

阿里云

云数据库 PolarDB 用户指南

文档版本：20220713

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. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

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

目录

1.概述	12
2.功能概览	13
3.数据迁移	22
3.1. 数据迁移方案概览	22
3.2. 【注意事项】MySQL 5.7迁移到PolarDB MySQL引擎8.0版本	22
3.3. 从RDS迁移至PolarDB	23
3.3.1. 一键升级RDS MySQL至PolarDB MySQL引擎	23
3.3.2. 一键克隆RDS MySQL至PolarDB MySQL引擎	34
3.3.3. RDS MySQL迁移至PolarDB MySQL	38
3.3.4. 包年包月RDS迁移至PolarDB后申请转单优惠退款	48
3.4. PolarDB间的数据迁移	49
3.4.1. PolarDB MySQL间迁移	49
3.5. 从其它数据库迁移至PolarDB	59
3.5.1. 自建MySQL迁移至PolarDB MySQL	59
3.5.2. 从自建MySQL迁移至PolarDB MySQL引擎（mysqldump工具...)	68
3.5.3. 从Amazon Aurora MySQL迁移至PolarDB MySQL	70
3.6. 从PolarDB迁移至其它数据库	78
3.6.1. PolarDB MySQL迁移至RDS MySQL	78
4.数据同步	88
4.1. 数据同步方案概览	88
4.2. PolarDB间的数据同步	88
4.2.1. PolarDB MySQL集群间的单向同步	88
4.2.2. PolarDB MySQL集群间的双向同步	94
4.3. PolarDB与其它数据库的数据同步	102
4.3.1. RDS MySQL同步至PolarDB MySQL集群	102
4.3.2. PolarDB MySQL同步至RDS MySQL	110

4.3.3. 从PolarDB MySQL同步至云原生数据仓库AnalyticDB MySQL	118
4.3.4. 从PolarDB MySQL同步至云原生数据仓库AnalyticDB Postgr...	125
4.3.5. 从PolarDB MySQL同步至Datahub	131
4.3.6. 从PolarDB MySQL同步到Kafka	137
4.3.7. 从ECS上的自建MySQL同步至PolarDB MySQL	143
4.3.8. 从通过专线、VPN网关或智能接入网关接入的自建MySQL同步至..	149
5.账号管理	155
5.1. 账号概述	155
5.2. 控制台账号	156
5.2.1. 注册和登录阿里云账号	156
5.2.2. 创建和管理RAM用户	157
5.2.3. 通过自定义策略授权RAM用户管理PolarDB	157
5.3. 数据库账号	159
5.3.1. 创建数据库账号	159
5.3.2. 管理数据库账号	162
5.3.3. 账号权限	165
5.3.4. 系统账号说明	166
6.数据安全和加密	167
6.1. 设置集群白名单	167
6.1.1. 设置白名单	167
6.1.2. 设置安全组	170
6.2. 设置SSL加密	172
6.3. 设置透明数据加密TDE	180
6.4. SQL防火墙	184
6.4.1. 概述	185
6.4.2. 设置黑名单规则	186
6.4.3. 设置白名单规则	193
6.5. 数据安全和加密常见问题	197

7.数据库代理企业版	199
7.1. 概述	199
7.2. 集群访问	202
7.2.1. 集群地址和主地址	202
7.2.2. 申请集群地址和主地址	205
7.2.3. 连接数据库集群	209
7.2.4. 如何解决集群连接失败的问题	212
7.2.5. 私有域名	213
7.2.6. 常见问题	215
7.3. 读写分离	216
7.3.1. 概述	216
7.3.2. 负载均衡	219
7.3.3. 一致性级别	220
7.3.4. 连接池	222
7.3.5. 连接保持	225
7.4. 动态脱敏	227
7.4.1. 概述	227
7.4.2. 管理脱敏规则	230
7.5. 配置数据库代理	233
7.6. 升级数据库代理的规格	236
7.7. 性能监控	237
7.8. 常见问题	238
8.数据库管理	240
9.变更集群配置	242
9.1. 概述	242
9.2. 手动变配	242
9.3. 自动变配（弹性扩缩容）	245
9.3.1. 集群规格自动变配（自动扩容）	245

9.3.2. 结合数据库自治服务DAS自动变配	247
9.3.3. 本地资源自动变配（秒级扩核）	250
9.4. 临时升配	253
9.5. 增加或删除节点	255
9.6. 历史库单节点版升级为历史库集群版	260
9.7. 手工扩容	261
10. 备份与恢复	262
10.1. 概述	262
10.2. 备份和恢复费用说明	265
10.3. 备份操作说明	266
10.3.1. 备份设置	266
10.3.2. 备份方式1：自动备份	272
10.3.3. 备份方式2：手动备份	272
10.4. 恢复操作说明	273
10.4.1. 全量恢复方式1：从备份集恢复	273
10.4.2. 全量恢复方式2：恢复到过去时间点	277
10.4.3. 库表恢复方式1：从备份集恢复	281
10.4.4. 库表恢复方式2：恢复到过去时间点	283
10.4.5. 误操作恢复最佳实践	285
10.5. 常见问题	286
11. 热备切换	289
11.1. 概述	289
11.2. 费用说明	289
11.3. 技术原理	289
11.4. 开启热备前后性能对比	293
11.5. 设置热备节点	296
12. 闪回查询	298
13. 高可用部署架构	301

13.1. 多可用区部署架构	301
13.1.1. 多可用区部署	301
13.1.2. 更换主可用区和交换机	302
13.2. 多节点部署架构	304
13.2.1. 多节点部署	304
13.2.2. 自动/手动主备切换	305
14.多主架构	307
14.1. 多主架构集群版	307
14.2. 使用说明	309
14.3. 最佳实践	312
14.3.1. DBLE与PolarDB多主架构结合使用最佳实践	312
14.3.2. Apache ShardingSphere-Proxy与PolarDB多主架构结合使...	315
15.历史库	321
15.1. 历史库	321
15.2. 使用说明	324
16.全球数据库	329
16.1. 概述	329
16.2. 技术架构	331
16.3. 异地灾备与异地多活	333
16.4. 跨地域部署GDN最佳实践	334
16.5. 创建与删除全球数据库网络	336
16.6. 添加与移除从集群	338
16.7. 连接全球数据库网络	342
17.HTAP	345
17.1. 列存索引（IMCI）	345
17.1.1. 概述	345
17.1.2. 添加只读列存节点	348
17.1.3. 配置集群地址实现行存和列存分流	350

17.1.3.1. 分流方案	350
17.1.3.2. 行存和列存自动分流	353
17.1.3.3. 行存和列存手动分流	357
17.1.4. 列存索引语法说明	358
17.1.4.1. 使用建表语句创建列存索引	358
17.1.4.2. 使用DDL语句动态增删列存索引	360
17.1.4.3. 设置压缩算法	364
18.多机并行（MPP）	366
18.1. 概述	366
18.2. 使用说明	369
19.PolarDB for AI	372
19.1. 概述	372
19.2. 开启PolarDB for AI功能	374
19.3. 使用说明	376
19.4. 算法介绍	380
19.4.1. LightGBM算法	380
19.4.2. DeepFM算法	382
19.4.3. K均值聚类算法（K-Means）	383
19.4.4. 随机森林回归算法（Random Forest Regression）	385
19.4.5. 梯度提升回归树算法（GBRT）	386
19.4.6. 线性回归算法（Linear Regression）	387
19.4.7. 支持向量回归算法（SVR）	388
20.集群回收站	390
20.1. 费用说明	390
20.2. 恢复已释放的集群	390
20.3. 彻底删除已释放的集群	394
21.监控与优化	395
21.1. 一键诊断	395

21.2. 自治中心	395
21.3. 会话管理	397
21.4. 实时性能	398
21.5. 空间分析	399
21.6. 锁分析	400
21.7. 诊断报告	401
21.8. 性能洞察	402
21.9. 性能监控	404
21.9.1. 查看性能监控指标	404
21.9.2. 修改监控频率	408
21.9.3. 创建报警规则	409
21.9.4. 管理报警规则	410
21.10. 慢SQL	411
21.11. SQL洞察	413
22.版本管理	418
23.计划内运维事件	420
23.1. 查看并管理计划内事件	420
23.2. 查询历史事件	424
24.参数配置	425
24.1. 设置集群参数和节点参数	425
24.2. 参数表达式	429
24.3. 使用参数模板	431
24.4. 高性能参数模板（8.0）	435
25.更多操作	441
25.1. 克隆集群	441
25.2. 开启Binlog	443
25.3. 设置可维护窗口	446
25.4. 重启节点	447

25.5. 查看或取消计划任务	447
25.6. 查看数据库存储用量	449
25.7. 释放集群	449
25.8. 集群保护锁	451
25.9. 标签	453
25.9.1. 绑定标签	453
25.9.2. 根据标签筛选集群	454
25.9.3. 查看集群绑定的标签	455
25.9.4. 解绑标签	456

1. 概述

PolarDB是阿里巴巴自研的新一代云原生关系型数据库，在存储计算分离架构下，利用了软硬件结合的优势，为用户提供具备极致弹性、高性能、海量存储、安全可靠的数据库服务。100%兼容MySQL 5.6/5.7/8.0, PostgreSQL 11, 高度兼容Oracle。

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 MySQL引擎 100%兼容原生MySQL和RDS MySQL，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 MySQL引擎。

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Cent OS、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
一个集群版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。更多信息请参见[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。

PolarDB控制台地址：[PolarDB控制台](#)。

2. 功能概览

本文汇总了PolarDB MySQL引擎各版本、系列支持的功能供您参考。

MySQL 8.0

类别	功能	MySQL 8.0			
		集群版	历史库单节点版	历史库集群版	单节点
数据迁移	数据迁移方案概览	✓	✓	✓	✓
集群管理	购买按量付费集群和购买包年包月集群	✓	□	✓	✓
	购买存储包和购买计算包	✓	✓	✓	✓
	包年包月转按量付费	✓	✓	✓	✓
	按量付费转包年包月	✓	✓	✓	✓
	手动续费	✓	✓	✓	✓
	自动续费	✓	✓	✓	✓
	释放集群	✓	✓	✓	✓
	设置集群参数和节点参数	✓	✓	✓	✓
	设置白名单	✓	✓	✓	✓
	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	✓	✓	✓	✓
账号	注册和登录阿里云账号	✓	✓	✓	✓
	创建和管理RAM用户	✓	✓	✓	✓
	创建数据库账号	✓	✓	✓	✓
	管理数据库账号	✓	✓	✓	✓
数据库	数据库管理	✓	✓	✓	✓
	申请集群地址和主地址	✓	✓	✓	✓

类别	功能	MySQL 8.0			
		集群版	历史库单节点版	历史库集群版	单节点
数据库代理	连接数据库集群	✓☺	✓☺	✓☺	✓☺
	私有域名	✓☺	✓☺	✓☺	✓☺
	读写分离	✓☺	☐	✓☺	☐
	动态脱敏	✓☺	☐	✓☺	☐
	配置数据库代理	✓☺	☐	✓☺	☐
变更集群配置	手动变配	✓☺	☐	✓☺	☐
	集群规格自动变配（自动扩容）	✓☺	☐	✓☺	☐
	本地资源自动变配（秒级扩核）	✓☺	☐	☐	☐
	增加或删除节点	✓☺	☐	✓☺	☐
	临时升配	✓☺	☐	☐	☐
	历史库单节点版升级为历史库集群版	☐	✓☺	☐	☐
高可用部署架构	多可用区部署和更换主可用区和交换机	✓☺	☐	☐	☐
	多节点部署和自动/手动主备切换	✓☺	☐	✓☺	☐
全球数据库	创建与删除全球数据库网络	✓☺	☐	☐	☐
	添加与移除从集群	✓☺	☐	☐	☐
	连接全球数据库网络	✓☺	☐	☐	☐
数据安全	设置透明数据加密TDE	✓☺	☐	☐	✓☺
	设置SSL加密	✓☺	✓☺	✓☺	✓☺
	自动备份	✓☺	✓☺	✓☺	✓☺
	手动备份	✓☺	✓☺	✓☺	✓☺
	全量恢复（从备份集恢复）	✓☺	✓☺	✓☺	✓☺
	全量恢复（恢复到过去时间点）	✓☺	✓☺	✓☺	✓☺

备份与恢复		MySQL 8.0			
类别	功能	集群版	历史库单节点版	历史库集群版	单节点
	库表恢复（从备份集恢复）	✓☺	☐	☐	☐
	库表恢复（恢复到过去时间点）	✓☺	☐	☐	☐
集群回收站	恢复已释放的集群	✓☺	✓☺	✓☺	✓☺
	彻底删除已释放的集群	✓☺	✓☺	✓☺	✓☺
诊断与优化	SQL洞察	✓☺	✓☺	☐	✓☺
	查看性能监控指标	✓☺	✓☺	✓☺	✓☺
	慢SQL	✓☺	✓☺	☐	☐
	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	版本管理	✓☺	✓☺	✓☺	✓☺
	开启Binlog	✓☺	☐	☐	✓☺
	ROLLUP性能增强	✓☺	✓☺	✓☺	✓☺
	表回收站	✓☺	✓☺	✓☺	✓☺
	Fast Query Cache	✓☺	✓☺	✓☺	✓☺
	并行查询（Parallel Query）	✓☺	☐	☐	✓☺
	Statement Outline	✓☺	✓☺	✓☺	✓☺
	热点行优化	✓☺	✓☺	✓☺	✓☺
	Performance Agent	✓☺	✓☺	✓☺	✓☺
	设置可维护窗口	✓☺	✓☺	✓☺	✓☺

其他功能	功能	MySQL 8.0			
		集群版	历史库单节点版	历史库集群版	单节点
	查看并管理计划内事件	✓☺	✓☺	✓☺	✓☺
	重启节点	✓☺	✓☺	✓☺	✓☺

MySQL 5.7

类别	功能	MySQL 5.7	
		集群版	单节点
数据迁移	数据迁移方案概览	✓☺	✓☺
集群管理	购买按量付费集群和购买包年包月集群	✓☺	✓☺
	购买存储包和购买计算包	✓☺	✓☺
	包年包月转按量付费	✓☺	✓☺
	按量付费转包年包月	✓☺	✓☺
	手动续费	✓☺	✓☺
	自动续费	✓☺	✓☺
	释放集群	✓☺	✓☺
	设置集群参数和节点参数	✓☺	✓☺
	设置白名单	✓☺	✓☺
	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	✓☺	✓☺
账号	注册和登录阿里云账号	✓☺	✓☺
	创建和管理RAM用户	✓☺	✓☺
	创建数据库账号	✓☺	✓☺
	管理数据库账号	✓☺	✓☺

类别	功能	MySQL 5.7	
		集群版	单节点
数据库	数据库管理	✓☺	✓☺
数据库代理	申请集群地址和主地址	✓☺	✓☺
	连接数据库集群	✓☺	✓☺
	私有域名	✓☺	✓☺
	读写分离	✓☺	☐
	动态脱敏	✓☺	☐
	配置数据库代理	✓☺	☐
弹性扩缩容	手动变配	✓☺	☐
	集群规格自动变配（自动扩容）	✓☺	☐
	本地资源自动变配（秒级扩核）	✓☺	☐
	增加或删除节点	✓☺	☐
	临时升配	✓☺	☐
高可用部署架构	多可用区部署和更换主可用区和交换机	✓☺	☐
	多节点部署和自动/手动主备切换	✓☺	☐
全球数据库	创建与删除全球数据库网络	☐	☐
	添加与移除从集群	☐	☐
	连接全球数据库网络	☐	☐
数据安全	设置透明数据加密TDE	✓☺	✓☺
	设置SSL加密	✓☺	✓☺
备份与恢复	自动备份	✓☺	✓☺
	手动备份	✓☺	✓☺
	全量恢复（从备份集恢复）	✓☺	✓☺
	全量恢复（恢复到过去时间点）	✓☺	✓☺
	库表恢复（从备份集恢复）	✓☺	☐
	库表恢复（恢复到过去时间点）	✓☺	☐

类别	功能	MySQL 5.7	
		集群版	单节点
集群回收站	恢复已释放的集群	✓☺	✓☺
	彻底删除已释放的集群	✓☺	✓☺
诊断与优化	SQL洞察	✓☺	✓☺
	查看性能监控指标	✓☺	✓☺
	慢SQL	✓☺	☐
	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	版本管理	✓☺	✓☺
	开启Binlog	✓☺	✓☺
	ROLLUP性能增强	☐	☐
	表回收站	☐	☐
	Fast Query Cache	✓☺	☐
	并行查询 (Parallel Query)	☐	☐
	Statement Outline	✓☺	✓☺
	热点行优化	☐	☐
	性能监控数据	☐	☐
其他功能	设置可维护窗口	✓☺	✓☺
	查看并管理计划内事件	✓☺	✓☺
	重启节点	✓☺	✓☺

MySQL 5.6

类别	功能	MySQL 5.6	
		集群版	单节点
数据迁移	数据迁移方案概览	✓☺	✓☺
集群管理	购买按量付费集群和购买包年包月集群	✓☺	✓☺
	购买存储包和购买计算包	✓☺	✓☺
	包年包月转按量付费	✓☺	✓☺
	按量付费转包年包月	✓☺	✓☺
	手动续费	✓☺	✓☺
	自动续费	✓☺	✓☺
	释放集群	✓☺	✓☺
	设置集群参数和节点参数	✓☺	✓☺
	设置白名单	✓☺	✓☺
	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	✓☺	✓☺
账号	注册和登录阿里云账号	✓☺	✓☺
	创建和管理RAM用户	✓☺	✓☺
	创建数据库账号	✓☺	✓☺
	管理数据库账号	✓☺	✓☺
数据库	数据库管理	✓☺	✓☺
数据库代理	申请集群地址和主地址	✓☺	✓☺
	连接数据库集群	✓☺	✓☺
	私有域名	✓☺	✓☺
	读写分离	✓☺	☐
	动态脱敏	✓☺	☐

类别	功能	MySQL 5.6	
		集群版	单节点
	配置数据库代理	✓☺	☐
弹性扩缩容	手动变配	✓☺	☐
	集群规格自动变配（自动扩容）	✓☺	☐
	本地资源自动变配（秒级扩核）	✓☺	☐
	增加或删除节点	✓☺	☐
	临时升配	✓☺	☐
高可用部署架构	多可用区部署和更换主可用区和交换机	✓☺	☐
	多节点部署和自动/手动主备切换	✓☺	☐
全球数据库	创建与删除全球数据库网络	✓☺	☐
	添加与移除从集群	✓☺	☐
	连接全球数据库网络	✓☺	☐
数据安全	设置透明数据加密TDE	✓☺	✓☺
	设置SSL加密	✓☺	✓☺
备份与恢复	自动备份	✓☺	✓☺
	手动备份	✓☺	✓☺
	全量恢复（从备份集恢复）	✓☺	✓☺
	全量恢复（恢复到过去时间点）	✓☺	✓☺
	库表恢复（从备份集恢复）	✓☺	☐
	库表恢复（恢复到过去时间点）	✓☺	☐
集群回收站	恢复已释放的集群	✓☺	✓☺
	彻底删除已释放的集群	✓☺	✓☺
	SQL洞察	✓☺	✓☺
	查看性能监控指标	✓☺	✓☺
	慢SQL	✓☺	☐
	自治中心	✓☺	☐

类别	功能	MySQL 5.6	
		集群版	单节点
诊断与优化	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	版本管理	✓☺	✓☺
	开启Binlog	✓☺	✓☺
	ROLLUP性能增强	☐	☐
	表回收站	☐	☐
	Fast Query Cache	✓☺	☐
	并行查询 (Parallel Query)	☐	☐
	Statement Outline	✓☺	✓☺
	热点行优化	✓☺	✓☺
	性能监控数据	☐	☐
其他功能	设置可维护窗口	✓☺	✓☺
	查看并管理计划内事件	✓☺	✓☺
	重启节点	✓☺	✓☺

3. 数据迁移

3.1. 数据迁移方案概览

云原生关系型数据库PolarDB提供了多种数据迁移方案，可满足不同上云、迁云的业务需求，使您可以在不影响业务的情况下平滑的实现引擎间的数据迁移。

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

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引擎（平滑迁移，推荐） • 一键克隆RDS MySQL至PolarDB MySQL引擎 • RDS MySQL迁移至PolarDB MySQL
PolarDB间的数据迁移	PolarDB MySQL间迁移
从其它数据库迁移至PolarDB	• 自建MySQL迁移至PolarDB MySQL • 从自建MySQL迁移至PolarDB MySQL引擎（mysqldump工具） • 从Amazon Aurora MySQL迁移至PolarDB MySQL
从PolarDB迁移至其它数据库	PolarDB MySQL迁移至RDS MySQL

3.2. 【注意事项】MySQL 5.7迁移到PolarDB MySQL引擎8.0版本

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

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

- [从RDS MySQL迁移至PolarDB MySQL](#)
- [从Amazon Aurora MySQL迁移至PolarDB MySQL](#)
- [PolarDB MySQL集群间的数据迁移](#)
- [从自建MySQL迁移至PolarDB MySQL](#)

客户端版本

您需要将MySQL客户端程序升级到如下版本：

- 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。
- 问题现象: 在PolarDB MySQL引擎 8.0版本数据库集群上执行带有 `kickout` 的SQL语句时, 会出现如下语法报错:

```
ERROR 1064 (42000): You have an error in your SQL syntax;
```

- 解决办法: 建议升级PolarDB MySQL引擎 8.0版本集群到最新修订版本。详细升级操作步骤, 请参见[版本管理](#)。

3.3. 从RDS迁移至PolarDB

3.3.1. 一键升级RDS MySQL至PolarDB MySQL引擎

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

升级方案

当前支持一键升级的源RDS MySQL的版本和存储类型, 以及目标PolarDB MySQL引擎的版本情况如下:

- 支持源RDS MySQL所有版本以及所有存储类型实例的迁移。不管是RDS MySQL 5.6、5.7还是8.0, 本地SSD盘还是云盘存储, 都支持一键升级至PolarDB MySQL引擎。

 **说明** 当前仅RDS MySQL高可用版系列支持一键升级至PolarDB MySQL引擎。

- 支持迁移RDS MySQL至相同或不同版本的PolarDB MySQL引擎。如支持RDS MySQL 5.6一键升级至PolarDB MySQL引擎 5.6, 也支持RDS MySQL 5.6升级至PolarDB MySQL引擎 8.0。

当前的一键升级功能通过物理迁移(流复制)和逻辑迁移(DTS数据同步)两种方式实现。

- **物理迁移(流复制)方式:** 先通过流复制的方式, 先从源RDS MySQL实例复制全量数据, 然后保持增量同步至创建的PolarDB MySQL引擎集群中。
- **逻辑迁移(DTS数据同步)方式:** 通过数据传输服务DTS, 创建数据同步任务, 将源RDS MySQL实例的库表结构和全量数据同步至创建的PolarDB MySQL引擎集群中, 然后保持增量数据同步。

 **说明** 当前逻辑迁移（DTS数据同步）方式处于灰度发布阶段。如有需求，请[提交工单](#)申请试用。

这两种方式与支持的版本和存储类型的对应关系如下：

迁移方式	源RDS MySQL实例版本和存储类型	目标PolarDB MySQL引擎集群版本
物理迁移（流复制）	RDS MySQL 5.6高可用版，且存储类型为本地SSD盘。	与RDS MySQL相同版本的PolarDB MySQL引擎集群。
	RDS MySQL 5.7高可用版，且存储类型为本地SSD盘。	
逻辑迁移（DTS数据同步）	RDS MySQL 5.7高可用版，且存储类型为云盘（ESSD PL1 云盘、ESSD PL2 云盘、ESSD PL3 云盘）。	- 与RDS MySQL相同版本的PolarDB MySQL引擎集群。 - 与RDS MySQL不同版本的PolarDB MySQL引擎集群。
	RDS MySQL 8.0高可用版，且存储类型为云盘（ESSD PL1 云盘、ESSD PL2 云盘、ESSD PL3 云盘）或本地SSD盘。	
	RDS MySQL 8.0高可用版，且存储类型为本地SSD盘。	

迁移方式对比

方案优势

一键升级功能具有如下优势：

- 可保留数据库原连接地址，无需应用程序修改任何连接配置即可切换至PolarDB。
- 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。
- 支持在线热迁移，迁移过程仅闪断一次（即当业务从RDS切换至PolarDB时）。
- 支持回滚，迁移失败可以在10分钟内恢复。
- 对于包年包月的RDS实例，数据从RDS迁移到PolarDB后，若业务已在PolarDB上稳定运行且不再需要RDS时，您可以申请转单优惠退款，避免浪费闲置的RDS资源，详情请参见[包年包月RDS迁移至PolarDB后申请转单优惠退款](#)。

前提条件

- 若要使用一键升级功能，源RDS实例版本需满足如下条件：
 - 针对RDS MySQL 5.6高可用版，内核小版本需为20190815或以上版本。
 - 针对RDS MySQL 5.7高可用版，内核小版本需为20200331或以上版本。
 - 针对RDS MySQL 8.0高可用版，目前无内核小版本要求。

 **说明** 您可以执行 `show variables like '%rds_release_date%';` 命令查看源RDS实例的内核小版本。如果源RDS实例的内核小版本低于上述指定版本，您可以将内核小版本升级到最新版。关于如何升级内核小版本，请参见[升级内核小版本](#)。

- 源RDS实例未开启TDE和SSL。若已开启，您可以选择手动创建DTS数据迁移任务将源RDS迁移至PolarDB。更多详情，请参见RDS MySQL迁移至PolarDB MySQL。
- 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（请参见创建账号），或者切换到高性能模式（参见【重要】RDS网络链路升级说明），才能进行一键升级。



使用限制

- 物理迁移（流复制）方式的使用限制如下：
 - 暂不支持跨地域迁移。
 - 迁移期间不允许对源RDS实例执行参数设置的操作。
- 逻辑迁移（DTS数据同步）方式的使用限制如下：

说明 DTS默认同步到目标数据库中时会取消外键约束，因此源数据库的级联、删除等操作不会同步到目标数据库。

类型	说明
源库限制	- 待同步的表需具备主键或唯一约束，且字段具有唯一性，否则可能会导致目标数据库中出现重复数据。 - 如同步对象为表级别，且需进行编辑（如表列名映射），则单次同步任务仅支持同步至多1000张表。当超出数量限制，任务提交后会显示请求报错，此时建议您拆分待同步的表，分批配置多个任务，或者配置整库的同步任务。 - Binlog日志： - 需开启Binlog，开启方法请参见设置实例参数，并且binlog_row_image为full。否则预检查阶段提示报错，且无法成功启动数据同步任务。 - 如为增量同步任务，DTS要求源数据库的本地Binlog日志保存24小时以上，如为全量同步和增量同步任务，DTS要求源数据库的本地Binlog日志至少保留7天以上（您可在全量同步完成后将Binlog保存时间设置为24小时以上），否则DTS可能因无法获取Binlog而导致任务失败，极端情况下甚至可能会导致数据不一致或丢失。由于您所设置的Binlog日志保存时间低于DTS要求的时间进而导致的问题，不在DTS的SLA保障范围内。如源为RDS MySQL，具体操作请参见本地日志（Binlog）。

类型	说明
其他限制	- 建议源和目标库版本保持一致，或者从低版本同步到高版本以保障兼容性。如为高版本同步至低版本，可能存在数据库兼容性问题。 - 执行数据同步前需评估源库和目标库的性能，同时建议业务低峰期执行数据同步。否则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。 - 全量初始化会并发执行INSERT操作，导致目标数据库的表产生碎片，因此全量初始化完成后目标实例的表空间比源实例的表空间大。 - 如同步对象为单个或多个表（非整库），那么在数据同步时，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。您可以使用数据管理DMS（Data Management Service）来执行在线DDL变更，请参见不锁表结构变更。 - 在DTS同步期间，不允许有除DTS外的数据写入目标库，否则会导致源库与目标库数据不一致。例如，有除DTS外的数据写入目标库时，使用DMS执行在线DDL变更，可能引起目标库数据丢失。

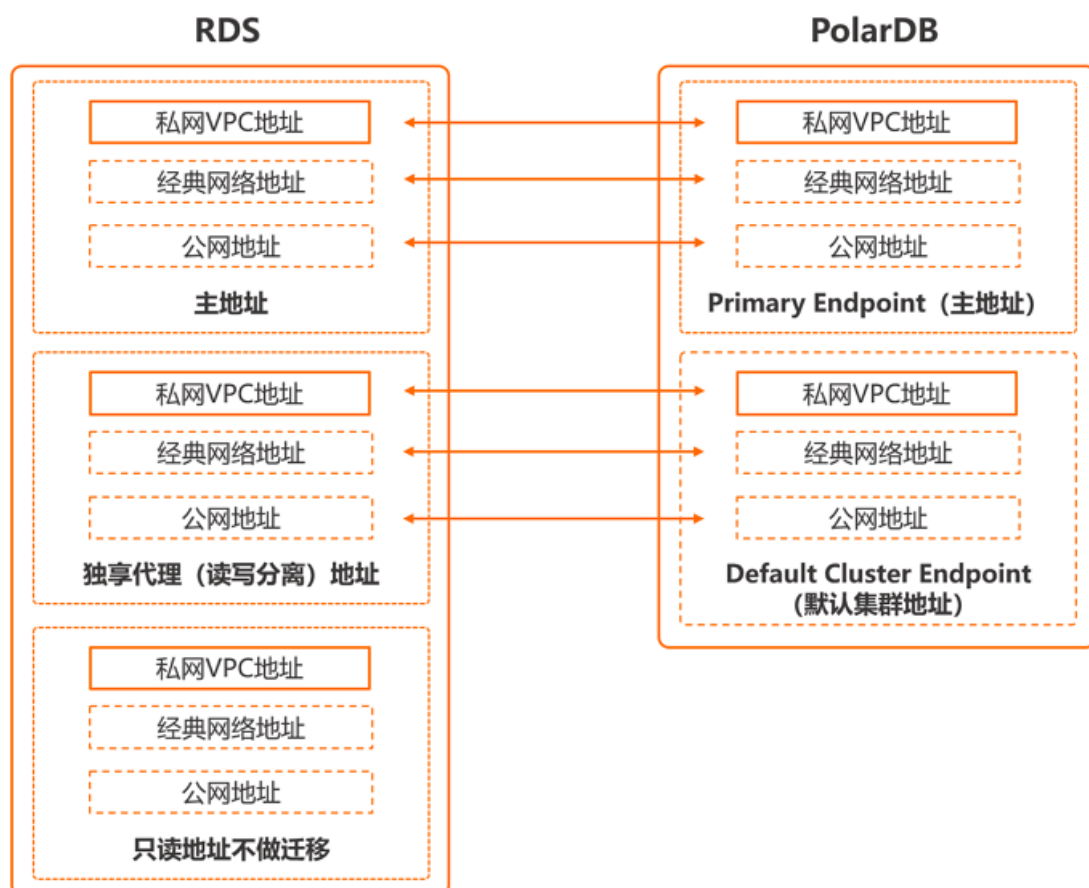
收费规则

- 物理迁移（流复制）方式的收费规则如下：
 - 从RDS迁移到PolarDB的操作完全免费，您只需承担购买PolarDB集群的费用。关于PolarDB集群的费用，详情请参见[计费项概览](#)。
 - 对于包年包月的RDS实例，从RDS迁移到PolarDB完成后，若确定业务已在PolarDB上稳定运行且不再需要RDS时，您可以申请转单优惠退款，避免浪费闲置的RDS资源，详情请参见[包年包月RDS迁移至PolarDB后申请转单优惠退款](#)。
- 逻辑迁移（DTS数据同步）方式的收费规则如下：
 您除了需承担购买PolarDB集群的费用之外，还需承担DTS所创建的同步任务产生的费用。但现阶段该功能处于试用阶段，创建的同步任务七天内不收取任何费用。具体如下：

迁移对象	费用
库表结构同步和全量数据同步	创建同步任务后，七天内不收取任何费用。 超过七天后，将会取消创建的同步任务。
增量数据同步	 说明 - 您可以前往新版DTS同步任务的列表页面查看同步任务的剩余时间。 - 若您对同步任务的时间有更长的需求，请提交工单联系阿里云技术支持。

带地址切换

一键升级RDS至PolarDB时支持带地址切换，系统会自动交换RDS和PolarDB上的连接地址，您无需在应用程序端修改任何配置即可自动连接到PolarDB。选择该切换方式后，RDS连接地址对应的PolarDB连接地址如下图所示。



使用带地址切换功能时，需注意以下几点：

- 带地址切换只会切换RDS和PolarDB的域名，Vswitch、Vip等配置不会切换。
- 仅当源RDS和目标PolarDB集群同时存在的连接地址才支持相互切换，默认情况下仅私网主地址支持带地址切换。
- 如需切换其他连接地址，您需在切换前创建好对应的连接地址，否则不会切换。关于如何为PolarDB集群和RDS实例创建连接地址，请参见[申请集群地址和主地址](#)和[设置连接地址](#)。
- 带地址切换不会切换端口，请确保RDS和PolarDB的连接端口一致（PolarDB和RDS默认使用的端口号均为3306），如需修改端口，请参见[修改内外网地址和端口（RDS）](#)或[修改连接地址和端口（PolarDB）](#)。
- 切换域名后，可能会存在DNS解析缓存问题，在缓存过期时间内可能会出现连接不上数据库或数据库只支持读操作无法执行写入操作等情况，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。

升级流程介绍

不管是物理迁移还是逻辑迁移方式，一键升级的流程均为如下：

步骤	说明
1、从RDS同步	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。

步骤	说明
2、迁移切换	- 在执行迁移切换时，您可以选择带连接切换，系统会自动交换RDS和PolarDB上的连接地址（PolarDB集群上需具备对应的连接地址，例如经典网络或公网地址），您无需在应用程序端修改任何配置即可自动连接到PolarDB。 - 执行本操作后，源RDS实例为只读状态，PolarDB集群为可读可写状态，同时会将PolarDB集群的新增数据同步到源RDS实例。  说明 迁移切换完成后，如果您发现数据存在异常等问题，可以执行迁移回滚，快速恢复至迁移前的状态。
3、完成迁移	- 您可以观察一段时间后确认业务运行正常，再执行本操作（需要在7天内完成）。 - 执行本操作后，PolarDB集群和源RDS实例间的数据同步关系将中断，此时RDS为可读可写状态。

步骤一：从RDS同步

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. 在控制台左上角，选择集群所在地域。
3. 单击**创建新集群**。
4. 选择商品类型为**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。
 - **按量付费**：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。

 **说明** PolarDB MySQL引擎已推出计算包。推荐购买按量付费的集群，因为按量付费的集群支持配合计算包使用，比包年包月付费方式更划算、更灵活。详情请参见**计费方式1：按量付费+计算包（推荐）**。

5. 设置如下参数。

参数	说明
创建方式	选择从RDS迁移。  说明 在正式迁移切换前PolarDB的读写状态为只读，且默认开启Binlog。迁移完成后，若原RDS为包年包月实例，可申请转单优惠退款。

参数	说明
地域	选择源RDS MySQL实例所在地域。  说明 新建的PolarDB集群也在此地域。
源RDS引擎	源RDS实例的引擎类型，固定为MySQL，不可变更。
源RDS版本	源RDS实例的版本，您可以选择5.6、5.7或8.0。
源RDS实例	选择源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。 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	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	目标PolarDB集群的数据库引擎版本。您可以选择与源RDS实例一致的版本，也可以选择跨版本。
系列	固定为集群版（2-16个节点）【推荐】，无需选择。
子系列	PolarDB MySQL引擎集群版支持通用规格和独享规格两种子系列。 关于两种子系列的详细对比，请参见如何选择通用规格和独享规格。
节点规格	按需选择，建议不低于源RDS实例规格。关于PolarDB节点规格，详情请参见 计算节点规格 。
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。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。
时区	设置集群时区，默认时区为UTC+08:00。
表名大小写	设置集群表名是否区分大小写，您可以选择不区分大小写（默认）或当本地数据库区分大小时，您可以选择区分大小写，便于您迁移数据。  说明 集群创建后该参数无法修改，请谨慎选择。

参数	说明
删除（释放）集群时	设置删除（释放）集群时的备份保留策略，默认保留最后一个备份（释放前自动备份）。 - 保留最后一个备份（释放前自动备份）：删除集群时保留最后一个备份。 - 保留全部备份：删除集群时保留所有备份。 - 不保留备份（释放后无法恢复）：删除集群时不保留任何备份。 说明 删除（释放）集群时保留备份可能会产生少量费用，您可以随时删除备份来节省成本。关于数据备份的计费规则，详情请参见备份存储（超出免费额度）计费规则。
集群名称	- 集群名称长度为2~128个字符，以大小写字母或中文开头，可包含数字、英文句号（.）、下划线（_）或短划线（-）。 - 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。
资源组	从已创建资源组中选择一个目标资源组。 说明 资源组是在单个云账号下将一组相关资源进行统一管理的容器，一个资源只能归属于一个资源组，详情请参见RAM资源分组与授权。

6. 设置购买时长（仅针对包年包月集群）和集群数量后，单击右侧的立即购买。

7. 在确认订单页面确认订单信息，阅读并选中服务协议，单击立即开通即可。

开通成功后，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，详情请参见[搭配存储包](#)。

8. 在支付页面，确认未支付订单信息和支付方式，单击订购。

9. 集群创建成功后，登录PolarDB控制台，确认目标PolarDB集群的复制延迟小于60秒即可进行[步骤二：迁移切换](#)操作。

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				
复制开始时间	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。					

说明

- 集群创建后开始从RDS实例同步数据，您需要在7天内进行**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**。

步骤二：迁移切换

- 进入**PolarDB控制台**。
- 找到目标集群，单击集群的ID。
- 在**基本信息**页面单击**迁移切换**。



说明

- 切换过程一般小于5分钟。
- 本操作将交换源RDS实例和目标PolarDB集群的读写状态（即将源RDS实例修改为只读，将PolarDB集群修改为可读可写），同时会更换复制方向（即将PolarDB集群的新增数据同步到RDS实例）。

- 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）。
 - 若您选择**带地址切换**（应用程序不用改连接配置），操作步骤如下：
 - 选中**带地址切换**（应用程序不用改连接配置），系统会自动交换RDS和PolarDB上的连接地址，您无需在应用程序端修改任何配置即可自动连接到PolarDB。

注意 使用带地址切换（应用程序不用改连接配置）前，请务必参见切换**注意事项**。

- 单击**确定**。
- 若您选择**不带地址切换**（应用程序需要改为新的PolarDB连接配置），操作步骤如下：
 - 选中**不带地址切换**（应用程序需要改为新的PolarDB连接配置），在迁移切换完成后，您需要尽快修改应用程序端的数据库连接地址。
 - 单击**确定**。
 - 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，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也可以选择**迁移回滚（可选）**。

步骤三：完成迁移

从RDS迁移后，您需要在7天内进行完成迁移操作。

 **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. 登录**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，期间迁移状态将显示为**关闭同步**，请耐心等待迁移完成。
- 您可以在**完成迁移**对话框内选择是否关闭PolarDB集群的Binlog。关闭Binlog会带来少量的写入性能提升，但关闭Binlog后PolarDB集群会自动重启使新配置生效。
- 如果不再需要源RDS实例，可以释放实例。若实例为包年包月实例，可申请RDS转单优惠退款。详情请参见**释放实例**和**包年包月RDS迁移至PolarDB后申请转单优惠退款**。

迁移回滚（可选）

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. 登录**PolarDB控制台**。
2. 找到目标集群，单击集群ID。
3. 在**基本信息**页面单击**迁移回滚**。



4. 在**开始回切**对话框中，选择**带地址回切（应用程序不用改连接配置）**或**不带地址回切（应用程序需要改为原RDS连接配置）**。
 - 若您选择**带地址回切（应用程序不用改连接配置）**，操作步骤如下：
 - 选中**带地址回切（应用程序不用改连接配置）**，系统会自动交换RDS和PolarDB上的连接地址，您无需在应用程序端修改任何配置即可自动回切到RDS。
 - 单击**确定**。此时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。
 - 若您选择**不带地址回切（应用程序需要改为原RDS连接配置）**，操作步骤如下：
 - 选中**不带地址回切（应用程序需要改为原RDS连接配置）**，在迁移切换完成后，您需要尽快修改应用程序端的数据库连接地址。

- b. 单击**确定**。此时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。
- c. 刷新页面，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。

查看数据同步任务（仅限逻辑迁移（DTS数据同步）方式）

常见问题

- Q：一键升级RDS MySQL至PolarDB MySQL引擎和**一键克隆RDS MySQL至PolarDB MySQL引擎**两者有什么区别？
A：两者间的区别如下表：

对比项	一键升级RDS MySQL至PolarDB MySQL引擎	一键克隆RDS MySQL至PolarDB MySQL引擎
是否支持迁移或同步增量数据	√	×
是否影响源RDS操作	×	×
源和目标的MySQL版本能否不同	√	√

 **说明** 表中√表示支持或需要，×表示不支持或不需要。

- Q：升级后的PolarDB MySQL引擎节点规格需要和源RDS MySQL的实例规格保持一致吗？
A：您可以按需选择PolarDB MySQL引擎的规格，建议不低于源RDS实例规格。所有集群版的PolarDB节点均为独享型，性能稳定可靠。详情请参见[计费项概览](#)。
- Q：升级前是否需要先购买PolarDB MySQL引擎集群？
A：您无需提前购买PolarDB MySQL引擎集群，**步骤一：从RDS同步**即包含购买和创建一个与源RDS实例数据相同的PolarDB集群的操作。
- Q：从RDS迁移会影响源RDS实例吗？
A：不会影响源RDS实例的正常运行。
- Q：平滑迁移对源RDS实例性能有影响吗？
A：迁移不会影响源RDS实例上的使用操作，但数据迁移涉及查询操作，会消耗源RDS实例一部分的查询性能。
- Q：平滑迁移对业务有影响吗？
A：平滑迁移能够保证迁移过程不丢失数据，停机（即暂停业务，不产生增量数据，而非停用数据库）时间小于10分钟，如果有需要还可以进行回滚。
- Q：取消迁移会有什么影响？
A：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。
- Q：升级完成后，将业务切换到PolarDB，应用程序端的连接地址是否需要修改？
A：您可以在**迁移切换**时选择**带地址切换（应用程序不用改连接配置）**，系统会自动交换RDS和PolarDB上的连接地址，您无需在应用程序端修改任何配置即可自动连接到PolarDB。
- Q：**迁移切换**时选择了**带地址切换（应用程序不用改连接配置）**，迁移完成后-为什么PolarDB集群仍然使用新的连接地址？
A：仅当源RDS和目标PolarDB集群同时存在的连接地址才支持相互切换，默认情况下仅私网主地址支持带地址切换。如需切换其他连接地址，您需在切换前创建好对应的连接地址，否则不会切换。关于如何为PolarDB集群和RDS实例创建连接地址，请参见[申请集群地址和主地址](#)和[设置连接地址](#)。

- Q: 源RDS实例中还包含只读实例, 若选择带地址切换(应用程序不用改连接配置), 只读实例的连接地址能否一并切换?
A: 不能。一键升级至PolarDB不会迁移源RDS只读实例的连接地址, 仅支持迁移源RDS主实例的连接地址和读写分离地址。您需要在应用程序端手动修改源RDS只读实例的连接地址为PolarDB的连接地址。
- Q: 业务成功切换后, 为什么连接不上PolarDB数据库或连接成功但只支持读操作无法执行写入操作?
A: 切换域名后, 可能会存在DNS解析缓存问题, 在缓存过期时间内可能会出现连接不上数据库或数据库只支持读操作无法执行写入操作等情况, 建议您刷新一下服务器的DNS缓存。
- Q: 一键升级到PolarDB前, 能否先进行兼容性测试并简单评估迁移工作量?
A: 您可以先通过[一键克隆RDS MySQL至PolarDB MySQL引擎](#)功能克隆一份数据到PolarDB进行兼容性测试和评估迁移工作量, 测试没有问题后再参照本文操作一键升级至PolarDB。
- Q: [迁移切换](#)后, 为什么在PolarDB控制台上看不到完成迁移按钮?
A: 若您已经执行过完成迁移操作, 该按钮将会消失, 避免您重复执行相同操作。
- Q: 一键升级至PolarDB后, 还需要在目标PolarDB集群中创建与源RDS实例相同的账号和密码吗?
A: 不需要。升级后PolarDB集群将包含源RDS实例的账号密码、数据库、IP白名单和必要的参数等信息。
- Q: 源RDS实例已开启了TDE或SSL, 如何再迁移至PolarDB集群?
A: 已开启了TDE或SSL的RDS实例不支持一键升级至PolarDB集群, 您可以选择创建DTS迁移任务将源RDS迁移至PolarDB。更多详情, 请参见[RDS MySQL迁移至PolarDB MySQL](#)。
- Q: 一键升级是否支持跨版本升级? 如将RDS MySQL 5.6升级至PolarDB MySQL引擎 8.0版本?
A: 支持。
- Q: 若在一键升级至PolarDB MySQL引擎前, 源RDS实例已开启了DTS数据同步任务, 升级时是否会影响该任务?
A: 不会。通过一键升级进行迁移时, 会先从RDS复制一份全量数据至一个新的PolarDB集群, 然后将增量数据保持同步至该PolarDB集群。源RDS上DTS数据同步任务的数据源仍然是源RDS, 数据迁移至PolarDB并不会影响源RDS上的运行和操作。
但迁移完成后, 如果您将业务切换到新的PolarDB, 且源RDS停止使用了, DTS的数据源是不会自动改到新PolarDB集群的, 此时, 您需要重新创建DTS同步任务, 将数据源改为PolarDB集群。

相关API

API	描述
CreateDBCluster	创建PolarDB集群。 ② 说明 一键升级时, 参数CreationOption取值需要为MigrationFromRDS。
DescribeDBClusterMigration	查询PolarDB集群的迁移状态。
ModifyDBClusterMigration	修改迁移任务, 进行任务的切换或回滚。
CloseDBClusterMigration	取消或完成迁移。

3.3.2. 一键克隆RDS MySQL至PolarDB MySQL引擎

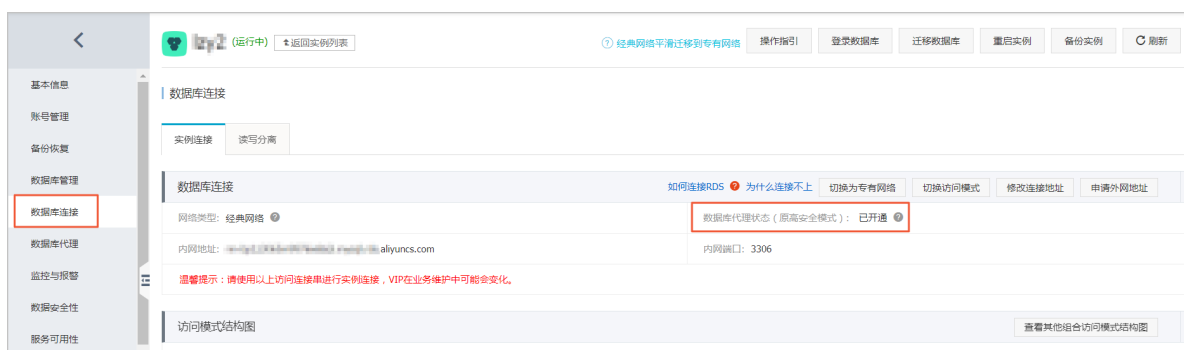
PolarDB支持从RDS MySQL一键克隆数据至新的PolarDB MySQL引擎集群。一键克隆功能将会新建一个与源RDS实例的数据相同的PolarDB集群, PolarDB集群包含源RDS实例的账号、数据库、IP白名单和必要的参数。

前提条件

- 源RDS实例版本需为RDS MySQL 5.6或5.7高可用版，且存储类型为本地SSD盘。
- 针对RDS MySQL 5.6，内核小版本需为20190815或以上版本。
- 针对RDS MySQL 5.7，内核小版本需为20200331或以上版本。

说明 您可以执行 `show variables like '%rds_release_date%';` 命令查看源RDS实例的内核小版本。如果源RDS实例的内核小版本低于上述指定版本，您可以将内核小版本升级到最新版。关于如何升级内核小版本，请参见[升级内核小版本](#)。

- 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），需要创建有权限账号（请参见[创建账号](#)），或者切换到高性能模式（参见[【重要】RDS网络链路升级说明](#)），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200 TB。
- 性能高：最高可达开源MySQL的6倍。

说明 该性能提升为在1024并发sysbench_insert场景下，PolarDB MySQL引擎与开启主备半同步Semi-Sync的开源MySQL的性能对比。

- 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[产品优势](#)。

注意事项

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[一键升级RDS MySQL至PolarDB MySQL引擎](#)。

功能亮点

- 免费
- 克隆过程数据0丢失

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. 单击创建新集群。
4. 选择商品类型为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。
 - 按量付费：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。

 **说明** PolarDB MySQL引擎已推出计算包。推荐购买按量付费的集群，因为按量付费的集群支持配合计算包使用，比包年包月付费方式更划算、更灵活。详情请参见[计费方式1：按量付费+计算包（推荐）](#)。

5. 设置如下参数。

参数	说明
地域	选择源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克隆。
源RDS引擎	源RDS实例的引擎类型，固定为MySQL，不可变更。
源RDS版本	源RDS实例的版本，您可以选择5.6或5.7。
源RDS实例	选择源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。 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	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集群的数据库引擎版本，默认与源RDS引擎版本保持一致，不可变更。
系列	固定为集群版（2-16个节点）【推荐】，无需选择。
子系列	PolarDB MySQL引擎集群版支持通用规格和独享规格两种子系列，其中： 关于两种类型的详细对比，请参见 如何选择通用规格和独享规格 。
节点规格	按需选择，建议不低于源RDS实例规格。关于PolarDB节点规格，详情请参见 计算节点规格 。
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。

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。
时区	设置集群时区，默认时区为UTC+08:00。
表名大小写	设置集群表名是否区分大小写，您可以选择不区分大小写（默认）或当本地数据库区分大小时，您可以选择区分大小写，便于您迁移数据。  说明 集群创建后该参数无法修改，请谨慎选择。
删除（释放）集群时	设置删除（释放）集群时的备份保留策略，默认保留最后一个备份（释放前自动备份）。 - 保留最后一个备份（释放前自动备份）：删除集群时保留最后一个备份。 - 保留全部备份：删除集群时保留所有备份。 - 不保留备份（释放后无法恢复）：删除集群时不保留任何备份。  说明 删除（释放）集群时保留备份可能会产生少量费用，您可以随时删除备份来节省成本，详情请参见释放集群。
集群名称	- 集群名称长度为2~128个字符，以大小写字母或中文开头，可包含数字、英文句号（.）、下划线（_）或短划线（-）。 - 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。
资源组	从已创建资源组中选择一个目标资源组。  说明 资源组是在单个云账号下将一组相关资源进行统一管理的容器，一个资源只能归属于一个资源组，详情请参见RAM资源分组与授权。

6. 设置购买时长（仅针对包年包月集群）和集群数量后，单击右侧的立即购买。

7. 在确认订单页面确认订单信息，阅读并选中服务协议，单击立即开通即可。

开通成功后，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，详情请参见[搭配存储包](#)。

8. 在支付页面，确认未支付订单信息和支付方式，单击**订购**。

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

常见问题

Q：从RDS克隆会影响源RDS实例吗？

A：不会影响源RDS实例的正常运行。

相关API

API	描述
CreateDBCluster	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[申请集群地址和主地址](#)。

3.3.3. RDS MySQL迁移至PolarDB MySQL

通过数据传输服务DTS（Data Transmission Service），可以帮助您将MySQL迁移至PolarDB MySQL引擎集群。

支持的源数据库

MySQL迁移至PolarDB MySQL引擎集群，支持源数据库MySQL为以下类型。本文以RDS MySQL实例为例介绍配置流程，其他类型的源数据库配置流程与本案例类似。

- RDS MySQL实例。
- 以下类型的自建数据库：
 - 有公网IP的自建数据库。
 - ECS上的自建数据库。
 - 通过专线、VPN网关或智能网关接入的自建数据库。
 - 通过数据库网关接入的自建数据库。

前提条件

- 已创建源RDS MySQL实例，创建方式，请参见[创建RDS MySQL实例](#)。
- 已创建目标PolarDB MySQL引擎集群，详情请参见[购买按量付费集群](#)和[购买包年包月集群](#)。

- 目标PolarDB MySQL引擎集群的存储空间须大于源RDS MySQL实例占用的存储空间。

注意事项

类型	说明
源库限制	- 带宽要求：源库所属的服务器需具备足够出口带宽，否则将影响数据迁移速率。 - 待迁移的表需具备主键或唯一约束，且字段具有唯一性，否则可能会导致目标数据库中出现重复数据。 - 如迁移对象为表级别，且需进行编辑（如表列名映射），则单次迁移任务仅支持迁移至多1000张表。当超出数量限制，任务提交后会显示请求报错，此时建议您拆分待迁移的表，分批配置多个任务，或者配置整库的迁移任务。 - 如需进行增量迁移，Binlog日志： - 需开启，并且binlog_format为row、binlog_row_image为full。否则预检查阶段提示报错，且无法成功启动数据迁移任务。  注意 如源实例自建MySQL是双主集群（两者互为主从），为保障DTS能获取全部的Binlog日志，则您需开启参数log_slave_updates。 - 如为增量迁移任务，DTS要求源数据库的本地Binlog日志保存24小时以上，如为全量迁移和增量迁移任务，DTS要求源数据库的本地Binlog日志至少保留7天以上（您可在全量迁移完成后将Binlog保存时间设置为24小时以上），否则DTS可能因无法获取Binlog而导致任务失败，极端情况下甚至可能会导致数据不一致或丢失。由于您所设置的Binlog日志保存时间低于DTS要求的时间进而导致的问题，不在DTS的SLA保障范围内。 - 源库的操作限制： - 在库表结构迁移和全量迁移阶段，请勿执行库或表结构变更的DDL操作，否则数据迁移任务会失败。 - 如仅执行全量数据迁移，请勿向源实例中写入新的数据，否则会导致源和目标数据不一致。为实时保持数据一致性，建议选择结构迁移、全量数据迁移和增量数据迁移。
其他限制	- 建议源和目标库的MySQL版本保持一致，以保障兼容性。 - 执行数据迁移前需评估源库和目标库的性能，同时建议业务低峰期执行数据迁移。否则全量数据迁移时DTS占用源和目标库一定读写资源，可能会导致数据库的负载上升。 - 由于全量数据迁移会并发执行INSERT操作，导致目标数据库的表产生碎片，因此全量迁移完成后目标数据库的表存储空间会比源实例的表存储空间大。 - 请确认DTS对数据类型为FLOAT或DOUBLE的列的迁移精度是否符合业务预期。DTS会通过 <code>ROUND (COLUMN, PRECISION)</code> 来读取这两类列的值。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。 - DTS会尝试恢复七天之内迁移失败任务。因此业务切换至目标实例前，请务必结束或释放该任务，或者将DTS访问目标实例账号的写权限用 <code>revoke</code> 命令回收掉。避免该任务被自动恢复后，源端数据覆盖目标实例的数据。

类型	说明
特殊情况	- 当源库为自建MySQL时 - 迁移时源库进行主备切换，会导致迁移任务失败。 - 由于DTS的延迟时间是根据迁移到目标库最后一条数据的时间戳和当前时间戳对比得出，源库长时间未执行DML操作可能导致延迟信息不准确。如果任务显示的延迟时间过大，您可以在源库执行一个DML操作来更新延迟信息。  说明 如果迁移对象选择为整库，您还可以创建心跳表，心跳表每秒定期更新或者写入数据。 - DTS会在源库定时执行CREATE DATABASE IF NOT EXISTS `test`命令以推进Binlog位点。 - 当目标库为PolarDB MySQL时 DTS会自动在PolarDB MySQL中创建数据库，如果待迁移的数据库名称不符合PolarDB MySQL的定义规范，您需要在配置迁移任务之前在PolarDB MySQL中创建数据库。相关操作，请参见数据库管理。

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 产品定价 。
增量数据迁移	收费，详情请参见 产品定价 。	

迁移类型

库表结构迁移

DTS将源库中迁移对象的结构定义迁移到目标库。

说明

- 目前DTS支持结构迁移的对象为表、视图、触发器、存储过程和存储函数。
- 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。

全量迁移

DTS将源库中迁移对象的存量数据，全部迁移到目标库中。

增量迁移

DTS在全量迁移的基础上，将源库的增量更新数据迁移到目标库中。通过增量数据迁移可以实现在自建应用不停机的情况下，平滑地完成数据迁移。

支持增量迁移的SQL操作

操作类型	SQL操作语句
DML	INSERT、UPDATE、DELETE
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	结构迁移	全量迁移	增量迁移
RDS MySQL实例	SELECT 权限	SELECT 权限	REPLICATION CLIENT、 REPLICATION SLAVE、 SHOW VIEW和SELECT 权限
PolarDB MySQL引擎集群	读写权限		

数据库账号创建及授权方法：

- RDS MySQL实例请参见[创建账号](#)和[修改账号权限](#)。
- PolarDB MySQL引擎集群请参见[创建数据库账号](#)。

操作步骤

1. 登录[新版DTS迁移任务的列表页面](#)。

 **说明** 您也可以登录[DMS数据管理服务](#)。在顶部菜单栏中，选择集成与开发（DTS）> 数据迁移。

2. 在页面左上角，选择迁移实例所属地域。



3. 单击[创建任务](#)，配置源库及目标库信息。

 **警告** 选择源和目标实例后，建议您仔细阅读页面上方显示的使用限制，以成功创建并执行迁移任务。

* 任务名称:

dtstd9l0mqfl

使用限制

1.待迁移的表需具备主键或唯一约束，且字段具有唯一性，否则可能会导致目标数据库中重复数据。2.如需进行增量迁移，Binlog日志需开启且至少保留24小时（建议3天以上）。3.请勿在链路创建阶段执行库或表结构变更的DDL操作，否则数据迁移链路建立会失败。4.如仅执行全量数据迁移，请勿向源实例中写入新的数据，否则会导致源和目标数据不一致。为实时保持数据一致性，建议选择结构迁移、全量数据迁移和增量数据迁移。5.DTS会自动在PolarDB MySQL中创建数据库，如果待迁移的数据库名称不符合RDS MySQL的定义规范，您需要在配置迁移任务之前在RDS MySQL中创建数据库。

更多限制，[请查看详情](#)

源库信息

选择已有的实例:

可以选择一个已有的实例进行快速配置

* 数据库类型: ⑦

DB2 iSeries(AS/400) DB2 LUW HBase Mariadb MongoDB MySQL
Oracle PolarDB MySQL PolarDB Oracle PolarDB PostgreSQL
PolarDB-X 1.0 PolarDB-X 2.0 PostgreSQL Redis SQLServer Teradata

* 接入方式:

阿里云实例 专线/VPN网关/智能网关 公网IP ECS自建数据库
云企业网CEN 数据库网关DG

* 实例地区:

华东1 (杭州)

是否跨阿里云账号: ⑦

不跨账号 跨账号

* RDS实例ID:

rm-xxxxxxx

* 数据库账号: ⑦

dtstest

* 数据库密码:

* 连接方式:

☒ 非加密连接 ☐ SSL安全连接

将连接信息保存为实例

目标库信息

选择已有的实例:

可以选择一个已有的实例进行快速配置

* 数据库类型: ⑦

AnalyticDB MySQL 3.0 AnalyticDB PostgreSQL DataHub DB2 LUW
ElasticSearch Kafka Maxcompute MySQL Oracle PolarDB MySQL
PolarDB-X 2.0

* 接入方式:

阿里云实例

* 实例地区:

华东1 (杭州)

* PolarDB实例ID:

pc-xxxxxxx

* 数据库账号: ⑦

dtstest

* 数据库密码:

将连接信息保存为实例

取消

测试连接以进行下一步

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。
	选择已有的实例	您可以按实际需求，选择是否使用已有实例。 - 如使用已有实例，数据库信息将自动填入，您无需重复输入。 - 如不使用已有实例，您需要输入下方的数据库信息。
	数据库类型	选择MySQL。

类别	配置	说明
源库信息	接入方式	选择 阿里云实例 。
	实例地区	选择源RDS MySQL实例所属地域。
	是否跨阿里云账号	本案例为同一阿里云账号间迁移，选择 不跨账号 。
	RDS实例ID	选择源RDS MySQL实例ID。
	数据库账号	填入源RDS MySQL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	填入该数据库账号对应的密码。
	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 MySQL实例的SSL加密功能，详情请参见 设置SSL加密 。
	将连接信息保存为实例或编辑连接信息模板	在选择已有的实例处是否选择了已有数据库的连接模板。 i. 是：您可以单击编辑连接信息模板自定义连接信息模板的名称。  说明 修改后的连接信息模板，将在下一次选择该连接模板作为实例连接信息时生效，不会影响原先已使用该模板配置的实例连接信息。 ii. 否：单击将连接信息保存为实例并在弹跳框中设置连接名称，可将该实例的连接信息保存为模板，便于下次自动传入。  说明 建议连接名称配置为具有业务意义的名称（无唯一性要求），便于后续识别。
目标库信	选择已有的实例	您可以按实际需求，选择是否使用已有实例。 - 如使用已有实例，数据库信息将自动填入，您无需重复输入。 - 如不使用已有实例，您需要输入下方的数据库信息。
	数据库类型	选择 PolarDB MySQL 。
	接入方式	选择 阿里云实例 。
	实例地区	选择目标PolarDB MySQL引擎集群所属地域。
	PolarDB MySQL实例ID	选择目标PolarDB MySQL引擎集群ID。
	数据库账号	填入目标PolarDB MySQL引擎集群的数据库账号，权限要求请参见 数据库账号的权限要求 。
	数据库密码	填入该数据库账号对应的密码。

信息类别	配置	说明
	将连接信息保存为实例或编辑连接信息模板	在选择已有的实例处是否选择了已有数据库的连接模板。 i. 是：您可以单击编辑连接信息模板自定义连接信息模板的名称。  说明 修改后的连接信息模板，将在下一次选择该连接模板作为实例连接信息时生效，不会影响原先已使用该模板配置的实例连接信息。 ii. 否：单击将连接信息保存为实例并在弹跳框中设置连接名称，可将该实例的连接信息保存为模板，便于下次自动传入。  说明 建议连接名称配置为具有业务意义的名称（无唯一性要求），便于后续识别。

4. 配置完成后，单击页面下方的**测试连接**以进行下一步。

警告

- 如果源或目标数据库是阿里云数据库实例（例如RDS MySQL、云数据库MongoDB版等）或ECS上的自建数据库，DTS会自动将对应地区DTS服务的IP地址添加到阿里云数据库实例的白名单或ECS的安全规则中，您无需手动添加，请参见[DTS服务器的IP地址段](#)；如果源或目标数据库是IDC自建数据库或其他云数据库，则需要您手动添加对应地区DTS服务的IP地址，以允许来自DTS服务器的访问。
- 上述场景中，DTS自动添加或您手动添加DTS服务的公网IP地址段可能会存在安全风险，一旦使用本产品代表您已理解和确认其中可能存在的安全风险，并且需要您做好基本的安全防护，包括但不限于加强账号密码强度防范、限制各网段开放的端口号、内部各API使用鉴权方式通信、定期检查并限制不需要的网段，或者使用通过内网（专线/VPN网关/智能网关）的方式接入。
- DTS任务完成或释放后，建议您手动检测并删除DTS相关的服务器IP地址段。

5. 配置任务对象及高级配置。

◦ 基础配置

基础配置

* 任务步骤:

☒ 库表结构迁移

☒ 全量迁移

☒ 增量迁移

❗ 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。

如果需要进行长期的数据实时同步，请使用数据同步功能。

注：增量迁移不支持trigger的同步[帮助文档](#)

* 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

* 同步对象:

自定义选择

源库对象

Q 支持正则，若全局搜索，请先展开树

> ☐

dttestdata

> ☐

dttestdata

> ☐

dttestdata

> ☐

dttestdata

> ☒

dttestdata

> ☐

dttestdata

> ☐

dttestdata

已选择对象

批量编辑 ?

Q 支持正则，若全局搜索，请先展开树

☐

dttestdata

【右键】选择对象
可以进行重命名或
DML、DDL配置

配置	说明
任务步骤	■ 如果只需要进行全量迁移，请同时选中库表结构迁移和全量迁移。 ■ 如果需要进行不停机迁移，请同时选中库表结构迁移、全量迁移和增量迁移。 ❓ 说明 如果未选择增量迁移，为保障数据一致性，数据迁移期间请勿在源实例中写入新的数据。

45

> 文档版本：20220713

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，数据迁移任务不会被启动。  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，请参见库表列名映射。 ■ 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，则会保留目标库中的该条记录，即源库中的该条记录不会迁移至目标库中。 ■ 表结构不一致的情况下，可能导致只能迁移部分列的数据或迁移失败。
同步对象	在同步对象框中单击待迁移的对象，然后单击  将其移动到已选择对象框。  说明
映射名称更改	■ 如需更改单个迁移对象在目标实例中的名称，请右击已选择对象中的迁移对象，设置方式，请参见库表列名单个映射。 ■ 如需批量更改迁移对象在目标实例中的名称，请单击已选择对象方框右上方的  设置方式，请参见库表列名批量映射。  说明 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。
过滤待迁移数据	支持设置条件过滤数据，详情请参见 通过SQL条件过滤任务数据 。
增量迁移的SQL操作	选择增量迁移DDL或DML操作，请右击已选择对象中的迁移对象，在弹跳框中选择所需增量迁移的DML和DDL操作。支持的操作，请参见 支持增量迁移的SQL操作 。

高级配置

高级配置

设置告警: ?
☒ 不设置 ☐ 设置

目标库对象名称大小写策略: ?
DTS默认策略

源表DMS_ONLINE_DDL过程中是否复制临时表到目标库: ?
☐ 是 ☒ 否

源库、目标库无法连接后的重试时间: ?
- 120 + 分钟

上一步配置源库及目标库信息

下一步保存任务并预检查

保存并返回列表

取消

配置	说明
设置告警	是否设置告警，当迁移失败或延迟超过阈值后，将通知告警联系人。 ■ 不设置：不设置告警。 ■ 设置：设置告警，您还需要设置告警阈值和告警联系人。
目标库对象名称大小写策略	您可以配置目标实例中迁移对象的库名、表名和列名的英文大小写策略。默认情况下选择 DTS默认策略 ，您也可以选择与源库、目标库默认策略保持一致。更多信息，请参见 目标库对象名称大小写策略 。
源表DMS_ONLINE_DDL过程中是否复制临时表到目标库	如源库使用 数据管理DMS（Data Management Service） 执行Online DDL变更，您可以选择是否迁移Online DDL变更产生的临时表数据。 ■ 是：迁移Online DDL变更产生的临时表数据。 ? 说明 Online DDL变更产生的临时表数据过大，可能会导致迁移任务延迟。 ■ 否：不迁移Online DDL变更产生的临时表数据，只迁移源库的原始DDL数据。 ? 说明 该方案会导致目标库锁表。

配置	说明
源、目标库无法连接重试时间	默认重试720分钟，您也可以在取值范围（10~720分钟）内自定义重试时间，建议设置30分钟以上。如果DTS在设置的时间内重新连接上源、目标库，同步任务将自动恢复。否则，同步任务将失败。  说明 ■ 针对同源或者同目标的多个DTS实例，网络重试时间以后创建任务的设置为准。 ■ 由于连接重试期间，DTS将收取任务运行费用，建议您根据业务需要自定义重试时间，或者在源和目标库实例释放后尽快释放DTS实例。

6. 上述配置完成后，单击页面右下角的下一步保存任务并预检查。

 说明

- 在迁移任务正式启动之前，会先进行预检查。只有预检查通过后，才能成功启动迁移任务。
- 如果预检查失败，单击具体检查项后的，查看失败详情。
 - 您可以根据提示修复后重新进行预检查。
 - 如无需修复告警检测项，您也可以选择**确认屏蔽**、**忽略告警项**并重新进行预检查，跳过告警检测项重新进行预检查。

7. 预检查通过率显示为100%时，单击下一步购买。

8. 在购买页面，选择数据迁移实例的链路规格，详细说明请参见下表。

类别	参数	说明
信息配置	链路规格	DTS为您提供了不同性能的迁移规格，迁移链路规格的不同会影响迁移速率，您可以根据业务场景进行选择，详情请参见 数据迁移链路规格说明 。

9. 配置完成后，阅读并选中《数据传输（按量付费）服务条款》。

10. 单击**购买并启动**，迁移任务正式开始，您可在数据迁移界面查看具体进度。

3.3.4. 包年包月RDS迁移至PolarDB后申请转单优惠退款

当您的业务从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，您可以通过工单为RDS实例申请转单优惠退款，避免浪费闲置的RDS资源。在符合前提条件的情况下，以购买RDS实例时实际支付的价格为基数，根据RDS实例的剩余时长按比例为您办理退款。

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。具体迁移过程，可参见[一键升级RDS MySQL至PolarDB MySQL引擎](#)。

退款规则

以购买源RDS实例时实际支付的价格为基数，根据源RDS实例的剩余时长按比例退款。退款流向可参见[退订后资金流向](#)。

假设源RDS实例的包年价格原价是20000元，购买时长为一年，购买时打折后实际支付价格（订单金额）为10000元。使用半年后，若您完成业务从RDS迁移至PolarDB，并申请了RDS转单优惠退款，则您可以获得5000元的退款金额。

退款申请

请[提交工单](#)联系售后客服申请RDS转单优惠退款。

3.4. PolarDB间的数据迁移

3.4.1. PolarDB MySQL间迁移

通过数据传输服务DTS（Data Transmission Service），可以帮助您实现PolarDB MySQL引擎集群间的数据迁移。

 **说明** 目前PolarDB MySQL引擎集群暂不支持升级MySQL版本至8.0版本，您可以创建一个新的PolarDB MySQL引擎集群（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 MySQL引擎集群来测试兼容性，测试完成后可释放该集群。

前提条件

- 已创建源和目标PolarDB MySQL引擎集群，详情请参见[购买按量付费集群](#)和[购买包年包月集群](#)。
- 目标PolarDB MySQL引擎集群的存储空间须大于源PolarDB MySQL引擎集群占用的存储空间。

注意事项

类型	说明
----	----

类型	说明
源库限制	- 带宽要求：源库所属的服务器需具备足够出口带宽，否则将影响数据迁移速率。 - 待迁移的表需具备主键或唯一约束，且字段具有唯一性，否则可能会导致目标数据库中重复数据。 - 如迁移对象为表级别，且需进行编辑（如表列名映射），则单次迁移任务仅支持迁移至多1000张表。当超出数量限制，任务提交后会显示请求报错，此时建议您拆分待迁移的表，分批配置多个任务，或者配置整库的迁移任务。 - 如需进行增量迁移，Binlog日志： - 需开启，并且loose_polar_log_bin为on。否则预检查阶段提示报错，且无法成功启动数据迁移任务。 - 如为增量迁移任务，DTS要求源数据库的本地Binlog日志保存24小时以上，如为全量迁移和增量迁移任务，DTS要求源数据库的本地Binlog日志至少保留7天以上（您可在全量迁移完成后将Binlog保存时间设置为24小时以上），否则DTS可能因无法获取Binlog而导致任务失败，极端情况下甚至可能会导致数据不一致或丢失。由于您所设置的Binlog日志保存时间低于DTS要求的时间进而导致的问题，不在DTS的SLA保障范围内。 - 源库的操作限制： - 在库表结构迁移和全量迁移阶段，请勿执行库或表结构变更的DDL操作，否则数据迁移任务会失败。 - 如仅执行全量数据迁移，请勿向源实例中写入新的数据，否则会导致源和目标数据不一致。为实时保持数据一致性，建议选择结构迁移、全量数据迁移和增量数据迁移。
其他限制	- 建议源和目标PolarDB MySQL引擎的MySQL版本保持一致，以保障兼容性。 - 不支持迁移源PolarDB MySQL引擎只读节点。 - 执行数据迁移前需评估源库和目标库的性能，同时建议业务低峰期执行数据迁移。否则全量数据迁移时DTS占用源和目标库一定读写资源，可能会导致数据库的负载上升。 - 由于全量数据迁移会并发执行INSERT操作，导致目标数据库的表产生碎片，因此全量迁移完成后目标数据库的表存储空间会比源实例的表存储空间大。 - 请确认DTS对数据类型为FLOAT或DOUBLE的列的迁移精度是否符合业务预期。DTS会通过 <code>ROUND(COLUMN, PRECISION)</code> 来读取这两类列的值。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。 - DTS会尝试恢复七天之内迁移失败任务。因此业务切换至目标实例前，请务必结束或释放该任务，或者将DTS访问目标实例账号的写权限用 <code>revoke</code> 命令回收掉。避免该任务被自动恢复后，源端数据覆盖目标实例的数据。
其他注意事项	DTS会在源库定时执行 <code>CREATE DATABASE IF NOT EXISTS `test`</code> 命令以推进Binlog位点。

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

迁移类型	链路配置费用	公网流量费用
增量数据迁移	收费，详情请参见 产品价格 。	

迁移类型说明

- 库表结构迁移

DTS将源库中迁移对象的结构定义迁移到目标库。

说明

- 目前DTS支持结构迁移的对象为表、视图、触发器、存储过程和存储函数。
- 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。

- 全量迁移

DTS将源库中迁移对象的存量数据，全部迁移到目标库中。

- 增量迁移

DTS在全量迁移的基础上，将源库的增量更新数据迁移到目标库中。通过增量数据迁移可以实现在自建应用不停机的情况下，平滑地完成数据迁移。

支持增量迁移的SQL操作

操作类型	SQL操作语句
DML	INSERT、UPDATE、DELETE
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引擎集群	待迁移对象的读权限
目标PolarDB MySQL引擎集群	迁移对象的读写权限

PolarDB MySQL引擎集群请参见[创建数据库账号](#)。

操作步骤

1. 登录[新版DTS迁移任务的列表页面](#)。

 **说明** 您也可以登录[DMS数据管理服务](#)。在顶部菜单栏中，选择集成与开发（DTS）> 数据迁移。

2. 在页面左上角，选择迁移实例所属地域。



3. 单击**创建任务**，配置源库及目标库信息。

 **警告** 选择源和目标实例后，建议您仔细阅读页面上方显示的使用限制，以成功创建并执行迁移任务。

* 任务名称:

dtsspwp135e

❗ 使用限制

1.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。2.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。
更多限制，[请查看详情](#)

源库信息

选择已有的实例:

可以选择一个已有的实例进行快速配置

* 数据库类型: ②

DB2 iSeries(AS/400)

DB2 LUW

HBase

Mariadb

MongoDB

MySQL

Oracle

PolarDB MySQL

PolarDB Oracle

PolarDB PostgreSQL

PolarDB-X 1.0

PolarDB-X 2.0

PostgreSQL

Redis

SQLServer

Teradata

* 接入方式:

阿里云实例

* 实例地区:

华东1 (杭州)

是否跨阿里云账号: ②

不跨账号

跨账号

* PolarDB实例ID:

pc-xxxxxx

* 数据库账号: ②

dtstest

* 数据库密码:

将连接信息保存为实例

目标库信息

选择已有的实例:

可以选择一个已有的实例进行快速配置

* 数据库类型: ②

AnalyticDB MySQL 3.0

AnalyticDB PostgreSQL

DataHub

Kafka

MySQL

Oracle

PolarDB MySQL

PolarDB-X 2.0

* 接入方式:

阿里云实例

* 实例地区:

华东1 (杭州)

* PolarDB实例ID:

pc-xxxxxx

* 数据库账号: ②

dtstest

* 数据库密码:

将连接信息保存为实例

取消

测试连接以进行下一步

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。
	选择已有的实例	您可以按实际需求，选择是否使用已有实例。 - 如使用已有实例，数据库信息将自动填入，您无需重复输入。 - 如不使用已有实例，您需要输入下方的数据库信息。
	数据库类型	选择PolarDB MySQL。
	接入方式	选择阿里云实例。
	实例地区	选择源PolarDB MySQL引擎集群所属地域。

53

> 文档版本：20220713

类别	配置	说明
源库信息	PolarDB MySQL实例ID	填入源PolarDB MySQL引擎集群ID。
	数据库账号	填入源PolarDB MySQL引擎集群的数据库账号，权限要求请参见 数据库账号的权限要求 。
	数据库密码	填入该数据库账号对应的密码。
	将连接信息保存为实例或编辑连接信息模板	在选择已有的实例处是否选择了已有数据库的连接模板。 i. 是：您可以单击编辑连接信息模板自定义连接信息模板的名称。  说明 修改后的连接信息模板，将在下一次选择该连接模板作为实例连接信息时生效，不会影响原先已使用该模板配置的实例连接信息。 ii. 否：单击将连接信息保存为实例并在弹跳框中设置连接名称，可将该实例的连接信息保存为模板，便于下次自动传入。  说明 建议连接名称配置为具有业务意义的名称（无唯一性要求），便于后续识别。
目标库信息	选择已有的实例	您可以按实际需求，选择是否使用已有实例。 - 如使用已有实例，数据库信息将自动填入，您无需重复输入。 - 如不使用已有实例，您需要输入下方的数据库信息。
	数据库类型	选择PolarDB MySQL。
	接入方式	选择阿里云实例。
	实例地区	选择目标PolarDB MySQL引擎集群所属地域。
	PolarDB MySQL实例ID	选择目标PolarDB MySQL引擎集群ID。
	数据库账号	填入目标PolarDB MySQL引擎集群的数据库账号，权限要求请参见 数据库账号的权限要求 。
	数据库密码	填入该数据库账号对应的密码。

类别	配置	说明
	将连接信息保存为实例或编辑连接信息模板	在选择已有的实例处是否选择了已有数据库的连接模板。 i. 是：您可以单击编辑连接信息模板自定义连接信息模板的名称。 ② 说明 修改后的连接信息模板，将在下一次选择该连接模板作为实例连接信息时生效，不会影响原先已使用该模板配置的实例连接信息。 ii. 否：单击将连接信息保存为实例并在弹跳框中设置连接名称，可将该实例的连接信息保存为模板，便于下次自动传入。 ② 说明 建议连接名称配置为具有业务意义的名称（无唯一性要求），便于后续识别。

4. 配置完成后，单击页面下方的**测试连接**以进行下一步。

警告

- 如果源或目标数据库是阿里云数据库实例（例如RDS MySQL、云数据库MongoDB版等）或ECS上的自建数据库，DTS会自动将对应地区DTS服务的IP地址添加到阿里云数据库实例的白名单或ECS的安全规则中，您无需手动添加，请参见[DTS服务器的IP地址段](#)；如果源或目标数据库是IDC自建数据库或其他云数据库，则需要您手动添加对应地区DTS服务的IP地址，以允许来自DTS服务器的访问。
- 上述场景中，DTS自动添加或您手动添加DTS服务的公网IP地址段可能会存在安全风险，一旦使用本产品代表您已理解和确认其中可能存在的安全风险，并且需要您做好基本的安全防护，包括但不限于加强账号密码强度防范、限制各网段开放的端口号、内部各API使用鉴权方式通信、定期检查并限制不需要的网段，或者使用通过内网（专线/VPN网关/智能网关）的方式接入。
- DTS任务完成或释放后，建议您手动检测并删除DTS相关的服务器IP地址段。

5. 配置任务对象及高级配置。

◦ 基础配置

基础配置

* 任务步骤:

☒ 库表结构迁移

☒ 全量迁移

☒ 增量迁移

① 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。

如果需要进行长期的数据实时同步，请使用数据同步功能。

注：增量迁移不支持trigger的同步[帮助文档](#)

* 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

* 同步对象:

自定义选择

源库对象

Q 支持正则，若全局搜索，请先展开树

> ☐

> ☐

> ☐

> ☐

> ☒ dtstestdata

> ☐

> ☐

已选择对象

批量编辑 ?

