPU使用率。
	内存	展示所选择节点的内存使用率。
	连接	展示所选择节点的当前总连接数和活跃连接数。
	操作	展示所选择节点的每秒操作数，包括DELETE、INSERT、UPDATE、REPLACE等等。
	内存缓冲池	展示所选择节点的缓冲池的脏块率、读命中率、利用率。
	I/O吞吐量	展示所选择节点的总I/O吞吐量、读I/O吞吐量、写I/O吞吐量。
	IOPS	展示所选择节点的每秒读写次数，包括每秒读写总次数、每秒读次数、每秒写次数。
	网络	展示所选择节点的每秒输入输出流量。
	扫描行数	展示所选择节点每秒插入、读取、更新、删除的行数。
	InnoDB读写数据量	展示所选择节点每秒从存储引擎读取的数据量和每秒往存储引擎写入的数据量。
	InnoDB缓冲池请求数	展示所选择节点缓冲池的每秒读写次数。
	InnoDB日志写	展示所选择节点日志的每秒写请求数量和每秒同步数据到磁盘的次数。

类别	监控项	说明
	临时表	展示所选择节点每秒创建的临时表数量。

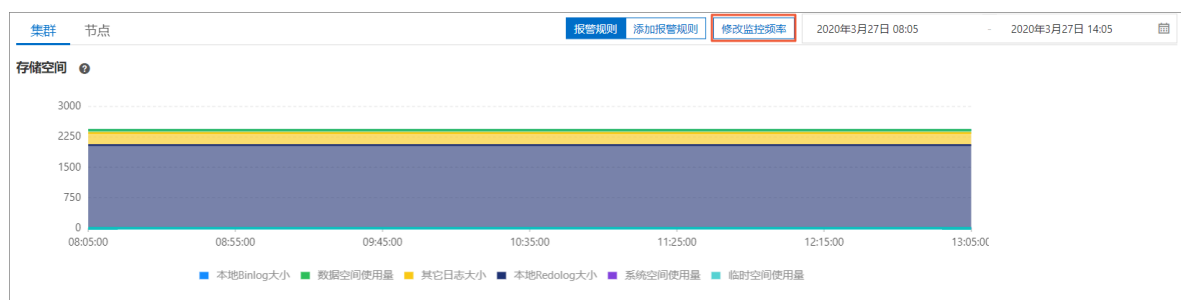
修改监控频率



说明：

基于分钟级别的监控采集频率已无法满足客户和运维人员对数据库的监控需求。PolarDB推出更细粒度的监控采集频率——秒级监控频率，可以解决日常监控数据粒度过粗，无法定位运维问题的需求，同时提高客户满意度。

1. 登录[PolarDB控制台](#)。
2. 在控制台左上角，选择集群所在地域。
3. 单击目标集群ID。
4. 在左侧导航栏中，选择**诊断与优化 > 性能监控**。
5. 单击**修改监控频率**。



6. 在监控频率设置对话框，选择监控频率。

- 监控频率为5秒时：
 - 当查询时间范围小于等于1小时，显示粒度为5秒。
 - 当查询时间范围小于等于1天，显示粒度为1分钟。
 - 当查询时间范围小于等于7天，显示粒度为10分钟。
 - 当查询时间范围小于等于30天，显示粒度为1小时。
 - 当查询范围大于30天，显示粒度为1天。
- 监控频率为60秒时：
 - 当查询范围小于等于1天，显示粒度为1分钟。
 - 当查询范围小于等于7天，显示粒度为10分钟。
 - 当查询范围小于等于30天，显示粒度为1小时。
 - 当查询范围大于30天，显示粒度为1天。

**说明：**

默认为60秒/次，您可以按需设置为5秒/次。

监控频率设置

监控频率:

☐ 5秒/次

☒ 60秒/次

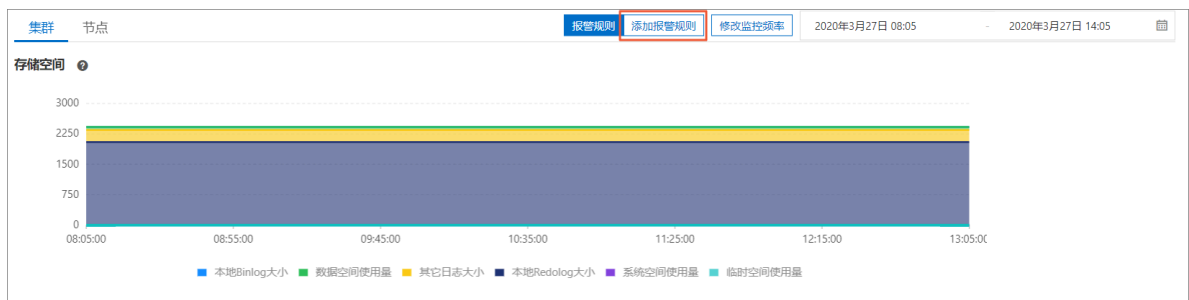
确认

取消

7. 单击确认。**添加报警规则**

1. 登录[PolarDB控制台](#)。
2. 在控制台左上角，选择集群所在地域。
3. 单击目标集群ID。
4. 在左侧导航栏中，选择。

5. 单击添加报警规则。



6. 在创建报警规则页面，选择资源范围，设置报警规则和通知方式后，单击确认即可。



说明：

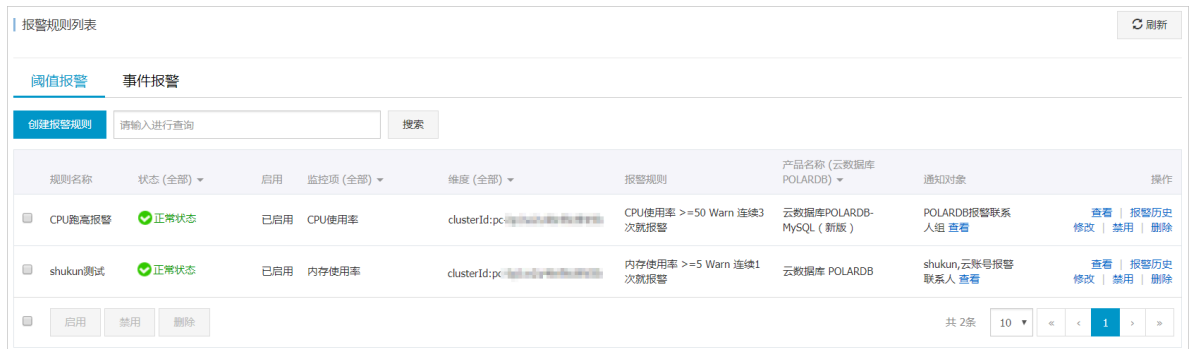
如果资源范围选择集群，还需要选择集群所在地域和目标集群。更多报警规则参数说明，请参见[#unique_207](#)。

成功添加报警规则后，当集群被监控的指标超过设定阈值后自动发送报警通知给指定的联系人。

报警规则管理

1. 登录[PolarDB控制台](#)。
2. 在控制台左上角，选择集群所在地域。
3. 单击目标集群ID。
4. 在左侧导航栏中，选择。

5. 单击报警规则，进入报警规则列表页面。



6. 在报警规则列表页面，您可以进行报警规则管理，具体操作如下。

- 单击操作栏中的**查看**，可查看报警规则的基本信息。
- 单击操作栏中的**报警历史**，可查看报警历史。
- 单击操作栏中的**修改**，可修改报警规则。
- 单击操作栏中的**禁用**，可禁用报警规则。
- 单击操作栏中的**删除**，可删除报警规则的。
- 单击通知对象栏中的**查看**，可查看报警联系组、报警联系人及报警通知方式。

相关API

API	描述
#unique_208	查询PolarDB集群的性能数据。
#unique_209	查询PolarDB集群节点的性能数据。
#unique_210	查询PolarDB集群监控数据的采集频率。
#unique_211	修改PolarDB集群监控数据的采集频率。

13.3 慢SQL

POLARDB for MySQL提供慢SQL分析功能，能够查看慢日志趋势和统计信息，并且提供SQL建议和诊断分析。

查看慢日志趋势和慢日志统计

- 登录[POLARDB控制台](#)。
- 在控制台左上角，选择集群所在地域。
- 单击目标集群ID。
- 在左侧导航栏中，选择**诊断与优化 > 慢SQL**。

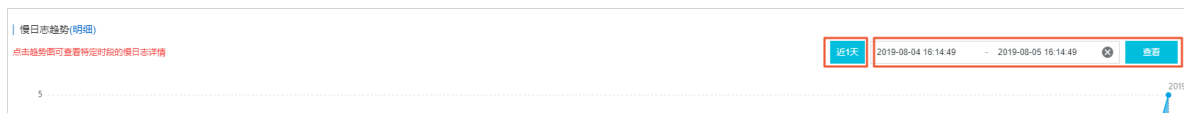
5. 您可以通过以下两种方式查看慢日志趋势：

- 单击**近1天**，查看最近24小时的慢日志情况。
- 设置查询起止日期，然后单击**查看**，可以查看最多7天的慢日志情况。



说明：

默认展示近1天的慢日志情况。



6. 查看慢日志趋势折线图，单击慢SQL发生的时间节点，即可在下方看到慢日志统计。



7. 您可以通过如下两种方式排查慢SQL问题：

- 单击目标慢SQL操作列的**样本**，可查看该次慢SQL的详细信息。

慢日志样本

执行时间	SQL模板	库名	客户端	用户	执行耗时 (秒)	锁等待耗时 (秒)	扫描行	返回行
2019-08-05 16:11:34	select * from company where n...		110		2.0	0.0	854.9K	313.8K

关闭

- 单击目标慢SQL操作列的**优化**，可查看诊断优化信息。

SQL诊断优化 (专家服务)

SQL

执行计划

ID	select_type	table	extra	rows	possible_k...	key	key_len	ref
1	SIMPLE	company		945820				

诊断结果

诊断时间: 2019-08-05 16:25:11

确认 取消



说明：

您也可以单击**专家服务**，付费购买数据库专家服务，为您提供云数据库产品本身能力范围之外的专业的数据库专家服务，主要包括紧急救援、健康诊断、性能调优、护航保障、数据迁移等。

导出慢日志

您可以单击**导出慢日志**将慢日志信息保存到本地查看。

慢日志统计 (时间范围: 2019-08-05 16:00:00 - 2019-08-05 17:00:00)

① 当1个小时慢日志数量多于15000条时,仅显示前15000条慢日志统计

SQL模板	库名	执行次数	平均执行时间 (秒)	最大执行时间 (秒)	平均锁等待时间 (秒)	最大锁等待时间 (秒)	平均扫描行	最大扫描行	平均返回行	最大返回行	操作
SELECT * FROM company ...		1	2.00	2	0.00	0	854.9K	855K	313.84K	314K	样本 优化
SELECT * FROM company ...		1	1.00	1	0.00	0	854.9K	855K	103.63K	104K	样本 优化

13.4 一键诊断

PolarDB MySQL的一键诊断融合了DAS部分功能，支持自治中心、会话管理、实时性能、空间分析、锁分析、诊断报告和性能洞察等功能，方便您查看数据库相关的诊断与优化结果。

前提条件

首次使用一键诊断功能需要对集群进行账号授权，详情请参见[账号授权](#)。

账号授权

1. 登录[PolarDB控制台](#)。
2. 在控制台左上角，选择集群所在地域。
3. 单击目标集群ID。
4. 在左侧导航栏中，选择**诊断与优化 > 一键诊断**。
5. 在弹出的对话框中输入**数据库账号**和**密码**，单击**授权**。

账号授权

* 数据库账号：

* 密码：

授权

6. 授权成功后，单击**完成**。

功能汇总

- **自治中心**：详情请参见[自治中心](#)。
- **会话管理**：详情请参见[会话管理](#)。
- **实时性能**：详情请参见[实时性能](#)。
- **空间分析**：详情请参见[空间分析](#)。
- **锁分析**：详情请参见[锁分析](#)。
- **性能洞察**：详情请参见[性能洞察](#)。
- **诊断报告**：详情请参见[诊断报告](#)。

13.5 自治中心

PolarDB MySQL的一键诊断融合了DAS部分功能，您可以通过其中的自治中心开启自治服务。自治服务开启后，DAS会在数据库出现异常时，自动进行根因分析，给出优化或止损建议，并自动进行优化或止损操作（需经您授权后才会开启优化操作）。

前提条件

目前仅支持PolarDB MySQL 5.6和PolarDB MySQL 5.7。

操作步骤

1. 登录[PolarDB控制台](#)。
2. 在控制台左上角，选择集群所在地域。
3. 单击目标集群ID。
4. 在左侧导航栏中单击**诊断与优化 > 一键诊断**。
5. 在自治中心页面右上角，单击**开关设置**。



6. 在**开关设置**对话框中，打开**开启自治服务**开关。



说明：

打开**开启自治服务**开关后，会同时默认开启自动SQL诊断，并输出SQL优化意见，但该功能不支持CPU规格为1核的实例。

开关设置

✕

开启自治服务 ☒ 开启后，会自动进行容量评估、会话分析、限流SQL分析、异常快照保存（对数据库不会增加额外负载）

自动索引创建和删除

关闭

SQL诊断 ^①

开启自动索引创建

开启自动索引删除

自动限流

☒

CPU利用率 > % 且

持续时间 > 分钟 最大限流时间 = 分钟

注意：CPU利用率至少大于80%，活跃会话数至少大于64且不可为小数，持续时间至少大于5分钟，最大限流时间不能为负数

自动扩容

☒

规格上限 观测窗口 静默期 ^①

在观测窗口内，CPU利用率 > 70%的时间超过50%

自动回缩 ^①

☒

在观测窗口内，CPU利用率 < 30%的时间超过99%，进行自动回缩 [详情](#)

确定

取消修改

7. 自治服务下包含**自动索引创建和删除**、**自动限流**、**自动扩容**和**自动回缩**服务。

- **自动索引创建和删除**：当打开**开启自治服务**开关后，默认开启**SQL诊断**。您也单击**开启自动索引创建**或**开启自动索引删除**开启创建或删除自动索引。
- **自动限流**：您可以自定义限流触发条件，当条件满足时会自动触发SQL限流，详情请参见[自动限流](#)。
- **自动扩容**：您可以自定义扩容触发条件，当条件满足时会自动触发扩容，详情请参见[自动弹性伸缩](#)。
- **自动回缩**：开启自动回缩后，在观测窗口内，当CPU利用率小于30%的时间超过99%时，将触发自动回缩。

8. 单击**确定**。

13.6 会话管理

PolarDB MySQL的一键诊断融合了DAS部分功能，您可以通过其中的会话管理功能查看目标实例的会话统计信息。

操作步骤

1. 登录PolarDB控制台。
2. 在控制台左上角，选择集群所在地域。
3. 单击目标集群ID。
4. 在左侧导航栏中单击**诊断与优化 > 一键诊断**。
5. 单击**会话管理**页签，进入会话管理页面。



6. 在**会话管理**页签，您可以查看**实例会话**和**会话统计**的详细情况。

- **实例会话**：支持查看异常会话、活跃会话及CPU利用率等当前实例会话信息。



说明：

- 您可以通过单击页面右上角的**10秒SQL请求**按钮，了解在10秒钟内，执行次数最多的SQL以及是否存在慢SQL等信息，详情请参见[10秒SQL分析](#)。
- 您可以通过单击**SQL限流**按钮，控制数据库请求访问量和SQL并发量，保障服务的可用性，详情请参见[SQL限流](#)。

- 您可以通过单击**优化**按钮，对SQL进行诊断和优化，详情请参见[SQL优化](#)。

- **会话统计**：支持查看会话总数、会话时间，以及按照不同统计项（如按用户统计）对会话数量进行统计。

会话统计		
概要		
用户	统计项	
会话总数	3	
运行中会话总数	0	
运行中会话最长时间	0	
按用户统计(1)		
用户	活跃数	总数
root	0	3
按访问来源统计(1)		
来源	活跃数	总数
阿里云ECS	0	3
按数据库统计(1)		
DB	活跃数	总数
	0	3

13.7 实时性能

PolarDB MySQL的一键诊断融合了DAS部分功能，您可以通过其中的实时性能功能，了解目标集群的QPS、TPS和网络流量等信息。

操作步骤

1. 登录[PolarDB控制台](#)。
2. 在控制台左上角，选择集群所在地域。
3. 单击目标集群ID。
4. 在左侧导航栏中，选择**诊断与优化 > 一键诊断**。
5. 单击**实时性能**页签，进入实时性能页面。