Q 支持正则，若全局搜索，请先展开树

☐ dtstestdata

【右键】选择对象
可以进行重命名或
DML、DDL配置

配置	说明
任务步骤	■ 如果只需要进行全量迁移，请同时选中库表结构迁移和全量迁移。 ■ 如果需要进行不停机迁移，请同时选中库表结构迁移、全量迁移和增量迁移。 ② 说明 如果未选择增量迁移，为保障数据一致性，数据迁移期间请勿在源实例中写入新的数据。

> 文档版本：20220713

56

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，数据迁移任务不会被启动。 ? 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，请参见库表列名映射。 ■ 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，则会保留目标库中的该条记录，即源库中的该条记录不会迁移至目标库中。 ■ 表结构不一致的情况下，可能导致只能迁移部分列的数据或迁移失败。
同步对象	在同步对象框中单击待迁移的对象，然后单击  将其移动到已选择对象框。 ? 说明
映射名称更改	■ 如需更改单个迁移对象在目标实例中的名称，请右击已选择对象中的迁移对象，设置方式，请参见库表列名单个映射。 ■ 如需批量更改迁移对象在目标实例中的名称，请单击已选择对象方框右上方的  设置方式，请参见库表列名批量映射。 ? 说明 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。
过滤待迁移数据	支持设置条件过滤数据，详情请参见 通过SQL条件过滤任务数据 。
增量迁移的SQL操作	选择增量迁移DDL或DML操作，请右击已选择对象中的迁移对象，在弹跳框中选择所需增量迁移的DML和DDL操作。支持的操作，请参见 支持增量迁移的SQL操作 。

高级配置

高级配置

设置告警: ?

☒ 不设置

☐ 设置

源表DMS_ONLINE_DDL过程中是否复制临时表到目标库: ?

☐ 是

☒ 否

源库、目标库无法连接后的重试时间: ?

-120+

分钟

上一步配置源库及目标库信息

下一步保存任务并预检查

保存并返回列表

取消

配置	说明
设置告警	是否设置告警，当迁移失败或延迟超过阈值后，将通知告警联系人。 ■ 不设置：不设置告警。 ■ 设置：设置告警，您还需要设置告警阈值和告警联系人。
源表 DMS_ONLINE_DDL 过程中是否复制临时表到目标库	如源库使用数据管理DMS（Data Management Service）执行Online DDL变更，您可以选择是否迁移Online DDL变更产生的临时表数据。 ■ 是：迁移Online DDL变更产生的临时表数据。 ? 说明 Online DDL变更产生的临时表数据过大，可能会导致迁移任务延迟。 ■ 否：不迁移Online DDL变更产生的临时表数据，只迁移源库的原始DDL数据。 ? 说明 该方案会导致目标库锁表。
源、目标库无法连接重试时间	默认重试720分钟，您也可以在取值范围（10~720分钟）内自定义重试时间，建议设置30分钟以上。如果DTS在设置的时间内重新连接上源、目标库，同步任务将自动恢复。否则，同步任务将失败。 ? 说明 ■ 针对同源或者同目标的多个DTS实例，网络重试时间以后创建任务的设置为准。 ■ 由于连接重试期间，DTS将收取任务运行费用，建议您根据业务需要自定义重试时间，或者在源和目标库实例释放后尽快释放DTS实例。

6. 上述配置完成后，单击页面右下角的下一步保存任务并预检查。

说明

- 在迁移任务正式启动之前，会先进行预检查。只有预检查通过后，才能成功启动迁移任务。
- 如果预检查失败，单击具体检查项后的，查看失败详情。
 - 您可以根据提示修复后重新进行预检查。
 - 如无需修复告警检测项，您也可以选择**确认屏蔽**、**忽略告警项**并重新进行预检查，跳过告警检测项重新进行预检查。

7. 预检查通过率显示为100%时，单击**下一步购买**。

8. 在**购买**页面，选择数据迁移实例的链路规格，详细说明请参见下表。

类别	参数	说明
信息配置	链路规格	DTS为您提供不同性能的迁移规格，迁移链路规格的不同会影响迁移速率，您可以根据业务场景进行选择，详情请参见 数据迁移链路规格说明 。

9. 配置完成后，阅读并选中《**数据传输（按量付费）服务条款**》。

10. 单击**购买并启动**，迁移任务正式开始，您可在数据迁移界面查看具体进度。

3.5. 从其它数据库迁移至PolarDB

3.5.1. 自建MySQL迁移至PolarDB MySQL

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引擎集群。

支持的源数据库

MySQL与PolarDB MySQL引擎集群间的迁移，支持源数据库MySQL为以下类型。本文以有公网IP的自建数据库为例介绍配置流程，其他类型的源数据库配置流程与本案例类似。

- RDS MySQL实例。
- 以下类型的自建数据库：
 - 有公网IP的自建数据库。
 - ECS上的自建数据库。
 - 通过专线、VPN网关或智能网关接入的自建数据库。
 - 通过数据库网关接入的自建数据库。

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。
- 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[迁移、同步或订阅本地数据库时需添加的IP白名单](#)。
- 已创建目标PolarDB MySQL引擎集群，详情请参见[购买按量付费集群](#)和[购买包年包月集群](#)。
- PolarDB MySQL引擎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

类型	说明
源库限制	- 带宽要求：源库所属的服务器需具备足够出口带宽，否则将影响数据迁移速率。 - 待迁移的表需具备主键或唯一约束，且字段具有唯一性，否则可能会导致目标数据库中出現重复数据。 - 如迁移对象为表级别，且需进行编辑（如表列名映射），则单次迁移任务仅支持迁移至多1000张表。当超出数量限制，任务提交后会显示请求报错，此时建议您拆分待迁移的表，分批配置多个任务，或者配置整库的迁移任务。 - 如需进行增量迁移，Binlog日志： - 需开启，并且binlog_format为row、binlog_row_image为full。否则预检查阶段提示报错，且无法成功启动数据迁移任务。  注意 如源实例自建MySQL是双主集群（两者互为主从），为保障DTS能获取全部的Binlog日志，则您需开启参数log_slave_updates。 - 如为增量迁移任务，DTS要求源数据库的本地Binlog日志保存24小时以上，如为全量迁移和增量迁移任务，DTS要求源数据库的本地Binlog日志至少保留7天以上（您可在全量迁移完成后将Binlog保存时间设置为24小时以上），否则DTS可能因无法获取Binlog而导致任务失败，极端情况下甚至可能会导致数据不一致或丢失。由于您所设置的Binlog日志保存时间低于DTS要求的时间进而导致的问题，不在DTS的SLA保障范围内。 - 源库的操作限制： - 在库表结构迁移和全量迁移阶段，请勿执行库或表结构变更的DDL操作，否则数据迁移任务会失败。 - 如仅执行全量数据迁移，请勿向源实例中写入新的数据，否则会导致源和目标数据不一致。为实时保持数据一致性，建议选择结构迁移、全量数据迁移和增量数据迁移。
其他限制	- 建议源和目标库的MySQL版本保持一致，以保障兼容性。 - 执行数据迁移前需评估源库和目标库的性能，同时建议业务低峰期执行数据迁移。否则全量数据迁移时DTS占用源和目标库一定读写资源，可能会导致数据库的负载上升。 - 由于全量数据迁移会并发执行INSERT操作，导致目标数据库的表产生碎片，因此全量迁移完成后目标数据库的表存储空间会比源实例的表存储空间大。 - 请确认DTS对数据类型为FLOAT或DOUBLE的列的迁移精度是否符合业务预期。DTS会通过 <code>ROUND (COLUMN, PRECISION)</code> 来读取这两类列的值。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。 - DTS会尝试恢复七天之内迁移失败任务。因此业务切换至目标实例前，请务必结束或释放该任务，或者将DTS访问目标实例账号的写权限用 <code>revoke</code> 命令回收掉。避免该任务被自动恢复后，源端数据覆盖目标实例的数据。

类型	说明
特殊情况	- 当源库为自建MySQL时 - 迁移时源库进行主备切换，会导致迁移任务失败。 - 由于DTS的延迟时间是根据迁移到目标库最后一条数据的时间戳和当前时间戳对比得出，源库长时间未执行DML操作可能导致延迟信息不准确。如果任务显示的延迟时间过大，您可以在源库执行一个DML操作来更新延迟信息。  说明 如果迁移对象选择为整库，您还可以创建心跳表，心跳表每秒定期更新或者写入数据。 - DTS会在源库定时执行CREATE DATABASE IF NOT EXISTS `test`命令以推进Binlog位点。 - 当目标库为PolarDB MySQL时 DTS会自动在PolarDB MySQL中创建数据库，如果待迁移的数据库名称不符合PolarDB MySQL的定义规范，您需要在配置迁移任务之前在PolarDB MySQL中创建数据库。相关操作，请参见数据库管理。

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 产品定价 。
增量数据迁移	收费，详情请参见 产品定价 。	

迁移类型说明

库表结构迁移

DTS将源库中迁移对象的结构定义迁移到目标库。

说明

- 目前DTS支持结构迁移的对象为表、视图、触发器、存储过程和存储函数。
- 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。

全量迁移

DTS将源库中迁移对象的存量数据，全部迁移到目标库中。

增量迁移

DTS在全量迁移的基础上，将源库的增量更新数据迁移到目标库中。通过增量数据迁移可以实现在自建应用不停机的情况下，平滑地完成数据迁移。

支持增量迁移的SQL操作

操作类型	SQL操作语句
DML	INSERT、UPDATE、DELETE

操作类型	SQL操作语句
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	库表结构迁移	全量迁移	增量迁移
自建MySQL数据库	SELECT 权限	SELECT 权限	增量数据迁移：待迁移对象的 SELECT 权限 REPLICATION CLIENT、 REPLICATION SLAVE、SHOW VIEW 建库建表的权限，以允许DTS创建库 dts，用于记录迁移期间的心跳数据
PolarDB MySQL引擎集群	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[自建MySQL创建账号并设置binlog](#)。
- PolarDB MySQL引擎集群请参见[创建数据库账号](#)。

操作步骤

1. 登录[新版DTS迁移任务的列表页面](#)。

 **说明** 您也可以登录[DMS数据管理服务](#)。在顶部菜单栏中，选择集成与开发（DTS）> 数据迁移。

2. 在页面左上角，选择迁移实例所属地域。



3. 单击**创建任务**，配置源库及目标库信息。

 **警告** 选择源和目标实例后，建议您仔细阅读页面上方显示的使用限制，以成功创建并执行迁移任务。

* 任务名称:

dtstzw4qnz6

❗ 使用限制

1.待迁移的表需具备主键或唯一约束，且字段具有唯一性，否则可能会导致目标数据库中出現重复数据。2.如需进行增量迁移，Binlog日志需开启且至少保留24小时（建议3天以上）。3.请勿在链路创建阶段执行库或表结构变更的DDL操作，否则数据迁移链路建立会失败。4.如仅执行全量数据迁移，请勿向源实例中写入新的数据，否则会导致源和目标数据不一致。为实时保持数据一致性，建议选择结构迁移、全量数据迁移和增量数据迁移。5.迁移时源库进行主备切换，会导致迁移任务失败。6.如果任务显示的延迟时间过大，您可以在源库执行一个DML操作来更新延迟信息。
更多限制, 请查看详情

源库信息

选择已有的实例:

可以选择一个已有的实例进行快速配置

* 数据库类型: ①

DB2 iSeries(AS/400)

DB2 LUW

HBase

Mariadb

MongoDB

MySQL

Oracle

PolarDB MySQL

PolarDB Oracle

PolarDB PostgreSQL

PolarDB-X 1.0

PolarDB-X 2.0

PostgreSQL

Redis

SQLServer

Teradata

* 接入方式:

阿里云实例

专线/VPN网关/智能网关

公网IP

ECS自建数据库

云企业网CEN

数据库网关DG

* 实例地区:

华东1（杭州）

* 主机名或IP地址:

* 端口:

3306

* 数据库账号: ②

dtstest

* 数据库密码:

将连接信息保存为实例

目标库信息

选择已有的实例:

可以选择一个已有的实例进行快速配置

* 数据库类型: ①

AnalyticDB MySQL 3.0

AnalyticDB PostgreSQL

DataHub

DB2 LUW

ElasticSearch

Kafka

MySQL

Oracle

PolarDB MySQL

PolarDB-X 2.0

* 接入方式:

阿里云实例

* 实例地区:

华东1（杭州）

* PolarDB实例ID:

pc-*****

* 数据库账号: ②

dtstest

* 数据库密码:

将连接信息保存为实例

取消

测试连接以进行下一步

类别	配置	说明
无	任务名称	DT S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。
	数据库类型	选择MySQL。
	接入方式	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，详情请参见准备工作概览。

63

> 文档版本：20220713

类别	配置	说明
源库信息	实例地区	选择源自建MySQL数据库所属地域。
	主机名或IP地址	填入源自建MySQL数据库的访问地址，本案例中填入公网地址。
	端口	填入源自建MySQL数据库的服务端口（需开放至公网），默认为3306。
	数据库账号	填入源自建MySQL数据库的账号，权限要求请参见 数据库账号的权限要求 。
	数据库密码	填入该数据库账号对应的密码。
目标库信息		
	数据库类型	选择PolarDB MySQL。
	接入方式	选择阿里云实例。
	实例地区	选择目标PolarDB MySQL引擎集群所属地域。
	PolarDB MySQL实例ID	选择目标PolarDB MySQL引擎集群ID。
	数据库账号	填入目标PolarDB MySQL引擎集群的数据库账号，权限要求请参见 数据库账号的权限要求 。
	数据库密码	填入该数据库账号对应的密码。

4. 配置完成后，单击页面下方的测试连接以进行下一步。

警告

- 如果源或目标数据库是阿里云数据库实例（例如RDS MySQL、云数据库MongoDB版等）或ECS上的自建数据库，DTS会自动将对应地区DTS服务的IP地址添加到阿里云数据库实例的白名单或ECS的安全规则中，您无需手动添加，请参见[DTS服务器的IP地址段](#)；如果源或目标数据库是IDC自建数据库或其他云数据库，则需要您手动添加对应地区DTS服务的IP地址，以允许来自DTS服务器的访问。
- 上述场景中，DTS自动添加或您手动添加DTS服务的公网IP地址段可能会存在安全风险，一旦使用本产品代表您已理解和确认其中可能存在的安全风险，并且需要您做好基本的安全防护，包括但不限于加强账号密码强度防范、限制各网段开放的端口号、内部各API使用鉴权方式通信、定期检查并限制不需要的网段，或者使用通过内网（专线/VPN网关/智能网关）的方式接入。
- DTS任务完成或释放后，建议您手动检测并删除DTS相关的服务器IP地址段。

5. 如果您的自建数据库具备白名单安全设置，您需要复制弹跳框中的DTS服务器IP地址，并加入自建数据库的白名单安全设置中。然后单击测试连接以进行下一步。

 警告

- 如果源或目标数据库是阿里云数据库实例（例如RDS MySQL、云数据库MongoDB版等）或ECS上的自建数据库，DTS会自动将对应地区DTS服务的IP地址添加到阿里云数据库实例的白名单或ECS的安全规则中，您无需手动添加，请参见[DTS服务器的IP地址段](#)；如果源或目标数据库是IDC自建数据库或其他云数据库，则需要您手动添加对应地区DTS服务的IP地址，以允许来自DTS服务器的访问。
- 上述场景中，DTS自动添加或您手动添加DTS服务的公网IP地址段可能会存在安全风险，一旦使用本产品代表您已理解和确认其中可能存在的安全风险，并且需要您做好基本的安全防护，包括但不限于加强账号密码强度防范、限制各网段开放的端口号、内部各API使用鉴权方式通信、定期检查并限制不需要的网段，或者使用通过内网（专线/VPN网关/智能网关）的方式接入。
- DTS任务完成或释放后，建议您手动检测并删除DTS相关的服务器IP地址段。

6. 配置任务对象及高级配置。

○ 基础配置

基础配置

* 任务步骤:

☒ 库表结构迁移

☒ 全量迁移

☒ 增量迁移

① 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。

如果需要进行长期的数据实时同步，请使用数据同步功能。

注：增量迁移不支持trigger的同步[帮助文档](#)

* 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

* 同步对象:

自定义选择

源库对象

Q 支持正则，若全局搜索，请先展开树

> ☐

> ☐

> ☐

> ☐

> ☒ dtstestdata

> ☐

> ☐

已选择对象

批量编辑 ?

Q 支持正则，若全局搜索，请先展开树

☐ dtstestdata

【右键】选择对象
可以进行重命名或
DML、DDL配置

配置

说明

配置	说明
任务步骤	- 如果只需要进行全量迁移，请同时选中库表结构迁移和全量迁移。 - 如果需要进行不停机迁移，请同时选中库表结构迁移、全量迁移和增量迁移。  说明 如果未选择增量迁移，为保障数据一致性，数据迁移期间请勿在源实例中写入新的数据。
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，数据迁移任务不会被启动。  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，请参见库表列名映射。 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，则会保留目标库中的该条记录，即源库中的该条记录不会迁移至目标库中。 - 表结构不一致的情况下，可能导致只能迁移部分列的数据或迁移失败。
同步对象	在同步对象框中单击待迁移的对象，然后单击  将其移动到已选择对象框。  说明
映射名称更改	- 如需更改单个迁移对象在目标实例中的名称，请右击已选择对象中的迁移对象，设置方式，请参见库表列名单个映射。 - 如需批量更改迁移对象在目标实例中的名称，请单击已选择对象方框右上方的  设置方式，请参见库表列名批量映射。  说明 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。
过滤待迁移数据	支持设置条件过滤数据，详情请参见 通过SQL条件过滤任务数据 。

配置	说明
增量迁移的SQL操作	选择增量迁移DDL或DML操作，请右击已选择对象中的迁移对象，在弹跳框中选择所需增量迁移的DML和DDL操作。支持的操作，请参见 支持增量迁移的SQL操作 。

高级配置

高级配置

设置告警: ?

☒ 不设置 ☐ 设置

目标库对象名称大小写策略: ?

DTS默认策略

源表DMS_ONLINE_DDL过程中是否复制临时表到目标库: ?

☐ 是 ☒ 否

源库、目标库无法连接后的重试时间: ?

- 120 +

分钟

上一步配置源库及目标库信息

下一步保存任务并预检查

保存并返回列表

取消

配置	说明
设置告警	是否设置告警，当迁移失败或延迟超过阈值后，将通知告警联系人。 ■ 不设置：不设置告警。 ■ 设置：设置告警，您还需要设置告警阈值和告警联系人。
目标库对象名称大小写策略	您可以配置目标实例中迁移对象的库名、表名和列名的英文大小写策略。默认情况下选择DTS默认策略，您也可以选择与源库、目标库默认策略保持一致。更多信息，请参见目标库对象名称大小写策略。
源表DMS_ONLINE_DDL过程中是否复制临时表到目标库	如源库使用数据管理DMS（Data Management Service）执行Online DDL变更，您可以选择是否迁移Online DDL变更产生的临时表数据。 ■ 是：迁移Online DDL变更产生的临时表数据。 ? 说明 Online DDL变更产生的临时表数据过大，可能会导致迁移任务延迟。 ■ 否：不迁移Online DDL变更产生的临时表数据，只迁移源库的原始DDL数据。 ? 说明 该方案会导致目标库锁表。

配置	说明
源、目标库无法连接重试时间	默认重试720分钟，您也可以在取值范围（10~720分钟）内自定义重试时间，建议设置30分钟以上。如果DTS在设置的时间内重新连接上源、目标库，同步任务将自动恢复。否则，同步任务将失败。  说明 ■ 针对同源或者同目标的多个DTS实例，网络重试时间以后创建任务的设置为准。 ■ 由于连接重试期间，DTS将收取任务运行费用，建议您根据业务需要自定义重试时间，或者在源和目标库实例释放后尽快释放DTS实例。

7. 上述配置完成后，单击页面右下角的下一步保存任务并预检查。

 说明

- 在迁移任务正式启动之前，会先进行预检查。只有预检查通过后，才能成功启动迁移任务。
- 如果预检查失败，单击具体检查项后的，查看失败详情。
 - 您可以根据提示修复后重新进行预检查。
 - 如无需修复告警检测项，您也可以选择确认屏蔽、忽略告警项并重新进行预检查，跳过告警检测项重新进行预检查。

8. 预检查通过率显示为100%时，单击下一步购买。

9. 在购买页面，选择数据迁移实例的链路规格，详细说明请参见下表。

类别	参数	说明
信息配置	链路规格	DTS为您提供了不同性能的迁移规格，迁移链路规格的不同会影响迁移速率，您可以根据业务场景进行选择，详情请参见 数据迁移链路规格说明 。

10. 配置完成后，阅读并选中《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，您可在数据迁移界面查看具体进度。

3.5.2. 从自建MySQL迁移至PolarDB MySQL引擎（mysqldump工具）

本文介绍如何使用mysqldump工具将自建MySQL数据库迁移至PolarDB MySQL引擎。

前提条件

目标PolarDB MySQL引擎集群需已完成如下操作：

- [创建数据库](#)
- [创建数据库账号](#)

- 设置白名单
- 申请公网连接地址

迁移方式对比

您可以通过mysqldump或DTS工具将自建MySQL数据库迁移至PolarDB MySQL引擎，下表对比了两种迁移方式的差异供您参考。

对比项	mysqldump	DTS
自建MySQL数据库版本	无限制	自建MySQL数据库版本为5.1、5.5、5.6、5.7或8.0版本。
是否支持结构迁移和全量数据迁移	支持	支持
是否支持增量数据迁移	不支持	支持
是否支持不停机迁移	不支持	支持

② 说明 使用DTS工具进行数据迁移的操作步骤，请参见[从自建MySQL迁移至PolarDB MySQL](#)。

注意事项

迁移后的表名不区分大小写，统一变为小写。

操作步骤

② 说明 本文以自建MySQL 8.0版本为例，在Linux系统上演示相关操作步骤。

1. 使用mysqldump导出自建数据库的数据、存储过程、触发器和函数。

② 说明 导出期间请勿进行数据更新，耐心等待导出完成。

- i. 在Linux命令行下导出自建数据库的数据，命令如下：

② 说明 针对自建数据库的连接地址：

- 若自建MySQL数据库部署在ECS实例上，请填入 127.0.0.1 。
- 若自建MySQL数据库部署在本地，请填入该数据库的公网连接地址。

```
mysqldump -h <自建数据库的连接地址> -u root -p --opt --default-character-set=utf8 --hex-blob <自建数据库名> --skip-triggers --skip-lock-tables > /tmp/<自建数据库名>.sql
```

示例

```
mysqldump -h 127.0.0.1 -u root -p --opt --default-character-set=utf8 --hex-blob testdb --skip-triggers --skip-lock-tables > /tmp/testdb.sql
```

- ii. (可选) 在Linux命令行下导出存储过程、触发器和函数, 命令如下:

 **说明** 若数据库中没有使用存储过程、触发器和函数, 可跳过该步骤。

```
mysqldump -h 127.0.0.1 -u root -p --opt --default-character-set=utf8 --hex-blob <自建数据库名> -R | sed -e 's/DEFINER[ ]*=[ ]*[^]*\*/\*/' > /tmp/<自建数据库名>Trigger.sql
```

示例

```
mysqldump -h 127.0.0.1 -u root -p --opt --default-character-set=utf8 --hex-blob testdb -R | sed -e 's/DEFINER[ ]*=[ ]*[^]*\*/\*/' > /tmp/testdbTrigger.sql
```

2. 将导出的文件导入到目标PolarDB集群中, 命令如下:

```
mysql -h <PolarDB集群连接地址> -P <PolarDB集群端口> -u <PolarDB集群账号> -p <PolarDB数据库名称> < /tmp/<自建数据库名>.sql  
mysql -h <PolarDB集群连接地址> -P <PolarDB集群端口> -u <PolarDB集群账号> -p <PolarDB数据库名称> < /tmp/<自建数据库名>Trigger.sql
```

示例

```
mysql -h polardbtest.mysql.polardb.rds.aliyuncs.com -P 3306 -u testuser -p testdb < /tmp/testdb.sql  
mysql -h polardbtest.mysql.polardb.rds.aliyuncs.com -P 3306 -u testuser -p testdb < /tmp/testdbTrigger.sql
```

 **说明**

- PolarDB数据库名称需要是PolarDB集群上已创建的数据库。创建数据库操作, 请参见[创建数据库](#)。
- PolarDB集群账号需要是高权限账号或具有读写权限的账号。

3. 导入成功后, 您可以登录PolarDB集群数据库中查看数据是否正常。具体操作, 请参见[连接数据库集群](#)。

常见问题

Q: Access denied; you need (at least one of) the SUPER privilege(s) for this operation 报错怎么解决?

A: SQL脚本里面包括SUPER权限的语句, 将相关语句删除再执行。

3.5.3. 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 产品定价 。
增量数据迁移	收费，详情请参见 产品定价 。	

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

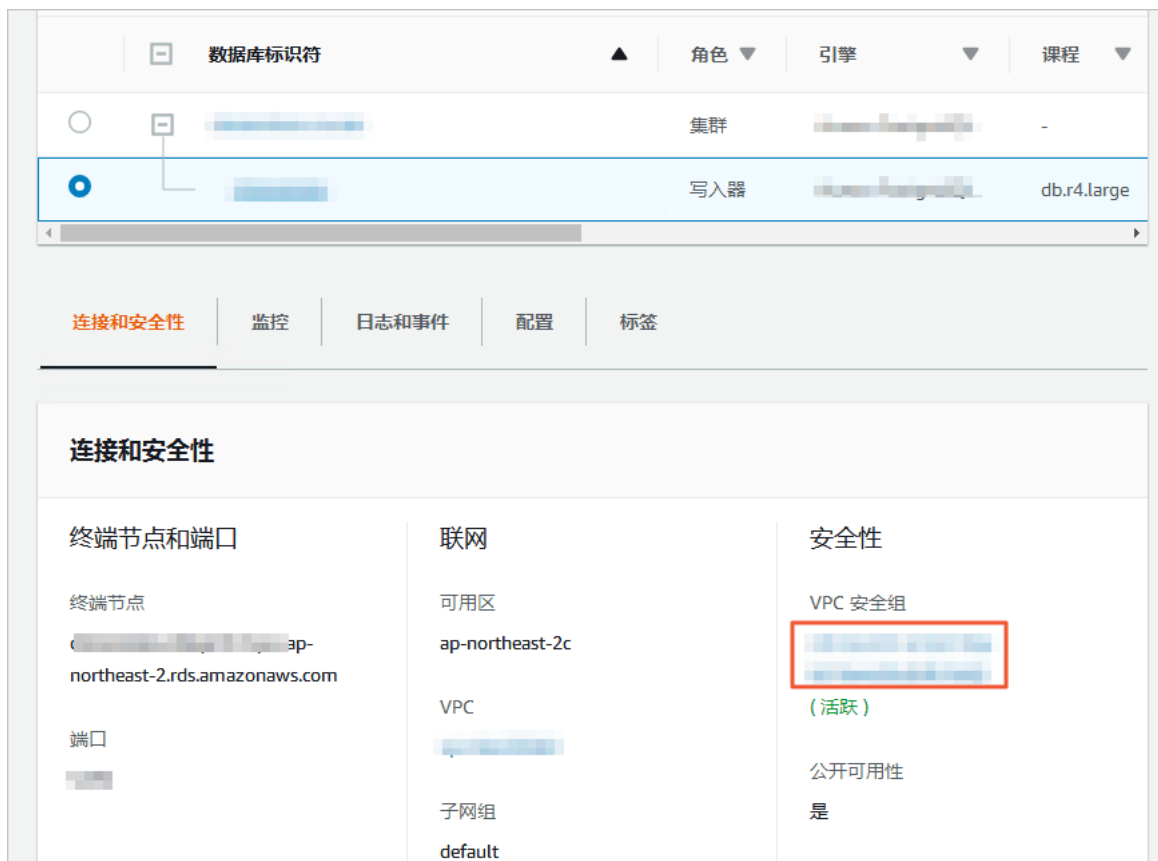
数据库	结构迁移	全量迁移	增量迁移
Amazon Aurora MySQL	迁移对象的SELECT权限	迁移对象的SELECT权限	REPLICATION SLAVE、REPLICATION CLIENT、SHOW VIEW和对迁移对象执行SELECT操作的权限
阿里云PolarDB MySQL	迁移对象的读写权限	迁移对象的读写权限	迁移对象的读写权限

数据库账号创建及授权方法：

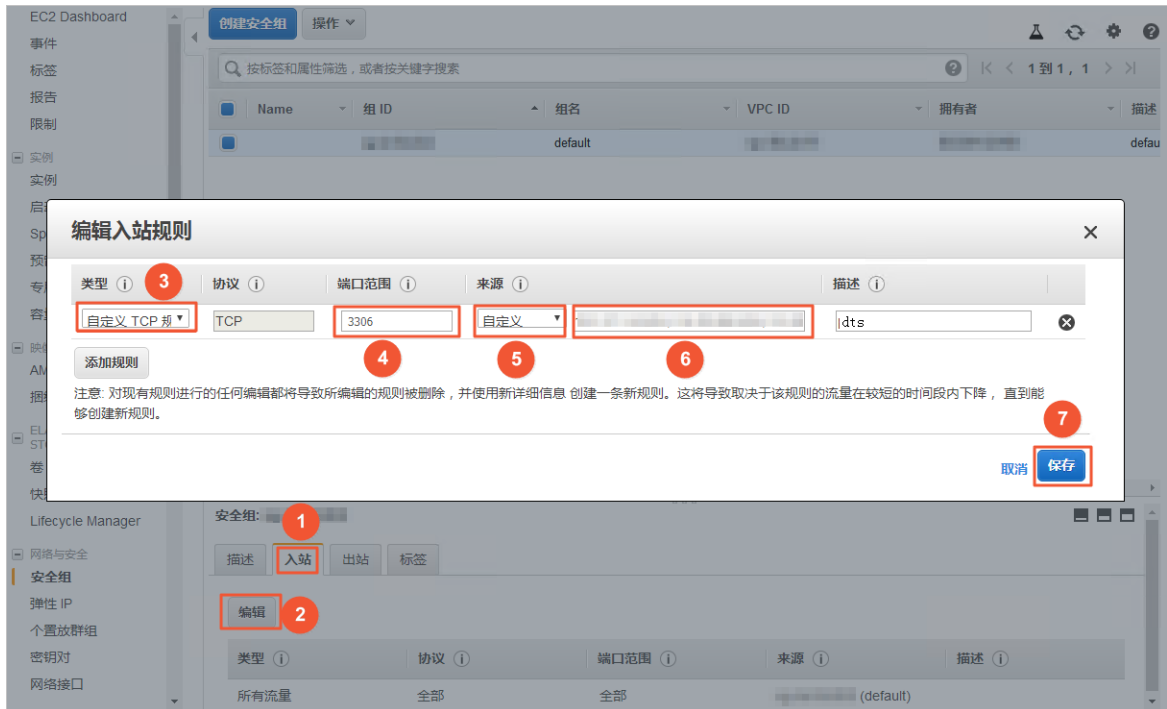
- Amazon Aurora MySQL请参见[自建MySQL创建账号并设置binlog](#)中创建账号的部分。
- 阿里云PolarDB MySQL请参见[创建账号](#)。

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。



- 在安全组设置页面，将对应区域的DTS服务器地址添加至入站规则中，IP地址段详情请参见[迁移、同步或订阅本地数据库时需添加的IP白名单](#)。



说明

- 您只需添加目标数据库所在区域对应的DTS IP地址段。例如，源数据库地区为新加坡，目标数据库地区为杭州，您只需要添加杭州地区的DTS IP地址段。
- 在加入IP地址段时，您可以一次性添加所需的IP地址，无需逐条添加入站规则。

- 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

- 登录[数据传输控制台](#)。
- 在左侧导航栏，单击[数据迁移](#)。
- 在迁移任务列表页面顶部，选择迁移的目标集群所属地域。
- 单击页面右上角的[创建迁移任务](#)。
- 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: 12.34.56.78

* 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-bp1234567890

* 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。
	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。 Amazon RDS 控制面板 数据库 Performance Insights 快照 Automated backups 预留实例 子网组 参数组 选项组 事件 事件订阅 Recommendations 相关 数据库标识符 角色 引擎 课程 状态 CPU 集群 实例 写入器 数据库 db.t2.small 8.50% 连接和安全性 监控 日志和事件 配置 标签 连接和安全性 终端节点和端口 终端节点 1.rds.amazonaws.com 端口 3306 联网 可用区 ap-southeast-1a VPC vpc- 子网组 default 安全性 VPC 安全组 rs-... (选择) 公开可用性 是 证书颁发机构

类别	配置	说明
	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.高级配置4.预检查

* 迁移类型：☒ 结构迁移 ☒ 全量数据迁移 ☒ 增量数据迁移

注：增量迁移不支持trigger的同步，详情请[参考文档](#)

注：DTS全量任务运行期间，不要清理DTS任务启动后源库产生的增量数据日志。源库如果过早清理日志，可能会导致DTS增量任务失败

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

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

* 源库、目标库无法连接后的重试时间

720

分钟

* 源表DMS_ONLINE_DDL过程中是否复制临时表到目标库：

☐ 是 ☒ 否

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，不会对源数据库数据（结构）造成影响。

2. 在做结构和全量迁移期间不要做DDL操作，否则可能导致任务失败。

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。

> 文档版本：20220713

76

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，详情请参见库表列映射。 - 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。
映射名称更改	如需更改迁移对象在目标实例中的名称，请使用对象名映射功能，详情请参见 库表列映射 。
源、目标库无法连接重试时间	默认重试12小时，您也可以自定义重试时间。如果DTS在设置的时间内重新连接上源、目标库，迁移任务将自动恢复。否则，迁移任务将失败。  说明 由于连接重试期间，DTS将收取任务运行费用，建议您根据业务需要自定义重试时间，或者在源和目标库实例释放后尽快释放DTS实例。
源表DMS_ONLINE_DDL过程中是否复制临时表到目标库	如源库使用数据管理DMS（Data Management Service）执行Online DDL变更，您可以选择是否迁移Online DDL变更产生的临时表数据。 - 是：迁移Online DDL变更产生的临时表数据。  说明 Online DDL变更产生的临时表数据过大，可能会导致迁移任务延迟。 - 否：不迁移Online DDL变更产生的临时表数据，只迁移源库的原始DDL数据。  说明 该方案会导致目标库锁表。

8. 上述配置完成后，单击页面右下角的**预检查并启动**。

 说明

- 在迁移任务正式启动之前，会先进行预检查。只有预检查通过后，才能成功启动迁移任务。
- 如果预检查失败，单击具体检查项后的 ，查看失败详情。
 - 您可以根据提示修复后重新进行预检查。
 - 如无需修复告警检测项，您也可以选择**确认屏蔽**、**忽略告警项**并重新进行预检查，跳过告警检测项重新进行预检查。

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. 将业务切换至阿里云PolarDB MySQL。

3.6. 从PolarDB迁移至其它数据库

3.6.1. PolarDB MySQL迁移至RDS MySQL

通过数据传输服务DTS（Data Transmission Service），您可以将PolarDB MySQL迁移至MySQL（包括自建MySQL、RDS MySQL实例）。

支持的目标数据库

PolarDB MySQL迁移至MySQL，支持目标数据库为以下类型。本文以RDS MySQL实例为例介绍配置流程，其他类型的目标数据库的配置流程与本案例类似。

- RDS MySQL实例。
- 以下类型的自建数据库：
 - 有公网IP的自建数据库。
 - ECS上的自建数据库。
 - 通过专线、VPN网关或智能网关接入的自建数据库。
 - 通过数据库网关接入的自建数据库。

前提条件

- 已创建源PolarDB MySQL引擎集群，详情请参见[购买按量付费集群](#)和[购买包年包月集群](#)。
- 目标RDS MySQL实例的存储空间须大于源集群占用的存储空间。

注意事项

类型	说明
----	----

类型	说明
源库限制	- 带宽要求：源库所属的服务器需具备足够出口带宽，否则将影响数据迁移速率。 - 待迁移的表需具备主键或唯一约束，且字段具有唯一性，否则可能会导致目标数据库中出现重复数据。 - 如迁移对象为表级别，且需进行编辑（如表列名映射），则单次迁移任务仅支持迁移至多1000张表。当超出数量限制，任务提交后会显示请求报错，此时建议您拆分待迁移的表，分批配置多个任务，或者配置整库的迁移任务。 - 如需进行增量迁移，Binlog日志： - 需开启，并且loose_polar_log_bin为on。否则预检查阶段提示报错，且无法成功启动数据迁移任务。 - 如为增量迁移任务，DTS要求源数据库的本地Binlog日志保存24小时以上，如为全量迁移和增量迁移任务，DTS要求源数据库的本地Binlog日志至少保留7天以上（您可在全量迁移完成后将Binlog保存时间设置为24小时以上），否则DTS可能因无法获取Binlog而导致任务失败，极端情况下甚至可能会导致数据不一致或丢失。由于您所设置的Binlog日志保存时间低于DTS要求的时间进而导致的问题，不在DTS的SLA保障范围内。 - 源库的操作限制： - 在库表结构迁移和全量迁移阶段，请勿执行库或表结构变更的DDL操作，否则数据迁移任务会失败。 - 如仅执行全量数据迁移，请勿向源实例中写入新的数据，否则会导致源和目标数据不一致。为实时保持数据一致性，建议选择结构迁移、全量数据迁移和增量数据迁移。
注意事项	- 不支持迁移源PolarDB MySQL引擎只读节点。 - 执行数据迁移前需评估源库和目标库的性能，同时建议业务低峰期执行数据迁移。否则全量数据迁移时DTS占用源和目标库一定读写资源，可能会导致数据库的负载上升。 - 由于全量数据迁移会并发执行INSERT操作，导致目标数据库的表产生碎片，因此全量迁移完成后目标数据库的表存储空间会比源实例的表存储空间大。 - 请确认DTS对数据类型为FLOAT或DOUBLE的列的迁移精度是否符合业务预期。DTS会通过 <code>ROUND(COLUMN,PRECISION)</code> 来读取这两类列的值。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。 - DTS会尝试恢复七天之内迁移失败任务。因此业务切换至目标实例前，请务必结束或释放该任务，或者将DTS访问目标实例账号的写权限用 <code>revoke</code> 命令回收掉。避免该任务被自动恢复后，源端数据覆盖目标实例的数据。 - DTS会在源库定时执 <code>CREATE DATABASE IF NOT EXISTS `test`</code> 命令以推进Binlog位点。
特殊情况	当目标库为RDS MySQL时，DTS会自动在RDS MySQL中创建数据库，如果待迁移的数据库名称不符合RDS MySQL的定义规范，您需要在配置迁移任务之前在RDS MySQL中创建数据库。相关操作，请参见创建数据库。

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按流量计费 详情请参

迁移类型	链路配置费用	公网流量费用
增量数据迁移	收费，详情请参见 产品定价 。	

迁移类型说明

- 库表结构迁移

DTS将源库中迁移对象的结构定义迁移到目标库。

说明

- 目前DTS支持结构迁移的对象为表、视图、触发器、存储过程和存储函数。
- 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。

- 全量迁移

DTS将源库中迁移对象的存量数据，全部迁移到目标库中。

- 增量迁移

DTS在全量迁移的基础上，将源库的增量更新数据迁移到目标库中。通过增量数据迁移可以实现在自建应用不停机的情况下，平滑地完成数据迁移。

支持增量迁移的SQL操作

操作类型	SQL操作语句
DML	INSERT、UPDATE、DELETE
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引擎集群	待迁移对象的读权限
RDS MySQL实例	迁移对象的读写权限

数据库账号创建及授权方法：

- PolarDB MySQL引擎集群请参见[创建数据库账号](#)。
- RDS MySQL实例请参见[创建账号](#)和[修改账号权限](#)。

操作步骤

1. 登录新版DTS迁移任务的列表页面。

 **说明** 您也可以登录DMS数据管理服务。在顶部菜单栏中，选择集成与开发（DTS）> 数据迁移。

2. 在页面左上角，选择迁移实例所属地域。



3. 单击创建任务，配置源库及目标库信息。

 **警告** 选择源和目标实例后，建议您仔细阅读页面上方显示的使用限制，以成功创建并执行迁移任务。

* 任务名称:

dtsspwpi35e

① 使用限制

1.源PolarDB MySQL集群需开启Binlog。2.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。3.不支持迁移源库的用户信息。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。
更多限制, [请查看详情](#)

源库信息

选择已有的实例:

可以选择一个已有的实例进行快速配置

* 数据库类型: ⑦

DB2 iSeries(AS/400)

DB2 LUW

HBase

MariaDb

MongoDB

MySQL

Oracle

PolarDB MySQL

PolarDB Oracle

PolarDB PostgreSQL

PolarDB-X 1.0

PolarDB-X 2.0

PostgreSQL

Redis

SQLServer

Teradata

* 接入方式:

阿里云实例

* 实例地区:

华东1（杭州）

是否跨阿里云账号: ⑦

不跨账号

跨账号

* PolarDB实例ID:

pc-bp1bjkuxc70v9l8c8

* 数据库账号: ⑦

dtstest

* 数据库密码:

将连接信息保存为实例

目标库信息

选择已有的实例:

可以选择一个已有的实例进行快速配置

* 数据库类型: ⑦

AnalyticDB MySQL 3.0

AnalyticDB PostgreSQL

DataHub

Kafka

MySQL

Oracle

PolarDB MySQL

PolarDB-X 2.0

* 接入方式:

阿里云实例

专线/VPN网关/智能网关

公网IP

ECS自建数据库

云企业网CEN

数据库网关DG

* 实例地区:

华东1（杭州）

* RDS实例ID:

rm-bp1imrtn6fq7hafz3

* 数据库账号: ⑦

dtstest

* 数据库密码:

* 连接方式:

☒ 非加密连接

☐ SSL安全连接

将连接信息保存为实例

取消

测试连接以进行下一步

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	数据库类型	选择PolarDB MySQL。
	接入方式	选择阿里云实例。
	实例地区	选择源PolarDB MySQL引擎集群所属地域。
	PolarDB MySQL实例ID	填入源PolarDB MySQL引擎集群ID。
	数据库账号	填入源PolarDB MySQL引擎集群的数据库账号，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	填入该数据库账号对应的密码。
目标库信息	数据库类型	选择MySQL。
	接入方式	选择阿里云实例。
	实例地区	选择目标RDS MySQL实例所属地域。
	RDS实例ID	选择目标RDS MySQL实例ID。
	数据库账号	填入目标RDS MySQL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	填入该数据库账号对应的密码。
	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 MySQL实例的SSL加密功能，详情请参见 设置SSL加密 。

4. 配置完成后，单击页面下方的测试连接以进行下一步。

警告

- 如果源或目标数据库是阿里云数据库实例（例如RDS MySQL、云数据库MongoDB版等）或ECS上的自建数据库，DTS会自动将对应地区DTS服务的IP地址添加到阿里云数据库实例的白名单或ECS的安全规则中，您无需手动添加，请参见[DTS服务器的IP地址段](#)；如果源或目标数据库是IDC自建数据库或其他云数据库，则需要您手动添加对应地区DTS服务的IP地址，以允许来自DTS服务器的访问。
- 上述场景中，DTS自动添加或您手动添加DTS服务的公网IP地址段可能会存在安全风险，一旦使用本产品代表您已理解和确认其中可能存在的安全风险，并且需要您做好基本的安全防护，包括但不限于加强账号密码强度防范、限制各网段开放的端口号、内部各API使用鉴权方式通信、定期检查并限制不需要的网段，或者使用通过内网（专线/VPN网关/智能网关）的方式接入。
- DTS任务完成或释放后，建议您手动检测并删除DTS相关的服务器IP地址段。

5. 配置任务对象及高级配置。

基础配置

基础配置

* 任务步骤:

☒ 库表结构迁移

☒ 全量迁移

☒ 增量迁移

① 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。

如果需要进行长期的数据实时同步，请使用数据同步功能。

注：增量迁移不支持trigger的同步[帮助文档](#)

* 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

* 同步对象:

自定义选择

源库对象

Q 支持正则，若全局搜索，请先展开树

> ☐

> ☐

> ☐

> ☐

> ☒ dtstestdata

> ☐

> ☐

已选择对象

批量编辑 ?

Q 支持正则，若全局搜索，请先展开树

☐ dtstestdata

【右键】选择对象
可以进行重命名或
DML、DDL配置

配置	说明
任务步骤	■ 如果只需要进行全量迁移，请同时选中库表结构迁移和全量迁移。 ■ 如果需要进行不停机迁移，请同时选中库表结构迁移、全量迁移和增量迁移。 ② 说明 如果未选择增量迁移，为保障数据一致性，数据迁移期间请勿在源实例中写入新的数据。

> 文档版本：20220713

84

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，数据迁移任务不会被启动。 ⓘ 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，请参见库表列名映射。 ■ 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，则会保留目标库中的该条记录，即源库中的该条记录不会迁移至目标库中。 ■ 表结构不一致的情况下，可能导致只能迁移部分列的数据或迁移失败。
同步对象	在同步对象框中单击待迁移的对象，然后单击  将其移动到已选择对象框。 ⓘ 说明
映射名称更改	■ 如需更改单个迁移对象在目标实例中的名称，请右击已选择对象中的迁移对象，设置方式，请参见库表列名单个映射。 ■ 如需批量更改迁移对象在目标实例中的名称，请单击已选择对象方框右上方的  设置方式，请参见库表列名批量映射。 ⓘ 说明 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。
过滤待迁移数据	支持设置条件过滤数据，详情请参见 通过SQL条件过滤任务数据 。
增量迁移的SQL操作	选择增量迁移DDL或DML操作，请右击已选择对象中的迁移对象，在弹跳框中选择所需增量迁移的DML和DDL操作。支持的操作，请参见 支持增量迁移的SQL操作 。

高级配置

高级配置

设置告警: ?

☒ 不设置

☐ 设置

源表DMS_ONLINE_DDL过程中是否复制临时表到目标库: ?

☐ 是

☒ 否

源库、目标库无法连接后的重试时间: ?

-120+

分钟

上一步配置源库及目标库信息

下一步保存任务并预检查

保存并返回列表

取消

配置	说明
设置告警	是否设置告警，当迁移失败或延迟超过阈值后，将通知告警联系人。 ■ 不设置：不设置告警。 ■ 设置：设置告警，您还需要设置告警阈值和告警联系人。
源表 DMS_ONLINE_DDL 过程中是否复制临时表到目标库	如源库使用数据管理DMS（Data Management Service）执行Online DDL变更，您可以选择是否迁移Online DDL变更产生的临时表数据。 ■ 是：迁移Online DDL变更产生的临时表数据。 ? 说明 Online DDL变更产生的临时表数据过大，可能会导致迁移任务延迟。 ■ 否：不迁移Online DDL变更产生的临时表数据，只迁移源库的原始DDL数据。 ? 说明 该方案会导致目标库锁表。
源、目标库无法连接重试时间	默认重试720分钟，您可以在取值范围（10~720分钟）内自定义重试时间，建议设置30分钟以上。如果DTS在设置的时间内重新连接上源、目标库，同步任务将自动恢复。否则，同步任务将失败。 ? 说明 ■ 针对同源或者同目标的多个DTS实例，网络重试时间以后创建任务的设置为准。 ■ 由于连接重试期间，DTS将收取任务运行费用，建议您根据业务需要自定义重试时间，或者在源和目标库实例释放后尽快释放DTS实例。

6. 上述配置完成后，单击页面右下角的下一步保存任务并预检查。

? 说明

- 在迁移任务正式启动之前，会先进行预检查。只有预检查通过后，才能成功启动迁移任务。
- 如果预检查失败，单击具体检查项后的, 查看失败详情。
 - 您可以根据提示修复后重新进行预检查。
 - 如无需修复告警检测项，您也可以选择**确认屏蔽**、**忽略告警项**并重新进行预检查，跳过告警检测项重新进行预检查。

7. 预检查通过率显示为100%时，单击**下一步购买**。

8. 在**购买**页面，选择数据迁移实例的链路规格，详细说明请参见下表。

类别	参数	说明
信息配置	链路规格	DTS为您提供了不同性能的迁移规格，迁移链路规格的不同会影响迁移速率，您可以根据业务场景进行选择，详情请参见 数据迁移链路规格说明 。

9. 配置完成后，阅读并选中《**数据传输（按量付费）服务条款**》。

10. 单击**购买并启动**，迁移任务正式开始，您可在数据迁移界面查看具体进度。

4. 数据同步

4.1. 数据同步方案概览

云原生关系型数据库PolarDB提供了多种数据同步方案，可满足不同的业务同步需求，使您可以在不影响业务的情况下平滑的实现引擎间的数据同步。

通过使用阿里云[数据传输服务（DTS）](#)，您可以实现PolarDB的实时同步。

数据同步

使用场景	文档链接
PolarDB间的数据同步	• PolarDB MySQL集群间的单向同步 • PolarDB MySQL集群间的双向同步
PolarDB与其它数据库的数据同步	• RDS MySQL同步至PolarDB MySQL集群 • PolarDB MySQL同步至RDS MySQL • 从PolarDB MySQL同步至云原生数据仓库AnalyticDB MySQL • 从PolarDB MySQL同步至云原生数据仓库AnalyticDB PostgreSQL • 从PolarDB MySQL同步至DataHub • 从PolarDB MySQL同步到Kafka • 从ECS上的自建MySQL同步至PolarDB MySQL • 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

4.2. PolarDB间的数据同步

4.2.1. 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](#)。支持的数据库版本，请参见[同步方案概览](#)。
- 源PolarDB MySQL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。
- 建议源和目标PolarDB MySQL集群的数据库版本保持一致，或者从低版本同步到高版本以保障兼容性。

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[数据同步拓扑介绍](#)。

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[源库存在触发器时如何配置同步作业](#)。
- 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[购买流程](#)。

 **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

* 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	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。
	数据库密码	填入数据库账号对应的密码。

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

如果源或目标数据库是阿里云数据库实例（例如RDS MySQL、云数据库MongoDB版等）或ECS上的自建数据库，DTS会自动将对应地区DTS服务的IP地址添加到阿里云数据库实例的白名单或ECS的安全规则中，您无需手动添加，请参见DT S服务器的IP地址段。

DTS任务完成或释放后，建议您手动删除添加的DTS服务器IP地址段。

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 (DML+DDL)

源库对象

若全局搜索，请先展开树

dtstestdata

Tables

Views

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

*源表DMS_ONLINE_DDL过程中是否复制临时表到目标库:

是

否

*源库、目标库无法连接后的重试时间

720

分钟

取消

上一步

下一步

配置项目	配置说明
------	------

91

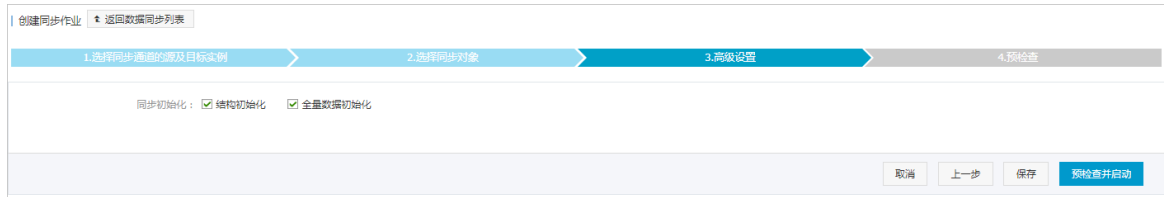
> 文档版本：20220713

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，您可以设置同步对象在目标实例中的名称来避免表名冲突。 ◦ 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。
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，详情请参见设置同步对象在目标实例中的名称。
映射名称更改	如需更改同步对象在目标实例中的名称，请使用对象名映射功能，详情请参见库表列映射。

配置项目	配置说明
源表DMS_ONLINE_DDL过程中是否复制临时表到目标库	如源库使用数据管理DMS（Data Management Service）执行Online DDL变更，您可以选择是否同步Online DDL变更产生的临时表数据。 - 是：同步Online DDL变更产生的临时表数据。  说明 Online DDL变更产生的临时表数据过大，可能会导致同步任务延迟。 - 否：不同步Online DDL变更产生的临时表数据，只同步源库的原始DDL数据。  说明 该方案会导致目标库锁表。
源、目标库无法连接重试时间	当源、目标库无法连接时，DTS默认重试720分钟（即12小时），您也可以自定义重试时间。如果DTS在设置的时间内重新连接上源、目标库，同步任务将自动恢复。否则，同步任务将失败。  说明 由于连接重试期间，DTS将收取任务运行费用，建议您根据业务需要自定义重试时间，或者在源和目标库实例释放后尽快释放DTS实例。

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，才能成功启动同步作业。
- 如果预检查失败，单击具体检查项后的，查看失败详情。

- 您可以根据提示修复后重新进行预检查。
- 如无需修复告警检测项，您也可以选择确认屏蔽、忽略告警项并重新进行预检查，跳过告警检测项重新进行预检查。

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	同步策略(全部)	操作
hangzhou-hangzhou-small	同步中	延时: 1376 毫秒 速度: 0.00RPS/(0.000MB/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 >				

4.2.2. 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。
- 如双向同步任务的源实例或目标实例位于海外地域，则仅支持同地域的双向同步，不支持跨地域的双向同步。例如，支持日本地域间的双向同步，不支持日本地域与法兰克福地域间的双向同步。

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。
- 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[源库存在触发器时如何配置同步作业](#)。
- 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，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

操作类型	SQL操作语句
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[购买流程](#)。

 **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。

i. 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

ii. 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
目标实例信息	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。
	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。
	数据库密码	填入数据库账号的密码。

iii. 单击页面右下角的授权白名单并进入下一步。

 说明

- 如果源或目标数据库是阿里云数据库实例（例如RDS MySQL、云数据库MongoDB版等）或ECS上的自建数据库，DTS会自动将对应地区DTS服务的IP地址添加到阿里云数据库实例的白名单或ECS的安全规则中，您无需手动添加，请参见[DTS服务器的IP地址段](#)。
- DTS任务完成或释放后，建议您手动删除添加的DTS服务器IP地址段。

iv. 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.选择同步对象

3.高级设置

4.预检查

同步策略: 双向同步 (DML+DDL)

是否过滤DDL: ☐ 是 ☒ 否

DML同步类型: ☒ Insert ☒ Delete ☒ Update

冲突修复策略: TaskFailed(遇到冲突, 任务报错退出)

源库对象

若全局搜索, 请先展开树

dtstestdata

Tables

Views

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

*源表DMS_ONLINE_DDL过程中是否复制临时表到目标库: ☐ 是 ☒ 否

*源库、目标库无法连接后的重试时间: 720 分钟

取消

上一步

下一步

类别	配置	说明
	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 DDL语法同步方向限制。为保障双向同步链路的稳定性，只支持正向同步DDL，不支持反向同步DDL。
	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
同步策略配置	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，您可以设置同步对象在目标实例中的名称来避免表名冲突。 ■ 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，详情请参见设置同步对象在目标实例中的名称。
映射名称更改	无	如需更改同步对象在目标实例中的名称，请使用对象名映射功能，详情请参见 库表列映射 。
源表 DMS_ONLINE_ DDL过程中是否 复制临时表到 目标库	无	如源库使用数据管理DMS（Data Management Service）执行 Online DDL变更，您可以选择是否同步 Online DDL变更产生的临时表数据。 - 是：同步 Online DDL变更产生的临时表数据。  说明 Online DDL变更产生的临时表数据过大，可能会导致同步任务延迟。 - 否：不同步 Online DDL变更产生的临时表数据，只同步源库的原始 DDL 数据。  说明 该方案会导致目标库锁表。
源、目标库无法连接重试时间	无	当源、目标库无法连接时，DTS默认重试720分钟（即12小时），您也可以自定义重试时间。如果DTS在设置的时间内重新连接上源、目标库，同步任务将自动恢复。否则，同步任务将失败。  说明 由于连接重试期间，DTS将收取任务运行费用，建议您根据业务需要自定义重试时间，或者在源和目标库实例释放后尽快释放DTS实例。

v. 上述配置完成后，单击页面右下角的下一步。

vi. 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。

vii. 上述配置完成后，单击页面右下角的预检查并启动。

?

说明

- 在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，才能成功启动同步作业。
- 如果预检查失败，单击具体检查项后的, 查看失败详情。
 - 您可以根据提示修复后重新进行预检查。
 - 如无需修复告警检测项，您也可以选择确认屏蔽、忽略告警项并重新进行预检查，跳过告警检测项重新进行预检查。

viii. 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。

i. 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 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	
☐ 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	

ii. 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。

执行结果

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

pc-bp

t5b

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

pc-bp

3hf

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

>

»

4.3. PolarDB与其它数据库的数据同步

4.3.1. RDS MySQL同步至PolarDB MySQL集群

通过数据传输服务DTS（Data Transmission Service），可以实现MySQL同步至PolarDB MySQL引擎集群。

支持的源数据库

MySQL同步至PolarDB MySQL引擎集群，支持源数据库MySQL为以下类型。本文以RDS MySQL实例为例介绍配置流程，其他类型的源数据库配置流程与本案例类似。

- RDS MySQL实例。
- ECS上的自建数据库。
- 通过专线、VPN网关或智能网关接入的自建数据库。
- 通过数据库网关接入的自建数据库。
- 通过云企业网CEN接入的自建数据库。

前提条件

- 已创建源RDS MySQL实例，创建方式，请参见[创建RDS MySQL实例](#)。
- 已创建目标PolarDB MySQL引擎集群，详情请参见[购买按量付费集群](#)和[购买包年包月集群](#)。
- PolarDB MySQL引擎集群的存储空间须大于RDS MySQL实例占用的存储空间。

注意事项

类型	说明
----	----

类型	说明
源库限制	- 待同步的表需具备主键或唯一约束，且字段具有唯一性，否则可能会导致目标数据库中出现重复数据。 - 如同步对象为表级别，且需进行编辑（如表列名映射），则单次同步任务仅支持同步至多1000张表。当超出数量限制，任务提交后会显示请求报错，此时建议您拆分待同步的表，分批配置多个任务，或者配置整库的同步任务。 - Binlog日志： - 需开启Binlog，开启方法请参见设置实例参数，并且binlog_row_image为full。否则预检查阶段提示报错，且无法成功启动数据同步任务。  注意 - 如源实例为自建MySQL，则需开启Binlog，并且binlog_format为row、binlog_row_image为full。 - 如源实例自建MySQL是双主集群（两者互为主从），为保障DTS能获取全部的Binlog日志，则您需开启参数log_slave_updates。具体操作请参见自建MySQL创建账号并设置binlog。 - 如为增量同步任务，DTS要求源数据库的本地Binlog日志保存24小时以上，如为全量同步和增量同步任务，DTS要求源数据库的本地Binlog日志至少保留7天以上（您可在全量同步完成后将Binlog保存时间设置为24小时以上），否则DTS可能因无法获取Binlog而导致任务失败，极端情况下甚至可能会导致数据不一致或丢失。由于您所设置的Binlog日志保存时间低于DTS要求的时间进而导致的问题，不在DTS的SLA保障范围内。如源为RDS MySQL，具体操作请参见本地日志（Binlog）。
其他限制	- 建议源和目标库版本保持一致，或者从低版本同步到高版本以保障兼容性。如为高版本同步至低版本，可能存在数据库兼容性问题。 - 执行数据同步前需评估源库和目标库的性能，同时建议业务低峰期执行数据同步。否则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。 - 全量初始化会并发执行INSERT操作，导致目标数据库的表产生碎片，因此全量初始化完成后目标实例的表空间比源实例的表空间大。 - 如同步对象为单个或多个表（非整库），那么在数据同步时，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。您可以使用数据管理DMS（Data Management Service）来执行在线DDL变更，请参见不锁表结构变更。 - 在DTS同步期间，不允许有除DTS外的数据写入目标库，否则会导致源库与目标库数据不一致。例如，有除DTS外的数据写入目标库时，使用DMS执行在线DDL变更，可能引起目标库数据丢失。

类型	说明
特殊情况	当源库为自建MySQL时 - 在同步时，如果源库进行主备切换，将会导致同步任务失败。 - 由于DTS的延迟时间是根据同步到目标库最后一条数据的时间戳和当前时间戳对比得出，源库长时间未执行DML操作可能导致延迟信息不准确。如果任务显示的延迟时间过大，您可以在源库执行一个DML操作来更新延迟信息。  说明 如果同步对象选择为整库，您还可以创建心跳表，心跳表每秒定期更新或者写入数据。 - DTS会在源库定时执行CREATE DATABASE IF NOT EXISTS `test`命令以推进Binlog位点。

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[数据同步拓扑介绍](#)。

支持同步的SQL

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

操作步骤

1. 登录[新版DTS同步任务的列表页面](#)。

 **说明** 您也可以登录[DMS数据管理服务](#)。在顶部菜单栏中，选择集成与开发（DTS），在左侧导航栏选择数据传输（DTS）> 数据同步。

2. 在页面左上角，选择同步实例所属地域。
3. 单击**创建任务**，配置源库及目标库信息。

 **警告** 选择源和目标实例后，建议您仔细阅读页面上方显示的使用限制，以成功创建并执行同步任务。

* 任务名称:

rdsmysql_polarbmysql

4 使用限制

1.待同步的表需具备主键或唯一约束，且字段具有唯一性，否则可能会导致目标数据库中出現重复数据。
2.Binlog日志需开启且至少保留24小时（建议3天以上）。

更多限制,查看详情

源库信息

* 数据库类型: ?

AS400DB2SQLServerMySQLPolarDB MySQLPolarDB-X 2.0PostgreSQL

* 接入方式:

阿里云实例ECS上的自建数据库

* 实例地区:

华东1（杭州）

是否跨阿里云账号: ?

不跨账号跨账号

* RDS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

目标库信息

* 数据库类型: ?

AnalyticDBPostgreSQLMySQLPolarDB MySQLPolarDB-X 2.0

* 接入方式:

阿里云实例

* 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:

取消

测试连接以进行下一步

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。
	数据库类型	选择MySQL。
	接入方式	选择阿里云实例。
	实例地区	选择源RDS MySQL实例所属地域。
	是否跨阿里云账号	本场景为同一阿里云账号间迁移，选择不跨账号。

类别	配置	说明
源库信息	RDS实例ID	选择源RDS MySQL实例ID。 说明 源和目标RDS MySQL实例可以不同或相同，即您可以使用DTS实现两个RDS MySQL实例间的数据迁移或同一RDS MySQL实例内的数据迁移。
	数据库账号	填入源RDS MySQL实例的数据库账号，需具备REPLICATION CLIENT、REPLICATION SLAVE、SHOW VIEW和SELECT权限。
	数据库密码	填入该数据库账号对应的密码。
	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 MySQL实例的SSL加密功能，详情请参见 设置SSL加密 。
目标库信息	数据库类型	选择PolarDB MySQL。
	接入方式	选择阿里云实例。
	实例地区	选择目标PolarDB MySQL引擎集群所属地域。
	PolarDB MySQL实例ID	选择目标PolarDB MySQL引擎集群ID。
	数据库账号	填入目标PolarDB MySQL引擎集群的数据库账号，需具备读写权限。
	数据库密码	填入该数据库账号对应的密码。

4. 配置完成后，单击页面下方的测试连接以进行下一步。

说明

- 如果源或目标数据库是阿里云数据库实例（例如RDS MySQL、云数据库MongoDB版等）或ECS上的自建数据库，DTS会自动将对应地区DTS服务的IP地址添加到阿里云数据库实例的白名单或ECS的安全规则中，您无需手动添加，请参见[DTS服务器的IP地址段](#)。
- DTS任务完成或释放后，建议您手动删除添加的DTS服务器IP地址段。

5. 配置任务对象及高级配置。

基础配置

基础配置

* 任务步骤:

☒ 库表结构同步

☒ 全量同步

☒ 增量同步

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

注：不支持trigger的同步[帮助文档](#)

* 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

* 同步对象:

自定义选择

源库对象

支持正则，若全局搜索，请先展开树

> ☐

> ☐

> ☐

> ☒ dtstestdata

> ☐

> ☐

已选择对象

批量编辑

?

支持正则，若全局搜索，请先展开树

☐ dtstestdata

【右键】选择对象
可以进行重命名或
DML、DDL配置

配置	说明
任务步骤	固定选中 增量同步 。默认情况下，您还需要同时选中 库表结构同步 和 全量同步 。预检查完成后，DTS会将源实例中待同步对象的全量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。
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，数据同步任务不会被启动。 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，请参见库表列名映射。 ■ 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，如在目标库遇到与源库主键的值相同的记录： ■ 全量期间，DTS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。 ■ 增量期间，DTS不会保留目标集群中的该条记录，即源库中的该条记录会覆盖至目标数据库中。 ■ 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。

107

> 文档版本：20220713

配置	说明
同步对象	在源库对象框中单击待同步对象，然后单击 将其移动至已选择对象框。  说明 同步对象选择的粒度为库、表、列。若选择的同步对象为表或列，其他对象（如视图、触发器、存储过程）不会被同步至目标库。
映射名称更改	- 如需更改单个同步对象在目标实例中的名称，请右击已选择对象中的同步对象，设置方式，请参见库表列名单个映射。 - 如需批量更改同步对象在目标实例中的名称，请单击已选择对象方框右上方的批量编辑，设置方式，请参见库表列名批量映射。
过滤待同步数据	支持设置WHERE条件过滤数据，请参见 通过SQL条件过滤任务数据 。
同步的SQL操作	请右击已选择对象中的同步对象，在弹跳框中选择所需同步的DML和DDL操作，支持的操作，请参见 支持同步的SQL 。

高级配置

高级配置

设置告警: 

☒ 不设置 ☐ 设置

目标库对象名称大小写策略: 

DTS默认策略

源表DMS_ONLINE_DDL过程中是否复制临时表到目标库: 

☐ 是 ☒ 否

源库、目标库无法连接后的重试时间: 

-120+

分钟

上一步配置源库及目标库信息

下一步保存任务并预检查

保存并返回列表

取消

配置	说明
设置告警	是否设置告警，当同步失败或延迟超过阈值后，将通知告警联系人。 - 不设置：不设置告警。 - 设置：设置告警，您还需要设置告警阈值和告警联系人。

配置	说明
目标库对象名称大小写策略	您可以配置目标实例中迁移对象的库名、表名和列名的英文大小写策略。默认情况下选择 DTS默认策略 ，您也可以选择与源库、目标库默认策略保持一致。更多信息，请参见 目标库对象名称大小写策略 。
源表 DMS_ONLINE_DDL过程中是否复制临时表到目标库	如源库使用数据管理DMS（Data Management Service）执行Online DDL变更，您可以选择是否同步Online DDL变更产生的临时表数据。 ■ 是：同步Online DDL变更产生的临时表数据。  说明 Online DDL变更产生的临时表数据过大，可能会导致同步任务延迟。 ■ 否：不同步Online DDL变更产生的临时表数据，只同步源库的原始DDL数据。  说明 该方案会导致目标库锁表。
源库、目标库无法连接后的重试时间	在同步任务连接失败时，DTS会立即进行持续的重试连接，默认持续重试时间为120分钟，您也可以在取值范围（10~1440分钟）内自定义重试时间，建议设置30分钟以上。如果DTS在设置的重试时间内重新连接上源库、目标库，同步任务将自动恢复。否则，同步任务将失败。  说明 ■ 针对同源或者同目标的多个DTS实例，如DTS实例A和DTS实例B，设置网络重试时间时A设置30分钟，B设置60分钟，则重试时间以低的30分钟为准。 ■ 由于连接重试期间，DTS将收取任务运行费用，建议您根据业务需要自定义重试时间，或者在源和目标库实例释放后尽快释放DTS实例。

6. 上述配置完成后，单击页面下方的**下一步保存任务并预检查**。

 说明

- 在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，才能成功启动同步作业。
- 如果预检查失败，单击具体检查项后的，查看失败详情。
 - 您可以根据提示修复后重新进行预检查。
 - 如无需修复告警检测项，您也可以选择**确认屏蔽、忽略告警项**并重新进行预检查，跳过告警检测项重新进行预检查。

7. 预检查通过率显示为**100%**时，单击**下一步购买**。

8. 在**购买**页面，选择数据同步实例的计费方式、链路规格，详细说明请参见下表。

类别	参数	说明
----	----	----

类别	参数	说明
信息配置	计费方式	- 预付费（包年包月）：在新建实例时支付费用。适合长期需求，价格比按量付费更实惠，且购买时长越长，折扣越多。 - 后付费（按量付费）：按小时扣费。适合短期需求，用完可立即释放实例，节省费用。
	链路规格	DTS为您提供不同性能的同步规格，同步链路规格的不同会影响同步速率，您可以根据业务场景进行选择，详情请参见 数据同步链路规格说明 。
	订购时长	在预付费模式下，选择包年包月实例的时长和数量，包月可选择1~9个月，包年可选择1~3年。  说明 该选项仅在付费类型为预付费时出现。

9. 配置完成后，阅读并勾选《[数据传输（按量付费）服务条款](#)》。

10. 单击**购买并启动**，同步任务正式开始，您可在数据同步界面查看具体任务进度。

4.3.2. PolarDB MySQL同步至RDS MySQL

通过数据传输服务DTS（Data Transmission Service），可以实现PolarDB MySQL引擎集群同步至MySQL（包括自建MySQL、RDS MySQL实例）。

支持的目标数据库

PolarDB MySQL引擎集群同步至MySQL，支持目标数据库为以下类型。本文以RDS MySQL实例为例介绍配置流程，其他类型的目标数据库的配置流程与本案例类似。

- RDS MySQL实例。
- ECS上的自建数据库。
- 通过专线、VPN网关或智能网关接入的自建数据库。
- 通过数据库网关接入的自建数据库。
- 通过云企业网CEN接入的自建数据库。

前提条件

- 已创建源PolarDB MySQL引擎集群，详情请参见[购买按量付费集群](#)和[购买包年包月集群](#)。
- 已创建目标RDS MySQL实例，创建方式，请参见[创建RDS MySQL实例](#)。
- 目标RDS MySQL实例的存储空间须大于源PolarDB MySQL引擎集群占用的存储空间。

注意事项

类型	说明
----	----

类型	说明
源库限制	- 待同步的表需具备主键或唯一约束，且字段具有唯一性，否则可能会导致目标数据库中出现重复数据。 - 如同步对象为表级别，且需进行编辑（如表列名映射），则单次同步任务仅支持同步至多1000张表。当超出数量限制，任务提交后会显示请求报错，此时建议您拆分待同步的表，分批配置多个任务，或者配置整库的同步任务。 - Binlog日志： - 需开启，并且loose_polar_log_bin为on。否则预检查阶段显示报错提示，且无法成功启动数据同步任务。 - 如为增量同步任务，DTS要求源数据库的本地Binlog日志保存24小时以上，如为全量同步和增量同步任务，DTS要求源数据库的本地Binlog日志至少保留7天以上（您可在全量同步完成后将Binlog保存时间设置为24小时以上），否则DTS可能因无法获取Binlog而导致任务失败，极端情况下甚至可能会导致数据不一致或丢失。由于您所设置的Binlog日志保存时间低于DTS要求的时间进而导致的问题，不在DTS的SLA保障范围内。
其他限制	- 不支持同步源PolarDB MySQL引擎只读节点。 - 执行数据同步前需评估源库和目标库的性能，同时建议业务低峰期执行数据迁移。否则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。 - 全量初始化会并发执行INSERT操作，导致目标数据库的表产生碎片，因此全量初始化完成后目标实例的表空间比源实例的表空间大。 - 对于表级别的数据同步，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。 - 对于表级别的数据同步，如果除DTS以外的数据没有写入目标库，您可以使用数据管理DMS（Data Management Service）来执行在线DDL变更，请参见不锁表结构变更。 - 在DTS同步期间，不允许有除DTS外的数据写入目标库，否则会导致源库与目标库数据不一致。例如，有除DTS外的数据写入目标库时，使用DMS执行在线DDL变更，可能引起目标库数据丢失。
其他注意事项	DTS会在源库定时执CREATE DATABASE IF NOT EXISTS `test`命令以推进Binlog位点。

优惠活动

DTS优惠活动，最低0折

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
- 一对一双向同步

关于各类同步架构的介绍及注意事项，请参见[数据同步拓扑介绍](#)。

支持同步的SQL操作

操作类型	SQL操作语句
DML	INSERT、UPDATE、DELETE

操作类型	SQL操作语句
DDL	• ALTER TABLE、ALTER VIEW • CREATE FUNCTION、CREATE INDEX、CREATE PROCEDURE、CREATE TABLE、CREATE VIEW • DROP INDEX、DROP TABLE • RENAME TABLE • TRUNCATE TABLE

操作步骤

1. 登录新版DTS同步任务的列表页面。

 **说明** 您也可以登录DMS数据管理服务。在顶部菜单栏中，选择集成与开发（DTS），在左侧导航栏选择数据传输（DTS）> 数据同步。

2. 在页面左上角，选择同步实例所属地域。
3. 单击创建任务，配置源库及目标库信息。

 **警告** 选择源和目标实例后，建议您仔细阅读页面上方显示的使用限制，以成功创建并执行同步任务。

* 任务名称:

polardbm_rdsmysql

❗ 使用限制

1. 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。

2. RENAME TABLE操作可能导致同步数据不一致，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。

3. 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。

4. 如果数据同步的源库设有主键或唯一约束，且记录的全字段设有唯一性，可能会出现重复数据。

更多限制,请查看详情

源库信息

* 数据库类型: ②

DB2 ISeries(AS/400)

DB2 LUW

SQLServer

MySQL

PolarDB MySQL

PolarDB-X 2.0

PostgreSQL

* 接入方式:

阿里云实例

* 实例地区:

华东1 (杭州)

* PolarDB实例ID:

pc-xxxxxx

* 数据库账号: ②

dtstest

* 数据库密码:

将连接信息保存为模版

目标库信息

* 数据库类型: ②

MySQL

PolarDB MySQL

PolarDB-X 2.0

* 接入方式:

阿里云实例

* 实例地区:

华东1 (杭州)

* RDS实例ID:

rm-xxxxxx

* 数据库账号: ②

dtstest

* 数据库密码:

* 连接方式:

非加密连接

SSL安全连接

将连接信息保存为模版

取消

测试连接以进行下一步

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	数据库类型	选择PolarDB MySQL。
	接入方式	选择阿里云实例。
	实例地区	选择源PolarDB MySQL引擎集群所属地域。
	PolarDB MySQL实例ID	选择源PolarDB MySQL引擎集群ID。
	数据库账号	填入源PolarDB MySQL引擎集群的数据库账号，需具备待同步对象的读权限。
	数据库密码	填入该数据库账号对应的密码。

类别	配置	说明
目标库信息	数据库类型	选择MySQL。
	接入方式	选择阿里云实例。
	实例地区	选择目标RDS MySQL实例所属地域。
	RDS实例ID	选择目标RDS MySQL实例ID。
	数据库账号	填入目标RDS MySQL实例的数据库账号，需具备读写权限。
	数据库密码	填入该数据库账号对应的密码。
	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 MySQL实例的SSL加密功能，详情请参见 设置SSL加密 。

4. 配置完成后，单击页面下方的[测试连接](#)以进行下一步。

 说明

- 如果源或目标数据库是阿里云数据库实例（例如RDS MySQL、云数据库MongoDB版等）或ECS上的自建数据库，DTS会自动将对应地区DTS服务的IP地址添加到阿里云数据库实例的白名单或ECS的安全规则中，您无需手动添加，请参见[DTS服务器的IP地址段](#)。
- DTS任务完成或释放后，建议您手动删除添加的DTS服务器IP地址段。

5. 配置任务对象及高级配置。

◦ 基础配置

基础配置

* 任务步骤:

☒ 库表结构同步

☒ 全量同步

☒ 增量同步

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

注：不支持trigger的同步[帮助文档](#)

* 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

* 同步对象:

自定义选择

源库对象

Q 支持正则，若全局搜索，请先展开树

> ☐

> ☐

> ☐

> ☐

> ☒ dtstestdata

> ☐

> ☐

已选择对象

批量编辑 ?