自治中心	会话管理	实时性能	空间分析	锁分析	诊断报告	性能洞察
------	------	-------------	------	-----	------	------

6. 在**实时性能**页签，您可以查看集群内不同节点的Server信息和buffer Pool等性能详情。



说明：

若需要了解详细的性能参数说明，请单击左上方**参数说明**进行查看。

实时性能 参数说明		剩余刷新次数: 903 次	暂停
当前节点: 选择节点 复制			
Server信息		连接信息	Buffer Pool
版本/运行时间		连接数上限/当前连接数/活跃连接	总页数/空闲页比例/脏页比例
5.6.16-log / 5天4时57分		8512 / - / 3	786432 / 99.98% / 0.00%

7. 单击**实时图表**页签，各个性能参数将以折线图的形式展现，方便您查看性能的变化趋势。



说明：

若您想了解具体的性能值，请单击**实时表格**页签进行查看。



13.8 空间分析

PolarDB MySQL的一键诊断融合了DAS部分功能，您可以通过空间分析，来查看目标实例的空间使用概况、空间剩余可用天数，以及数据库中某个表的空间使用情况、空间碎片、空间异常诊断等。

操作步骤

1. 登录[PolarDB控制台](#)。
2. 在控制台左上角，选择集群所在地域。
3. 单击目标集群ID。
4. 在左侧导航栏中，选择**诊断与优化 > 一键诊断**。
5. 单击**空间分析**页签，进入空间分析页面。



6. 在空间分析页面，您可以查看以下信息。

- 单击**空间概况**页签，您可以查看库表空间概况和空间变化趋势等数据。



- 单击**数据空间**页签，您可以查看每个数据库和单个表的的空间使用情况。单击表名可查看对应表的字段和索引。

5.6.3节点 类型: 主实例 所属集群: 接入状态: 连接正常

当前结果分析时间 2020年6月23日 16:17:59

空间概况 数据空间

数据空间

库名 (点击查看详情)	数据库名	存储引擎	表空间	表空间占比	索引空间	数据空间	碎片率	索引数	平均行长
(576.00 KB)	表空间								
(13.03 MB)	表名 (点击查看详情)	数据引擎	存储引擎						
(587.52 MB)	tablespace	disttestdata	InnoDB						
(0 B)		disttestdata	InnoDB						

表名: tablespace, 库名: tablespace

字段名	类型	描述	是否	自增	缺省值
orderid	int(11)		NO	NO	
username	char(32)		YES	NO	
ordertime1	datetime		YES	NO	
commodity	varchar(32)		YES	NO	
phonenumber	int(11)		YES	NO	
address	text		YES	NO	

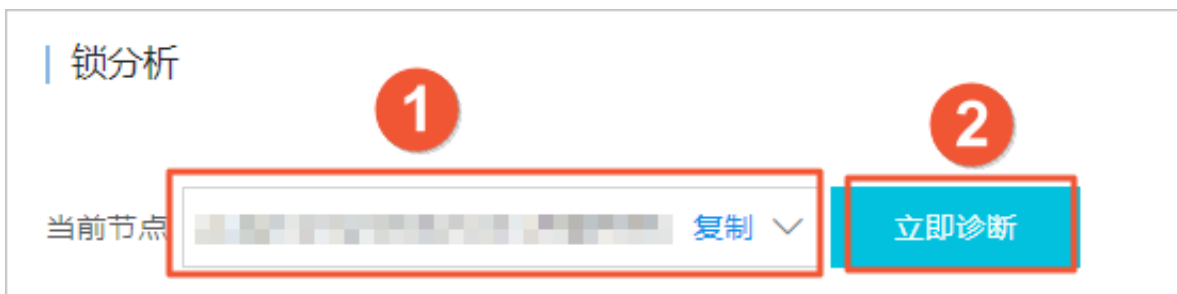
13.9 锁分析

PolarDB MySQL的一键诊断融合了DAS部分功能，您可以通过其中的锁分析直观地查看和分析数据库最近一次发生的死锁。

操作步骤

1. 登录**PolarDB控制台**。
2. 在控制台左上角，选择集群所在地域。
3. 单击目标集群ID。
4. 在左侧导航栏中单击**诊断与优化 > 一键诊断**。
5. 单击**锁分析**页签，进入锁分析页面。

6. 在锁分析页面，从**当前节点**列表中，选择目标实例ID后，单击**立即诊断**。



7. 在诊断列表里找到存在死锁的诊断结果，单击其右侧**详情**栏中的**查看详情**。

诊断生成时间	发生时间	发现死锁	详情
2020年5月11日 22:24:55	2020年4月10日 20:58:04	是	查看详情

8. 在**锁分析**对话框中，查看死锁诊断结果的详情列表，您也可以单击**查看死锁日志**，查看最近一次的死锁日志详情。

锁分析

查看死锁日志

发生时间: 2020年4月10日 20:58:04

	事务 1(已回滚)	事务 2
Session ID		
请求类型	update	updating
事务ID		
涉及表		
等待锁		
等待锁索引名	uk_ipport	PRIMARY
等待锁类型	X waiting	X locks rec but not gap waiting
持有锁		
持有锁索引名		uk_ipport
持有锁类型	X waiting	X locks rec but not gap waiting
事务SQL		

13.10 诊断报告

PolarDB MySQL的一键诊断融合了DAS部分功能，支持自定义诊断报告创建条件并查看诊断报告。

操作步骤

1. 登录[PolarDB控制台](#)。
2. 在控制台左上角，选择集群所在地域。
3. 单击目标集群ID。
4. 在左侧导航栏中，选择**诊断与优化 > 一键诊断**。
5. 单击**诊断报告**页签，进入诊断报告页面。



6. 在诊断报告页面，您可以通过以下两种方式生成诊断报告。

- **自动生成诊断报告：**

- a. 单击实例诊断报告列表旁边的**自动生成报告设置**按钮。
- b. 在弹出的对话框内，选中**定时触发**复选框。
- c. 选中相应的日期如**全部**复选框，并设置诊断报告生成的时间。
- d. 单击**确认**。



- **手动生成诊断报告：**单击**发起诊断**生成即时诊断报告。



7. 在实例诊断报告列表找到已完成的诊断，在右侧**操作**栏中，单击**查看报告**。

ID	诊断生成时间	开始时间	结束时间	告警数量	诊断状态	操作
772761df1cad9018c5269b119...	2020年5月17日 14:31:53	2020年5月16日 14:31:45	2020年5月17日 14:31:45	暂无	诊断完成	查看报告

8. 在实例诊断报告详情页中，您可以查看实例的健康状况、会话列表、死锁和性能趋势等情况。

实例诊断报告详情			
(2020年5月16日 14:31:45 ~ 2020年5月17日 14:31:45)			
得分: 100 和分详情			
实例基本信息			
实例ID: [实例ID]		实例来源: [实例来源]	
数据库类型: [数据库类型]		库名: [库名]	
节点	角色	规格	最大连接数
[节点ID]	只读节点	规格: 5核16GB 磁盘: 160GB	8000

13.11 性能洞察

PolarDB MySQL的一键诊断融合了DAS的部分功能，您可以通过其中的性能洞察，快速评估数据库负载，找到性能问题的源头，提升数据库的稳定性。

背景信息

用于性能洞察的数据主要来源如下：

- 如果目标实例已经开启了performance_schema，直接采集和分析performance_schema中的数据。
- 如果目标实例未开启performance_schema，则采集和分析活跃会话数据。

操作步骤

1. 登录[PolarDB控制台](#)。
2. 在控制台左上角，选择集群所在地域。
3. 单击目标集群ID。
4. 在左侧导航栏中，选择**诊断与优化 > 一键诊断**。
5. 单击**性能洞察**页签。

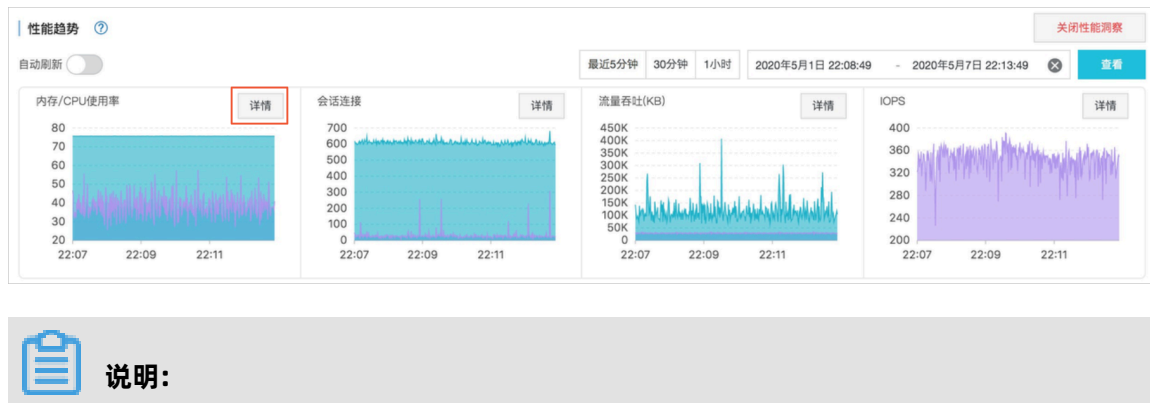
6. 单击开启性能洞察。



7. 在弹出的对话框内，单击确定。

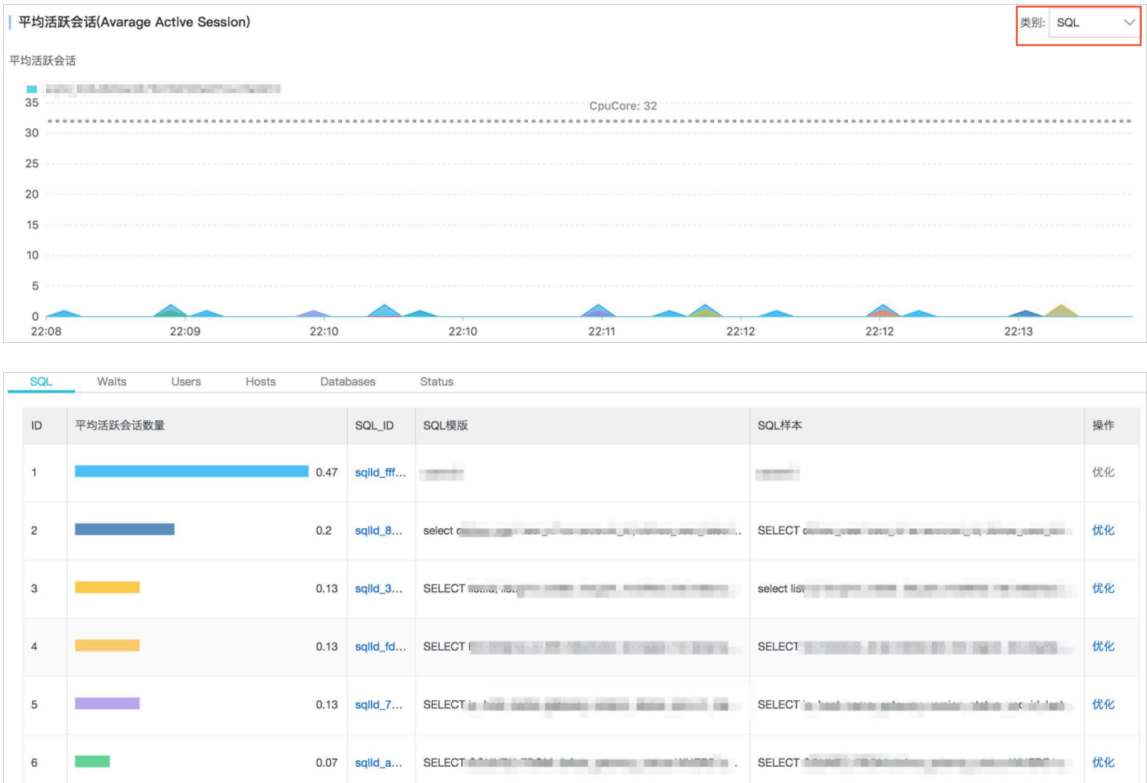
8. 您可以在性能洞察页面查看和管理如下信息。

- 在**性能趋势**区域，您可以查看特定时段的数据库性能情况。若您需要查看某个具体性能（如CPU使用率），可以单击该性能名称右侧的**详情**按钮进行查看。



可供查看的时间范围不能超过7天。

- 在**平均活跃会话**区域，您可以查看不同类别（如SQL）会话的变化趋势图和相关多维负载信息列表，确定性能问题源头。



14 其他操作

14.1 开启Binlog

本文介绍如何为PolarDB集群开启Binlog功能。

前提条件

集群创建于2019年4月5日之后。如果是在2019年4月5日之前创建的集群，暂时需要[提交工单](#)进行小版本升级，之后即可在控制台手动开启Binlog。

背景信息

PolarDB是一款完全兼容MySQL的云原生数据库，默认使用了更高级别的物理日志代替Binlog，但为了更好地与MySQL生态融合，PolarDB上线了开启Binlog的功能，开启Binlog后您就可以连接[ElasticSearch](#)、[AnalyticDB](#)等数据产品，也可以搭建[PolarDB到RDS](#)、[RDS到PolarDB](#)或PolarDB之间的数据实时同步。

收费说明

Binlog的空间属于集群存储空间的一部分，需要收取[存储费用](#)。

注意事项

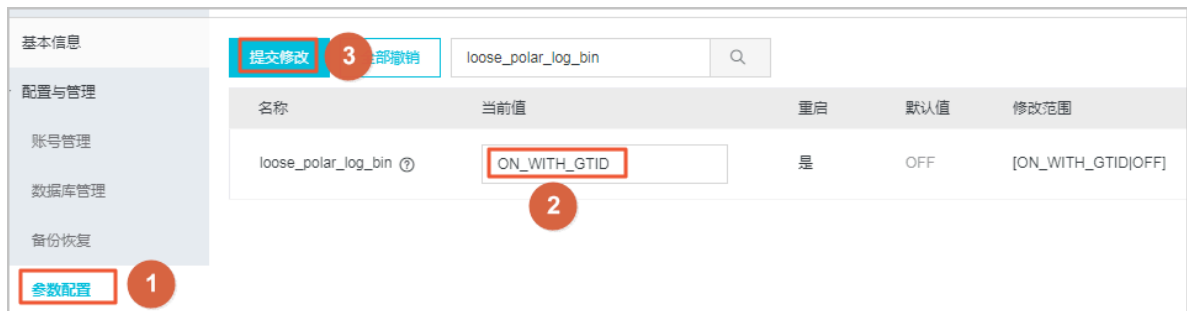
- 开启后Binlog默认保存2周，超出两周的Binlog文件会被自动删除。您可以修改参数`loose_expire_logs_hours`（取值范围0~2376，单位：小时）以设置Binlog的保存时长。0表示不自动删除Binlog文件。
- Binlog功能默认关闭，开启Binlog需要重启实例，会造成连接中断，重启前请做好业务安排，谨慎操作。
- 开启Binlog后会导致写性能下降，读性能不受影响。

- 拉取、订阅或同步Binlog（例如使用DTS等工具）时，建议使用PolarDB的**主地址**，因为直接指向生成Binlog的主节点，具有更好的兼容性和稳定性。您可以在**基本信息**页面查看主地址，如下图所示。



开启Binlog的操作步骤

1. 登录PolarDB控制台。
2. 选择地域。
3. 找到目标集群，单击**集群名称**列的集群ID。
4. 在左侧导航栏中选择**配置与管理** > **参数配置**。
5. 搜索loose_polar_log_bin，修改当前值，并单击**提交修改**。



说明：

- 数据库引擎为MySQL 5.6的PolarDB集群，当前值修改为ON_WITH_GTID；
- 数据库引擎为MySQL 8.0的PolarDB集群，当前值修改为ON。

6. 在右侧提示重启的对话框中单击**确定**。



说明：

如果报错提示Custins minor version does not support current action，请[提交工单](#)开启。

常见问题

- Binlog能保存多久？

答：Binlog默认保存2周，超出两周的Binlog文件会被自动删除。您可以修改参

数`loose_expire_logs_hours`（取值范围0~2376，单位：小时）以设置Binlog的保存时长。0表示不自动删除Binlog文件。

- 开启Binlog后可以关闭吗？

答：将参数`loose_polar_log_bin`修改为OFF并提交即可关闭。关闭后已有的Binlog不会被删除。

- 开启Binlog对性能影响有多大？

答：测试数据显示，在64线程并发下，开启Binlog后会有30%~40%的写性能衰减（性能衰减随并发增加而减少，后续将会持续优化），读性能不受影响。对于读多写少的业务场景，整体数据库性能影响较小，例如读写比是4:1的数据库，整体性能影响大约是10%。

14.2 设置集群参数

本文将介绍如何通过控制台修改集群参数。

背景信息

您可以通过PolarDB控制台修改集群参数，关于参数的详细说明，请参见[参数说明](#)。

注意事项

- 请按照控制台上规定的**修改范围**修改参数值。

<

集群 pc-

运行中

基本信息

配置与管理

账号管理

数据库管理

备份恢复

参数配置

诊断与优化

日志与审计

提交修改

全部撤销

请输入

Q

名称	当前值	重启	默认值	修改范围
autocommit	ON	否	ON	[ON OFF]
automatic_sp_privileges	ON	否	ON	[ON OFF]
back_log	3000	是	3000	[0-65535]
binlog_checksum	CRC32	否	CRC32	[NONE CRC32]
binlog_rows_query_log_events	OFF	否	OFF	[ON OFF]
binlog_row_image	FULL	否	FULL	[full minimal nolib]
binlog_stmt_cache_size	32768	否	32768	[4096-4294967295]
character_set_filesystem	binary	否	binary	[utf8 latin1 gbk utf8mb4 binary]
character_set_server	utf8	是	utf8	[utf8 latin1 gbk utf8mb4]
connect_timeout	10	否	10	[2-31536000]

- 部分参数修改后需要重启全部节点，重启前请做好业务安排，谨慎操作。详情请参见[参数设置页面中的重启列](#)。

<

集群

pc-1

运行中

基本信息

配置与管理

账号管理

数据库管理

备份恢复

参数配置

诊断与优化

日志与审计

提交修改

全部撤销

请输入

Q

名称	当前值	重启	默认值	修改范围
autocommit [?]	ON	否	ON	[ON OFF]
automatic_sp_privileges [?]	ON	否	ON	[ON OFF]
back_log [?]	3000	是	3000	[0-65535]
binlog_checksum [?]	CRC32	否	CRC32	[NONE CRC32]
binlog_rows_query_log_events [?]	OFF	否	OFF	[ON OFF]
binlog_row_image [?]	FULL	否	FULL	[full minimal noblob]
binlog_stmt_cache_size [?]	32768	否	32768	[4096-4294967295]
character_set_filesystem [?]	binary	否	binary	[utf8 latin1 gbk utf8mb4 binary]
character_set_server [?]	utf8	是	utf8	[utf8 latin1 gbk utf8mb4]
connect_timeout [?]	10	否	10	[2-31536000]

操作步骤

1. 登录[PolarDB控制台](#)。
2. 在控制台左上角，选择集群所在地域。
3. 单击目标集群ID。
4. 在左侧导航栏中，选择[配置与管理](#) > [参数配置](#)。
5. 修改一个或多个参数的当前值，单击[提交修改](#)。

提交修改

全部撤销

请输入

Q

名称	当前值	重启	默认值	修改范围
autocommit ?	ON	否	ON	[ON OFF]
automatic_sp_privileges ?	ON	否	ON	[ON OFF]
back_log ?	4000	是	3000	[0-65535]
binlog_checksum ?	CRC32	否	CRC32	[NONE CRC32]
binlog_rows_query_log_events ?	OFF	否	OFF	[ON OFF]

6. 在弹出的**保存改动**对话框中，单击**确定**。



相关API

API	描述
#unique_213	查看集群的参数
#unique_214	修改集群的参数

14.3 设置可维护窗口

本文为您介绍如何设置可维护窗口，以便您在维护过程中不会影响业务。

背景信息

在阿里云平台上，为保障云数据库POLARDB的稳定性，后端系统会不定期对集群进行维护操作，确保集群平稳运行。您可以根据业务规律，将可维护窗口设置在业务低峰期，以免维护过程中对业务造成影响。

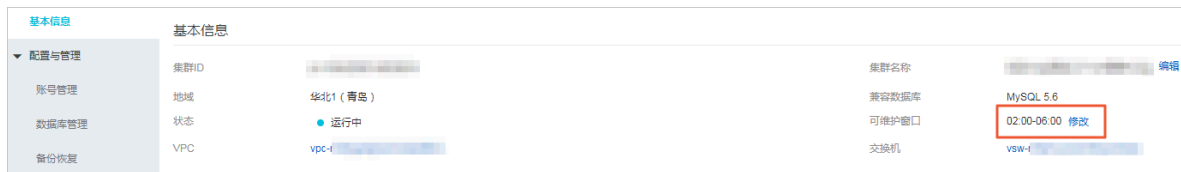
注意事项

- 在进行正式维护前，POLARDB 会给阿里云账号中设置的联系人发送短信和邮件，请注意查收。
- 集群维护当天，为保障整个维护过程的稳定性，集群会在所设置的可维护窗口之前的一段时间，进入集群维护中的状态，当集群处于该状态时，数据库本身正常的数据库访问不会受到任何影响，但该集群的控制台上，除了账号管理、数据库管理和添加 IP 白名单外，其他涉及变更类的功能均无法使用（如常用的升降级、重启等操作均无法重启），查询类如性能监控等可以正常查阅。
- 在进入集群所设置的可维护窗口后，集群会在该段时间内发生1到2次的连接闪断，请确保您的应用程序具有重连机制。闪断后，集群即可恢复到正常状态。

操作步骤

1. 登录[POLARDB控制台](#)。

2. 在控制台左上角，选择集群所在地域。
3. 单击目标集群ID。
4. 在**基本信息**中的**可维护窗口**后，单击**修改**。



5. 在编辑**可维护窗口**中，选择集群的可维护窗口，单击**提交**。

相关API

API	描述
CreateDBCluster	创建数据库集群。
ModifyDBClusterMaintainTime	修改集群可运维时间。

14.4 重启节点

本文介绍当数据库节点出现连接数满或性能问题时，如何手动重启节点。

背景信息

当数据库节点出现连接数满或性能问题时，您可以手动重启节点。重启节点会造成连接中断，重启前请做好业务安排，谨慎操作。

操作步骤

1. 登录[PolarDB控制台](#)。
2. 在控制台左上角，选择集群所在地域。
3. 单击目标集群ID。
4. 在**基本信息**页面下方**节点信息**里找到需要重启的节点。

The screenshot shows the 'Node Information' (节点信息) table. At the top, there are buttons for 'Add Node' (增加节点) and 'Reduce Node' (减少节点). The table has the following columns: Node Name (节点名称), Region (可用区), Status (状态), Current Role (当前角色), Configuration (规格), Max Connections (最大连接数), Max IOPS (最大IOPS), and Actions (操作). There are three rows of nodes, all in 'Running' (运行中) status. The 'Restart' (重启) button in the 'Actions' column for each node is highlighted with a red box.

节点名称	可用区	状态	当前角色	规格	最大连接数	最大IOPS	操作
pi-xxxxxx	杭州 可用区G	运行中	主节点	4核 32GB	5000	36000	重启
pi-xxxxxx	杭州 可用区G	运行中	只读节点	4核 32GB	5000	36000	重启
pi-xxxxxx	杭州 可用区G	运行中	只读节点	4核 32GB	5000	36000	重启

5. 单击**操作**栏中的**重启**。
6. 在弹出的对话框中，单击**确认**。

相关API

API	描述
#unique_216	重启数据库节点。

14.5 待处理事件

当PolarDB出现待处理事件时，会在控制台提醒您及时处理。

前提条件

您有未处理的运维事件。



说明：

当您有未处理的运维事件时，可以在控制台左侧导航栏的**待处理事件**看到提醒。



背景信息

PolarDB运维事件（数据库软件升级、硬件维护与升级）除了在短信、语音、邮件或站内信通知之外，还会在控制台进行通知。您可以查看具体的事件类型、任务ID、集群名称、切换时间等，也可以手动修改切换时间。

修改切换时间

1. 登录[PolarDB控制台](#)。

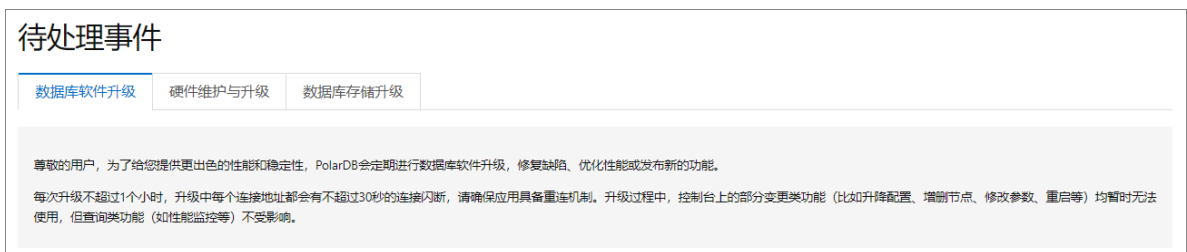
2. 在左侧导航栏中，单击待处理事件。



说明：

强制要求预约时间的运维事件会弹窗提醒，请尽快完成预约。

3. 在待处理事件页面，选择相应的事件类型。



说明：

不同的事件类型页面会显示不同的通知信息。

4. 在下方事件列表查看事件的详细信息，如需修改切换时间，请在左侧勾选对应的实例，然后单击修改切换时间，在弹出的对话框中设置时间并单击确认。



说明：

切换时间不能晚于最晚操作时间。

历史事件

您可以在PolarDB控制台里查看已完成的事件。

1. 登录PolarDB控制台。
2. 在左侧导航栏中，单击**历史事件**。

云数据库PolarDB 全球数据库网络 (GDN) 集群列表 集群回收站 待处理事件 历史事件	历史事件					
	数据库软件升级 硬件维护与升级 数据库存储升级					
	任务ID	集群名称	兼容性	开始时间	切换时间	最终开始时间
	118261	pc-t...	100% 兼容 MySQL 5.6	2019年10月22日 02:10:00	2019年10月22日 02:10:00	2019年10月23日 23:59:59
	118260	pc-b...	100% 兼容 MySQL 5.6	2019年10月22日 02:00:00	2019年10月22日 02:00:00	2019年10月23日 23:59:59
	118258	pc-b...	100% 兼容 MySQL 5.6	2019年10月22日 02:20:00	2019年10月22日 02:20:00	2019年10月23日 23:59:59

3. 在**历史事件**页面，选择相应的事件类型。不同的事件类型页面会显示不同的信息。

14.6 标签

14.6.1 绑定标签

在集群数量较多的情况下，您可以创建多个标签，为集群绑定不同的标签对其进行分类，之后通过标签进行集群筛选。

注意事项

- 标签由一对键值组成，键在同账号同地域下唯一，值无此限制。
- 每个集群最多可以绑定10个标签。相同的标签键会被覆盖。
- 不同地域的标签信息是独立的。

操作步骤

1. 登录PolarDB控制台。
2. 在控制台左上角，选择集群所在地域。
3. 在**集群列表**页面，将鼠标移动到目标集群**标签**栏的  图标上。

4. 单击**编辑标签**。

集群列表 POLARDB简介								
创建新集群	购买存储包	集群ID	请输入	Q	标签		刷新	上
集群名称	状态	兼容性	节点数	主节点配置	已使用数据	付费类型	标签	操作
pc-t...	运行中	100% 兼容 MySQL 5.6	2	4核 16GB	2.34 GB	按小时付费 2019年8月21日 15:28:15 创建	暂无标签 编辑标签	升降配 增删节点
pc-b... pc-b...	运行中	100% 兼容 MySQL 8.0	2	4核 16GB	3.71 GB	按小时付费 2019年8月5日 13:51:42 创建	升降配 增删节点	
pc-t... pc-b...	运行中	100% 兼容 MySQL 5.6	3	2核 4GB	2.83 GB	包年包月 2020年1月15日 00:00:00 到期	升降配 增删节点	

5. 在编辑标签对话框，单击**新建标签**或**已有标签**。

- **新建标签：**

设置标签的**键**和**值**后，单击**确定**。

编辑标签

注：每个集群最多可绑定 10 个标签，单次操作绑定/解绑标签的数量分别不能超过 5 个。

绑定： 已有标签 键： key 值： val 确定 取消

确定 取消



说明：

完成标签的新建后，您可以在其它实例中绑定已创建的标签。

- **已有标签：**

单击需要绑定的**标签键**即可。

编辑标签

注：每个集群最多可绑定 10 个标签，单次操作绑定/解绑标签的数量分别不能超过 5 个。

绑定： 已有标签 新建标签

标签键

te... 1

a...

a...

te... 2

确定 取消

6. 重复以上步骤，完成所有标签的绑定，之后单击对话框右下角的**确定**。

相关API

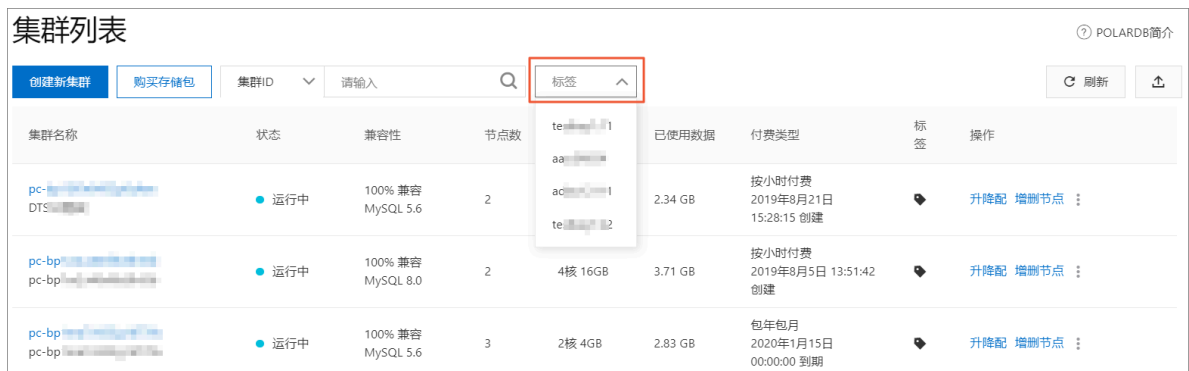
API	说明
#unique_218	该接口用于为POLARDB集群绑定标签。

14.6.2 根据标签筛选集群

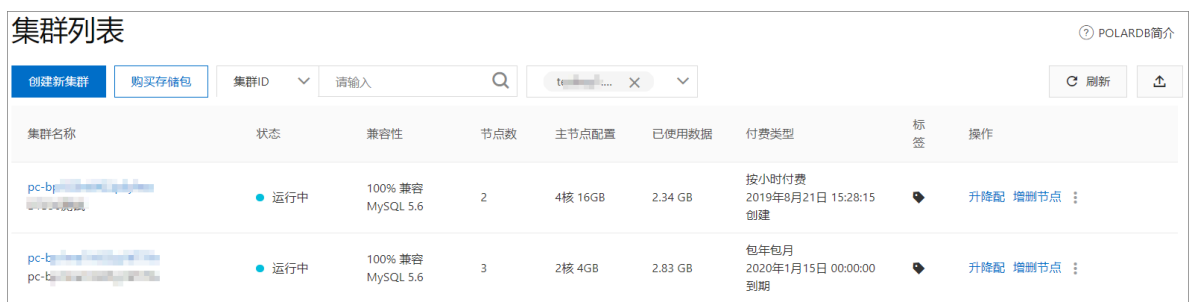
为POLARDB集群绑定标签后，您可以在集群列表中通过标签进行筛选，找出绑定指定标签的集群。