Q 支持正则，若全局搜索，请先展开树

☐ dtstestdata

【右键】选择对象
可以进行重命名或
DML、DDL配置

配置	说明
任务步骤	固定选中 增量同步 。默认情况下，您还需要同时选中 库表结构同步 和 全量同步 。预检查完成后，DTS会将源实例中待同步对象的全量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。
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，数据同步任务不会被启动。 ? 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，请参见库表列名映射。 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，如在目标库遇到与源库主键的值相同的记录： 全量期间，DTS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。 增量期间，DTS不会保留目标集群中的该条记录，即源库中的该条记录会覆盖至目标数据库中。 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。

115

> 文档版本：20220713

配置	说明
同步对象	在源库对象框中单击待同步对象，然后单击 将其移动至已选择对象框。  说明 同步对象选择的粒度为库、表、列。若选择的同步对象为表或列，其他对象（如视图、触发器、存储过程）不会被同步至目标库。
映射名称更改	■ 如需更改单个同步对象在目标实例中的名称，请右击已选择对象中的同步对象，设置方式，请参见库表列名单个映射。 ■ 如需批量更改同步对象在目标实例中的名称，请单击已选择对象方框右上方的批量编辑，设置方式，请参见库表列名批量映射。
过滤待同步数据	支持设置WHERE条件过滤数据，请参见 通过SQL条件过滤任务数据 。
同步的SQL操作	请右击已选择对象中的同步对象，在弹跳框中选择所需同步的DML和DDL操作，支持的操作，请参见 支持同步的SQL操作 。

高级配置

高级配置

设置告警: 

☒ 不设置 ☐ 设置

源表DMS_ONLINE_DDL过程中是否复制临时表到目标库: 

☐ 是 ☒ 否

源库、目标库无法连接后的重试时间: 

-120+

分钟

上一步配置源库及目标库信息

下一步保存任务并预检查

保存并返回列表

取消

配置	说明
设置告警	是否设置告警，当同步失败或延迟超过阈值后，将通知告警联系人。 ■ 不设置：不设置告警。 ■ 设置：设置告警，您还需要设置告警阈值和告警联系人。

配置	说明
源表 DMS_ONLINE_ DDL过程中是否 复制临时表到 目标库	如源库使用数据管理DMS（Data Management Service）执行Online DDL变更，您可以选择是否同步Online DDL变更产生的临时表数据。 ■ 是：同步Online DDL变更产生的临时表数据。  说明 Online DDL变更产生的临时表数据过大，可能会导致同步任务延迟。 ■ 否：不同步Online DDL变更产生的临时表数据，只同步源库的原始DDL数据。  说明 该方案会导致目标库锁表。
源库、 目标库 无法连 接后的 重试时 间	在同步任务连接失败时，DTS会立即进行持续的重试连接，默认持续重试时间为120分钟，您也可以在此取值范围（10~1440分钟）内自定义重试时间，建议设置30分钟以上。如果DTS在设置的重试时间内重新连接上源库、目标库，同步任务将自动恢复。否则，同步任务将失败。  说明 ■ 针对同源或者同目标的多个DTS实例，如DTS实例A和DTS实例B，设置网络重试时间时A设置30分钟，B设置60分钟，则重试时间以低的30分钟为准。 ■ 由于连接重试期间，DTS将收取任务运行费用，建议您根据业务需要自定义重试时间，或者在源和目标库实例释放后尽快释放DTS实例。

6. 上述配置完成后，单击页面下方的下一步保存任务并预检查。

 说明

- 在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，才能成功启动同步作业。
- 如果预检查失败，单击具体检查项后的，查看失败详情。
- 您可以根据提示修复后重新进行预检查。
- 如无需修复告警检测项，您也可以选择确认屏蔽、忽略告警项并重新进行预检查，跳过告警检测项重新进行预检查。

7. 预检查通过率显示为100%时，单击下一步购买。

8. 在购买页面，选择数据同步实例的计费方式、链路规格，详细说明请参见下表。

类别	参数	说明
	计费方式	◦ 预付费（包年包月）：在新建实例时支付费用。适合长期需求，价格比按量付费更实惠，且购买时长越长，折扣越多。 ◦ 后付费（按量付费）：按小时扣费。适合短期需求，用完可立即释放实例，节省费用。

类别 信息配置	参数	说明
	链路规格	DTS为您提供不同性能的同步规格，同步链路规格的不同会影响同步速率，您可以根据业务场景进行选择，详情请参见 数据同步链路规格说明 。
	订购时长	在预付费模式下，选择包年包月实例的时长和数量，包月可选择1~9个月，包年可选择1~3年。  说明 该选项仅在付费类型为预付费时出现。

9. 配置完成后，阅读并勾选《数据传输（按量付费）服务条款》。

10. 单击**购买并启动**，同步任务正式开始，您可在数据同步界面查看具体任务进度。

4.3.3. 从PolarDB MySQL同步至云原生数据仓库

AnalyticDB MySQL

云原生数据仓库AnalyticDB 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同步到云原生数据仓库AnalyticDB 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

- 已创建目标云原生数据仓库AnalyticDB MySQL集群，详情请参见[创建云原生数据仓库AnalyticDB MySQL（2.0）](#)或[创建云原生数据仓库AnalyticDB MySQL（3.0）](#)。
- 确保目标云原生数据仓库AnalyticDB 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](#)。
- 如果同步的目标为云原生数据仓库AnalyticDB MySQL（2.0），源PolarDB MySQL中待同步的对象不能包含云原生数据仓库AnalyticDB 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

- 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）。
- 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** 如果同步的目标为云原生数据仓库AnalyticDB MySQL（3.0），您可以使用DMS提供的相关功能来执行在线DDL变更，详情请参见[DDL无锁变更](#)。

- 由于云原生数据仓库AnalyticDB MySQL（3.0）本身的使用限制，当云原生数据仓库AnalyticDB 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同步失败。

术语/概念对应关系

PolarDB MySQL	云原生数据仓库AnalyticDB MySQL
数据库	- 云原生数据仓库AnalyticDB MySQL（2.0）：表组 - 云原生数据仓库AnalyticDB MySQL（3.0）：数据库
表	- 云原生数据仓库AnalyticDB MySQL（2.0）：表 - 云原生数据仓库AnalyticDB MySQL（3.0）：表

 **说明** 关于云原生数据仓库AnalyticDB MySQL中表组和表的相关介绍，请参见[基本概念](#)。

支持同步的SQL操作

目标数据库版本	支持的SQL操作
云原生数据仓库AnalyticDB MySQL2.0	- DDL操作：ADD COLUMN、DROP COLUMN、MODIFY COLUMN - DML操作：INSERT、UPDATE、DELETE
云原生数据仓库AnalyticDB MySQL3.0	- DDL操作：CREATE TABLE、DROP TABLE、RENAME TABLE、TRUNCATE TABLE、ADD COLUMN、DROP COLUMN、MODIFY 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，详情请参见修复因变更字段类型导致的同步失败。

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。
云原生数据仓库AnalyticDB 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。
云原生数据仓库AnalyticDB 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[创建云原生数据仓库AnalyticDB MySQL数据库账号](#)。

数据类型映射关系

详情请参见[结构初始化涉及的数据类型映射关系](#)。

操作步骤

1. [购买数据同步实例](#)。

 **说明** 购买时，选择源实例为PolarDB、目标实例为AnalyticDB 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.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。
	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。

类别	配置	说明
目标实例信息	版本	根据目标云原生数据仓库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	选择目标云原生数据仓库AnalyticDB MySQL的集群ID。
	数据库账号	填入云原生数据仓库AnalyticDB 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

- 如果源或目标数据库是阿里云数据库实例（例如RDS MySQL、云数据库MongoDB版等）或ECS上的自建数据库，DTS会自动将对应地区DTS服务的IP地址添加到阿里云数据库实例的白名单或ECS的安全规则中，您无需手动添加，请参见[DTS服务器的IP地址段](#)。
- DTS任务完成或释放后，建议您手动删除添加的DTS服务器IP地址段。

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

注：DTS全量任务运行期间，不要清理DTS任务启动后源库产生的增量数据日志。源库如果过早清理日志，可能会导致DTS增量任务失败

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

多表旧并：☐ 是 ☒ 否

同步操作类型：☒ Insert ☒ Update ☒ Delete ☒ Alter Table ☒ Truncate Table
☒ Create Table ☒ Drop Table

源库对象

若全局搜索，请先展开树

dtstestdata

Tables

Views

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

*源表OMS_ONLINE_DDL过程中是否复制临时表到目标库：☒ 是 ☐ 否

*源库、目标库无法连接后的重试时间 720 分钟

取消

上一步

下一步

预检查并启动

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，详情请参见设置同步对象在目标实例中的名称。 ◦ 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 <code>__dts_data_source</code> 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，支持的同步操作详情请参见 支持同步的SQL操作 ，默认情况下都处于选中状态。

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，详情请参见设置同步对象在目标实例中的名称。
映射名称更改	如需更改同步对象在目标实例中的名称，请使用对象名映射功能，详情请参见 库表列映射 。
源表 DMS_ONLINE_DDL过程中是否复制临时表到目标库	如源库使用数据管理DMS（Data Management Service）执行Online DDL变更，您可以选择是否同步Online DDL变更产生的临时表数据。 - 是：同步Online DDL变更产生的临时表数据。 说明 Online DDL变更产生的临时表数据过大，可能会导致同步任务延迟。 - 否：不同步Online DDL变更产生的临时表数据，只同步源库的原始DDL数据。 说明 该方案会导致目标库锁表。
源、目标库无法连接重试时间	当源、目标库无法连接时，DTS默认重试720分钟（即12小时），您也可以自定义重试时间。如果DTS在设置的时间内重新连接上源、目标库，同步任务将自动恢复。否则，同步任务将失败。 说明 由于连接重试期间，DTS将收取任务运行费用，建议您根据业务需要自定义重试时间，或者在源和目标库实例释放后尽快释放DTS实例。

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。

创建同步作业

返回数据同步列表

1. 选择同步源和目标实例

2. ADS账号授权

3. 选择同步对象

4. 预检查

按任意键到库表名会显示源库表名

ADP实例ID	ADP实例名	类型(全部)	主键列	分区列	分区数	定义状态(全部)
说明 选择了结构初始化后，您需要定义待同步的表在云原生数据库AnalyticDB 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，才能成功启动同步作业。
- 如果预检查失败，单击具体检查项后的，查看失败详情。

- 您可以根据提示修复后重新进行预检查。
- 如无需修复告警检测项，您也可以选择**确认屏蔽**、**忽略告警项**并重新进行预检查，跳过告警检测项重新进行预检查。

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	耗时: 1376 毫秒 速度: 0.00RPS/(0.000MB/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					

4.3.4. 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。

- 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同步失败。
- 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。

 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**新增同步对象**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[购买流程](#)。

 说明 购买时，选择源实例为PolarDB、目标实例为AnalyticDB 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。
	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。
	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。

目标实例信息类别	配置	说明
	数据库账号	填入云原生数据仓库AnalyticDB PostgreSQL的初始账号，详情请参见 创建数据库账号 。  说明 您也可以填入具备RDS_SUPERUSER权限的账号，创建方法请参见用户权限管理。
	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

- 如果源或目标数据库是阿里云数据库实例（例如RDS MySQL、云数据库MongoDB版等）或ECS上的自建数据库，DTS会自动将对应地区DTS服务的IP地址添加到阿里云数据库实例的白名单或ECS的安全规则中，您无需手动添加，请参见[DTS服务器的IP地址段](#)。
- DTS任务完成或释放后，建议您手动删除添加的DTS服务器IP地址段。

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

注：DTS全量任务运行期间，不要清理DTS任务启动后源库产生的增量数据日志。源库如果过早清理日志，可能会导致DTS增量任务失败

目标已存在表的处理模式：☒ 预检查并报错拦截 ☐ 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

dtstestdata

Tables

Views

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

*源表DMS_ONLINE_DDL过程中是否复制临时表到目标库：☒ 是 ☐ 否

*源库、目标库无法连接后的重试时间 分钟

*为目标对象添加引号 ☒ 是 ☐ 否

取消

上一步

下一步

预检查并启动

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

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。

129

> 文档版本：20220713

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，详情请参见设置同步对象在目标实例中的名称。
映射名称更改	无	如需更改同步对象在目标实例中的名称，请使用对象名映射功能，详情请参见库表列映射。
源表 DMS_ONLINE_DDL过程中是否复制临时表到目标库	无	如源库使用数据管理DMS（Data Management Service）执行Online DDL变更，您可以选择是否同步Online DDL变更产生的临时表数据。 ◦ 是：同步Online DDL变更产生的临时表数据。  说明 Online DDL变更产生的临时表数据过大，可能会导致同步任务延迟。 ◦ 否：不同步Online DDL变更产生的临时表数据，只同步源库的原始DDL数据。  说明 该方案会导致目标库锁表。
源、目标库无法连接重试时间	无	当源、目标库无法连接时，DTS默认重试720分钟（即12小时），您也可以自定义重试时间。如果DTS在设置的时间内重新连接上源、目标库，同步任务将自动恢复。否则，同步任务将失败。  说明 由于连接重试期间，DTS将收取任务运行费用，建议您根据业务需要自定义重试时间，或者在源和目标库实例释放后尽快释放DTS实例。

9. 设置待同步的表在云原生数据仓库AnalyticDB PostgreSQL中的主键列和分布列信息。

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，才能成功启动同步作业。
- 如果预检查失败，单击具体检查项后的，查看失败详情。

- 您可以根据提示修复后重新进行预检查。
- 如无需修复告警检测项，您也可以选择[确认屏蔽](#)、[忽略告警项](#)并重新进行预检查，跳过告警检测项重新进行预检查。

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。

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: 1376 毫秒 速度: 0.00RPS/(0.000MB/s)	按量付费	单向同步	暂停同步 转包年包月 升级更多

4.3.5. 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[购买流程](#)。

 **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。

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_DataHub

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: DataHub

实例地区: 华东1 (杭州)

* Project: dtstestdata

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。

> 文档版本：20220713

132

类别	配置	说明
目标实例信息	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。
	实例类型	固定为DataHub，不可变更。
目标实例信息	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。
	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

dtstestdata

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

dtstestdata (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

* 源库、目标库无法连接后的重试时间

720 分钟

* 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

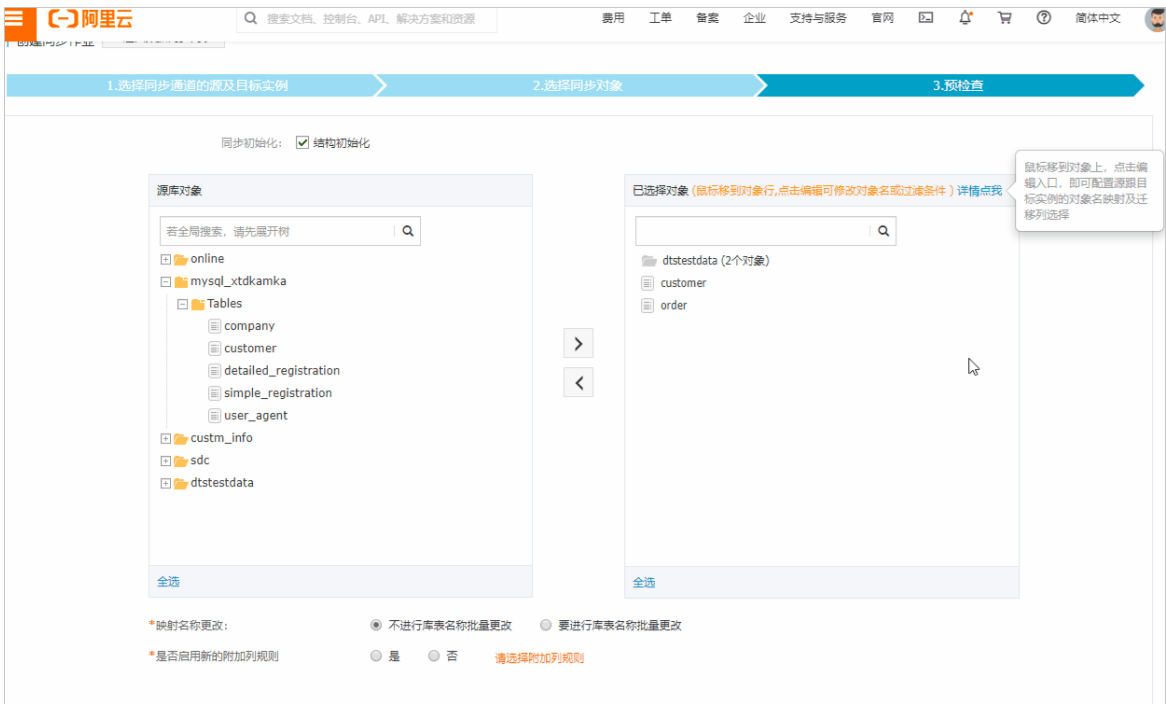
说明

133

> 文档版本：20220713

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。
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，详情请参见设置同步对象在目标实例中的名称。
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。
映射名称更改	如需更改同步对象在目标实例中的名称，请使用对象名映射功能，详情请参见库表列映射。
源表 DMS_ONLINE_DDL过程中是否复制临时表到目标库	如源库使用数据管理DMS（Data Management Service）执行Online DDL变更，您可以选择是否同步Online DDL变更产生的临时表数据。 - 是：同步Online DDL变更产生的临时表数据。  说明 Online DDL变更产生的临时表数据过大，可能会导致同步任务延迟。 - 否：不同步Online DDL变更产生的临时表数据，只同步源库的原始DDL数据。  说明 该方案会导致目标库锁表。
源、目标库无法连接重试时间	当源、目标库无法连接时，DTS默认重试720分钟（即12小时），您也可以自定义重试时间。如果DTS在设置的时间内重新连接上源、目标库，同步任务将自动恢复。否则，同步任务将失败。  说明 由于连接重试期间，DTS将收取任务运行费用，建议您根据业务需要自定义重试时间，或者在源和目标库实例释放后尽快释放DTS实例。

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，才能成功启动同步作业。
- 如果预检查失败，单击具体检查项后的 ⓘ，查看失败详情。

- 您可以根据提示修复后重新进行预检查。
- 如无需修复告警检测项，您也可以选择确认屏蔽、忽略告警项并重新进行预检查，跳过告警检测项重新进行预检查。

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。



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 `dtstestdata` 的前缀。

dtstestdata	dtstestdata	dtstestdata	dtstestdata	dtstestdata	dtstestdata	dtstestdata	dtstestdata	dtstestdata	dtstestdata	dtstestdata
dtstestdata	dtstestdata	dtstestdata	dtstestdata	dtstestdata	dtstestdata	dtstestdata	dtstestdata	dtstestdata	dtstestdata	dtstestdata
10006	dtstestdata	dtstestdata	1574832130000000000	U	dtstestdata	dtstestdata	customer	1574832130	Y	N
10006	dtstestdata	dtstestdata	1574832130000000000	U	dtstestdata	dtstestdata	customer	1574832130	N	Y
10009	dtstestdata	dtstestdata	1574832919000000000	D	dtstestdata	dtstestdata	customer	1574832919	Y	N
10112	dtstestdata	dtstestdata	1574832919000000000	I	dtstestdata	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	数据类型	说明
<code>dtstestdata</code> <code>dtstestdata</code> <code>dtstestdata</code> <code>dtstestdata</code>	<code>new_dts_sync_dts_record_id</code>	String	增量日志的记录ID，为该日志唯一标识。  说明 • 正常情况下是全局唯一自增的，在容灾的情况下会有回退且无法保证自增和唯一。 • 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（分别记录更新前和更新后的值），且 <code>dtstestdata</code> <code>dtstestdata</code> 的值相同。
<code>dtstestdata</code> <code>dtstestdata</code>	<code>new_dts_sync_dts_operation_flag</code>	String	操作类型，取值： • I: INSERT操作。 • D: DELETE操作。 • U: UPDATE操作。
<code>dtstestdata</code> <code>dtstestdata</code>	<code>new_dts_sync_dts_instance_id</code>	String	数据库的server ID。暂不支持显示实际的值，目前固定为 <code>null</code> 。
<code>dtstestdata</code> <code>dtstestdata</code>	<code>new_dts_sync_dts_db_name</code>	String	数据库名称。
<code>dtstestdata</code> <code>dtstestdata</code>	<code>new_dts_sync_dts_table_name</code>	String	表名。
<code>dtstestdata</code> <code>dtstestdata</code>	<code>new_dts_sync_dts_utc_timestamp</code>	String	操作时间戳，即binlog的时间戳（UTC时间）。
<code>dtstestdata</code> <code>dtstestdata</code>	<code>new_dts_sync_dts_before_flag</code>	String	所有列的值是否更新前的值，取值：Y或N。
<code>dtstestdata</code> <code>dtstestdata</code>	<code>new_dts_sync_dts_after_flag</code>	String	所有列的值是否更新后的值，取值：Y或N。

关于dtstestdata_before_flag和dtstestdata_after_flag的补充说明

对于不同的操作类型，增量日志中的 `dtstestdata_before_flag` 和 `dtstestdata_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 `dtb_before_flag` 取值为N，`dtb_after_flag` 取值为Y，示例如下。

dtb_id	dtb_name	dtb_address	dtb_record_id	dtb_operation_flag	dtb_instance_id	dtb_db_name	dtb_table_name	dtb_utc_timestamp	dtb_before_flag	dtb_after_flag
10112		北京市	1574832919000000000	I		dtstestdata	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 `dtb_record_id`、`dtb_operation_flag` 及 `dtb_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 `dtb_before_flag` 取值为Y，`dtb_after_flag` 取值为N。第二条增量日志记录了更新后的值，所以 `dtb_before_flag` 取值为N，`dtb_after_flag` 取值为Y，示例如下。

dtb_id	dtb_name	dtb_address	dtb_record_id	dtb_operation_flag	dtb_instance_id	dtb_db_name	dtb_table_name	dtb_utc_timestamp	dtb_before_flag	dtb_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 `dtb_before_flag` 取值为Y，`dtb_after_flag` 取值为N，示例如下。

dtb_id	dtb_name	dtb_address	dtb_record_id	dtb_operation_flag	dtb_instance_id	dtb_db_name	dtb_table_name	dtb_utc_timestamp	dtb_before_flag	dtb_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](#)。

4.3.6. 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0.10.1.0-2.7.0版本。
- 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[新增同步对象](#)。

操作步骤

1. 购买数据同步作业，详情请参见[购买流程](#)。

 **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 请先点击右侧按钮，获取Topic列表后选择具体的Topic 获取Topic列表

* 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，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。

类别	配置	说明
目标实例信息	实例类型	根据Kafka集群的部署位置选择，本文以ECS上的自建数据库为例介绍配置流程。  说明 当选择为其他实例类型时，您还需要执行相应的准备工作，详情请参见准备工作概览。
	实例地区	购买数据同步实例时选择的目标实例地域，不可变更。
	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

- 如果源或目标数据库是阿里云数据库实例（例如RDS MySQL、云数据库MongoDB版等）或ECS上的自建数据库，DTS会自动将对应地区DTS服务的IP地址添加到阿里云数据库实例的白名单或ECS的安全规则中，您无需手动添加，请参见[DTS服务器的IP地址段](#)。
- DTS任务完成或释放后，建议您手动删除添加的DTS服务器IP地址段。

8. 配置同步策略和同步对象信息。

1.选择同步通道的源及目标实例

4.预检查

同步架构: 单向同步

投递到kafka的数据格式: ☒ DTS Avro ☐ Canal Json ?

同步到kafka Partition策略: ☒ 全部投递至Partition 0 ?
☐ 按库名+表名的hash值投递到不同Partition
☐ 按主键的hash值投递到不同Partition
注: 数据同步作业正式启动后, 请勿修改目标topic的Partition数量, 否则将导致同步失败

源库对象

若全局搜索, 请先展开树

chw

Tables

chw02

test_polar2

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

dtstest0415 源库名:chw (1个对象)

dtstest0415 源表名:tw02

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

* 源库、目标库无法连接后的重试时间 720 分钟 ?

取消

上一步

下一步

鼠标移到对象行, 即可看到目标实例的对象信息。鼠标左键单击, 即可将对象移动到已选对象区域。为保证兼容性, 将大写库名映射为小写, 库名映射需要保持大写。

配置	说明
投递到kafka的数据格式	同步到Kafka集群中的数据以avro格式或者Canal Json格式存储, 定义详情请参见 Kafka集群的数据存储格式 。
同步到Kafka Partition策略	根据业务需求选择同步的策略, 详细介绍请参见 Kafka Partition同步策略说明 。
同步对象	在源库对象区域框中, 选择需要同步的对象 (选择的粒度为表), 然后单击  图标将其移动到已选对象区域框中。 ? 说明 DTS会自动将表名映射为步骤6选择的Topic名称。如果需要更换同步的目标Topic, 请参见步骤9。
映射名称更改	如需更改同步对象在目标实例中的名称, 请使用对象名映射功能, 详情请参见 库表列映射 。

配置	说明
源、目标库无法连接重试时间	当源、目标库无法连接时，DTS默认重试720分钟（即12小时），您也可以自定义重试时间。如果DTS在设置的时间内重新连接上源、目标库，同步任务将自动恢复。否则，同步任务将失败。  说明 由于连接重试期间，DTS将收取任务运行费用，建议您根据业务需要自定义重试时间，或者在源和目标库实例释放后尽快释放DTS实例。

9. （可选）在已选择对象区域框中，将鼠标指针放置在目标Topic名上，然后单击Topic名后出现的编辑，在弹出的对话框中设置源表在目标Kafka集群中的Topic名称、Topic的Partition数量、Partition Key等信息。

编辑表

注意：编辑表名或列名后，目标数据库的表名列名将为修改后的名称。
编辑表名后会根据新的表名创建1个Topic，如果目标端已有同名的Topic，则不会创建新的Topic

* 数据库表名称：

dtstopic_new

过滤条件：

id>1000

验证语法

验证通过

* 设置新建Topic的Partition数量：

12

全选

列名

类型

Partition Key

address

varchar(32)

id

int(11)

name

varchar(32)

确定

配置	说明
----	----

141

> 文档版本：20220713

配置	说明
数据库表名称	设置源表同步到的目标Topic名称。  说明 如果设置的Topic名称在目标Kafka集群中不存在，您还需要设置该Topic的Partition数量。
过滤条件	- 过滤条件支持标准的SQL WHERE语句（仅支持 <code>=</code>、<code>!=</code>、<code><</code> 和 <code>></code> 操作符），只有满足WHERE条件的数据才会被同步到目标Topic。本案例填入 <code>id>1000</code>。 - 过滤条件中如需使用引号，请使用单引号（<code>'</code>），例如 <code>address in('hangzhou','shanghai')</code>。
设置新Topic的Partition数量	在下拉列表中，选择新Topic的Partition数量。  说明 只有当设置的目标Topic名称在目标Kafka集群中不存在时，您才需要配置本参数。
设置Partition Key	当您在步骤8中选择同步策略为按主键的hash值投递到不同Partition时，您可以配置本参数，指定单个或多个列作为Partition Key来计算Hash值，DTS将根据计算得到的Hash值将不同的行投递到目标Topic的各Partition中。

10. 上述配置完成后单击页面右下角的下一步。

11. 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

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](#)

过滤选项：☒ 忽略增量同步阶段的 DDL

取消

上一步

保存

预检查并启动

配置	说明
同步初始化	默认选择结构初始化和全量数据初始化，DTS会在增量数据同步之前，将源库中待同步对象的结构和存量数据，同步到目标库。
过滤选项	默认选择忽略增量同步阶段的 DDL，即增量同步阶段源库执行的DDL操作不会被DTS同步至目标库。

12. 上述配置完成后，单击页面右下角的预检查并启动。

? 说明

- 在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，才能成功启动同步作业。
- 如果预检查失败，单击具体检查项后的，查看失败详情。
 - 您可以根据提示修复后重新进行预检查。
 - 如无需修复告警检测项，您也可以选择**确认屏蔽**、**忽略告警项**并重新进行预检查，跳过告警检测项重新进行预检查。

13. 在**预检查**对话框中显示**预检查**通过后，关闭**预检查**对话框，数据同步作业正式开始。

您可以在**数据同步**页面，查看数据同步状态。

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				

4.3.7. 从ECS上的自建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ECS上的自建MySQL为例介绍配置流程。

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

- 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

操作类型	SQL操作语句
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[源库存在触发器时如何配置同步作业](#)。
- 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

为自建MySQL创建账号并设置binlog

 **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[数据同步拓扑介绍](#)。

操作步骤

1. 购买数据同步作业，详情请参见[购买流程](#)。

 **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

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。
	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。

145

> 文档版本：20220713

类别 目标实例信息	配置	说明
	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

- 如果源或目标数据库是阿里云数据库实例（例如RDS MySQL、云数据库MongoDB版等）或ECS上的自建数据库，DTS会自动将对应地区DTS服务的IP地址添加到阿里云数据库实例的白名单或ECS的安全规则中，您无需手动添加，请参见[DTS服务器的IP地址段](#)。
- DTS任务完成或释放后，建议您手动删除添加的DTS服务器IP地址段。

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 (DML+DDL)

源库对象

若全局搜索，请先展开树

dtstestdata

Tables

Views

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

*源表DMS_ONLINE_DDL过程中是否复制临时表到目标库:

☐ 是

☒ 否

*源库、目标库无法连接后的重试时间

720

分钟

取消

上一步

下一步

配置项目	配置说明
------	------

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，您可以设置同步对象在目标实例中的名称来避免表名冲突。 ◦ 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。
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，详情请参见设置同步对象在目标实例中的名称。
映射名称更改	如需更改同步对象在目标实例中的名称，请使用对象名映射功能，详情请参见库表列映射。
源表DMS_ONLINE_DDL过程中是否复制临时表到目标库	如源库使用数据管理DMS (Data Management Service) 执行Online DDL变更，您可以选择是否同步Online DDL变更产生的临时表数据。 ◦ 是：同步Online DDL变更产生的临时表数据。  说明 Online DDL变更产生的临时表数据过大，可能会导致同步任务延迟。 ◦ 否：不同步Online DDL变更产生的临时表数据，只同步源库的原始DDL数据。  说明 该方案会导致目标库锁表。

配置项目	配置说明
源、目标库无法连接重试时间	当源、目标库无法连接时，DTS默认重试720分钟（即12小时），您也可以自定义重试时间。如果DTS在设置的时间内重新连接上源、目标库，同步任务将自动恢复。否则，同步任务将失败。  说明 由于连接重试期间，DTS将收取任务运行费用，建议您根据业务需要自定义重试时间，或者在源和目标库实例释放后尽快释放DTS实例。

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，才能成功启动同步作业。
 - 如果预检查失败，单击具体检查项后的, 查看失败详情。
- 您可以根据提示修复后重新进行预检查。

■ 如无需修复告警检测项，您也可以选择**确认屏蔽**、**忽略告警项**并重新进行预检查，跳过告警检测项重新进行预检查。

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

耗时: 1376 毫秒
速度: 0.00RPS/(0.000MB/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

>

»

4.3.8. 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[本地IDC接入至阿里云方案概览](#)。

 **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[配置DTS与专线、VPN网关或智能接入网关间的路由](#)。

- 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）。
- 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[数据同步拓扑介绍](#)。

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[源库存在触发器时如何配置同步作业](#)。
- 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

为自建MySQL创建账号并设置binlog

 **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[购买流程](#)。

 **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)

* 已和源端数据库联通的VPC: 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。
	已和源端数据库联通的VPC	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。

目标实例信息类别	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

- 如果源或目标数据库是阿里云数据库实例（例如RDS MySQL、云数据库MongoDB版等）或ECS上的自建数据库，DTS会自动将对应地区DTS服务的IP地址添加到阿里云数据库实例的白名单或ECS的安全规则中，您无需手动添加，请参见[DTS服务器的IP地址段](#)。
- DTS任务完成或释放后，建议您手动删除添加的DTS服务器IP地址段。

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 (DML+DDL)

源库对象

若全局搜索，请先展开树

dtstestdata

Tables

Views

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

*源表DMS_ONLINE_DDL过程中是否复制临时表到目标库:

☐ 是

☒ 否

*源库、目标库无法连接后的重试时间

720

分钟

取消

上一步

下一步

配置项目	配置说明
------	------

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，您可以设置同步对象在目标实例中的名称来避免表名冲突。 ◦ 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。
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，详情请参见设置同步对象在目标实例中的名称。
映射名称更改	如需更改同步对象在目标实例中的名称，请使用对象名映射功能，详情请参见库表列映射。
源表DMS_ONLINE_DDL过程中是否复制临时表到目标库	如源库使用数据管理DMS (Data Management Service) 执行Online DDL变更，您可以选择是否同步Online DDL变更产生的临时表数据。 ◦ 是：同步Online DDL变更产生的临时表数据。  说明 Online DDL变更产生的临时表数据过大，可能会导致同步任务延迟。 ◦ 否：不同步Online DDL变更产生的临时表数据，只同步源库的原始DDL数据。  说明 该方案会导致目标库锁表。

配置项目	配置说明
源、目标库无法连接重试时间	当源、目标库无法连接时，DTS默认重试720分钟（即12小时），您也可以自定义重试时间。如果DTS在设置的时间内重新连接上源、目标库，同步任务将自动恢复。否则，同步任务将失败。  说明 由于连接重试期间，DTS将收取任务运行费用，建议您根据业务需要自定义重试时间，或者在源和目标库实例释放后尽快释放DTS实例。

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，才能成功启动同步作业。
 - 如果预检查失败，单击具体检查项后的, 查看失败详情。
- 您可以根据提示修复后重新进行预检查。

■ 如无需修复告警检测项，您也可以选择**确认屏蔽**、**忽略告警项**并重新进行预检查，跳过告警检测项重新进行预检查。

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: 1376 毫秒

速度: 0.00RPS/(0.000MB/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. 账号管理

5.1. 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，且以阿里云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，可以管理集群中所有的普通账号和数据库。 • 可以创建数据库和普通账号，并且把某个数据库授权给某个普通账号，从而使该普通账号有权限在该数据库中进行增删改查。 • 开放了更多权限，可满足个性化和精细化的权限管理需求，例如可为不同用户分配不同表的查询权限。 • 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。 • 无创建数据库和普通账号的权限，只能管理被授权的数据库。 • 不能管理其他账号，也不能断开其他账号的连接。

相关API

API	描述
CreateAccount	创建账号。
DescribeAccounts	查看账号列表。
ModifyAccountDescription	修改账号备注。
ModifyAccountPassword	修改账号密码。
GrantAccountPrivilege	账号授权。
RevokeAccountPrivilege	撤销账号权限。
ResetAccount	重置账号权限。

API	描述
DeleteAccount	删除账号。

5.2. 控制台账号

5.2.1. 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](#)，单击右上角的立即注册。
- 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](#)。

 扫码快速登录

 账号密码登录

输入邮箱/账号名/8位ID

请输入登录密码

登录

[忘记密码](#) | [忘记账号名](#) | [免费注册](#)

其它登录方式

RAM用户登录

- 子账号的[登录入口](#)。

[使用主帐号登录](#)

子用户登录

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

5.2.2. 创建和管理RAM用户

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则需要创建RAM用户。本文介绍如何创建RAM用户并为RAM用户授权以及RAM用户如何登录控制台。

前提条件

登录阿里云账号或有RAM权限的子账号。

- 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`。

操作步骤

- 创建RAM用户，请参见[创建RAM用户](#)。
- 在用户页面下为RAM用户授权，请参见[在用户页面为RAM用户授权](#)。
- 在授权页面下为RAM用户授权，请参见[在授权页面为RAM用户授权](#)。
- 使用RAM用户登录控制台，请参见[RAM用户登录阿里云控制台](#)。

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](#)。

5.2.3. 通过自定义策略授权RAM用户管理PolarDB

如果RAM提供的系统策略无法满足您的业务需求，您可以通过创建自定义策略对云原生关系型数据库PolarDB进行精细化权限管理（例如资源或操作级别的授权）。

前提条件

使用RAM进行权限管理前，请确保您已完成[账号注册](#)。

背景信息

- 权限策略是用语法结构描述的一组权限的集合，可以精确地描述被授权的资源集、操作集以及授权条件，详细的语言规范请参见[权限策略语法和结构](#)。
- 使用自定义策略对PolarDB进行精细化权限管理前，请先了解PolarDB的权限定义，详情请参见[RAM资源授权](#)。

 **说明** 如果需要自定义各种权限类型组合或授予某些表级别权限等场景，您可以通过数据管理DMS推出的数据库账号权限管理功能进行灵活管控，详情请参见[MySQL数据库账号权限管理](#)。

操作步骤

1. 创建自定义权限策略。具体操作请参见[创建自定义权限策略](#)。

常见的自定义权限策略：

- 示例1：授权目标RAM用户管理2个指定的PolarDB集群。
假设您的账号下拥有多个PolarDB集群，但作为RAM管理员，您希望仅授权其中的2个集群（集群ID分别为i-001和i-002）给目标RAM用户，那么您可以创建如下权限策略：

```
{
  "Statement": [
    {
      "Action": "polardb:*",
      "Effect": "Allow",
      "Resource": [
        "acs:polardb:*:*:/i-001",
        "acs:polardb:*:*:/i-002"
      ]
    },
    {
      "Action": "polardb:Describe*",
      "Effect": "Allow",
      "Resource": "*"
    }
  ],
  "Version": "1"
}
```

说明

- 被授予该权限策略的RAM用户号可以查看所有的集群及资源，但只能管理已被授权的2个集群（即集群i-001和集群i-002）。同时作为RAM管理员，您仍然可以使用API、CLI或SDK直接管理上述两个集群。
- `Describe*` 在权限策略中是必须的，否则被授权的RAM用户无法在控制台看到任何集群。

- 示例2：仅授权PolarDB的部分功能给目标RAM用户。
假设您希望仅授权PolarDB的部分功能给目标RAM用户，则可以创建如下权限策略：

```
{
  "Statement": [
    {
      "Action": [
        "polardb:Describe*",
        "polardb:CreateBackup",
        "polardb>DeleteBackup",
        "polardb:ModifyDBClusterAccessWhitelist"
      ],
      "Resource": "*",
      "Effect": "Allow"
    }
  ],
  "Version": "1"
}
```

② 说明

- 被授予该权限策略的目标RAM用户仅可以对账号下所有的PolarDB集群进行集群信息查询、备份查询、创建备份、删除备份、修改白名单操作，但不允许进行其他任何操作。
- PolarDB支持通过RAM进行API级别的访问控制，您可以通过相关API对PolarDB进行细粒度的权限访问控制，详情请参见[支持RAM的云服务](#)和[API概览](#)。

2. 为RAM用户授权自定义策略。具体操作请参见[为RAM用户授权](#)。

5.3. 数据库账号

5.3.1. 创建数据库账号

本文为您介绍如何创建PolarDB MySQL引擎数据库账号，以及高权限账号与普通账号的区别。

背景信息

PolarDB支持高权限账号和普通账号这两种数据库账号，您可以在控制台管理所有账号。

② 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，可以管理集群中所有的普通账号和数据库。 • 可以创建数据库和普通账号，并且把某个数据库授权给某个普通账号，从而使该普通账号有权限在该数据库中进行增删改查。 • 开放了更多权限，可满足个性化和精细化的权限管理需求，例如可为不同用户分配不同表的查询权限。 • 可以断开任意账号的连接。

账号类型	说明
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。 - 无创建数据库和普通账号的权限，只能管理被授权的数据库。 - 不能管理其他账号，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. 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 > 账号管理**。
5. 单击 **创建账号**。
6. 在创建账号面板，设置以下参数：

参数	说明
账号名	填写账号名称。账号名称需符合如下要求： - 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等。
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。
密码	设置账号的密码。密码需符合如下要求： - 至少包含大写字母、小写字母、数字或特殊字符中的任意三种。 - 长度为8~32个字符。 - 特殊字符为： !@#%&*()_+ =
确认密码	再次输入密码。
备注说明	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 <code>http://</code> 或 <code>https://</code> 开头。 - 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. 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 > 账号管理。
5. 单击创建账号。
6. 在创建账号面板，设置以下参数：

参数	说明
账号名	填写账号名称。账号名称需符合如下要求： ◦ 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等。
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。 - 从未授权数据库列表选择一个或多个数据库，单击图标将其添加到已授权数据库列表中。 - 在已授权数据库列表中，为目标数据库选择读写、只读、仅DML、仅DDL或只读+索引权限。  说明 如果您需要自定义各种权限类型组合或授予某些表级别权限等，请单击未授权数据库列表下方的自定义权限，通过数据管理DMS推出的数据库账号权限管理功能进行灵活管控，详情请参见MySQL数据库账号权限管理。
密码	设置账号的密码。密码需符合如下要求： ◦ 至少包含大写字母、小写字母、数字或特殊字符中的任意三种。 ◦ 长度为8~32个字符。 ◦ 特殊字符为： !@#%&*()_+ -=
确认密码	再次输入密码。
备注说明	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 <code>http://</code> 或 <code>https://</code> 开头。 ◦ 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），您可以通过重置高权限账号的权限，使其恢复到初始状态。

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 > 账号管理。
5. 单击目标账号操作栏中的重置权限。
6. 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。

下一步

申请集群地址和主地址

相关API

API	描述
CreateAccount	创建账号。
DescribeAccounts	查看账号列表。
ModifyAccountDescription	修改账号备注。
ModifyAccountPassword	修改账号密码。
GrantAccountPrivilege	账号授权。
RevokeAccountPrivilege	撤销账号权限。
ResetAccount	重置账号权限。

5.3.2. 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引擎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

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，单击配置与管理 > 账号管理。
5. 找到目标账号，单击操作栏中的修改密码。

账号名	状态	所属数据库	备注	类型	操作
	可用		-	高权限账号	重置权限 修改密码 删除
	可用		-	普通账号	修改权限 修改密码 删除

6. 在弹出的对话框中，输入并确认新密码，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. 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，单击配置与管理 > 账号管理。
5. 找到目标账号，单击操作栏中的重置权限。

账号名	状态	所属数据库	备注	类型	操作
	可用		-	高权限账号	重置权限 修改密码 删除
	可用		-	普通账号	修改权限 修改密码 删除

6. 在弹出的对话框中，单击确定。

修改普通账号权限

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，单击配置与管理 > 账号管理。
5. 找到目标账号，单击操作栏中的修改权限。

账号名	状态	所属数据库	备注	类型	操作
	可用		-	高权限账号	重置权限 修改密码 删除
	可用		-	普通账号	修改权限 修改密码 删除

6. 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. 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，单击配置与管理 > 账号管理。
5. 找到目标账号，单击操作栏中的删除。

账号名	状态	所属数据库	备注	类型	操作
	可用		-	高权限账号	重置权限 修改密码 删除
	可用		-	普通账号	修改权限 修改密码 删除

6. 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引擎 8.0版本：

<pre>ALTER USER {username} IDENTIFIED BY '{password}'</pre>	
参数	说明
username	待修改密码的账号。
password	账号的密码。

- PolarDB MySQL引擎 5.6或5.7版本：

<pre>SET PASSWORD FOR 'username'@'host' = PASSWORD('password')</pre>	
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。

<pre>GRANT privileges ON databasename.tablename TO 'username'@'host' WITH GRANT OPTION;</pre>	
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（*）。
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
CreateAccount	创建账号。
DescribeAccounts	查看账号列表。
ModifyAccountDescription	修改账号备注。
ModifyAccountPassword	修改账号密码。
GrantAccountPrivilege	账号授权。
RevokeAccountPrivilege	撤销账号权限。
ResetAccount	重置账号权限。
DeleteAccount	删除账号。

5.3.3. 账号权限

本文介绍PolarDB高权限账号和普通账号对应的权限。

高权限账号

PolarDB高权限账号拥有如下权限。

账号类型	权限
高权限账号	SELECT、INSERT、UPDATE、DELETE、CREATE DROP、RELOAD、PROCESS、REFERENCES、INDEX ALTER、LOCK TABLES、EXECUTE、TRIGGER、CREATE TEMPORARY TABLES REPLICATION SLAVE、REPLICATION CLIENT、CREATE VIEW、SHOW VIEW CREATE ROUTINE、ALTER ROUTINE、CREATE USER、EVENT

普通账号

PolarDB普通账号拥有如下权限。

权限类型	权限
读写	SELECT、INSERT、UPDATE、DELETE、CREATE、EXECUTE DROP、REFERENCES、INDEX、ALTER、CREATE TEMPORARY TABLES EVENT、CREATE VIEW、SHOW VIEW、CREATE ROUTINE、ALTER ROUTINE LOCK TABLES、TRIGGER、PROCESS、REPLICATION SLAVE、REPLICATION CLIENT
只读	SELECT、LOCK TABLES、SHOW VIEW PROCESS、REPLICATION SLAVE、REPLICATION CLIENT
仅DML	SELECT、INSERT、UPDATE、DELETE、LOCK TABLES CREATE TEMPORARY TABLES、EXECUTE、TRIGGER、EVENT SHOW VIEW、PROCESS、REPLICATION SLAVE、REPLICATION CLIENT

权限类型	权限
仅DDL	CREATE、DROP、INDEX、ALTER、CREATE TEMPORARY TABLES CREATE VIEW、SHOW VIEW、CREATE ROUTINE、ALTER ROUTINE LOCK TABLES、PROCESS、REPLICATION SLAVE、REPLICATION CLIENT
只读+索引	SELECT、INDEX、LOCK TABLES、SHOW VIEW

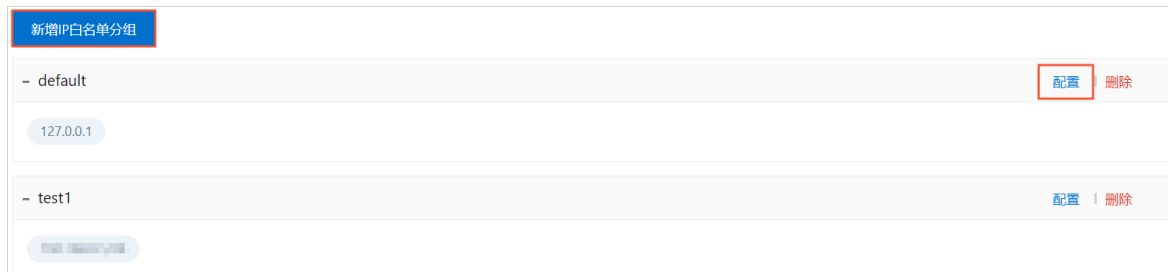
5.3.4. 系统账号说明

PolarDB MySQL引擎提供了多种类型的系统内部账号，用于后台管控运维等操作。通常情况下您不需要关注系统账号的权限及操作。

账号名称	说明
root（5.6和8.0版本） aliyun_root（5.7版本）	本地运维账号。用于实例管控，例如修改内核参数、查询实例状态等。
aurora	远程管控账号。实例故障时，管控系统通过该账号登录实例执行管控操作，例如主备切换、实例监控等。
replicator	主从复制账号。用于主节点和备节点之间的数据同步。

 **说明** 在您创建数据库账号时，请勿与以上系统账号重名。

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，单击配置与管理 > 集群白名单。
5. 在集群白名单页面，您可以新增IP白名单分组或配置已有白名单。



- o 新增白名单分组
 - a. 单击新增IP白名单分组。

b. 在新增IP白名单分组对话框，输入分组名称和允许访问的IP白名单地址。

新增白名单

只有已添加到白名单中的IP地址才可以访问该POLARDB集群

- 可以填写IP地址（如192.168.0.1）或IP段（如192.168.0.0/24）

- 多个IP需要用英文逗号隔开，如192.168.0.1,192.168.0.0/24

- 127.0.0.1表示禁止任何IP地址访问

* 分组名称

请输入白名单分组名称，如：my_polardb_ip_list0/120

由小写字母、数字、下划线（_）组成，字母开头，字母或数字结尾，长度2~120个字符

* 组内白名单

请输入IP地址，如：192.168.0.1,192.168.100.0/24

新白名单将于1分钟后生效

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

确定

取消

② 说明 IP白名单分组名称需满足如下要求：

- 由小写字母、数字、下划线（_）组成。
- 由字母开头、字母或数字结尾。
- 长度为2~120个字符。

o. 配置白名单

a. 单击目标IP白名单分组名称右侧的配置。

b. 在配置白名单对话框，输入允许访问的IP白名单地址。

配置白名单

只有已添加到白名单中的IP地址才可以访问该POLARDB集群

- 可以填写IP地址（如192.168.0.1）或IP段（如192.168.0.0/24）

- 多个IP需要用英文逗号隔开，如192.168.0.1,192.168.0.0/24

- 127.0.0.1表示禁止任何IP地址访问

* 分组名称

abc1236/120

由小写字母、数字、下划线（_）组成，字母开头，字母或数字结尾，长度2~120个字符

* 组内白名单

127.0.0.1

新白名单将于1分钟后生效

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

确定

取消

说明

- 每个集群都默认包含一个 default 的白名单分组，且只包含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。

6. 单击确定即可。

说明 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。

相关API

API	描述
DescribeDBClusterAccessWhitelist	查看允许访问数据库集群的IP名单。
ModifyDBClusterAccessWhitelist	修改允许访问数据库集群的IP名单。

6.1.2. 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。本文将介绍如何设置ECS安全组。

使用场景

创建PolarDB MySQL引擎集群后，暂时无法访问该集群。您还需要在PolarDB MySQL引擎集群白名单中添加ECS安全组后，该安全组中的ECS实例才可以访问PolarDB MySQL引擎集群。

说明

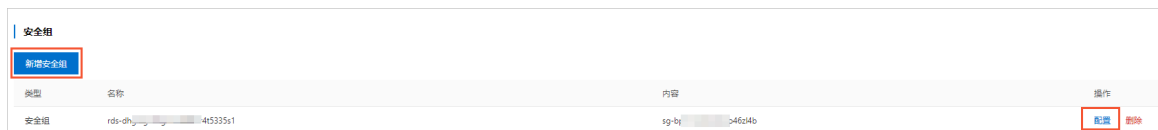
- 关于ECS安全组的更多信息，以及如何在ECS中配置安全组，请参见[创建安全组](#)。
- 您可以在白名单中配置ECS的IP，IP白名单和安全组中的ECS实例都可以访问该PolarDB MySQL引擎集群。

注意事项

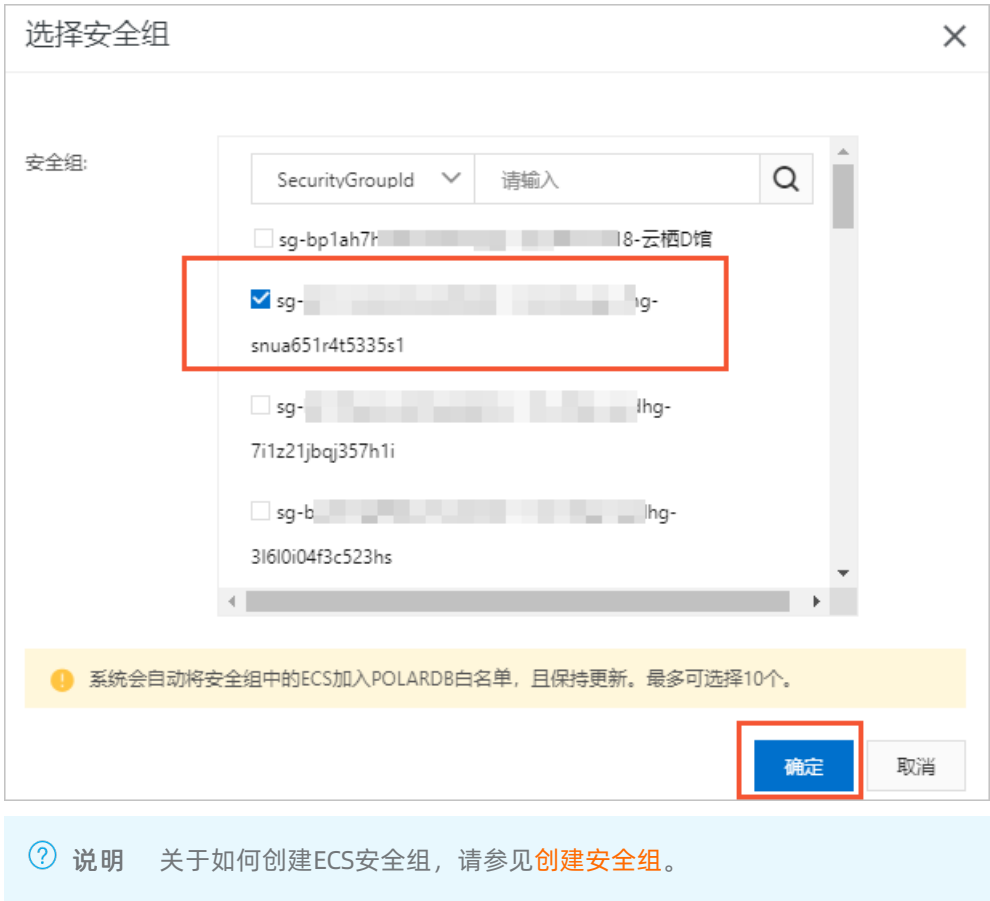
- PolarDB MySQL引擎集群中只能添加与自身网络类型相同的ECS安全组。例如，集群为专有网络VPC时，只能添加VPC类型的ECS安全组。
- 一个PolarDB MySQL引擎集群中最多可以添加10个安全组。

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，单击集群ID。
4. 在左侧导航栏，单击配置与管理 > 集群白名单。
5. 在集群白名单页面，您可以单击新增安全组或单击目标安全组操作栏中的配置按钮来更改安全组设置。



6. 在选择安全组对话框，选中目标安全组，单击确定即可。



相关API

API	描述
DescribeDBClusterAccessWhitelist	查看允许访问数据库集群的IP名单。
ModifyDBClusterAccessWhitelist	修改允许访问数据库集群的IP名单。

6.2. 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。

背景信息

SSL是Netscape公司所提出的安全保密协议，在浏览器和Web服务器之间构造安全通道来进行数据传输，采用RC4、MD5、RSA等加密算法实现安全通讯。国际互联网工程任务组（IETF）对SSL 3.0进行了标准化，标准化后更名为安全传输层协议（TLS）。由于SSL这一术语更为常用，因此本文所述SSL加密实际是指TLS加密。

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正常工作。更新证书有效期的详细说明，请参见[更新证书有效期](#)。
- 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。

- 修改受保护地址后，SSL证书会自动更新，同时重启您的数据库实例，请谨慎操作。
- 更新证书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。
- 开启SSL加密的PolarDB连接地址需要少于64个字符，如何修改连接地址请参见[配置数据库代理](#)。

开启SSL加密和下载证书

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。



说明

- PolarDB MySQL引擎5.6版本和PolarDB MySQL引擎5.7版本的集群仅支持为主地址配置SSL。
- PolarDB MySQL引擎8.0版本的集群支持为主地址、集群地址和自定义地址配置SSL。

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格式文件：用于Windows系统中导入CA证书。
- .pem格式文件：用于其他系统或应用中导入CA证书。
- .jks格式文件：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。

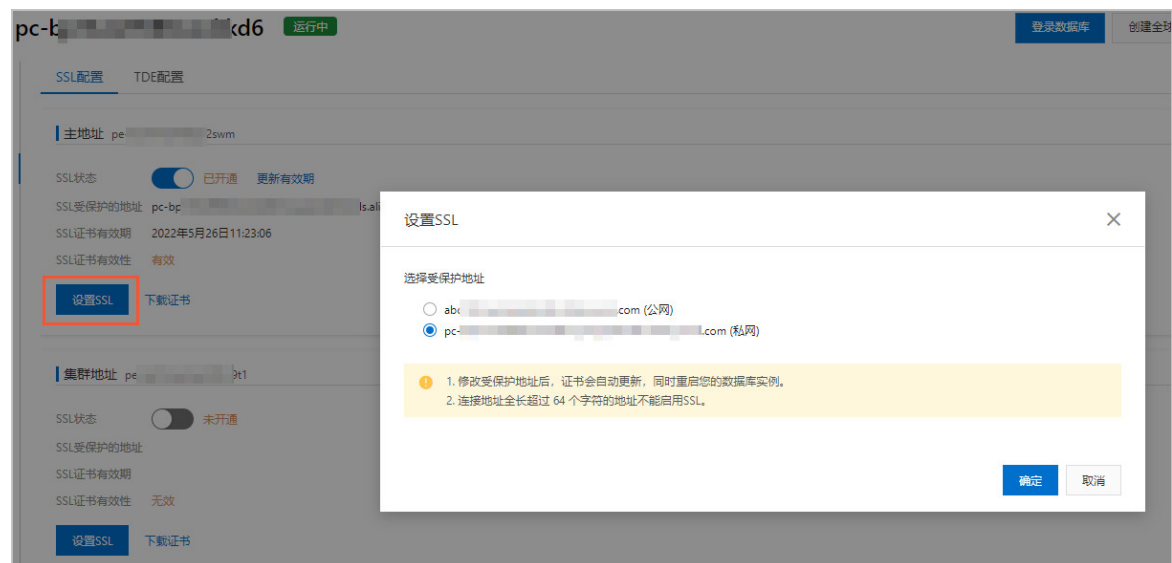
② 说明 在Java中使用JKS证书文件时，jdk7和jdk8需要修改默认的jdk安全配置，在连接PolarDB数据库的服务器的 `jre/lib/security/java.security` 文件中，修改如下两项配置：

```
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
```

如果您需要修改SSL受保护的地址，可以单击设置SSL进行修改。



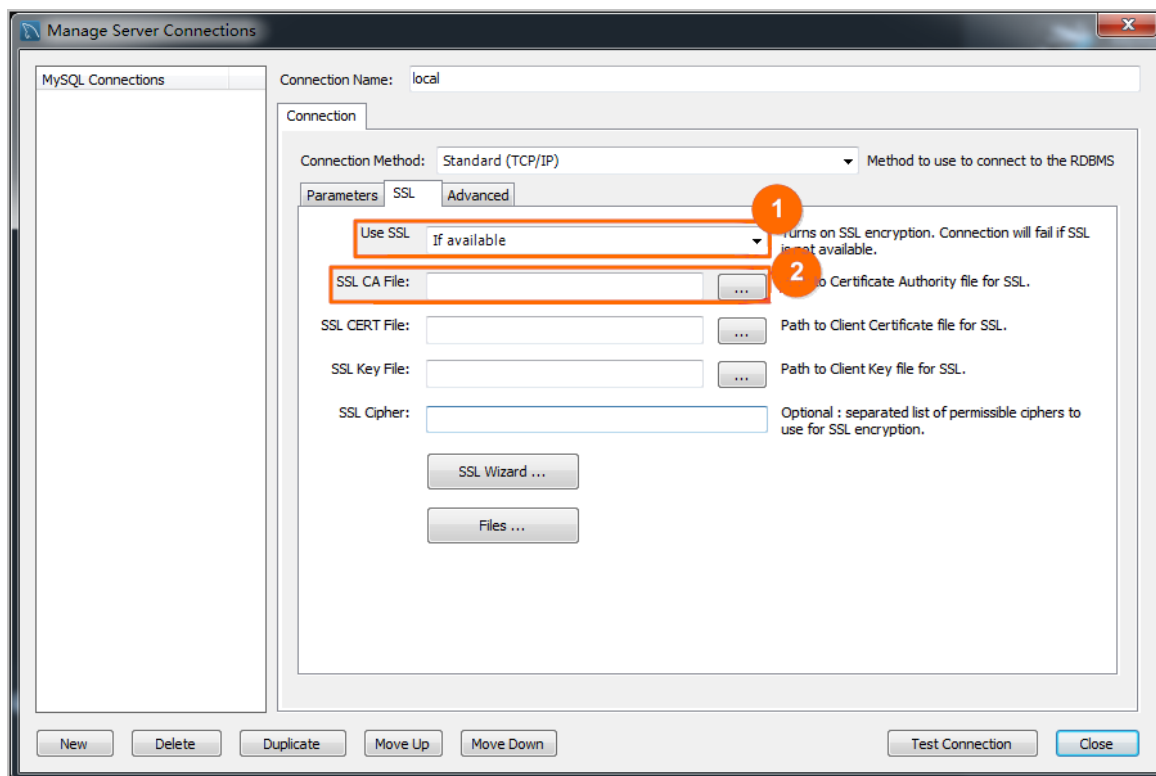
🔊 注意 修改受保护地址后，SSL证书会自动更新，同时重启您的数据库实例，请谨慎操作。

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证书的配置方法。其它应用或者客户端请参见对应产品的使用说明。

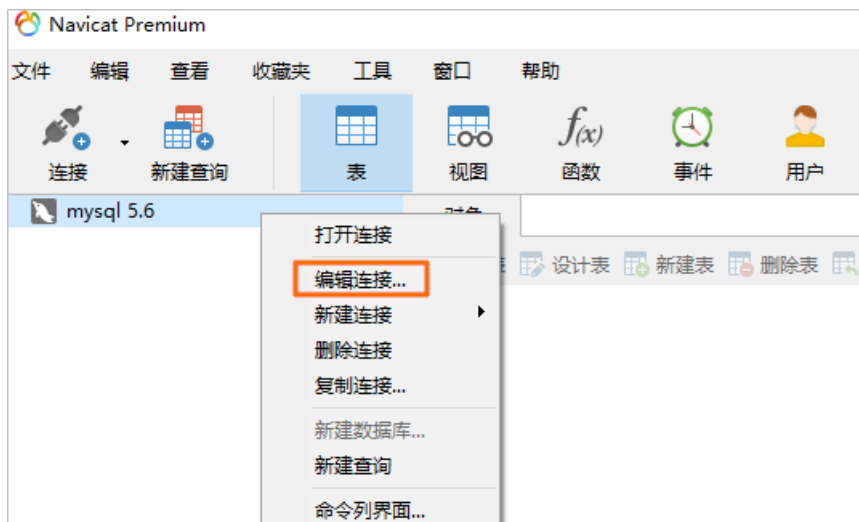
MySQL Workbench配置方法

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。

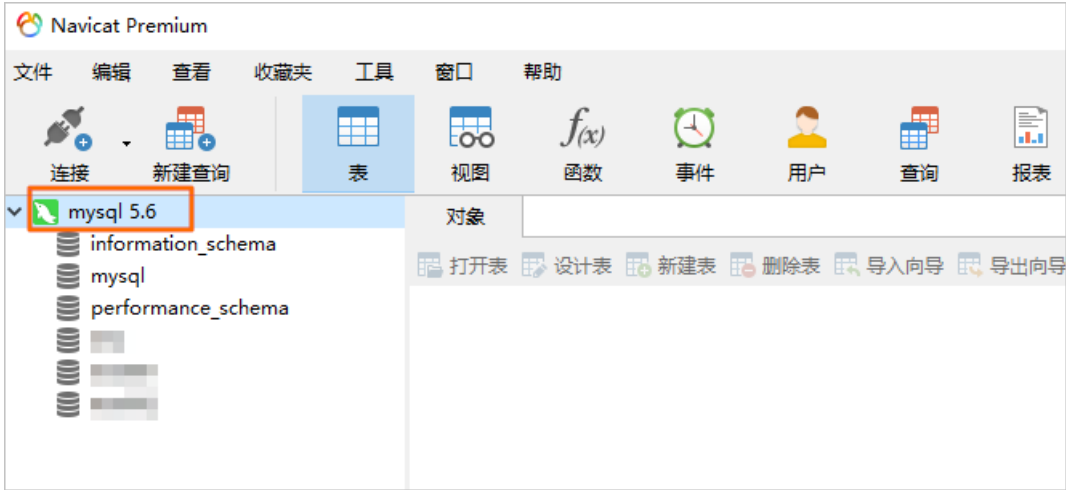


4. 单击确定。

 说明

如果报 Connection with same connection name already exists in the project. 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，您需要手动更新证书有效期，以下内容将为您介绍如何更新证书有效期。

 **说明** 更新证书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。

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. 在弹出的对话框中，单击确定。
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](#)。

开启证书自动轮换

开启证书自动轮换后，在证书即将过期的10天内，PolarDB会在集群的可维护时间窗口内自动更新证书。

 **说明** 自动更新证书将会重启集群，请您谨慎选择。

1. 在SSL配置页签中，单击高级选项。

SSL配置 TDE配置

主地址 pe-xxxxx-im

SSL状态 ☒ 已开通 [更新有效期](#)

SSL受保护的地址 pc-bp-xxxxx.com

SSL证书有效期 2022年11月16日 11:56:17

SSL证书有效性 有效

设置SSL 下载证书

集群地址 pe-xxxxx-r0ly

SSL状态 ☐ 未开通

SSL受保护的地址

SSL证书有效期

SSL证书有效性 无效

设置SSL 下载证书

高级选项

2. 在高级选项对话框中，选择开启证书自动轮转，并单击确定。

高级选项

证书自动轮转 ☒ 开启 ☐ 关闭

⚠ 轮转触发的时间点、规则、会导致实例重启等。

确认 取消

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状态右侧滑块来关闭SSL加密。
6. 在弹出的对话框中，单击确定。

相关API

API	描述
DescribeDBClusterSSL	调用DescribeDBClusterSSL接口查询PolarDB集群SSL设置。
ModifyDBClusterSSL	调用ModifyDBClusterSSL接口设置PolarDB集群SSL加密的开通、关闭或更新CA证书。

6.3. 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

- 仅产品系列为**集群版**或**单节点**的PolarDB MySQL引擎集群支持TDE，**历史库**不支持TDE。集群版和单节点的PolarDB MySQL引擎集群须满足如下版本要求：

产品系列	版本	是否支持TDE
集群版	5.6	Revision version需为5.6.1.0.21或以上。
	5.7	Revision version需为5.7.1.0.3或以上。
	8.0	Revision version需为8.0.1.1.1或以上。
单节点	5.6	Revision version需为5.6.1.0.21或以上。
	5.7	Revision version需为5.7.1.0.3或以上。
	8.0	Revision version需为8.0.1.1.1或以上。
历史库	8.0	不支持。

- 已开通KMS。具体操作请参见[开通密钥管理服务](#)。
- 已授权RDS访问KMS。具体操作请参见[授权RDS访问KMS](#)。

背景信息

PolarDB MySQL引擎的TDE加密采用国际流行的AES算法，密钥长度为256比特。TDE加密使用的密钥由密钥管理服务（KMS）产生和管理，PolarDB MySQL引擎不提供加密所需的密钥和证书。部分可用区不仅可以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 MySQL引擎使用。

注意事项

- 如果是IO bound场景，开通TDE后，可能会对数据库性能产生一定影响。
- 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。

操作步骤

注意

- 启用TDE加密会导致PolarDB集群重启，请谨慎操作。
- TDE开通后将无法关闭。

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. 在左侧导航栏单击配置与管理 > 安全管理。
5. 在TDE配置页签，打开TDE状态开关。



6. 在设置TDE对话框中，选择使用服务密钥（阿里云自动生成）或使用已有自定义密钥。

设置TDE

☒ 使用服务密钥（阿里云自动生成）

☐ 使用已有自定义密钥

高级选项:

☒ 开启后，所有新建的表自动加密

ⓘ 开通注意事项

1. 开启TDE会重启PolarDB实例。

2. 禁用密钥，设置了密钥计划删除，或者删除了密钥材料都会造成密钥不可用。

3. 撤销授权关系后，重启以后会造成PolarDB实例不可用。

4. 当前页面操作人必须是主账号或者是具有AliyunSTSAssumeRoleAccess权限的账号。

5. 开通TDE大约10分钟完成。

6. 目前支持的密钥类型为 Aliyun_AES_256， Aliyun_SM4。

确定

取消

❓ 说明 目前TDE支持的密钥类型为 Aliyun_AES_256 、 Aliyun_SM4 。

- 如果您选择使用服务密钥（阿里云自动生成），单击确定，开通TDE。
- 如果您选择使用已有自定义密钥，下拉选择由密钥管理服务（KMS）产生的密钥，单击确定，开通TDE。

设置TDE

使用服务密钥（阿里云自动生成）

使用已有自定义密钥

请选择

ee1c

caa1a19

高级选项:

开启后，所有新建的表自动加密

⚠ 开通注意事项

1. 开启TDE会重启PolarDB实例。

2. 禁用密钥，设置了密钥计划删除，或者删除了密钥材料都会造成密钥不可用。

3. 撤销授权关系后，重启以后会造成PolarDB实例不可用。

4. 当前页面操作人必须是主账号或者是具有AliyunSTSAssumeRoleAccess权限的账号。

5. 开通TDE大约10分钟完成。

6. 目前支持的密钥类型为 Aliyun_AES_256, Aliyun_SM4。

确定

取消

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[创建密钥](#)。
- 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。
 - 需要使用阿里云账号或者具有AliyunSTSAssumeRoleAccess权限的账号。

开通TDE大约需要10分钟时间，请您耐心等待。

高级选项

说明 仅当集群版本为PolarDB MySQL引擎8.0版本且内核小版本为8.0.1.1.15及以上时，支持开启高级选项功能。

开通TDE时，您可以在设置TDE对话框中，开启高级选项功能。该功能开启后，所有新建的表将自动加密。

设置TDE

☒ 使用由阿里云自动生成的密钥

☐ 使用已有自定义密钥

高级选项:

☒ 开启后，所有新建的表自动加密

ⓘ 开通注意事项

1. 开启TDE会重启PolarDB实例。

2. 禁用密钥，设置了密钥计划删除，或者删除了密钥材料都会造成密钥不可用。

3. 撤销授权关系后，重启以后会造成PolarDB实例不可用。

4. 当前页面操作人必须是主账号或者是具有AliyunSTSAssumeRoleAccess权限的账号。

5. 开通TDE大约10分钟完成。

6. 目前支持的密钥类型为 Aliyun_AES_256, Aliyun_SM4。

确定

取消

加密和解密操作

❓ 说明 如果打开了高级选项功能，所有新建的表将自动加密，无须手动进行加密操作。对于现存的表，用户还需要进行以下操作才能进行数据加密。

开启TDE之后，您还需要登录数据库对MySQL的表进行DDL操作才能进行数据加密或者解密，不同内核版本的加密和解密操作存在差异，具体如下：

操作类型	PolarDB MySQL 5.6	PolarDB MySQL 5.7&PolarDB MySQL 8.0
加密	<pre>alter table <tablename> block_format=encrypted;</pre>	<pre>alter table <tablename> encryption= 'Y';</pre>
解密	<pre>alter table <tablename> block_format=default;</pre>	<pre>alter table <tablename> encryption= 'N';</pre>

❓ 说明 通过以上alter table命令对表进行加密或解密时，该表会处于锁定状态。

6.4. SQL防火墙

6.4.1. 概述

PolarDB的Proxy提供了SQL防火墙功能，该功能通过设置黑白名单规则来识别需要放行和拦截的SQL语句。本文主要介绍SQL防火墙功能相关内容。

背景信息

在实际应用场景中，如果数据库被攻击者撞库成功，攻击者可能会通过批量拖库的方式获取所有信息。除此之外，在日常运维操作中，如果执行的SQL语句中缺少 `where` 条件，且执行的语句为DELETE语句时，会发生误删除数据库中的表的意外事件。

为了在意外发生之前拦截这类高危操作，Proxy提供了一种SQL拦截能力，通过设置黑名单规则和白名单规则两种规则类型来识别需要拦截和放行的SQL语句。

前提条件

PolarDB数据库代理版本需为2.5.1或以上。如何查看和升级当前数据库代理版本，请参见[版本管理](#)。

 **说明** 数据库代理版本为2.5.1以下的版本时，如果需要使用此功能，请[提交工单](#)联系技术支持升级数据库代理版本。

使用限制

- 当前功能只对新购买的集群生效，存量集群暂时不可用。
- 开启SQL防火墙功能后，Proxy会在您当前的集群创建一个数据库 `database:proxy_auditing` 和两张表，其中，`sql_list` 表用于保存黑白名单规则中的参数化SQL语句，`org_sql_list` 表用于保存黑名单规则中具体的SQL语句。且每张表最多保存500,000条SQL语句。
- 第一次创建新规则时，只对新建立的连接生效。修改规则后，5s后可以对之前的连接生效。
- 使用指定模式创建的黑白名单SQL语句，在创建时不会被拦截，创建成功后才会进行拦截。
- 暂不支持对同一账号和Endpoint同时启用黑名单和白名单两个功能。
- 不支持Multi-Statement。所有的Multi-Statement都会被拦截。
- Proxy无法解析的SQL语句，会被记录到防火墙审计日志中，但不进行拦截。例如，含有错误语法的SQL语句SELECT。
- 使用指定模式创建的黑白名单SQL语句，即使对应的黑白名单规则都被禁用或被删除，其在数据库表中的SQL语句还是会被保留。如果对应的黑白名单再次被启用，同账号的规则还是会生效。如果您想彻底删除黑白名单，可通过超级账号连接到对应集群的主节点上删除对应的SQL语句。
- 黑白名单功能只能拦截普通sql command(3)，或者prepare command(22)；其它command不会被拦截。例如，通过MySQL命令行连接集群后，`use db;`这条SQL语句会被命令行工具转为COM_INIT_DB(1)命令，该命令不会被拦截。

性能影响

- 启用黑名单规则后，会产生一定的性能开销（10%以内）。
- 启用白名单规则后，且不产生报警日志的情况下，会产生一定的性能开销（10%左右）。
- 如果执行的所有SQL语句全部产生报警日志，在正常的业务情况下，请求速率会下降20%~30%。但是，正常情况下不会发生所有SQL均触发报警日志的情况，而是偶尔会有SQL语句触发报警日志。所以在正常的业务情况下，开启黑白名单功能对客户的请求速率都不会有明显的影。

功能简介

- **黑名单规则**：您可以通过配置黑名单规则的方式，来拦截指定类型或具体的SQL语句，当执行符合黑名单规则的SQL语句时，Proxy会对符合规则的SQL语句进行拦截。具体请参见[设置黑名单规则](#)。

- **白名单规则**：该功能的核心思想是使用业务SQL训练得出SQL白名单列表。开启防护模式之后，只有白名单中的SQL可以得到放行。具体请参见[设置白名单规则](#)。

6.4.2. 设置黑名单规则

您可以在PolarDB控制台新增、启用或禁用黑名单规则，同时您还可以修改或删除已有的规则。本文将介绍黑名单规则的概念以及相关的操作步骤。

黑名单规则

Proxy提供的一种拦截能力，您可以通过配置黑名单规则来拦截指定类型的SQL语句或具体的SQL语句。

当前支持通过以下三种方式配置黑名单规则：

- **固定规则模式**：在控制台设置的常见的黑名单规则模式，每一种规则可以设定对某个账号以及某个集群生效。当前支持的常见规则请参见[配置SQL防火墙规则](#)。
- **用户自定义SQL参数化模式**：该模式支持将您在数据库中执行的SQL语句中的所有变量进行参数化处理，并生成参数化模板记录到数据库中。当执行符合参数化模板的SQL语句时，Proxy会进行拦截。
- **用户自定义具体SQL模式**：该模式支持对您指定的具体的SQL语句进行拦截，而不将SQL语句中的变量进行参数化处理，其它任意参数的SQL语句不受该模式影响。

新增黑名单规则

1. 登录[PolarDB控制台](#)。
2. 在控制台左上角，选择集群所在地域。
3. 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SQL防火墙页签，单击页面左侧的**新增**。
6. 在**新建SQL防火墙规则**对话框中，根据您需要新建的黑名单模式，设置对应的参数信息。
 - **固定规则模式**。
 - a. 选择**新建固定规则模式**时，设置以下参数。配置SQL防火墙规则

参数名称		是否必填	描述
基本信息	规则名称	是	SQL防火墙规则的名称。规则名称需满足如下要求： ■ 使用数字和英文字母组成。 ■ 字符长度不能超过30个字符。
	规则描述	否	描述该规则的相关信息，便于后续管理。  说明 字符长度不能超过64个字符。
	Endpoint	是	需要应用当前规则的Endpoint。
	规则类型	是	需要新增的规则类型。选择 黑名单规则 。
	当前模式	否	SQL防火墙规则的模式。固定为 防护模式 。即检测到符合黑名单中的SQL语句时，进行拦截。

参数名称		是否必填	描述
	数据库账号名	否	需要应用当前规则的数据库账号名称。支持以下选项： ■ 所有数据库账号：表示当前规则对该集群中的所有数据库账号生效，此时右侧文本框无需填写任何内容。 ■ 包含：表示当前规则仅对目标数据库账号生效，此时右侧文本框中必须填入一个或多个数据库账号名，多个账号之间使用英文逗号（,）分隔。 ■ 不包含：表示当前规则会对该集群中，除目标数据库账号外的所有数据库账号生效，此时右侧文本框中必须填入一个或多个数据库账号名，多个账号之间使用英文逗号（,）分隔。  说明 文本框中填入的数据库账号名称需满足如下格式中的任意一种： ■ 账号名。示例：<code>user</code>。 ■ 账号名@完整IP地址。示例：<code>user@10.0.0.0</code>。
	拦截*号	否	是否拦截含有 <code>*</code> 号的SQL语句。取值： ■ 开启：拦截含有 <code>*</code> 号的SQL语句。 ■ 关闭：不拦截含有 <code>*</code> 号的SQL语句。
	拦截特定类型	否	是否拦截特定类型的SQL语句。取值： ■ 开启：拦截特定类型的SQL语句。如果选择开启拦截特定类型，则至少选择其中一种类型。目前支持的类型有： ■ <code>CREATE</code> ■ <code>DROP</code> ■ <code>ALTER</code> ■ <code>TRUNCATE</code> ■ <code>RENAME</code> ■ <code>INSERT</code> ■ <code>UPDATE</code> ■ <code>SELECT</code> ■ <code>DELETE</code> ■ 关闭：不拦截特定类型的SQL语句。

参数名称		是否必填	描述
规则配置	拦截无where条件	否	是否拦截无where条件的SQL语句。取值： ■ 开启：拦截指定类型且无where条件的SQL语句。如果选择开启拦截无where条件，则至少选择其中一种类型。目前支持的类型有： ■ UPDATE ■ SELECT ■ DELETE ■ 关闭：不拦截无where条件的SQL语句。  说明 目前只针对 SELECT 、 UPDATE 和 DELETE 语句且语句中至少包含一个表名时生效。 SELECT 1； 这类语句不会被Proxy拦截。
	拦截特定列名	否	是否拦截含有特定列名的SQL语句。取值： ■ 开启：拦截含有特定列名的SQL语句。如果选择开启拦截特定列名，支持以下选项： ■ 所有：表示当前规则对该集群中的所有数据库列名生效，此时右侧文本框无需填写任何内容。 ■ 包含：表示当前规则仅对该集群中指定的列名生效，此时右侧文本框中必须填入一个或多个列名，多个列名之间使用英文逗号（,）分隔。 ■ 不包含：表示当前规则会对该集群中，除指定的列名外的所有数据库列名生效，此时右侧文本框中必须填入一个或多个列名，多个列名之间使用英文逗号（,）分隔。 ■ 关闭：不拦截含有特定列名的SQL语句。

参数名称		是否必填	描述
	拦截特定函数	否	是否拦截特定函数。取值： ■ 开启：拦截含有特定函数的SQL语句。如果选择开启拦截特定函数，支持以下选项： ■ 所有：表示当前规则对在该集群的数据库中执行的所有函数生效，此时右侧文本框无需填写任何内容。 ■ 包含：表示当前规则仅对在该集群的数据库中执行的特定的函数生效，此时右侧文本框中必须填入一个或多个函数名，多个函数名之间使用英文逗号（,）分隔。 ■ 不包含：表示当前规则会对在该集群的数据库中执行的，除指定的函数名外的所有函数生效，此时右侧文本框中必须填入一个或多个函数名，多个函数名之间使用英文逗号（,）分隔。 ■ 关闭：不拦截含有特定函数的SQL语句。

参数名称		是否必填	描述
	拦截特定函数引用特定列名	否	是否拦截含有特定函数且引用特定列名的SQL语句。取值： ■ 开启：拦截含有特定函数且引用特定列名的SQL语句。如果选择开启拦截特定函数引用特定列名，右侧文本框中必须填入一个或多个函数名和列名。其中： ■ 如果函数名选择包含，列名也选择包含时。表示该规则对该集群中使用特定函数，且引用特定列名的SQL语句生效。 ■ 如果函数名选择包含，列名选择不包含。表示该规则对该集群中使用特定函数，且引用指定列名之外的SQL语句生效。 ■ 如果函数名选择不包含，列名选择包含。表示该规则对该集群中使用特定函数之外的函数，且引用指定列名的SQL语句生效。 ■ 如果函数名选择不包含，列名也选择不包含。表示该规则对该集群中使用特定函数之外的函数，且引用指定列名之外的列名的SQL语句生效。 ■ 关闭：不拦截含有特定函数且引用特定列名的SQL语句。

- b. 规则设置完成后，单击**确认**。
- o. 用户自定义SQL参数化模式。

a. 选择新建**用户自定义SQL参数化模式**时，设置您需要的参数信息，具体的参数信息请参见[配置SQL防火墙规则](#)。

?

说明

当您使用**用户自定义SQL参数化模式**时，可以不开启规则配置中的任何拦截规则。

- b. 规则设置完成后，单击**确认**。

- c. 使用上述步骤中的数据库账号名连接指定的数据库Endpoint地址。通过在指定的SQL语句前添加 `HINT` 命令 `hint(/* store_to_blacklist */)` 来指定要拦截的SQL语句。例如，拦截SQL语句 `select id from sqlblack_test where id = 1;`，执行如下命令：

```
/* store_to_blacklist */ select id from sqlblack_test where id = 1;
```

参数化后的模板如下：

```
select id from sqlblack_test where id = ?
```

其中，`?` 表示任意值。

等待5s，当使用该账号在该指定的集群中执行符合以上参数化模板的SQL语句时，会被Proxy进行拦截。拦截后显示的信息如下：

```
ERROR 1141 (HY000): This SQL is rejected by SQL Firewall. Access denied for user 'xxx'@'x.x.x.x' to database 'xzh': This SQL is in blacklist bl_test.
```

其中，`bl_test` 为黑名单规则表。

② 说明

- 如果使用MySQL命令行，需要加 `-c` 选项，否则 `HINT` 命令不生效。
- 被参数化后的SQL语句，需要等待5s才会生效。

o. 用户自定义具体SQL模式。

- a. 选择新建用户自定义具体SQL模式时，设置您需要的参数信息，具体的参数信息请参见[配置SQL防火墙规则](#)。

② 说明 当您使用用户自定义具体SQL模式时，可以不开启规则配置中的任何拦截规则。

- b. 规则设置完成后，单击**确认**。

- c. 使用上述步骤中的数据库账号名连接指定的数据库Endpoint地址。通过在指定的SQL语句前添加 `HINT` 命令 `hint(/* original_store_to_blacklist */)` 来指定要拦截的SQL语句。例如，拦截SQL语句 `update t set k = 2 where id = 2;`。

```
/* original_store_to_blacklist */ update t set k = 2 where id = 2;
```

等待5s，当使用该账号在该指定的集群中执行该具体的SQL语句 `update t set k = 2 where id = 2;` 时，会被Proxy进行拦截，而其他参数的SQL语句不受影响。拦截后显示的信息如下：

```
ERROR 1141 (HY000): This SQL is rejected by SQL Firewall. Access denied for user 'xxx'@'x.x.x.x' to database 'xzh': This SQL is in blacklist bl_test.
```

其中，`bl_test` 为黑名单规则表。

② 说明

- 如果使用MySQL命令行，需要加 `-c` 选项，否则 `HINT` 命令不生效。
- 被参数化后的SQL语句，需要等待5s才会生效。

启用或禁用黑名单规则

1. 登录PolarDB控制台。
2. 在控制台左上角，选择集群所在地域。
3. 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SQL防火墙页签，找到需要启用或禁用的规则，打开启/禁规则开关。

规则名称	描述	启/禁规则	操作
			修改 删除

 **说明** 您也可以在规则列表中同时选中多个目标规则，然后单击列表下方的启用或禁用按钮，批量启用或禁用规则。

6. 在弹出的启用或禁用对话框中，单击确定。

 **说明** 禁用指定模式创建的黑名单规则时，即使对应的黑名单规则被禁用，其对应数据库中的黑名单表中的SQL语句还是会被保留。如果对应的黑名单再次被启用，同账号的规则还是会生效。如果您想彻底禁用黑名单，可通过超级账号连接到对应集群的主节点，删除 `proxy_auditing.sql_list` 表中的SQL语句。删除SQL语句后，需要等待5s才会生效。且在进行删除操作时，请勿直接使用 `DROP` 命令删除表。

修改SQL黑名单规则

1. 登录PolarDB控制台。
2. 在控制台左上角，选择集群所在地域。
3. 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SQL防火墙页签，找到需要修改的规则，单击操作列的修改，在弹出的修改SQL防火墙规则对话框中，根据业务需要修改规则参数。规则参数的具体说明，请参见配置SQL防火墙规则。

规则名称	描述	启/禁规则	操作
			修改 删除

 **说明** 修改规则时，不支持修改规则名称。

6. 规则修改完成后，单击确认。

 **说明** 用户自定义SQL参数化模式规则中的参数化SQL语句，以及用户自定义具体SQL模式规则中的具体SQL语句不支持在控制台修改，只能在数据库表中进行删除后重新添加。

删除黑名单规则

1. 登录PolarDB控制台。
2. 在控制台左上角，选择集群所在地域。
3. 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SQL防火墙页签，找到需要删除的规则，单击操作列的删除。

规则名称	描述	启/禁规则	操作
			修改 删除

说明 您也可以在规则列表中同时选中多个目标规则，然后单击列表下方的删除，批量删除规则。

6. 在弹出的删除对话框中，单击**确定**。

说明 删除指定模式创建的黑名单规则时，即使对应的黑名单规则被删除，其对应数据库中 `proxy_auditing.sql_list` 表中的SQL语句还是会被保留。如果您想彻底删除黑名单，可通过超级账号连接到对应集群的主节点，删除 `proxy_auditing.sql_list` 表中的SQL语句。删除SQL语句后，需要等待5s才会生效。且在进行删除操作时，请勿直接使用 `DROP` 命令删除表。

取消指定模式黑名单规则

- 用户自定义SQL参数化模式黑名单规则。

您可以通过以下两种方式取消用户自定义SQL参数化模式黑名单规则。

- 通过**启用或禁用黑名单规则**或**删除黑名单规则**取消用户自定义SQL参数化模式黑名单规则。

说明 如果仅在控制台禁用该模式的黑名单规则，而没有在数据库中 `proxy_auditing.sql_list` 表里删除对应的参数化SQL语句，则在控制台再次启用对应账号的该黑名单规则时，该规则仍然会生效。

- 通过您的超级账号连接到Endpoint主节点，删除 `proxy_auditing.sql_list` 表中相应的SQL语句。等待5s，该参数化SQL语句将不再被拦截。

说明 在进行删除操作时，请勿直接使用 `DROP` 命令删除表。

- 用户自定义具体SQL模式黑名单规则。

您可以通过以下两种方式取消用户自定义具体SQL模式黑名单规则。

- 通过**启用或禁用黑名单规则**或**删除黑名单规则**取消用户自定义具体SQL模式黑名单规则。

说明 如果仅在控制台禁用该模式的黑名单规则，而没有在数据库中 `proxy_auditing.org_sql_list` 表里删除对应的SQL语句，则在控制台再次启用对应账号的该黑名单规则时，该规则仍然会生效。

- 通过您的超级账号连接到Endpoint主节点，删除 `proxy_auditing.org_sql_list` 表中相应的SQL语句。等待5s，该SQL语句将不再被拦截。

说明 在进行删除操作时，请勿直接使用 `DROP` 命令删除表。

6.4.3. 设置白名单规则

您可以在PolarDB控制台新增、启用或禁用白名单规则，同时您还可以修改或删除已有的规则。本文将介绍白名单规则的概念以及相关的操作步骤。

白名单规则

设置白名单规则后，任何不包含在白名单中的SQL语句都会被拦截或报警。此模式用于保护您的主业务使用的账号。即只允许此账号执行业务中使用的SQL语句，不允许执行任何业务外的SQL语句。由于实际业务中使用的SQL语句可能会很多，单个SQL语句录入会相当耗时，为了提升工作效率和使用体验。Proxy提供了以下三种白名单模式：

- **训练模式**：Proxy只收集SQL语句，而不进行SQL语句拦截和报警。
- **检验模式**：检测到不包含在白名单中的SQL语句时，只记录不进行SQL拦截。
- **防护模式**：检测到不包含在白名单中的SQL语句时，进行SQL拦截并记录。

您也可以在控制台设置多个白名单，每个白名单可以通过不同的账号去训练。开启**检验模式**和**防护模式**后，也可以单独为每个白名单指定执行账号。

新增白名单规则

1. 登录**PolarDB控制台**。
2. 在控制台左上角，选择集群所在地域。
3. 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**SQL防火墙**页签，单击页面左侧的**新增**。
6. 在**新建SQL防火墙规则**对话框中，根据您需要新建的白名单模式，设置对应的参数信息。

配置SQL防火墙规则

参数名称		是否必填	描述
基本信息	规则名称	是	SQL防火墙规则的名称。规则名称需满足以下条件： ◦ 使用数字和英文字母组成。 ◦ 字符长度不能超过30个字符。
	规则描述	否	描述该规则的相关信息，便于后续管理。 ⓘ 说明 字符长度不能超过64个字符。
	Endpoint	是	需要应用当前规则的Endpoint。
	规则类型	是	需要新增的规则类型。选择 白名单规则 。
	当前模式	否	SQL防火墙规则的模式。取值： ◦ 训练模式：只收集SQL语句，不进行记录和拦截。 ◦ 检验模式：检测到白名单外的SQL语句时，只记录不进行拦截。 ◦ 防护模式：检测到白名单外的SQL语句时，进行记录并拦截。

参数名称		是否必填	描述
规则配置	数据库账号名	否	需要应用当前规则的数据库账号名称。支持以下选项： - 所有数据库账号：表示当前规则对该集群中的所有数据库账号生效，此时右侧文本框无需填写任何内容。 - 包含：表示当前规则仅对目标数据库账号生效，此时右侧文本框中必须填入一个或多个数据库账号名，多个账号之间使用英文逗号（,）分隔。 - 不包含：表示当前规则会对该集群中，除目标数据库账号外的所有数据库账号生效，此时右侧文本框中必须填入一个或多个数据库账号名，多个账号之间使用英文逗号（,）分隔。  说明 文本框中填入的数据库账号名称需满足如下格式中的任意一种： - 账号名。示例：user。 - 账号名@完整IP地址。示例：user@10.0.0.0。

7. 规则设置完成后，单击**确认**。
8. 根据您选择的模式，选择执行的操作步骤。
- 如果当前模式为**训练模式**，请执行以下步骤。

a. 使用上述步骤中的**数据库账号名**连接指定的数据库Endpoint地址。

b. 使用该账号执行业务中需要添加到白名单中的SQL语句，Proxy会将SQL语句进行参数化处理，并保存到数据库中的白名单中。例如，

```
update t set k = 2 where id = 2;
```

参数化后的SQL语句如下：

```
update t set k = ? where id = ?
```

其中，**?** 表示任意值。该参数化后的SQL语句 `update t set k = ? where id = ?` 会被收集到白名单中。

 说明 您也可以在任何白名单模式下，在业务SQL语句前添加 `HINT 命令` `hint (/* store_to_whitelist */)`，将业务SQL语句进行参数化处理并添加到白名单中。

- 如果当前模式为**检验模式**，请执行以下步骤。
- a. 使用上述步骤中的**数据库账号名**连接指定的数据库Endpoint地址。

- b. 使用该账号执行业务中使用的SQL语句，检测是否还存在没有被收集到白名单中的SQL语句。例如：

```
update t set k = 2 where k = 2
```

如果该SQL语句不包含在白名单中，Proxy会将该SQL语句放行并记录。且执行结果如下：

```
Query OK, 0 rows affected (0.03 sec)
Rows matched:1 Changed: 0 Warnings:0
```

- o 如果当前模式为防护模式，请执行以下步骤。
 - a. 使用上述步骤中的数据库账号名连接指定的数据库Endpoint地址。
 - b. 使用该账号执行业务中使用的SQL语句。例如，

```
select id from t where id = 1;
```

如果该SQL语句没有包含在白名单中，Proxy会进行记录和拦截，且执行结果如下：

```
ERROR 1141 (HY000): This SQL is rejected by SQL Firewall. Access denied for user
'xzh'@'x.x.x.x' to database 'xzh': This SQL is not in whitelist wl_test.
```

说明

- 每次更新业务时，都需要进行一次SQL语句训练，否则，新业务的SQL语句将不能够被执行。
- 您可以在控制台新增多个白名单，每个白名单可以通过不同的账号去训练。启用检验模式和防护模式后，也可以单独为每个白名单指定要对哪些账号起作用。

启用或禁用白名单规则

1. 登录PolarDB控制台。
2. 在控制台左上角，选择集群所在地域。
3. 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SQL防火墙页签，找到需要启用或禁用的规则，打开启/禁规则开关。

规则名称	描述	启/禁规则	操作
			修改 删除

说明 您也可以同时在规则列表中同时选中多个目标规则，然后单击列表下方的启用或禁用按钮，批量启用或禁用规则。

6. 在弹出的启用或禁用对话框中，单击确定。

说明 禁用白名单规则时，即使对应的白名单规则被禁用，其对应数据库中的白名单表中的SQL语句还是会被保留。如果对应的白名单再次被启用，同账号的规则还是会生效。如果您想彻底禁用白名单，可通过超级账号连接到对应集群的主节点，删除 `proxy_auditing.sql_list` 表中的SQL语句。删除SQL语句后，需要等待5s才会生效。且在进行删除操作时，请勿直接使用 `DROP` 命令删除表。

修改白名单规则

1. 登录PolarDB控制台。

2. 在控制台左上角，选择集群所在地域。
3. 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SQL防火墙页签，找到需要修改的规则，单击操作列的修改，在弹出的修改SQL防火墙规则对话框中，根据业务需要修改规则参数。规则参数的具体说明，请参见配置SQL防火墙规则。

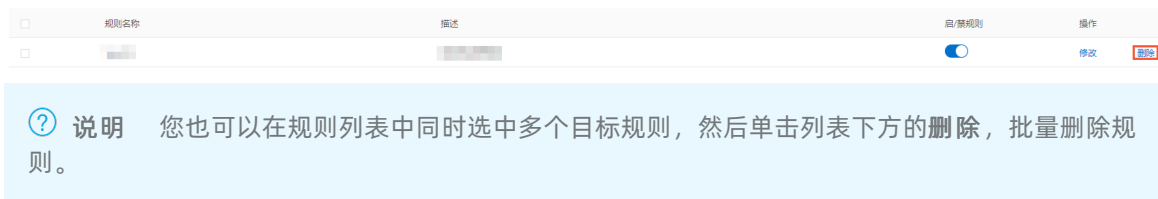


6. 规则修改完成后，单击确认。

说明 数据库中被参数化后的SQL语句不支持在控制台修改，只能删除数据库表 `proxy_auditing.sql_list` 中的参数化SQL语句后重新添加。

删除白名单规则

1. 登录PolarDB控制台。
2. 在控制台左上角，选择集群所在地域。
3. 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SQL防火墙页签，找到需要删除的规则，单击操作列的删除。



6. 在弹出的删除对话框中，单击确定。

说明 删除白名单规则时，即使对应的白名单规则被删除，其对应数据库中的 `proxy_auditing.sql_list` 表中的SQL语句还是会被保留。如果您想彻底删除白名单，可通过超级账号连接到对应集群的主节点，删除 `proxy_auditing.sql_list` 表中的SQL语句。删除SQL语句后，需要等待5s才会生效。且在进行删除操作时，请勿直接使用 `DROP` 命令删除表。

6.5. 数据安全和加密常见问题

本文档介绍了数据安全和加密的常见问题。

设置集群白名单常见问题

- Q: 如何指定服务器只能访问集群中的某个节点？
A: 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。
- Q: IP白名单最多支持填写多少个IP地址？
A: 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。
- Q: 已添加ECS的IP地址到IP白名单中，为何还是无法访问集群？
A: 请按以下步骤进行排查：

- i. 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。
- ii. 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[ClassicLink](#)打通经典网络到专有网络的网络。

- iii. 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。
- Q: 什么原因会导致公网连接失败?
A: 公网连接失败的原因如下所示：
 - i. 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。
 - ii. 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[申请集群地址和主地址](#)确定真正的公网地址。
 - Q: 如何限制某个用户只能从特定的IP地址访问PolarDB?
A: 您可以使用以下命令创建高权限账号，并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';|
```

设置SSL加密常见问题

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。

设置透明数据加密TDE常见问题

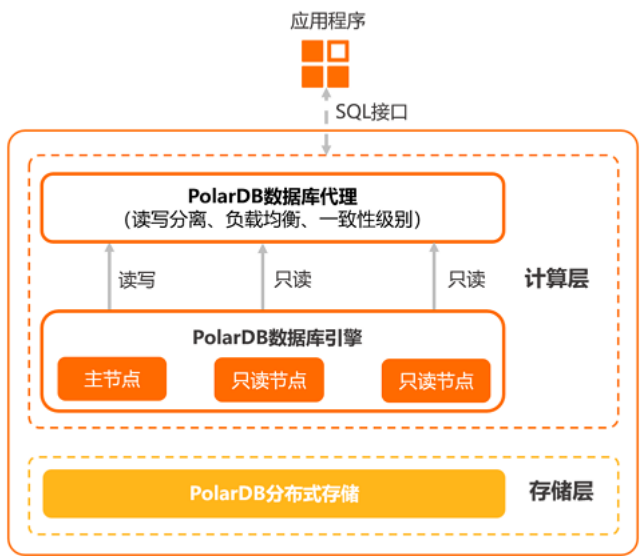
- Q: 开启TDE后，常用数据库工具（Navicat等）还能正常使用吗？
A: 可以正常使用。
- Q: 加密后查看数据为什么还以明文显示？
A: 查询数据时会解密并读取到内存，所以是明文显示。开启TDE后存储的数据是加密的。

7.数据库代理企业版

7.1. 概述

PolarDB数据库代理是位于数据库和应用程序之间的网络代理服务，用于代理应用程序访问数据库时的所有请求，具有高可用、高性能、可运维、简单易用等特点，支持自动读写分离、负载均衡、一致性级别、连接池等高级功能。您可以连接PolarDB集群地址使用数据库代理的各项功能。

PolarDB数据库代理介绍



PolarDB集群版是一个由多节点构成的数据库集群，包括一个主节点和多个只读节点。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PolarDB数据库代理提供，集群地址分为可读可写和只读两种读写模式，可读可写模式支持读写分离，只读模式支持按连接数负载。

支持两种数据库代理版本

数据库代理企业版提供两种版本：企业通用版和企业独享版。

- **企业通用版**：配套集群子系列的通用规格，它可以共享CPU物理资源，可根据业务负载，提供智能秒级资源弹性扩展能力。
- **企业独享版**：配套集群子系列的独享规格，它可以独占CPU物理资源，具有更好的性能稳定性。

两种版本的差异如下表所示：

对比项	企业通用版	企业独享版
计费类型	付费（约占计算节点总费用的5~10%） 当前暂时免费使用，未来收费时间待定	
资源类型	共享CPU物理资源，可根据业务负载，提供智能秒级资源弹性扩展能力	独占物理资源，具有更好的性能稳定性
部署架构	高可用冗余架构	
实例规格	最低配置：2核	

对比项	企业通用版	企业独享版
性能	集群存储最大IOPS相比历史版本可提升50%，不同规格集群的最大IOPS可参见 计算节点规格	
只读节点配置	只读节点无需与主节点配置保持一致，可以根据业务负载降配从而节省成本	
只读节点数量	支持1~15个只读节点	
连接地址	1个主地址+7个集群地址	
热备切换	连接/事务不中断，短暂阻塞5~10秒	
一致性	- 最终一致性 - 会话一致性 - 全局一致性	
连接池	支持	
事务拆分	支持	
防闪断（连接保持）	支持	
数据脱敏（安全）	支持	
变配业务无感	支持	
多主架构	支持	
计算节点秒级弹性扩展	支持	独占资源，无需支持
Proxy限流保护（即将上线）	支持	

费用

数据库代理企业版目前可免费使用，未来将收费，收费时间待定。

数据库代理企业版的费用策略如下：

系列	购买类型	费用策略
单节点和历史库单节点版	不论是新购还是存量，都不支持数据库代理功能。	
集群版和历史库集群版	新购集群	在2021年12月9日后，新购集群仅提供数据库代理企业版。
	存量按量付费集群	对于存量的按量付费集群，数据库代理于2021年12月9日已自动切换到数据库代理企业版。
	存量包年包月集群	对于存量的包年包月集群，数据库代理于2021年12月9日已自动切换到数据库代理企业版。

使用限制

仅PolarDB集群版和历史库集群版支持集群地址和数据库代理相关功能。PolarDB单节点和历史库单节点版采用单节点架构无需集群地址和数据库代理功能。

注意事项

- 数据库代理默认的集群地址和自定义的集群地址都不支持压缩协议。
- 使用集群地址后，若未启用事务拆分，事务内的所有请求都会路由到主节点。
- 使用集群地址后，当执行 `SHOW PROCESSLIST` 命令时，系统会将所有节点的结果合并后返回。
- 若执行了Multi Statements或调用了存储过程，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功能。
- PolarDB集群地址本身并没有最大连接数的限制，连接数的限制主要由后端数据库中计算节点的规格决定。对于读写模式的集群地址由于一个应用连接会与后端数据库中每个计算节点各建立一个连接，所以应用能使用的最大连接数就是单个计算节点的最大连接数；对于只读模式的集群地址由于一个应用连接只会与后端数据库中的一个计算节点建立连接，所以应用能够使用的最大连接数为该只读集群地址下所有只读节点的最大连接数之和。对于读写模式的集群地址可以通过[事务级连接池](#)功能来增加应用的最大可用连接。
- 新增只读节点或重启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。新增只读节点或重启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或重启后的只读节点上，需要断开该连接并重新建立连接，例如，重启应用程序。
- 请勿在运行Multi Statements或调用存储过程时修改环境变量，例如 `set names utf8mb4;select * from t1;`，否则可能会导致请求路由到只读节点和主节点查到的数据不一致。

读写分离

PolarDB集群版自带读写分离功能。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，详情请参见[读写分离](#)。

负载均衡

PolarDB支持基于负载的自动调度策略，读请求将在多个只读节点中按照活跃连接数自动调度，来保证多个只读节点间的负载均衡。负载均衡包括主库是否接受读和事务拆分功能：

- **主库是否接受读**
在确保一致性及事务正确性前提下，将查询SQL仅发送到只读节点，来降低主节点的负载，确保主节点稳定。详情请参见[主库是否接受读](#)。
- **事务拆分**
PolarDB提供了事务拆分功能，旨在保证读写一致性的前提下，将事务中的读请求发送到只读节点，减轻主节点的压力。详情请参见[事务拆分](#)。

一致性级别

由于PolarDB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提供了最终一致性、会话读一致性以及全局一致性级别供您选择，详情请参见[一致性级别](#)。

连接池

PolarDB支持会话级连接池和事务级连接池，您可以根据业务场景选择合适的连接池，帮助降低因大量连接导致的数据库负载压力，详情请参见[连接池](#)。

连接保持

PolarDB新增支持连接保持功能，避免由于一些运维操作（如升级配置、主备切换或升级小版本等）或非运维操作故障（如节点所在服务器故障）导致的连接闪断或新建连接短暂失败的问题，进一步提高PolarDB的高可用性。具体请参见[连接保持](#)。

动态脱敏

当应用程序发起数据查询请求时，系统会在数据库内部对敏感数据进行变换后再返回给应用程序。开始查询前，仅需指定需要进行脱敏查询的数据库账号，以及需要脱敏的数据库、表或列的名称即可。详情请参见[动态脱敏](#)。

相关API

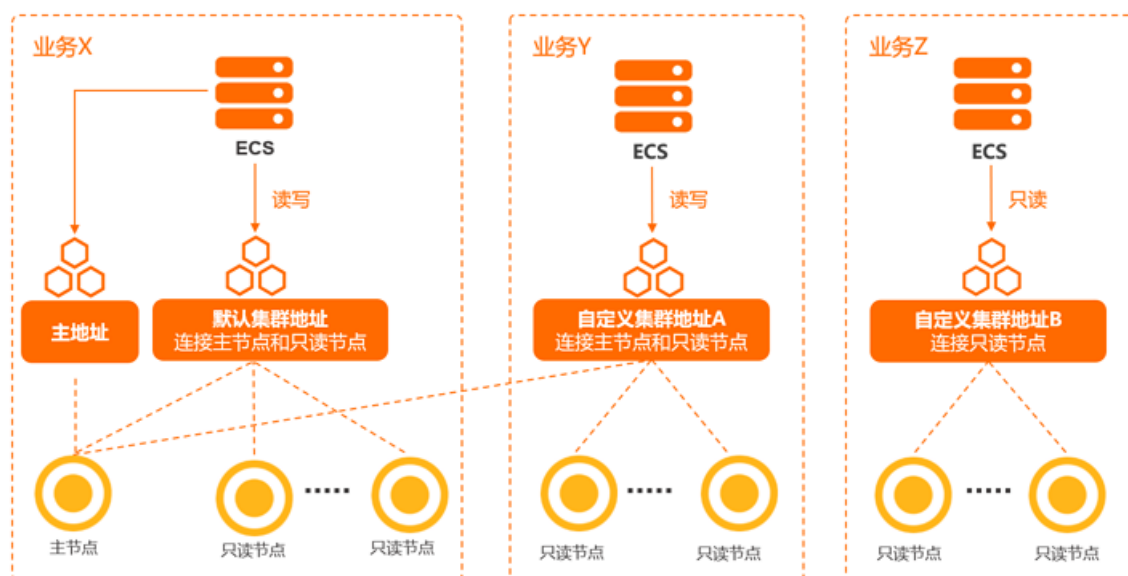
API	描述
CreateDBEndpointAddress	创建PolarDB集群的公网地址。
CreateDBClusterEndpoint	创建PolarDB自定义集群地址。
DescribeDBClusterEndpoints	查询PolarDB集群的地址信息。
ModifyDBClusterEndpoint	修改PolarDB集群地址属性。
ModifyDBEndpointAddress	修改PolarDB集群的连接地址（如自定义集群地址）。
DeleteDBEndpointAddress	释放PolarDB集群地址（除了自定义集群地址的私网地址）。
DeleteDBClusterEndpoint	释放PolarDB自定义集群地址。

7.2. 集群访问

7.2.1. 集群地址和主地址

PolarDB集群的连接地址（又称为Endpoint）分为集群地址和主地址两种类型。在连接PolarDB集群时，您可以填写PolarDB集群的集群地址或主地址。本文将分别介绍集群地址和主地址的作用和区别。

集群地址和主地址对比说明



地址类型	地址说明	适用场景	支持的网络类型
集群地址 (推荐)	集群地址通过数据库代理实现。集群地址有以下特点： • PolarDB的数据库代理支持读写分离功能。应用程序只需连接一个集群地址，即可连接到多个节点，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。 • PolarDB默认提供1个集群地址，您还可以根据业务需求创建最多6个自定义的集群地址。集群地址可以设置连接到指定的节点，以及设置读写模式等。 • 您需要连接PolarDB集群地址才能使用数据库代理的各项功能。关于数据库代理的说明，请参见数据库代理。  说明 PolarDB支持创建单节点集群地址。若此节点故障，该集群地址可能会有最多1小时的不可用，请勿用于生产环境。	• 适用于有隔离需求的业务。您可以根据业务需要将有隔离需求的业务使用对应的集群地址连接至PolarDB集群。 • 集群地址支持配置可读可写（自动读写分离）和只读2种读写模式，因此也适用于纯只读的业务。 假设购买了包含1个主节点和4个只读节点的PolarDB集群，现在需要将A业务（纯只读）和B业务（可读可写）都连接至该集群。您可以将只读节点1和只读节点2组成一个集群地址a（只读模式）提供给业务A，主节点、只读节点3和只读节点4组成一个集群地址b（可读可写（自动读写分离）模式）提供给业务B，以实现两个业务在数据库使用上的物理隔离，避免相互影响。	• 私网 • 公网 • 私网（经典网络）
主地址	每个PolarDB集群均支持且仅支持1个主地址。主地址有以下特点： • 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。	适用于不需要读写分离的业务。	

私网、公网和经典网络地址

网络类型	说明	使用场景
私网	• 通过私网的连接地址访问可以发挥PolarDB的最佳性能。 • 创建集群时会默认生成一个私网的连接地址，该地址支持修改但无法释放，关于如何修改，请参见修改连接地址。	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	• 您可以申请或释放公网的连接地址，详情请参见申请集群地址和主地址。 • 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。

网络类型	说明	使用场景
经典网络	- 您可以申请或释放私网（经典网络）的连接地址，详情请参见申请集群地址和主地址。 - 集群之间不通过网络进行隔离，只能依靠集群自身的白名单策略来阻挡非法访问，详情请参见设置白名单。	只有当业务处于经典网络中时，才能通过私网地址（经典网络）访问数据库集群。

集群地址的读写模式

集群地址支持可读可写（自动读写分离）和只读两种读写模式，下表汇总了两种模式下集群地址的差异供您参考。

 **说明** 设置集群地址读写模式的详细步骤，请参见[配置数据库代理](#)。

对比项	可读可写（自动读写分离）	只读
服务节点选择方式	支持如下3种方式： - 仅选择主节点。 - 仅选择（1个或多个）只读节点。 - 同时选择主节点和（1个或多个）只读节点。  说明 可读可写模式下： - 所有写请求只会发往主节点，不论主节点是否已被添加在服务节点中。 - 针对读请求，您可以通过设置主库是否接受读配置来决定主节点是否处理读请求。	支持如下2种方式： - 仅选择（1个或多个）只读节点。 - 同时选择主节点和（1个或多个）只读节点。  说明 只读模式下： - 所有读请求按照负载均衡的方式转发到各只读节点。 - 所有读请求不会转发到主节点。即使主节点已被添加在服务节点中，也不会生效。 - 不支持创建仅包含一个主节点的集群地址。
主库是否接受读	支持。 更多详情，请参见 读写分离 。	该功能是为了降低主节点负载，只读地址的主节点不处理任何读写请求，因此无需该功能。
事务拆分	支持。 更多详情，请参见 事务拆分 。  说明 仅当一致性级别为会话一致性（中）或全局一致性（强）时，支持该配置。	该功能是为了降低主节点负载，只读地址的主节点不处理任何读写请求，因此无需该功能。
一致性级别	支持最终一致性（弱）、会话一致性（中）和全局一致性（强）。 更多详情，请参见 一致性级别 。	最终一致性（弱），因为该地址不执行任何写操作。

对比项	可读可写（自动读写分离）	只读
连接池	支持会话级和事务级连接池。 更多详情，请参见连接池。 ② 说明 - 该功能是PolarDB数据库代理（Proxy）的连接池功能，并不影响客户端的连接池功能。如果客户端已经支持连接池，您可以选择关闭Proxy的连接池功能。 - 关闭Proxy的连接池功能后，一个客户端连接会通过Proxy连到该地址下配置的所有节点上（包括主节点和只读节点），可用的业务连接总数受主节点的最大连接数限制。	不支持。 ② 说明 PolarDB会在只读集群地址下配置的所有只读节点（主节点不参与连接分配）间均衡地分配连接，一个客户端连接会通过Proxy仅连接到一个只读节点，可用的业务连接总数是所有只读节点上的连接数之和。
并行查询	暂不支持，以免影响主节点。	支持。更多详情，请参见 并行查询（Parallel Query） 。

7.2.2. 申请集群地址和主地址

您可以通过PolarDB集群的集群地址和主地址来连接数据库集群。本文将主要介绍如何申请集群地址和主地址。

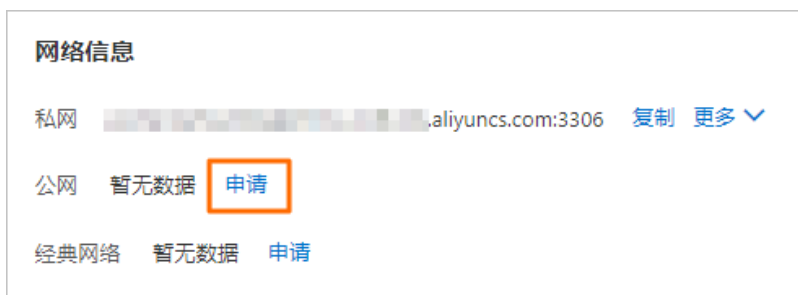
申请公网和经典网络的集群地址和主地址

1. 登录[PolarDB控制台](#)。
2. 在控制台左上角，选择集群所在地域。
3. 找到目标集群，单击集群ID。
4. 在集群基本信息页的[链接地址](#)区域，您可以通过以下任意一种方式来申请连接地址。
 - 方式一：
 - a. 单击[链接地址](#)区域右上角图标切换视图。
 - b. 单击[申请](#)。



- 方式二：
 - a. 单击目标集群地址右侧的[编辑配置](#)。

b. 在弹出的对话框中的网络信息区域，单击**申请**。



❓ 说明 仅支持申请公网和经典网络连接地址。集群创建时会默认生成一个私网连接地址，因此该地址无需申请。

5. 在弹出的对话框中，设置连接地址前缀，单击**确定**。

- ❓ 说明 连接地址前缀需满足如下条件：
- 由小写字母、数字、中划线（-）组成，6~30个字符。
 - 以小写字母开头，以数字或字母结尾。

集群地址申请成功之后，会自动支持数据库代理功能。您可以根据需要修改数据库代理的相关设置，具体请参见[配置数据库代理](#)。

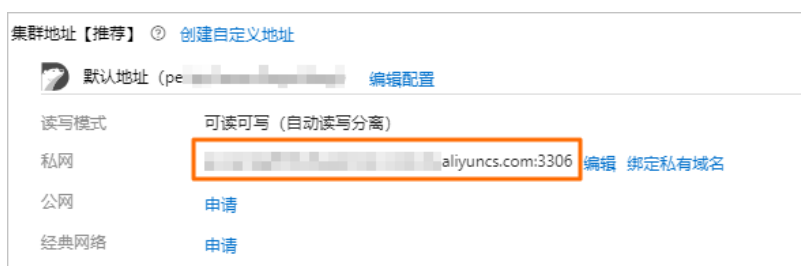
新增自定义集群地址

您也可以在集群基本信息页的**链接地址**区域，单击**创建自定义地址**，自定义一个连接地址，并配置数据库代理相关的功能。具体请参见[配置数据库代理](#)。

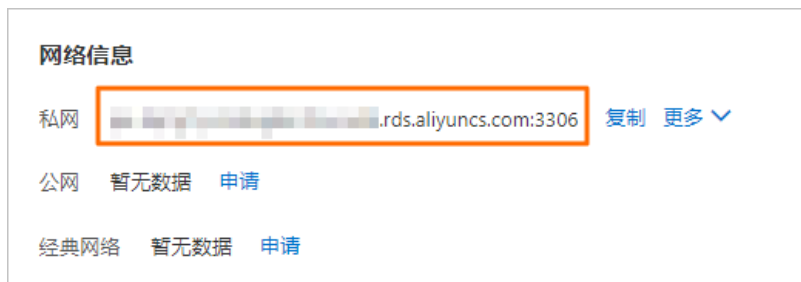
查看连接地址和端口

您可以通过以下任意一种方式来查看连接地址和端口信息：

- 单击**链接地址**区域右上角的☰图标切换视图，即可查看目标连接地址和端口信息。



- 单击目标集群地址右侧的**编辑配置**，即可在弹出的对话框中查看网络信息，包括连接地址和端口号。



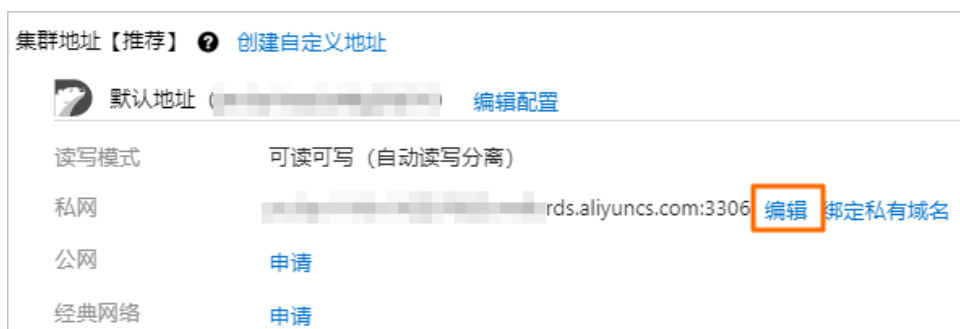
说明

- 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，可以单击绑定私有域名进行绑定。仅私网连接地址支持设置私有域名绑定，详情请参见[私有域名](#)。
- PolarDB MySQL引擎连接地址的默认端口号为3306，您也可以自定义端口号，详情请参见[修改连接地址](#)。

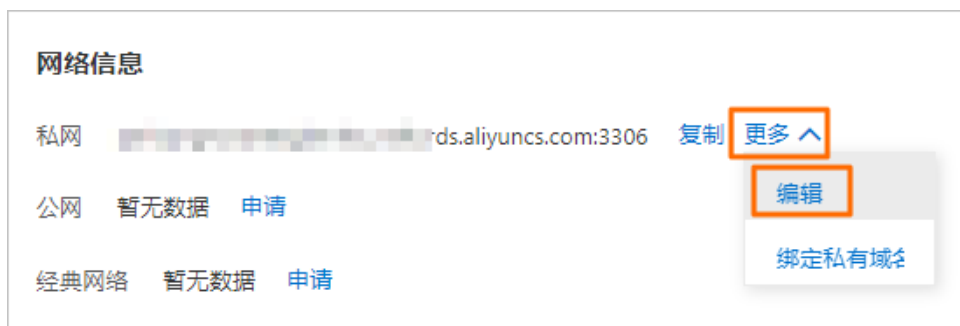
修改连接地址

您可以通过以下任意一种方式来修改连接地址和端口配置。

- 单击[连接地址](#)区域右上角的图标切换视图。然后找到目标连接地址，单击[编辑](#)。



- 单击目标集群地址右侧的[编辑配置](#)。然后在弹出的对话框中的网络信息区域，单击[更多 > 编辑](#)。



注意

- 连接地址前缀需满足如下条件：
 - 由小写字母、数字、中划线（-）组成，6~30个字符。
 - 以小写字母开头，以数字或字母结尾。
- 端口范围为3000~3500。
- 如果该连接地址正在启用SSL，修改会导致集群重启。
- 如果该连接地址正在启用SSL，则修改后的连接地址全长不能超过64个字符。

释放连接地址

 **警告**

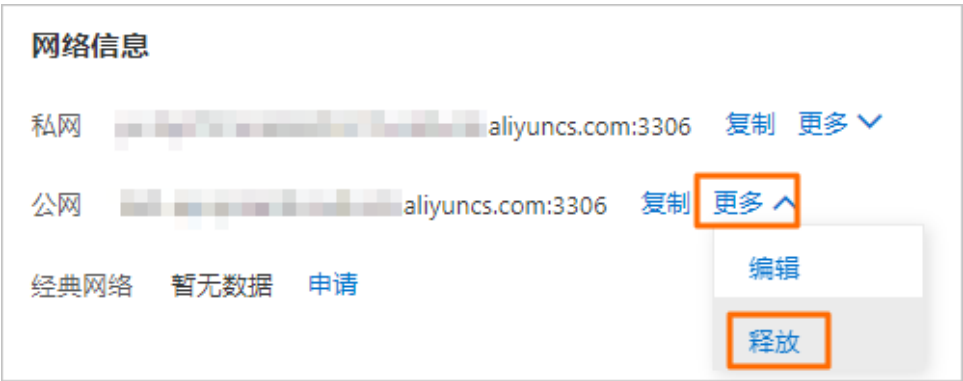
- 释放连接地址前，请确认您的应用程序已使用新的连接地址进行访问。
- 连接地址释放后无法恢复，只能重新申请，详情请参见[申请公网和经典网络的集群地址和主地址](#)。
- 默认集群地址仅支持修改地址配置，但无法被释放，自定义集群地址可以被释放。

您可以通过以下任意一种方式来释放连接地址：

- 单击[链接地址](#)区域右上角的图标切换视图。然后找到目标连接地址，单击释放。



- 单击目标集群地址右侧的[编辑配置](#)。在弹出的对话框中的网络信息区域，单击更多 > 释放。



 **说明** 仅公网和经典网络的连接地址支持被释放。

下一步

[连接数据库集群](#)

相关API

API	描述
DescribeDBClusterEndpoints	查询集群的地址信息。
CreateDBEndpointAddress	创建集群的公网地址。
ModifyDBEndpointAddress	修改集群默认访问地址。
DeleteDBEndpointAddress	释放集群地址。

7.2.3. 连接数据库集群

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

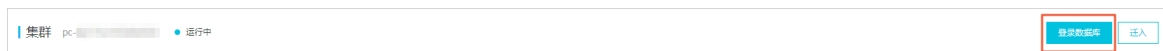
前提条件

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. 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. 在基本信息页面右上角，单击登录数据库。



5. 在弹出的对话框中，输入PolarDB MySQL集群中创建的数据库账号和数据库密码，单击登录。

登录实例

* 数据库类型

POLARDB-MySQL

✓

* 实例地区

华东1 (杭州)

✓

* 实例ID

pc-bp-xxxxxx

✓

* 数据库账号

xxxxxx

✓

* 数据库密码

xxxxxx

✓

☐ 记住密码 ?

✓ 测试连接

登录

取消

 **说明** 请确保登录的数据库账号拥有目标数据库的权限，否则您无法在DMS控制台左侧菜单栏中找到目标数据库。如何修改数据库账号权限，请参见[修改普通账号权限](#)。

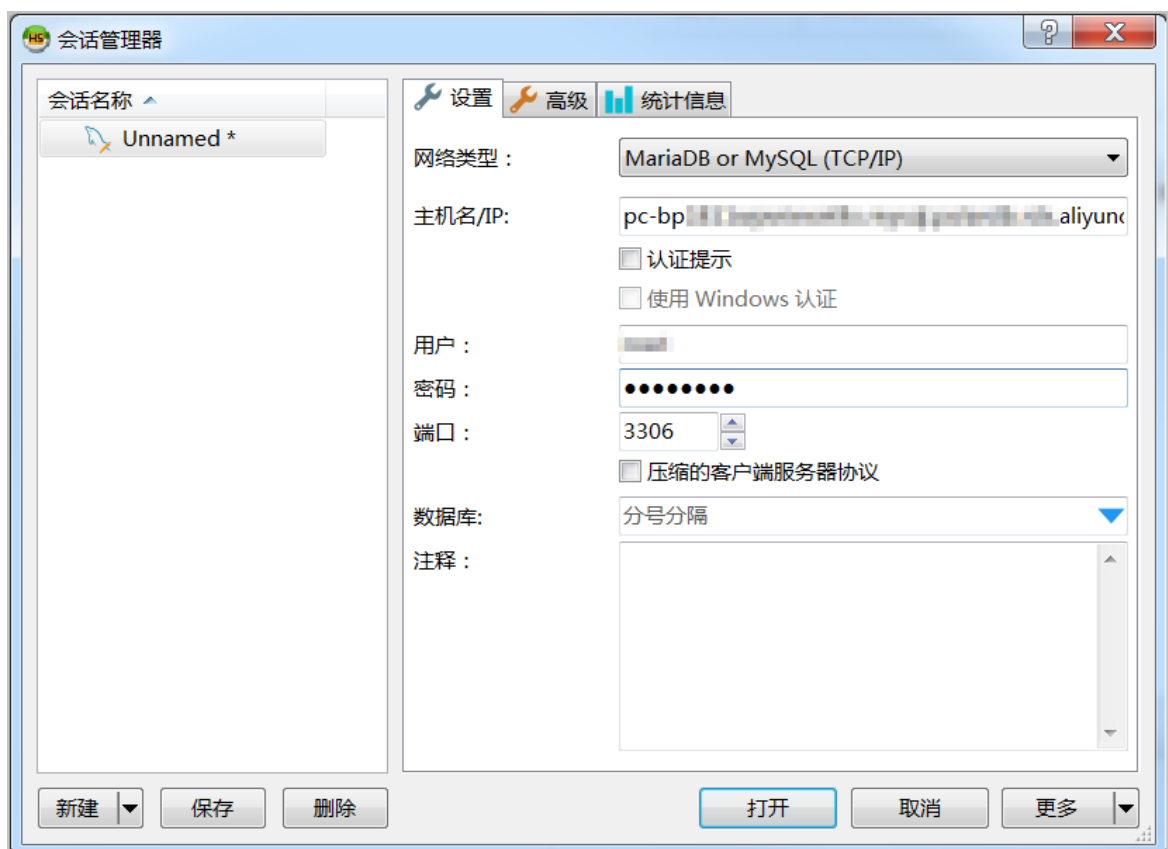
6. 登录DMS后请刷新页面，在左侧导航栏中，单击已登录实例。
7. 在已登录实例列表中，单击目标集群名称，找到并双击目标数据库名称，即可切换到目标数据库进行管理。



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。

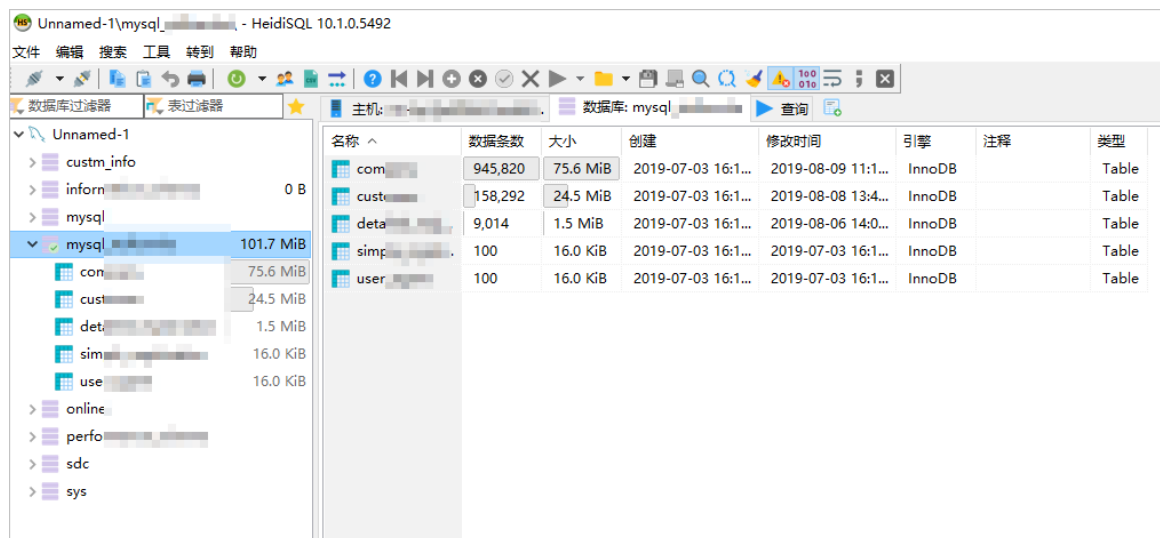
1. 启动HeidiSQL客户端。
2. 在左下角单击新建。



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（杭州）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（默认为3306）。

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引擎数据库集群，连接方式如下：

```
mysql -h<连接地址> -p<端口> -u<用户名> -p<密码>
```

示例：

```
mysql -hpc-bp1867z0z****kjd.rwlb.rds.aliyuncs.com -p3306 -uroot12 -pPass*****33
```

```
[root@izbp1g... ~]# mysql -hpc-bp...d.rwlb.rds.aliyuncs.com -p3306 -uroot12 -pPas...233
Warning: Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 536887153
Server version: 5.7.28-log Source distribution