操作步骤

1. 登录[PolarDB控制台](#)。
2. 在控制台左上角，选择集群所在地域。
3. 在**集群列表**页面，单击**标签**，选择目标标签。



4. 选择指定标签后，**集群列表**将显示所有绑定该标签的集群。



相关API

API	说明
#unique_219	该接口用于查询一个或多个POLARDB集群已绑定的标签列表，或者查询一个或多个标签绑定的POLARDB集群列表。

14.6.3 查看集群绑定的标签

您可以在集群列表中查看POLARDB集群绑定了哪些标签。

操作步骤

1. 登录[PolarDB控制台](#)。
2. 在控制台左上角，选择集群所在地域。
3. 在页面，将鼠标移动到目标集群栏的  图标上。

4. 查看目标集群绑定的标签，如下图所示。

集群列表								
创建新集群 购买存储包		集群ID	请输入	Q	标签	刷新 上传		
集群名称	状态	兼容性	节点数	主节点配置	已使用数据	付费类型	标签	操作
pc-bp- 	运行中	100% 兼容 MySQL 5.6	2	4核 16GB	2.34 GB	按小时付费 2019年8月21日 15:28:15 创建	 te- 1111 编辑标签	升降配 增删节点 更多
pc-bj- 	运行中	100% 兼容 MySQL 8.0	2	4核 16GB	3.71 GB	按小时付费 2019年8月5日 13:51:42 创建		升降配 增删节点 更多

相关API

API	说明
#unique_219	该接口用于查询一个或多个POLARDB集群已绑定的标签列表，或者查询一个或多个标签绑定的POLARDB集群列表。

14.6.4 解绑标签

当集群不再需要某个标签时，您可以将该标签从POLARDB集群上解绑。

注意事项

标签从一个集群解绑后，如果没有绑定到其它集群，则该标签自动被删除。

操作步骤

1. 登录[PolarDB控制台](#)。
2. 在控制台左上角，选择集群所在地域。
3. 在页面，将鼠标移动到目标集群栏的  图标上。

4. 单击。

集群列表

创建新集群 购买存储包 集群ID 请输入 标签 刷新 上传

集群名称	状态	兼容性	节点数	主节点配置	已使用数据	付费类型	标签	操作
pc-l...	运行中	100% 兼容 MySQL 5.6	2	4核 16GB	2.34 GB	按小时付费 2019年8月21日 15:28:15 创建	暂无标签	升降配 增删节点
pc-b...	运行中	100% 兼容 MySQL 8.0	2	4核 16GB	3.71 GB	按小时付费 2019年8月5日 13:51:42 创建	编辑标签	升降配 增删节点
pc-b...	运行中	100% 兼容 MySQL 5.6	3	2核 4GB	2.83 GB	包年包月 2020年1月15日 00:00:00 到期		升降配 增删节点

5. 在编辑标签对话框，单击目标标签右侧的 图标。

编辑标签

te: ... X a: ... X ac: ... 1... X

注：每个集群最多可绑定 10 个标签，单次操作绑定/解绑标签的数量分别不能超过 5 个。

绑定： 已有标签 新建标签

确定 取消

6. 单击确定。



说明：

在一个集群中解绑标签不影响其它绑定了该标签的集群。

相关API

API	说明
#unique_220	该接口用于将标签从POLARDB集群上解绑。

15 内核功能

15.1 PolarDB MySQL内核兼容性说明

PolarDB完全兼容MySQL 5.6和MySQL 8.0，您可以在不修改应用程序任何代码和配置的情况下，将MySQL数据库迁移至PolarDB。本文将为您介绍PolarDB MySQL的内核兼容性。

- 兼容ANSI/ISO SQL标准，PolarDB MySQL 5.6/8.0支持修改SQL兼容模式为ANSI，您可以将集群参数sql_mode修改为ANSI实现该需求。如何修改集群参数请参见[设置集群参数](#)。
- 支持0至3.51版本的ODBC。
- 支持W3C和XPath标准的XML功能。
- PolarDB MySQL 8.0支持RFC 7159和ECMAScript标准（ECMA-262）的原生JSON数据类型。

15.2 小版本升级

PolarDB MySQL支持手动升级内核小版本，内核小版本的升级涉及性能提升、新功能或问题修复等。

注意事项

- 升级内核小版本会重启集群，请您尽量在业务低峰期执行升级操作，或确保您的应用有自动重连机制。
- 升级内核小版本后无法降级。

操作步骤

1. 登录[PolarDB控制台](#)。

2. 在页面左上角，选择集群所在地域。



3. 找到目标集群，单击集群ID。
4. 在基本信息页面中单击升级到最新版本。



说明：

如果您的集群内核小版本已经是最新版本，将不会显示升级到最新版本按钮。



5. 在升级到最新版本对话框中单击确定。



说明：

升级过程有60s左右的连接闪断，请确保业务有重连机制。

15.3 开启Binlog

本文介绍如何为PolarDB集群开启Binlog功能。

前提条件

集群创建于2019年4月5日之后。如果是在2019年4月5日之前创建的集群，暂时需要[提交工单](#)进行小版本升级，之后即可在控制台手动开启Binlog。

背景信息

PolarDB是一款完全兼容MySQL的云原生数据库，默认使用了更高级别的物理日志代替Binlog，但为了更好地与MySQL生态融合，PolarDB上线了开启Binlog的功能，开启Binlog后您就可以连接ElasticSearch、AnalyticDB等数据产品，也可以搭建PolarDB到RDS、RDS到PolarDB或PolarDB之间的数据实时同步。

收费说明

Binlog的空间属于集群存储空间的一部分，需要收取[存储费用](#)。

注意事项

- 开启后Binlog默认保存2周，超出两周的Binlog文件会被自动删除。您可以修改参数`loose_expire_logs_hours`（取值范围0~2376，单位：小时）以设置Binlog的保存时长。0表示不自动删除Binlog文件。
- Binlog功能默认关闭，开启Binlog需要重启实例，会造成连接中断，重启前请做好业务安排，谨慎操作。
- 开启Binlog后会导致写性能下降，读性能不受影响。
- 拉取、订阅或同步Binlog（例如使用DTS等工具）时，建议使用PolarDB的**主地址**，因为直接指向生成Binlog的主节点，具有更好的兼容性和稳定性。您可以在**基本信息**页面查看主地址，如下图所示。



开启Binlog的操作步骤

1. 登录[PolarDB控制台](#)。
2. 选择地域。
3. 找到目标集群，单击**集群名称**列的集群ID。
4. 在左侧导航栏中选择**配置与管理 > 参数配置**。

5. 搜索loose_polar_log_bin，修改当前值，并单击提交修改。



说明：

- 数据库引擎为MySQL 5.6的PolarDB集群，当前值修改为ON_WITH_GTID；
- 数据库引擎为MySQL 8.0的PolarDB集群，当前值修改为ON。

6. 在右侧提示重启的对话框中单击确定。



说明：

如果报错提示Custins minor version does not support current action，请[提交工单](#)开启。

常见问题

- Binlog能保存多久？

答：Binlog默认保存2周，超出两周的Binlog文件会被自动删除。您可以修改参

数`loose_expire_logs_hours`（取值范围0~2376，单位：小时）以设置Binlog的保存时长。0表示不自动删除Binlog文件。

- 开启Binlog后可以关闭吗？

答：将参数`loose_polar_log_bin`修改为OFF并提交即可关闭。关闭后已有的Binlog不会被删除。

- 开启Binlog对性能影响有多大？

答：测试数据显示，在64线程并发下，开启Binlog后会有30%~40%的写性能衰减（性能衰减随并发增加而减少，后续将会持续优化），读性能不受影响。对于读多写少的业务场景，整体数据库性能影响较小，例如读写比是4:1的数据库，整体性能影响大约是10%。

15.4 Thread Pool

为了发挥出PolarDB MySQL的最佳性能，阿里云提供线程池（Thread Pool）功能，将线程和会话分离，在拥有大量会话的同时，只需要少量线程完成活跃会话的任务即可。

优势

MySQL默认的线程使用模式是会话独占模式，每个会话都会创建一个独占的线程。当大量的会话存在时，会导致大量的资源竞争，大量的系统线程调度和缓存失效也会导致性能急剧下降。

阿里云PolarDB的线程池实现了不同类型SQL操作的优先级及并发控制机制，将连接数始终控制在最佳连接数附近，使PolarDB数据库在高连接大并发情况下始终保持高性能。线程池的优势如下：

- 当大量线程并发工作时，线程池会自动调节并发的线程数量在合理的范围内，从而避免线程调度工作过多和大量缓存失效。
- 大量的事务并发执行时，线程池会将语句和事务分为不同的优先级，分别控制语句和事务的并发数量，从而减少资源竞争。
- 线程池给予管理类的SQL语句更高的优先级，保证这些语句优先执行。这样在系统负载很高时，新建连接、管理、监控等操作也能够稳定执行。
- 线程池给予复杂查询SQL语句相对较低的优先级，并且有最大并发数的限制。这样可以避免过多的复杂SQL语句将系统资源耗尽，导致整个数据库服务不可用。

前提条件

集群版本为PolarDB MySQL 5.6或PolarDB MySQL 8.0。

使用Thread Pool

Thread Pool设计了以下七个参数，您可以在控制台进行修改。具体请参见[设置集群参数](#)。

参数名称	说明
loose_thread_pool_enabled	是否开启线程池功能。取值： • ON • OFF 默认值：OFF。  说明： 开启或关闭线程池功能无需重启实例。

参数名称	说明
loose_thread_pool_high_prio_mode	线程池高优先级队列模式。取值： - transactions：已开启事务的SQL将被加入高优先级队列，并被赋予thread_pool_high_prio_tickets这个门票，接下来执行的SQL都将被放入高优先级队列，直到门票被耗尽。 - statements：所有SQL都会加入高优先级队列。 - none：所有SQL都不加入高优先级队列。 默认值：transactions。  说明： 仅PolarDB MySQL 5.6支持该参数。
loose_thread_pool_high_prio_tickets	高优先级队列的单次门票数。 取值：0~4294967295。 默认值：4294967295。  说明： 仅PolarDB MySQL 5.6支持该参数。
loose_thread_pool_idle_timeout	释放线程池空闲线程的时间阈值，超过此时间，没有服务任何请求的空闲线程将被释放。 取值：0~31536000。 单位：秒，默认值：60。  说明： 仅PolarDB MySQL 5.6支持该参数。
loose_thread_pool_max_threads	线程池中允许的最大工作线程数。 取值：1~100000。 默认值：100000。  说明： 仅PolarDB MySQL 5.6支持该参数。

参数名称	说明
loose_thread_pool_oversubscribe	每个组中允许的活跃线程的数量。 活跃线程是指正在执行SQL语句的线程，但是不包括以下两种情形： - SQL语句在等待磁盘I/O。 - SQL语句在等待事务提交。 取值：1~1000。 默认值：10。
loose_thread_pool_stall_limit	判断线程池进入拥塞状态的时间阈值。 当线程池进入拥塞状态时，系统会创建新的线程来服务SQL。 取值：1~18446744073709551615。 单位：毫秒。默认值：30。  说明： 仅PolarDB MySQL 5.6支持该参数。

查询Thread Pool状态

您可以通过如下命令查询Thread Pool状态：

```
select * from information_schema.THREAD_POOL_STATUS;
```

返回示例如下所示。

```
mysql>select * from information_schema.THREAD_POOL_STATUS;
+-----+-----+-----+-----+
+-----+-----+-----+-----+
+-----+-----+-----+-----+
| ID      | THREAD_COUNT | ACTIVE_THREAD_COUNT | WAITING_THREAD_COUNT |
| DUMP_THREAD_COUNT | CONNECTION_COUNT | LOW_QUEUE_COUNT |
| HIGH_QUEUE_COUNT |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
+-----+-----+-----+-----+
| 1 | 0 | 2 | 1 | 0 | 0 |
| 0 | 1 | 2 | 0 | 0 | 0 |
| 1 | 2 | 2 | 0 | 0 | 0 |
| 1 | 3 | 2 | 0 | 0 | 0 |
| 1 | 4 | 2 | 0 | 0 | 0 |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
+-----+-----+-----+-----+
```

返回行数：[5]，耗时：7 ms.

参数说明如下。

参数	说明
ID	线程池ID。
THREAD_COUNT	线程池中的线程数。
ACTIVE_THREAD_COUNT	线程池中的活跃线程数。
WAITING_THREAD_COUNT	线程池中正在等待磁盘I/O、事务提交的线程数。
DUMP_THREAD_COUNT	线程池中DUMP类长连接数量。
CONNECTION_COUNT	线程池中已建立的用户连接数。
LOW_QUEUE_COUNT	线程池中低优先级队列中等待的请求数。
HIGH_QUEUE_COUNT	线程池中高优先级队列中等待的请求数。

Sysbench测试

如下是开启线程池和不开启线程池的性能对比。从测试结果可以看出线程池在高并发的情况下有着明显的性能优势。

图 15-1: OLTP无索引更新测试

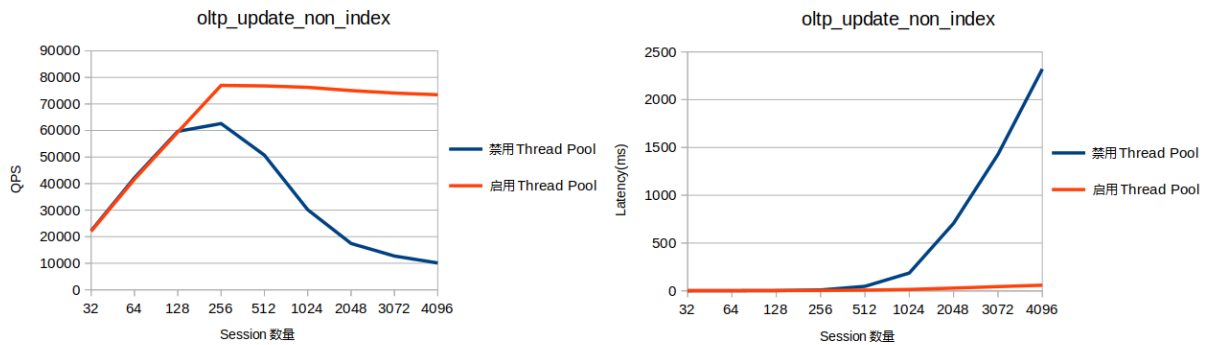


图 15-2: OLTP只写测试

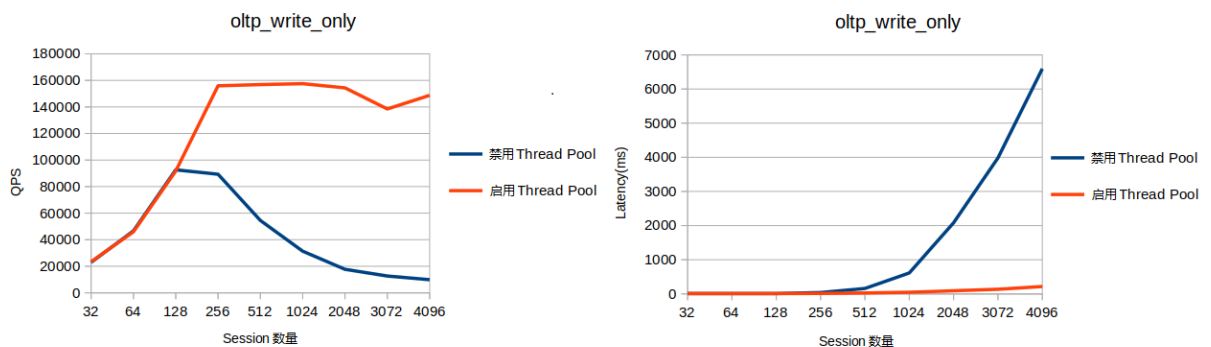


图 15-3: OLTP只读测试

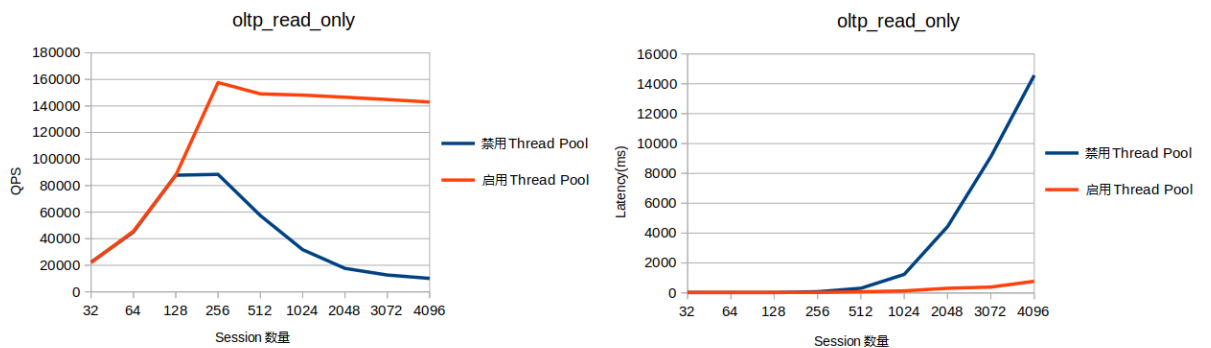
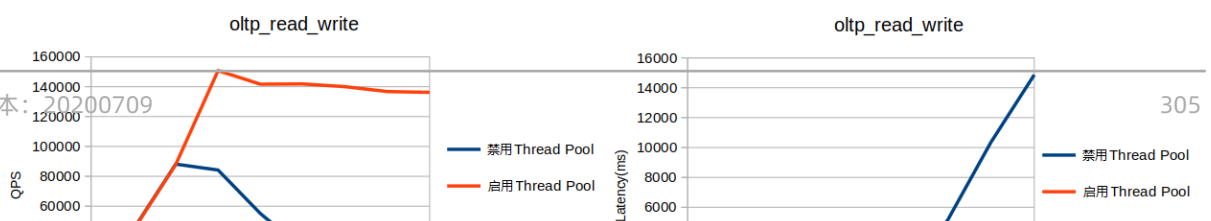