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> 
```

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 集群地址和主地址 。	pc- bpxxxxxxxxxxxxxxxx.mysql.polar db.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集群的数据库账号名称。	root
-p	数据库账号的登录密码。 ❓ 说明 该参数为必填参数。 - 如果不填写该参数，在 Enter password 后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

7.2.4. 如何解决集群连接失败的问题

本文将介绍使用DMS（Data Management Service）和通用MySQL客户端连接PolarDB MySQL引擎集群失败的问题原因以及解决方法。

集群白名单未配置或配置有误

问题原因：

- 白名单设置中默认地址为 127.0.0.1，表示禁止任何IP地址访问PolarDB集群。白名单设置中没有添加对端的IP地址。
- 白名单设置中IP地址的格式不正确。
- 在白名单设置中添加的设备公网IP地址并非设备真正的出口IP地址。

解决方法：

- 在白名单设置中添加对端的IP地址，具体操作请参见[设置白名单](#)。
- 将白名单设置中IP地址的格式设置正确。例如将 `0.0.0.0` 修改为 `0.0.0.0/0`。
- 请获取准确的设备公网IP地址，并添加到白名单中。

用户未创建数据库账号或当前数据库账号没有访问数据库的权限

问题原因：

- 当前数据库尚未创建数据库账号。
- 当前账号没有访问该数据库的权限。

解决方法：

- 在PolarDB集群中创建数据库账号，并为该账号授予当前数据库的权限。具体操作请参见[创建数据库账号](#)。
- 在PolarDB集群中修改当前数据库账号的权限。具体操作请参见[重置高权限账号权限](#)和[修改普通账号权限](#)。

私网、公网地址使用错误

问题原因：地址类型（公网、私网）使用错误。

解决方法：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。

网络类型不匹配

问题原因：网络类型不同，ECS实例采用经典网络，PolarDB集群采用VPC网络。

解决方法：

- （推荐）将ECS实例从经典网络迁移到VPC网络，具体操作请参见[ECS实例从经典网络迁移到专有网络](#)。

❓ 说明 网络切换后，ECS实例和PolarDB集群必须处于同一个VPC网络中，才能通过内网互通。

- 使用[ClassicLink](#)功能，使经典网络中的ECS实例可以和VPC网络中的云资源通过内网互通。
- ECS实例使用PolarDB的公网连接地址连接集群，这种方式的性能、安全性、稳定性较差。

7.2.5. 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`developer.aliyundoc.com`，把数据库迁移至PolarDB集群，PolarDB集群的连接地址是`image.developer.aliyundoc.com`，为了保留原来的域名不变，可以创建一个私有域名将`developer.aliyundoc.com`绑定（CNAME）到`image.developer.aliyundoc.com`。绑定成功后在指定VPC中访问`developer.aliyundoc.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，具体费用，请参见[产品定价](#)。

绑定私有域名

- 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[链接地址](#)区域，单击右上角图标切换视图。
- 5. 单击目标私网地址右侧的[绑定私有域名](#)。



- 6. 在绑定私有域名对话框中，输入私有域名的前缀和后缀。



私有域名格式为 `<前缀>.<后缀>`，详细信息请参见下表。

配置	说明
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。

配置	说明
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），具体信息，请参见[PolarDB服务关联角色](#)。

- 单击**确定**。
- 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。

相关API

API	描述
ModifyDBEndpointAddress	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。

7.2.6. 常见问题

本文汇总了PolarDB集群访问相关的常见问题。

- 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: 阿里云不会对PolarDB公网地址产生的流量进行收费。
- Q: 一个集群最多允许创建多少个单节点集群地址？
A: 一个集群最多仅支持6个自定义集群地址，其中包括单节点集群地址。如何创建单节点集群地址，详情请参见[新增自定义集群地址](#)。

警告 创建单节点集群地址后，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: 当出现系统故障需要进行主备切换时，单节点集群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[自动/手动主备切换](#)。
- 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7个集群地址，其中1个为默认集群地址，另外6个为自定义集群地址。
- Q: 可以修改集群地址吗？
A: 默认集群地址和自定义集群地址都支持修改配置，详情请参见[修改连接地址](#)。
- 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[释放连接地址](#)。

7.3. 读写分离

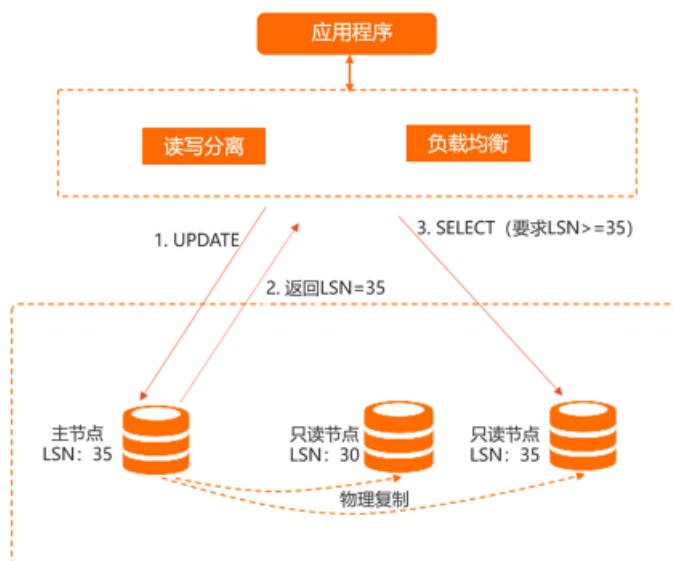
7.3.1. 概述

PolarDB MySQL引擎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

当客户端通过读写分离地址与后端建立连接后，读写分离中间件会自动与主节点和各个只读节点建立连接。在同一个连接内（同一个session内），读写分离中间件会根据各个数据库节点的数据同步程度，来选择合适的节点，在保证数据正确的基础上（写操作之后的读有正确的结果），实现读写请求的负载均衡。



- 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，响应延迟较大。而PolarDB读写分离直接内置在已有的高安全链路中，没有任何额外的组件产生消耗，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，才能将写请求发往主节点而将读请求发往所有节点。

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，PolarDB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如下：

- 只发往主节点：
 - 所有DML操作（INSERT、UPDATE、DELETE、SELECT FOR UPDATE）。
 - 所有DDL操作（建表或库、删表或库、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。
 - KILL（SQL语句中的KILL，非命令KILL）。

- 发往只读节点或主节点：

 **说明** 仅当主库是否接受读选择为是时会发往主节点。

- 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。
 - SHOW PROCESSLIST。

 **说明** 执行SHOW PROCESSLIST命令后，PolarDB将返回所有节点的PROCESSLIST信息。

只读模式转发逻辑如下：

- 不允许执行DDL、DML操作。
- 所有读请求按照负载均衡的方式转发到各只读节点。
- 所有读请求不会转发到主节点。即使主节点已被添加在服务节点中，也不会生效。

功能特性

PolarDB的读写分离功能主要包含以下特性。

- **负载均衡**

PolarDB支持基于负载的自动调度策略，读请求将在多个只读节点中按照活跃连接数自动调度，来保证多个只读节点间的负载均衡。

负载均衡包括主库是否接受读和事务拆分功能：

◦ 主库是否接受读（原主库不接受读）

主库是否接受读选择为否后，普通的读请求将不再发往主节点。而事务内，一致性要求的读请求还是会被发往主节点，以保证业务的需求。另外，当所有只读节点出现故障后，读请求也会发往主节点。如果业务对一致性的要求较低，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节点，也可以通过事务拆分功能来减少真正事务前的读请求发往主节点。广播的请求（例如SET或PREPARE）还是会被发往主节点。

② 说明

- 仅当读写模式为可读可写（自动读写分离）时，支持设置主库是否接受读。主库是否接受读功能默认关闭，关于如何修改主库是否接受读设置，请参见[配置数据库代理](#)。
- 主库是否接受读配置更改后会立即生效。

◦ 事务拆分

开启事务拆分前，数据库代理会将事务内的所有请求都发送到主节点来保证一个会话连接中事务读写的一致性，但这会导致主节点负载过大。开启事务拆分后，数据库代理会识别当前事务的状态，在保证业务中读写一致性的前提下，将正式启动事务前的读请求通过负载均衡模块分流至只读节点。详情请参见[事务拆分](#)。

● 一致性级别

由于PolarDB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，可能会出现一个写请求之后的读请求无法获得最新的写结果，从而导致无法从只读节点中读取最新数据。PolarDB提供了最终一致性、会话读一致性以及全局一致性级别供您选择，详情请参见[一致性级别](#)。

● 连接池

PolarDB支持会话级连接池和事务级连接池，您可以根据业务场景选择合适的连接池，帮助降低因大量连接导致的数据库负载压力，详情请参见[连接池](#)。

● 连接保持

PolarDB新增支持连接保持功能，避免由于一些运维操作（如升级配置、主备切换或升级小版本等）或非运维操作故障（如节点所在服务器故障）导致的连接闪断或新建连接短暂失败的问题，进一步提高PolarDB的高可用性。具体请参见[连接保持](#)。

Hint语法使用说明

PolarDB支持如下Hint语法：

🔔 注意

- Hint的路由优化级别最高，不受一致性级别和事务拆分的约束，使用前请进行评估。
- 仅读写模式为可读可写（自动读写分离）的集群地址支持Hint语法，只读模式下的集群地址和主地址均不支持Hint语法。关于集群地址的读写模式信息，请参见[集群地址的读写模式](#)。

- 支持在SQL语句前加上 `/*FORCE_MASTER*/` 或 `/*FORCE_SLAVE*/` 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。

- 支持在SQL语句前加上 `/*force_node='<节点ID>'*/` 强制指定在某节点执行某查询命令。

例如 `/*force_node='pi-bpxxxxxxx'*/ show processlist`，该 `show processlist` 命令只在 `pi-bpxxxxxxx` 节点执行。如果该节点发生故障，则返回报错 `force hint server node is not found, please check.`。

- 支持在SQL语句前加上 `/*force_proxy_internal*/set force_node = '<节点ID>'` 强制指定在某节点执行所有查询命令。

例如 `/*force_proxy_internal*/set force_node = 'pi-bpxxxxxxx'`，执行该命令后，后续所有查询命令只发往 `pi-bpxxxxxxx` 节点，如果该节点发生故障，则返回报错 `set force node 'rr-bpxxxxx' is not found, please check.`。

说明

- 若您需要通过MySQL官方命令行执行上述Hint语句，请在命令行中加上-c参数，否则该Hint会被MySQL官方命令行过滤导致Hint失效，具体请参见[MySQL官方命令行](#)。
- 通常不建议使用 `/*force_proxy_internal*/` 语法，会导致后续所有查询请求都发往该节点，读写分离失效。
- 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。

7.3.2. 负载均衡

PolarDB支持基于负载的自动调度策略，读请求将在多个只读节点中按照活跃连接数自动调度，来保证多个只读节点间的负载均衡。本文主要介绍负载均衡的两大功能主库是否接受读和事务拆分功能。

主库是否接受读

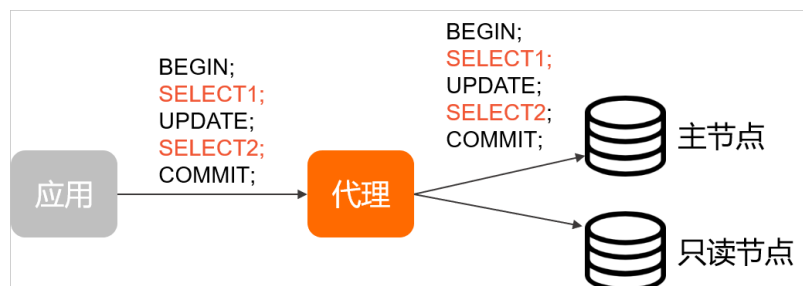
主库是否接受读选择与否，普通的读请求将不再发往主节点。而事务内，一致性要求的读请求还是会被发往主节点，以保证业务的需求。另外，当所有只读节点出现故障后，读请求也会发往主节点。如果业务对一致性的要求较低，可以通过设置一致性级别为最终一致性来减少读请求到主节点，也可以通过事务拆分功能来减少真正事务前的读请求发往主节点。广播的请求（例如SET或PREPARE）还是会被发往主节点。关于如何修改主库是否接受读设置，请参见[配置数据库代理](#)。

说明

- 仅当读写模式为可读可写（自动读写分离）时，支持设置主库是否接受读。
- 主库是否接受读配置更改后会立即生效。

事务拆分

当使用PolarDB可读可写模式集群地址时，读写请求会由数据库代理（Proxy）分发到主节点和只读节点。为了保证一个会话连接中事务读写一致性，代理会将这个会话中所有在事务中的请求都发送到主节点。例如，某些数据库客户端驱动（如JDBC）默认将请求封装在事务中，因此应用的请求都会被发送到主节点，导致主节点压力大，而只读节点几乎没有压力，如下图所示。



为了解决上述问题，PolarDB提供了事务拆分功能，旨在保证业务中读写一致性的前提下，将事务中读请求发送到只读节点，以减轻主节点的压力。您不需要改动应用的代码或配置就可以将事务中的读压力从主节点转移到只读节点，从而提高主节点的稳定性。

 **说明** 仅对读已提交（Read Committed）事务隔离级别的会话支持事务拆分功能。

Proxy会将事务中第一个写请求前的读请求发送到只读节点，从而减轻主节点的负载，由于事务中未提交（COMMIT）的数据在只读节点上处于不可见的状态，为了保障事务中读写一致性，第一个写请求后的所有读写请求仍路由到主节点。如需开启事务拆分服务，请参见[配置数据库代理](#)。

7.3.3. 一致性级别

PolarDB提供了三种一致性级别：最终一致性、会话一致性和全局一致性，满足您在不同场景下对一致性级别的要求。

问题与解决方案

MySQL的读写分离代理功能，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，从而实现了读写分离。但是当数据库负载很高时，例如对大表执行DDL（如加字段）操作或大批量插入数据的时候，延迟会非常严重，从而导致无法从只读节点中读取最新数据。MySQL的读写分离代理功能无法解决由于延迟导致的查询不一致问题。

PolarDB采用了异步物理复制方式实现了主节点和只读节点间的数据同步。主节点的数据更新后，相关的更新会应用到只读节点，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确保了主节点和只读节点间数据的最终一致。PolarDB提供了如下三种一致性级别，满足您在不同场景下对一致性级别的要求：

- **最终一致性**
- **会话一致性**
- **全局一致性**

 **说明** 如何修改一致性级别，请参见[配置数据库代理](#)。

最终一致性

- **功能介绍**

PolarDB是读写分离的架构，传统的读写分离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，例如在一个会话内连续执行如下查询，最后的SELECT结果可能会不同（具体的访问结果由主从复制的延迟决定）。

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

- **适用场景**

若需要减轻主节点压力，让尽量多的读请求路由到只读节点，您可以选择最终一致性。

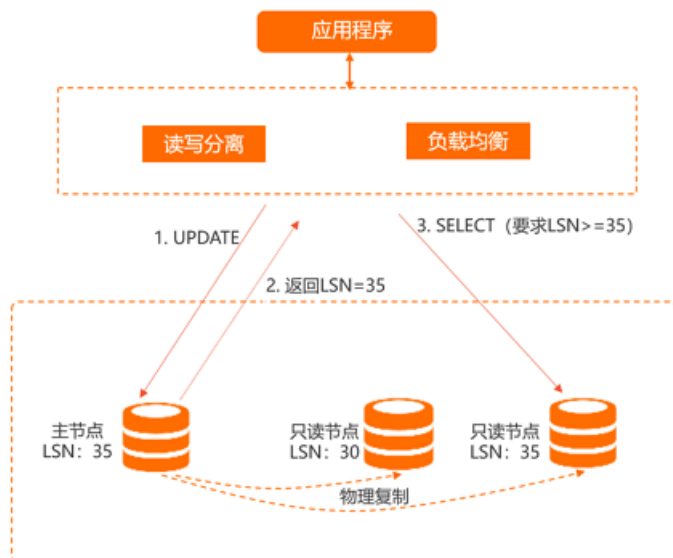
会话一致性

- **功能介绍**

针对最终一致性导致查询结果不同的问题，通常需要将业务进行拆分，将一致性要求高的请求直接发往主节点，而可以接受最终一致性的请求则通过读写分离发往只读节点。这既增加了主节点的压力，影响读写分离的效果，又增加了应用开发的负担。

为解决上述问题，PolarDB提供了会话一致性（也称因果一致性）。会话一致性保证了同一个会话内，一定能够查询到读请求执行前已更新的数据，确保了数据单调性。

在PolarDB的链路中间层做读写分离的同时，中间层会追踪各个节点已经应用的Redo日志位点，即日志序号（Log Sequence Number，简称LSN）。同时每次数据更新时PolarDB会记录此次更新的位点为Session LSN。当有新请求到来时，PolarDB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。且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，又保证了读写分离负载均衡的效果。

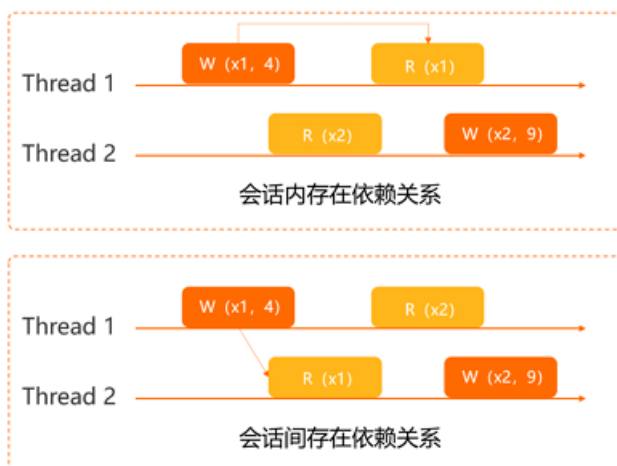
● 适用场景

PolarDB的一致性级别越高，对主库的压力越大，集群性能也越低。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。

全局一致性

● 功能介绍

在部分应用场景中，除了会话内部有逻辑上的因果依赖关系，会话之间也存在依赖关系，例如在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送出去。对数据库来说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，此时会话一致性便无法保证查询结果的一致性。因此PolarDB提供了全局一致性来解决该问题。



每个读请求到达PolarDB数据库代理时，代理都会先去主节点确认当前最新的LSN位点，假设为LSN0（为了减少每次读请求都去获得主节点的最新LSN，内部做了批量优化），然后等待所有只读节点的LSN都更新至主节点的LSN0位点后，代理再将读请求发送至只读节点。这样就能保证该读请求能够读到至请求发起时刻为止，任意一条已完成更新的数据。

全局一致性提供如下两个配置参数：

参数	说明
ConsistTimeout	全局一致性读超时时间，即允许用于只读节点的LSN更新至主节点最新LSN的时间。若超出该时间，PolarDB代理将根据ConsistTimeoutAction参数设置进行对应操作。 取值范围0~300000，默认值为20，单位为毫秒。
ConsistTimeoutAction	全局一致性读超时策略，若未能在参数ConsistTimeout设置的时间内将只读节点的LSN更新至主节点最新LSN，PolarDB代理将根据ConsistTimeoutAction参数设置进行对应操作。 取值范围如下： 0：将读请求发往主节点（默认）。 1：代理返回一个错误报文 <code>wait replication complete timeout, please retry</code> 给应用端。

 说明 如何修改全局一致性读超时时间和全局一致性读超时策略，请参见[配置数据库代理](#)。

● 适用场景

当主从延迟较高时，使用全局一致性可能会导致更多的请求被路由到主节点，造成主节点压力增大，业务延迟也可能增加。因此建议在读多写少的场景下选择全局一致性。

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。
- 若对不同会话间的一致性需求较高，可以选择如下方案之一：
 - 使用HINT将特定查询强制发送至主节点执行。

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

说明

- 若您需要通过MySQL官方命令行执行上述Hint语句，请在命令行中加上-c参数，否则该Hint会被MySQL官方命令行过滤导致Hint失效，具体请参见[MySQL官方命令行](#)。
- Hint的路由优先级最高，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。

- 选择全局一致性。

7.3.4. 连接池

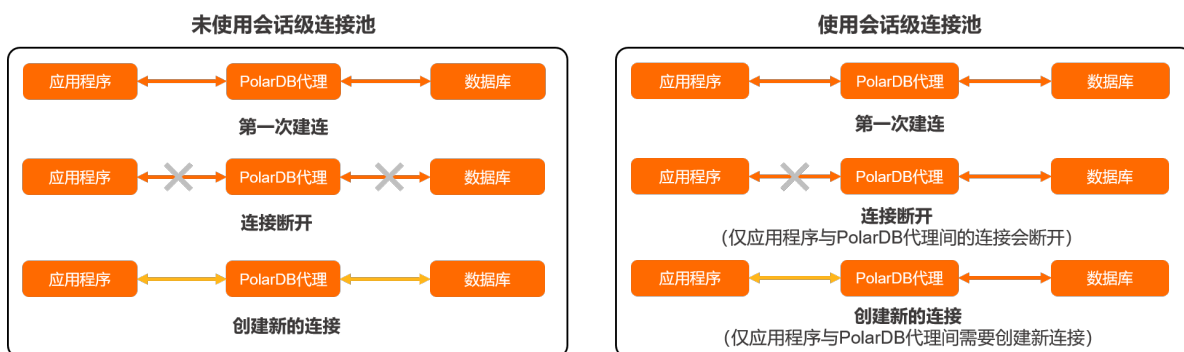
PolarDB支持会话级连接池和事务级连接池，您可以根据业务场景选择合适的连接池，帮助降低因大量连接导致的数据库负载压力。

注意事项

- 更改连接池设置后，仅对新建连接生效。如何修改连接池设置，请参见[配置数据库代理](#)。
- 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权限出错。
- 本文介绍的功能是PolarDB数据库代理的连接池功能。该功能并不影响客户端的连接池功能，如果客户端已经支持连接池，则可以不使用PolarDB数据库代理的连接池功能。

会话级连接池

- 工作原理

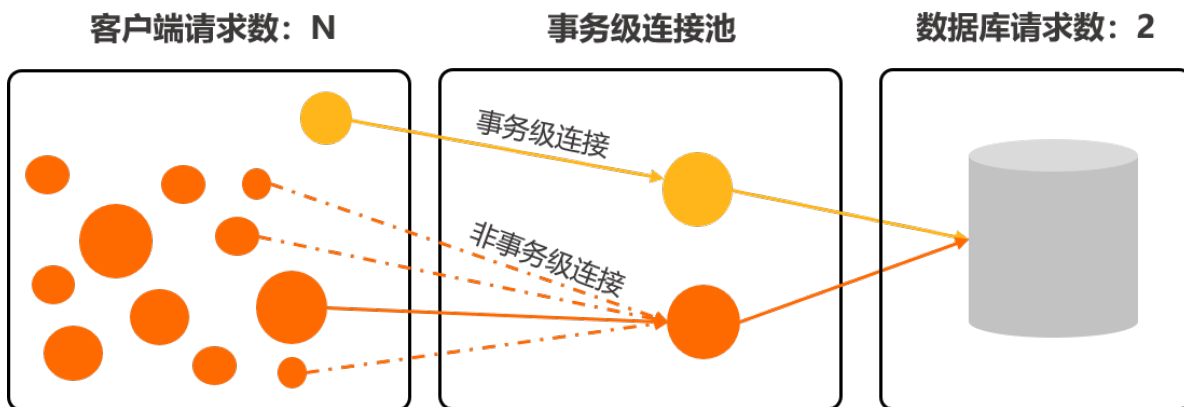


会话级连接池用于减少短连接业务频繁建立新连接导致MySQL负载高。当您的连接断开时，系统会判断当前的连接是否是一个闲置的连接，如果是闲置连接，系统将会代理该连接并保留在连接池中一小段时间，如果这时新的连接建立的话就会直接从连接池里获得连接（命中的条件包括 `user`、`clientip` 和 `db name` 等），从而减少与数据库的建连开销。如果没有可用的连接，则走正常连接流程，重新与数据库建立一个新的连接。

- 使用限制
 - 会话级连接池并不能减少数据库的并发连接数。该优化只能通过降低应用与数据库的建连速率来减少MySQL主线程的开销，从而更好地处理业务请求，但是连接池里空闲的连接会短暂占您的连接数。
 - 会话级连接池也不能解决由于存在大量慢SQL，导致的连接堆积问题，此类问题的核心是先解决慢SQL问题。

事务级连接池

- 工作原理



事务级连接池主要用于减少直接连接到数据库的业务连接数，以及减少短连接场景下频繁建连带来的负载。

开启事务级连接池后，客户端与PolarDB代理间可以存在上千个连接，但代理与后端数据库间可能只存在几十或几百个连接。

PolarDB代理本身并没有最大连接数的限制，连接数的限制主要由后端数据库中计算节点的规格决定。未开启事务级连接池时，每条由客户端发起的连接都需要在后端主节点和所有只读节点上各创建一个对应的连接。

开启事务级连接池后，当客户端发送请求时，会先与PolarDB代理建连，代理不会马上将其与后端数据库建连，而是先从事务级连接池里查找是否存在可用的连接（判断是否为可用连接的条件：`user`、`dbname`和系统变量这3个参数值是否一致）。若不存在，代理会与数据库创建一个新连接；若存在，则从连接池里直接拿出并使用，并在当前事务结束后将该连接放回事务级连接池，方便下个请求继续使用。

- 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
 - 创建临时表
 - 修改用户变量
 - 大报文（例如16 MB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，但是无法保证调用结果的正确性。其中：
 - 1.13.11及以上的数据代理版本支持在 `SELECT SQL_CALC_FOUND_ROWS * FROM t1 LIMIT *` 语句后直接使用 `SELECT FOUND_ROWS()` 命令。但MySQL官方已不推荐该用法，建议您将 `SELECT FOUND_ROWS()` 替换为 `SELECT COUNT(*) FROM t1` 进行查询，详情请参见[FOUND_ROWS\(\)](#)。
 - 支持在 `INSERT` 后直接使用 `SELECT LAST_INSERT_ID()` 语句，来保证查询结果正确性。
- 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，如果业务依赖其他会话级别的系统变量，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` 或者SQL洞察中显示的IP地址和端口，可能会与客户端实际的IP地址和端口不一致。
- 数据库代理会将所有节点上的SHOW PROCESSLIST结果合并后再返回最终结果，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命令可能会返回一个错误，但是实际上KILL命令已经正常执行成功，可通过SHOW PROCESSLIST确定相应的连接是否断开。

连接池的选择

您可以根据如下建议评估选择是否开启连接池以及开启何种类型的连接池：

- 若业务使用的多为长连接且连接数较少，或者业务本身已具备较好的连接池，那么您可以不使用PolarDB的连接池功能。

- 若业务使用的连接数较多（如连接数需求上万）或使用的是Serverless服务（即连接数会随着业务端服务器的扩容而线性增加），且确认您的业务不涉及上述事务级连接池使用限制的场景，那么您可以选择开启事务级连接池。
- 若业务使用的纯短连接，且业务使用场景中包含上述事务级连接池使用限制的场景，那么您可以考虑开启会话级连接池。

7.3.5. 连接保持

PolarDB新增支持连接保持功能，避免由于一些运维操作（如升级配置、主备切换或升级小版本等）或非运维操作故障（如节点所在服务器故障）导致的连接闪断或新建连接短暂失败的问题，进一步提高PolarDB的高可用性。

前提条件

- PolarDB数据库代理（PolarProxy）版本需为2.4.7或以上。

 说明 若2.4.7之前版本需使用连接保持功能，请[提交工单](#)联系技术支持。

- 集群版本需为PolarDB MySQL引擎5.6、5.7或8.0版本且产品系列为集群版。

背景信息

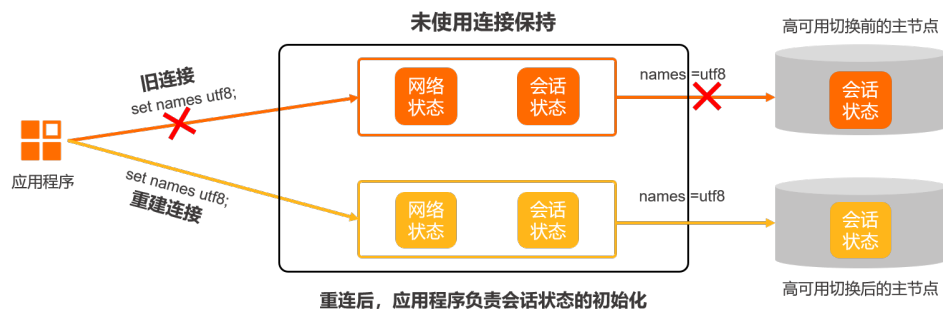
PolarDB已通过高可用组件实现了主节点出现故障时快速进行主备切换，来提供高可用的集群服务。但此类切换过程会对应用程序服务造成影响，导致连接闪断、新建连接短暂失败等问题。引起应用程序服务短暂不可用的场景通常分为如下两种：

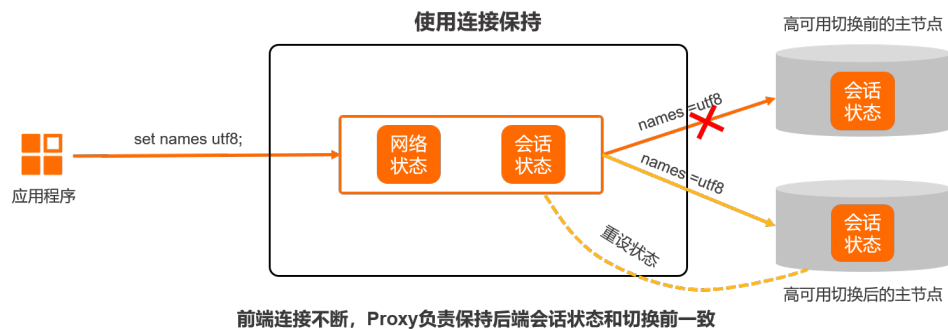
- 计划内（Switchover）：即由数据库控制台或后台管控发起的各种运维操作，如[升级配置](#)、[自动/手动主备切换](#)或[升级小版本](#)等。
- 计划外（Failover）：其它非运维操作引起的故障，比如主节点崩溃，节点所在的主机出现故障等。

通常的解决办法是重启应用程序或保证应用程序具备重连机制，但由于开发周期等原因，在开发设计应用程序的前期可能并未考虑到该问题，导致出现大量的非预期行为甚至应用程序的服务不可用。因此PolarDB新增支持连接保持功能，避免由于一些运维操作或非运维操作故障导致的应用程序服务短暂不可用问题，进一步提高PolarDB集群的高可用性。

实现原理

从连接建立的角度看，PolarDB中的一个会话（Session）包含了一个前端连接（即应用程序和Proxy的连接）和一个后端连接（即Proxy和后端数据库节点的连接）。开启连接保持功能后，当Proxy与旧主节点（即高可用切换前的主节点）连接断开时，Proxy与前端应用的连接保持不断（即应用程序看到的Session），同时Proxy会与新主节点（即高可用切换后的主节点）重新建立连接并且恢复之前的会话状态，以实现应用程序端无感知的高可用切换。





MySQL的连接（会话）通常包括系统变量、用户变量、临时表、字符集编码、事务状态和的Prepare语句状态信息。本文以字符集编码的状态信息为例介绍开启连接保持功能前后的会话状态。

假设应用程序和Proxy间建立了一个连接，并执行了 `set names utf8;` 命令，此时 `names=utf8` 就是这个连接的一个状态。当Proxy在新旧主节点中进行切换时，需要保留这个状态，否则会出现字符集乱码问题。所以连接保持的核心在于保证连接切换前后的会话状态一致。

❓ 说明 当Proxy将连接从旧主节点切换到新主节点的过程中，会存在短暂的新老数据库同时不可写不可读的时间（具体时长受数据库压力影响），所以在切换时，Proxy会暂时停止将应用程序的连接请求路由到后端数据库，并根据新数据库可读可写能力的恢复时间决定路由方向：

- 若新数据库在60s内恢复了可写可读能力，Proxy会将请求路由到新数据库。
- 若新数据库的可读可写能力未能在60s内恢复，Proxy会断开与应用程序间的连接，应用程序需重新发起连接请求（即此时与未使用连接保持功能时的行为一致）。

如何使用

- 数据库代理（PolarProxy）版本为2.4.7及以上的PolarDB MySQL引擎集群，该功能默认开启，无需手动开启。
- 数据库代理（PolarProxy）版本为2.4.7之前的PolarDB MySQL引擎集群，请[提交工单](#)联系技术支持开启该功能。

注意事项

连接保持功能无法保持如下场景中的连接：

- 当连接切换时，连接（会话）上存在临时表。
- 当连接切换时，Proxy正在从数据库接收结果报文但只接收了部分结果报文，比如执行SELECT语句后，需从数据库接收一个包含100 MB数据的结果报文，但切换时数据库只传输了10 MB数据。
- 当连接切换时，连接（会话）上有正在执行中的事务（如 `begin;insert into;` ）。

❓ 说明 针对上述场景中的最后两种，若该场景是由于计划内（Switchover）的操作导致的连接切换，Proxy会先保留该请求并等待2s，如果剩余的结果报文能在2s内全部返回或事务能在2s内执行完成，Proxy将在等待时间结束后执行连接切换，从而提高连接保持的概率。

性能测试

- 测试环境
 - 测试用集群如下：
 - PolarDB MySQL引擎8.0版本集群（默认包含1个主节点和2个只读节点）。
 - 节点规格为4核16 GB独享规格（polar.mysqlx4.large）。

- 测试工具：SysBench。
- 测试数据如下：
 - 20张表，其中每张表包含10000行记录数。
 - 并发度为20。
- 测试方法
在不同运维场景下，测试PolarDB集群的连接保活率（即执行运维操作前后的连接保持比例）。
- 测试结果
在如下测试的运维场景中，PolarDB集群均能保持100%的连接保活率。

说明

- 当前升级集群规格仅支持在未跳级升级规格（如从4核升级至8核）的场景下能够保持100%的连接保活率，如果是从4核升级到16核（或者更大规格），可能出现连接闪断。
- 若删除只读节点时触发了数据库代理节点收缩，将导致部分连接的闪断。
- 内核小版本升级的场景中仅包括数据库内核引擎的小版本升级，而不包含数据库代理的小版本升级，数据库代理的小版本升级会出现连接闪断。

运维场景	保活率
主备切换	100%
内核小版本升级	100%
升级集群规格	100%
增删节点	100%

7.4. 动态脱敏

7.4.1. 概述

本文主要介绍了PolarDB数据库代理（Proxy）提供的动态脱敏功能。

前提条件

PolarDB数据库代理版本需为2.4.12或以上。如何查看和升级当前数据库代理版本，请参见[版本管理](#)。

说明 当前PolarDB数据库代理版本为1.x.x时，暂不支持通过控制台升级到2.4.12及以上，请[提交工单](#)联系技术支持进行升级。

脱敏方案

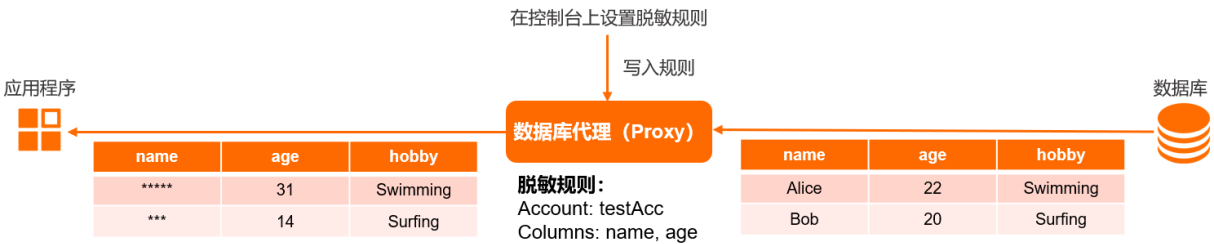
在数据库使用中，需要实时地从生产环境中的数据库（即生产库）获取最新的客户数据来进行报表生成、数据分析、开发测试等。但为了不泄露真实的客户个人信息（Personal Identifiable Information），需要将这些数据进行脱敏处理后才能提供给第三方使用。目前脱敏方案包括动态脱敏和静态脱敏两种，PolarDB数据库代理使用的是动态脱敏方案。

脱敏方案对比

脱敏方案	说明	优点	缺点
动态脱敏	当应用程序发起数据查询请求时，系统会在数据库内部对敏感数据进行变换后再返回给应用程序。 开始查询前，仅需指定需要进行脱敏查询的数据库账号，以及需要脱敏的数据库、表或列的名称即可。	• 无需修改业务端任何代码，使用成本低。 • 使用生产库的实时数据，能有效保证数据的实时性。	由于需要对实时数据进行脱敏，因此相较于镜像库，生产库的查询性能会略受影响。
静态脱敏	将生产库的全量数据导入至一个镜像数据库中，同时在导出数据过程中对敏感数据进行加密或脱敏。	应用程序访问的不是真正的生产库而是镜像库，因此脱敏过程完全不会影响正常业务对生产库的使用。	• 需要在数据导入的工具中开发一套修改敏感数据的组件，开发成本高。 • 镜像数据库中的数据会落后于生产库的数据，实时性较差。

工作原理

在PolarDB控制台上设置完脱敏规则后，控制台会将规则写入至Proxy中。当应用程序通过规则中指定的账号连接至数据库并查询规则中指定的列时，Proxy会先将数据库返回的数据脱敏转换，再将脱敏后的数据返回给客户端。



上图的脱敏规则为：

- 仅当使用 `testAcc` 账号连接数据库进行查询时生效。
- 仅对查询结果中 `name` 和 `age` 列的数据进行脱敏。

当应用程序使用 `testAcc` 连接数据库，并查询某个表中的 `name`、`age` 和 `hobby` 时，Proxy会将 `name` 和 `age` 列的数据进行脱敏后再返回，`hobby` 列的数据则未进行脱敏原样返回。

Proxy会对不同类型的数据进行不同的脱敏处理，具体脱敏处理说明如下表所示。

数据类型	脱敏处理说明	示例
整数型（TINYINT、SMALLINT、MEDIUMINT、INT、BIGINT）	Proxy会根据该数据的类型选择一个符合该类型定义的随机值。	• 原始值：12345 • 脱敏值（随机）：28175
小数型（DECIMAL、FLOAT、DOUBLE）		• 原始值：1.2345 • 脱敏值（随机）：8.2547

数据类型	脱敏处理说明	示例
时间型（DATE、TIME、DATETIME、TIMESTAMP、YEAR）		- 原始值：2021-01-01 00:00:00 - 脱敏值（随机）：4926-12-13 17:23:07
除上述3种类型之外的数据	Proxy会统一使用星号（*）进行替换	- 原始值：Elon Musk - 脱敏值：*****

注意事项

- 动态脱敏功能仅对集群地址（包括默认集群地址和自定义集群地址）生效，使用主地址连接数据库进行查询时，动态脱敏功能不生效。查看集群连接地址的操作步骤，请参见[查看连接地址和端口](#)。
- 查询数据时，若查询结果中包含需要脱敏的列，且单行数据超过16 MB时，执行该查询命令的连接会断开。
例如，若已有一张包含了 name 和 description 列的表 person，其中 name 列需要脱敏，且 description 列中的数据超过了16 MB。当执行 SELECT name, description FROM person 时，执行该查询命令的连接则会断开。
- 若需要脱敏的数据列作为函数入参，脱敏功能会失效。
例如，已创建了一条需要对 name 列数据进行脱敏的规则，当执行 SELECT CONCAT(name, '') FROM person 进行查询时，应用程序仍然能够读取到 name 的原始值。
- 若需要脱敏的数据列同时被用于UNION中，脱敏功能可能会失效。
脱敏失效示例：已创建了一条需要对 name 列数据进行脱敏的规则，当执行 SELECT hobby FROM person UNION SELECT name FROM person 语句进行查询时，应用程序仍然能够读取到 name 的原始值。

使用方法

详情请参见[管理脱敏规则](#)。

附录：性能影响

脱敏功能会在以下3种场景中对集群的性能产生影响。

 说明 本文使用了集群的只读QPS值进行对比。

场景说明		性能影响
账号是否加入脱敏规则	查询是否命中规则	
否	否	此时等同于当前账号未使用脱敏功能，性能没有损耗。
是	否	此时Proxy仅会分析结果集中的列定义数据，不会对真正的查询结果（即行数据）进行脱敏。 性能损耗约为6%（即开启脱敏功能前后的只读QPS值相差约6%）。
	是	此时Proxy不仅会分析结果集中的列定义数据，还会对真正的查询结果（即行数据）进行脱敏。 此时性能损耗与结果集大小相关，结果行数越多损耗越大。 在返回单行结果的情况下，性能损耗约为6%。

7.4.2. 管理脱敏规则

您可以在PolarDB控制台上新建以及启用或禁用脱敏规则，同时您还可以修改或删除已有的脱敏规则。本文将介绍相关操作步骤。

前提条件

PolarDB数据库代理版本需为2.4.12或以上。如何查看和升级当前数据库代理版本，请参见[版本管理](#)。

注意事项

- 动态脱敏功能仅对集群地址（包括默认集群地址和自定义集群地址）生效，使用主地址连接数据库进行查询时，动态脱敏功能不生效。查看集群连接地址的操作步骤，请参见[查看连接地址和端口](#)。
- 查询数据时，若查询结果中包含需要脱敏的列，且单行数据超过16 MB时，执行该查询命令的连接会断开。
例如，若已有一张包含了 `name` 和 `description` 列的表 `person`，其中 `name` 列需要脱敏，且 `description` 列中的数据超过了16 MB。当执行 `SELECT name, description FROM person` 时，执行该查询命令的连接则会断开。
- 若需要脱敏的数据列作为函数入参，脱敏功能会失效。
例如，已创建了一条需要对 `name` 列数据进行脱敏的规则，当执行 `SELECT CONCAT(name, '') FROM person` 进行查询时，应用程序仍然能够读取到 `name` 的原始值。
- 若需要脱敏的数据列同时被用于UNION中，脱敏功能可能会失效。
脱敏失效示例：已创建了一条需要对 `name` 列数据进行脱敏的规则，当执行 `SELECT hobby FROM person UNION SELECT name FROM person` 语句进行查询时，应用程序仍然能够读取到 `name` 的原始值。

新建脱敏规则

1. 登录[PolarDB控制台](#)。
2. 在控制台左上角，选择集群所在地域。
3. 找到目标集群，单击集群ID。
4. 在左侧导航栏，选择配置与管理 > 规则管理。
5. 单击页面左上角的新增，在新建脱敏规则对话框中，设置以下参数。

配置脱敏规则

参数		是否必填	说明
基本信息	规则名称	必填	脱敏规则的名称。字符长度不能超过30个字符。
	规则描述	选填	规则的描述。字符长度不能超过64个字符。
	启/禁规则	不涉及	启/禁规则开关。  说明 新建脱敏规则时，启/禁规则开关默认打开。

参数		是否必填	说明
规则配置	数据库账号名	选填	需要应用当前规则的数据库账号名称。支持如下选项： - 所有数据库账号：表示当前规则对该集群中的所有数据库账号生效，此时右侧文本框无需填写任何内容。 - 包含：表示当前规则仅对目标数据库账号生效，此时右侧文本框中必须填入一个或多个数据库账号名，多个账号间用英文逗号（,）分隔。 - 不包含：表示当前规则会对该集群中，除目标数据库账号外的所有数据库账号生效，此时右侧文本框中必须填入一个或多个数据库账号名，多个账号间用英文逗号（,）分隔。  说明 文本框中填入的数据库账号名称需满足支持如下格式中的任意一种： - 账号名。示例：user。 - 账号名@完整IP。示例：user@1.1.1.1。 - 账号名@含通配符的IP。示例：user@1.1.1.%、user@%.1.1.1 或 user@1.%.1。 - 账号名@IP/掩码。示例：user@1.1.1.0/255.255.255.0。
	数据库名	选填	需要应用当前规则的数据库名称。支持如下选项： - 所有数据库：表示当前规则对该集群中的所有数据库生效，此时右侧文本框中无需填入任何内容。 - 包含：表示当前规则仅对目标数据库生效，此时右侧文本框中必须填入一个或多个数据库名称，多个名称间用英文逗号（,）分隔。
	表名	选填	需要应用当前规则的表名称。支持如下选项： - 所有表：表示当前规则对该集群中的所有表生效，此时右侧文本框中无需填入任何内容。 - 包含：表示当前规则仅对目标表生效，此时右侧文本框中必须填入一个或多个表名称，多个名称间用英文逗号（,）分隔。
	字段名	必填	需要应用当前规则的字段名称。支持同时输多个字段名称，多个名称间用英文逗号（,）分隔。

6. 单击**确认**。

启用或禁用规则

- 1. 登录**PolarDB控制台**。
- 2. 在控制台左上角，选择集群所在地域。
- 3. 找到目标集群，单击集群ID。
- 4. 在左侧导航栏，选择**配置与管理 > 规则管理**。

5. 找到目标规则，打开或关闭启/禁规则开关。

☐	规则名称	描述	启/禁规则	操作
☐				修改 删除

? 说明

- 您也可以在规则列表中同时选中多个目标规则，然后单击列表下方的启用或禁用进行批量启用或禁用规则。
- 被禁用的脱敏规则不会被删除，而是继续保留在规则列表里。如有需要，您可以再次启用该规则。

6. 在弹出的对话框中，单击确定即可。

修改规则

- 登录PolarDB控制台。
- 在控制台左上角，选择集群所在地域。
- 找到目标集群，单击集群ID。
- 在左侧导航栏，选择配置与管理 > 规则管理。
- 找到目标规则，单击右侧操作中的修改，在弹出的对话框中，根据业务需要修改规则参数。规则参数的具体说明，请参见配置脱敏规则。

☐	规则名称	描述	启/禁规则	操作
☐				修改 删除

? 说明 仅支持修改规则描述和规则配置中的各参数，不支持修改规则名称。

6. 单击确认。

删除脱敏规则

- 登录PolarDB控制台。
- 在控制台左上角，选择集群所在地域。
- 找到目标集群，单击集群ID。
- 在左侧导航栏，选择配置与管理 > 规则管理。
- 找到目标规则，单击右侧操作栏中的删除。

☐	规则名称	描述	启/禁规则	操作
☐				修改 删除

? 说明 您也可以在规则列表中同时选中多个目标规则，然后单击列表下方的删除进行批量删除。

6. 在弹出的对话框中，单击确定即可。

相关API

API	描述
DescribeMaskingRules	查询PolarDB集群下所有的脱敏规则或目标脱敏规则的详情。
ModifyMaskingRules	修改脱敏规则配置或新增脱敏规则。
DeleteMaskingRules	删除目标脱敏规则。

7.5. 配置数据库代理

本文介绍如何通过PolarDB的集群地址来配置数据库代理。

前提条件

产品系列需为PolarDB MySQL引擎集群版和历史库集群版。单节点和历史库单节点版均不支持该功能。更多关于3个系列的介绍，请参见[产品系列](#)。

注意事项

- 仅PolarDB MySQL引擎8.0版本集群在配置数据库代理时支持开启并行查询和设置并行度。
- 对于数据库代理（PolarProxy）版本为1.6.0及以上版本的PolarDB MySQL引擎集群，您可以直接新增自定义集群地址；若您需要为1.6.0之前版本的集群新增自定义地址，请[提交工单](#)联系售后服务。

操作步骤

- 登录[PolarDB控制台](#)。
- 在控制台左上角，选择集群所在地域。
- 找到目标集群，单击集群ID。
- 在集群基本信息页的[链接地址](#)区域，找到目标集群地址，单击目标集群地址名称右侧的[编辑配置](#)。
- 在弹出的对话框内，根据业务需要修改集群地址的配置。具体配置项如下表所示：

配置表

配置项		说明
网络信息		PolarDB为每个集群地址默认提供了私网连接地址，若需要修改该地址或申请公网和经典网络连接地址，请参见 申请集群地址和主地址 。
集群设置	读写模式	集群地址的读写模式，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。
	地址名称	输入集群地址的名称。

配置项		说明
服务节点	可选节点和已选节点	从左侧可选节点框内（包括主节点和所有只读节点），选中想要加入集群地址用于处理读请求的节点，单击图标，将其移动到右侧已选节点框中。  说明 节点的选择不影响读写模式： - 读写模式为可读可写（自动读写分离）时，无论已选节点是否选中主节点，写请求都只会发往主节点。 - 读写模式为只读时，所有读请求按照负载均衡的方式转发到各只读节点，不会转发到主节点。即使主节点已被添加在已选节点中，也不会生效。
	新节点自动加入	新增的节点是否要自动添加到该地址中。
负载均衡设置	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。
	主库是否接受读	- 选择否，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。 - 选择是，查询SQL可以发送到主节点和只读节点。 关于主库是否接受读的更多介绍，请参见主库是否接受读。  说明 仅可读可写（自动读写分离）模式下支持该配置。
	事务拆分	开启或关闭事务拆分。关于事务拆分的更多介绍，请参见事务拆分。  说明 仅可读可写（自动读写分离）模式下支持该配置。
一致性设置	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（弱）、会话一致性（中）和全局一致性（强），详情请参见一致性级别。 - 读写模式为只读时，默认一致性级别为最终一致性（弱）且不可更改。  说明 一致性级别修改后对所有连接立即生效。
	全局一致性读超时时间	等待只读节点同步到最新数据的超时时间，单位为ms。取值范围为0~6000 ms，默认取值为20 ms。  说明 仅当一致性级别为全局一致性（强）时支持该配置。

配置项		说明
	全局一致性读超时策略	在只读节点等待超时后，PolarDB的默认策略，取值范围如下： 0，发送该请求到主节点（默认值） 1，SQL报错（wait replication complete timeout, please retry）  说明 仅当一致性级别为全局一致性（强）时支持该配置。
连接池设置	连接池	您可以选择关闭（默认选项）、开启会话级连接池或开启事务级连接池。关于连接池的更多介绍，请参见连接池。  说明 仅当读写模式为可读可写（自动读写分离）时支持该配置。
HTAP优化	并行查询	开启或关闭并行查询，设置并行度，并选择并行引擎。并行查询默认关闭。详情请参见并行查询（Parallel Query）。并行引擎包括单机并行和多机并行(MPP)。 - 单机并行：即并行查询（Parallel Query），提供了能够有效利用单节点内多核CPU进行并行加速的能力。详情请参见并行查询（Parallel Query）。 - 多机并行(MPP)：提供了能够有效利用集群内所有节点的计算资源进行并行加速的能力。  说明 仅当集群版本为PolarDB MySQL引擎8.0版本的集群地址在只读模式下支持该配置。
	行存/列存自动引流	开启或关闭行存和列存自动引流功能。详情请参见行存和列存自动引流。  说明 PolarDB MySQL引擎8.0.1版本且内核小版本需为8.0.1.1.22及以上的集群地址在可读可写（自动读写分离）模式下，且服务节点的已选节点中包含了至少一个只读列存节点时，才支持该配置。

6. 单击**确定**即可。

相关API

API	描述
DescribeDBClusterEndpoints	查询集群地址。
ModifyDBClusterEndpoint	修改集群地址。
DeleteDBClusterEndpoint	释放自定义集群地址。

7.6. 升级数据库代理的规格

数据库代理企业版默认的规格可以满足用户绝大部分的应用场景，当客户进行压力测试或者业务吞吐量很高的时候，可以考虑升级数据库代理企业版的规格，避免数据库代理成为瓶颈。本文档介绍了升级数据库代理企业版规格的步骤。

背景信息

当前数据库代理企业版的规格，会根据数据库节点以及节点规格的变化自动触发增加或减少资源，无需用户选择。具体规则如下：

- **企业通用版**：集群中数据库节点总核数的1/6。向上取整，以2 vCPU为粒度，最小2 vCPU，最大64 vCPU。
- **企业独享版**：集群中数据库节点总核数的1/4。向上取整，以2 vCPU为粒度，最小2 vCPU，最大128 vCPU。

例如，当前集群的数据库代理为**企业独享版**，且集群包含2个数据库节点，每个节点都为4核，则数据库代理占用的CPU核数为2核。

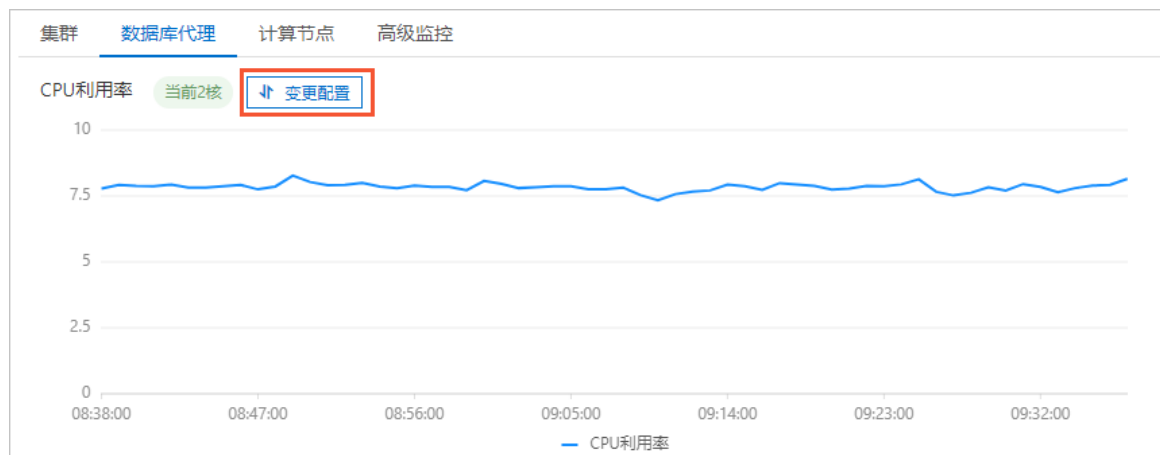
注意事项

此升级过程会将新的会话连接至新规格的数据库代理，持续24小时后，依然连接原规格数据库代理的会话会出现闪断。建议您在业务低谷期执行升级操作，并确保应用具备重连机制。

操作步骤

本文以数据库代理企业独享版为例进行说明。

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，单击目标集群ID。
4. 在**诊断与优化 > 性能监控**页的**数据库代理**页签，单击**变更配置**。



5. 在**变更配置**对话框中，选择数据库代理企业版规格的升级方案。

升级方案如下：

- 标准配置：企业通用版vCPU的数量为集群中数据库节点总核数的1/6，企业独享版vCPU的数量为集群中数据库节点总核数的1/4，向上取整。
- 标准配置*2：vCPU的数量是标准配置的2倍。
- 标准配置*4：vCPU的数量是标准配置的4倍。



变更配置对话框截图。对话框标题为“变更配置”，右上角有关闭按钮。下方有三个配置选项：
- 标准配置 2核
- 标准配置*2 4核 (被选中)
- 标准配置*4 8核
下方有一个黄色背景的备注区域，包含以下信息：
- 备注：
- 此变更过程会将新的会话连接至新规格的数据库代理，持续24小时后，依然链接原规格的数据库代理的会话会出现闪断。
- 标准配置的核数计算方法：
- Proxy企业通用版：集群数据库节点总核数1/6（共享），以2 vCPU为粒度向上取整，最小2 vCPU，最大64 vCPU。
- Proxy企业独享版：集群数据库总核数1/4（独享），以2 vCPU为粒度向上取整，最小2 vCPU，最大128 vCPU。
底部有三个按钮：
- 立即升级 (蓝色)
- 可维护窗口升级 (02:00-03:00)
- 取消

6. 单击**立即升级**或**可维护窗口升级**。

若选择在**可维护窗口升级**，您还可以在**计划任务**页查看该任务的具体信息或取消该任务，详情请参见[查看或取消计划任务](#)。

7. 在弹出的对话框中，单击**确定**即可。

7.7. 性能监控

数据库代理企业版提供了性能监控项和秒级监控频率，方便您掌握集群的运行状态并通过细粒度的监控数据快速定位运维问题。

注意事项

- 2021年12月9日之前购买的集群需要升级数据库代理版本才能使用性能监控功能，2021年12月9日之后购买的集群可直接使用性能监控功能。
- 数据库代理升级期间，会将新的会话连接至新规格的数据库代理，持续24小时后，依然连接原规格数据库代理的会话会出现闪断。建议您在业务低谷期执行升级操作，并确保应用具备重连机制。

操作步骤

- 进入数据库代理性能监控页面。
 - 登录[PolarDB控制台](#)。
 - 在控制台左上角，选择集群所在地域。

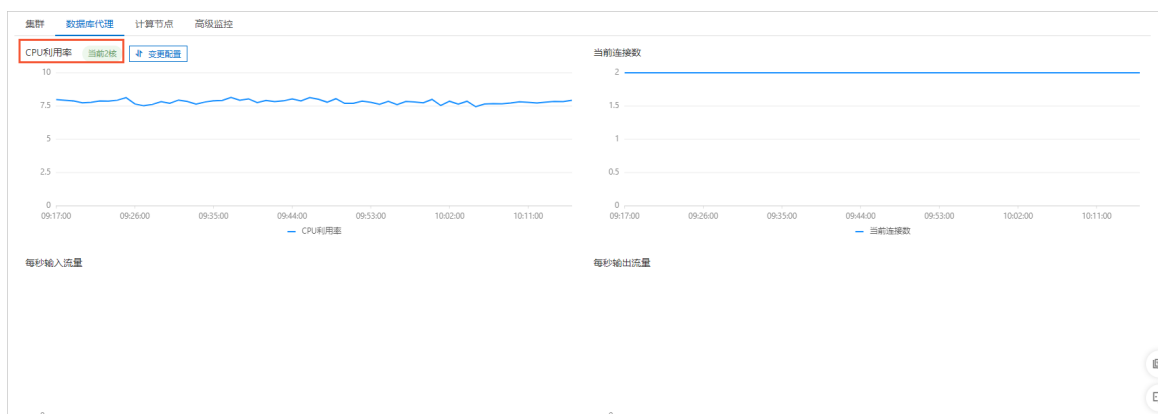
- iii. 找到目标集群，单击集群ID。
 - iv. 在左侧导航栏中，选择**诊断与优化 > 性能监控**。
 - v. 在**性能监控**页面，单击**数据库代理**页签。
2. （可选）升级数据库代理版本。

说明 如果是2021年12月9日之前购买的集群，请先按照以下操作升级数据库代理版本；如果是2021年12月9日之后购买的集群，请忽略该步骤。



- i. 在**数据库代理**页签，单击**规格升级**或**可维护窗口升级**。
若选择在**可维护窗口升级**，您还可以在**计划任务**页查看该任务的具体信息或取消该任务，详情请参见[查看或取消计划任务](#)。
 - ii. 在弹出的对话框中，单击**确定**即可。
3. 查看数据库代理监控指标数据。

在**数据库代理**页签，您可以根据业务需求查看数据库代理的监控指标信息。



CPU利用率表示数据库代理所占用集群节点的CPU核数的利用率。数据库代理占用的集群节点的CPU核数计算公式：**企业通用版**为集群中数据库节点总核数的1/6，**企业独享版**为集群中数据库节点总核数的1/4，向上取整。

例如，当前**企业独享版**集群包含2个数据库节点，每个节点都为4核，则数据库代理占用的CPU核数为2核。

7.8. 常见问题

本文汇总了PolarDB数据库代理相关的常见问题。

- Q：为什么刚插入的语句，立即查的时候查不到？

A：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，详情请参见[会话一致性](#)。

- Q：是否支持0毫秒延迟的读取？

A：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。如何查看主地址，请参见[查看连接地址和端口](#)。

- Q：为什么当主节点负载很高时，只读节点上负载却很低？

A：因为在默认情况下，事务中的请求都会路由到主节点。此时，您可以尝试如下解决办法：

- 若在Sysbench压测中，您可以在0.5版本的Sysbench中加上 `--oltp-skip-trx=on` 或在1.0版本的Sysbench中加上 `--skip-trx=on` 去掉事务后，再进行压测。
- 若在实际业务中，由于事务较多导致主节点负载过高时，您可以开启读写分离下的事务拆分功能减轻主节点的压力，详情请参见[事务拆分](#)。

- Q：为什么某个节点请求数比别的节点多？

A：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。

- Q：新增的只读节点会自动加入到读写分离吗？

A：新增只读节点之后新建的读写分离连接才会转发请求到该只读节点。若需要使新增只读节点之前建立的读写分离连接也转发请求到新增的只读节点，则需要通过重启应用程序等操作断开该连接并重新建立连接。

- Q：数据库代理企业版的企业通用版和企业独享版有什么区别？

A：

- **企业通用版**：对应集群子系列的通用规格，它可以共享CPU物理资源，可根据业务负载，提供智能秒级资源弹性扩展能力。
- **企业独享版**：对应集群子系列的独享规格，它可以独占CPU物理资源，具有更好的性能稳定性。

- Q：数据库代理历史版本（免费版）能否继续使用？

A：不能继续使用

- 存量集群的数据库代理（免费版）于2021年12月9日已自动切换到数据库代理企业版，并于2022年4月1日开始收费。
- 新购集群仅提供数据库代理企业版，并于2022年4月1日正式开始收费。

8. 数据库管理

您可以在控制台管理所有的PolarDB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. 找到目标集群，单击集群ID。
4. 在左侧导航栏中，单击配置与管理 > 数据库管理。
5. 单击创建数据库。
6. 在创建数据库面板中，设置以下参数。

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。 - 长度不超过64个字符。 - 数据库名称在实例内必须是唯一的。 ❓ 说明 请勿使用保留关键字作为数据库名称，如 <code>test</code>、<code>mysql</code> 等。
支持字符集	选择数据库支持的字符集，例如utf8mb4、utf8、gbk、latin1等。
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，取值范围为读写、只读、仅DML、仅DDL或只读+索引。
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 <code>http://</code> 或 <code>https://</code> 开头。 - 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. 找到目标集群，单击集群ID。
4. 在左侧导航栏中，单击配置与管理 > 数据库管理。
5. 找到目标数据库，单击操作栏中的删除。

6. 在弹出的对话框中，单击确定。

相关API

API	描述
CreateDatabase	创建数据库。
DescribeDatabases	查看数据库列表信息。
ModifyDBDescription	修改数据库描述。
DeleteDatabase	删除数据库。

9. 变更集群配置

9.1. 概述

本文简要介绍了集群变更配置的功能以及支持的使用场景。

PolarDB集群支持在线变配，变更配置过程中无需对数据库加锁。它支持三维扩展能力，集群变配分钟级生效：

- 计算能力的纵向扩展：集群规格升降配，支持对主节点和只读节点分别升降配，即只读节点的规格可以与主节点规格不一致。

 **说明** PolarDB集群提供了更灵活的变配方案，只读节点的规格可以单独变配。无需与主节点规格保持一致，您可以根据实际业务需求，灵活分配不同的节点规格。

- 计算能力横向扩展：增加或删除只读节点，最多扩展至16个计算节点。
- 存储空间的横向扩展：按容量计费（按量付费）的存储空间采用Serverless架构，容量随数据量的变化而自动在线调整；按空间计费（包年包月）的存储空间，支持通过手工扩容的方式来扩展存储容量。当您的数据量较大时，推荐使用存储包以降低存储成本，详情请参见[购买存储包](#)。

集群变更配置功能支持以下使用场景：

- **手动变配**：您可以手动调整PolarDB MySQL引擎集群的规格。具体操作请参见[手动变配](#)。

 **说明** 产品系列需为PolarDB MySQL引擎集群版或历史库集群版。单节点和历史库单节点版均不支持该场景。

- **自动变配（弹性扩缩容）**：您可以设置自动扩缩容参数，PolarDB MySQL引擎集群版实例将根据您设置的参数自动扩容和自动回缩。具体操作请参见[集群规格自动变配（自动扩容）](#)和[本地资源自动变配（秒级扩核）](#)。您可以通过体验馆案例[PolarDB MySQL集群自动扩容和回缩](#)，体验PolarDB MySQL引擎自感知、自运维的云服务。
- **临时升配**：您可以临时升级包年包月PolarDB集群的规格，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。具体操作请参见[临时升配](#)。

 **说明** 仅包年包月的PolarDB集群支持临时升配。

- **增删节点**：您可以手动增加或删除只读节点。具体操作请参见[增加或删除节点](#)。

 **说明** 产品系列需为PolarDB MySQL引擎集群版、多主架构集群版或历史库集群版。单节点和历史库单节点版均不支持该场景。

- **历史库单节点版升级为历史库集群版**：您可以将存量的历史库单节点版升级为历史库集群版独享规格。具体操作请参见[历史库单节点版升级为历史库集群版](#)。
- **手工扩容**：您可以手动扩展存储空间容量。具体操作请参见[手工扩容](#)。

9.2. 手动变配

您可以根据业务波动变更PolarDB集群的配置。如果您的业务量比较平稳，当您需要变更集群节点规格时，推荐使用手动变配的方式。本文介绍如何手动升级或降级PolarDB集群的规格。

前提条件

- 产品系列需为PolarDB MySQL引擎集群版和历史库集群版。单节点和历史库单节点版均不支持该功能。更多关于产品系列的介绍，请参见[产品系列](#)。
- PolarDB集群没有正在进行的配置变更。

费用说明

具体的费用说明，请参见[变更配置费用说明](#)。

 **说明** 推荐您使用[按量付费+计算包](#)的购买方式，随时可以升级或降级配置，费用更划算。

注意事项

- 集群规格的升降级不会对集群中已有数据造成任何影响。
- 规格变更的耗时主要与集群计算节点数量有关（约5分钟/计算节点。例如，包含2个计算节点的集群，规格变更预计10分钟左右完成）。此外，规格变更的耗时还与数据库负载、库表数量等因素有关。
- 规格变更过程中，每个连接地址都会有不超过30秒的连接闪断。建议您在业务低谷期执行变更，并确保应用具备自动重连机制。
- PolarDB集群变更配置期间，只读请求相比读写请求的滞后时间，可能比正常运行状态的滞后时间更长。
- 您可以对整个集群进行规格升降级，也可对集群中的单个节点进行规格升降级。

主节点和只读节点可单独变配规格

PolarDB集群支持对主节点和只读节点分别升降配，即只读节点的规格可以与主节点规格不一致。

使用限制

- 同一集群中，确保至少有一个只读节点与主节点规格保持一致。
- 只读节点规格建议不要与主节点规格相差过大：
 - 只读节点的内存不能小于主节点内存的一半。
 - 只读节点的CPU规格与主节点CPU规格对应规则如下：

主节点CPU规格（core）	允许只读节点最小CPU规格（core）
2	2
4	4
8	4
16	8
32	16
64	32
88	32

- [开启热备](#)后的只读节点需要与主节点规格保持一致。
- [只读列存节点](#)场景下建议只读列存节点规格比主节点规格更高。

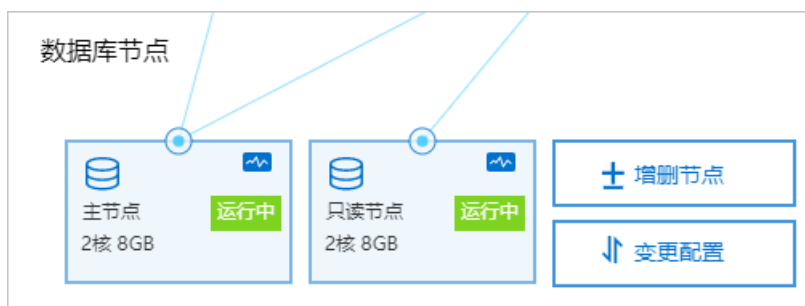
 **说明** PolarDB多主架构集群版不受以上使用限制。

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. 您可以按照如下两种方式中的任意一种进入变更配置对话框：
 - 方法一：
在集群列表页，找到目标集群，单击右侧操作栏中的变更配置。



- 方法二：
 - a. 在集群列表页，单击目标集群ID。
 - b. 在基本信息页的数据库节点区域，单击变更配置。



4. 您可以根据业务需要选择升配或降配，单击确定。



❗ 说明 仅包年包月集群支持临时升配，详情请参见临时升配。

5. 在变配页面中，选择所需的节点规格和切换时间并选中服务协议，单击立即购买。

说明

- 同一集群中，至少确保一个只读节点与主节点规格保持一致。
- 您可以对整个集群进行规格升降级，也可对集群中的单个节点进行规格升降级。
- 仅升配和降配时支持切换时间选项，临时升配不支持该选项。
- 您可以选择切换时间为立即切换或定时切换。若选择定时切换，可选的时间范围为未来24小时内的任意时间点，升级配置的任务将在选定时间点后的30分钟内完成。您还可以在计划任务页查看该任务的具体信息或取消该任务，详情请参见[查看或取消计划任务](#)。

6. 在支付页面中，确认订单信息，单击订购。**说明**

- 规格变更过程中，每个连接地址都会有不超过30秒的连接闪断，建议您在业务低谷期执行变更，并确保应用具备自动重连机制。
- 规格变更的耗时主要与集群计算节点数量有关（约5分钟/计算节点。例如，包含2个计算节点的集群，规格变更预计10分钟左右完成）。此外，规格变更的耗时还与数据库负载、库表数量等因素有关。

相关API

API	描述
ModifyDBNodeClass	变更PolarDB集群节点规格。
ModifyDBNodesClass	独立变更PolarDB集群单个节点的规格。

9.3. 自动变配（弹性扩缩容）

9.3.1. 集群规格自动变配（自动扩容）

您可以在PolarDB MySQL引擎集群的**基本信息**页面设置自动扩缩容参数，PolarDB MySQL引擎集群将根据您设置的参数自动扩容和自动回缩。本文档介绍了在PolarDB MySQL引擎集群的**基本信息**页面设置自动扩缩容参数的操作步骤。

使用限制

- 产品系列需为PolarDB MySQL引擎集群版和历史库集群版。单节点和历史库单节点版均不支持该功能。更多关于3个系列的介绍，请参见[产品系列](#)。
- PolarDB集群没有正在进行的配置变更。
- 按量付费和包年包月的PolarDB MySQL引擎集群都支持自动扩容和自动回缩。

费用说明

具体的费用说明，请参见[变更配置费用说明](#)。

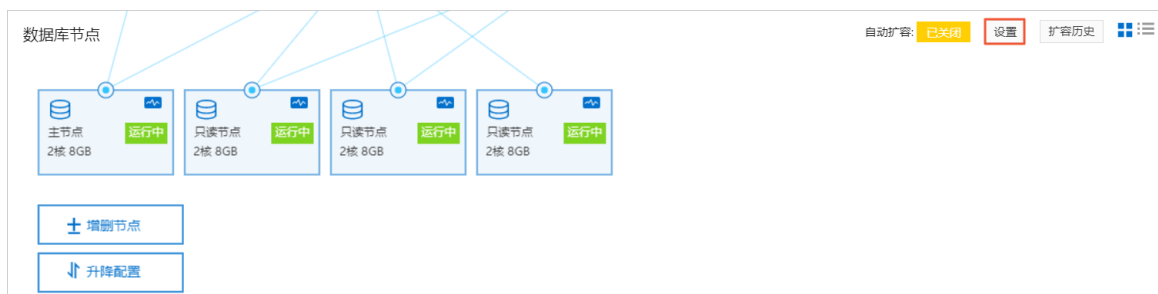
若您的业务量波动较大且频繁，推荐您购买[PolarDB计算包](#)并配合PolarDB MySQL引擎的自动扩缩容功能一起使用，当集群配置发生调整时，计算包能根据当前规格自动进行抵扣。

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。
- 规格变更过程中，可能会出现一次约30秒的闪断，建议您在业务低谷期执行变更，并确保应用具备自动重连机制。
- PolarDB集群变更配置期间，只读请求相比读写请求的滞后时间，可能比正常运行状态的滞后时间更长。

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. 在集群列表页，单击目标集群ID。
4. 在基本信息页的数据库节点区域，单击右上角的设置。



5. 在弹出的对话框中，设置以下参数：

参数名称	参数说明
自动扩容	打开自动扩容的开关。
观测窗口	在整个观测窗口期内，若CPU平均使用率大于等于设定值，则在观测窗口结束后，PolarDB将根据集群的实时读写流量情况选择增加节点或升配的扩容方式。例如观测窗口期是5分钟，自动扩容的时间是10分钟，所以您需要等待15分钟左右才能看到自动扩容的效果。
CPU平均使用率	自动扩容的触发阈值。当CPU平均使用率大于或等于设置的值时，就会触发自动扩容。
扩容规格上限	自动扩容的规格上限。触发自动扩容后，会逐级扩容PolarDB的集群规格，例如从4核到8核再到16核，直到升级至扩容规格的上限。
只读节点数量上限	自动扩容只读节点的数量上限。触发自动扩容后，会逐个增加PolarDB的只读节点，直到增加至上限。 说明 自动扩容的节点会加入集群默认地址，自定义地址需要配置新节点自动加入。关于如何配置新节点自动加入，详情请参见配置数据库代理。

参数名称	参数说明
自动回缩	打开自动回缩的开关。  说明 开启自动回缩后，在静默期内如果CPU平均使用率小于30%的时间超过静默期时间的99%，则在静默期结束后，将触发自动逐级回缩PolarDB集群规格，直到回缩至原始规格。
静默期	两次自动扩容或自动回缩的最小间隔时间。在静默期内，PolarDB会持续进行观测，但不会触发自动扩容或缩容。若在静默期与观测窗口同时结束，且该观测窗口内CPU使用率达到调整阈值，PolarDB会在静默期和观测窗口同时结束时触发自动规格调整。

6. 单击**确定**。

9.3.2. 结合数据库自治服务DAS自动变配

PolarDB MySQL引擎集群的**一键诊断**功能融合了数据库自治服务DAS的部分功能，您可以通过其中的**自治中心**开启自动扩缩容服务。本文档介绍了开启自动扩缩容服务的操作步骤。您可以通过体验馆案例**PolarDB MySQL集群自动扩容和回缩**，体验PolarDB MySQL引擎自感知、自运维的云服务。

使用限制

- 产品系列需为PolarDB MySQL引擎集群版。单节点、历史库单节点版和历史库集群版均不支持该功能。更多关于3个系列的介绍，请参见**产品系列**。
- PolarDB集群没有正在进行的配置变更。
- 按量付费和包年包月的PolarDB MySQL引擎集群都支持自动扩容和自动回缩。

费用说明

具体的费用说明，请参见**变更配置费用说明**。

若您的业务量波动较大且频繁，推荐您购买**PolarDB计算包**并配合DAS提供的自动扩缩容功能一起使用，当集群配置发生调整时，计算包能根据当前规格自动进行抵扣。

注意事项

- 数据库自治服务DAS自动变配功能在**自动扩容**时不支持对主节点和只读节点分别变配，自动回缩后所有只读节点会跟主节点变成同一规格，请谨慎使用。

说明 当PolarDB集群中节点数量大于2个且主节点与只读节点规格不一致，并开启自动扩容时。在整个观测窗口内，如果CPU平均利用率大于或等于设定值时，PolarDB会根据集群的实时读写流量情况选择以下几种可能的扩容方式：

- 增加与主节点同一规格的只读节点。
- 增加与集群中某个只读节点同一规格的只读节点。

说明 该扩容方式适用于只读节点的CPU平均利用率大于或等于设定值时的场景。

- 与主节点升级到同一规格。

开启自动回缩后，在静默期内如果CPU平均使用率小于30%的时间超过静默期时间的99%，则在静默期结束后，将会触发自动逐级回缩PolarDB集群规格，有以下几种可能的回缩方式：

- 删除增加的只读节点。

说明 未被删除的只读节点规格保持不变。

- 所有只读节点会跟主节点变成同一规格。

- 集群规格的升降级不会对集群中已有数据造成任何影响。
- 规格变更过程中，可能会出现一次约30秒的闪断，建议您在业务低谷期执行变更，并确保应用具备自动重连机制。
- PolarDB集群变更配置期间，只读请求相比读写请求的滞后时间，可能比正常运行状态的滞后时间更长。

操作步骤

- 登录PolarDB控制台。
- 在控制台左上角，选择集群所在地域。
- 在集群列表页，单击目标集群ID。
- 在左侧导航栏中，选择**诊断与优化 > 一键诊断**。
- 单击**自治中心**页签。
- 在**自治中心**区域，单击**自治功能开关**。



- 在设置对话框中的**自治功能开关**页签下，先开启**自治服务**，然后再陆续打开**自动扩容**和**自动回缩**。

说明

- 打开自动扩容后，在整个观测窗口内如果CPU平均利用率大于或等于设定值，则在观测窗口结束后，PolarDB将根据集群的实时读写流量情况选择增加节点或升配的扩容方式。例如观测窗口期是5分钟，自动扩容的时间是10分钟，所以您需要等待15分钟左右才能看到自动扩容的效果。
- 打开自动回缩后，在静默期内如果CPU平均使用率小于30%的时间超过静默期时间的99%，则在静默期结束后，将触发自动逐级回缩PolarDB集群规格，直到回缩至原始规格。

设置

自治功能开关

事件订阅设置

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

开启自动索引创建

自动限流

自动扩容

CPU平均利用率 >= 80 %

规格上限 = 88核710G

只读节点数量上限 = 3 个

观测窗口 = 5分钟

在观测窗口内，CPU利用率 > 70%的时间超过50%

自动回缩

静默期 = 30分钟

在观测窗口内，CPU利用率 < 30%的时间超过99%，进行自动回缩

确定

取消

8. 设置自动扩容和自动回缩参数，单击确定。

参数名称	参数说明
CPU平均利用率	自动扩容的触发阈值。当CPU平均使用率大于或等于设置的值时，就会触发自动扩容。
规格上限	自动扩容的规格上限。触发自动扩容后，会逐级扩容PolarDB的集群规格，例如从4核到8核再到16核，直到升级至扩容规格的上限。
只读节点数量上限	自动扩容只读节点的数量上限。触发自动扩容后，会逐个增加PolarDB的只读节点，直到增加至上限。 说明 自动扩容的节点会加入集群默认地址，自定义地址需要配置新节点自动加入。关于如何配置新节点自动加入，详情请参见配置数据库代理。

参数名称	参数说明
观测窗口	在整个观察窗口期内，若CPU平均使用率大于等于设定值，则在观测窗口结束后，PolarDB将根据集群的实时读写流量情况选择增加节点或升配的扩容方式。例如观测窗口期是5分钟，自动扩容的时间是10分钟，所以您需要等待15分钟左右才能看到自动扩容的效果。
静默期	两次自动扩容或自动回缩的最小间隔时间。在静默期内，PolarDB会持续进行观测，但不会触发自动扩容或缩容。若在静默期与观测窗口同时结束，且该观测窗口内CPU使用率达到调整阈值，PolarDB会在静默期和观测窗口同时结束时触发自动规格调整。

9.3.3. 本地资源自动变配（秒级扩核）

PolarDB MySQL提供本地资源自动弹性伸缩功能，当观测窗口内数据库实例的CPU平均使用率达到设定的阈值时，会在原规格的基础上，将数据库实例的CPU核数按照设定值进行扩容。

前提条件

- 目前仅支持PolarDB MySQL集群版通用规格的集群支持该功能。
- 请前往DAS控制台创建[DAS服务关联角色](#)。

注意事项

- 此功能当前可免费试用，暂不收费。
- 开启本地资源自动弹性伸缩后，建议关闭PolarDB的集群规格自动扩容功能，因为两者是冲突的。如果首先触发了本地资源自动弹性伸缩，再触发集群规格自动扩容时，本地资源自动弹性伸缩会失效。

与集群规格自动变配的对比

本地资源自动变配功能，与[集群规格自动变配（自动扩容）](#)相比，有如下区别：

对比项	本地资源自动变配	集群规格自动变配
扩容内容	仅能扩容CPU、IOPS，无法扩容内存、连接数等。	可以同时扩容CPU、IOPS、内存、连接数等。
扩容速度	执行速度快： ● 扩容观察窗口最短为30秒。 ● 触发扩容后执行完成可达到秒级。	执行速度较慢： ● 扩容观察窗口最短为5分钟。 ● 涉及规格扩展，速度较慢。
扩容过程	不涉及规格切换，扩容过程中无闪断。	需要规格切换，有闪断过程。

通过以上对比可以得出，本地资源自动变配相比集群规格自动变配，有如下明显优势：

- 扩容速度快，可实现秒级扩容。
- 扩容过程中无闪断。

操作步骤

1. 登录[DAS控制台](#)。
2. 在左侧导航栏单击[管理和设置](#)。

3. 在自动弹性伸缩策略列表区域右侧单击**新增策略**。

4. 在**新增策略**页面设置如下参数：

参数	说明
策略名称	策略的名称。
模式	策略的模式，请选择 本地资源自动弹性伸缩 。
引擎类型	数据库引擎的类型，目前只支持PolarDB MySQL。
规格类型	所选数据库引擎的规格，目前只支持共享型，即通用规格。
CPU平均利用率	自动扩容的触发阈值。当CPU平均使用率大于或等于设置的值时，就会触发自动扩容。
扩容观测窗口	实例性能的观测时间。系统会在选定的观测期间周期性检测实例的CPU使用率，当CPU平均使用率达到触发阈值时，则会触发自动扩容。
CPU扩容步长	CPU每次扩容增加的核数，取值为2~31之间的整数。 例如，若原规格的CPU核数为4核，当设置CPU扩容步长为2时，触发后会将CPU核数扩容为6核；如果扩容完成后，CPU平均使用率依旧满足自动扩容的触发条件，则会继续逐级扩容，直到达到扩容的上限。  说明 单个数据库实例的CPU扩容上限为4×原始CPU核数，最大为32核。 例如，若当前CPU为4核，则最多会扩容到16核。
缩容观测窗口	实例性能的缩容观测时间。系统会在选定的回缩观测期间周期性检测实例的CPU使用率，当CPU使用率低于30%的时间占比超过99%，则会触发自动性能回缩。 每次缩容操作，会按照扩容步长设置的CPU核数，进行缩容，直至回缩到原有规格。

新增策略

* 策略名称

秒级扩核

* 模式

本地资源自动弹性伸缩

* 引擎类型

PolarDB MySQL

* 规格类型

共享型

CPU平均利用率 >=

40

%

扩容观测窗口

30秒

CPU扩容步长

2

核

(请输入2~31之间的整数,但是单个实例的vCPU上限为Min [32核, 4*原始vCPU],例如当前vCPU 4核,CPU扩容步长为2,仅会扩容16核CPU)

缩容观测窗口

1分钟

5. 单击确定。
6. 返回到策略列表后，在对应策略操作列的应用。
7. 在应用策略页，勾选需要应用此策略的PolarDB MySQL实例，并单击。

应用策略

* 已选择策略

秒级扩容

选择应用的实例

实例ID/实例别名

pc-

实例ID/实例别名

pc-

(通用规格)

>

<

1 项

1 项

确认

取消

> 文档版本：20220713

252

8. 单击**确认**，为选中的数据库实例应用此自动弹性伸缩策略。

查看本地资源自动弹性伸缩的结果

1. 在DAS控制台首页左侧导航栏单击**实例监控**。
2. 找到目标实例，单击实例ID，进入目标实例详情页。
3. 在左侧导航栏中，单击**自治中心**。
4. 在自治中心页，查看选定时间范围发生的**弹性伸缩事件**。
5. 单击**弹性伸缩事件的详情**，查看自动弹性伸缩的详细信息。



9.4. 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和临时升配订单。
- 产品系列需为PolarDB MySQL引擎集群版和历史库集群版。单节点和历史库单节点版均不支持该功能。更多关于产品系列的介绍，请参见[产品系列](#)。

背景介绍

临时升配是指临时升级规格，您可以对整个集群进行规格升级，也可对集群中的单个节点进行规格升级。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 不支持临时降配，如需降级请参见[手动变配](#)。

注意事项

- 临时升配过程和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。
- 临时升配期间不支持普通的[手动变配](#)、自动变配、[增加或删除节点](#)。
- 临时升配的最短时间为1小时，建议升配时间最长不超过14天。
- 临时升配后如果性能不够或需要延长还原时间，可以在还原时间到达之前最多再进行1次升配，此次设置的还原时间不能早于第1次。

主节点和只读节点可单独变配规格

PolarDB集群支持对主节点和只读节点分别临时升级规格，即只读节点的规格可以与主节点规格不一致。

使用限制

- 同一集群中，确保至少有一个只读节点与主节点规格保持一致。
- 只读节点规格建议不要与主节点规格相差过大：
 - 只读节点的内存不能小于主节点内存的一半。

- 只读节点的CPU规格与主节点CPU规格对应规则如下：

主节点CPU规格（core）	允许只读节点最小CPU规格（core）
2	2
4	4
8	4
16	8
32	16
64	32
88	32

- 开启**热备**后的只读节点需要与主节点规格保持一致。
- 只读**列存节点**场景下建议只读列存节点规格比主节点规格更高。

 **说明** PolarDB多主架构集群版不受以上使用限制。

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 × 1.5 × N。

若您的业务量波动较大且频繁，推荐您购买**PolarDB计算包**并配合DAS提供的**自动扩、缩容**功能一起使用，当集群配置发生调整时，计算包能根据当前规格自动进行抵扣。

 **说明** 若您使用的PolarDB MySQL引擎集群购买方式为包年包月，则需要将其转为按量付费后才能使用计算包进行抵扣，详情请参见**包年包月转按量付费**。

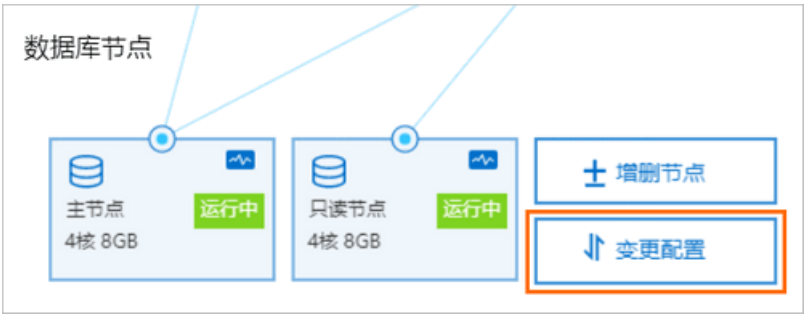
操作步骤

- 登录**PolarDB控制台**。
- 在控制台左上角，选择集群所在地域。
- 在**集群列表**页，找到目标集群。
- 您可以选择如下两种方式中的任意一种进入**变更配置（包年包月）**页面：
 - 单击目标集群操作栏中的**变更配置**。



- 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**变更配置**。



5. 在**变更配置（包年包月）**页面，选中**临时升配**，单击**确定**。

 **说明** 仅包年包月集群支持临时升配。

6. 在弹出的对话框中，设置如下参数。

参数	说明
节点	为当前节点选择升级后的目标节点规格。  说明 确保一个只读节点与主节点规格保持一致，其他节点规格均可选择配置。
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1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. 选中服务协议，单击**立即购买**完成支付。

8. 在支付页面，确认待支付订单，单击**支付**。

9.5. 增加或删除节点

创建PolarDB集群后，您可以根据实际需求手动增加期望规格的只读节点或删除不需要的只读节点。

前提条件

- 产品系列需为PolarDB MySQL引擎集群版、多主架构集群版和历史库集群版。单节点和历史库单节点版均不支持该功能。更多关于产品系列介绍，请参见[产品系列](#)。
- PolarDB集群没有正在进行的配置变更。

主节点和只读节点可单独变配规格

您可以根据实际需求增加期望规格的只读节点，即只读节点的规格可以与主节点规格不一致。

使用限制

- 同一集群中，最多包含15个只读节点，最少包含一个只读节点（用于保障集群的高可用）。
- 新增的只读节点规格建议不要与主节点规格相差过大：
 - 新增的只读节点的内存不能小于主节点内存的一半。
 - 新增的只读节点的CPU规格与主节点CPU规格对应规则如下：

主节点CPU规格（core）	允许只读节点最小CPU规格（core）
2	2
4	4
8	4
16	8
32	16
64	32
88	32

- 同一集群中，确保至少有一个只读节点与主节点规格保持一致。
- 开启热备后的只读节点需要与主节点规格保持一致。
- 只读列存节点场景下建议只读列存节点规格比主节点规格更高。

 说明 PolarDB多主架构集群版不受以上使用限制。

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（也称预付费），则增加的节点也是包年包月。
- 如果集群为按量付费（也称后付费或按小时付费），则增加的节点也是按量付费。
- 若您的业务量波动较大且频繁，推荐您购买PolarDB计算包并配合DAS提供的自动扩、缩容功能一起使用，当集群配置发生调整时，计算包能根据当前规格自动进行抵扣。

 说明

- 包年包月和按量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（详情请参见计费项概览），存储费用仍然按实际使用量收费，与节点数量无关。
- 若您使用的PolarDB MySQL引擎集群计费方式为包年包月，则需要将其转为按量付费后才能使用计算包进行抵扣，详情请参见包年包月转按量付费。

节点数量对性能的影响

详情请参见性能测试（OLTP）。

注意事项

- PolarDB MySQL引擎**集群版**和历史库集群版支持同时增加多个只读节点，最多可增加至15个只读节点。
- 仅PolarDB MySQL引擎集群支持同时删除多个只读节点，但集群中必须保留至少一个只读节点，以保障集群的高可用。
- 增加一个只读节点预计耗时5分钟，具体耗时受新增节点数量、库表数量、数据库负载等因素影响。增加节点的过程中，对数据库无任何影响。
- 删除只读节点时，该节点上的连接会发生闪断，其他节点不受影响。建议您在业务低谷期执行删除节点操作，并确保应用具备自动重连机制。如果您的应用连接的是集群地址，PolarDB可自动屏蔽删除的节点，不需要修改应用配置。

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，您可以重启应用断开该读写分离连接并重新建立连接，请求将转发到新增的只读节点。

1. 登录**PolarDB控制台**。
2. 在控制台左上角，选择集群所在地域。
3. 您可以按照如下两种方式中的任意一种进入**增删节点向导**页面：
 - 通过**集群列表**页进入**增删节点向导**：找到目标集群，单击操作栏的**增删节点**。

集群ID/名称	运行状态	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 5.7	2	8核 64GB	2.36 GB	包年包月 2020年9月28日 00:00:00 到期		升降配 增删节点 更多

- 通过目标集群**基本信息**页进入**增删节点向导**：
 - a. 找到目标集群，单击集群ID，进入**集群基本信息**页。
 - b. 在**数据库节点**区域，单击图标切换视图。
 - c. 单击**增删节点**。

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

4. 选中**增加节点**并单击**确定**。

增删节点向导

阿里云数据库专家推荐：包年包月计费不灵活？添加节点只用几天却要付一年的钱？推荐使用 [计算包](#)，随时弹，都划算！

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](#)

确定

取消

- 单击**+**增加一个只读节点，选择**切换时间**并选中服务协议，单击去支付。

说明

- 若需要为您的PolarDB MySQL引擎集群同时增加多个节点，请重复单击**+**增加一个只读节点进行添加。
- 您可以选择**切换时间**为**立即切换**或**定时切换**。若选择**定时切换**，可选的时间范围为未来24小时内的任意时间点，增加节点的任务将在选定时间点后的30分钟内完成。您还可以在**计划任务**页查看该任务的具体信息或取消该任务，详情请参见[查看或取消计划任务](#)。

- 在支付页面确认待支付订单，单击**订购**。

删除只读节点

- 登录[PolarDB控制台](#)。
- 在控制台左上角，选择集群所在地域。
- 您可以按照如下两种方式中的任意一种进入**增删节点向导**页面：
 - 通过**集群列表**页进入**增删节点向导**：找到目标集群，单击操作栏的**增删节点**。

集群ID/名称	运行状态	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-xxxxx	运行中	100%兼容 MySQL 5.7	2	8核 64GB	2.36 GB	包年包月 2020年9月28日 00:00:00 到期		升降配 增删节点 更多

- 通过目标集群**基本信息**页进入**增删节点向导**：
 - 找到目标集群，单击**集群ID**，进入**集群基本信息**页。
 - 在**数据库节点**区域，单击图标切换视图。
 - 单击**增删节点**。

数据库节点							自动扩容	已关闭	设置	扩容历史	更多
节点名称	可用区	状态	当前角色	规格	最大IOPS	Follower 优先级					
pi-xxxxxx		运行中	主节点	4核 16GB	32000	1					
pi-xxxxxx		运行中	只读节点	4核 16GB	32000	1					

4. 选中删除节点并单击确定。

增删节点向导

阿里云数据库专家推荐：包年包月计费不灵活？添加节点只用几天却要付一年的钱？推荐使用 [计算包](#)，随时弹，都划算！

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](#)

确定

取消

5. 单击目标节点名称前的



图标进行删除。

说明 仅PolarDB MySQL引擎集群支持同时删除多个只读节点，但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. 选中服务协议，单击立即购买。

说明 节点被删除后会停止计费并进行退款，详情请参见[退款或停止计费](#)。

相关API

API	描述
CreateDBNodes	增加PolarDB集群节点。
ModifyDBNodesClass	独立变更PolarDB集群单个节点的规格。
ModifyDBNodeClass	变更PolarDB集群节点规格。
RestartDBNode	重启PolarDB集群节点。
DeleteDBNodes	删除PolarDB集群节点。

9.6. 历史库单节点版升级为历史库集群版

历史库支持历史库单节点版和历史库集群版两种类型。如果您需要将历史库单节点版升级为历史库集群版，请参考本文档进行操作。

前提条件

使用该功能之前，请先[提交工单](#)联系技术支持进行数据处理。

背景信息

当前已不支持购买历史库单节点版，但存量的历史库单节点版可继续使用。您可以将存量的历史库单节点版升级为历史库集群版独享规格。关于历史库集群版的详细说明，请参见[历史库](#)。

注意事项

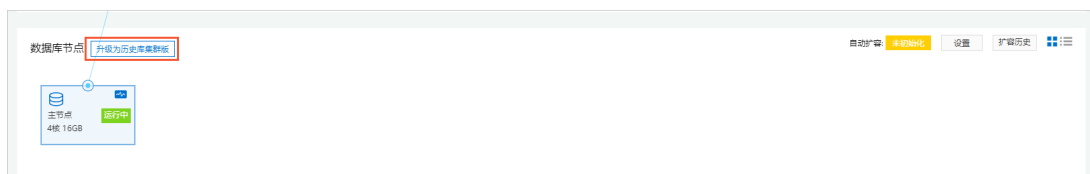
- 历史库单节点版升级为历史库集群版，系统会自动重启节点。一般情况下，重启需要3~5分钟的时间，具体耗时取决于升级前积攒的Redo日志量。极端情况下，重启耗时可能达到半小时。在此过程中，您的业务将会不可用，请您谨慎操作，建议在业务低峰期进行。
- 升级成功后，集群中的所有节点会继续使用主地址。如果您需要使用集群地址，需要自行设置。
- 升级为历史库集群版后，无法回退至历史库单节点版。若有需要，可再增加节点。此时增加节点不会导致节点重启。增删节点的操作以及注意事项与集群版一致，具体请参见[增加或删除节点](#)。

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. 您可以按照如下两种方式中的任意一种进入历史库变配对话框：
 - 方法一：
在**集群列表**页，找到目标集群，单击右侧操作栏中的**升级为历史库集群版**。



- 方法二：
 - a. 在**集群列表页**，单击目标集群ID。
 - b. 在**基本信息页**的**数据库节点区域**，单击**升级为历史库集群版**。



4. 单击+增加一个只读节点，选择切换时间并选中服务协议，单击去支付。

说明

- 若需要为您的PolarDB MySQL引擎集群同时增加多个节点，请重复单击+ 增加一个只读节点进行添加。
- 您可以选择切换时间为立即切换或定时切换。若选择定时切换，可选的时间范围为未来24小时内的任意时间点，增加节点的任务将在选定时间点后的30分钟内完成。您还可以在计划任务页查看该任务的具体信息或取消该任务，详情请参见[查看或取消计划任务](#)。

5. 在支付页面确认待支付订单，单击**订购**。

9.7. 手工扩容

存储类型按空间计费（包年包月）的集群可以通过手工扩容的方式来扩展存储空间容量。本文档介绍了手工扩容的操作步骤。

背景信息

如果预购的存储空间被占满，系统则会禁止新数据的写入。建议您及时进行手工扩容。

注意事项

仅存储类型按空间计费（包年包月）的集群支持手工扩容，按容量计费（按量付费）的存储空间采用Serverless方式，购买时无需选择容量，随着数据增长而在线自动扩容。

操作步骤

- 登录[PolarDB控制台](#)。
- 在控制台左上角，选择集群所在地域。
- 找到目标集群，单击集群ID。
- 在基本信息页面的数据库分布式存储区域，单击**手工扩容**。



- 在变配页面中，选择所需的存储空间和切换时间并选中服务协议，单击**立即购买**。

说明 您可以选择切换时间为立即切换或定时切换。若选择定时切换，可选的时间范围为未来24小时内的任意时间点，升级配置的任务将在选定时间点后的30分钟内完成。您还可以在计划任务页查看该任务的具体信息或取消该任务，详情请参见[查看或取消计划任务](#)。

- 在支付页面，确认待支付订单，单击**支付**。

10. 备份与恢复

10.1. 概述

数据是企业的核心资产。随着业务发展，企业数据呈现出规模化、爆炸式的增长，业务应用要求实时、在线的快速处理。对于数据库运维人员来说，保护企业核心数据的任务越来越有挑战性，例如数据误删除、相关系统漏洞和勒索病毒、硬件故障，甚至自然灾害都可能造成数据的丢失。因此，备份和恢复是数据库非常重要的功能。

PolarDB支持数据备份和物理日志备份。数据备份即将某个时间点上集群的全量数据生成一个备份集（快照）；物理日志备份即记录生成备份集后的增量数据。通过一个完整的数据备份以及后续一段时间的Redo日志备份，就可以将整个PolarDB集群或特定库表恢复到任意时间点。

数据备份

数据备份按照存储位置可分为一级备份和二级备份。

备份存储位置	是否默认开启	保留时长	特点	查看备份大小																		
一级备份	是	3~14天。	- 一级备份采用ROW（Redirect-on-Write）快照的方式，直接保存在PolarDB分布式存储系统上。每次保存时，一级备份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。 - PolarDB集群备份和恢复功能均采用多线程并行处理，并通过其它技术创新，10分钟内即可完成从备份集（快照）恢复到一个新的集群。 ❓ 说明 一级备份默认开启，无法关闭。	一级备份（快照）总大小如下图所示。 一级备份（快照）总大小 386.00 MB 其中冗余数据为20%当前存储容量，大约为 1.76 GB 备份功能FAQ <table><tr><th>备份ID</th><th>备份开始时间</th><th>备份结束时间</th><th>状态</th><th>一级备份时点</th><th>备份方法</th><th>备份类型</th><th>大小</th><th>存储位置</th></tr><tr><td>1</td><td>2021年3月28日 15:10:05</td><td>2021年3月28日 15:10:15</td><td>备份完成</td><td>2021年3月28日 15:10:06</td><td>物理备份</td><td>全量备份</td><td>3.4 GB</td><td>一级备份</td></tr></table> ❓ 说明 PolarDB集群一级备份（快照）总大小是所有一级备份独占的物理空间之和（即上图中①所示），而非逻辑数据大小之和（即上图中②所示），PolarDB集群的数据与多个一级备份（快照）会复用相同的物理数据块，在计时时只会计算一次。更多关于备份的问题，请参见常见问题。	备份ID	备份开始时间	备份结束时间	状态	一级备份时点	备份方法	备份类型	大小	存储位置	1	2021年3月28日 15:10:05	2021年3月28日 15:10:15	备份完成	2021年3月28日 15:10:06	物理备份	全量备份	3.4 GB	一级备份
备份ID	备份开始时间	备份结束时间	状态	一级备份时点	备份方法	备份类型	大小	存储位置														
1	2021年3月28日 15:10:05	2021年3月28日 15:10:15	备份完成	2021年3月28日 15:10:06	物理备份	全量备份	3.4 GB	一级备份														

备份存储位置	是否默认开启	保留时长	特点	查看备份大小																		
二级备份	否	30~7300 天。 - 开启删除集群前永久保留来永久保存。	- 二级备份是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。保存成本较低，但使用二级备份恢复数据的速度较慢。 - 开启二级备份后，若一级备份超出您设置的保留时间，将会被自动转存为二级备份，转存速度约为150 MB/秒。 - 二级备份支持同地域备份和跨地域备份，详情请参见同地域备份和跨地域备份。  说明 - 同地域备份和跨地域备份当前处于灰度发布阶段。如有需求，请提交工单申请使用。 - 跨地域备份暂时免费使用，未来收费时间待定。 - 若一级备份未能在下一个一级备份开始转存前完成，则下一个一级备份将会被直接删除而不会被转存为二级备份。例如将PolarDB集群的一级备份的备份时间设置为每日凌晨1点，保留时间为24小时，PolarDB集群在1月1日凌晨1点生成一级备份A，2日凌晨生成一级备份B，备份A在2号凌晨1点超过保留时间并开始转存为二级备份，由于该备份文件较大转存时间较长，到3号凌晨1点时该转存任务仍未完成，则此时备份B在3号凌晨1点到期后将会被直接删除而不会被转存为二级备份。	二级备份大小如下图所示，二级备份总大小即为每个二级备份文件大小之和。 <table><tr><th>备份集ID</th><th>备份开始时间</th><th>备份结束时间</th><th>状态</th><th>一致性快照时间戳</th><th>备份方法</th><th>备份类型</th><th>大小</th><th>存储位置</th></tr><tr><td>1</td><td>2021年03月24日 15:10:19</td><td>2021年03月24日 15:10:29</td><td>备份完成</td><td>2021年03月24日 15:10:22</td><td>快速备份</td><td>全量备份</td><td>424.20 MB</td><td>二级备份</td></tr></table>	备份集ID	备份开始时间	备份结束时间	状态	一致性快照时间戳	备份方法	备份类型	大小	存储位置	1	2021年03月24日 15:10:19	2021年03月24日 15:10:29	备份完成	2021年03月24日 15:10:22	快速备份	全量备份	424.20 MB	二级备份
备份集ID	备份开始时间	备份结束时间	状态	一致性快照时间戳	备份方法	备份类型	大小	存储位置														
1	2021年03月24日 15:10:19	2021年03月24日 15:10:29	备份完成	2021年03月24日 15:10:22	快速备份	全量备份	424.20 MB	二级备份														

物理日志备份	是	保留时长	特点	查看备份大小
备份存储位置	默认			
特点				
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。日志备份方式包括同地域备份和跨地域备份，最短保留时间为3天，最长保留时间为7300天。您可以通过开启删除集群前永久保留功能永久保存。				

说明

- 日志备份默认开启同地域备份，无法关闭。
- 跨地域备份当前处于灰度发布阶段。如有需求，请[提交工单](#)申请试用。

借助日志备份可以实现任意时间点的一致性备份：通过一个完整的数据全量备份（快照）以及后续一段时间的日志备份，就可以将PolarDB集群恢复到任意时间点（Point-In-Time Recovery，简称PITR），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。恢复到任意时间点时，应用Redo日志的恢复速度大概是20~70秒/GB，整个恢复时间是备份集（快照）恢复时间以及应用Redo日志恢复时间之和。

查看备份大小

日志备份大小如下图所示，日志备份总大小即为每个日志备份文件大小之和。

数据备份列表	物理日志备份列表	备份策略设置	有关数据恢复，请查看“不同场景下恢复方式概述”
2022年4月30日	2022年5月30日		
日志ID	日志名	日志大小	日志开始/结束时间
		1.00 GB	2022年5月30日 10:52:43

同地域备份和跨地域备份

备份说明

备份类型	说明	是否默认开启	使用场景	优势
同地域备份	备份存储在同一个地域的不同可用区。	是。 说明 开启二级备份时，默认开启同地域备份。	长期归档。	按需设置更低的备份频率进行转储，实现更低的成本。
跨地域备份	备份存储在除当前地域外的其他地域。	否，需要手动开启。	异地备份、等保三级。	低RPO，适用于安全加密的非公网环境，按需设置低频转储，实现更低的成本。

说明

低频二级备份：即二级备份的备份周期设置为比一级备份的备份周期更低的频率。

支持备份地域

地域	同地域备份	跨地域备份
西南1（成都）	支持	不支持。

地域	同地域备份	跨地域备份
华北2（北京）	支持	支持 华北2（北京） 跨地域备份至 华东2（上海）。
华东2（上海）	支持	支持 华东2（上海） 跨地域备份至 华北2（北京）。
华东2（上海）金融云	支持	支持 华东2（上海）金融云 跨地域备份至 华南1（深圳）金融云。
华南1（深圳）金融云	支持	支持 华南1（深圳）金融云 跨地域备份至 华东2（上海）金融云。

10.2. 备份和恢复费用说明

本文介绍PolarDB备份和恢复功能的费用相关说明。

费用说明

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。根据备份类型，备份所占的空间分为一级备份、二级备份和日志备份。三者的价格有所差异：

- 一级备份和日志备份的存储空间有一定的免费额度。当超过该免费额度时，PolarDB将根据超过的存储容量和保存时长收取一定费用。
- 二级备份（同地域备份）没有免费额度，PolarDB会根据存储容量和保存时长收取一定费用。

 **说明** 跨地域备份暂时免费使用，未来收费时间待定。

具体价格如下：

价格表（单位：元/GB/小时）

地域	一级备份	二级备份	日志备份
中国内地（大陆）	0.003	0.00021	0.00021
中国香港及海外	0.0042	0.000294	0.000294

 **说明** 超出免费额度时的存储空间费用，除正常按量付费方式，其中的一级备份费用推荐使用存储包进行抵扣，更划算。更多详情，请参见[备份存储（超出免费额度）计费规则](#)。

快照备份后，如果您对数据进行了修改，会导致快照备份变大，造成备份费用增长。另外，碎片整理涉及到数据修改，也会导致快照备份增长。

例如：假设数据库数据量为100 GB，在进行快照备份后，修改了数据库中的10 GB数据，则：

- 数据存储按照100 GB收费，快照备份存储按照10 GB收费；
- 如果删除了该数据库，且保留其快照备份，则快照备份存储按照100 GB收费。

如何减少备份费用

当前PolarDB不支持将备份文件下载至本地。若需要减少备份费用，可通过以下几种方式：

- 选择适当的备份频率：您可在[备份策略设置](#)中，选择适当的数据备份频率。

 说明 可以设置低频二级备份，减少备份费用。

- 缩短一级备份保留的时间：您可在**备份策略设置**中，适当缩短**数据备份保留**中一级备份的保留时长。
- 购买存储包抵扣一级备份的费用：一级备份的费用可使用存储包进行抵扣，更划算。更多详情，请参见[备份存储（超出免费额度）计费规则](#)。

10.3. 备份操作说明

10.3.1. 备份设置

PolarDB支持数据备份和Redo物理日志备份。数据备份即将某个时间点上集群的全量数据生成一个备份集（快照）；Redo物理日志备份即记录生成备份集后的增量数据。您可设置数据备份和Redo物理日志备份的备份策略，如备份方式，数据备份文件保存时长、存储位置，日志备份文件保存时长等。

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. 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**备份策略设置**页签，单击**编辑**。
6. 在**备份策略设置**对话框中，设置**数据备份策略**、**日志备份策略**、**通用备份策略**的相关参数。

备份策略设置

数据备份

使用存储层的快照技术对数据库进行无锁备份

* 一级备份

一级备份结构图

备份方式

☒ 常规备份 (按备份周期)

☐ 高频备份

备份周期

☒ 周一

☒ 周二

☒ 周三

☒ 周四

☒ 周五

☒ 周六

☒ 周日

备份开始时间

18:00 - 19:00

备份保留时长

7

天

3天

14天

* 二级备份

二级备份结构图

备份开关

☒

开启二级备份, 打开会产生额外的费用, 删除备份文件可节省成本

备份方式

☒ 同地域备份

☐ 跨地域备份

备份周期

☒ 周一

☒ 周二

☒ 周三

☒ 周四

☒ 周五

☐ 周六

☐ 周日

同地域备份保留

30

天

30天

7300天

☐ 删除集群前永久保留

日志备份

离线保存每一个数据库Redo日志文件

备份方式

☒ 同地域备份

☐ 跨地域备份

确定

取消

7. 单击确定。

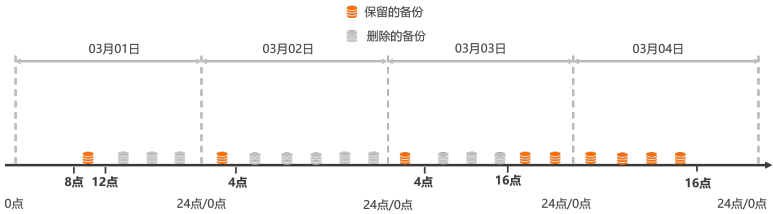
数据备份策略

数据备份策略包括一级备份策略和二级备份策略。

备份类型	参数	说明
------	----	----

267

> 文档版本：20220713

备份类型	参数	说明
一级备份	备份方式	自动备份的频率，可选择常规备份或高频备份。 ● 常规备份（按备份周期）：自动备份默认每日1次，你可以设置数据自动备份的周期和自动备份开始时间。 说明 ○ 出于安全考虑，常规备份的频率为每周至少2次。 ○ 备份文件不可删除。 ● 高频备份：PolarDB支持增强保护功能，能够缩短备份周期，增加备份密度，从而提升恢复速度。 24小时内可以自动执行多次备份。您可以根据业务选择开启每2、3或4小时进行一次增强备份 最近24小时增强保护功能开启后，备份完成时间在24小时内的备份会全部保留。超过24小时的，系统将仅保留每日00:00点后完成的第一个备份，其他均被删除。 假设您在3月1日8:00设置了每4小时创建一个备份，那么系统将自动在2小时内（即3月1日8:00~12:00）完成一个备份，并以每4小时一个备份的频率持续创建。 假设当前时间为3月4日16:00，系统将保留如下备份： ○ 最近24小时内（3月3日16:00~3月4日16:00）完成的备份。 ○ 3月3日0:00~4:00间完成的备份。 ○ 3月2日0:00~4:00间完成的备份。 ○ 3月1日8:00~12:00间完成的备份  那么4小时后，即3月4日20:00，系统将保留如下备份： ○ 最近24小时内（3月3日20:00~3月4日20:00）完成的备份。 ○ 3月3日0:00~4:00间完成的备份。 ○ 3月2日0:00~4:00间完成的备份。 ○ 3月1日8:00~12:00间完成的备份 

备份类型	参数	说明
	备份周期	设置数据自动备份的周期。  说明 仅当选择常规备份时，需要配置该参数。
	备份开始时间	设置数据自动备份的开始时间。  说明 仅当选择常规备份时，需要配置该参数。
	备份频率	设置备份频率，目前支持最近24小时，每2小时备份一次、最近24小时，每3小时备份一次或最近24小时，每4小时备份一次。  说明 仅当选择高频备份时，需要配置该参数。
	备份保留时长	设置一级备份保留时间。  说明 - 一级备份默认开启且默认保留时间为7天。 - 最短保留时间为3天，最长保留时间为14天。
	备份开关	开启或关闭二级备份。  说明 - 二级备份默认关闭。开启二级备份会产生额外的费用，删除备份文件可节省成本。二级备份费用详情，请参见备份存储（超出免费额度）计费规则。 - 低频转储规则：开启低频二级备份后，备份周期选择某一天时，则会选择当天最早的一个备份集进行转储。 - 开启低频二级备份后，对于后续产生的备份集，只有满足低频转储规则时，才能进行转储。
	备份方式	可选择同地域备份或跨地域备份。  说明 - 默认开启同地域备份。 - 如果同时开启同地域备份和跨地域备份，则会同时转储两份备份数据。

备份策略	参数	说明
	备份周期	设置二级备份的备份周期。  说明 仅一级备份选择的日期可选。
	同地域备份保留	设置同地域备份保留时间。  说明 - 最短保留时间为30天，最长保留时间为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。
	跨地域备份地域	选择需要跨地域备份的地域。
	跨地域备份保留	设置跨地域备份保留时间。  说明 - 最短保留时间为30天，最长保留时间为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。

日志备份策略

日志备份策略包括Redo物理日志的保留时长。

参数	说明
备份方式	可选择同地域备份或跨地域备份。  说明 - 同地域日志备份默认开启，无法关闭。 - 仅当二级备份开启跨地域备份时，日志备份才可选择跨地域备份。

参数	说明
同地域备份保留	设置同地域日志备份的保留时长。  说明 - 最短保留时间为3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。
跨地域备份地域	与二级备份中的跨地域备份地域一致，无需选择。
跨地域备份保留	设置跨地域日志备份的保留时长。  说明 - 最短保留时间为3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。

通用备份策略

此外，您还可以设置删除集群时的备份文件保留策略。

参数	说明
集群备份保留	设置删除集群时的备份保留策略。 - 删除集群时会自动备份，并永久保留该集群的所有备份集。：删除集群时保留所有备份。 - 删除集群时会自动备份，永久保留该备份集。：删除集群时保留最后一个备份。 - 删除集群时，立即删除该集群的所有备份集。：删除集群时不保留任何备份。  说明 - 如果您选择了删除集群时会自动备份，并永久保留该集群的所有备份集。或删除集群时会自动备份，永久保留该备份集。策略，删除PolarDB集群时，系统会自动发起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，更多内容请参见恢复已释放的集群。 - 如果是跨地域备份，当选择删除集群时，立即删除该集群的所有备份集。策略时，则在删除集群时，会同时删除跨地域备份的所有备份集。如果选择其他策略，则执行设定的过期策略。

相关API

API	描述
DescribeBackupPolicy	查询PolarDB集群备份策略。
ModifyBackupPolicy	修改PolarDB集群备份策略。

10.3.2. 备份方式1：自动备份

自动备份默认开启，PolarDB会按照设置的备份策略自动备份数据，以达到定时定期备份的效果，保证数据的安全性。集群创建之初，PolarDB即会按照每日备份一次的频率，自动进行数据备份。您可以根据业务需求，在控制台配置自动备份的频率、备份保存时长等参数。

注意事项

- 自动备份的备份文件不可删除，您可以通过备份设置中的[数据备份保留](#)设置自动备份文件的保留时长。
- 快照备份后，如果您对数据进行了修改，会导致快照备份变大，造成备份费用增长。另外，碎片整理涉及到数据修改，也会导致快照备份增长。
例如：假设数据库数据量为100 GB，在进行快照备份后，修改了数据库中的10 GB数据，则：
 - 数据存储按照100 GB收费，快照备份存储按照10 GB收费；
 - 如果删除了该数据库，且保留其快照备份，则快照备份存储按照100 GB收费。

设置自动备份策略

- 登录[PolarDB控制台](#)。
- 在控制台左上角，选择集群所在地域。
- 找到目标集群，单击集群ID。
- 在左侧导航栏中，选择[配置与管理](#) > [备份恢复](#)。
- 在[备份策略设置](#)页签，单击[编辑](#)。
- 在备份策略设置对话框中，设置数据备份策略、日志备份策略、通用备份策略的相关参数，具体参数介绍，请参见[备份设置](#)。
- 完成备份设置后，单击[确定](#)。

相关API

API	描述
CreateBackup	创建PolarDB集群全量快照备份。
DescribeBackups	查询PolarDB集群备份信息。
DeleteBackup	删除PolarDB集群备份。
DescribeBackupPolicy	查询PolarDB集群自动备份策略。
ModifyBackupPolicy	修改PolarDB集群自动备份策略。

10.3.3. 备份方式2：手动备份

手动备份即主动备份，您可以根据业务需求随时进行手动备份，达到即时备份的效果，确保数据可靠性。本文介绍手动备份的相关操作步骤。

注意事项

- 每个集群最多可以有3个主动创建的备份。
- 手动备份的备份文件可以删除。
- 快照备份后，如果您对数据进行了修改，会导致快照备份变大，造成备份费用增长。另外，碎片整理涉及到数据修改，也会导致快照备份增长。
例如：假设数据库数据量为100 GB，在进行快照备份后，修改了数据库中的10 GB数据，则：
 - 数据存储按照100 GB收费，快照备份存储按照10 GB收费；
 - 如果删除了该数据库，且保留其快照备份，则快照备份存储按照100 GB收费。

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，单击集群ID。
4. 在左侧导航栏中，选择配置与管理 > 备份恢复。
5. 在数据备份列表页签，单击创建备份。
6. 在弹出的对话框中，单击确定即可。

相关API

API	描述
CreateBackup	创建PolarDB集群全量快照备份。
DescribeBackups	查询PolarDB集群备份信息。
DeleteBackup	删除PolarDB集群备份。

10.4. 恢复操作说明

10.4.1. 全量恢复方式1：从备份集恢复

全量恢复是指将PolarDB全量历史数据恢复至一个新的集群，验证新集群数据后，您可以再将恢复后的数据迁移至原集群。全量恢复支持从备份集恢复和恢复到过去时间点两种恢复方式。本文介绍如何从备份集恢复全量历史数据。

注意事项

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。

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，单击集群ID。
4. 在左侧导航栏中，选择配置与管理 > 备份恢复。

5. 恢复数据到新集群。

- 同地域恢复：
找到目标备份集，单击右侧操作栏中的恢复数据到新集群。

备份集ID	备份开始/结束时间	备份大小	状态	一致性快照时间	备份方法	是否有效	备份类型	存储位置	备份策略	操作
1-1000002	2022年6月27日 06:17:26-2022年6月27日 06:18:27	2.32 GB	备份完成	2022年6月27日 06:17:29	快照备份	有效	全量备份	一级备份	自动备份	恢复数据到新集群 删除备份
1-1000036	2022年6月26日 06:17:13-2022年6月26日 06:18:13	2.32 GB	备份完成	2022年6月26日 06:17:16	快照备份	有效	全量备份	一级备份	自动备份	恢复数据到新集群 删除备份

- 跨地域恢复：
 - 在备份恢复页面，选择需要恢复的备份集所在的地域。
 - 找到目标备份集，单击右侧操作栏中的恢复数据到新集群。

数据备份列表

物理日志备份列表

备份策略设置

有关数据恢复，请查看“不同场景下恢复方式概述”

创建备份

选择1 (杭州)

2022年6月11日

2022年6月12日

一级备份 (快照)

总大小 308.00 MB (其中免费额度为50%当前存储容量, 大约为 1.18 GB)

备份集ID	备份开始/结束时间	备份大小	状态	一致性快照时间	备份方法	是否有效	备份类型	存储位置	备份策略	操作
1-1000007	2022年6月12日 06:01:05-2022年6月12日 06:02:06	30.53 MB	备份完成	2022年6月12日 06:01:09	快照备份	有效	全量备份	二级备份	自动备份	恢复数据到新集群 删除备份

6. 在克隆实例页面，选择新集群的付费模式。

- 包年包月：在创建集群时支付计算节点的费用，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。
- 按量付费：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。

说明 PolarDB MySQL引擎已推出计算包。推荐购买按量付费的集群，因为按量付费的集群支持配合计算包使用，比包年包月付费方式更划算、更灵活。详情请参见[计费方式1：按量付费+计算包（推荐）](#)。

7. 设置以下参数。

参数	说明
操作类型	选择从备份集恢复数据。
备份集	选择需要恢复的备份集。 说明 此处展示的为各备份集的备份开始时间，您可以根据该时间确定是否需要恢复的备份集。
地域	选择恢复到的目标地域。 说明 - 如果开启跨地域备份，支持恢复到源地域和目标地域。 - 如果未开启跨地域备份，默认与源集群所在地域一致，无需选择。

参数	说明
主可用区	选择集群的主可用区。  说明 在有两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	固定为专有网络，无需选择。  说明 如需使用经典网络请先任选一个VPC专有网络，待集群创建完成后，再申请经典网络连接地址。申请经典网络连接地址的步骤，请参见申请集群地址和主地址。
VPC网络	选择集群所在的 VPC网络 和 VPC交换机 ，建议与原集群一致。
VPC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。
兼容性	默认与原集群兼容性一致，无需选择。 例如原集群兼容性为MySQL 8.0（与MySQL 8.0完全兼容），此处兼容性也固定为MySQL 8.0。
系列	默认与原集群系列一致，无需选择。 例如原集群系列为集群版（2-16个节点）【推荐】，此处系列也固定为集群版（2-16个节点）【推荐】。更多详情，请参见概述。
子系列	PolarDB MySQL引擎集群版支持通用规格和独享规格两种子系列，其中： 关于两种类型的详细对比，请参见如何选择通用规格和独享规格。  说明 仅当原集群系列为集群版（2-16个节点）【推荐】时，支持该配置。系列为单节点（入门级）或历史库单节点版时，不支持该配置。
节点规格	选择节点规格，不同规格有不同的最大存储容量和性能。更多详情，请参见计算节点规格。  说明 为了保障恢复后的集群运行正常，建议选择高于原集群的节点规格。
节点个数	◦ 若原集群系列为集群版（2-16个节点）【推荐】，节点个数默认为2，无需选择。  说明 集群版集群默认包含1主1只读两个节点，创建成功后最多可添加至1主15只读节点。关于如何添加节点，请参见增加或删除节点。 ◦ 若原集群系列为单节点（入门级）或历史库单节点版，节点个数默认为1，无需选择。

参数	说明
数据库代理类型	PolarDB支持企业通用版和企业独享版两种数据库代理类型。 ◦ 企业通用版：配套集群子系列的通用规格。它可以共享CPU物理资源，可根据业务负载，提供智能秒级资源弹性扩展能力。 ◦ 企业独享版：配套集群子系列的独享规格。它可以独占CPU物理资源，具有更好的性能稳定性。 相对于数据库代理历史版本（免费版），数据库代理企业版将支持多主架构、计算节点秒级弹性扩展、Proxy限流保护等更多高级功能。具体请参见数据库代理企业版概述。  说明 数据库代理企业版当前可免费使用，暂不收费，未来收费时间待定。
存储类型	PolarDB支持PSL5和PSL4两种存储类型： ◦ PSL5：PolarDB历史版本中支持的存储类型，即2022年06月07日之前购买的PolarDB集群默认的存储类型。性能好，可靠性和可用性更强。 ◦ PSL4：PolarDB全新推出的存储类型，采用阿里巴巴自研的硬件压缩盘（Smart-SSD）技术，在物理SSD磁盘层面压缩、解压缩存储的数据，保持性能影响可控的情况下，使单位容量数据的存储价格更低。  说明 ■ 当前仅PolarDB MySQL引擎8.0版本支持PSL4存储类型。 ■ 对于已创建的集群，存储类型不支持切换。如需切换存储类型，建议您购买一个新的集群并配置预期的存储类型，将原有集群的数据迁移到新集群即可。 关于两种存储类型的详细对比，请参见如何选择PSL4和PSL5。
存储费用	您购买时无需选择容量，PolarDB会根据实际使用量按小时计费，您也可以预购存储包，如何购买存储包请参见 购买存储包 。
集群名称	输入集群名称，集群名称需满足如下要求： ◦ 不能以 <code>http://</code> 或 <code>https://</code> 开头。 ◦ 长度为2~256个字符。 如果留空，系统将为自动生成一个集群名称，创建集群后还可以修改。
购买时长	选择集群的购买时长。  说明 仅当付费模式为包年包月时支持设置该参数。
购买数量	选择集群的 购买数量 。

8. 阅读并选中服务协议，并根据集群的**付费模式**完成后续购买操作。

- **按量付费**
单击**立即购买**即可。
- **包年包月**

- a. 单击**立即购买**。
- b. 在**支付**页面，确认未支付订单信息和支付方式，单击**订购**。

 **说明** 购买成功后，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。

后续步骤

1. 登录到新集群来验证数据。登录集群的操作步骤，请参见[连接数据库集群](#)。
2. 迁移数据到原集群。
确认新集群的数据之后，您可以将需要的数据从新集群迁移回原集群。详情请参见[PolarDB MySQL集群间的数据迁移](#)。

 **说明** 数据迁移是指将一个集群（称为源集群）的数据复制到另一个集群（称为目标集群），迁移操作不会对源集群造成影响。

相关API

API	描述
CreateDBCluster	PolarDB的数据恢复需要通过CreateDBCluster来实现。  说明 参数CreationOption取值需要为CloneFromPolarDB。

相关视频

10.4.2. 全量恢复方式2：恢复到过去时间点

全量恢复是指将PolarDB全量历史数据恢复至一个新的集群，验证新集群数据后，您可以再将恢复后的数据迁移至原集群。全量恢复支持从备份集恢复和恢复到过去时间点两种恢复方式。本文介绍如何将全量历史数据恢复到过去时间点。

注意事项

- 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。
- 可恢复至的时间点的范围与备份设置中的**日志备份保留期限**有关，默认为7天。

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，单击集群ID。
4. 在左侧导航栏中，选择**配置与管理 > 备份恢复**。
5. 按时间点恢复。
 - 同地域恢复：
在**备份恢复**页面，单击**按时间点恢复**。



- 跨地域恢复：
 - 在备份恢复页面，选择需要恢复的数据所在的地域。
 - 单击按时间点恢复。



6. 在克隆实例页面，选择新集群的付费模式。
- 包年包月：在创建集群时支付计算节点的费用，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。
 - 按量付费：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。

说明 PolarDB MySQL引擎已推出计算包。推荐购买按量付费的集群，因为按量付费的集群支持配合计算包使用，比包年包月付费方式更划算、更灵活。详情请参见[计费方式1：按量付费+计算包（推荐）](#)。

7. 设置以下参数。

参数	说明
操作类型	选择恢复到过去时间点。
备份时间点	选择需要恢复的时间点。 说明 该时间点与备份设置中的日志备份保留期限有关，默认为7天。
地域	选择恢复到的目标地域。 说明 - 如果开启跨地域备份，支持恢复到源地域和目标地域。 - 如果未开启跨地域备份，默认与源集群所在地域一致，无需选择。

参数	说明
主可用区	选择集群的主可用区。  说明 在有两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	固定为专有网络，无需选择。  说明 如需使用经典网络请先任选一个VPC专有网络，待集群创建完成后，再申请经典网络连接地址。申请经典网络连接地址的步骤，请参见申请集群地址和主地址。
VPC网络	选择集群所在的 VPC网络 和 VPC交换机 ，建议与原集群一致。
VPC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。
兼容性	默认与原集群兼容性一致，无需选择。 例如原集群兼容性为MySQL 8.0（与MySQL 8.0完全兼容），此处兼容性也固定为MySQL 8.0。
系列	默认与原集群系列一致，无需选择。 例如原集群系列为集群版（2-16个节点）【推荐】，此处系列也固定为集群版（2-16个节点）【推荐】。更多详情，请参见概述。
子系列	PolarDB MySQL引擎集群版支持通用规格和独享规格两种子系列，其中：关于两种类型的详细对比，请参见如何选择通用规格和独享规格。  说明 仅当原集群系列为集群版（2-16个节点）【推荐】时，支持该配置。系列为单节点（入门级）或历史库单节点版时，不支持该配置。
节点规格	选择节点规格，不同规格有不同的最大存储容量和性能。更多详情，请参见计算节点规格。  说明 为了保障恢复后的集群运行正常，建议选择等于或高于原集群的节点规格。

参数	说明
节点个数	- 若原集群系列为集群版（2-16个节点）【推荐】，节点个数默认为2，无需选择。  说明 集群版集群默认包含1主1只读两个节点，创建成功后最多可添加至1主15只读节点。关于如何添加节点，请参见增加或删除节点。 - 若原集群系列为单节点（入门级）或历史库单节点版，节点个数默认为1，无需选择。
数据库代理类型	PolarDB支持企业通用版和企业独享版两种数据库代理类型。 企业通用版：配套集群子系列的通用规格。它可以共享CPU物理资源，可根据业务负载，提供智能秒级资源弹性扩展能力。 企业独享版：配套集群子系列的独享规格。它可以独占CPU物理资源，具有更好的性能稳定性。 相对于数据库代理历史版本（免费版），数据库代理企业版将支持多主架构、计算节点秒级弹性扩展、Proxy限流保护等更多高级功能。具体请参见数据库代理企业版概述。  说明 数据库代理企业版当前可免费使用，暂不收费，未来收费时间待定。
存储类型	PolarDB支持PSL5和PSL4两种存储类型： PSL5：PolarDB历史版本中支持的存储类型，即2022年06月07日之前购买的PolarDB集群默认的存储类型。性能好，可靠性和可用性更强。 PSL4：PolarDB全新推出的存储类型，采用阿里巴巴自研的硬件压缩盘（Smart-SSD）技术，在物理SSD磁盘层面压缩、解压缩存储的数据，保持性能影响可控的情况下，使单位容量数据的存储价格更低。  说明 - 当前仅PolarDB MySQL引擎8.0版本支持PSL4存储类型。 - 对于已创建的集群，存储类型不支持切换。如需切换存储类型，建议您购买一个新的集群并配置预期的存储类型，将原有集群的数据迁移到新集群即可。 关于两种存储类型的详细对比，请参见如何选择PSL4和PSL5。
存储费用	您购买时无需选择容量，PolarDB会根据实际使用量按小时计费，您也可以预购存储包，如何购买存储包请参见购买存储包。
集群名称	输入集群名称，集群名称需满足如下要求： - 不能以 <code>http://</code> 或 <code>https://</code> 开头。 - 长度为2~256个字符。 如果留空，系统将为自动生成一个集群名称，创建集群后还可以修改。

参数	说明
购买时长	选择集群的购买时长。 ❓ 说明 仅当付费模式为包年包月时支持设置该参数。
购买数量	选择集群的购买数量。

8. 阅读并选中服务协议，并根据集群的付费模式完成后续购买操作。

- 按量付费
单击立即购买即可。
- 包年包月
 - a. 单击立即购买。
 - b. 在支付页面，确认未支付订单信息和支付方式，单击订购。

❓ 说明 购买成功后，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。

后续步骤

1. 登录到新集群来验证数据。登录集群的操作步骤，请参见[连接数据库集群](#)。
2. 迁移数据到原集群。
确认新集群的数据之后，您可以将需要的数据从新集群迁移回原集群。详情请参见[PolarDB MySQL集群间的数据迁移](#)。

❓ 说明 数据迁移是指将一个集群（称为源集群）的数据复制到另一个集群（称为目标集群），迁移操作不会对源集群造成影响。

相关API

API	描述
CreateDBCluster	PolarDB的数据恢复需要通过CreateDBCluster来实现。 ❓ 说明 参数CreationOption取值需要为CloneFromPolarDB。

10.4.3. 库表恢复方式1：从备份集恢复

库表恢复是指仅恢复指定的部分库或部分表到原集群。例如游戏业务中有时仅需恢复某个或某些玩家的数据，此时可使用库表恢复方式。库表恢复支持从备份集恢复和恢复到过去时间点两种备份方式。从本文介绍如何从备份集恢复指定的库或表。

使用限制

- 仅PolarDB集群版支持库表恢复，且版本需为以下版本之一：

- PolarDB MySQL引擎5.6版本且内核小版本需为5.6.1.0.25及以上。关于如何升级小版本，请参见[版本管理](#)。
- PolarDB MySQL引擎5.7版本且内核小版本需为5.7.1.0.8及以上。
- PolarDB MySQL引擎8.0版本且内核小版本需为8.0.1.1.14及以上。

 **说明** PolarDB MySQL引擎内核小版本8.0.2暂时不支持库表恢复功能。

- PolarDB单节点版、历史库单节点版、历史库集群版均不支持库表恢复。
- 开启透明数据加密TDE的集群暂不支持库表恢复功能。
- GDN中的集群不支持库表恢复功能。

注意事项

- 当前库表恢复方式中的从备份集恢复，只支持从一级备份的备份集里恢复，不支持从二级备份恢复。
- 库表恢复只会恢复指定的表，操作时请确认已选中所有需要恢复的表。

 **说明** 若无法确定所有涉及的表，建议您将整个集群的历史数据恢复到一个新集群后再迁移至原集群。更多详情，请参见[全量恢复方式1：从备份集恢复](#)和[全量恢复方式2：恢复到过去时间点](#)。

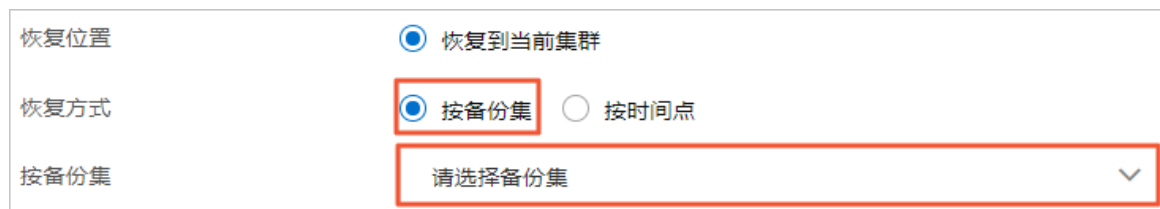
- 当选择[按时间点](#)的方式进行库表恢复时，若最近一份备份集生成时间点到指定恢复的时间点内指定的表已被删除，库表恢复会失败。
- 每次最多支持恢复100张表。若选择恢复库，则支持恢复的表数量为该库下所有的表。

 **说明** 若您需要恢复的表数量较多，建议您选择恢复全量历史数据到新集群中的方案，详细操作步骤，请参见[全量恢复方式1：从备份集恢复](#)和[全量恢复方式2：恢复到过去时间点](#)。

- 集群内的表低于50000张才可以使用库表恢复功能，超过50000张表时无法使用。
- 库表恢复功能不支持恢复触发器（Trigger），若原表设置了Trigger，该Trigger不会被恢复。
- 库表恢复功能不支持恢复外键（Foreign Key），若原表设置了Foreign Key，该Foreign Key不会被恢复。

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[备份恢复](#)页面，单击库（表）恢复。
6. 在弹出的对话框中，选择恢复方式为按备份集，并在备份集列表中选择目标备份集。



对话框显示了恢复配置选项：

- 恢复位置**：包含一个单选按钮，当前选中“恢复到当前集群”。
- 恢复方式**：包含两个单选按钮，当前选中“按备份集”，另一个为“按时间点”。
- 按备份集**：下方是一个下拉菜单，显示“请选择备份集”，右侧有一个向下箭头。

7. 在需要恢复的库和表区域左侧，选中需要恢复的目标库，并在右侧选中目标表。

需要恢复的库和表

请输入

Q

库名

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

☐

☐

☐

☒

- PolarDB MySQL引擎5.6版本且内核小版本需为5.6.1.0.25及以上。关于如何升级小版本，请参见[版本管理](#)。
- PolarDB MySQL引擎5.7版本且内核小版本需为5.7.1.0.8及以上。
- PolarDB MySQL引擎8.0版本且内核小版本需为8.0.1.1.14及以上。

 说明 PolarDB MySQL引擎内核小版本8.0.2暂时不支持库表恢复功能。

- PolarDB单节点版、历史库单节点版、历史库集群版均不支持库表恢复。
- 开启透明数据加密TDE的集群暂不支持库表恢复功能。
- GDN中的集群不支持库表恢复功能。

注意事项

- 库表恢复只会恢复指定的表，操作时请确认已选中所有需要恢复的表。

 说明 若无法确定所有涉及的表，建议您将整个集群的历史数据恢复到一个新集群后再迁移至原集群。更多详情，请参见[全量恢复方式1：从备份集恢复](#)和[全量恢复方式2：恢复到过去时间点](#)。

- 当选择按时间点的方式进行库表恢复时，若最近一份备份集生成时间点到指定恢复的时间点内指定的表已被删除，库表恢复会失败。
- 每次最多支持恢复100张表。若选择恢复库，则支持恢复的表数量为该库下所有的表。

 说明 若您需要恢复的表数量较多，建议您选择恢复全量历史数据到新集群中的方案，详细操作步骤，请参见[全量恢复方式1：从备份集恢复](#)和[全量恢复方式2：恢复到过去时间点](#)。

- 集群内的表低于50000张才可以使用库表恢复功能，超过50000张表时无法使用。
- 库表恢复功能不支持恢复触发器（Trigger），若原表设置了Trigger，该Trigger不会被恢复。
- 库表恢复功能不支持恢复外键（Foreign Key），若原表设置了Foreign Key，该Foreign Key不会被恢复。

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. 在备份恢复页面，单击库（表）恢复。
6. 在弹出的对话框中，选择恢复方式为按时间点，并选择需要恢复至的时间点。

恢复位置	☒ 恢复到当前集群
恢复方式	☐ 按备份集 ☒ 按时间点
可恢复时间点	2021-03-30 14:37:43 至 2021-04-06 14:37:43
恢复至	2021-03-30 14:37:43 

说明 恢复至的时间点选择需符合可恢复时间点的限制，同时离该时间点最近一份全量备份集中必须存在需要恢复的表，才能正常使用按时间点恢复功能。可恢复时间点的范围与备份设置中的日志备份保留期限有关，默认为7天。

7. 在需要恢复的库和表区域左侧，选中需要恢复的目标库，并在右侧选中目标表。

需要恢复的库和表

请输入

Q

库名

☐

☐

☒

☐

库名

表名

☐

☒

<<

<

1

>

>>

已选择的库和表

库名	恢复后库名	表名	恢复后表名	操作
				删除

说明

- 若选中目标库后但不指定具体的数据表，默认恢复该库下的所有数据。
- 选中目标库或目标表后，系统会自动在原库或原表名称（如 `test`）后加 `_backup` 作为新库或新表的名称（如 `test_backup`），您也可以自定义恢复后的库名或表名。

8. 单击**确定**即可。

相关API

API	描述
DescribeMetaList	调用DescribeMetaList接口查看支持恢复的库表元信息详情。
RestoreTable	调用RestoreTable恢复PolarDB库表。

10.4.5. 误操作恢复最佳实践

数据误操作有多种场景，例如：误删表，误删数据库，误删集群，误操作表中数据（如整体覆盖、误删除/修改表中的列/行/数据）。

根据场景的不同以及对应数据库引擎版本的不同，PolarDB MySQL引擎提供了多种方式，帮助您恢复数据。

请根据[不同场景下恢复方式概览](#)中的介绍，选择对应的数据恢复方式，并根据操作步骤，完成数据恢复的完整操作流程。

10.5. 常见问题

本文介绍PolarDB MySQL引擎备份与恢复功能的常见问题。

备份FAQ

- Q：一级备份（快照）总大小是否为所有一级备份（快照）之和？
A：一级备份（快照）总大小不是所有备份（快照）之和，即并非下图中的②，而是①所示的大小。

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	备份方法	备份类型	大小	存储位置
	2021年3月28日 15:10:05	2021年3月28日 15:10:15	备份完成	2021年3月28日 15:10:08	快照备份	全量备份	3.49 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。以中国内地价格为例，PolarDB数据库的一级备份总大小为700 GB，数据库存储用量为1000 GB，那么每小时费用为(700 GB-500 GB)×0.003元/GB=0.6元。
- Q：存储包是否支持抵扣备份空间的费用？
A：支持，购买存储包后，若抵扣完账号下所有PolarDB集群的存储空间用量后存储包容量仍有剩余，将自动按照1:1.6的比例抵扣一级备份中超过免费额度的空间用量，直至抵扣完存储包中的所有容量。若存储包中剩余容量不足以抵扣一级备份所用用量，超出部分将正常按量付费。更多关于存储包抵扣的问题，请参见[存储包](#)。
- Q：手动备份仅支持一级备份吗？
A：是的。
- Q：手动备份保留时长是多久？

A：手动备份的备份文件的保留时长，由备份设置中的数据备份保留中一级备份设置的具体时长决定。

数据备份列表

物理日志备份列表

备份策略设置

有关数据恢复，请查看“不同场景下恢复方式概览”

备份策略设置

编辑

数据备份

使用存储层的快照技术对数据库进行无锁备份

一级备份

常规备份（按备份周期）（如需提升备份频率，可切换为高频备份）

备份周期（周一、周二）

备份开始时间（14:00 - 15:00）

备份保留时长（7天）

二级备份

开启

备份周期（周一、周二）

同城备份保留（30天）

跨地域备份地域（华北2（北京））

跨地域备份保留（30天）

日志备份

离线保存每一个数据库Redo日志文件

同城备份保留（7天）

跨地域备份地域（华北2（北京））

跨地域备份保留（30天）

通用

集群备份保留

删除集群时会自动备份，永久保留该备份集。

- Q：如何查看二级备份的大小？

A：您可以在控制台上备份列表页签下查看二级备份的大小。

数据备份列表	物理日志备份列表	备份策略设置	有关数据恢复，请查看“不同场景下恢复方式概览”
创建备份	华东1（杭州）	2022年5月11日	2022年6月12日
一级备份（快照）总大小 327.00 MB（其中免费额度为50%当前存储用量，大约为 1.18 GB）			
备份ID	备份开始/结束时间	备份大小	状态
1-7	2022年6月12日 06:01:06-2022年6月12日 06:02:06	30.53 MB	备份完成
1-2	2022年6月5日 06:01:27-2022年6月5日 06:02:27	22.50 MB	备份完成

- Q：如何将PolarDB MySQL引擎的备份集下载到本地？

A：由于PolarDB的备份集采用存储层的快照备份机制，导出的物理文件无法在本地普通的文件系统中恢复，因此下载快照备份集到本地无意义，但您可以通过mysql dump工具下载或通过阿里云的DBS服务备份PolarDB MySQL并手动下载备份集。

恢复FAQ

- Q：数据误操作时，例如误删除（表、数据库、集群等）、误修改（整体覆盖或误修改表中的列、行、数据等），如何进行恢复？

A：您可以根据业务场景及对应数据库引擎版本，选择不同的方式恢复数据。具体场景选择及操作步骤，请参见不同场景下恢复方式概览。

- Q: 之前可以使用库表恢复, 为什么突然不能用了?
A: 可能您集群内的表数量过多。集群内的表低于50000张才可以使用库表恢复功能, 超过50000张表时无法使用。
- Q: 支持自定义恢复后的新库或新表的名称吗?
A: 支持。
- Q: 没有数据备份可以按时间点恢复吗?
A: 不可以。因为按时间点恢复是先将所选时间点前的一个全量数据备份恢复到集群, 然后根据Redo日志增量恢复数据到所选时间点。

11.热备切换

11.1. 概述

PolarDB提供了热备切换功能，您可事先为集群中的只读节点开启热备功能，从而在主备切换的过程中实现快速切换和连接保持。

PolarDB面向高可用场景，推出了新的方案——快速热备切换。该方案从切换速度和切换体验两个方面，整合了多项技术，针对切换场景进行了优化，包括计划内的切换如集群升降配，以及计划外的容灾切换。热备切换功能的核心技术请参见[技术原理](#)。

适用版本

仅PolarDB MySQL引擎集群版5.7、8.0版本支持热备功能。

适用场景

优化闲置节点

对于一致性读要求高和写多读少、数据变化快的业务场景，直接使用主地址服务可以简化业务逻辑。PolarDB基于容灾的需求，默认创建至少一个只读节点。对于不提供读扩展的闲置只读节点，开启热备切换后，可以优化主节点的刷脏性能和高可用容灾的效率。

变配需求

热备节点通过全局预热系统优化热备切换耗时，引入连接保持和事务续传实现了应用视角的无感运维。开启热备和事务续传之后，在[主动运维](#)、[升降配](#)、[小版本升级](#)三个场景下，PolarDB都可以实现连接和事务不中断，业务客户端报错量显著降低。

数据导入

新实例创建或者业务低峰期，用户会有批量导入数据的需求。您可以选择在数据导入阶段临时开启热备，主节点就可以保留更多的资源服务于数据写入请求，从而使集群的运行状态更加平滑。

只读高可用

对于分析型业务的可用性有高要求的用户，PolarDB支持只读节点和热备节点的动态转换。可以额外配置一个热备节点。该热备节点不仅作为主节点的容灾备库，也可以作为分析型业务只读节点的容灾节点，起到一备多的效果。

11.2. 费用说明

本文档介绍了热备功能的费用。

热备功能本身不收取费用，只收取开启热备的只读节点的费用。关于只读节点的费用请参见[按量付费计费规则](#)和[包年包月计费规则](#)。

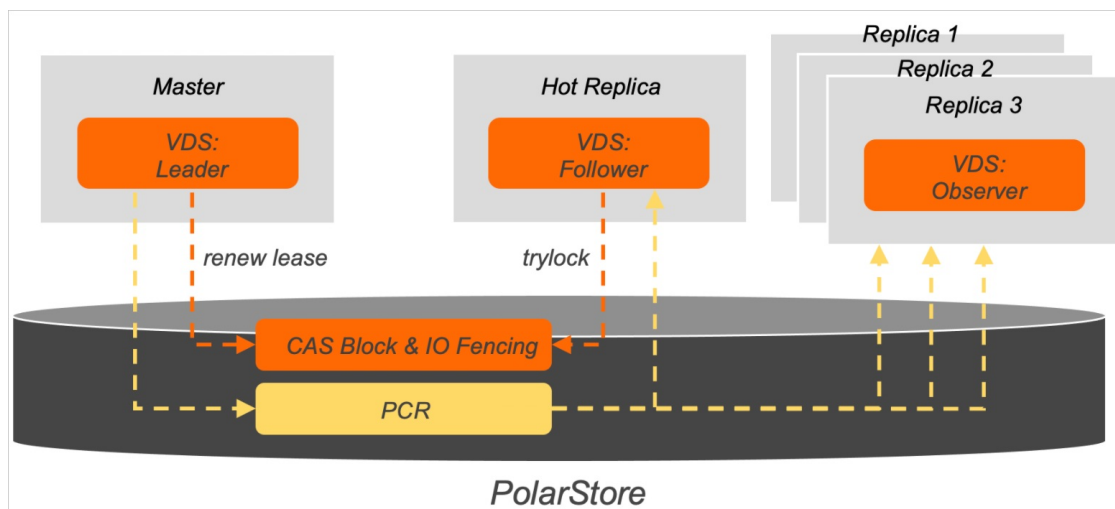
11.3. 技术原理

本文介绍了PolarDB热备切换功能的技术原理。

PolarDB热备切换功能的核心技术如下：

- 引入全新的高可用模块VDS（Voting Disk Service），该模块基于共享存储架构，实现自治的集群节点管理，大幅降低故障检测和集群选主的耗时；
- 新增支持全局预热系统的热备节点，通过对存储引擎内部的多个模块提前预热，优化热备切换的耗时；
- 支持连接保持和事务续传功能。在集群升降配或小版本升级过程中，开启连接保持和事务续传功能后，系统会尽可能保证用户连接和事务不中断，实现基本无感知的主动运维。

全新的高可用系统VDS



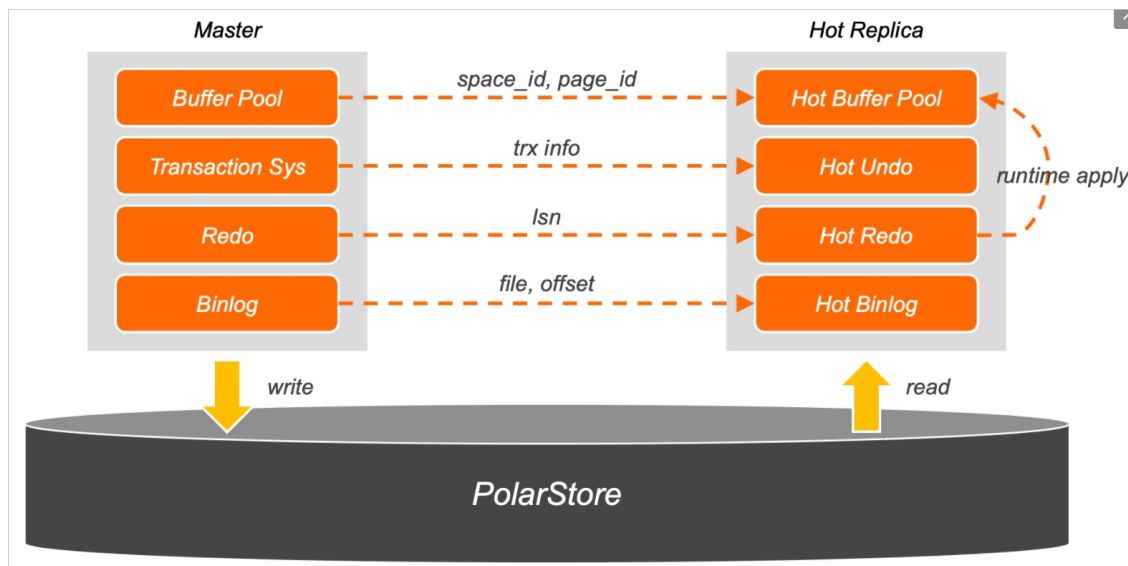
热备功能开启后，PolarDB会启用VDS。VDS借助PolarDB共享存储架构，可以实现集群节点的自治管理、故障检测和集群选主。VDS的架构说明如下：

- VDS中每个计算节点有独立的VDS线程，分为三种不同的角色：Leader、Follower和Observer。其中Leader对应PolarDB的主节点，Follower对应PolarDB的热备节点，Observer对应PolarDB的只读节点。PolarDB集群中可以包括1个Leader、1个Follower和多个Observer。
- VDS在共享存储（PolarStore）上维护了2个数据模块，分别是CAS Block和PCR（Polar Cluster Registry）。
 - CAS Block是PolarStore提供的支持Compare-And-Swap操作的原子数据块。VDS基于CAS Block实现了分布式租约锁，并在数据块中记录了锁持有者、租约时间等元信息。PolarDB的主节点和热备节点，通过续租和加锁语义，完成故障探测和集群选主。
 - PCR是一个保存了PolarDB节点管理元数据的普通文件，负责维护整个集群的拓扑状态。VDS中的Leader角色有PCR的写入权限，Follower和Observer角色只有只读权限。当VDS中的Follower角色升级为Leader时，原来的Leader角色会失效，只有最新的Leader角色才拥有共享存储的写入权限。

正常情况下，主节点对外提供读写服务，并通过VDS中的Leader定期续租。当主节点故障时，热备切换流程如下：

1. VDS Follower在租约超期之后会加锁成功变为Leader，从而使热备节点升级为新的主节点。
2. 故障的主节点恢复后，加锁失败，会自动降级为新的热备节点。
3. 在集群选主流程结束后，PCR会将新的拓扑信息广播给所有的VDS Observer。这样只读节点就能够自动连接到新的主节点，并恢复LSN、Binlog等同步链路。

全局预热系统



热备节点不同于只读节点，它不对外提供读服务，主要负责提升高可用的切换速度和切换体验。全局预热系统是热备切换中最核心的模块，主要负责实时同步主节点的元信息，将一些关键数据提前加载进内存，来提升未来的切换速度。全局预热系统包含4个模块：Hot Buffer Pool、Hot Undo、Hot Redo和Hot Binlog。

- **Hot Buffer Pool**

Hot Buffer Pool预热模块会实时监控主节点Buffer Pool的最近最少使用算法LRU（Least Recently Used）链表，并将相关信息发送给热备节点。热备节点会智能地选择最热的数据页，并将它们提前批量加载到内存中，用来缓解只读节点切换为主节点后，Buffer Pool命中率大幅下降带来的性能抖动。

- **Hot Undo**

Hot Undo预热模块是对事务系统的预热。在切换过程中，PolarDB需要从Undo Page中找出悬挂事务并进行回滚。只读节点只会服务于分析类型的大查询，不会访问主节点未提交的事务。因此在切换流程中存在Undo Page的IO等待。热备切换通过提前预热Undo Page和事务信息，提升了切换速度。

- **Hot Redo**

Hot Redo预热模块会将热备节点和只读节点的Redo日志实时缓存在内存的Redo Hash中。

- **Hot Binlog**

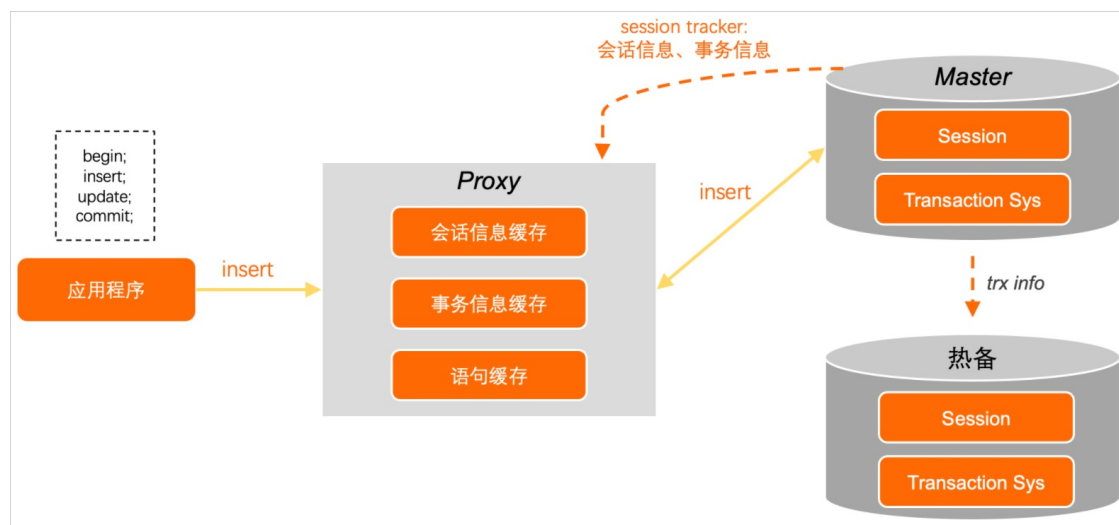
如果集群开启了Hot Binlog，切换过程中，处于Prepare状态的InnoDB事务还依赖Binlog信息来决策提交或者回滚事务。对于大事务场景，完整读取和解析最后一个Binlog文件时，经常需要耗费数秒甚至数分钟的时间。热备节点通过后台线程，异步将最新的Binlog数据缓存在IO Cache中，并提前进行解析，来彻底解决切换耗时的问题。

连接保持和事务续传

当集群执行升降配、小版本升级、主动运维或者故障容灾等操作时，需要进行主备切换。这类切换会对应用服务造成影响，导致连接闪断、新建连接短暂失败等问题，增加了应用开发的复杂性和风险。

PolarDB支持[连接保持](#)功能。连接保持的原理是数据库代理在应用程序和PolarDB之间，扮演了连接桥接的角色。当数据库进行主备切换时，数据库代理将新的后端连接（即数据库代理和数据库节点的连接）桥接到原有的前端连接（即应用程序和数据库代理的连接）上，同时恢复之前的会话状态，包括原来的系统变量、用户变量、字符集编码等信息。

连接保持功能只能作用于空闲的连接，如果切换瞬间，当前的会话有正在执行的事务，一方面数据库代理无法从PolarDB中找回原有事务的上下文，另一方面新的主节点会将未提交的悬挂事务回滚，释放这些事务持有的行锁。在这种场景下，连接保持就会失效，而PolarDB最新推出的事务续传功能可以解决这一问题。事务续传配合连接保持，可以实现对应用程序完全无感知的高可用切换。

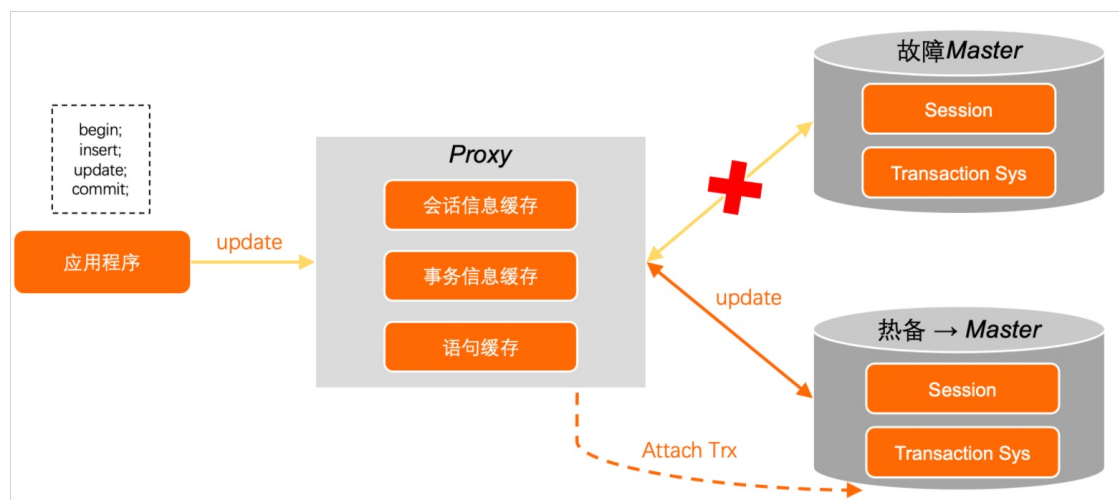


相比传统使用Binlog的逻辑复制，PolarDB基于物理复制的架构，可以在热备节点上重建与主节点完全一致的事务。

以上图为例，假设应用程序完整的事务是begin-insert-update-commit。事务开始执行后，Proxy在将SQL语句转发给主节点的同时，会缓存最近执行的SQL语句。当insert操作在主节点执行后，PolarDB会自动保存最新一条语句的Savepoint来作为事务信息的一部分。通过Session Tracker将当前的会话信息和事务信息返回给Proxy，Proxy会将这些数据临时保存到内部的缓存中。其中，会话信息用于连接保持，包括字符集、用户变量等信息；事务信息用于事务续传，包括trx_id、undo_no等信息。此外，事务信息会通过物理复制，持续同步到热备节点中。

假设应用程序在PolarDB执行update的过程中，主节点出现故障。通过热备切换选主之后，新的主节点通过Redo Log可以构建出所有的未提交事务，并异步等待未提交的事务，暂时不进行回滚。然后，Proxy探测到发生了主备切换，会利用自身缓存的会话信息和事务信息，借助PolarDB的Attach Trx接口，桥接事务状态。PolarDB会根据Proxy信息，判断相关的事务信息是否有效。如果事务有效，会把事务信息绑定到这个连接上，并回滚到最后一条语句（即上文的update）对应undo_no的Savepoint中。

当事务桥接成功后，Proxy就可以从SQL语句缓存中，把最近一次未执行成功的update操作重新发送给新的主节点。从用户角度来看，整个主备切换过程中，应用程序只感知到一条执行时间变慢的update语句，不会接收到任何连接报错或事务报错。



11.4. 开启热备前后性能对比

本文档介绍了开启热备功能前后的性能差异。

对比总览

在关闭Binlog的前提下，热备节点（开启热备切换功能）与只读节点的性能差异总览如下：

对比维度	热备节点	只读节点
服务能力	不提供服务	提供只读服务
switchover（主动运维，如小版本升级、规格升降配）	TPS跌零5秒左右	TPS跌零10秒以上
failover（故障容灾，如主节点宕机）	TPS跌零5秒左右	连接中断报错60秒左右

通过以上数据可以看出：

- 主动运维时，开启热备的只读节点相比未开启的情况，业务中断时间更短。
- 故障容灾时，开启热备的只读节点相比未开启的情况，业务中断时间更短，并且可以实现连接和事务不中断，业务客户端报错量显著降低。

详细数据

开启与关闭热备功能的切换效率的详细数据如下：

- switchover（主动运维，如小版本升级、规格升降配）
 - 普通只读节点（未开启热备和事务续传）：TPS跌零10秒以上。