图 15-4: OLTP读写测试



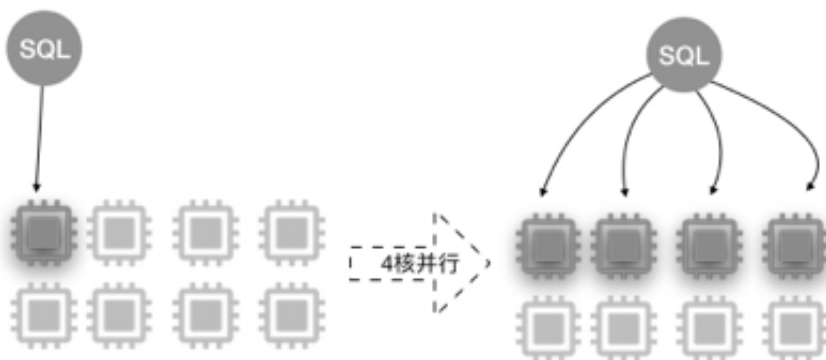
15.5 并行查询

15.5.1 并行查询

PolarDB MySQL 8.0重磅推出并行查询框架，当您的查询数据量到达一定阈值，就会自动启动并行查询框架，从而使查询耗时指数级下降。

在存储层将数据分片到不同的线程上，多个线程并行计算，将结果流水线汇总到总线程，最后总线程做些简单归并返回给用户，提高查询效率。

并行查询（Parallel Query）利用多核CPU的并行处理能力，以8核 32G配置为例，示意图如下所示。



应用场景

并行查询适用于大部分SELECT语句，例如大表查询、多表连接查询、计算量较大的查询。对于非常短的查询，效果不太显著。

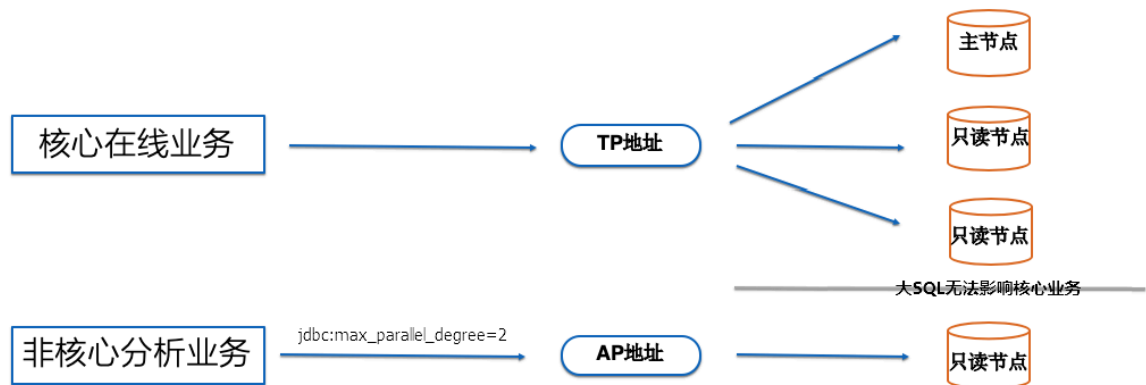
- 轻分析类业务

报表查询通常SQL复杂而且比较耗费时间，通过并行查询可以加速单次查询效率。

- 系统资源相对空闲

并行查询会使用更多的系统资源，只有当系统的CPU较多、IO负载不高、内存够大的时候，才可以充分使用并行查询来提高资源利用率和查询效率。

- 在离线混合场景



不同的业务用不同的连接地址，使用不同的数据库节点，避免互相影响。您可以在只读地址（AP地址）上开启并行查询。

并行查询的使用方法

- 通过系统参数来控制并行查询

PolarDB通过Global全局参数max_parallel_degree来控制每一条SQL最多使用多少个线程并行执行，默认值是0，您可以在使用过程中随时修改该参数，无需重启数据库，具体操作请参见[设置集群参数](#)。

并行查询推荐设置以及相关说明如下：

- 并行度参数从低到高，逐渐增加，建议不要超过CPU核数的四分之一。集群CPU核数大于等于8才支持开启并行查询，小规模集群不建议开启。例如，刚开始使用并行查询时，设置

max_parallel_degree参数为2，试运行一天后，如果CPU压力不大，可以往上增加；如果CPU压力较大，停止增加。

- max_parallel_degree参数为0时，表示关闭并行查询；max_parallel_degree参数为1时，表示开启并行查询，但并行度只有1。
- 并行度参数max_parallel_degree为了保持兼容性以及MySQL配置文件的约定俗成，阿里云在控制台上增加了前缀loose，因此参数名称是loose_max_parallel_degree，这样保证其他版本接受该参数时也不会报错，详情请参见[Program Option Modifiers](#)。
- 打开并行查询功能时，需要设置innodb_adaptive_hash_index参数为OFF，innodb_adaptive_hash_index参数会影响并行查询的性能。

除了Global集群级别，您也可以单独调整某Session内SQL查询的并行度。例如，通过Session级环境变量，加到JDBC的连接串配置中，则可以对某个应用程序单独设置并行度。

```
set max_parallel_degree = n
```

- 使用Hint

使用Hint语法可以对单个语句进行控制，例如系统默认关闭并行查询情况下，但需要对某个高频的慢SQL查询进行加速，此时就可以使用Hint对特定SQL进行加速。

开启并行查询（以下任意一种方式）：

```
SELECT /*+PARALLEL(x)*/ ... FROM ...; -- x > 0  
SELECT /*+ SET_VAR(max_parallel_degree=n) */ * FROM ... // n > 0
```

关闭并行查询（以下任意一种方式）：

```
SELECT /*+NO_PARALLEL()*/ ... FROM ...;  
SELECT /*+ SET_VAR(max_parallel_degree=0) */ * FROM ...
```

Hint高级用法

并行查询提供了PARALLEL和NO_PARALLEL两种Hint。

- 通过PARALLEL Hint可以强制查询并行执行，同时可以指定并行度和并行扫描的表。
- 通过NO_PARALLEL Hint可以强制查询串行执行，或者指定不选择某些表作为并行扫描的表。

Hint语法如下所示：

```
/*+ PARALLEL ([ [query_block] [table_name] [degree] ]) */
```

```
/*+ NO_PARALLEL [( [query_block] [table_name][, table_name] )] */
```

其中参数说明如下所示。

参数	说明
query_block	应用Hint的query block名称。
table_name	应用Hint的表名称。
degree	并行度。

示例：

```
SELECT /*+PARALLEL()*/ * FROM t1, t2;
-- 设置force_parallel_mode为true（表示当表记录数小于20000行时，依旧会强制并行），
-- 并行度用系统默认max_parallel_degree，如果max_parallel_degree > 0,
-- 则打开并行，如果max_parallel_degree等于0时，依旧时关闭并行

SELECT /*+PARALLEL(8)*/ * FROM t1, t2;
-- 强制并行度8并行执行，
-- 并设置force_parallel_mode为true（表示当表记录数小于20000行时，依旧会强制并行），
-- 并行度设置max_parallel_degree为8

SELECT /*+ SET_VAR(max_parallel_degree=8) */ * FROM ...
-- 设置并行度max_parallel_degree为8,
-- 但force_parallel_mode为false（表示当表记录数小于20000行时会自动关闭并行）

SELECT /*+PARALLEL(t1)*/ * FROM t1, t2;
-- 选择t1表并行，对t1表执行/*+PARALLEL()*/ 语法

SELECT /*+PARALLEL(t1 8)*/ * FROM t1, t2;
-- 强制并行度8且选择t1表并行执行，对t1表执行/*+PARALLEL(8)*/语法

SELECT /*+PARALLEL(@subq1)*/ SUM(t.a) FROM t WHERE t.a =
(SELECT /*+QB_NAME(subq1)*/ SUM(t1.a) FROM t1);
--强制subquery并行执行，并行度用系统默认max_parallel_degree,
-- 如果max_parallel_degree > 0, 则打开并行，max_parallel_degree等于0时，依旧时关闭并行

SELECT /*+PARALLEL(@subq1 8)*/ SUM(t.a) FROM t WHERE t.a =
(SELECT /*+QB_NAME(subq1)*/ SUM(t1.a) FROM t1);
--强制subquery并行执行，并行度设置max_parallel_degree为8

SELECT SUM(t.a) FROM t WHERE t.a =
(SELECT /*+PARALLEL()*/ SUM(t1.a) FROM t1);
--强制subquery并行执行，
-- 并行度用系统默认max_parallel_degree,
-- 如果max_parallel_degree > 0, 则打开并行，max_parallel_degree等于0时，依旧时关闭并行

SELECT SUM(t.a) FROM t WHERE t.a =
(SELECT /*+PARALLEL(8)*/ SUM(t1.a) FROM t1);
--强制subquery并行执行，设置并行度max_parallel_degree为8

SELECT /*+NO_PARALLEL()*/ * FROM t1, t2;
-- 禁止并行执行
```

```

SELECT /*+NO_PARALLEL(t1)*/ * FROM t1, t2;
-- 只对t1表禁止并行，当系统打开并行时，有可能对t2进行并行扫描，并行执行

SELECT /*+NO_PARALLEL(t1, t2)*/ * FROM t1, t2;
-- 同时对t1和t2表禁止并行

SELECT /*+NO_PARALLEL(@subq1)*/ SUM(t.a) FROM t WHERE t.a =
  (SELECT /*+QB_NAME(subq1)*/ SUM(t1.a) FROM t1);
--禁止subquery 并行执行

SELECT SUM(t.a) FROM t WHERE t.a =
  (SELECT /*+NO_PARALLEL()*/ SUM(t1.a) FROM t1);
--禁止subquery 并行执行

```

**说明：**

对于不支持并行的查询或者并行扫描的表，PARALLEL Hint不生效。

并行子查询的选择方式（并行子查询详细信息请参见[并行执行计划](#)中的子查询支持部分。）也可以通过Hint来进行控制，语法及说明如下：

```

/*+ PQ_PUSHDOWN [( [query_block])] */ 对应的子查询会选择push down的并行子查询执行策略
/*+ NO_PQ_PUSHDOWN [( [query_block])] */ 对应的子查询会选择 shared access的并行子查询执行策略

```

示例：

```

#子查询选择 push down并行策略
EXPLAIN SELECT /*+ PQ_PUSHDOWN(@qb1) */ * FROM t2 WHERE t2.a =
  (SELECT /*+ qb_name(qb1) */ a FROM t1);

#子查询选择 shared access并行策略
EXPLAIN SELECT /*+ NO_PQ_PUSHDOWN(@qb1) */ * FROM t2 WHERE t2.a =
  (SELECT /*+ qb_name(qb1) */ a FROM t1);

#不加query block进行控制
EXPLAIN SELECT * FROM t2 WHERE t2.a =
  (SELECT /*+ NO_PQ_PUSHDOWN() */ a FROM t1);

```

- 强制优化器选择并行执行

PolarDB优化器可能不选择并行执行查询，如果希望优化器忽略代价，尽可能选择并行计划，可以设置如下参数：

```
set force_parallel_mode = on
```

**说明：**

这是一个调试参数，不建议在生产环境中使用。由于并行查询使用场景的限制，有些情况下即便设置了该参数，优化器也可能不选择并行。

相关参数和变量

表 15-1: 系统参数

参数名	级别	描述
max_parallel_degree	Global、Session	单个查询的最大并行度，即最多使用多少个 worker 进行并行执行。 - 取值范围：[0-1024] - 默认值：0，表示关闭并行计算。  说明： PolarDB 优化器可能会对主查询和子查询分别并行执行，如果同时并行执行，它们的最大 worker 数不能超过 max_parallel_degree，整个查询使用的 worker 数将会是主查询和子查询使用的 worker 数之和。
force_parallel_mode	Session	强制 PolarDB 优化器忽略代价，尽可能的使用并行查询。 - 取值范围：[ON, OFF] - 默认值：OFF  注意： 这是个调试参数，打开后，PolarDB 优化器会尽可能的选择并行查询，但不能保证一定使用并行查询。

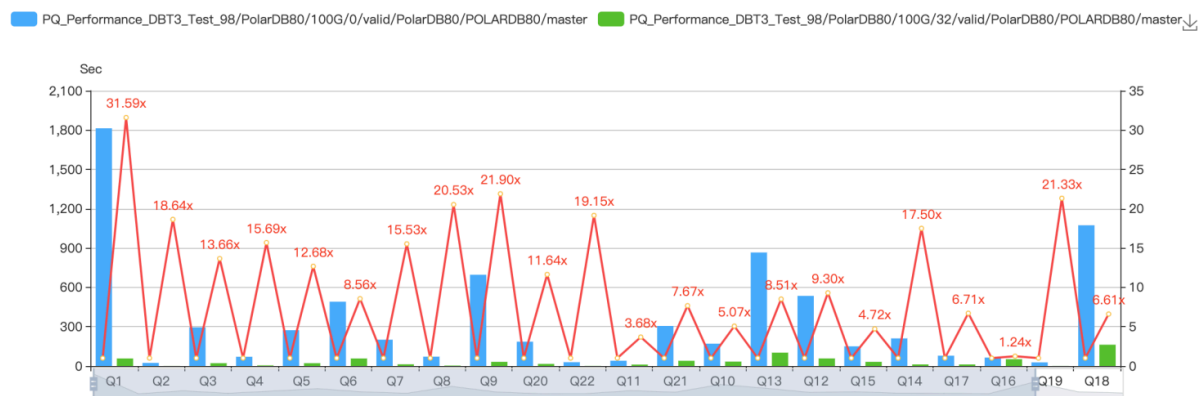
表 15-2: 状态变量

变量名	级别	描述
Parallel_workers_created	Session、Global	从 Session 启动开始，生成 Parallel Worker 的个数。
Gather_records	Session、Global	Gather 记录总数。
PQ_refused_over_memory_soft_limit	Session、Global	由于内存限制没有启用并行的查询数。
PQ_refused_over_total_workers	Session、Global	由于总 Worker 数限制没有启用并行的查询数。

变量名	级别	描述
Total_used_query_memory	Global	当前总的已使用的查询内存（Virtual Memory）。
Total_running_parallel_workers	Global	当前正在运行的Parallel Worker的数目。

性能指标

本次测试将使用TPC-H生成100G数据来测试PolarDB MySQL 8.0集群的性能指标。测试用的PolarDB集群规格为32核256GB，并行度max_parallel_degree分别设置为32和0，具体测试步骤请参见[#unique_224](#)。



通过以上测试结果图得出结论，TPC-H中95%的SQL可以被加速，70%的SQL加速比超过8倍。

并行执行计划

以下为您介绍EXPLAIN执行计划输出中与并行查询相关的内容。

- 并行扫描

在并行扫描中，每个Worker并行独立扫描数据表中的数据。Worker扫描产生的中间结果集将会返回给Leader线程，Leader线程通过Gather操作收集产生的中间结果，并将所有结果汇总返回到客户端。