```
[ 36s ] thds: 64 tps: 2708.19 qps: 54140.88 (r/w/o: 37904.72/10819.78/5416.39) lat (ms,95%): 26.20 err/s: 0.00 reconn/s: 0.00
[ 37s ] thds: 64 tps: 2703.00 qps: 54030.10 (r/w/o: 37810.07/10815.02/5405.01) lat (ms,95%): 26.20 err/s: 0.00 reconn/s: 0.00
[ 38s ] thds: 64 tps: 2698.80 qps: 54018.04 (r/w/o: 37814.23/10806.21/5397.60) lat (ms,95%): 26.20 err/s: 0.00 reconn/s: 0.00
[ 39s ] thds: 64 tps: 2703.02 qps: 54084.38 (r/w/o: 37854.27/10824.08/5406.04) lat (ms,95%): 26.20 err/s: 0.00 reconn/s: 0.00
[ 40s ] thds: 64 tps: 2694.98 qps: 53889.61 (r/w/o: 37730.73/10768.92/5389.96) lat (ms,95%): 26.68 err/s: 0.00 reconn/s: 0.00
[ 41s ] thds: 64 tps: 2704.02 qps: 54137.33 (r/w/o: 37921.23/10806.07/5410.03) lat (ms,95%): 26.20 err/s: 0.00 reconn/s: 0.00
[ 42s ] thds: 64 tps: 2703.02 qps: 53980.34 (r/w/o: 37753.24/10822.07/5405.03) lat (ms,95%): 26.68 err/s: 0.00 reconn/s: 0.00
[ 43s ] thds: 64 tps: 2696.99 qps: 54029.83 (r/w/o: 37838.88/10796.97/5393.98) lat (ms,95%): 26.20 err/s: 0.00 reconn/s: 0.00
[ 44s ] thds: 64 tps: 2715.99 qps: 54255.82 (r/w/o: 37964.88/10859.96/5430.98) lat (ms,95%): 26.20 err/s: 0.00 reconn/s: 0.00
[ 45s ] thds: 64 tps: 2714.98 qps: 54295.60 (r/w/o: 38013.72/10851.92/5429.96) lat (ms,95%): 26.20 err/s: 0.00 reconn/s: 0.00
[ 46s ] thds: 64 tps: 2711.02 qps: 54248.37 (r/w/o: 37974.26/10851.07/5423.04) lat (ms,95%): 26.20 err/s: 0.00 reconn/s: 0.00
[ 47s ] thds: 64 tps: 2714.02 qps: 54231.31 (r/w/o: 37947.22/10856.06/5428.03) lat (ms,95%): 26.20 err/s: 0.00 reconn/s: 0.00
[ 48s ] thds: 64 tps: 2713.99 qps: 54246.76 (r/w/o: 37977.84/10839.95/5428.98) lat (ms,95%): 26.20 err/s: 0.00 reconn/s: 0.00
[ 49s ] thds: 64 tps: 2700.02 qps: 54050.31 (r/w/o: 37839.21/10816.06/5395.03) lat (ms,95%): 26.20 err/s: 0.00 reconn/s: 0.00
[ 50s ] thds: 64 tps: 2710.98 qps: 54088.55 (r/w/o: 37840.68/10823.91/5423.95) lat (ms,95%): 26.20 err/s: 0.00 reconn/s: 0.00
[ 51s ] thds: 64 tps: 2550.12 qps: 50488.44 (r/w/o: 35268.70/10179.49/5040.24) lat (ms,95%): 26.20 err/s: 0.00 reconn/s: 0.00
[ 52s ] thds: 64 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,95%): 0.00 err/s: 0.00 reconn/s: 0.00
[ 53s ] thds: 64 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,95%): 0.00 err/s: 0.00 reconn/s: 0.00
[ 54s ] thds: 64 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,95%): 0.00 err/s: 0.00 reconn/s: 0.00
[ 55s ] thds: 64 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,95%): 0.00 err/s: 0.00 reconn/s: 0.00
[ 56s ] thds: 64 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,95%): 0.00 err/s: 0.00 reconn/s: 0.00
[ 57s ] thds: 64 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,95%): 0.00 err/s: 0.00 reconn/s: 0.00
[ 58s ] thds: 64 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,95%): 0.00 err/s: 0.00 reconn/s: 0.00
[ 59s ] thds: 64 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,95%): 0.00 err/s: 0.00 reconn/s: 0.00
[ 60s ] thds: 64 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,95%): 0.00 err/s: 0.00 reconn/s: 0.00
[ 61s ] thds: 64 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,95%): 0.00 err/s: 0.00 reconn/s: 0.00
[ 62s ] thds: 64 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,95%): 0.00 err/s: 0.00 reconn/s: 0.00
[ 63s ] thds: 64 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,95%): 0.00 err/s: 0.00 reconn/s: 0.00
[ 64s ] thds: 64 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,95%): 0.00 err/s: 0.00 reconn/s: 0.00
```

- 热备只读节点（开启热备和事务续传）：TPS跌零5秒左右。

```
[ 33s ] thds: 64 tps: 2427.11 qps: 48339.19 (r/w/o: 33788.53/9696.44/4854.22) lat (ms,95%): 30.81 err/s: 0.00 reconns/s: 0.00
[ 34s ] thds: 64 tps: 2391.22 qps: 48200.34 (r/w/o: 33794.04/9621.87/4784.43) lat (ms,95%): 30.81 err/s: 0.00 reconns/s: 0.00
[ 35s ] thds: 64 tps: 2427.71 qps: 48447.26 (r/w/o: 33919.98/9671.85/4855.43) lat (ms,95%): 30.81 err/s: 0.00 reconns/s: 0.00
[ 36s ] thds: 64 tps: 2456.12 qps: 48941.31 (r/w/o: 34219.61/9813.46/4908.23) lat (ms,95%): 30.26 err/s: 0.00 reconns/s: 0.00
[ 37s ] thds: 64 tps: 2402.02 qps: 48214.46 (r/w/o: 33790.32/9616.09/4808.05) lat (ms,95%): 30.81 err/s: 0.00 reconns/s: 0.00
[ 38s ] thds: 64 tps: 2377.98 qps: 47430.59 (r/w/o: 33179.71/9494.92/4755.96) lat (ms,95%): 31.37 err/s: 0.00 reconns/s: 0.00
[ 39s ] thds: 64 tps: 2395.97 qps: 48032.41 (r/w/o: 33600.59/9640.88/4790.94) lat (ms,95%): 30.81 err/s: 0.00 reconns/s: 0.00
[ 40s ] thds: 64 tps: 2360.57 qps: 47201.33 (r/w/o: 33074.94/9407.26/4719.13) lat (ms,95%): 31.37 err/s: 0.00 reconns/s: 0.00
[ 41s ] thds: 64 tps: 2465.96 qps: 49155.27 (r/w/o: 34367.49/9855.85/4931.93) lat (ms,95%): 29.72 err/s: 0.00 reconns/s: 0.00
[ 42s ] thds: 64 tps: 2436.03 qps: 48872.56 (r/w/o: 34237.39/9764.11/4871.06) lat (ms,95%): 29.72 err/s: 0.00 reconns/s: 0.00
[ 43s ] thds: 64 tps: 2311.01 qps: 46276.24 (r/w/o: 32385.17/9265.05/4626.02) lat (ms,95%): 33.12 err/s: 0.00 reconns/s: 0.00
[ 44s ] thds: 64 tps: 2382.00 qps: 47578.10 (r/w/o: 33299.07/9519.02/4760.01) lat (ms,95%): 31.37 err/s: 0.00 reconns/s: 0.00
[ 45s ] thds: 64 tps: 2303.99 qps: 46139.88 (r/w/o: 32303.92/9223.98/4611.99) lat (ms,95%): 33.12 err/s: 0.00 reconns/s: 0.00
[ 46s ] thds: 64 tps: 2266.00 qps: 45233.03 (r/w/o: 31658.02/9044.01/4531.00) lat (ms,95%): 33.72 err/s: 0.00 reconns/s: 0.00
[ 47s ] thds: 64 tps: 2448.98 qps: 48876.56 (r/w/o: 34201.69/9776.91/4897.96) lat (ms,95%): 29.72 err/s: 0.00 reconns/s: 0.00
[ 48s ] thds: 64 tps: 2443.01 qps: 49073.25 (r/w/o: 34396.17/9791.05/4886.02) lat (ms,95%): 29.72 err/s: 0.00 reconns/s: 0.00
[ 49s ] thds: 64 tps: 2369.98 qps: 47161.57 (r/w/o: 32979.70/9444.91/4736.96) lat (ms,95%): 31.37 err/s: 0.00 reconns/s: 0.00
[ 50s ] thds: 64 tps: 2412.03 qps: 48338.63 (r/w/o: 33826.44/9692.13/4820.06) lat (ms,95%): 30.26 err/s: 0.00 reconns/s: 0.00
[ 51s ] thds: 64 tps: 2453.00 qps: 49064.93 (r/w/o: 34346.95/9805.99/4911.99) lat (ms,95%): 29.72 err/s: 0.00 reconns/s: 0.00
[ 52s ] thds: 64 tps: 2462.99 qps: 49209.86 (r/w/o: 34455.90/9826.97/4926.99) lat (ms,95%): 29.72 err/s: 0.00 reconns/s: 0.00
[ 53s ] thds: 64 tps: 2450.97 qps: 48999.48 (r/w/o: 34314.64/9789.90/4894.95) lat (ms,95%): 29.72 err/s: 0.00 reconns/s: 0.00
[ 54s ] thds: 64 tps: 350.02 qps: 6975.32 (r/w/o: 4876.22/1394.06/705.03) lat (ms,95%): 30.26 err/s: 0.00 reconns/s: 0.00
[ 55s ] thds: 64 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,95%): 0.00 err/s: 0.00 reconns/s: 0.00
[ 56s ] thds: 64 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,95%): 0.00 err/s: 0.00 reconns/s: 0.00
[ 57s ] thds: 64 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,95%): 0.00 err/s: 0.00 reconns/s: 0.00
[ 58s ] thds: 64 tps: 484.98 qps: 9434.61 (r/w/o: 6520.73/1950.92/962.96) lat (ms,95%): 4683.57 err/s: 0.00 reconns/s: 0.00
[ 59s ] thds: 64 tps: 2108.03 qps: 42603.70 (r/w/o: 29945.49/8433.14/4225.07) lat (ms,95%): 38.94 err/s: 0.00 reconns/s: 0.00
[ 60s ] thds: 64 tps: 2245.97 qps: 44820.47 (r/w/o: 31301.63/9032.89/4485.95) lat (ms,95%): 34.33 err/s: 0.00 reconns/s: 0.00
[ 61s ] thds: 64 tps: 2187.98 qps: 43689.61 (r/w/o: 30575.73/8737.92/4375.96) lat (ms,95%): 34.95 err/s: 0.00 reconns/s: 0.00
```

- failover（故障容灾，如主节点宕机）
 - 普通只读节点（未开启热备和事务续传）：连接中断报错60秒左右。

```
[ 8s ] thds: 64 tps: 2739.73 qps: 54652.67 (r/w/o: 38253.27/10919.94/5479.47) lat (ms,95%): 26.20 err/s: 0.00 reconn/s: 0.00
[ 9s ] thds: 64 tps: 2729.13 qps: 54679.69 (r/w/o: 38282.88/10937.54/5459.27) lat (ms,95%): 26.20 err/s: 0.00 reconn/s: 0.00
[ 10s ] thds: 64 tps: 2735.17 qps: 54688.38 (r/w/o: 38288.37/10932.68/5467.34) lat (ms,95%): 26.20 err/s: 0.00 reconn/s: 0.00
[ 11s ] thds: 64 tps: 2739.96 qps: 54792.19 (r/w/o: 38345.43/10964.84/5481.92) lat (ms,95%): 26.20 err/s: 0.00 reconn/s: 0.00
[ 12s ] thds: 64 tps: 2731.81 qps: 54665.11 (r/w/o: 38272.27/10928.22/5464.61) lat (ms,95%): 26.20 err/s: 0.00 reconn/s: 0.00
[ 13s ] thds: 64 tps: 2733.08 qps: 54711.56 (r/w/o: 38309.09/10939.31/5463.16) lat (ms,95%): 26.20 err/s: 0.00 reconn/s: 0.00
[ 14s ] thds: 64 tps: 2726.08 qps: 54524.63 (r/w/o: 38159.14/10911.33/5454.16) lat (ms,95%): 26.20 err/s: 0.00 reconn/s: 0.00
[ 15s ] thds: 64 tps: 2723.08 qps: 54487.65 (r/w/o: 38146.15/10897.33/5444.16) lat (ms,95%): 26.20 err/s: 0.00 reconn/s: 0.00
[ 16s ] thds: 64 tps: 2728.77 qps: 54461.46 (r/w/o: 38098.82/10905.09/5457.54) lat (ms,95%): 26.20 err/s: 0.00 reconn/s: 0.00
[ 17s ] thds: 64 tps: 2727.01 qps: 54562.25 (r/w/o: 38201.18/10907.05/5454.03) lat (ms,95%): 26.20 err/s: 0.00 reconn/s: 0.00
[ 18s ] thds: 64 tps: 2727.06 qps: 54522.20 (r/w/o: 38170.84/10896.24/5455.12) lat (ms,95%): 26.20 err/s: 0.00 reconn/s: 0.00
[ 19s ] thds: 64 tps: 2725.02 qps: 54567.37 (r/w/o: 38206.26/10910.07/5451.04) lat (ms,95%): 26.20 err/s: 0.00 reconn/s: 0.00
[ 20s ] thds: 64 tps: 2734.14 qps: 54535.77 (r/w/o: 38155.94/10912.55/5467.28) lat (ms,95%): 26.20 err/s: 0.00 reconn/s: 0.00
[ 21s ] thds: 64 tps: 2723.65 qps: 54563.98 (r/w/o: 38200.09/10916.60/5447.30) lat (ms,95%): 26.20 err/s: 0.00 reconn/s: 0.00
[ 22s ] thds: 64 tps: 2721.34 qps: 54447.70 (r/w/o: 38113.69/10890.34/5443.67) lat (ms,95%): 26.20 err/s: 0.00 reconn/s: 0.00
[ 23s ] thds: 64 tps: 2727.85 qps: 54551.03 (r/w/o: 38184.92/10910.41/5455.70) lat (ms,95%): 26.20 err/s: 0.00 reconn/s: 0.00
[ 24s ] thds: 64 tps: 2716.98 qps: 54330.60 (r/w/o: 38041.72/10856.92/5431.96) lat (ms,95%): 26.20 err/s: 0.00 reconn/s: 0.00
[ 25s ] thds: 64 tps: 2725.96 qps: 54504.28 (r/w/o: 38148.50/10903.86/5451.93) lat (ms,95%): 26.20 err/s: 0.00 reconn/s: 0.00
[ 26s ] thds: 64 tps: 2718.91 qps: 54422.29 (r/w/o: 38094.80/10887.66/5439.83) lat (ms,95%): 26.20 err/s: 0.00 reconn/s: 0.00
[ 27s ] thds: 64 tps: 2725.08 qps: 54460.59 (r/w/o: 38114.11/10894.32/5452.16) lat (ms,95%): 26.20 err/s: 0.00 reconn/s: 0.00
[ 28s ] thds: 64 tps: 2723.81 qps: 54433.12 (r/w/o: 38093.28/10898.23/5441.61) lat (ms,95%): 26.20 err/s: 0.00 reconn/s: 0.00
[ 29s ] thds: 64 tps: 2706.19 qps: 54206.77 (r/w/o: 37962.63/10827.76/5416.38) lat (ms,95%): 26.20 err/s: 0.00 reconn/s: 0.00
[ 30s ] thds: 64 tps: 2727.94 qps: 54494.88 (r/w/o: 38126.21/10912.78/5455.89) lat (ms,95%): 26.20 err/s: 0.00 reconn/s: 0.00
[ 31s ] thds: 64 tps: 2729.05 qps: 54616.00 (r/w/o: 38243.70/10915.20/5457.10) lat (ms,95%): 26.20 err/s: 0.00 reconn/s: 0.00
[ 32s ] thds: 64 tps: 2731.98 qps: 54622.69 (r/w/o: 38240.78/10917.94/5463.97) lat (ms,95%): 26.20 err/s: 0.00 reconn/s: 0.00
[ 33s ] thds: 64 tps: 2729.00 qps: 54642.01 (r/w/o: 38261.01/10922.00/5459.00) lat (ms,95%): 26.20 err/s: 0.00 reconn/s: 0.00
[ 34s ] thds: 64 tps: 2413.85 qps: 48485.04 (r/w/o: 33991.95/9669.40/4823.70) lat (ms,95%): 26.20 err/s: 0.00 reconn/s: 0.00
[ 35s ] thds: 64 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,95%): 0.00 err/s: 0.00 reconn/s: 0.00 0:36秒
[ 36s ] thds: 64 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,95%): 0.00 err/s: 0.00 reconn/s: 0.00

FATAL: 'thread_run' function failed: /usr/local/share/sysbench/oltp_common.lua:405: SQL error, errno = 2013, state = 'HY000'
connection to MySQL server during query
(last message repeated 2 times)
FATAL: mysql_drv_query() returned error 2013 (Lost connection to MySQL server during query) for query 'BEGIN'
FATAL: 'thread_run' function failed: /usr/local/share/sysbench/oltp_common.lua:405: SQL error, errno = 2013, state = 'HY000'
connection to MySQL server during query
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)

Running the test with following options:
Number of threads: 64
Report intermediate results every 1 second(s)
Initializing random number generator from current time

Initializing worker threads...

Threads started!