- 多表并行连接

并行查询会将多表连接操作完整的下推到Worker上去执行。PolarDB优化器只会选择一个自认为最优的表进行并行扫描，而除了该表外，其他表都是一般扫描。每个Worker会将连接结果集返回给Leader线程，Leader线程通过Gather操作进行汇总，最后将结果返回给客户端。

- 并行排序

PolarDB优化器会根据查询情况，将ORDER BY下推到每个Worker里执行，每个Worker将排序后的结果返回给Leader，Leader通过Gather Merge Sort操作进行归并排序，最后将排序结果返回到客户端。

- 并行分组

PolarDB优化器会根据查询情况，将GROUP BY下推到Worker上去并行执行。每个Worker负责部分数据的GROUP BY。Worker会将GROUP BY的中间结果返回给Leader，Leader通过Gather操作汇总所有数据。这里PolarDB优化器会根据查询计划情况来自动识别是否需要再次在Leader上进行GROUP BY。例如，如果GROUP BY使用了Loose Index Scan，Leader上将不会进行再次GROUP BY；否则Leader会再次进行GROUP BY操作，然后把最终结果返回到客户端。

- 并行聚集

并行查询执行聚集函数下推到Worker上并行执行。并行聚集是通过两次聚集来完成的。第一次，参与并行查询部分的每个Worker执行聚集步骤。第二次，Gather或Gather Merge节点将每个Worker产生的结果汇总到Leader。最后，Leader会将所有Worker的结果再次进行聚集得到最终结果。

- 子查询支持

在并行查询下子查询有四种执行策略：

- 在Leader线程中串行执行

当子查询不可并行执行时，例如在2个表JOIN，在JOIN条件上引用了用户的function，此时子查询会在leader线程上进行串行查询。

- 在Leader上并行执行（Leader会启动另一组worker）

生成并行计划后，在Leader上执行的计划包含有支持并行执行的子查询，但这些子查询不能提前并行执行（即不能采用Shared access）。例如，当前如果子查询中包括window function就采用这样执行策略。

- Shared access

生成并行计划后，Worker的执行计划引用了可并行执行的子查询，PolarDB优化器会选择先提前并行执行这些子查询，让Worker可以直接访问这些子查询的结果。

- pushed down

生成并行计划后，Worker执行计划引用了相关子查询，这些子查询会被整体推送到worker上执行。

并行执行计划示例

以下以使用pq_test表来测试并行查询为例。

表结构如下：

```
mysql> SHOW CREATE TABLE pq_test\G
***** 1. row *****
      Table: pq_test
      Create Table: CREATE TABLE `pq_test` (
        `id` BIGINT(20) NOT NULL AUTO_INCREMENT,
        `help_topic_id` INT(10) UNSIGNED NOT NULL,
        `name` CHAR(64) NOT NULL,
        `help_category_id` SMALLINT(5) UNSIGNED NOT NULL,
        `description` TEXT NOT NULL,
        `example` TEXT NOT NULL,
        `url` TEXT NOT NULL,
        PRIMARY KEY (`id`)
      ) ENGINE=InnoDB AUTO_INCREMENT=21495809 DEFAULT CHARSET=utf8
      1 row in set (0.00 sec)
```

表大小如下：

```
mysql> SHOW TABLE STATUS\G
***** 1. row *****
      Name: pq_test
      Engine: InnoDB
      Version: 10
      Row_format: Dynamic
      Rows: 20064988
      Avg_row_length: 1898
      Data_length: 38085328896
      Max_data_length: 0
      Index_length: 0
      Data_free: 4194304
      Auto_increment: 21495809
      Create_time: 2019-07-30 01:35:27
      Update_time: NULL
      Check_time: NULL
      Collation: utf8_general_ci
      Checksum: NULL
      Create_options:
      Comment:
      1 row in set (0.02 sec)
```

查询SQL：

```
SELECT COUNT(*) FROM pq_test;
```

- 不使用并行查询的EXPLAIN输出如下：

```
mysql> SET max_parallel_degree=0; EXPLAIN SELECT COUNT(*) FROM pq_test\G
Query OK, 0 rows affected (0.02 sec)
***** 1. row *****
      Id: 1
      Select_type: SIMPLE
      Table: pq_test
      Partitions: NULL
      Type: index
```

```
Possible_keys: NULL
Key: PRIMARY
Key_len: 8
Ref: NULL
Rows: 20064988
Filtered: 100.00
Extra: Using index
1 row in set, 1 warning (0.03 sec)
```

- 使用并行查询的EXPLAIN输出如下：

```
mysql> EXPLAIN SELECT COUNT(*) FROM pq_test\G
***** 1. row *****
    id: 1
  select_type: SIMPLE
    table: &gather2&gt;
  partitions: NULL
    type: ALL
Possible_keys: NULL
    key: NULL
   key_len: NULL
    ref: NULL
   rows: 20064988
  filtered: 100.00
   extra: NULL
***** 2. row *****
    id: 2
  select_type: SIMPLE
    table: pq_test
  partitions: NULL
    type: index
Possible_keys: NULL
    key: PRIMARY
   key_len: 8
    ref: NULL
   rows: 10032494
  filtered: 100.00
   extra: Parallel scan (2 workers); Using index
2 rows in set, 1 warning (0.00 sec)
```

从该EXPLAIN可以看到，并行计划中包含Gather操作，该操作负责汇总所有Worker返回的中间结果。另外，从执行计划输出的Extra信息中看到：pq_test表使用了Parallel scan（并行扫描），期望用2个Workers来并行执行。

- 带子查询的语句使用并行查询的EXPLAIN输出如下：

```
mysql> EXPLAIN SELECT a, (select sum(t2.b) from t2 where t2.a = t1.b) FROM t1 WHERE
(a, b) IN (SELECT b, MAX(a) FROM t2 GROUP BY b)\G
***** 1. row *****
    id: 1
  select_type: PRIMARY
    table: <gather1>
  partitions: NULL
    type: ALL
Possible_keys: NULL
    key: NULL
   key_len: NULL
    ref: NULL
   rows: 2
  filtered: 100.00
   extra: NULL
```

```

***** 2. row *****
  id: 1
  select_type: PRIMARY
  table: t1
  partitions: NULL
  type: ALL
  possible_keys: NULL
  key: NULL
  key_len: NULL
  ref: NULL
  rows: 2
  filtered: 100.00
  Extra: Parallel scan (1 workers); Using where
***** 3. row *****
  id: 2
  select_type: DEPENDENT SUBQUERY
  table: t2
  partitions: NULL
  type: ALL
  possible_keys: NULL
  key: NULL
  key_len: NULL
  ref: NULL
  rows: 2
  filtered: 50.00
  Extra: Parallel pushdown; Using where
***** 4. row *****
  id: 3
  select_type: SUBQUERY
  table: <gather3>
  partitions: NULL
  type: ALL
  possible_keys: NULL
  key: NULL
  key_len: NULL
  ref: NULL
  rows: 1
  filtered: 100.00
  Extra: Shared access; Using temporary
***** 5. row *****
  id: 3
  select_type: SIMPLE
  table: t2
  partitions: NULL
  type: ALL
  possible_keys: NULL
  key: NULL
  key_len: NULL
  ref: NULL
  rows: 2
  filtered: 100.00
  Extra: Parallel scan (1 workers); Using temporary
5 rows in set, 2 warnings (0.02 sec)

```

这里对于select_type是SUBQUERY的2个子查询，Extra分别标识了Parallel pushdown和Shared access，分别表示2个子查询的并行执行策略，一个是整个下推到Worker去执行，一个是提前并行执行并Share执行结果给所有worker。

15.5.2 并行查询使用示例

本文以TPC-H为例，为您介绍并行查询使用示例。案例中所有示例，使用的数据量是TPC-H中SF=100GB，使用PolarDB节点规格为88核710G的配置在主节点进行测试。

- [GROUP BY & ORDER BY支持](#)
- [AGGREGATE函数支持 \(SUM/AVG/COUNT\)](#)
- [JOIN支持](#)
- [BETWEEN函数 & IN函数支持](#)
- [LIMIT支持](#)
- [INTERVAL函数支持](#)
- [CASE WHEN支持](#)
- [LIKE支持](#)

GROUP BY & ORDER BY支持

未开启并行查询，耗时1563.32秒，开启并行查询后，耗时只用49.65秒，提升31.48倍。

原始SQL语句，如下所示：

```
SELECT l_returnflag,
       l_linestatus,
       Sum(l_quantity) AS sum_qty,
       Sum(l_extendedprice) AS sum_base_price,
       Sum(l_extendedprice * (1 - l_discount)) AS sum_disc_price,
       Sum(l_extendedprice * (1 - l_discount) * (1 + l_tax)) AS sum_charge,
       Avg(l_quantity) AS avg_qty,
       Avg(l_extendedprice) AS avg_price,
       Avg(l_discount) AS avg_disc,
       Count(*) AS count_order
FROM   lineitem
WHERE  l_shipdate <= date '1998-12-01' - INTERVAL '93' day
GROUP BY l_returnflag,
         l_linestatus
ORDER BY l_returnflag,
         l_linestatus ;
```

未开启并行查询，耗时1563.32秒。

```
mysql> SELECT /*+ SET_VAR(max_parallel_degree=0) */ l_returnflag,
-> l_linestatus,
-> Sum(l_quantity) AS sum_qty,
-> Sum(l_extendedprice) AS sum_base_price,
-> Sum(l_extendedprice * (1 - l_discount)) AS sum_disc_price,
-> Sum(l_extendedprice * (1 - l_discount) * (1 + l_tax)) AS sum_charge,
-> Avg(l_quantity) AS avg_qty,
-> Avg(l_extendedprice) AS avg_price,
-> Avg(l_discount) AS avg_disc,
-> Count(*) AS count_order
-> FROM lineitem
-> WHERE l_shipdate <= date '1998-12-01' - INTERVAL '93' day
-> GROUP BY l_returnflag,
-> l_linestatus
-> ORDER BY l_returnflag,
-> l_linestatus ;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| l_returnflag | l_linestatus | sum_qty | sum_base_price | sum_disc_price | sum_charge | avg_qty | avg_price | avg_disc | count_order |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| A | F | 3775127758.00 | 5660776097194.45 | 5377736398183.9374 | 5592847429515.927026 | 25.499370 | 38236.116984 | 0.050002 | 148047881 |
| N | F | 98553062.00 | 147771098385.98 | 140384965965.0348 | 145999793032.775829 | 25.501557 | 38237.199389 | 0.049985 | 3864590 |
| N | O | 7421794698.00 | 11128967412861.24 | 10572526740167.5066 | 10995437028892.436612 | 25.500030 | 38237.248190 | 0.049998 | 291050427 |
| R | F | 3775724970.00 | 5661603032745.34 | 5378513563915.4097 | 5593662252666.916161 | 25.500066 | 38236.697258 | 0.050001 | 148067261 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (26 min 3.32 sec)
```

开启并行查询后，耗时只用49.65秒，提升31.48 倍。

```
mysql> SELECT /*+ SET_VAR(max_parallel_degree=32) */ l_returnflag,
->      l_linestatus,
->      Sum(l_quantity) AS sum_qty,
->      Sum(l_extendedprice) AS sum_base_price,
->      Sum(l_extendedprice * (1 - l_discount)) AS sum_disc_price,
->      Sum(l_extendedprice * (1 - l_discount) * (1 + l_tax)) AS sum_charge,
->      Avg(l_quantity) AS avg_qty,
->      Avg(l_extendedprice) AS avg_price,
->      Avg(l_discount) AS avg_disc,
->      Count(*) AS count_order
-> FROM lineitem
-> WHERE l_shipdate <= date '1998-12-01' - INTERVAL '93' day
-> GROUP BY l_returnflag,
->      l_linestatus
-> ORDER BY l_returnflag,
->      l_linestatus ;
```

l_returnflag	l_linestatus	sum_qty	sum_base_price	sum_disc_price	sum_charge	avg_qty	avg_price	avg_disc	count_order
A	F	3775127758.00	5660776097194.45	5377736398183.9374	5592847429515.927026	25.499370	38236.116984	0.050002	148047881
N	F	98553062.00	147771098385.98	140384965965.0348	145999793032.775829	25.501557	38237.199389	0.049985	3864590
N	O	7421794698.00	11128967412861.24	10572526740167.5066	10995437028892.436612	25.500030	38237.248190	0.049998	291050427
R	F	3775724970.00	5661603032745.34	5378513563915.4097	5593662252666.916161	25.500066	38236.697258	0.050001	148067261

4 rows in set (49.65 sec)

AGGREGATE函数支持 (SUM/AVG/COUNT)

未开启并行查询，耗时1563.32秒，开启并行查询后，耗时只用49.65秒，提升31.48 倍。

原始SQL语句，如下所示：

```
SELECT l_returnflag,
       l_linestatus,
       Sum(l_quantity) AS sum_qty,
       Sum(l_extendedprice) AS sum_base_price,
       Sum(l_extendedprice * (1 - l_discount)) AS sum_disc_price,
       Sum(l_extendedprice * (1 - l_discount) * (1 + l_tax)) AS sum_charge,
       Avg(l_quantity) AS avg_qty,
       Avg(l_extendedprice) AS avg_price,
       Avg(l_discount) AS avg_disc,
       Count(*) AS count_order
FROM lineitem
WHERE l_shipdate <= date '1998-12-01' - INTERVAL '93' day
GROUP BY l_returnflag,
         l_linestatus
ORDER BY l_returnflag,
         l_linestatus ;
```

未开启并行查询，耗时1563.32秒。

```
mysql> SELECT /*+ SET_VAR(max_parallel_degree=0) */ l_returnflag,
->      l_linestatus,
->      Sum(l_quantity) AS sum_qty,
->      Sum(l_extendedprice) AS sum_base_price,
->      Sum(l_extendedprice * (1 - l_discount)) AS sum_disc_price,
->      Sum(l_extendedprice * (1 - l_discount) * (1 + l_tax)) AS sum_charge,
->      Avg(l_quantity) AS avg_qty,
->      Avg(l_extendedprice) AS avg_price,
->      Avg(l_discount) AS avg_disc,
->      Count(*) AS count_order
-> FROM lineitem
-> WHERE l_shipdate <= date '1998-12-01' - INTERVAL '93' day
-> GROUP BY l_returnflag,
->      l_linestatus
-> ORDER BY l_returnflag,
->      l_linestatus ;
```

l_returnflag	l_linestatus	sum_qty	sum_base_price	sum_disc_price	sum_charge	avg_qty	avg_price	avg_disc	count_order
A	F	3775127758.00	5660776097194.45	5377736398183.9374	5592847429515.927026	25.499370	38236.116984	0.050002	148047881
N	F	98553062.00	147771098385.98	140384965965.0348	145999793032.775829	25.501557	38237.199389	0.049985	3864590
N	O	7421794698.00	11128967412861.24	10572526740167.5066	10995437028892.436612	25.500030	38237.248190	0.049998	291050427
R	F	3775724970.00	5661603032745.34	5378513563915.4097	5593662252666.916161	25.500066	38236.697258	0.050001	148067261

4 rows in set (26 min 3.32 sec)

开启并行查询后，耗时只用49.65秒，提升31.48 倍。


```
mysql> SELECT /*+ SET_VAR(max_parallel_degree=32) */ l_returnflag,
-> l_linestatus,
-> Sum(l_quantity) AS sum_qty,
-> Sum(l_extendedprice) AS sum_base_price,
-> Sum(l_extendedprice * (1 - l_discount)) AS sum_disc_price,
-> Sum(l_extendedprice * (1 - l_discount) * (1 + l_tax)) AS sum_charge,
-> Avg(l_quantity) AS avg_qty,
-> Avg(l_extendedprice) AS avg_price,
-> Avg(l_discount) AS avg_disc,
-> Count(*) AS count_order
-> FROM lineitem
-> WHERE l_shipdate <= date '1998-12-01' - INTERVAL '93' day
-> GROUP BY l_returnflag,
-> l_linestatus
-> ORDER BY l_returnflag,
-> l_linestatus ;
```

l_returnflag	l_linestatus	sum_qty	sum_base_price	sum_disc_price	sum_charge	avg_qty	avg_price	avg_disc	count_order
A	F	3775127758.00	5660776097194.45	5377736398183.9374	5592847429515.927026	25.499370	38236.116984	0.050002	148047881
N	F	98553062.00	147771098385.98	140384965965.0348	145999793032.775829	25.501557	38237.199389	0.049985	3864590
N	O	7421794698.00	11128967412861.24	10572526740167.5066	10995437028892.436612	25.500030	38237.248190	0.049998	291050427
R	F	3775724970.00	5661603032745.34	5378513563915.4097	5593662252666.916161	25.500066	38236.697258	0.050001	148067261