[ 1s ] thds: 64 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,95%): 0.00 err/s: 0.00 reconn/s: 0.00
[ 2s ] thds: 64 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,95%): 0.00 err/s: 0.00 reconn/s: 0.00
[ 3s ] thds: 64 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,95%): 0.00 err/s: 0.00 reconn/s: 0.00
[ 4s ] thds: 64 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,95%): 0.00 err/s: 0.00 reconn/s: 0.00
[ 5s ] thds: 64 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,95%): 0.00 err/s: 0.00 reconn/s: 0.00
[ 6s ] thds: 64 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,95%): 0.00 err/s: 0.00 reconn/s: 0.00
[ 7s ] thds: 64 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,95%): 0.00 err/s: 0.00 reconn/s: 0.00
[ 8s ] thds: 64 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,95%): 0.00 err/s: 0.00 reconn/s: 0.00
[ 9s ] thds: 64 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,95%): 0.00 err/s: 0.00 reconn/s: 0.00
[ 10s ] thds: 64 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,95%): 0.00 err/s: 0.00 reconn/s: 0.00 02:17秒
[ 11s ] thds: 64 tps: 2542.18 qps: 51469.30 (r/w/o: 36126.28/10199.69/5143.33) lat (ms,95%): 28.16 err/s: 0.00 reconn/s: 0.00
```

- 热备只读节点（开启热备和事务续传）：TPS跌零5秒左右。

```
[ 23s ] thds: 64 tps: 2257.82 qps: 45260.44 (r/w/o: 31662.51/9077.29/4520.64) lat (ms,95%): 33.72 err/s: 0.00 reconn/s: 0.00
[ 24s ] thds: 64 tps: 2326.91 qps: 46344.22 (r/w/o: 32464.75/9227.64/4651.82) lat (ms,95%): 32.53 err/s: 0.00 reconn/s: 0.00
[ 25s ] thds: 64 tps: 2283.15 qps: 45898.11 (r/w/o: 32156.18/9171.62/4570.31) lat (ms,95%): 32.53 err/s: 0.00 reconn/s: 0.00
[ 26s ] thds: 64 tps: 2277.45 qps: 45298.86 (r/w/o: 31683.20/9060.77/4554.89) lat (ms,95%): 33.12 err/s: 0.00 reconn/s: 0.00
[ 27s ] thds: 64 tps: 2272.40 qps: 45662.89 (r/w/o: 32006.51/9111.58/4544.79) lat (ms,95%): 32.53 err/s: 0.00 reconn/s: 0.00
[ 28s ] thds: 64 tps: 2299.27 qps: 46039.50 (r/w/o: 32234.85/9207.10/4597.55) lat (ms,95%): 32.53 err/s: 0.00 reconn/s: 0.00
[ 29s ] thds: 64 tps: 2222.77 qps: 44138.48 (r/w/o: 30825.84/8867.09/4445.54) lat (ms,95%): 34.33 err/s: 0.00 reconn/s: 0.00
[ 30s ] thds: 64 tps: 2259.98 qps: 45548.56 (r/w/o: 31980.69/9045.91/4521.96) lat (ms,95%): 33.12 err/s: 0.00 reconn/s: 0.00
[ 31s ] thds: 64 tps: 2274.05 qps: 45362.07 (r/w/o: 31722.75/9097.21/4542.11) lat (ms,95%): 33.12 err/s: 0.00 reconn/s: 0.00
[ 32s ] thds: 64 tps: 2334.92 qps: 46711.43 (r/w/o: 32680.90/9355.68/4674.84) lat (ms,95%): 31.94 err/s: 0.00 reconn/s: 0.00
[ 33s ] thds: 64 tps: 2318.20 qps: 46435.94 (r/w/o: 32517.76/9284.79/4633.39) lat (ms,95%): 31.94 err/s: 0.00 reconn/s: 0.00
[ 34s ] thds: 64 tps: 2309.92 qps: 46073.39 (r/w/o: 32213.87/9236.68/4622.84) lat (ms,95%): 32.53 err/s: 0.00 reconn/s: 0.00
[ 35s ] thds: 64 tps: 2331.92 qps: 46619.46 (r/w/o: 32655.92/9300.69/4662.85) lat (ms,95%): 31.94 err/s: 0.00 reconn/s: 0.00
[ 36s ] thds: 64 tps: 888.25 qps: 17879.04 (r/w/o: 12529.53/3572.01/1777.50) lat (ms,95%): 31.37 err/s: 0.00 reconn/s: 0.00
[ 37s ] thds: 64 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,95%): 0.00 err/s: 0.00 reconn/s: 0.00
[ 38s ] thds: 64 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,95%): 0.00 err/s: 0.00 reconn/s: 0.00
[ 39s ] thds: 64 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,95%): 0.00 err/s: 0.00 reconn/s: 0.00
[ 40s ] thds: 64 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,95%): 0.00 err/s: 0.00 reconn/s: 0.00
[ 41s ] thds: 64 tps: 0.00 qps: 0.00 (r/w/o: 0.00/0.00/0.00) lat (ms,95%): 0.00 err/s: 0.00 reconn/s: 0.00
[ 42s ] thds: 64 tps: 69.99 qps: 1462.81 (r/w/o: 1043.87/278.96/139.98) lat (ms,95%): 6594.16 err/s: 0.00 reconn/s: 0.00
[ 43s ] thds: 64 tps: 1061.02 qps: 21101.34 (r/w/o: 14733.24/4247.07/2121.03) lat (ms,95%): 28.67 err/s: 0.00 reconn/s: 0.00
[ 44s ] thds: 64 tps: 1144.81 qps: 22881.15 (r/w/o: 16011.31/4580.23/2289.61) lat (ms,95%): 28.67 err/s: 0.00 reconn/s: 0.00
[ 45s ] thds: 64 tps: 2385.06 qps: 47750.12 (r/w/o: 33432.79/9546.22/4771.11) lat (ms,95%): 29.72 err/s: 0.00 reconn/s: 0.00
[ 46s ] thds: 64 tps: 2324.10 qps: 46336.94 (r/w/o: 32384.35/9311.39/4641.19) lat (ms,95%): 33.12 err/s: 0.00 reconn/s: 0.00
[ 47s ] thds: 64 tps: 2309.83 qps: 46180.51 (r/w/o: 32356.55/9198.30/4625.65) lat (ms,95%): 30.26 err/s: 0.00 reconn/s: 0.00
[ 48s ] thds: 64 tps: 2318.07 qps: 46553.51 (r/w/o: 32623.06/9293.30/4637.15) lat (ms,95%): 29.72 err/s: 0.00 reconn/s: 0.00
[ 49s ] thds: 64 tps: 2292.02 qps: 45800.44 (r/w/o: 32053.31/9165.09/4582.04) lat (ms,95%): 29.72 err/s: 0.00 reconn/s: 0.00
[ 50s ] thds: 64 tps: 2284.95 qps: 45601.99 (r/w/o: 31910.29/9124.80/4566.90) lat (ms,95%): 29.19 err/s: 0.00 reconn/s: 0.00
[ 51s ] thds: 64 tps: 2168.99 qps: 43422.82 (r/w/o: 30406.87/8672.96/4342.98) lat (ms,95%): 29.19 err/s: 0.00 reconn/s: 0.00
```

视频展示

热备RO与普通RO主动运维的表现对比

本视频展示了在主动运维场景下，普通RO与热备RO的性能差异对比。从视频中实验结果可知，在主动运维场景（如主节点切换、规格升降配）下，热备RO的业务中断时间更短，且可以实现事务不中断。

热备RO与普通RO故障容灾的表现对比

本视频展示了在故障容灾场景下，普通RO与热备RO的性能差异对比。从视频中实验结果可知，在故障容灾 failover 场景（如主节点宕机）下，热备RO可以实现连接和事务不中断，且不出现中断报错。

11.5. 设置热备节点

创建集群时，不可设置热备节点，只能在节点配置中为集群中已有的只读节点开启热备功能。

前提条件

仅PolarDB MySQL引擎集群版5.7、8.0版本支持热备功能。

注意事项

- 当只读节点未开启热备时，主备切换过程中可能会出现20~30秒左右的闪断，因此切换前请务必确保应用具备重连机制；当只读节点开启了热备功能时，主备切换将在3~10秒内完成。
- 只读节点开启热备功能后，该节点将不再提供读服务。
- 热备节点规格需要与主节点规格保持一致。
- 热备切换功能与**列存索引**功能互斥，若集群中已存在只读列存节点，则该集群中的任何只读节点都不支持开启热备功能。

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，单击集群ID。
4. 在**基本信息**页面的**数据库节点**区域，单击右上角图标切换视图。

5. 在数据库节点区域，定位到需要设置热备功能的只读节点，单击操作列的开启热备。

数据库节点

自动扩容: 已关闭 设置 扩容历史

新增节点 变更配置 主备切换

节点名称	可用区	状态	当前角色	规格	最大IOPS	Failover 优先级	操作
pi-xxxxxx	新加坡 可用区A	运行中	主节点	4核 16GB	48000	1	重启
pi-xxxxxx	新加坡 可用区A	运行中	只读节点	4核 16GB	48000	1	重启 开启热备

说明 第一次开启热备功能时，集群中的所有节点会依次重启，后续开启或关闭热备功能时，节点无需再进行重启。

12. 闪回查询

通过闪回查询（Flashback Query）功能，您可以高效查询实例、数据库、数据表在过去某个时间点的信息。

语法结构

```
SELECT <column_name_list> FROM <table_name> AS OF <TIMESTAMP>;
```

前提条件

PolarDB集群版本需满足如下条件之一：

- PolarDB MySQL引擎8.0.2版本且修订版本为8.0.2.2.2及以上
- PolarDB MySQL引擎8.0.1版本且修订版本为8.0.1.1.24及以上
- PolarDB MySQL引擎5.7版本且修订版本为5.7.1.0.21及以上
- PolarDB MySQL引擎5.6版本且修订版本为5.6.1.0.32及以上

如何确认集群版本，详情请参见[查询版本号](#)。

语法说明

- 使用该功能前，您需要在PolarDB集群的参数配置中先开启 `innodb_backquery_enable` 参数。

 **说明** 如果您在开启 `innodb_backquery_enable` 参数之前执行闪回查询，系统会返回报错 `ERROR 1815 (HY000): Internal error: the backquery_time set is out of range, too old`。

- PolarDB提供了以下参数来更精确的控制闪回功能：

参数名称	数据类型	说明
<code>innodb_backquery_enable</code>	BOOL	打开或者关闭闪回查询功能。取值： ◦ ON：打开 ◦ OFF：关闭（默认）
<code>innodb_backquery_window</code>	ULONG	闪回查询支持的时间长度。 取值范围：1~2592000。单位为秒。默认值为86400s。

语法示例

- 准备测试数据：
在 `2021-08-31 13:51` 创建 `products` 表，并插入数据。

```

create table products (
    prod_id bigint(10) primary key NOT NULL,
    prod_name varchar(20) NOT NULL,
    cust_id bigint(10) NULL,
    createtime datetime NOT NULL DEFAULT NOW()
);
INSERT INTO products(prod_id,prod_name,cust_id,createtime)
values
(101,'Book',1,NOW()), (102,'Apple',1,NOW()), (103,'Beef',2,NOW()), (104,'Bread',3,NOW()), (105,'Cheese',4,NOW());

```

- 查询 products 表中的数据：

```

SELECT * FROM products;
+-----+-----+-----+-----+
| prod_id | prod_name | cust_id | createtime          |
+-----+-----+-----+-----+
| 101 | Book | 1 | 2021-08-31 13:51:22 |
| 102 | Apple | 1 | 2021-08-31 13:51:24 |
| 103 | Beef | 2 | 2021-08-31 13:51:26 |
| 104 | Bread | 3 | 2021-08-31 13:51:27 |
| 105 | Cheese | 4 | 2021-08-31 13:51:29 |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

```

- 更新测试数据：

在 2021-08-31 14:18 对 products 表数据进行了更新。

```

UPDATE products SET prod_id = 110, createtime = NOW() WHERE prod_name = "Book";
UPDATE products SET prod_id = 119, createtime = NOW() WHERE prod_name = "Apple";
SELECT * FROM products;
+-----+-----+-----+-----+
| prod_id | prod_name | cust_id | createtime          |
+-----+-----+-----+-----+
| 103 | Beef | 2 | 2021-08-31 13:51:26 |
| 104 | Bread | 3 | 2021-08-31 13:51:27 |
| 105 | Cheese | 4 | 2021-08-31 13:51:29 |
| 110 | Book | 1 | 2021-08-31 14:18:21 |
| 119 | Apple | 1 | 2021-08-31 14:18:22 |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

```

- 执行闪回查询：

查看 products 表中 2021-08-31 14:00:00 这个历史时间点的数据。

```
SELECT * FROM products AS of TIMESTAMP '2021-08-31 14:00:00';
```

```
+-----+-----+-----+-----+
| prod_id | prod_name | cust_id | createtime |
+-----+-----+-----+-----+
|      101 | Book      |      1 | 2021-08-31 13:51:22 |
|      102 | Apple     |      1 | 2021-08-31 13:51:24 |
|      103 | Beef      |      2 | 2021-08-31 13:51:26 |
|      104 | Bread     |      3 | 2021-08-31 13:51:27 |
|      105 | Cheese    |      4 | 2021-08-31 13:51:29 |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

13.高可用部署架构

13.1. 多可用区部署架构

13.1.1. 多可用区部署

PolarDB MySQL引擎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。

前提条件

- 集群所在的地域需要包含两个及以上的可用区。
- 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。



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。关于如何新建集群，详情请参见[购买按量付费集群](#)和[购买包年包月集群](#)。

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。

多可用区费用说明

多可用区功能暂时不需要支付额外费用。

说明 当前单可用区集群也可免费升级至多可用区集群。

查看集群所属可用区

在集群的基本信息页面，可查看数据分布的可用区。

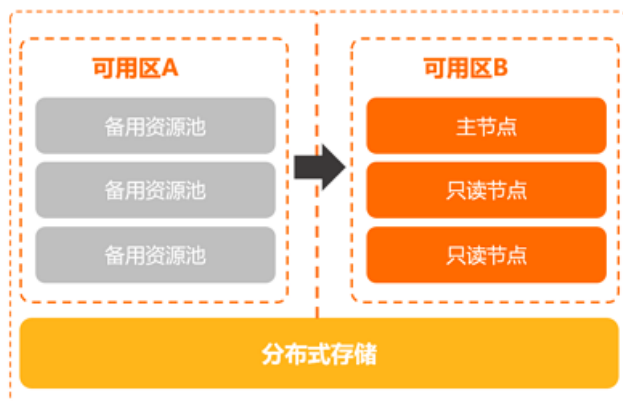
地域	华东1（杭州）	VPC		付费类型	按小时付费		
数据分布的可用区	杭州可用区I（主），杭州可用区H	交换机		创建时间	2020年9月24日 17:36:46		
兼容性	100% 兼容 MySQL 8.0	可维护窗口	02:00-03:00	系列	标准版		

13.1.2. 更换主可用区和交换机

PolarDB支持更换主可用区和交换机，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。

使用场景和注意事项

更换集群主可用区的功能适用于灾难恢复或者让ECS就近访问的场景。



- 主备可用区切换过程中，会出现1~2次约30秒的连接闪断。建议您在业务低谷期执行切换操作并确保应用具备重连机制。
- 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。
- 如果目标可用区不是备可用区，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。
- 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。

操作步骤

- 登录PolarDB控制台。
- 在控制台左上角，选择集群所在地域。
- 找到目标集群，单击集群ID。
- 在基本信息页面，单击更换主可用区。



- 在弹出的对话框中，选择目标可用区和目标交换机，并根据业务需要选择生效时间。



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。
- 如果目标可用区不是备可用区，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。
- 若目标可用区无交换机，您需要先[创建交换机](#)。
- 您可以选择生效时间为立即生效或可维护窗口生效。若选择在可维护窗口生效，您还可以在计划任务页查看该计划任务的具体信息或取消该任务，详情请参见[查看或取消计划任务](#)。

6. 单击**确定**即可。

常见问题

- Q：需要多长时间才能完成可用区的更换？
- 若您选择的目标可用区是备可用区，则更换主可用区时不需要迁移数据。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。
- 若您选择的目标可用区不是备可用区，则更换主可用区时还需要迁移数据。系统执行迁移时间的长短跟数据容量有关，若数据量较大，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。



- Q：更换主可用区所需的时间是否等于业务不可用时间？例如当需要将备可用区更换为目标主可用区时，平均耗时为5分钟/节点，若集群中有4个节点，是否意味着业务会有约20分钟的不可用时间？
A：更换主可用区所需的时间不等于业务不可用时间。整个切换过程中，仅在主备切换时会出现1~2次约30秒的闪断时间。建议您在业务低谷期执行并确保应用具备重连机制。
- Q：在更换主可用区的过程中，集群是如何保持正常运行的？

A：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，因此您仍然可通过数据库连接地址正常访问集群。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。

13.2. 多节点部署架构

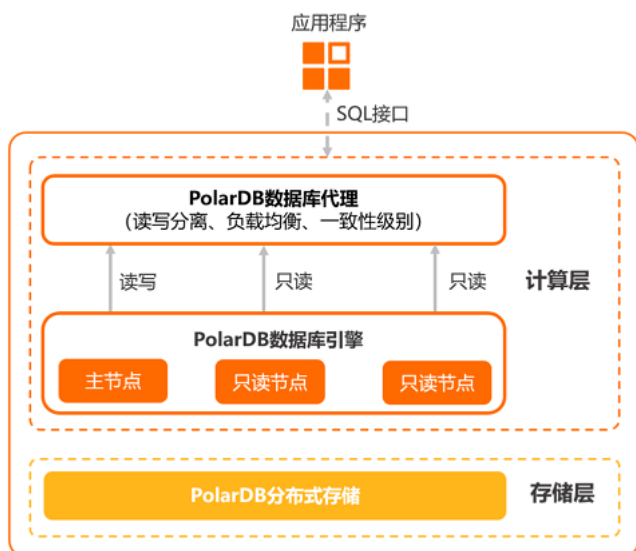
13.2.1. 多节点部署

PolarDB集群版的主可用区内包含了多个节点（计算节点），其中一个节点是主节点，其他节点为只读节点。

使用限制

仅PolarDB集群版和历史库集群版支持多节点，单节点和历史库单节点版不支持。更多关于产品系列的功能对比，请参见[产品系列](#)。

多节点架构



每个PolarDB集群都包含了1个主节点和多个只读节点。其中，只读节点最多15个，最少1个。同一集群中，所有节点的规格相同。

多节点的架构可用于保障集群的高可用，当系统发生故障时，可读写的主节点和只读节点之间会自动进行故障切换（Failover）。

此外，通过PolarDB的数据库代理功能，可在这些节点的基础上实现读写分离等功能，具体可参见[数据库代理](#)。

增加/删除只读节点

您可以根据实际需要手动增加或删除只读节点，进而调整集群的性能。具体操作可参见[增加或删除节点](#)。

说明

- 增加一个只读节点预计耗时5分钟，具体耗时受新增节点数量、库表数量、数据库负载等因素影响。增加节点的过程中，对数据库无任何影响。
- 删除只读节点时，该节点上的连接会发生闪断，其他节点不受影响。建议您在业务低谷期执行删除节点操作，并确保应用具备自动重连机制。
- 推荐使用默认集群地址连接应用，当增加/删除只读节点后PolarDB会自动进行调整，不需要修改应用配置：当增加只读节点后，PolarDB支持自动识别新节点，并自动分流到新节点，达到负载均衡；当删除只读节点后，PolarDB支持自动屏蔽失效节点。

13.2.2. 自动/手动主备切换

当系统发生故障时，PolarDB集群会自动进行主备切换。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。

注意事项

不论是自动切换还是手动切换，当只读节点未开启热备时，主备切换过程中可能会出现20~30秒左右的闪断，因此切换前请务必确保应用具备重连机制；当只读节点开启了热备功能时，主备切换将在5~10秒内完成。关于如何开启只读节点的热备功能，请参见[设置热备节点](#)。

自动主备切换

PolarDB集群版采用双活（Active-Active）的高可用集群架构。当系统发生故障时，可读写的主节点和只读节点之间会自动进行故障切换（Failover），系统自动选举新的主节点。

集群中每个节点都有一个故障切换（Failover）优先级，该优先级决定了故障切换时每个节点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：

- 系统找出当前可以被选举的所有只读节点。
- 选择优先级最高的一个或多个只读节点。
- 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。

您可在集群基本信息页面的数据库节点区域，查看并设置集群中每个节点的Failover优先级。

数据库节点							自动扩容: 已关闭	设置	扩容历史	☰
新增节点 变更配置 主备切换										
节点名称	可用区	状态	当前角色	规格	最大IOPS	Failover 优先级				
pi-...	新加坡 可用区A	运行中	主节点	4核 16GB	48000	1				
pi-...	新加坡 可用区A	运行中	只读节点	4核 16GB	48000	1				

说明 当只读节点未开启热备时，主备切换过程中可能会出现20~30秒左右的闪断，因此切换前请务必确保应用具备重连机制；当只读节点开启了热备功能时，主备切换将在5~10秒内完成。

手动主备切换

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。手动切换适用于高可用演练，或者需要指定某个节点为主节点的场景。

- 登录[PolarDB控制台](#)。
- 在控制台左上角，选择集群所在地域。

- 找到目标集群，单击集群ID。
- 在基本信息页面的数据库节点区域，单击右上角图标切换视图。
- 单击主备切换。



- 在弹出的对话框中，选择新主节点，单击确定，开始切换。

说明

- 如果新主节点为未开启热备功能的只读节点，切换过程中可能会出现20~30秒左右的闪断，因此切换前请务必确保应用具备重连机制；
- 如果新主节点为开启热备功能的只读节点，主备切换将在5~10秒内完成。

相关API

API	描述
FailoverDBCluster	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。

14.多主架构

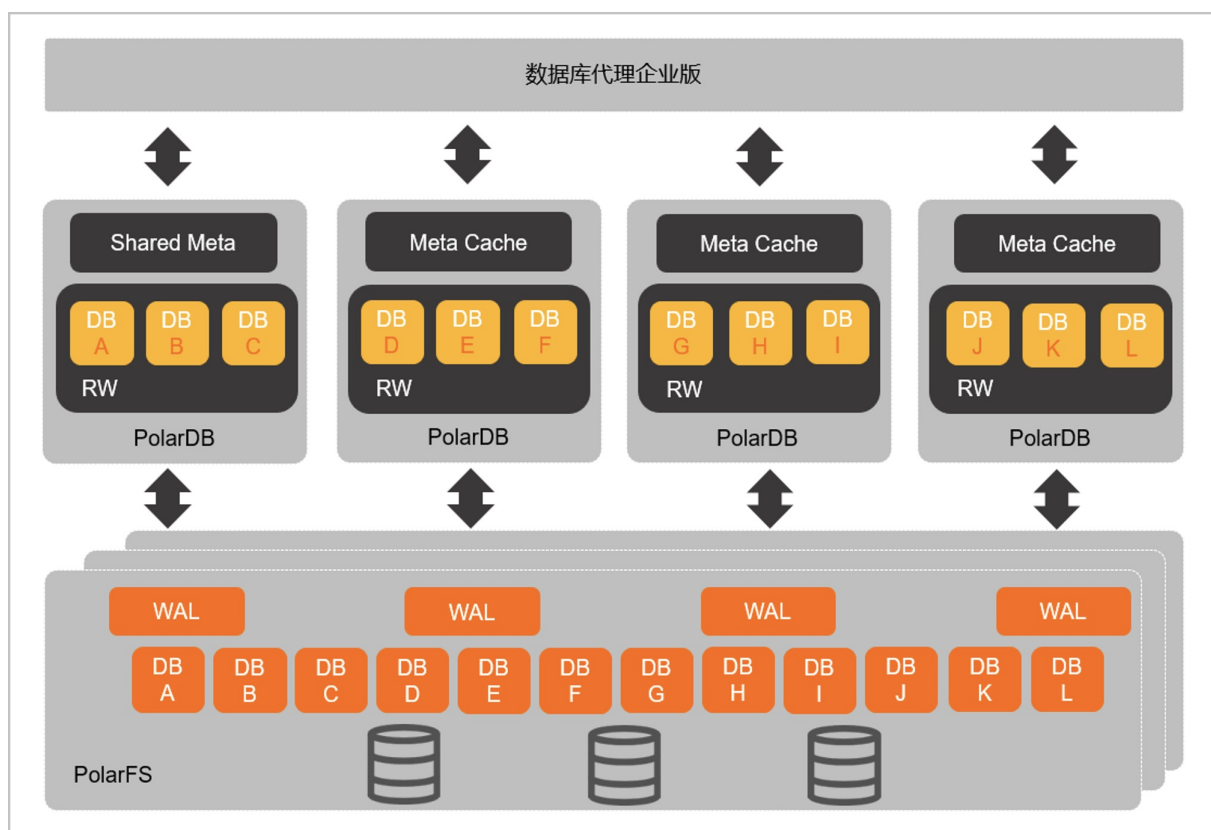
14.1. 多主架构集群版

本文将详细介绍多主架构集群版的相关信息。

随着PolarDB MySQL引擎客户的不断增加，大规模头部客户不断涌入，部分头部客户业务体量规模庞大，使得目前PolarDB MySQL引擎的单写（一写多读）架构在特定场景下，写性能出现瓶颈。

PolarDB MySQL引擎全新推出多主架构，实现从一写多读架构到多写多读架构的升级，主要面向多租户、游戏、电商等高并发读写的应用场景。

多主架构的架构图如下：



集群中所有的数据文件都存放在共享存储（PolarStore）中，各个RW节点通过分布式文件系统（PolarFileSystem）共享底层存储（PolarStore）中的数据文件。用户可以通过集群地址访问整个集群，数据库代理会自动转发SQL命令到正确的RW节点。

核心优势和能力

- 支持不同数据库在不同计算节点并发写入。
- 集群版支持1个写节点和最多15个只读节点，多主架构集群版最多可支持32个节点同时写入。
- 支持数据库跨节点动态调度，秒级完成切换，极大提升集群整体并发读写能力。

适用场景

多主架构主要面向SaaS多租户、游戏、电商等高并发读写的应用场景。

- SaaS多租户场景：满足高并发性能需求，实现租户间负载均衡

场景特点：租户的数据库数量变化较快，负载变化较大，需要经常在不同的实例之间调配数据库资源，以便达到最佳用户体验。

解决方案：多主架构可帮助客户快速将租户的数据库在不同RW节点间进行切换，从而实现负载均衡。

- **分服游戏场景：更好的性能和扩展能力，支持世界服架构**

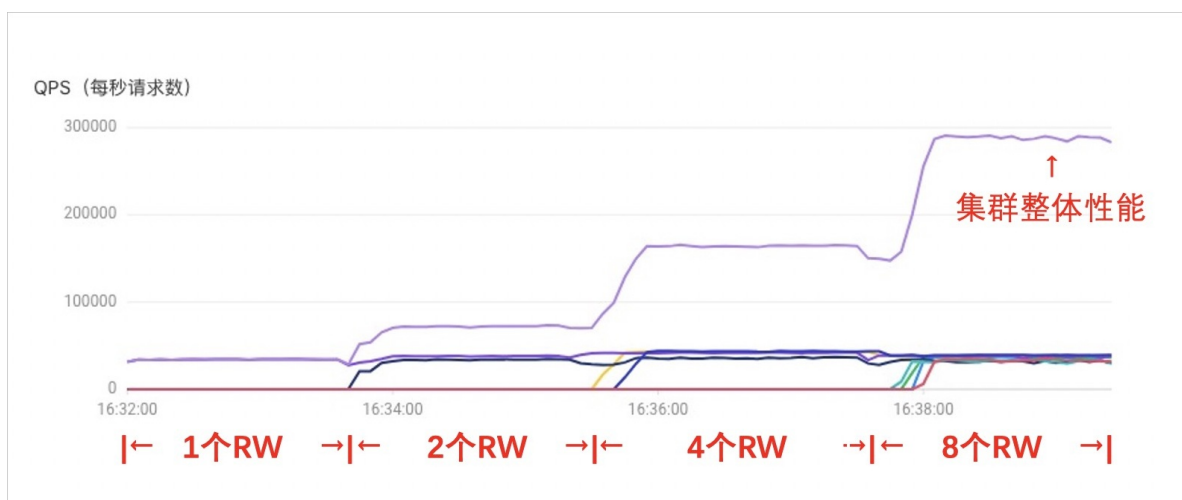
场景特点：在游戏成长期，数据库负载较大，且呈现为不断增长的趋势特点。通常表现为在游戏成长期期间，会不断增加数据库，导致RW节点负荷也不断增加。而在游戏衰退期，数据库负载逐渐减少，数据库会不断合并，导致RW节点的负荷也呈减少趋势。

解决方案：游戏成长期，可快速将部分数据库切换到新的RW节点，实现负载均衡；游戏衰退期，可快速将数据库聚合到少量RW节点，快速降低运作成本。

性能提升情况

经测试，随着集群中的数据库切换至更多的主节点（RW）上，集群整体并发读写能力几乎呈线性提升。测试示例如下：

- 测试背景：集群包含8个数据库，8个RW节点。
- 测试过程：初始情况下，8个数据库全部负载在其中一个RW节点上，然后对所有数据库同步执行相同的压力测试。压测期间，将8个数据库分别平均切换到2个RW节点、4个RW节点、8个RW节点上，观察集群整体的性能变化趋势。
- 性能变化趋势如下，以QPS为例：



从上图可以看出，随着数据库切换至更多的RW上时，集群整体并发读写能力得到了极大的提升，几乎呈现为线性提升。

支持的内核版本

目前仅PolarDB MySQL引擎8.0内核版本支持多主架构集群版。

节点规格与定价

多主架构集群版支持7种独享规格和6种通用规格的节点。更多详情，请参见[计算节点规格](#)。

关于多主架构集群版的计费详情，请参见[产品计费](#)。

如何使用

目前该产品系列还处于灰度发布阶段，如需使用请[提交工单](#)申请开通。具体使用方法，请参见[使用说明](#)。

❓ 说明 多主架构集群版当前仅面向企业用户灰度开放。

14.2. 使用说明

PolarDB MySQL引擎多主架构集群版实现从一写多读架构到多写多读架构的升级；支持不同数据库在不同计算节点并发写入；支持数据库跨节点动态调度，秒级完成切换，极大提升实例整体并发读写能力。本文介绍多主架构集群版的使用说明。

前提条件

- 已购买多主架构集群版集群。具体操作请参见[购买按量付费集群](#)和[购买包年包月集群](#)。
- 已创建高权限账号。具体操作请参见[创建高权限账号](#)。
- 已连接到数据库集群。具体操作请参见[连接数据库集群](#)。

使用限制

- 每个数据库的数据只能通过一个节点写入。没有分配数据库的节点，不能进行读写操作。
- 不支持跨RW节点的数据查询。如果一个查询SQL中包含多个RW节点上的数据库，系统则会报错。建议您先把所有数据库的访问点调整到其中一个RW节点上，再进行查询操作。
- 仅提供集群地址，不支持主地址。

创建数据库时指定RW节点

在指定的RW节点上创建数据库。语法结构如下：

```
CREATE DATABASE name [POLARDB_WRITE_NODE master_id];
```

 **说明** 每个数据库的数据只能通过一个节点写入。没有分配数据库的节点，不能进行读写操作。

示例：在RW1上创建一个数据库 `db1`

```
CREATE DATABASE db1 POLARDB_WRITE_NODE 1;
```

如果需要在RW2上创建数据库 `db1`，把上述示例中的1换成2即可。

删除在指定RW节点上创建的数据库

删除在指定RW节点上创建的数据库。语法结构如下：

```
DROP DATABASE name;
```

示例：删除在RW1节点上创建的数据库 `db1`

```
DROP DATABASE db1;
```

删除数据库时，无需指定POLARDB_WRITE_NODE。

RW节点切换

把数据库的访问点切换到其他RW节点上。语法结构如下：

```
ALTER DATABASE name POLARDB_WRITE_NODE master_id;
```

示例：把数据库 `db1` 切换到RW2上

```
ALTER DATABASE db1 POLARDB_WRITE_NODE 2;
```

指定SQL语句执行的RW节点

 **注意** 该功能仅适用于非数据查询的语句，例如查询information_schema、查询status变量等。如果需要查询数据，例如查询语句 `SELECT * FROM table1`，不需要指定RW节点，数据库代理会自动选择正确的RW节点执行查询。

如果需要把某条SQL语句发送到指定的RW节点，则需要执行以下SQL语句锁定某个RW节点：

```
ALTER SESSION POLARDB_WRITE_NODE master_id;
```

示例：查询RW1节点上 `innodb_buffer_pool_size` 这个变量的值

```
ALTER SESSION POLARDB_WRITE_NODE 1;    #把SQL语句发送到RW1节点。
SHOW VARIABLES LIKE 'innodb_buffer_pool_size'; #查询RW1节点上innodb_buffer_pool_size的值。
```

 **说明** 如果在执行某条SQL语句时不指定RW节点，那么数据库代理会随机选择一个RW节点执行该SQL语句。

执行以下命令可以解锁指定执行SQL语句的RW节点：

```
RESET SESSION POLARDB_WRITE_NODE;
```

查询节点信息

- 执行以下命令可以查询某个RW节点上的数据库分布情况：

```
ALTER SESSION POLARDB_WRITE_NODE master_id;
SELECT * FROM INFORMATION_SCHEMA.INNODB_MASTER_GLOBAL_LOCK_INFO;
```

示例：查询RW1节点上的数据库分布情况

```
ALTER SESSION POLARDB_WRITE_NODE 1;
SELECT * FROM INFORMATION_SCHEMA.INNODB_MASTER_GLOBAL_LOCK_INFO;
```

查询结果如下：

```

+-----+-----+-----+-----+-----+-----+
| table_name          | table_id          | space_id | s_lock_count | lock_mode | current_lsn |
+-----+-----+-----+-----+-----+-----+
| mysql/global_ddl_lock | 1043956280258737140 | 0 | 0 | SLS_X | 24311657 |
| db2                  | 27980076883382651 | 0 | 0 | SLS_X | 36087702 |
| db1                  | 27980076883383418 | 0 | 0 | SLS_X | 34339564 |
| mysql                | 3381631115268247737 | 0 | 0 | SLS_IX | 0 |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

```

上述查询结果中的每一行就是一个数据库的信息（虽然列名是table_name）。其中，mysql/global_ddl_lock、mysql 为MySQL内部使用的数据库，并非用户创建的数据库。

- 执行以下命令可以查询整个集群上所有数据库的分布情况：

 **说明** 仅支持通过高权限账号查询，不能通过用户新建的账号进行查询。

```
SELECT * FROM INFORMATION_SCHEMA.INNODB_CC_GLOBAL_LOCK_INFO;
```

查询结果如下：

```

+-----+-----+-----+-----+-----+-----+
| master_id | table_name          | table_id          | lock_mode | current_lsn |
+-----+-----+-----+-----+-----+-----+
| 2 | db5                | 27980076883382398 | SLS_X    | 18866566 |
| 1 | mysql/global_ddl_lock | 1043956280258737140 | SLS_X    | 24311657 |
| 1 | db1                | 27980076883383418 | SLS_X    | 34339564 |
| 1 | db2                | 27980076883382651 | SLS_X    | 36087702 |
| 2 | db4                | 27980076883383165 | SLS_X    | 18855954 |
| 1 | mysql              | 3381631115268247737 | SLS_IX   | 0 |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

```

上述查询结果中的每一行就是一个数据库的信息（虽然列名是table_name），表示这个数据库是在对应的RW节点（master_id）下操作的。其中，mysql/global_ddl_lock、mysql 为MySQL内部使用的数据库，并非用户创建的数据库。

查询数据库的元信息

执行以下命令可以查询所有RW节点上的表信息：

```
SELECT * FROM INFORMATION_SCHEMA.TABLES;
```

执行以下命令可以查询当前执行SQL查询的RW节点信息：

```
SELECT * FROM INFORMATION_SCHEMA.INNODB_TABLES;
```

14.3. 最佳实践

14.3.1. DBLE与PolarDB多主架构结合使用最佳实践

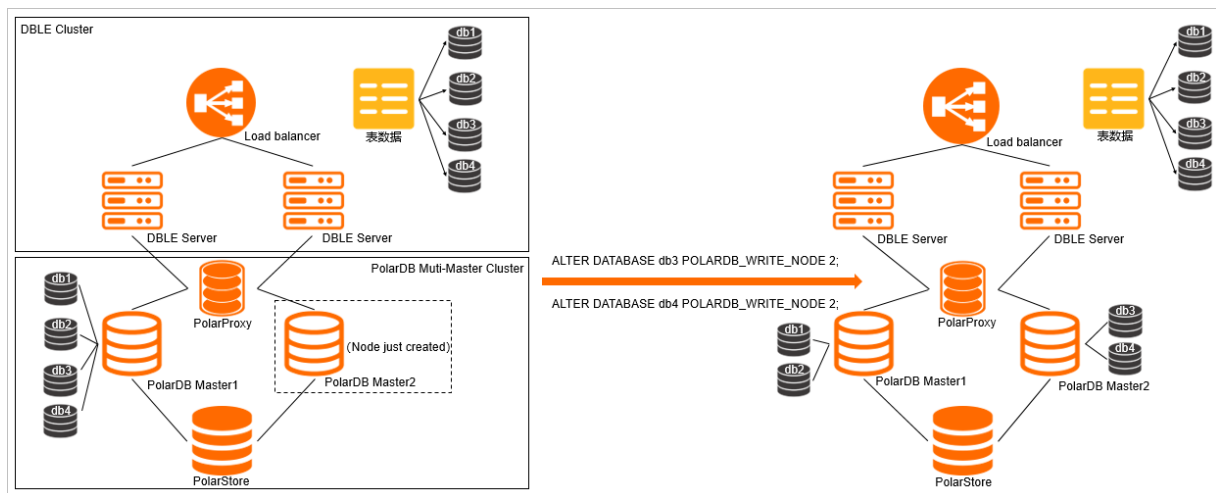
本文介绍DBLE与PolarDB多主架构结合使用的最佳实践。

原理介绍

DBLE是一个基于Java开发的数据库中间件，通过在应用和数据库之间部署一套代理，来帮助您实现非应用侵入式的分库分表。

PolarDB多主架构通过将不同的逻辑数据库分散到不同的主节点（RW），来实现读写能力水平扩展。

DBLE和PolarDB多主架构结合使用，通过DBLE将数据分散至多个数据库，再结合PolarDB多主架构实现读写能力扩展。原理图如下所示：



- 如果仅使用DBLE，在需要提升数据库集群读写能力时，需要先使用传统数据库运维手段进行数据复制和备份恢复，再配合业务系统停写、DBLE配置更新、数据验证、业务系统恢复等步骤来提升数据库集群的读写能力。其中，数据库的运维操作尤为复杂耗时。
- 使用PolarDB多主架构之后，您可以省去大部分手工扩展数据库集群的工作。您只需要通过控制台或OpenAPI为PolarDB多主架构集群增加节点即可。在业务低峰期，配合业务系统的停写，再执行 `ALTER DATABASE name POLARDB_WRITE_NODE x` 等操作，即可在数秒内将数据库集群的读写能力扩展到新增的节点上。大幅减少了数据库停写维护的时间，提高了业务系统的SLA。

DBLE的`db.xml`配置文件中可以配置数据库组（dbGroup）。建议将同一个PolarDB集群在多个数据库组中配置多次，使得数据库组与`sharding.xml`中的`shardingNode`一一对应。通过这样的配置方式，能够减少在DBLE维护的长连接中执行 `use database` 语句的频率，提高查询执行的性能。与此同时，需要确保DBLE配置的最大连接数总数不超过PolarDB集群的**最大连接数**。

注意事项

- 如果多个查询均需要开启显式事务，需要注意不同事务中操作表的顺序，确保不同的表按照同一顺序先后执行，避免因顺序问题导致死锁。
 示例：以下两个事务中，事务1会按照先后顺序去获取 `table_1` 和 `table_2` 两个表中 `id=1` 的行锁，事务2则按照相反的顺序获取行锁。如果两个事务分别在两个MySQL连接中同时执行，MySQL会判定发生死锁，为了保证其中一个事务的顺利执行，另外一个事务会被回滚。
 事务1：

```
update table_1 set c = 1 where id = 1;
update table_2 set c = 2 where id = 1;
```

事务2:

```
update table_2 set c = 1 where id = 1;  
update table_1 set c = 2 where id = 1;
```

DBLE存在死锁风险。在DBLE的实现中，一个前端连接可以对应多个后端连接，即使是将查询转发至同一分片，如果查询的表名不同，也会使用不同的后端连接。因此，上述示例中的两个事务经过了DBLE代理之后，会转换为四个MySQL连接。事务1的第二个连接，等待事务2的第一个连接释放行锁，进而导致事务1无法提交。而事务2的第二个连接，等待事务1的第一个连接释放行锁。由此两个事务形成了死锁。

 **说明** DBLE操作涉及多个表和多个分片的事务，存在**部分提交的风险**，请您谨慎评估这些风险。

- 应用系统应使用数据库连接池连接DBLE的数据端口，以便在连接断开时自动恢复。

性能测试

使用Sysbench构造基准测试，查看DBLE结合PolarDB多主架构的弹性扩容能力。

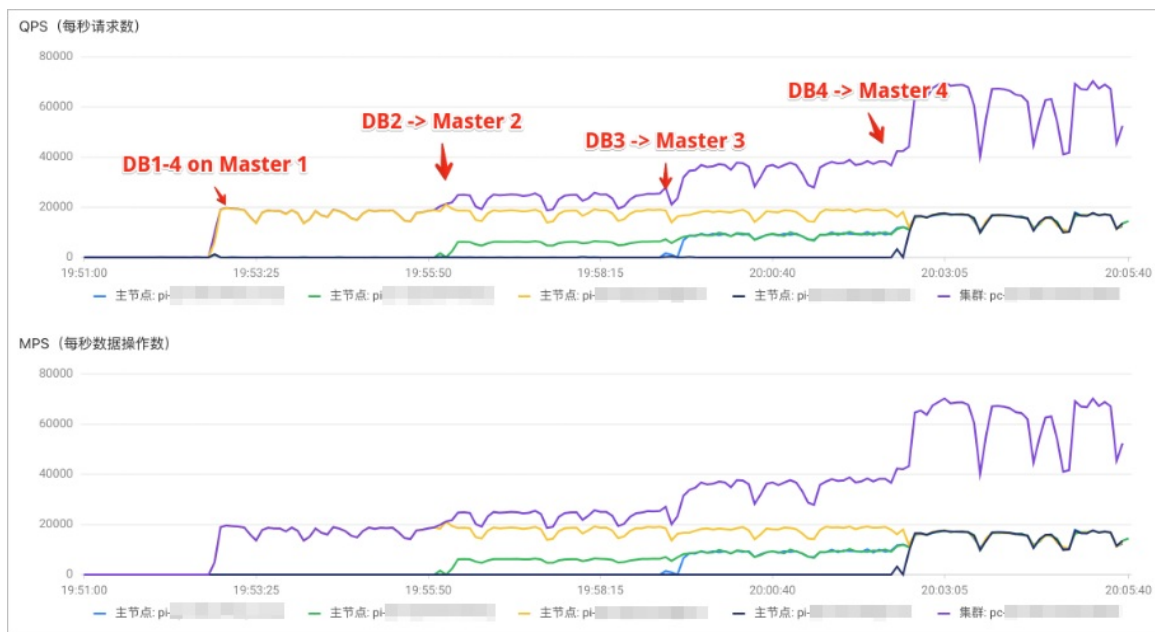
测试环境

PolarDB多主架构集群包含4个数据库，4个RW节点（4核16 GB）。DBLE以非集群部署于ECS，DBLE提供1个逻辑库4个逻辑表，每个逻辑表均按 `id%4` 散列到PolarDB多主架构集群的4个数据库上。

测试过程

初始状态下，4个数据库全部负载于其中一个RW节点，再使用Sysbench启动256个线程对所有逻辑表执行基准测试。测试期间，将3个数据库依次切换至空闲的RW节点，最终每个RW节点上均有且仅有一个数据库。查看集群整体的性能变化趋势。

- 如果业务的写入以自动提交（`autoCommit=1`）为主，PolarDB多主架构结合DBLE可以实现接近线性的性能提升。
 - QPS与MPS:



○ TPS与CPU使用率：



- 如果业务的写入以显式事务（autoCommit=0）为主，PolarDB多主架构结合DBLE可以实现仅次于线性的性能提升。以256个线程进行写入测试，弹性扩容效果如下：

○ QPS与MPS：



◦ TPS与CPU使用率：



14.3.2. Apache ShardingSphere-Proxy与PolarDB多主架构结合使用最佳实践

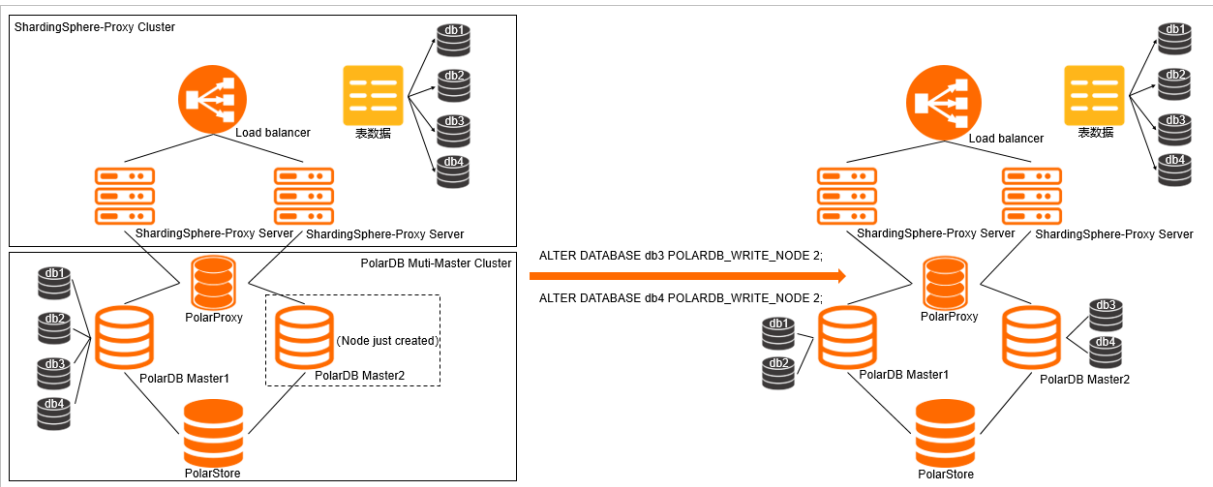
本文介绍Apache ShardingSphere-Proxy与PolarDB多主架构结合使用的最佳实践。

原理介绍

Apache ShardingSphere是开源的分布式数据库增强计算引擎，通过在应用和数据库之间部署ShardingSphere-Proxy代理，来实现非应用侵入式的分库分表。

PolarDB多主架构通过将不同的逻辑数据库分散至不同的主节点（RW），来实现读写能力水平扩展。

ShardingSphere-Proxy结合PolarDB多主架构，将数据通过ShardingSphere分散至多个数据库，再结合PolarDB多主架构实现读写能力扩展。原理图如下所示：



- 如果仅使用ShardingSphere，在需要提升数据库集群读写能力时，按照其[文档推荐的方式](#)，需要先使用传统数据库运维手段进行数据复制和备份恢复，再配合业务系统停写、规则切换、数据验证、业务系统恢复等步骤来提升数据库集群的读写能力，其中，数据库的运维操作尤为复杂耗时。

- 使用PolarDB多主架构之后，您可以省去绝大部分手工扩展数据库集群的工作，您只需要通过控制台或OpenAPI为PolarDB多主架构集群增加节点，然后在业务低峰期，执行 `ALTER DATABASE name POLARDB_WRITE_NODE x` 等操作，即可在数秒内将数据库集群的读写扩展到新增的节点上，大幅减少了数据库停写维护的时间，提高了业务系统的SLA。

PolarDB多主架构在执行 `ALTER DATABASE name POLARDB_WRITE_NODE x` 切换数据库所在的主节点时，目标数据库会有秒级的不可访问，显示为查询阻塞或获取表全局锁失败错误。如果能够在切换节点期间避免访问该数据库，或者避免访问该库的查询阻塞过久抢占访问其他库的资源，均可以减少切换数据库所在的RW节点时对业务的影响。

- 切换节点期间，避免访问切换中的数据库。

ShardingSphere借助[注册中心实现集群管理](#)，在 `/metadata/${schemeName}/dataSources` 路径下，记录了一个逻辑库的数据源。ShardingSphere会感知该路径值的变化，并动态更新和重建相关连接池。通过联动注册中心配置变更和PolarDB进行多主架构的弹性扩容，先禁用指定的数据源，再执行 `ALTER DATABASE`，最后再恢复数据源。您可以将业务影响控制在预期范围（只影响与需要迁移的数据库相关的查询）的情况下获得接近线性的数据库读写能力扩展。

- 切换节点期间，避免访问切换中的数据库时查询阻塞过久。

使用JDBC URL配置ShardingSphere的数据源，您可以在URL中配置connectTimeout和socketTimeout参数。通过将这两个超时时长配置为500ms，也可以使访问到正在切换的数据库的查询，尽可能快地超时失败，从而释放线程和连接资源去访问其他数据库。

ShardingSphere-Proxy的 `config-sharding.yaml` 配置文件中可以配置数据源，建议将每一个分库都配置成一个数据源，同时在JDBC URL中配置connectTimeout和socketTimeout参数，结合业务的属性和系统所在网络环境设置合理的超时时间，例如：

- 如果ShardingSphere-Proxy和PolarDB多主架构集群部署在同一地区，且网络RTT较低，则connectTimeout可以设置为5~10s。
- 如果业务是以在线事务为主，大部分的SQL查询都能够在1s以内返回数据，则socketTimeout也可以设置为5~10s。

一般情况下，可以借助监控系统来合理规划超时时间，将超时时间设置为业务查询耗时P99的3倍左右，既可以避免影响正常的业务查询，又可以避免局部故障阻塞过久影响整体业务。

性能测试

测试环境

PolarDB多主架构集群包含4个数据库，4个RW节点（4C16GB），ShardingSphere-Proxy以内存模式部署于ECS，ShardingSphere-Proxy提供1个逻辑库4张逻辑表，每张逻辑表均按 `id%4` 散列于PolarDB多主架构集群的4个数据库。

测试过程

初始状态下，4个数据库全部负载于其中一个RW节点，使用Sysbench启动2048个线程对所有逻辑表执行基准测试。测试期间，将3个数据库依次切换至空闲的RW节点，最终每个RW节点上均有且仅有一个数据库，查看集群整体的性能变化趋势。

- 如果业务的写入以自动提交（autoCommit=1）为主，PolarDB多主架构结合ShardingSphere-Proxy可以实现接近线性的性能提升。
 - 配置较短的超时时间。

■ QPS与MPS：

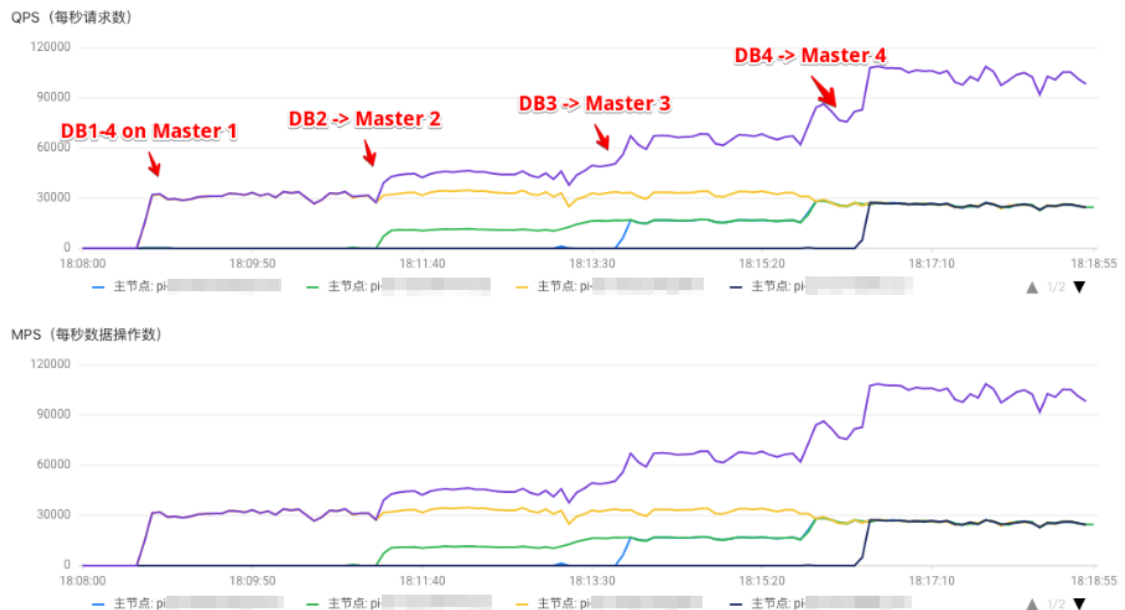


■ TPS与CPU使用率：



- 联动注册中心。

- QPS与MPS:



- TPS与CPU使用率:



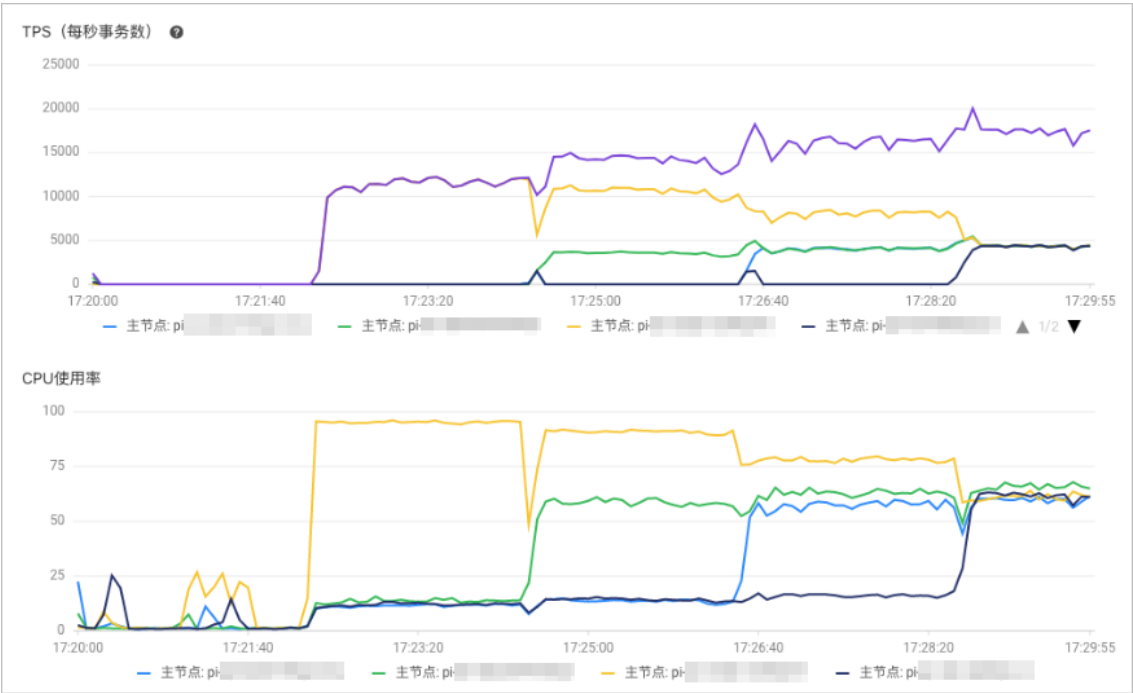
- 如果业务的写入以显式事务（autoCommit=0）为主，PolarDB多主架构结合ShardingSphere-Proxy可以实现仅次于线性的性能提升。以256个线程进行写入测试，弹性扩容效果如下：

配置较短的超时时间。

■ QPS与MPS：

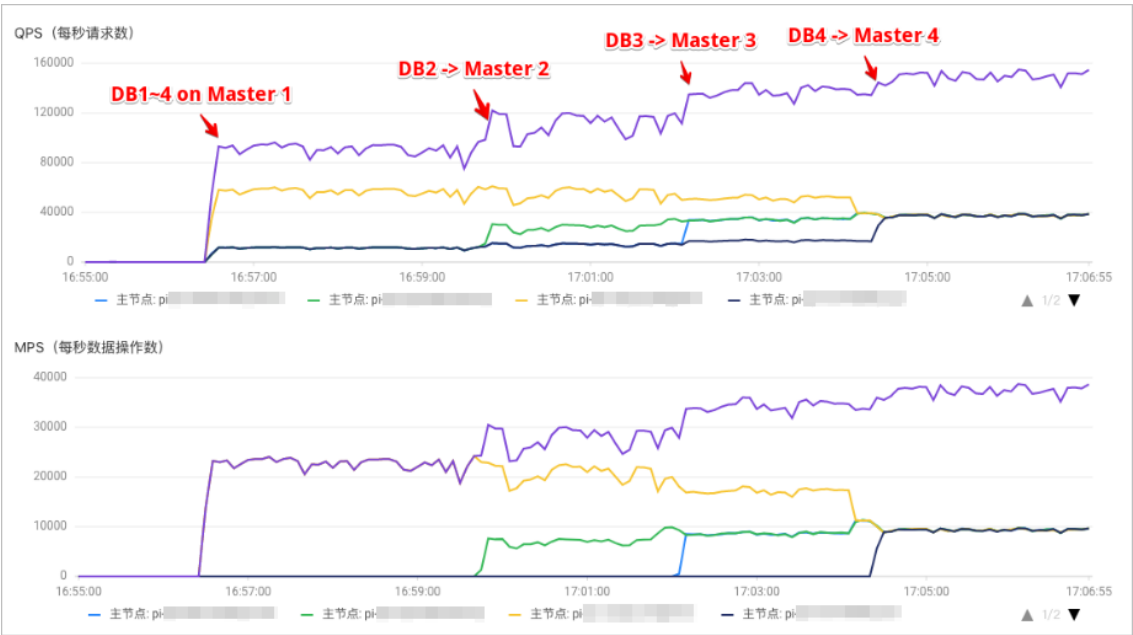


■ TPS与CPU使用率：

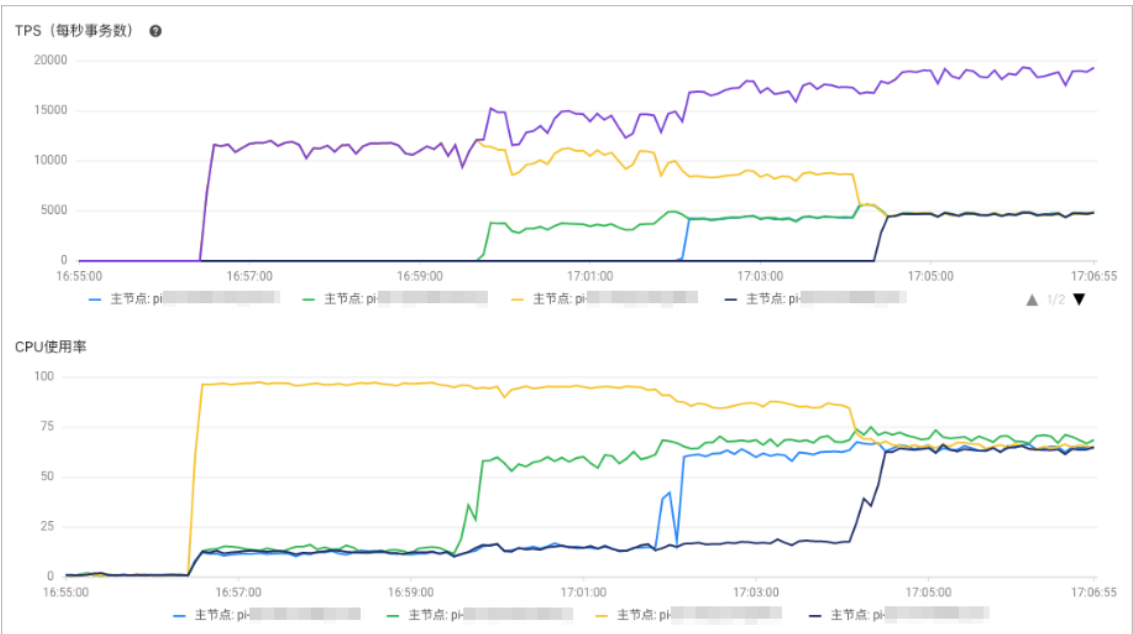


◦ 联动注册中心。

■ QPS与MPS：



■ TPS与CPU使用率：



15.历史库

15.1. 历史库

本文介绍PolarDB历史库的产品优势、架构及适用场景。

挑战和诉求：历史数据归档

- 历史数据归档的挑战

大部分业务数据的读写特征，都是最新产生的数据会被更频繁地读取或更新，而更久之前的数据（如1年前的聊天记录或订单信息）很少被访问。随着业务发展，数据库系统中会积累大量访问频率很低甚至为0的数据，这些数据的积累容易导致如下问题：

- 历史数据和最新数据存储在同一数据库系统中，导致磁盘空间不足。
- 大量数据共享数据库系统的内存、缓存空间、磁盘IOPS等，导致性能问题。
- 数据量太大导致数据备份时间过长甚至备份失败；同时如何存放备份数据也是一个问题。

针对如上问题，一种做法是对历史数据做归档，将长期不使用的数据迁移至以文件形式存储的廉价存储设备上，如阿里云OSS或者阿里云数据库DBS服务。然而，在实际业务中，历史数据并不完全是静态的，针对几个月甚至几年前的历史数据，依旧可能存在实时地、低频地查询甚至更新需求。例如，在阿里巴巴内部，对淘宝或天猫历史订单的查询、对企业级办公软件钉钉历史聊天记录的查询或对菜鸟海量历史物流订单的查询等。

- 历史数据归档的诉求

为了解决历史数据的读取和更新问题，可以使用一个单独的数据库用来存储归档的数据，即历史库。业务对单独的历史库一般有如下诉求：

- 具备大容量存储空间，支持业务持续将线上数据保存到历史库中，而无需担心容量问题。
- 与在线数据库系统使用相同的访问接口，如都支持MySQL协议等，确保应用程序端无需修改任何代码即可同时访问在线库和历史库。
- 成本低廉，如支持通过压缩减少数据所占磁盘空间、使用廉价存储介质等，确保可以使用较小的代价保存海量的数据。
- 具备一定的读写能力，能够满足低频读写的需求。

MySQL作为世界上使用最广泛的开源数据库系统，一直缺乏一个既能满足大容量低成本要求，又具备一定读写能力的历史数据归档存储方案。虽然业界曾经推出过一些高压压缩引擎，如TokuDB、MyRocks等，但受限于单物理机磁盘容量限制，存储的数据量有限。

解决方案：PolarDB历史库

为应对如上历史数据归档存储方面的挑战和诉求，PolarDB基于如下技术创新和突破，推出了历史库产品系列：

- 阿里巴巴自研的基于LSM-tree架构的存储引擎X-Engine提供了强大的数据压缩能力，满足了归档数据库低存储成本的要求。通过LSM-Tree（Log-Structured Merge-Tree）层次化架构和Zstandard（ZSTD）压缩算法实现了更高的数据压缩率，对比使用InnoDB作为存储引擎，最高可节省70%的存储空间。更多关于X-Engine存储引擎的详情，请参见[X-Engine简介](#)。但由于采用了X-Engine引擎，在使用历史库时也存在一些限制（尤其是与MySQL的兼容性限制），具体限制请参见[使用限制](#)。
- PolarDB借助于共享分布式存储服务，实现了存储容量在线平滑扩容，同时计算资源和存储资源间采用高速网络互联，并通过RDMA协议进行数据传输，使I/O性能不再成为瓶颈。集成到PolarDB的X-Engine引擎同样获得了这些技术优势。
如下技术创新将X-Engine移植进PolarDB，从而进入PolarDB双引擎时代：
 - 合并X-Engine的事务WAL日志流和InnoDB的REDO日志流，实现了一套日志流和传输通道同时服务于InnoDB引擎和X-Engine引擎，管控逻辑以及与共享存储的交互逻辑无需做任何改变，同时未来新增其他

引擎时也可以复用这套架构。

- 将X-Engine的IO模块对接到PolarDB InnoDB所使用的用户态文件系统PFS上，实现了InnoDB与X-Engine共享同一个分布式块设备。同时依靠底层分布式存储实现了快速备份。

历史库计算节点架构

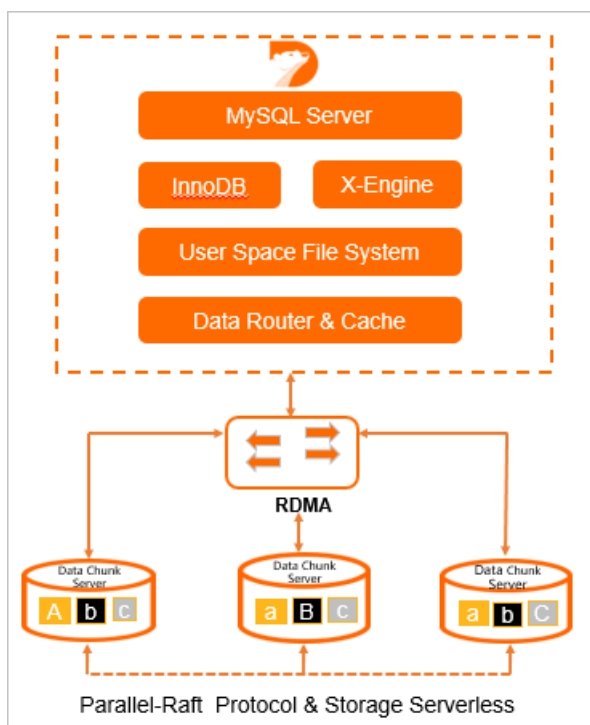
历史库支持历史库单节点版和历史库集群版两个版本。历史库单节点版采用单节点架构来提供服务。历史库集群版采用多节点架构基于共享存储实现了一写多读，集群中有一个主节点（可读可写）和至少一个只读节点，支持独享规格和通用规格两个子系列。

❓ 说明 当前已不支持购买历史库单节点版，但存量的历史库单节点版可继续使用。您可以将历史库单节点版升级到历史库集群版独享规格，具体操作请参见[历史库单节点版升级为历史库集群版](#)。

历史库单节点版

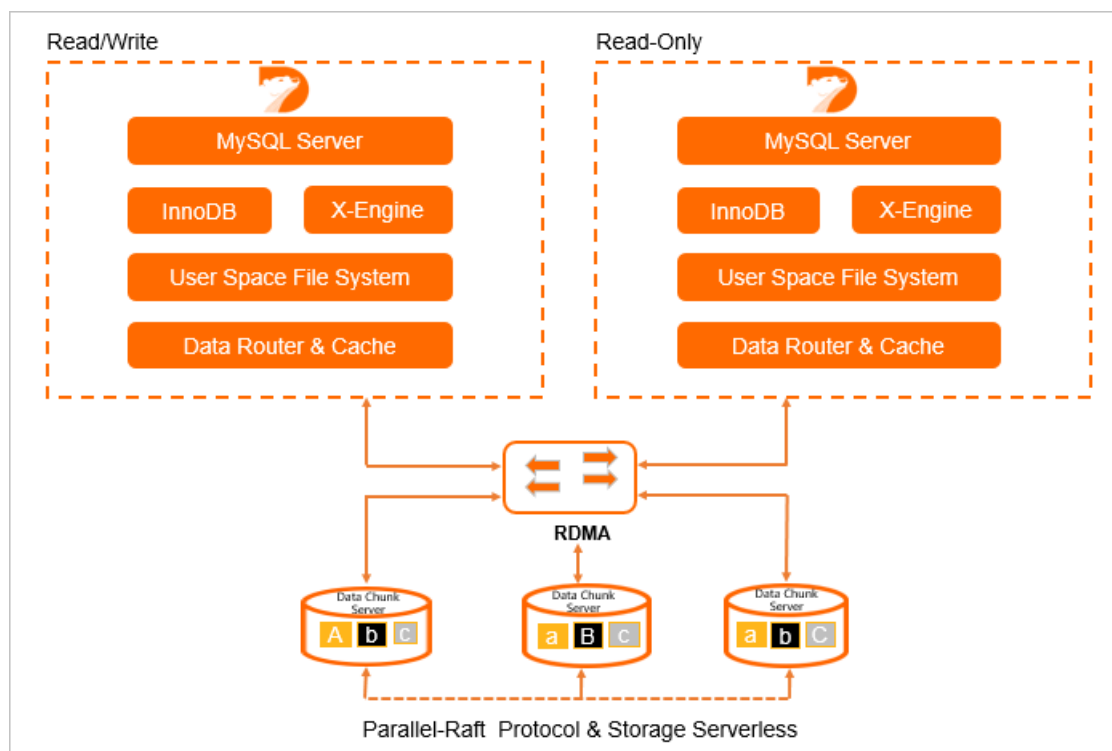
历史库单节点版采用一个计算节点，独享计算规格，可以有效地节省Proxy代理层的成本和Redo日志同步的开销。在历史库场景中，通常需要巨大的存储容量，而读写量较少，此时主节点的计算资源都无法利用完，更无需读节点提供的读扩展能力。在主节点和只读节点规格相同的情况下，相当于浪费了一半的计算资源。借助X-Engine引擎的数据压缩能力，历史库能够有效降低存储成本。历史库单节点版中使用单个主节点来提供服务，省去了只读节点的计算资源成本。当然，去除了只读节点，在灾难场景（如主节点出现异常崩溃）下，PolarDB需要更长的恢复时间。但是依靠底层分布式存储提供的高可用能力，历史库单节点版依然能够保证99.95%的可用性。对大多数用户来说，在仅需低频读写的场景（很多时候数据需异步批量导入至历史库）中，用稍低一点的可用性换取存储成本的节省是可接受的。

历史库单节点架构下，日常不提供只读节点。在需要对节点进行运维操作（如小版本升级需要重启节点）时，会将系统内部部署的临时只读节点升级为主节点，来减少对历史库的读写影响。历史库单节点架构图如下：



历史库集群版

历史库集群版在历史库单节点版的基础上，基于共享存储实现了一写多读，集群中有一个主节点（可读可写）和至少一个只读节点。历史库集群版多节点架构不仅继承了历史库单节点的优势，还可用于保障集群的高可用，当系统发生故障时，可读写的主节点和只读节点之间会自动进行故障切换（Failover），保证了服务可用性不低于99.99%。历史库集群版多节点架构图如下：



产品优势

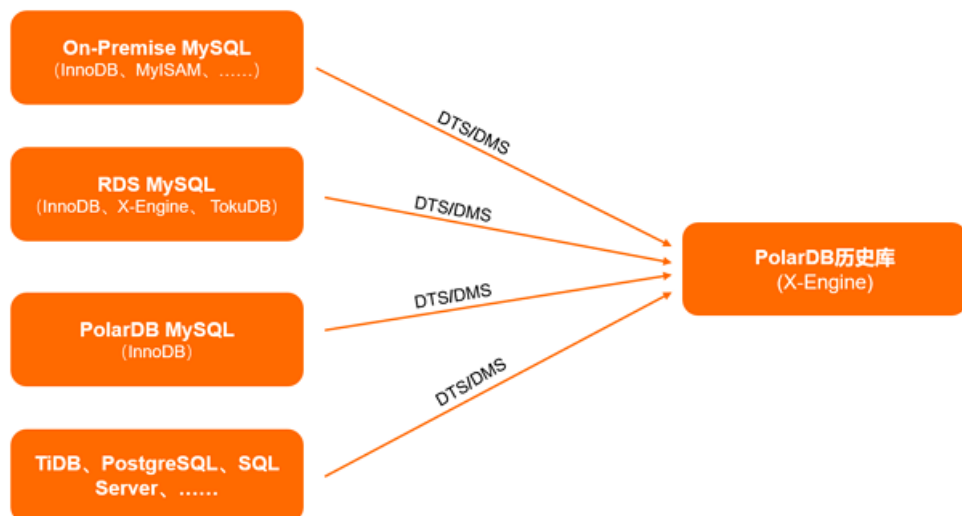
- 超大存储容量。200 TB的存储空间加上X-Engine数据压缩能力，可提供500 TB以上的原始数据存储容量，同时存储空间采用Serverless方式，购买时无需选择容量，随着数据增长而在线自动扩容，只按实际数据量大小收费。
- PolarDB历史库与官方MySQL的协议一致，相比于将历史数据备份到HBase等NoSQL产品，应用程序端无需修改任何代码即可同时访问在线库和历史库。
- 借助PolarDB底层共享存储的快速备份能力，实现对海量数据的快速备份，备份数据可以上传到OSS等廉价存储设备，确保数据永不丢失。
- 历史库集群版多节点架构借助X-Engine引擎的数据压缩能力，不仅能够有效降低存储成本，还可用于保障集群的高可用。当系统发生故障时，可读写的主节点和只读节点之间会自动进行故障切换（Failover），保证了服务可用性不低于99.99%。

适用场景

PolarDB历史库提供了超大存储容量，它可以同时作为多个业务历史数据的汇聚地，以方便对所有历史数据进行集中存储和管理，主要适用于如下几个场景：

- 将PolarDB历史库作为线下自建数据库实例的冷数据存储地，线下自建数据库包括但不限于MySQL、TiDB、PostgreSQL、SQL Server等关系型数据库。
- 将PolarDB历史库作为阿里云RDS MySQL或者PolarDB MySQL引擎数据库服务的归档存储地，将访问较少的历史数据迁移到PolarDB MySQL引擎X-Engine中存储，释放在线数据库的空间以降低成本并提升性能。
- 直接将PolarDB历史库作为大容量关系数据库使用，以满足一些写入数据量巨大，但读频次较低的业务的需求（如系统监控日志等）。

您可以通过阿里云DTS持续实时地将在线库的数据迁移至PolarDB历史库，或通过阿里云DMS周期性地将在线数据批量导入到PolarDB历史库。



支持的内核版本

目前支持PolarDB MySQL引擎8.0版本。

节点规格与定价

历史库支持历史库单节点版和历史库集群版两种规格。更多详情，请参见[计算节点规格](#)。

关于历史库单节点版和历史库集群版的计费规则，请参见[计算节点计费规则](#)。

常见问题

Q：历史库单节点版仅使用了单个主节点，是如何保证服务可用性和数据可靠性呢？

A：历史库单节点版是基于单个计算节点提供特定用途的数据库产品。虽然只有一个节点，但历史库单节点版借助秒级计算调度、分布式多副本存储等技术，依旧可以保证服务的高可用性和数据的高可靠性。

相关视频

15.2. 使用说明

PolarDB历史库使用X-Engine代替了InnoDB作为默认存储引擎，具有较高的数据压缩率，非常适合对计算诉求不高但需要存储一些归档类数据（如钉钉消息等数据）的业务。本文介绍PolarDB历史库相关的使用说明。

前提条件

仅PolarDB MySQL引擎8.0版本支持历史库产品系列。更多详情，请参见[PolarDB历史库](#)。

创建PolarDB历史库（X-Engine）

PolarDB历史库的创建方式与PolarDB集群创建方式一致，您只需在售卖页选择兼容性为MySQL 8.0，且系列为历史库（高压压缩率）即可。其他参数说明，请参见[购买按量付费集群](#)和[购买包年包月集群](#)。

兼容性	☒ MySQL 8.0	☐ MySQL 5.7	☐ MySQL 5.6	☐ PostgreSQL 11	☐ 兼容Oracle语法
与 MySQL 8.0 完全兼容					
系列	☐ 集群版（2-16个节点）【推荐】	☐ 单节点（入门级）	☒ 历史库（高压压缩率）		

在历史库中创建表

在PolarDB历史库上创建表的方式与在使用InnoDB引擎的数据库上创建方式一致。由于历史库默认引擎为X-Engine，若建表时未指定引擎，默认会创建X-Engine引擎的表。例如，使用如下命令创建一张表（未指定引擎）：

```
CREATE TABLE test_arc.t1 (id int PRIMARY KEY,c1 varchar(10));
```

使用 `show create table` 命令查看建表语句详情，结果如下：

Table	t1
Create Table	CREATE TABLE `t1` (`id` int(11) NOT NULL, `c1` varchar(10) COLLATE utf8mb4_general_ci DEFAULT NULL, PRIMARY KEY (`id`)) ENGINE=XENGINE DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci

表创建成功后，后续使用方法与InnoDB一样，数据会存储在X-Engine引擎上。

说明

- PolarDB历史库同时也支持创建InnoDB表，例如使用DTS迁移数据时，可能会出现迁移的表引擎仍然为InnoDB。若需要将其转换为X-Engine表，请参见[InnoDB引擎转换为X-Engine引擎](#)。
- 您可以通过如下命令查看当前数据库的默认引擎：

```
show variables like '%default_storage_engine%';
```

使用限制

- 与InnoDB引擎共存时的资源分配限制

使用X-Engine引擎时，由于95%的内存会提供给X-Engine引擎用做写入缓存和BlockCache以加速读写速度，留给InnoDB Buffer Pool的内存非常少，所以在X-Engine引擎的集群中，请尽量避免使用InnoDB引擎表存储太多数据，否则会因为缓存命中率低而导致性能大幅降低。建议使用PolarDB历史库时，所有的表都使用相同的X-Engine引擎，避免引擎混用影响集群性能。

- 引擎功能限制

X-Engine引擎在功能上有如下限制。

分类	功能	说明
SQL功能	外键	不支持
	临时表	不支持
	分区表（partition）	不支持（所有与partition相关的创建及增删改查操作均不支持）
	Generated Column	不支持
	Handler API	不支持
	最大列长度 （longblob/longtext/json）	32 MB

列属性分类	功能	说明
	GIS地理数据类型	所有GIS相关数据类型均不支持（包含 geometry、point、linestring、polygon、multipoint、multilinestring、multipolygon、geometrycollection）
索引	哈希索引	不支持
	空间索引	不支持（所有与全文索引相关的创建、使用均不支持）
事务	事务隔离级别	支持如下2个隔离级别： ◦ 读已提交（RC） ◦ 可重复读（RR）
	单个事务支持的最大数据量	32 MB
	Savepoint	不支持
	XA事务	即将支持
锁	锁粒度	◦ 支持表级别锁和行级别锁 ◦ 不支持GAP锁
	Skip Locked	不支持
	Lock Nowait	不支持
字符集	非索引列支持的字符集	非索引列支持所有的字符集（校对规则）
	索引列支持的字符集	◦ latin1 (latin1_bin) ◦ gbk (gbk_chinese_ci、gbk_bin) ◦ utf8 (utf8_general_ci、utf8_bin) ◦ utf8mb4 (utf8mb4_0900_ai_ci、utf8mb4_general_ci、utf8mb4_bin)
主从复制	Binlog格式	支持如下3种格式： ◦ stmt ◦ row ◦ mixed  说明 默认为row，采用 stmt/mixed在特定并发场景可能存在数据安全性问题。

 **说明** 对于X-Engine引擎中其他未列出的功能，默认其功能特性与InnoDB引擎相同。

- 大事务功能限制

X-Engine目前不支持大事务。当一个事务修改的行数等于或大于10000行时，X-Engine会启用 `commit in middle` 功能，即X-Engine会在内部提交该事务，并重新开启一个子事务继续执行该事务。但 `commit in middle` 并不能遵循严格意义上的事务原子性（Atomicity），因此您在使用过程中需要注意以下几点：

- 假设需要开启一个事务插入大量数据，但在数据插入过程中，由于 `commit in middle` 功能，已经先提交了一部分数据，其它请求就可以访问到插入的数据。
- 假设需要开启一个事务修改大量数据，已经执行了 `commit in middle` 的事务将无法回滚。

```
drop table t1;
create table t1(c1 int primary key , c2 int)ENGINE=xengine;
begin;
call insert_data(12000); //插入12000行数据，触发commit in middle，前10000行数据已经提交。
rollback; // 回滚只能把最后2000条数据回滚。
select count(*) from t1; // 这里仍然能够查询到10000条数据。
+-----+
| count(*) |
+-----+
|    10000 |
+-----+
1 row in set (0.00 sec)
```

- 假设需要开启一个需要修改且删除大量数据的事务，由于 `commit in middle` 功能，DELETE操作读不到本事务中新插入的行，导致新插入的数据未被DELETE。

```
drop table t1;
create table t1(c1 int primary key , c2 int)ENGINE=xengine;
call insert_data(10000);
begin;
insert into t1 values(10001,10001), (10002,10002);
delete from t1 where c1 >= 0; // delete操作触发commit in middle，导致delete操作没有读到本事务插入的行。
commit;
select * from t1;
+-----+-----+
| c1    | c2    |
+-----+-----+
| 10001 | 10001 |
| 10002 | 10002 |
+-----+-----+
2 rows in set (0.00 sec)
```

参数说明

② 说明

- 针对表中可修改的参数，您可以根据业务需要进行修改。具体操作步骤，请参见[设置集群参数和节点参数](#)。
- 表中所有参数在PolarDB控制台上都已加上MySQL配置文件的兼容性前缀 `loose_`。

类别	参数	说明	是否支持修改	参数修改后，是否需要重启集群生效
性能	xengine_batch_group_max_group_size	事务流水线最大分组数。	否	不涉及
	xengine_batch_group_max_leader_wait_time_us	事务流水线的最大等待时间。	否	不涉及
	xengine_batch_group_slot_array_size	事务流水线最大batch大小。	否	不涉及
	xengine_parallel_read_threads	并行扫描并发数	是	否
	xengine_parallel_wal_recover	并行恢复	否	不涉及
内存	xengine_block_cache_size	读block缓存的大小。	否	不涉及
	xengine_row_cache_size	行缓存的大小。	否	不涉及
	xengine_write_buffer_size	单Memtable的最大大小。	否	不涉及
	xengine_block_size	磁盘上数据block大小。	否	不涉及
	xengine_db_write_buffer_size	所有subtable的Active Memtable的总大小限制。	否	不涉及
	xengine_db_total_write_buffer_size	所有subtable的Active Memtable/Immutable memtable的总大小限制。	否	不涉及
	xengine_scan_add_blocks_limit	每个请求在范围扫描时，可以加到BlockCache中的Block数目。	是	否
compaction	xengine_flush_delete_percent_trigger	当Memtable中记录数超过此数目时，则xengine_flush_delete_record_trigger参数生效。	否	不涉及
锁	xengine_max_row_locks	单SQL请求中，最大可以锁定的行数。	否	不涉及
	xengine_lock_wait_timeout	锁等待超时时间。	是	否

16.全球数据库

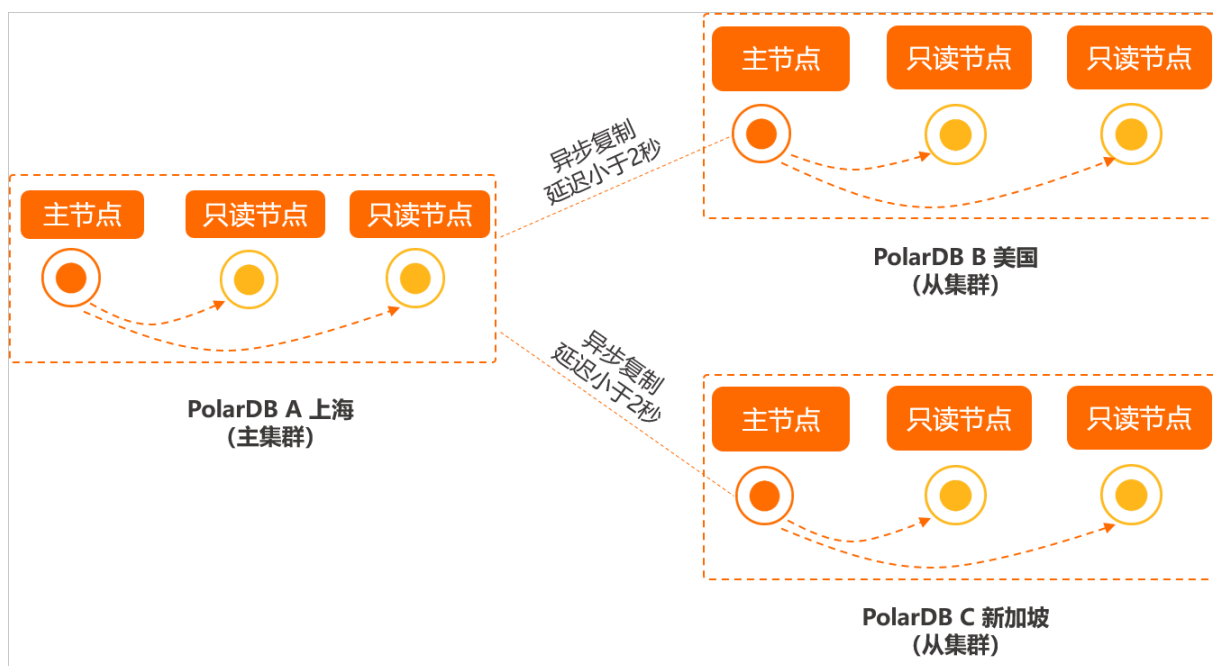
16.1. 概述

全球数据库网络（Global Database Network，简称GDN）是由分布在全球多个地域的多个PolarDB集群组成的网络。本文将介绍GDN特性等详情。

GDN网络中所有集群的数据保持同步，网络中每个集群均提供读写服务，并且提供异地灾备能力，可满足如下应用场景：

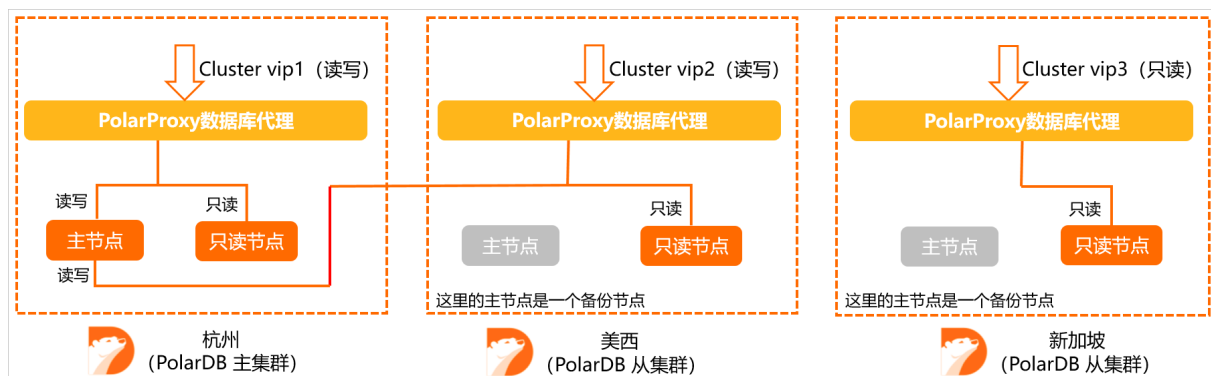
- 异地多活
如果业务部署在多个地域，传统网络下，数据库在主地域，其它地域的应用需要跨地域访问主地域的数据库，网络延迟会导致数据库性能低下，带来不良的用户体验。通过全球数据库网络的跨地域低延迟同步、跨地域读写分离、本地就近读取等特性，可以确保各地域的应用访问数据库时的延迟小于2秒。
- 异地灾备
不论业务部署在一个或多个地域，都能通过全球数据库网络实现异地容灾。当主集群出现地域级别的故障时，您只需要手动将您的业务切换到从集群。

❓ 说明 主从集群切换在10分钟之内完成（实际测试5分钟以内），切换过程中，可能会出现60秒以内的连接闪断。建议您在业务低谷期执行切换操作并确保应用具备重连机制。



请求路由说明

GDN中每个集群的读写请求路由，是由对应集群的Cluster vip（Endpoint）配置决定。例如Cluster vip2配置为可读可写，并且主库接受读，那么读请求就会路由到主集群的主节点，这时请求延迟可能会较大；如果Cluster vip3配置为只读，那么读请求只会路由到从集群的只读节点，不会路由到主集群和从集群的主节点。关于如何配置集群的Cluster vip，请参见[配置数据库代理](#)。如下图所示：



❓ **说明** 从集群的Endpoint地址，如果配置为可读可写，那么写请求以及其它的广播语法（如set语句）会被路由到主集群的主节点；如果开启了会话一致性功能，那么读请求也可能被路由到主集群的主节点。

优势

- 无需修改业务代码：可以直接从同城部署扩展到多地部署，详情请参见[跨地域部署](#)。
- 跨地域读写分离、就近读取：GDN内所有集群均可读可写，读请求被直接发送到本地集群，写请求被转发至主集群。详情请参见[跨地域读写分离](#)。
- 配置灵活：主、从集群拥有相互独立的配置，如集群规格、白名单和参数值等。详情请参见[创建全球数据库网络](#)。
- 跨地域低延时同步：通过多通道物理复制，高压力下全球同步延迟小于2秒。详情请参见[跨地域低延时同步](#)。

费用

目前全球数据库网络（GDN）跨地域传输流量0折优惠，您只需要支付每个PolarDB集群自身的费用，关于PolarDB集群自身费用，请参见[计费项概览](#)。

支持的地域和集群

- 地域：覆盖全球包括中国内地、中国香港及海外地域在内的10多个地域。
- GDN中的集群版本需为以下版本之一：
 - PolarDB MySQL引擎8.0.1版本
 - PolarDB MySQL引擎5.7版本且内核小版本需为5.7.1.0.13及以上
 - PolarDB MySQL引擎5.6版本且内核小版本需为5.6.1.0.27及以上
- 主集群和从集群的数据库引擎版本需保持一致，必须都为MySQL 8.0、MySQL 5.7或MySQL 5.6。