4 rows in set (49.65 sec)

JOIN支持

未开启并行查询，耗时21.73秒，开启并行查询后，耗时1.37秒，提升15.86倍。

原始SQL语句，如下所示：

```
select sum(l_extendedprice*(1 - l_discount)) as revenue
from lineitem, part
where ( p_partkey = l_partkey and p_brand = 'Brand#12'
      and p_container in ('SM CASE', 'SM BOX', 'SM PACK', 'SM PKG')
      and l_quantity >= 6 and l_quantity <= 6 + 10
      and p_size between 1 and 5
      and l_shipmode in ('AIR', 'AIR REG')
      and l_shipinstruct = 'DELIVER IN PERSON' )
or ( p_partkey = l_partkey and p_brand = 'Brand#13'
      and p_container in ('MED BAG', 'MED BOX', 'MED PKG', 'MED PACK')
      and l_quantity >= 10 and l_quantity <= 10 + 10
      and p_size between 1 and 10
      and l_shipmode in ('AIR', 'AIR REG')
      and l_shipinstruct = 'DELIVER IN PERSON' )
or ( p_partkey = l_partkey and p_brand = 'Brand#24'
      and p_container in ('LG CASE', 'LG BOX', 'LG PACK', 'LG PKG')
      and l_quantity >= 21 and l_quantity <= 21 + 10
      and p_size between 1 and 15
      and l_shipmode in ('AIR', 'AIR REG')
      and l_shipinstruct = 'DELIVER IN PERSON' );
```

未开启并行查询，耗时21.73秒。

```
mysql> select /*+ SET_VAR(max_parallel_degree=0) */ sum(l_extendedprice* (1 - l_discount)) as revenue
-> from lineitem, part
-> where ( p_partkey = l_partkey and p_brand = 'Brand#12'
->         and p_container in ('SM CASE', 'SM BOX', 'SM PACK', 'SM PKG')
->         and l_quantity >= 6 and l_quantity <= 6 + 10
->         and p_size between 1 and 5
->         and l_shipmode in ('AIR', 'AIR REG')
->         and l_shipinstruct = 'DELIVER IN PERSON' )
-> or ( p_partkey = l_partkey and p_brand = 'Brand#13'
->         and p_container in ('MED BAG', 'MED BOX', 'MED PKG', 'MED PACK')
->         and l_quantity >= 10 and l_quantity <= 10 + 10
->         and p_size between 1 and 10
->         and l_shipmode in ('AIR', 'AIR REG')
->         and l_shipinstruct = 'DELIVER IN PERSON' )
-> or ( p_partkey = l_partkey and p_brand = 'Brand#24'
->         and p_container in ('LG CASE', 'LG BOX', 'LG PACK', 'LG PKG')
->         and l_quantity >= 21 and l_quantity <= 21 + 10
->         and p_size between 1 and 15
->         and l_shipmode in ('AIR', 'AIR REG')
->         and l_shipinstruct = 'DELIVER IN PERSON' );

+-----+
| revenue |
+-----+
| 317693789.3851 |
+-----+
1 row in set (21.73 sec)
```

开启并行查询后，耗时1.37秒，提升15.86倍。

```
mysql> select /*+ SET_VAR(max_parallel_degree=32) */ sum(l_extendedprice* (1 - l_discount)) as revenue
-> from lineitem, part
-> where ( p_partkey = l_partkey and p_brand = 'Brand#12'
->         and p_container in ('SM CASE', 'SM BOX', 'SM PACK', 'SM PKG')
->         and l_quantity >= 6 and l_quantity <= 6 + 10
->         and p_size between 1 and 5
->         and l_shipmode in ('AIR', 'AIR REG')
->         and l_shipinstruct = 'DELIVER IN PERSON' )
-> or ( p_partkey = l_partkey and p_brand = 'Brand#13'
->         and p_container in ('MED BAG', 'MED BOX', 'MED PKG', 'MED PACK')
->         and l_quantity >= 10 and l_quantity <= 10 + 10
->         and p_size between 1 and 10
->         and l_shipmode in ('AIR', 'AIR REG')
->         and l_shipinstruct = 'DELIVER IN PERSON' )
-> or ( p_partkey = l_partkey and p_brand = 'Brand#24'
->         and p_container in ('LG CASE', 'LG BOX', 'LG PACK', 'LG PKG')
->         and l_quantity >= 21 and l_quantity <= 21 + 10
->         and p_size between 1 and 15
->         and l_shipmode in ('AIR', 'AIR REG')
->         and l_shipinstruct = 'DELIVER IN PERSON' );

+-----+
| revenue |
+-----+
| 317693789.3851 |
+-----+
1 row in set (1.37 sec)
```

BETWEEN函数 & IN函数支持

未开启并行查询，耗时21.73秒，开启并行查询后，耗时1.37秒，提升15.86倍。

原始SQL语句，如下所示：

```
select sum(l_extendedprice* (1 - l_discount)) as revenue
```

```

from lineitem, part
where ( p_partkey = l_partkey and p_brand = 'Brand#12'
      and p_container in ('SM CASE', 'SM BOX', 'SM PACK', 'SM PKG')
      and l_quantity >= 6 and l_quantity <= 6 + 10
      and p_size between 1 and 5
      and l_shipmode in ('AIR', 'AIR REG')
      and l_shipinstruct = 'DELIVER IN PERSON' )
or ( p_partkey = l_partkey and p_brand = 'Brand#13'
    and p_container in ('MED BAG', 'MED BOX', 'MED PKG', 'MED PACK')
    and l_quantity >= 10 and l_quantity <= 10 + 10
    and p_size between 1 and 10
    and l_shipmode in ('AIR', 'AIR REG')
    and l_shipinstruct = 'DELIVER IN PERSON' )
or ( p_partkey = l_partkey and p_brand = 'Brand#24'
    and p_container in ('LG CASE', 'LG BOX', 'LG PACK', 'LG PKG')
    and l_quantity >= 21 and l_quantity <= 21 + 10
    and p_size between 1 and 15
    and l_shipmode in ('AIR', 'AIR REG')
    and l_shipinstruct = 'DELIVER IN PERSON' );

```

未开启并行查询，耗时21.73秒。

```

mysql> select /*+ SET_VAR(max_parallel_degree=0) */ sum(l_extendedprice* (1 - l_discount)) as revenue
-> from lineitem, part
-> where ( p_partkey = l_partkey and p_brand = 'Brand#12'
->         and p_container in ('SM CASE', 'SM BOX', 'SM PACK', 'SM PKG')
->         and l_quantity >= 6 and l_quantity <= 6 + 10
->         and p_size between 1 and 5
->         and l_shipmode in ('AIR', 'AIR REG')
->         and l_shipinstruct = 'DELIVER IN PERSON' )
-> or ( p_partkey = l_partkey and p_brand = 'Brand#13'
->      and p_container in ('MED BAG', 'MED BOX', 'MED PKG', 'MED PACK')
->      and l_quantity >= 10 and l_quantity <= 10 + 10
->      and p_size between 1 and 10
->      and l_shipmode in ('AIR', 'AIR REG')
->      and l_shipinstruct = 'DELIVER IN PERSON' )
-> or ( p_partkey = l_partkey and p_brand = 'Brand#24'
->      and p_container in ('LG CASE', 'LG BOX', 'LG PACK', 'LG PKG')
->      and l_quantity >= 21 and l_quantity <= 21 + 10
->      and p_size between 1 and 15
->      and l_shipmode in ('AIR', 'AIR REG')
->      and l_shipinstruct = 'DELIVER IN PERSON' );

+-----+
| revenue |
+-----+
| 317693789.3851 |
+-----+
1 row in set (21.73 sec)

```

开启并行查询后，耗时1.37秒，提升15.86倍。

```
mysql> select /*+ SET_VAR(max_parallel_degree=32) */ sum(l_extendedprice* (1 - l_discount)) as revenue
-> from   lineitem, part
-> where  ( p_partkey = l_partkey and p_brand = 'Brand#12'
->         and p_container in ('SM CASE', 'SM BOX', 'SM PACK', 'SM PKG')
->         and l_quantity >= 6 and l_quantity <= 6 + 10
->         and p_size between 1 and 5
->         and l_shipmode in ('AIR', 'AIR REG')
->         and l_shipinstruct = 'DELIVER IN PERSON' )
-> or ( p_partkey = l_partkey and p_brand = 'Brand#13'
->         and p_container in ('MED BAG', 'MED BOX', 'MED PKG', 'MED PACK')
->         and l_quantity >= 10 and l_quantity <= 10 + 10
->         and p_size between 1 and 10
->         and l_shipmode in ('AIR', 'AIR REG')
->         and l_shipinstruct = 'DELIVER IN PERSON' )
-> or ( p_partkey = l_partkey and p_brand = 'Brand#24'
->         and p_container in ('LG CASE', 'LG BOX', 'LG PACK', 'LG PKG')
->         and l_quantity >= 21 and l_quantity <= 21 + 10
->         and p_size between 1 and 15
->         and l_shipmode in ('AIR', 'AIR REG')
->         and l_shipinstruct = 'DELIVER IN PERSON' );
+-----+
| revenue |
+-----+
| 317693789.3851 |
+-----+
1 row in set (1.37 sec)
```

LIMIT支持

未开启并行查询，耗时339.22 秒，开启并行查询后，耗时29.31秒，提升11.57倍。

原始SQL语句，如下所示：

```
select l_shipmode, sum(case when o_orderpriority = '1-URGENT' or o_orderpriority = '2-
HIGH' then 1
    else 0
end) as high_line_count, sum(case when o_orderpriority <> '1-URGENT' and o_orderpri
ority <> '2-HIGH' then 1
    else 0
end) as low_line_count
from   orders, lineitem
where  o_orderkey = l_orderkey
and l_shipmode in ('MAIL', 'TRUCK')
and l_commitdate < l_receiptdate
and l_shipdate < l_commitdate
and l_receiptdate >= date '1996-01-01'
and l_receiptdate < date '1996-01-01' + interval '1' year
group by l_shipmode
order by l_shipmode limit 10;
```

未开启并行查询，耗时339.22 秒。

```
mysql> select /*+ SET_VAR(max_parallel_degree=0) */
  l_shipmode, sum(case when o_orderpriority = '1-URGENT' or o_orderpriority = '2-HIGH' then 1
    else 0
  end) as high_line_count, sum(case when o_orderpriority <> '1-URGENT' and o_orderpriority <> '2-HIGH' then 1
    else 0
  end) as low_line_count
from orders, lineitem
where o_orderkey = l_orderkey
and l_shipmode in ('MAIL', 'TRUCK')
and l_commitdate < l_receiptdate
and l_shipdate < l_commitdate
and l_receiptdate >= date '1996-01-01'
and l_receiptdate < date '1996-01-01' + interval '1' year
group by l_shipmode
order by l_shipmode limit 10;
```

l_shipmode	high_line_count	low_line_count
MAIL	625339	937117
TRUCK	625580	937993

2 rows in set (339.22 sec)

开启并行查询后，耗时29.31秒，提升11.57倍。

```
mysql> select /*+ SET_VAR(max_parallel_degree=32) */
  l_shipmode, sum(case when o_orderpriority = '1-URGENT' or o_orderpriority = '2-HIGH' then 1
    else 0
  end) as high_line_count, sum(case when o_orderpriority <> '1-URGENT' and o_orderpriority <> '2-HIGH' then 1
    else 0
  end) as low_line_count
from orders, lineitem
where o_orderkey = l_orderkey
and l_shipmode in ('MAIL', 'TRUCK')
and l_commitdate < l_receiptdate
and l_shipdate < l_commitdate
and l_receiptdate >= date '1996-01-01'
and l_receiptdate < date '1996-01-01' + interval '1' year
group by l_shipmode
order by l_shipmode limit 10;
```

l_shipmode	high_line_count	low_line_count
MAIL	625339	937117
TRUCK	625580	937993

2 rows in set (29.31 sec)

INTERVAL函数支持

未开启并行查询，耗时220.87秒，开启并行查询后，耗时7.75秒，提升28.5倍。

原始SQL语句，如下所示：

```
select
  100.00 * sum(case when p_type like 'PROMO%' then l_extendedprice * (1 - l_discount)
    else 0
  end) / sum(l_extendedprice * (1 - l_discount)) as promo_revenue
from lineitem, part
where l_partkey = p_partkey
and l_shipdate >= date '1996-01-01'
and l_shipdate < date '1996-01-01' + interval '1' month limit 10;
```

未开启并行查询，耗时220.87秒。

```
mysql> select /*+ SET_VAR(max_parallel_degree=0) */
-> 100.00 * sum(case when p_type like 'PROMO%' then l_extendedprice * (1 - l_discount)
-> else 0
-> end) / sum(l_extendedprice * (1 - l_discount)) as promo_revenue
-> from lineitem, part
-> where l_partkey = p_partkey
-> and l_shipdate >= date '1996-01-01'
-> and l_shipdate < date '1996-01-01' + interval '1' month limit 10;

^@+-----+
| promo_revenue |
+-----+
| 16.6415897388 |
+-----+
1 row in set (3 min 40.87 sec)
```

开启并行查询后，耗时7.75秒，提升28.5倍。

```
mysql>
mysql> select /*+ SET_VAR(max_parallel_degree=32) */
-> 100.00 * sum(case when p_type like 'PROMO%' then l_extendedprice * (1 - l_discount)
-> else 0
-> end) / sum(l_extendedprice * (1 - l_discount)) as promo_revenue
-> from lineitem, part
-> where l_partkey = p_partkey
-> and l_shipdate >= date '1996-01-01'
-> and l_shipdate < date '1996-01-01' + interval '1' month limit 10;

+-----+
| promo_revenue |
+-----+
| 16.6415897388 |
+-----+
1 row in set (7.75 sec)
```

CASE WHEN支持

未开启并行查询，耗时220.87秒，开启并行查询后，耗时7.75秒，提升28.5倍。

原始SQL语句，如下所示：

```
select
  100.00 * sum(case when p_type like 'PROMO%' then l_extendedprice * (1 - l_discount)
  else 0
end) / sum(l_extendedprice * (1 - l_discount)) as promo_revenue
from lineitem, part
where l_partkey = p_partkey
and l_shipdate >= date '1996-01-01'
and l_shipdate < date '1996-01-01' + interval '1' month limit 10;
```

未开启并行查询，耗时220.87秒。

```
mysql> select /*+ SET_VAR(max_parallel_degree=0) */
-> 100.00 * sum(case when p_type like 'PROMO%' then l_extendedprice * (1 - l_discount)
-> else 0
-> end) / sum(l_extendedprice * (1 - l_discount)) as promo_revenue
-> from lineitem, part
-> where l_partkey = p_partkey
-> and l_shipdate >= date '1996-01-01'
-> and l_shipdate < date '1996-01-01' + interval '1' month limit 10;

^@+-----+
| promo_revenue |
+-----+
| 16.6415897388 |
+-----+
1 row in set (3 min 40.87 sec)
```

开启并行查询后，耗时7.75秒，提升28.5倍。

```
mysql>
mysql> select /*+ SET_VAR(max_parallel_degree=32) */
-> 100.00 * sum(case when p_type like 'PROMO%' then l_extendedprice * (1 - l_discount)
-> else 0
-> end) / sum(l_extendedprice * (1 - l_discount)) as promo_revenue
-> from lineitem, part
-> where l_partkey = p_partkey
-> and l_shipdate >= date '1996-01-01'
-> and l_shipdate < date '1996-01-01' + interval '1' month limit 10;

+-----+
| promo_revenue |
+-----+
| 16.6415897388 |
+-----+
1 row in set (7.75 sec)
```

LIKE支持

未开启并行查询，耗时427.46秒，开启并行查询后，耗时33.72秒，提升12.68倍。

原始SQL语句，如下所示：

```
select s_name, s_address from
supplier, nation where
s_suppkey in
( select ps_suppkey from partsupp where
ps_partkey in ( select p_partkey from part where p_name like 'dark%')
and ps_availqty > (select 0.0005 * sum(l_quantity) as col1
from lineitem, partsupp
where l_partkey = ps_partkey and l_suppkey = ps_suppkey
and l_shipdate >= date '1993-01-01' and l_shipdate < date '1993-01-01' + interval '1'
year)
)
and s_nationkey = n_nationkey and n_name = 'JORDAN'
order by s_name limit 10;
```

未开启并行查询，耗时427.46秒。


```
mysql>
mysql> select /*+ SET_VAR(max_parallel_degree=0) */ s_name, s_address from
-> supplier, nation where
-> s_suppkey in
-> ( select ps_suppkey from partsupp where
->      ps_partkey in ( select p_partkey from part where p_name like 'dark%')
->      and ps_availqty>(select 0.0005 * sum(l_quantity) as col1
->      from lineitem, partsupp
->      where l_partkey = ps_partkey and l_suppkey = ps_suppkey
->      and l_shipdate >= date '1993-01-01' and l_shipdate < date '1993-01-01' + interval '1' year)
-> )
-> and s_nationkey = n_nationkey and n_name = 'JORDAN'
-> order by s_name limit 10;
ct p_partkey from part where p_name like 'dark%')
      and ps_availqty>(select 0.0005 * sum(l_quantity) as col1
      from lineitem, partsupp
      where l_partkey = ps_partkey and l_suppkey = ps_suppkey
      and l_shipdate >= date '1993-01-01' and l_shipdate < date '1993-01-01' + interval '1' year)
      )
and s_nationkey = n_nationkey and n_name = 'JORDAN'
order by s_name limit 10;
^@^@^@^@^@Empty set (7 min 7.46 sec)
```

开启并行查询后，耗时33.72秒，提升12.68倍。

```
mysql> select /*+ SET_VAR(max_parallel_degree=32) */ s_name, s_address from
-> supplier, nation where
-> s_suppkey in
-> ( select ps_suppkey from partsupp where
->      ps_partkey in ( select p_partkey from part where p_name like 'dark%')
->      and ps_availqty>(select 0.0005 * sum(l_quantity) as col1
->      from lineitem, partsupp
->      where l_partkey = ps_partkey and l_suppkey = ps_suppkey
->      and l_shipdate >= date '1993-01-01' and l_shipdate < date '1993-01-01' + interval '1' year)
-> )
-> and s_nationkey = n_nationkey and n_name = 'JORDAN'
-> order by s_name limit 10;
^@Empty set (33.72 sec)
```

SUBQUERY支持

未开启并行查询，耗时16.23秒，开启并行查询后，耗时1.4秒，提升11.59倍。

原始SQL语句，如下所示：

```
select
  s_acctbal,
  s_name,
  n_name,
  p_partkey,
  p_mfgr,
  s_address,
  s_phone,
  s_comment
from
  part,
  supplier,
  partsupp,
  nation,
  region
where
  p_partkey = ps_partkey
  and s_suppkey = ps_suppkey
```



```

and p_size = 35
and p_type like '%STEEL'
and s_nationkey = n_nationkey
and n_regionkey = r_regionkey
and r_name = 'AMERICA'
and ps_supplycost = (
    select
        min(ps_supplycost)
    from
        partsupp,
        supplier,
        nation,
        region
    where
        p_partkey = ps_partkey
        and s_suppkey = ps_suppkey
        and s_nationkey = n_nationkey
        and n_regionkey = r_regionkey
        and r_name = 'AMERICA'
)
order by
    s_acctbal desc,
    n_name,
    s_name,
    p_partkey;
limit 100;

```

未开启并行查询，耗时16.23秒。

```

mysql> select /*+ SET_VAR(max_parallel_degree=0) */
    s_acctbal,
    s_name,
    n_name,
    p_partkey,
    p_mfgr,
    s_address,
    s_phone,
    s_comment
from
    part,
    supplier,
    partsupp,
    nation,
    region
where
    p_partkey = ps_partkey
    and s_suppkey = ps_suppkey
    and p_size = 35
    and p_type like '%STEEL'
    and s_nationkey = n_nationkey
    and n_regionkey = r_regionkey
    and r_name = 'AMERICA'
    and ps_supplycost = (
        select
            min(ps_supplycost)
        from
            partsupp,
            supplier,
            nation,
            region
        where
            p_partkey = ps_partkey
            and s_suppkey = ps_suppkey
            and s_nationkey = n_nationkey
            and n_regionkey = r_regionkey
            and r_name = 'AMERICA'
    )
order by
    s_acctbal desc,
    n_name,
    s_name,
    p_partkey
limit 100;

```

s_acctbal	s_name	n_name	p_partkey	p_mfgr	s_address	s_phone	s_comment
9999.99	Supplier#00627583	CANADA	3127576	Manufacture#4	DN0XH2R9 1XXcAZtG4fux1Z	13-841-451-5653	uffily about the thin asymptotes—regular,busy instructions behind the Tiresias'
9999.57	Supplier#00389448	PERU	5139442	Manufacture#4	gThQW53f5LRhkpLJ00IUgsfZ0Xp10HN.4Qw	27-224-704-7021	tealthy,sly asymptotes according to the idle depths

开启并行查询后，耗时1.4秒，提升11.59倍。

```
mysql> select /*+ SET_VAR(max_parallel_degree=32) */
s_acctbal,
s_name,
n_name,
p_partkey,
p_mfgr,
s_address,
s_phone,
s_comment
from
part,
supplier,
partsupp,
nation,
region
where
p_partkey = ps_partkey
and s_suppkey = ps_suppkey
and p_size = 35
and p_type like '%STEEL'
and s_nationkey = n_nationkey
and n_regionkey = r_regionkey
and r_name = 'AMERICA'
and ps_supplycost = (
select
min(ps_supplycost)
from
partsupp,
supplier,
nation,
region
where
p_partkey = ps_partkey
and s_suppkey = ps_suppkey
and s_nationkey = n_nationkey
and n_regionkey = r_regionkey
and r_name = 'AMERICA'
)
order by
s_acctbal desc,
n_name,
s_name,
p_partkey
limit 100;
```

s_acctbal	s_name	n_name	p_partkey	p_mfgr	s_address	s_phone	s_comment
9999.99	Supplier#00627583	CANADA	3127576	Manufacture#4	DN0XHzR9 1XXcAZtG4fux1Z	13-841-451-5653	uffily about the thin asymptotes---regular,busy

GROUP BY WITH ROLLUP支持

GROUP BY WITH ROLLUP详细介绍请参见[MySQL-with rollup函数运用](#)和[MySQL ROLLUP](#)。

未开启并行查询，耗时318.73秒，开启并行查询后，耗时22.30秒，提升14.29倍。

原始SQL语句，如下所示：

```
select
l_returnflag,
l_linestatus,
sum(l_quantity) as sum_qty,
sum(l_extendedprice) as sum_base_price,
sum(l_extendedprice * (1 - l_discount)) as sum_disc_price,
sum(l_extendedprice * (1 - l_discount) * (1 + l_tax)) as sum_charge,
avg(l_quantity) as avg_qty,
avg(l_extendedprice) as avg_price,
avg(l_discount) as avg_disc,
count(*) as count_order
from
lineitem
where
l_shipdate <= date_sub('1998-12-01', interval ':1' day)
group by
l_returnflag,
l_linestatus
with rollup
order by
l_returnflag,
l_linestatus;
```

未开启并行查询，耗时318.73秒。

```

root@localhost:dbt3 8.0.13-rds-dev> select
->   l_returnflag,
->   l_linestatus,
->   sum(l_quantity) as sum_qty,
->   sum(l_extendedprice) as sum_base_price,
->   sum(l_extendedprice * (1 - l_discount)) as sum_disc_price,
->   sum(l_extendedprice * (1 - l_discount) * (1 + l_tax)) as sum_charge,
->   avg(l_quantity) as avg_qty,
->   avg(l_extendedprice) as avg_price,
->   avg(l_discount) as avg_disc,
->   count(*) as count_order
-> from
->   lineitem
-> where
->   l_shipdate <= date_sub('1998-12-01', interval '1' day)
-> group by
->   l_returnflag,
->   l_linestatus
-> with rollup
-> order by
->   l_returnflag,
->   l_linestatus;

```

l_returnflag	l_linestatus	sum_qty	sum_base_price	sum_disc_price	sum_charge	avg_qty	avg_price	avg_disc	count_order
NULL	NULL	1526902118.00	2289567239008.24	2175081814581.5635	2262103960452.919838	25.501098	38238.521048	0.050000	59875936
A	NULL	376833718.00	565037383437.68	536782889857.2197	558261263942.606353	25.500331	38236.069808	0.050005	14777601
A	F	376833718.00	565037383437.68	536782889857.2197	558261263942.606353	25.500331	38236.069808	0.050005	14777601
N	NULL	773043634.00	1159154050514.15	1101191254365.0758	1145250897418.143233	25.497847	38233.200251	0.050000	30317997
N	F	9830917.00	14735826224.79	13998714160.5500	14559295139.030004	25.516888	38247.950728	0.049977	385271
N	O	763212717.00	1144418224289.36	1087192540204.5258	1130691602279.113229	25.497601	38233.010394	0.050000	29932726
R	NULL	377024766.00	565375805056.41	537107670359.2680	558591799092.170252	25.508535	38251.886057	0.049997	14780338
R	F	377024766.00	565375805056.41	537107670359.2680	558591799092.170252	25.508535	38251.886057	0.049997	14780338

8 rows in set, 3 warnings (5 min 18.73 sec)

开启并行查询后，耗时22.30秒，提升14.29倍。

```

root@localhost:dbt3 8.0.13-rds-dev> select /*+ SET_VAR(max_parallel_degree=32) */
->   l_returnflag,
->   l_linestatus,
->   sum(l_quantity) as sum_qty,
->   sum(l_extendedprice) as sum_base_price,
->   sum(l_extendedprice * (1 - l_discount)) as sum_disc_price,
->   sum(l_extendedprice * (1 - l_discount) * (1 + l_tax)) as sum_charge,
->   avg(l_quantity) as avg_qty,
->   avg(l_extendedprice) as avg_price,
->   avg(l_discount) as avg_disc,
->   count(*) as count_order
-> from
->   lineitem
-> where
->   l_shipdate <= date_sub('1998-12-01', interval '1' day)
-> group by
->   l_returnflag,
->   l_linestatus
-> with rollup
-> order by
->   l_returnflag,
->   l_linestatus;

```

l_returnflag	l_linestatus	sum_qty	sum_base_price	sum_disc_price	sum_charge	avg_qty	avg_price	avg_disc	count_order
NULL	NULL	1526902118.00	2289567239008.24	2175081814581.5635	2262103960452.919838	25.501098	38238.521048	0.050000	59875936
A	NULL	376833718.00	565037383437.68	536782889857.2197	558261263942.606353	25.500331	38236.069808	0.050005	14777601
A	F	376833718.00	565037383437.68	536782889857.2197	558261263942.606353	25.500331	38236.069808	0.050005	14777601
N	NULL	773043634.00	1159154050514.15	1101191254365.0758	1145250897418.143233	25.497847	38233.200251	0.050000	30317997
N	F	9830917.00	14735826224.79	13998714160.5500	14559295139.030004	25.516888	38247.950728	0.049977	385271
N	O	763212717.00	1144418224289.36	1087192540204.5258	1130691602279.113229	25.497601	38233.010394	0.050000	29932726
R	NULL	377024766.00	565375805056.41	537107670359.2680	558591799092.170252	25.508535	38251.886057	0.049997	14780338
R	F	377024766.00	565375805056.41	537107670359.2680	558591799092.170252	25.508535	38251.886057	0.049997	14780338

8 rows in set, 3 warnings (22.30 sec)

INSERT ... SELECT/REPLACE ... SELECT支持

未开启并行查询，耗时182.82秒，开启并行查询后，耗时23.25秒，提升7.86倍。

原始SQL语句，如下所示：

```

insert into line_item_ap
SELECT  l_returnflag,
        l_linestatus,
        Sum(l_quantity)                AS sum_qty,
        Sum(l_extendedprice)           AS sum_base_price,
        Sum(l_extendedprice * (1 - l_discount)) AS sum_disc_price,
        Sum(l_extendedprice * (1 - l_discount) * (1 + l_tax)) AS sum_charge,
        Avg(l_quantity)                AS avg_qty,

```

```

        Avg(l_extendedprice)      AS avg_price,
        Avg(l_discount)          AS avg_disc,
        Count(*)                 AS count_order
FROM   lineitem
WHERE  l_shipdate <= date '1998-12-01' - INTERVAL '93' day
GROUP BY l_returnflag,
         l_linestatus
ORDER BY l_returnflag,
         l_linestatus ;

```

未开启并行查询，耗时182.82秒。

```

mysql> insert into line_item_ap
-> SELECT  l_returnflag,
->         l_linestatus,
->         Sum(l_quantity)                AS sum_qty,
->         Sum(l_extendedprice)           AS sum_base_price,
->         Sum(l_extendedprice * (1 - l_discount)) AS sum_disc_price,
->         Sum(l_extendedprice * (1 - l_discount) * (1 + l_tax)) AS sum_charge,
->         Avg(l_quantity)                AS avg_qty,
->         Avg(l_extendedprice)           AS avg_price,
->         Avg(l_discount)               AS avg_disc,
->         Count(*)                      AS count_order
-> FROM    lineitem
-> WHERE   l_shipdate <= date '1998-12-01' - INTERVAL '93' day
-> GROUP BY l_returnflag,
->         l_linestatus
-> ORDER BY l_returnflag,
->         l_linestatus ;
Query OK, 4 rows affected (3 min 2.82 sec)

```

开启并行查询后，耗时23.25秒，提升7.86倍。

```

mysql> insert into line_item_ap
-> SELECT  l_returnflag,
->         l_linestatus,
->         Sum(l_quantity)                AS sum_qty,
->         Sum(l_extendedprice)           AS sum_base_price,
->         Sum(l_extendedprice * (1 - l_discount)) AS sum_disc_price,
->         Sum(l_extendedprice * (1 - l_discount) * (1 + l_tax)) AS sum_charge,
->         Avg(l_quantity)                AS avg_qty,
->         Avg(l_extendedprice)           AS avg_price,
->         Avg(l_discount)               AS avg_disc,
->         Count(*)                      AS count_order
-> FROM    lineitem
-> WHERE   l_shipdate <= date '1998-12-01' - INTERVAL '93' day
-> GROUP BY l_returnflag,
->         l_linestatus
-> ORDER BY l_returnflag,
->         l_linestatus ;
Query OK, 4 rows affected (23.25 sec)

```

15.5.3 并行查询使用限制与串行执行结果兼容问题

本文为您介绍并行查询的使用限制以及与串行执行结果可能不兼容的地方，帮助您正确使用并行查询功能。

并行查询的使用限制

PolarDB会持续迭代并行查询的能力，目前以下情况暂时无法享受并行查询带来的性能提升：

- 查询系统表或临时表或非InnoDB表
- 使用全文索引的查询
- 存储过程Procedures
- Recursive CTE

- Windows Functions（函数本身不能并行计算，整个查询可以并行）
- GIS（函数本身不能并行计算，整个查询可以并行）
- Index Merge
- 串行化隔离级别事务内的查询语句

与串行执行结果可能不兼容的地方

- 错误提示次数可能会增多

在串行执行出错误提示的查询，在并行执行情况下，可能每个工作线程都会提示错误，导致总体错误提示数会增多。

- 精度问题

并行查询的执行过程中，可能会出现比串行执行多出中间结果的存储，如果中间结果是浮点型，可能会导致浮点部分精度差别，导致最终结果有细微的差别。

- 结果集顺序差别

因为是多线程扫描，返回的结果集可能与执行顺序不一致。例如使用了限制（limit），并且不带order by的情况下，更容易出现与串行结果顺序不同的现象。

- 网络包或者中间结果长度超出max_allowed_packet允许的最大长度

并行查询的执行过程中，相比串行执行可能会多出中间结果。如果中间结果的长度超出了max_allowed_packet定义的最大长度，可能出现错误提示，可以通过增加max_allowed_packet参数的值来解决。修改参数请参见[设置集群参数](#)。