开始使用

详情请参见[创建全球数据库网络](#)。

相关视频

GDN

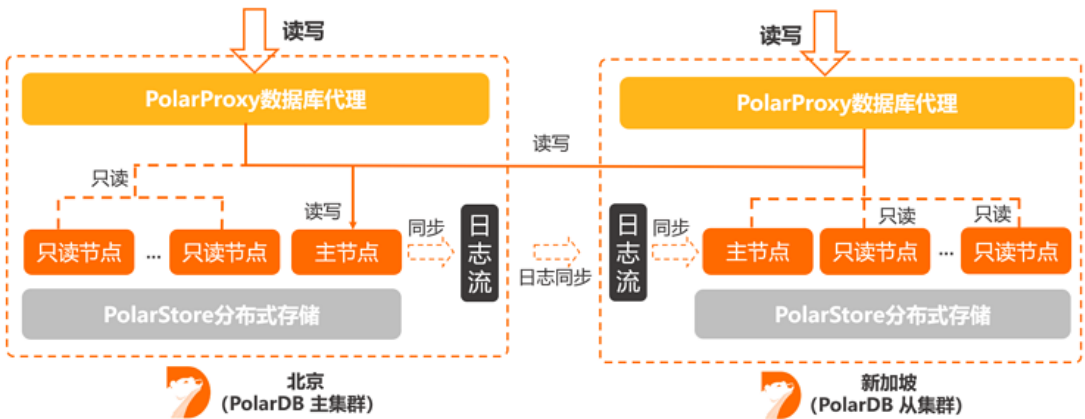
相关文档

- [技术架构](#)
- [异地灾备与异地多活](#)
- [创建与删除全球数据库网络](#)
- [添加与移除从集群](#)
- [连接全球数据库网络](#)

16.2. 技术架构

本文介绍全球数据库网络（Global Database Network，简称GDN）的技术架构。

GDN架构



跨地域部署

- 一个主集群（Primary），其他为从集群（Secondary），多个PolarDB集群之间数据保持同步。

❓ 说明 当前一个GDN中包含一个主集群和最多四个从集群，若您需要添加更多从集群，请[提交工单](#)联系技术支持进行添加。

- GDN中的每个集群都默认包含2个节点，最多可增加至16个节点，详情请参见[增加只读节点](#)。

跨地域低延时同步

全球数据库网络支持通过异步复制机制帮助实现跨地域复制数据，并通过物理日志、并行处理等技术降低主从集群间跨地域复制的延迟。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，大大降低了非中心区域应用访问的读取延迟。创建异地从集群并保持数据同步，不会影响现有主集群的稳定性和性能。

❓ 说明 创建从集群时，建议选择与主集群一致的节点规格以保证低延迟同步效果。如何创建从集群，请参见[添加从集群](#)。

如下是以GDN张家口-美西地域为例的低延时同步的测试结果。

测试集群规格和拓扑	Sysbench压力测试	峰值QPS/TPS	美西从集群到张家口主集群的延迟
	OLTP_INSERT插入负载	82655/82655	小于1秒

GDN张家口-美西地域 测试集群规格和拓扑 PolarDB MySQL引擎 16核128 GB	Sysbench压力测试	峰值QPS/TPS	美西从集群到张家口主集群的延迟
	OLTP_WRITE_ONLY只写负载	157953/26325	小于1秒
	OLTP_READ_WRITE读写负载	136758/6837	小于1秒

跨地域读写分离

- 特点

- 每个集群均可读写。
- 读请求通常情况下会直接发往本地，而写请求则被转发至主集群。

 **说明** 从集群中的主节点主要用于异步复制主集群数据。为减少跨地域下物理复制的延迟，读请求默认发往本地从集群下的只读节点。

- 无需在应用程序上修改代码，拆分读写请求。

- 实现方式

GDN的跨地域读写分离功能需依赖PolarDB集群的集群连接地址实现。如何设置GDN的集群连接地址，具体操作请参见[连接全球数据库网络](#)。

- 转发逻辑

目标节点	转发的请求
只发往主节点	- 所有DML操作（INSERT、UPDATE、DELETE、SELECT FOR UPDATE）。 - 所有DDL操作（建表或库、删表或库、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。

 **说明** 仅当开启主库是否接受读时会发往主节点。

目标节点	转发的请求
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。

说明 从集群中的主节点主要用于异步复制主集群数据而不处理任何读写请求，因此表中的主节点指主集群中的主节点，只读节点指从集群中的只读节点。

相关文档

- 概述
- 异地灾备与异地多活
- 创建与删除全球数据库网络
- 添加与移除从集群

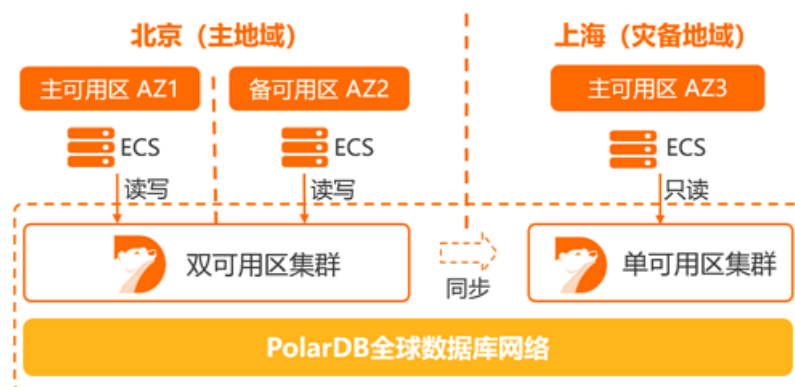
16.3. 异地灾备与异地多活

异地灾备与异地多活是全球数据库网络（Global Database Network，简称GDN）中两种典型的应用场景。本文介绍这两种典型场景的业务架构及部署流程。

异地灾备

通过异地灾备实现跨地域高可用，提升数据安全性和系统可用性。当发生机房或数据中心级别故障时，可以快速恢复业务。可以实现两地三中心、两地四中心、三地六中心等架构。

- 典型行业：银行、证券、保险、互金等。
- 业务架构（以两地三中心架构为例）
数据库由两个PolarDB集群组成：



- 北京的双可用区集群，覆盖AZ 1和AZ 2。
- 上海的单可用区集群。

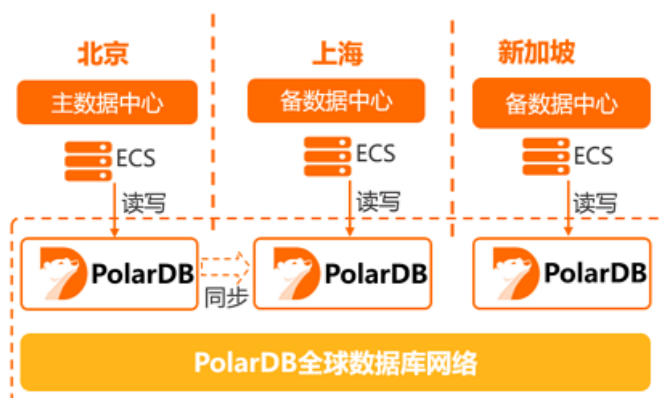
应用在北京，对AZ 1的数据库进行本地读写。

- 当北京AZ 1故障时，优先切换到北京AZ 2。
- 当北京AZ 1和AZ 2均故障时，切换到上海AZ 3。

异地多活（多地部署）

企业的用户遍布全国或全球，数据需要打通，多地域都可以读写。全球地域都能访问同一个数据库，读写请求尽量发往本地集群。

- 典型行业：游戏、跨境电商、本地生活（外卖）、新零售（门店）
- 业务架构示例如下：
 - 各地域的应用程序均直接就近读写各地域的数据库，性能最佳。
 - 应用程序只需配置一个连接地址，无需修改代码即可将业务平滑扩展到更多地域，从1个数据中心，扩展到2个、3个甚至更多的数据中心。
 - 上海、新加坡的集群规格无需与北京保持一致，可以灵活选择。



部署流程

1. 创建一个GDN，将已有集群作为GDN中的主集群。详细操作步骤，请参见[创建全球数据库网络](#)。
2. 在GDN中，添加新集群作为从集群。详细操作步骤，请参见[添加从集群](#)。
3. 连接GDN。详细操作步骤，请参见[连接全球数据库网络](#)。

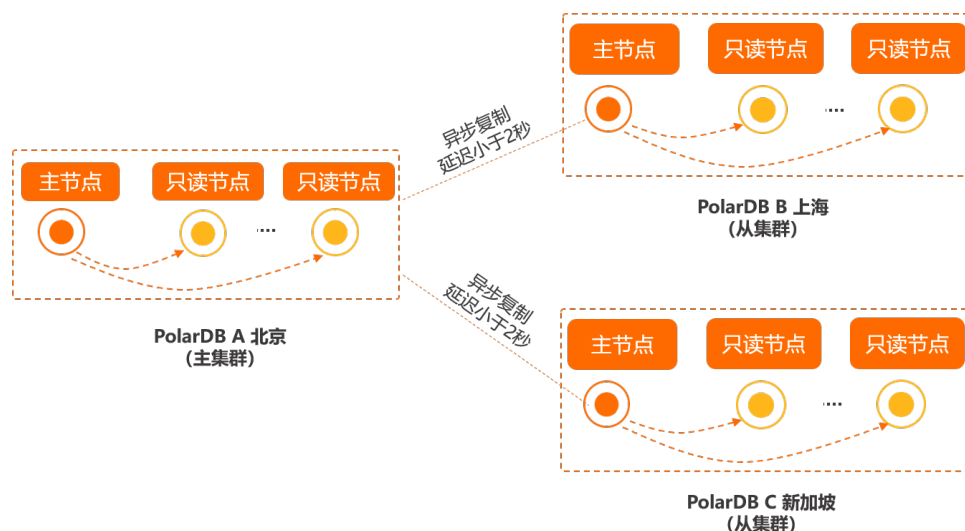
更多关于跨地域部署GDN的最佳实践，请参见[跨地域部署GDN最佳实践](#)。

16.4. 跨地域部署GDN最佳实践

本文介绍跨地域部署GDN的完整流程和具体操作步骤。

部署规划

- 架构
某游戏公司原业务主要在北京，现需将业务拓展至上海和新加坡，要求不同地域间的数据需要打通，全球地域都能访问同一个数据库，且读写请求尽量发往本地集群。



● 流程

- 创建GDN。
- 添加从集群。
- 连接全球数据库网络。
- (可选) 购买计算包。
- (可选) 购买存储包。

创建GDN

创建一个GDN后，默认包含一个主集群。详细操作步骤，请参见[创建GDN](#)。

说明 目前全球数据库网络（GDN）跨地域传输流量0折优惠，您只需要支付每个PolarDB集群自身的费用，关于PolarDB集群自身费用，请参见[计费项概览](#)。

添加从集群

在目标GDN中添加从集群，组成一个完整的全球数据网络。详细操作步骤，请参见[添加从集群](#)。

说明

- 创建从集群时，节点规格建议与主集群的节点规格一致以保证低延迟同步效果，只读节点的个数无需与主集群保持一致，您可以根据本地从集群的读请求量自行选择。
- GDN具备跨地域低延迟同步特性，从集群能自动与主集群保持数据同步，无需购买额外的数据传输工具（如DTS）。

连接全球数据库网络

GDN创建成功后，您即可通过集群地址连接GDN进行管理，详细操作步骤，请参见[连接全球数据库网络](#)。

(可选) 购买计算包

若创建主集群或从集群时选择付费方式为[按量付费](#)，您还可以购买计算包来抵扣按量付费集群的计算节点费用。购买计算包后，[按量付费](#)集群的计算节点费用将由计算包自动抵扣费用，更加划算。详细操作步骤，请参见[购买计算包](#)。

说明

- 您可以购买多个计算包叠加使用，一个计算包可以被多个集群共享使用。
- 每个计算包的有效期为1年、3年或5年，有效期结束后剩余的计算包容量会自动失效。
- 若计算包剩余计算量不足以抵扣，超出部分的计算节点费用将按量付费。
- 计算包不支持续费或升降级，若计算包容量不够、即将到期或已过期，您可以再次购买计算包叠加使用。
- 仅当您的账号下有按量付费的PolarDB集群时，计算包容量才会被消耗。

（可选）购买存储包

存储包支持抵扣如下两种费用：

- **按容量计费（按量付费）的存储类型费用**
PolarDB集群的存储类型都是根据数据量和存储时长单独收费的，若您需要的数据量较大（如1000 GB或以上），推荐配合存储包一起使用，以降低存储成本。存储空间费用的抵扣比例为1:1，即1 GB存储包容量可抵扣1 GB存储空间用量。

说明 当前存储包仅可抵扣按容量计费（按量付费）的存储空间费用，不能抵扣按空间计费（包年包月）的存储空间费用。

- **一级备份（超出免费额度）费用**
购买存储包后，若抵扣完PolarDB集群的存储空间用量后，存储包容量仍有剩余，将自动按照1:1.6的比例（即1 GB存储包容量抵扣1.6 GB一级备份用量）抵扣一级备份中超过免费额度的部分。若存储包中剩余容量不足以抵扣一级备份的空间用量，超出部分将正常按量付费。

详细操作步骤，请参见[购买存储包](#)。

说明

- 存储包分为中国内地通用和中国香港及海外通用两种类型，且每种类型的存储包只允许购买一个。
- 存储包由资源包类型（中国内地通用或中国香港及海外通用）规定地域内的所有集群共享使用。更多信息，请参见[常见问题](#)。
- 存储包容量不够时，您可以选择[升级存储包](#)，但暂不支持降级已购存储包。
- 超出存储包容量的部分以按量付费的方式收取费用。

例如，GDN部署规划中3个集群的存储容量均为300 GB，那么您可以购买一个500 GB的中国内地通用存储包抵扣北京主集群和上海从集群的存储费用，再购买一个300 GB的中国香港及海外通用存储包抵扣新加坡从集群的存储费用，超出的100 GB则按量付费。

16.5. 创建与删除全球数据库网络

全球数据库网络（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均保持同步。当您的业务部署在多个地域时，利用GDN可以实现应用访问数据库的低延迟和高稳定性。本文介绍如何新建和删除全球数据库网络。

前提条件

已创建主PolarDB集群。关于如何创建集群，请参见[购买按量付费集群](#)和[购买包年包月集群](#)。

使用限制

- GDN中的集群版本需为以下版本之一：
 - PolarDB MySQL引擎8.0.1版本
 - PolarDB MySQL引擎5.7版本且内核小版本需为5.7.1.0.13及以上
 - PolarDB MySQL引擎5.6版本且内核小版本需为5.6.1.0.27及以上
- 主集群和从集群的数据库引擎版本需保持一致，必须都为MySQL 8.0、MySQL 5.7或MySQL 5.6。
- 一个GDN中包含一个主集群和最多四个从集群，集群的地域和可用区不受限制，多个集群可以存在于同一个地域的同一个可用区。

 **说明** 若您需要添加更多从集群，请[提交工单](#)联系技术支持进行添加。

- 一个集群只能属于一个GDN。
- GDN中的集群不支持2核4 GB和2核8 GB节点规格。
- GDN中的集群不支持库表恢复功能。

费用

目前全球数据库网络（GDN）跨地域传输流量0折优惠，您只需要支付每个PolarDB集群自身的费用，关于PolarDB集群自身费用，请参见[计费项概览](#)。

创建全球数据库网络

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](#)。

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	pc- 华东1（杭州） pc- 华东2（上海） 更多	2020年3月23日 13:46:54	添加从集群 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	选择一个已有集群作为GDN的主集群。  说明 不支持PolarDB MySQL引擎8.0.2版本。

5. 完成配置后，单击[确定](#)。

 **说明** 全球数据库网络创建成功后，您可以添加从集群，详细操作步骤请参见[添加与移除从集群](#)。

删除全球数据库网络

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GDN，单击操作栏的删除。

注意

- 当全球数据库网络中只有一个主集群时，才可以被删除。
- GDN删除后将无法恢复，请谨慎处理。
- GDN删除后，连接GDN的应用程序将无法访问数据库，请及时修改应用程序的相关连接代码。

4. 在[删除全球数据库网络](#)对话框，单击确定。

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。
- Q：GDN创建后，能否更换GDN中的主集群？
A：暂不支持在创建GDN后更换主集群。您可以重新创建一个GDN，并选择需要的集群作为主集群，然后将原GDN释放即可。

 **说明** 原GDN释放前需先移除所有从集群。

相关视频

16.6. 添加与移除从集群

全球数据库网络（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，利用GDN可以实现应用访问数据库的低延迟和高稳定性。本文介绍如何添加和移除从集群。

前提条件

[创建全球数据库网络](#)

注意事项

- GDN中的集群版本需为以下版本之一：
 - PolarDB MySQL引擎8.0.1版本
 - PolarDB MySQL引擎5.7版本且内核小版本需为5.7.1.0.13及以上
 - PolarDB MySQL引擎5.6版本且内核小版本需为5.6.1.0.27及以上
- 主集群和从集群的数据库引擎版本需保持一致，必须都为MySQL 8.0、MySQL 5.7或MySQL 5.6。
- 从集群只能新建，不能将已经创建好的集群添加为从集群。
- 创建从集群时，节点规格建议与主集群的节点规格一致以保证低延迟同步效果，只读节点的个数无需与主

集群保持一致，您可以根据本地从集群的读请求量自行选择。

- 一个GDN中包含一个主集群和最多四个从集群，集群的地域和可用区不受限制，多个集群可以存在于同一个地域的同一个可用区。

 说明 若您需要添加更多从集群，请[提交工单](#)联系技术支持进行添加。

- 一个集群只能属于一个GDN。
- GDN中的集群不支持2核4 GB和2核8 GB节点规格。

费用

目前全球数据库网络（GDN）跨地域传输流量0折优惠，您只需要支付每个PolarDB集群自身的费用，关于PolarDB集群自身费用，请参见[计费项概览](#)。

添加从集群

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GDN，单击操作栏中的[添加从集群](#)。

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- GDN测试	运行中	100% 兼容 MySQL 8.0	pc- 华东1（杭州） pc- 华东2（上海） 更多	2020年3月23日 13:46:54	添加从集群 删除

 说明 GDN不支持将现有集群添加为从集群。

4. 在弹出的售卖页中，选择[包年包月](#)或[按量付费](#)。

 说明 PolarDB MySQL引擎已推出计算包。推荐购买按量付费的集群，因为按量付费的集群支持配合计算包使用，比包年包月付费方式更划算、更灵活。详情请参见[计费方式1：按量付费+计算包（推荐）](#)。

5. 设置如下参数。

参数	说明
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。
创建方式	选择创建从集群。
全球数据库网络	选择需要加入的GDN。  说明 默认为创建从集群前选择的GDN。

参数	说明
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	固定为VPC专有网络，无需选择。  说明 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，详情请参见创建和管理专有网络。
兼容性	支持MySQL 8.0、MySQL 5.7和MySQL 5.6，请与主集群的兼容性保持一致。
系列	仅支持集群版（2-16个节点）【推荐】，无需选择。
小版本号	仅支持8.0.1（与MySQL 8.0.13完全兼容）【推荐】，暂不支持8.0.2（与MySQL 8.0.18完全兼容）【公测】版本。关于各小版本的具体信息，请参见内核发布记录。  说明 如果兼容性选择MySQL 5.7或MySQL 5.6，则不支持该参数。
子系列	PolarDB MySQL引擎集群版支持通用规格和独享规格两种子系列，其中： - 独享规格：每个集群会独占所分配到的计算资源（如CPU），而不会与同一服务器上的其他集群共享资源，性能更加稳定可靠。 - 通用规格：同一服务器上的不同集群，会互相充分利用彼此空闲的计算资源（如CPU），通过复用计算资源享受规模红利，性价比更高。 关于两种类型的详细对比，请参见如何选择通用规格和独享规格。

参数	说明
节点规格	您可以按需选择节点规格。关于各节点规格的具体信息，请参见 计算节点规格 。
节点个数	系统默认为每个集群版（2-16个节点）【推荐】的集群，创建规格相同的两个节点（一主一读写），无需选择。  说明 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，请参见产品架构。
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。
时区	设置集群时区，默认时区为 UTC+08:00 。
表名大小写	设置集群表名是否区分大小写，默认不区分大小写。当本地数据库区分大小时，您可以选择区分大小写，便于您迁移数据。  说明 集群创建后该参数无法修改，请谨慎选择。
删除（释放）集群时	设置删除（释放）集群时的备份保留策略，默认保留最后一个备份（释放前自动备份）。 - 保留最后一个备份（释放前自动备份）：删除集群时保留最后一个备份。 - 保留全部备份：删除集群时保留所有备份。 - 不保留备份（释放后无法恢复）：删除集群时不保留任何备份。  说明 删除（释放）集群时保留备份可能会产生少量费用，您可以随时删除备份来节省成本，详情请参见释放集群。
集群名称	- 集群名称长度为2~128个字符，以大小写字母或中文开头，可包含数字、英文句号（.）、下划线（_）或短划线（-）。 - 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。
资源组	从已创建资源组中选择一个目标资源组。  说明 资源组是在单个云账号下将一组相关资源进行统一管理的容器，一个资源只能归属于一个资源组，详情请参见RAM资源分组与授权。

6. 设置购买时长（仅针对包年包月集群）和集群数量后，单击右侧的**立即购买**。

7. 在**确认订单**页面确认订单信息，阅读并选中服务协议。

- 若您选择的**商品类型**是**按量付费**，单击**立即开通**即可。
- 若您选择的**商品类型**是**包年包月**，单击**去支付**，并在支付页面，确认未支付订单信息和支付方式，单击**订购**。

开通成功后，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，详情请参见[搭配存储包](#)。

移除从集群

- 登录[PolarDB控制台](#)。
- 单击左侧导航栏中的**全球数据库网络（GDN）**。
- 找到目标GDN，单击目标**全球数据库网络ID/名称**。

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	pc- 华东1（杭州） pc- 华东2（上海） 更多	2020年3月23日 13:46:54	添加从集群 删除

- 在**集群列表**区域，找到目标**从集群**，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下所有集群（包括正在脱离的从集群）的连接地址仍然可用，您仍然可以通过集群的连接地址正常访问数据库。
- 只有从集群支持脱离GDN，主集群无法脱离。
- 从GDN脱离后，从集群将中断与主集群的数据同步，同时系统将设置该集群为读写模式。
- 从GDN脱离后，该集群不能再作为从集群加入GDN，请谨慎操作。

集群列表									
集群ID	地域	集群地址	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）	查看	4核 16GB	2.34 GB	运行中	主集群	-	按小时付费	脱离
pc-	华东2（上海）	查看	2核 8GB	5.40 GB	运行中	从集群	1 秒	按小时付费	脱离

- 在弹出的对话框中，单击**确定**即可。

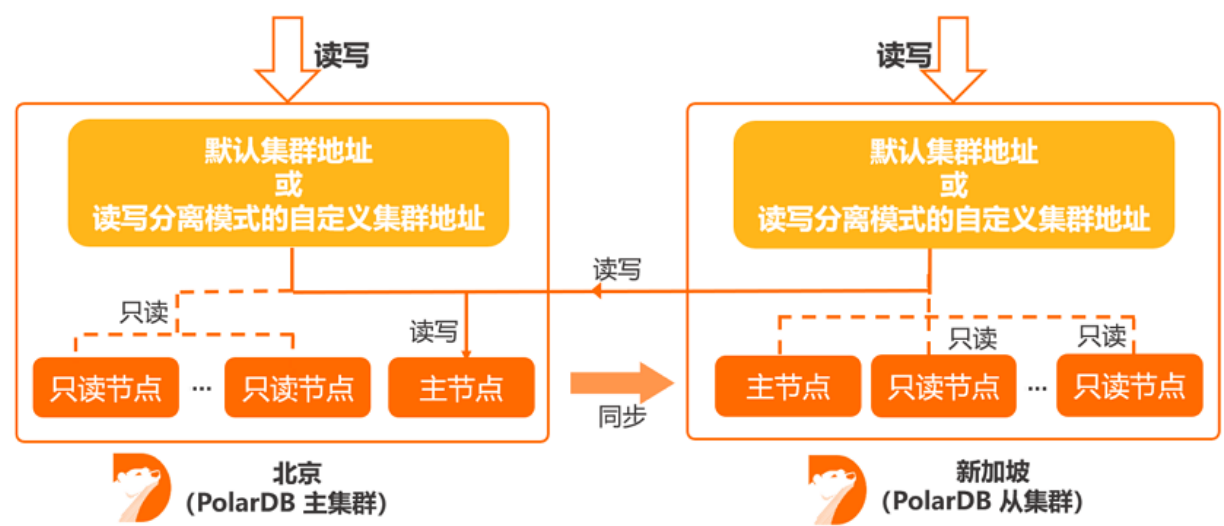
16.7. 连接全球数据库网络

全球数据库网络（Global Database Network，简称GDN）是由分布在全球多个地域的多个PolarDB集群组成的网络，在本文介绍如何查看GDN的集群地址并连接GDN。

连接地址说明

GDN不提供统一的连接地址，但GDN中每个子集群（主集群和从集群）都提供了独立的集群地址，各个地域的应用使用本地集群的集群地址连接到GDN。

GDN中所有从集群都与主集群保持数据同步。读请求通常情况下会直接发往本地，而写请求则被转发至主集群。更多关于GDN读写分离服务的介绍，请参见[跨地域读写分离](#)。



仅如下集群地址支持GDN读写分离服务：

- 默认集群地址。
- 读写模式为可读可写（自动读写分离）的自定义集群地址。

说明

- 主地址和只读模式的自定义集群地址不支持GDN读写分离服务。
- 关于如何修改读写模式，请参见[配置数据库代理](#)。
- 为降低主从集群间的复制延迟对业务可能带来的影响，建议您在从集群上设置自定义集群地址时，将主库是否接受读设置为否，并设置一致性级别为最终一致性（弱）。具体配置介绍，可参见[配置数据库代理](#)。

查看集群地址

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GDN，单击目标全球数据库网络ID/名称。

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-xxxxxx 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，找到目标集群。
5. 单击集群地址栏的查看，即可在弹出的对话框中查看集群地址详情。



说明

- 当前仅支持查看默认集群的连接地址（包括私网和公网地址）信息。
- 如需查看更多连接地址详情，请单击[查看或管理更多地址](#)。关于如何管理连接地址，请参见[申请集群地址和主地址](#)。

连接全球数据库集群

各个地域的应用使用本地集群的集群地址连接到GDN，GDN会自动进行读写分离。关于如何连接数据库集群，请参见[连接数据库集群](#)。

相关API

API	描述
DescribeGlobalDatabaseNetworks	查看账号下所有全球数据库网络（GDN）的详情。
CreateDBClusterEndpoint	创建PolarDB自定义集群地址。
DescribeDBClusterEndpoints	查询PolarDB集群的连接地址信息。
ModifyDBClusterEndpoint	修改PolarDB自定义集群地址的属性，包括读写模式、新节点是否自动加入本地址、一致性级别等。
DeleteDBClusterEndpoint	释放PolarDB自定义集群地址。

17.HTAP

17.1. 列存索引（IMCI）

17.1.1. 概述

本文介绍了PolarDB MySQL引擎的列存索引特性。

简介

当前PolarDB MySQL引擎主要面向OLTP场景，广泛应用于在线业务，日常产生大量的数据。但是，PolarDB MySQL基于行存的查询性能并不能满足所有应用场景的需求。通常情况下，为了实现复杂分析型查询，需要将数据从PolarDB MySQL导出，然后导入到外部专用的OLAP系统中再进行分析查询。这样一来，需要使用两套数据库系统，架构复杂性、运维工作量和成本都会大大增加。

PolarDB MySQL引擎重磅推出的列存索引（In-Memory Column Index，简称IMCI）面向OLAP场景大数据量复杂查询。通过列存索引，PolarDB MySQL实现了一体化的实时事务处理和实时数据分析的能力，成为一站式HTAP数据库产品解决方案。通过一套数据库系统，即可满足业务的OLTP及OLAP需求。

说明

使用过程中如果遇到问题，建议[提交工单](#)获取全面的技术支持。

版本要求

- 仅PolarDB MySQL集群版，小版本号为8.0.1的集群支持列存索引特性，且具体内核版本需为8.0.1.1.22及以上。
- 单节点、历史库单节点版、历史库集群版均不支持列存索引。

实现形态

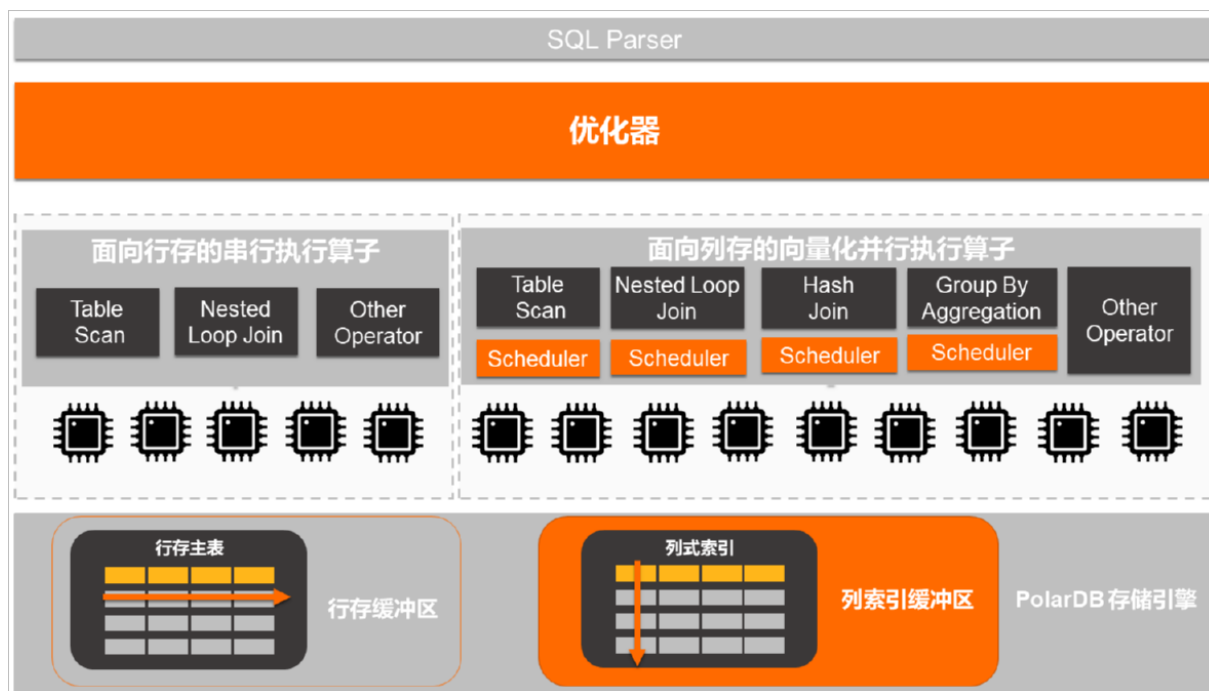
为了实现OLAP和OLTP的计算资源隔离，当前仅支持在只读节点上实现列存索引，即OLAP查询请求只发给只读节点，而不会发给主节点。至于OLAP查询请求是发给只读行存节点，还是只读列存节点，可根据[行存/列存分流方案](#)进行配置。

费用说明

该功能可免费使用，只收取只读列存节点的费用。只读列存节点按照普通的计算节点收费，具体请参见[按量付费计费规则](#)和[包年包月计费规则](#)。同时，列存索引也会占用一定的存储空间，存储空间计费具体请参见[存储类型计费规则](#)。

技术原理

列存索引特性在PolarDB MySQL引擎中的功能架构图如下：从以上架构图可以看到，PolarDB MySQL引擎从存储引擎、执行算子、优化器三个层面设计了列存索引的特性：



- 存储引擎：支持实时事务级别一致性的行列混合存储；
- 执行算子：面向列存的向量化并行执行算子，支持极速的单表和多表查询；
- SQL Parser/优化器：面向行列混合存储的CBO优化器，可以根据代价自动选择行存或者列存执行查询请求；

在此架构下，PolarDB MySQL引擎实现了100%兼容MySQL协议的基础上，同时获得数个数量级的查询加速效果。

核心优势

PolarDB MySQL引擎依托列存索引特性，具备如下优势：

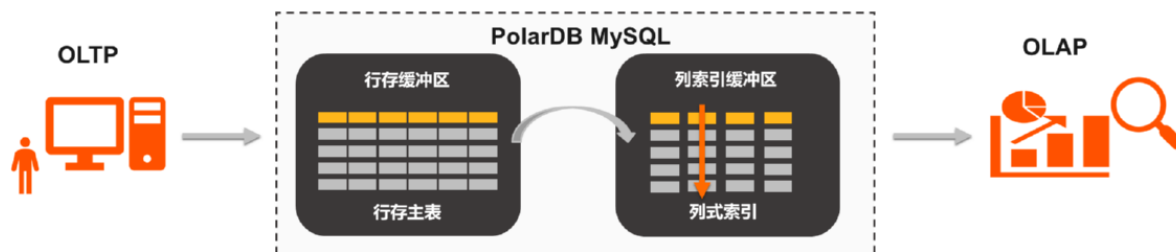
- 100%兼容MySQL：列存具有与MySQL一致的数据类型系统，支持灵活的类型转换，100%兼容MySQL协议；
- 极致的HTAP性能：PolarDB在OLTP方面本身具备极致性能。列存索引使其OLAP性能也与专用的OLAP数据库系统处于同一水平；
- 行列混合存储，降低成本：同时支持行存储和列存储两种格式，且实时保证行列的事务级一致。列存更具有低成本的优势。

适用场景

PolarDB MySQL引擎的列存索引特性提供了一站式HTAP产品体验，可以应用于多种业务场景：

- 对在线数据有实时数据分析需求的场景，如实时报表；
- 专用数据仓库场景：依托PolarDB提供的海量数据存储能力，汇聚多个上游数据源，将其作为专用数据仓库使用；

- ETL数据加速计算场景：依托PolarDB基于列存索引提供的强大而灵活的计算能力，在PolarDB中使用SQL来实现ETL功能。



性能提升情况

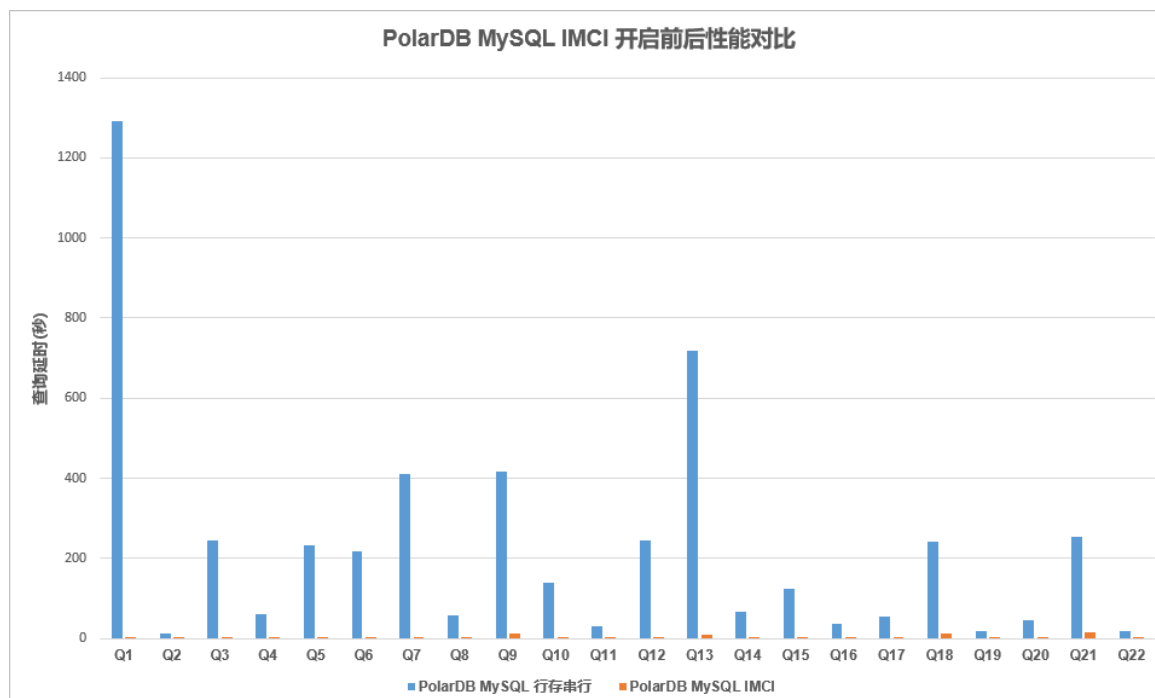
列存索引功能对SQL查询操作有明显的加速作用，查询性能甚至可以提升百倍。接下来我们以标准TPC-H测试的数据表和SQL为例，体验列存索引对查询的加速效果。

- 测试方法：**TPC-H是业界常用的一套基准，由TPC委员会制定发布，用于评测数据库的分析型查询能力。TPC-H查询包含8张数据表、22条复杂的SQL查询，大多数查询包含若干表Join、子查询和Group by聚合等。

说明

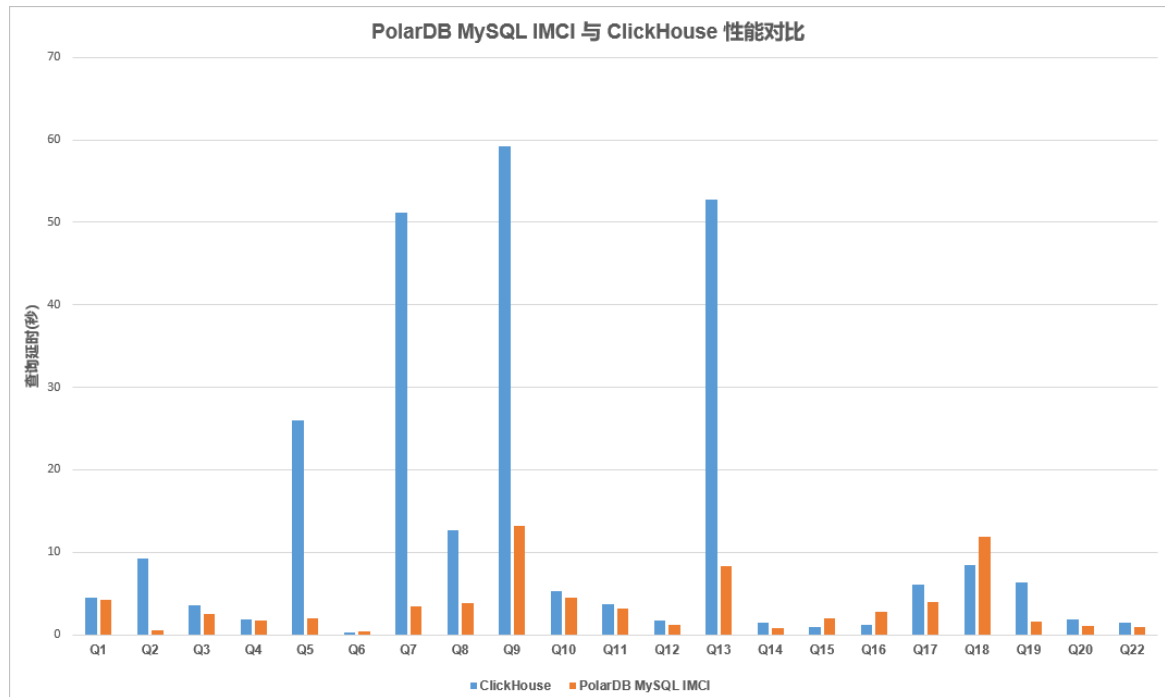
本文的TPC-H的实现基于TPC-H的基准测试，并不能与已发布的TPC-H基准测试结果相比较，本文中的测试并不符合TPC-H基准测试的所有要求。

- 数据量：**100 GB。
 - 测试结果：**
 - PolarDB MySQL开启列存索引前后性能对比
- 下图展示了PolarDB MySQL开启列存索引前后分别执行TPC-H的22条查询语句的查询响应时间的对比。



- PolarDB MySQL开启列存索引后与ClickHouse性能对比

下图展示了PolarDB MySQL开启了列存索引后，执行TPC-H的21条查询语句（Q21除外，ClickHouse不支持）的查询响应时间，与相同数据量和数据结构的ClickHouse数据库的对比。



- 结论：

- 列存索引对大多数的复杂查询操作都有加速作用，查询性能提升非常明显，甚至可达到百倍。
- 与传统OLAP数据库ClickHouse相比：PolarDB MySQL开启列存索引后，与ClickHouse性能相比各有优劣。其中PolarDB MySQL在单表Scan/AGG、Join等场景中表现突出。未来PolarDB MySQL的列存索引特性将在聚合加速、窗口函数等方面持续优化和突破。

使用说明

1. 首先您需要[添加只读列存节点](#)。
2. 然后根据业务情况，配置包含列存节点的集群地址，实现[行存/列存分流](#)。
3. 然后通过集群地址[连接数据库集群](#)后，基于[列存索引语法说明](#)在数据表中创建列存索引。
4. 若有需要，可[设置压缩算法](#)，降低存储成本。

17.1.2. 添加只读列存节点

PolarDB MySQL引擎集群默认包含一个主节点和一个只读行存节点。您可以在新增只读节点的时候选择新增只读行存节点或者只读列存节点。本章节介绍如何添加只读列存节点。

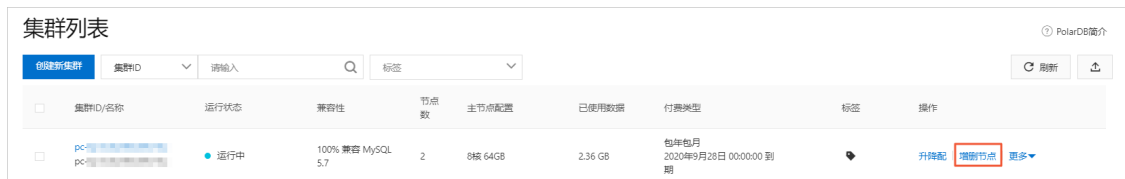
注意事项

列存索引功能与[热备切换](#)功能互斥。若集群中已有开启热备功能的只读节点，则不支持在该集群中再添加只读列存节点。

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. 您可以按照如下两种方式中的任意一种进入[增删节点向导](#)页面：

- 通过集群列表页进入增删节点向导：
找到目标集群，单击操作栏的增删节点。



- 通过目标集群基本信息页进入增删节点向导：
 - 找到目标集群，单击集群ID，进入集群基本信息页。
 - 在数据库节点区域，单击  图标切换视图。
 - 单击增删节点。



- 选中增加节点并单击确定。



- 单击+增加一个只读节点。
- 列存索引选择开启。

子系列

独享规格

独占计算资源，性能更稳定

节点

您已经添加了3个节点，还可以增加12个节点

撤销所有规格变更

pi-l-

4核 16 GB（独享）

pi-l-

4核 16 GB（独享）

pi-l-

4核 16 GB（独享）

+ 增加一个只读节点

有关 PolarDB 节点规格，详见 [规格与定价](#)。

列存索引

关闭

开启

此功能处于灰度发布阶段

数据库代理类型

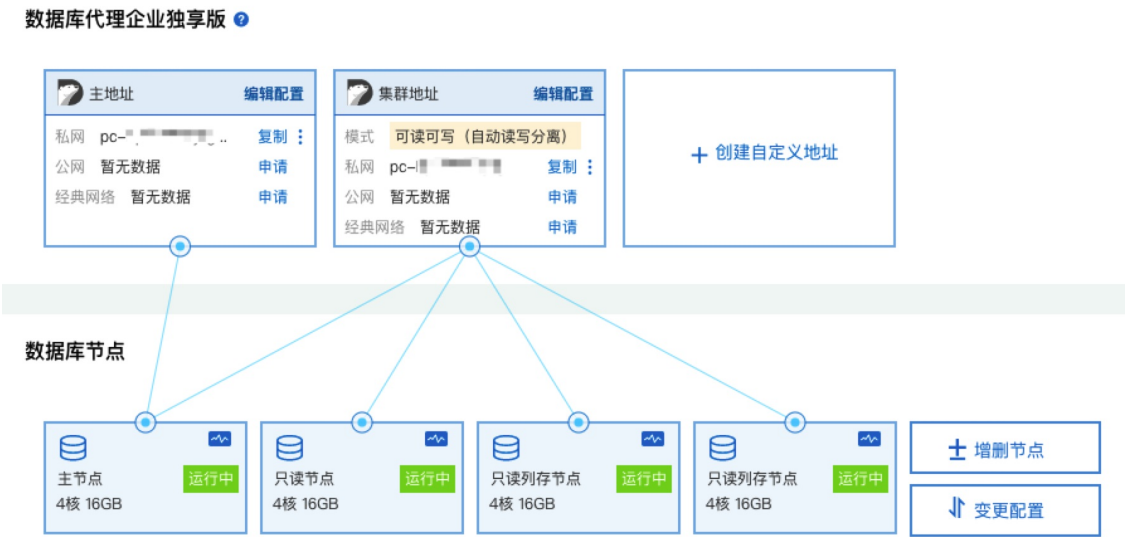
企业独享版

独占计算资源，性能更稳定

7. 选择切换时间并选中服务协议，单击去支付。

执行结果

支付完成后，返回集群的基本信息页面，稍等片刻后，只读列存节点即添加完成。



从上图可以看到，只读列存节点即为新添加的列存索引节点；只读节点即为原有的行存节点。

17.1.3. 配置集群地址实现行存和列存分流

17.1.3.1. 分流方案

如果您的业务中既包含OLTP类业务又包含OLAP类业务，则为了实现业务的最大性能，您可以基于列存索引特性，通过数据库代理的集群地址功能，实现OLAP查询引流到列存节点、OLTP查询引流到行存节点。当前支持手动分流和自动分流两种方案。

背景信息

PolarDB MySQL引擎支持两种只读节点类型：

- 只读行存节点：该类型的只读节点基于行存储处理读请求。行存节点在处理OLTP读请求时具有极致的性能。
- 只读列存节点：该类型的只读节点基于列存储处理读请求。列存节点在处理OLAP读请求（如复杂的SQL和分析型SQL）时，具有远超行存节点的性能。

 **说明** 集群初始包含一个主节点和一个只读行存节点。您可以在[添加只读节点](#)的时候选择添加的节点是行存或是列存。

为了实现SQL查询的最大性能，您可以通过配置集群地址（Endpoint），实现OLAP查询引流到列存节点、OLTP查询引流到行存节点。当前支持手动分流和自动分流两种方案。

OLTP与OLAP业务混合负载场景下行存和列存自动分流

如果您的业务中，OLAP类型与OLTP类型的请求是通过同一个应用程序访问数据库，此时可实现将两类业务的读请求按照扫描的行数进行自动分流，分别分流到列存节点或行存节点。

分流规则：

- OLTP类业务：一般包含读和写的请求。写请求统一由主节点处理，读请求由只读行存节点或主节点处理。
- OLAP类业务：一般仅包含读请求。读请求统一由只读列存节点处理。

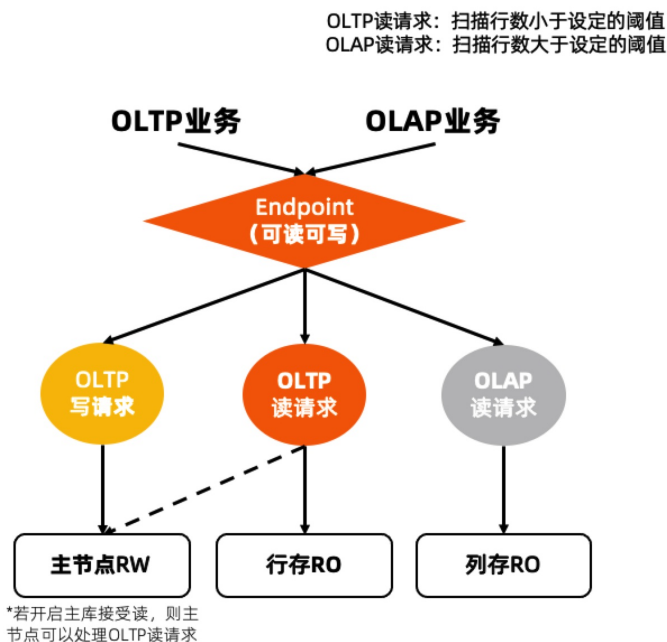
自动分流方案（读写模式需为可读可写（自动读写分离））：

- 主节点和只读列存节点分流：由于主节点也属于行存节点，因此也可处理OLTP类读请求。该方案下，写请求以及OLTP类读请求分流至主节点，OLAP类读请求分流至只读列存节点。
- 只读行存节点和只读列存节点分流：该方案下，写请求依旧分流至主节点，OLTP类读请求分流至只读行存节点，OLAP类读请求分流至只读列存节点。

自动分流判断标准：

- 低于扫描行数阈值的SQL请求将被引流至行存节点上（或者主节点）执行。多个行存节点的情况下，具体引流至哪个行存节点，根据负载均衡自动判定。
- 高于扫描行数阈值的SQL请求将被引流至列存节点上执行。多个列存节点的情况下，具体引流至哪个列存节点，根据负载均衡自动判定。

OLTP/OLAP业务混合负载场景下自动引流



❓ 说明 若要实现行存和列存自动分流，不建议将集群地址的读写模式设置为只读。因为只读模式下，业务会按连接负载均衡，无法保证OLAP业务请求会路由到列存节点，因此业务会出现时快时慢的情况。

具体配置方式可参见[行存和列存自动分流](#)。

OLTP与OLAP业务独立场景下行存和列存手动分流

若您的业务中，OLAP类型与OLTP类型的业务基于不同应用程序访问数据库，则可分别为这些应用程序配置不同的集群地址，然后将行存节点和列存节点分别配置到不同集群地址（Endpoint）的服务节点中，从而实现行存和列存分流。

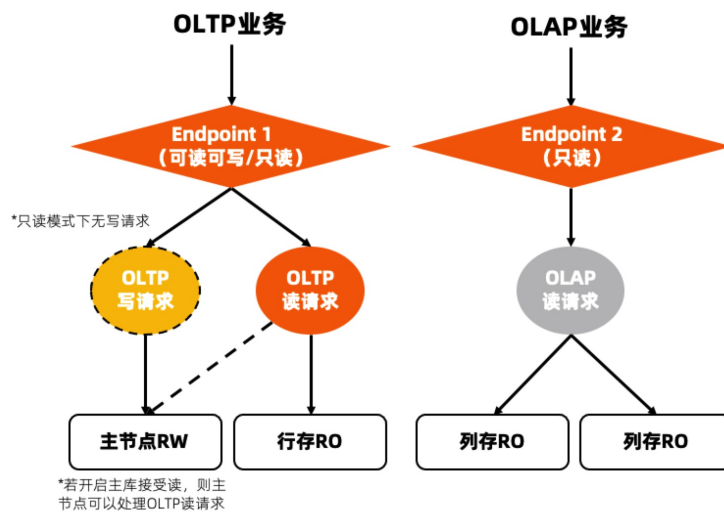
分流规则：

- OLTP类业务：一般包含读和写的请求。写请求统一由主节点处理，读请求由只读行存节点或主节点处理。
- OLAP类业务：一般仅包含读请求。读请求统一由只读列存节点处理。

手动分流方案（读写模式为可读可写（自动读写分离）或只读均可）：

- 将OLTP类的应用与不含只读列存节点的集群地址进行关联，读请求由主节点或只读行存节点处理。
- 将OLAP类的应用与仅包含只读列存节点的集群地址进行关联，读请求由只读列存节点处理。

OLTP/OLAP业务独立场景下手动引流



具体配置方式可参见[行存和列存手动分流](#)。

17.1.3.2. 行存和列存自动分流

如果您的业务中，OLAP类型的查询与OLTP类型的查询通过同一个应用程序访问数据库，此时如果要实现分流，集群地址的读写模式需设置为可读可写（自动读写分离），并开启自动分流功能。自动分流功能开启后，数据库代理基于业务流入的查询扫描行数来实现自动分流，从而实现SQL查询的最大性能。当SQL请求的扫描行数超过一定数量时，数据库代理可自动将该请求分流至只读列存节点处理，反之该请求则由行存节点或主节点来处理。

自动分流方案说明

PolarDB MySQL引擎的数据库代理功能支持将SQL语句的扫描行数作为判断指标，通过判断扫描行数是否会超过特定阈值（该阈值可配置），决定将该SQL请求分流到行存或是列存节点上处理，从而发挥出行存和列存的最佳性能。

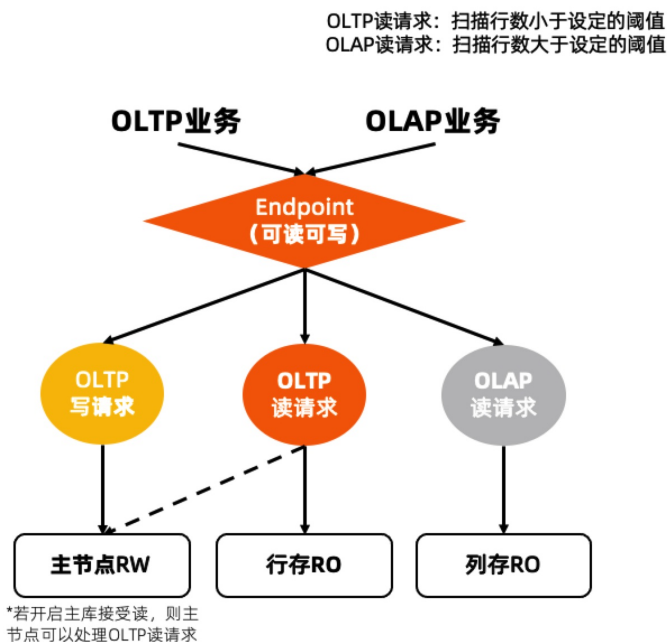
分流规则：

- OLTP类业务：一般包含读和写的请求。写请求统一由主节点处理，读请求由只读行存节点或主节点处理。
- OLAP类业务：一般仅包含读请求。读请求统一由只读列存节点处理。

自动分流方案（读写模式需为可读可写（自动读写分离））：

- 主节点和只读列存节点分流：由于主节点也属于行存节点，因此也可处理OLTP类读请求。该方案下，写请求以及OLTP类读请求分流至主节点，OLAP类读请求分流至只读列存节点。
- 只读行存节点和只读列存节点分流：该方案下，写请求依旧分流至主节点，OLTP类读请求分流至只读行存节点，OLAP类读请求分流至只读列存节点。

OLTP/OLAP业务混合负载场景下自动引流



使用限制

- 仅当集群地址的读写模式为可读可写（自动读写分离），且服务节点包含了至少一个只读列存节点时，支持列存和行存自动引流。
- 当集群地址的读写模式为只读时，暂不支持列存和行存自动引流功能，而是按连接负载均衡，导致无法保证复杂的业务请求会路由到列存节点。因此，若该模式下服务节点同时包含列存节点和行存节点，业务会出现时快时慢的情况。

步骤1：开启列存和行存自动引流功能

1. 在集群基本信息页的数据库代理企业版区域，找到目标集群地址，单击目标集群地址名称右侧的**编辑配置**。
2. 设置读写模式为可读可写（自动读写分离）。
3. 在**服务节点**中，选择主节点和需要参与处理请求的行存和列存只读节点。

说明 服务节点中需要至少选择一个只读列存节点。

示例1：在下图中，除了主节点外，服务节点还包含1个只读节点（即只读行存节点）和2个只读列存节点。若此时开启自动引流功能后，写请求将引流至主节点，OLAP类读请求则引流至只读列存节点，OLTP类读请求则引流至只读行存节点。

集群设置

读写模式 ☐ 只读 ☒ 可读可写（自动读写分离）

地址名称 HTAP自动引流示例 10/127

服务节点

可选节点	已选节点
无节点	☐ 主节点 : [模糊] ☐ 只读节点 : [模糊] ☐ 只读列存节点 : [模糊] ☐ 只读列存节点 : [模糊]
☐ 0 项	☐ 4 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。

示例2：在下图中，服务节点包括主节点和1个只读列存节点。若此时开启自动引流功能后，写请求将引流至主节点，OLAP类读请求将引流至只读列存节点，OLTP类读请求将引流至主节点。

集群设置

读写模式 ☐ 只读 ☒ 可读可写（自动读写分离）

地址名称 请输入集群地址名称 0/127

服务节点

可选节点	已选节点
☐ 只读节点 : [模糊] ☐ 只读列存节点 : [模糊]	☐ 主节点 : [模糊] ☐ 只读列存节点 : [模糊]
☐ 2 项	☐ 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。

? 说明 可读可写（自动读写分离）模式下，无论主节点是否已被添加在服务节点中，所有写请求只会发往主节点。

4. 在右侧HTAP优化类别中，开启行存/列存自动引流。

编辑地址配置 (pe-xxxxxx-xxxxxx) 如何设置集群地址? ×

网络信息

私网 pc-xxxxxx-xxxxxx-xxxxxx.com:3306 [复制](#) [更多](#) [v](#)

公网 暂无数据 [申请](#)

经典网络 暂无数据 [申请](#)

集群设置

读写模式 ☐ 只读 ☒ 可读可写 (自动读写分离)

地址名称 4/127

服务节点

可选节点

无节点

☐ 0 项

已选节点

☐ 主节点: xxxxxxxx-xxxxxx-xxxxxx-xxxxxx
☐ 只读节点: xxxxxxxx-xxxxxx-xxxxxx-xxxxxx
☐ 只读列存节点: xxxxxxxx-xxxxxx-xxxxxx-xxxxxx
☐ 只读列存节点: xxxxxxxx-xxxxxx-xxxxxx-xxxxxx

☐ 4 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

负载均衡设置

负载均衡策略 基于负载的自动调度

主库是否接受读 ☐ 否 ☒ 是

事务拆分 ☐ 开启 ☒ 关闭

一致性设置

一致性级别 ☒ 最终一致性 (弱) ☐ 会话一致性 (中) ☐ 全局一致性 (强)

连接池设置

连接池 ☒ 关闭 ☐ 会话级 ☐ 事务级

HTAP优化

并行查询 为了避免影响主库，读写地址暂不支持。

行存/列存自动引流 ☒ 开启 ☐ 关闭

动态脱敏

全部规则 0, 启用规则 0, [规则管理](#)

禁用规则 0

5. 设置其他数据库代理相关的参数，具体可参见[配置数据库代理](#)。

步骤2：配置自动引流阈值

开启了列存和行存自动引流后，您需要设置SQL语句的扫描行数的阈值。阈值配置完成后，数据库代理将此阈值作为判断依据。若业务侧请求的扫描行数大于该阈值，则将该请求引流至列存节点执行；若小于该阈值，则将该请求引流至行存节点执行。

当前SQL语句的扫描行数的阈值由下表中的参数决定，您可以在集群的[参数配置](#)页面，根据自身业务情况修改参数的值，从而调整自动分流效果。

参数名称	描述
loose_imci_ap_threshold	行存节点或列存节点当前SQL语句的扫描行数的阈值。默认值：500000。 ? 说明 如果该参数值大于500000，则路由至列存节点。
loose_cost_threshold_for_imci	列存节点内部当前SQL语句的扫描行数的阈值。默认值：50000。 ? 说明 如果该参数值大于50000，则选择列存执行计划。否则，选择行存执行计划。

您可以通过 `show status like 'Last_query_imci_cost'` 命令精确查询上一条SQL语句的执行代价，从而判断如何调整上表中的参数值。

例如，执行以下语句，可查看上一条SQL语句的执行代价：

```
show status like 'Last_query_imci_cost';
```

查询结果如下：

```
+-----+-----+
| Variable_name | Value |
+-----+-----+
| Last_query_imci_cost | 2 |
+-----+-----+
1 row in set (0.01 sec)
```

从以上查询结果可以看出，该SQL语句的执行代价为2。

此时若要实现将该SQL查询分流至列存节点的列存计划执行，则需要将loose_imci_ap_threshold和loose_cost_threshold_for_imci参数的值设置为1。

17.1.3.3. 行存和列存手动分流

如果您的业务中，OLAP类型与OLTP类型的业务基于不同的应用程序访问数据库，则可分别为这些应用程序配置不同的集群地址，然后将行存节点和列存节点分别配置到不同集群地址（Endpoint）的服务节点中，从而实现行存和列存分流。

手动分流方案说明

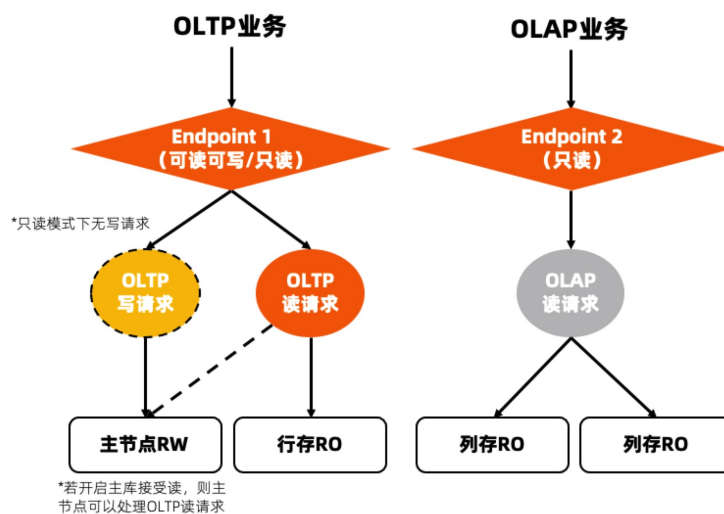
分流规则：

- OLTP业务：一般包含读和写的请求。写请求统一转发至主节点处理，读请求由只读行存节点或主节点处理。
- OLAP业务：一般仅包含读请求。读请求统一由只读列存节点处理。

手动分流方案（读写模式为可读可写（自动读写分离）或只读均可）：

- 将OLTP类的应用与不含只读列存节点的集群地址进行关联，读请求由主节点或只读行存节点处理。
- 将OLAP类的应用与仅包含只读列存节点的集群地址进行关联，读请求由只读列存节点处理。

OLTP/OLAP业务独立场景下手动引流



操作步骤

您需要为OLTP类业务和OLAP类业务分别配置不同的集群地址，配置集群地址的方式可参见[申请集群地址和主地址](#)。

在配置集群地址的过程中，有如下注意事项：

- 针对OLTP类业务的集群地址：
 - 若在只读模式下，服务节点只需包含只读行存节点。
 - 若在可读可写（自动读写分离）模式下，服务节点建议包含至少一个只读行存节点。此时若开启了主库是否接受读，则读请求也会发往主节点。

 **说明** 可读可写（自动读写分离）模式下，无论主节点是否已被添加在服务节点中，所有写请求只会发往主节点。

- 针对OLAP类业务的集群地址：由于OLAP类业务一般仅包含读请求，因此建议设置为只读模式。该模式下服务节点需要包含至少一个只读列存节点。

17.1.4. 列存索引语法说明

17.1.4.1. 使用建表语句创建列存索引

本文介绍了如何在建表的时候创建列存索引。

前提条件

在添加完只读列存节点，并配置好对应的集群地址后，您可以通过集群地址连接到集群，使用SQL语句创建并管理列存索引。

- [添加只读列存节点](#)。
- 配置集群地址。为了实现行存和列存的分流，目前有手动和自动两种分流方案。您可以根据业务情况选择一种[分流方案](#)，并配置集群地址。
- [通过集群地址连接到数据库集群](#)。

创建列存索引

- CREATE TABLE语法：
 - 在建表时，您只需要在CREATE TABLE语句的COMMENT字段里增加COLUMNAR=1字符串，即可创建列存索引。其余语法均不变，且不受影响。
 - COLUMNAR=1可以单独加在列的COMMENT字段中，单独对该列生效；也可以加在CREATE TABLE语句末尾的COMMENT字段中，对表中所有支持的数据类型的列生效。

 说明

- 8.0.1.1.25版本及之后的版本，列存索引支持BLOB和TEXT类型。
- 列存索引暂不支持ENUM、SET、BIT和地理信息等数据类型。

示例：

```
CREATE TABLE t1(  
  col1 INT COMMENT 'COLUMNAR=1',  
  col2 DATETIME COMMENT 'COLUMNAR=1',  
  col3 VARCHAR(200)  
) ENGINE InnoDB;  
CREATE TABLE t2(  
  col1 INT,  
  col2 DATETIME,  
  col3 VARCHAR(200)  
) ENGINE InnoDB COMMENT 'COLUMNAR=1';
```

从以上示例中可以看到：

- 表 `t1` 的建表语句中，为指定列（`col1`和`col2`）创建了列存索引。
- 表 `t2` 的建表语句中，为全表创建了列存索引，即覆盖了表中的所有数据类型能够支持的列，即 `col1`、`col2`、`col3`。
- **CREATE TABLE LIKE**语法：若使用**CREATE TABLE LIKE**命令创建表，如果源表包含列存索引，则目标表也将包含相同的列存索引。
- **CREATE TABLE ... SELECT**语法：若使用**CREATE TABLE ... SELECT**命令创建表，则可在表的**COMMENT**字段里增加**COLUMNAR=1**字符串，为全表的列创建列存索引。但使用**CREATE TABLE ... SELECT**命令创建表时，无法为指定的列单独创建列存索引。

示例：

```
CREATE TABLE t3(  
  col1 INT,  
  col2 DATETIME,  
  col3 VARCHAR(200)  
) ENGINE InnoDB;  
CREATE TABLE t4 COMMENT 'COLUMNAR=1' SELECT col1, col2 FROM t3;
```

从以上示例中可以看到：

表 `t4` 的**COMMENT 'COLUMNAR=1'**是对全表生效的，即对 `t4` 的所有列（`col1`,`col2`，以及后续新增的列）创建列存索引。

查看表的列存索引结构信息

- 语法：可以使用 `SHOW CREATE TABLE <tablename> FULL` 语句查看表的列存索引的结构信息。
- 示例：

```

SHOW CREATE TABLE test.t1\G
***** 1. row *****
      Table: t1
Create Table: CREATE TABLE `t1` (
  `id` int(11) NOT NULL,
  `coll` int(11) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='columnar=1'
SHOW CREATE TABLE test.t1 FULL\G
***** 1. row *****
      Table: t1
Create Table: CREATE TABLE `t1` (
  `id` int(11) NOT NULL,
  `coll` int(11) DEFAULT NULL,
  PRIMARY KEY (`id`),
  COLUMNAR INDEX (`id`,`coll`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='columnar=1'

```

从以上示例中可以看到：

- `SHOW CREATE TABLE <tablename>` 命令默认不输出COLUMNAR INDEX的定义，只会展示COMMENT信息。
- `SHOW CREATE TABLE <tablename> FULL` 命令可以输出COLUMNAR INDEX的列存索引的结构信息。

17.1.4.2. 使用DDL语句动态增删列存索引

本文介绍了如何在建表之后，通过DDL语句动态创建和删除列存索引。

前提条件

在添加完只读列存节点，并配置好对应的集群地址后，您可以通过集群地址连接到集群，使用SQL语句创建并管理列存索引。

- **添加只读列存节点。**
- **配置集群地址。**为了实现行存和列存的分流，目前有手动和自动两种分流方案。您可以根据业务情况选择一种**分流方案**，并配置集群地址。
- **通过集群地址连接到数据库集群。**

创建列存索引

- 语法：
 - 可在ALTER TABLE语句后增加COMMENT 'COLUMNAR=1'字段，为表创建对全表生效的列存索引。
 - 可在ALTER TABLE ... MODIFY COLUMN ...语句后增加COMMENT 'COLUMNAR=1'字段，为指定列创建列存索引。

② 说明

- 8.0.1.1.25版本及之后的版本，列存索引支持BLOB和TEXT类型。
- 列存索引暂不支持ENUM、SET、BIT和地理信息等数据类型。

- 示例：

```
CREATE TABLE t5(
  col1 INT,
  col2 DATETIME,
  col3 VARCHAR(200)
) ENGINE InnoDB;
-- 创建默认列存索引
ALTER TABLE t5 COMMENT 'COLUMNAR=1';
-- 创建指定列的列存索引
ALTER TABLE t5 MODIFY COLUMN col1 INT COMMENT 'COLUMNAR=1',
MODIFY COLUMN col2 DATETIME COMMENT 'COLUMNAR=1';
```

删除列存索引

- 语法：
 - 可在ALTER TABLE语句后增加COMMENT 'COLUMNAR=0'字段，删除对全表生效的列存索引。
 - 可在ALTER TABLE ... MODIFY COLUMN ...语句后增加COMMENT 'COLUMNAR=0'字段，删除指定列的列存索引。
- 示例：

```
-- 创建指定列的列存索引
CREATE TABLE t6(
  col1 INT COMMENT 'COLUMNAR=1',
  col2 DATETIME COMMENT 'COLUMNAR=1',
  col3 VARCHAR(200)
) ENGINE InnoDB;
-- 删除指定列的列存索引
ALTER TABLE t6 MODIFY COLUMN col1 INT COMMENT 'COLUMNAR=0',
MODIFY COLUMN col2 DATETIME COMMENT 'COLUMNAR=0';
-- 创建默认列存索引
CREATE TABLE t7(
  col1 INT,
  col2 DATETIME,
  col3 VARCHAR(200)
) ENGINE InnoDB COMMENT 'COLUMNAR=1';
-- 删除默认列存索引
ALTER TABLE t7 COMMENT 'COLUMNAR=0';
```

修改列存索引定义

- 语法：
 - 可在ALTER TABLE ... MODIFY COLUMN ...语句后增加COMMENT 'COLUMNAR=1'字段，为列存索引增加一列。
 - 可在ALTER TABLE ... MODIFY COLUMN ...语句后增加COMMENT 'COLUMNAR=0'字段，从列存索引删除一列。

② 说明

- 8.0.1.1.25版本及之后的版本，列存索引支持BLOB和TEXT类型。
- 列存索引暂不支持ENUM、SET、BIT和地理信息等数据类型。

- 示例：

```
CREATE TABLE t8(
  col1 INT COMMENT 'COLUMNAR=1',
  col2 DATETIME COMMENT 'COLUMNAR=1',
  col3 VARCHAR(200)
) ENGINE InnoDB;
-- 为列存索引新增一列
ALTER TABLE t8 MODIFY COLUMN col3 VARCHAR(200) COMMENT 'COLUMNAR=1';
-- 从列存索引删除一列
ALTER TABLE t8 MODIFY COLUMN col2 DATETIME COMMENT 'COLUMNAR=0';
```

创建覆盖多列的列存索引

在OLAP应用中，涉及到的表大多数为大宽表。您可以通过表的COMMENT来简化大宽表创建列存索引的方式，即默认为宽表在所有支持的数据类型的列上创建列存索引，同时可以只指定少量的无需有列存索引覆盖的列。

说明

- 8.0.1.1.25版本及之后的版本，列存索引支持BLOB和TEXT类型。
- 列存索引暂不支持ENUM、SET、BIT和地理信息等数据类型。

例如，若使用如下语句建表：

```
CREATE TABLE t9(
  col1 INT, col2 INT, col3 INT,
  col4 DATETIME, col5 TIMESTAMP,
  col6 CHAR(100), col7 VARCHAR(200),
  col8 TEXT, col9 BLOB
) ENGINE InnoDB;
```

可使用如下命令为表创建列存索引：

```
ALTER TABLE t9 COMMENT 'COLUMNAR=1', MODIFY COLUMN col7 VARCHAR(200) COMMENT 'COLUMNAR=0';
```

结果如下：

```
SHOW CREATE TABLE t9 FULL\G
***** 1. row *****
      Table: t9
Create Table: CREATE TABLE `t9` (
  `col1` int(11) DEFAULT NULL,
  `col2` int(11) DEFAULT NULL,
  `col3` int(11) DEFAULT NULL,
  `col4` datetime DEFAULT NULL,
  `col5` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  `col6` char(100) DEFAULT NULL,
  `col7` varchar(200) DEFAULT NULL COMMENT 'COLUMNAR=0',
  `col8` text,
  `col9` blob,
  COLUMNAR INDEX (`col1`,`col2`,`col3`,`col4`,`col5`,`col6`,`col8`,`col9`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT 'COLUMNAR=1'
```

从以上示例可以看到：col7列被指定过滤掉，没有被列存索引覆盖。

但受限于InnoDB Online DDL的实现方式，上述示例中的DDL语句 `ALTER TABLE t9 COMMENT 'COLUMNAR=1', MODIFY COLUMN col7 VARCHAR(200) COMMENT 'COLUMNAR=0';` 会以online rebuild的方式实现，从而性能较差。您可以尝试调整为以下方式：

```
-- 先过滤掉无需覆盖的列，只修改列的COMMENT属性
ALTER TABLE t9 MODIFY COLUMN col7 VARCHAR(200) COMMENT 'COLUMNAR=0';
-- 修改表COMMENT属性并且添加默认列存索引
ALTER TABLE t9 COMMENT 'COLUMNAR=1';
```

增加列的同时添加列存索引

您可以在使用ALTER TABLE ADD COLUMN时，在COMMENT中指定COLUMNAR属性，从而达到在列上创建列存索引的目的。

说明

- 8.0.1.1.25版本及之后的版本，列存索引支持BLOB和TEXT类型。
- 列存索引暂不支持ENUM、SET、BIT和地理信息等数据类型。

例如，若使用如下语句建表，建表时创建列存索引，并覆盖col1和col2两列：

```
CREATE TABLE t10 (
  col1 INT COMMENT 'COLUMNAR=1',
  col2 DATETIME COMMENT 'COLUMNAR=1',
  col3 VARCHAR(200)
) ENGINE InnoDB;
```

可执行如下命令，为表 t10 新加一列col4，并且被列存索引覆盖：

```
ALTER TABLE t10 ADD col4 DATETIME DEFAULT NOW() COMMENT 'COLUMNAR=1';
```

由于涉及到列存索引的变更，因此不再是INSTANT DDL。该DDL语句在加列的同时会删除旧的列存索引，并且新建覆盖col1、col2、col4的列存索引。

查看索引状态

在启用列存索引特性后，当前OLAP查询请求会发给只读列存节点，而不会发给主节点，从而保证OLAP和OLTP的计算资源隔离。基于这种隔离限制，将新建列存索引或者修改列存索引定义的Online DDL优化为异步DDL，其逻辑是在主节点上完成表及索引的元数据修改操作，然后通过Redo日志同步给只读列存节点。只读列存节点会在数据字典修改生效后启动后台线程并发构建列存索引。

异步DDL的逻辑效果是在RW节点上成功执行DDL语句并且在数据字典上已生效。但列存索引在构建完成前无法提供查询服务。您会发现，在执行完相应DDL语句后，若立即进行OLAP查询，依旧是通过行存的方式执行。需要等待列存索引构建完成后，再进行OLAP查询，列存索引才会生效。

您可以在只读列存节点上查询INFORMATION_SCHEMA.IMCI_INDEXES获取列存索引的创建状态。

例如，若使用如下语句建表：

```
CREATE TABLE t11 (
  col1 INT, col2 DATETIME, col3 VARCHAR(200)
) ENGINE InnoDB;
```

然后执行如下DDL语句创建列存索引：

```
ALTER TABLE t11 COMMENT 'COLUMNAR=1';
```

该DDL语句的效果类似于INSTANT DDL，会在主节点上很快执行完成。但此时查询：

```
SELECT * FROM INFORMATION_SCHEMA.IMCI_INDEXES WHERE TABLE_NAME = 't11';
```

可发现返回结果的STATE字段并不是COMMITTED（如RECOVERING），则表示列存索引依旧在创建过程中。

TABLE_ID	SCHEMA_NAME	TABLE_NAME	NUM_COLS	PACK_SIZE	ROW_ID	STATE	MEM_SIZE
xxxx	test	t11	3	65536	0	RECOVERING	0

若STATE为COMMITTED，则表示列存索引已创建完成。此时进行OLAP查询，则会通过列存索引执行。

17.1.4.3. 设置压缩算法

为了提高列存模式下的压缩效率，降低存储成本，您可以为创建的列存索引设置压缩算法。本文介绍如何设置和修改列存索引的压缩算法。

支持的压缩算法

当前PolarDB MySQL引擎支持两种IMCI的压缩算法：LZ4压缩、ZSTD压缩。

- LZ4：一种无损数据压缩算法，压缩速度大于每核500 MB/s。具体可参见[LZ4 GitHub](#)。
- ZSTD（全称Zstandard）：一种无损数据压缩算法，压缩速度与LZ4相当。具体可参见[Zstandard GitHub](#)。

建表时设置压缩算法

- 语法：
在建表时，您只需要在CREATE TABLE语句的COMMENT字段里设置codec_opt参数，即可设置IMCI的压缩算法。

```
COMMENT 'COLUMNAR=1 codec_opt={LZ4}'
```

codec_opt有效的取值为 `LZ4`、`ZSTD` 和 `NONE`，对应LZ4压缩、ZSTD压缩和不使用压缩。

说明

- imci_default_codec参数定义了缺省压缩算法。当COMMENT中缺省codec_opt时，将根据该参数的值采用对应的压缩算法。该参数默认值为 `ZSTD`。
- 您可以执行SET命令在SESSION级别修改imci_default_codec的值，或者[提交工单](#)申请全局性修改该值。

在表的COMMENT字段中指定的压缩算法将应用在表的所有列索引上。如果同时在列的COMMENT中设置了压缩算法，将以列的配置优先。

- 示例：

```
CREATE TABLE t12 (
  col1 INT,
  col2 DATETIME,
  col3 VARCHAR(200)
) ENGINE InnoDB COMMENT 'COLUMNAR=1 codec_opt={LZ4}';
SET imci_default_codec="{LZ4}";
CREATE TABLE t13 (
  col1 INT COMMENT 'codec_opt={NONE}',
  col2 DATETIME,
  col3 VARCHAR(200) 'codec_opt={ZSTD}',
) ENGINE InnoDB COMMENT 'COLUMNAR=1';
```

从以上示例中可以看到：

- 表 `t12` 使用 `COMMENT` 字段指定了表的默认压缩算法为 LZ4。col1、col2 和 col3 列的数据都使用 LZ4 算法进行压缩。
- 表 `t13` 未设置表的压缩算法，col1 单独配置为不开启压缩；col2 未设置 `codec_opt`，将使用参数 `imci_default_codec` 配置的压缩算法，使用 LZ4 算法压缩；col3 设置 `codec_opt={ZSTD}`，由于列的配置优先级高于表以及默认配置，因此 col3 使用 ZSTD 压缩算法。

修改列的压缩算法

目前尚不支持修改单列数据的压缩算法。若要修改，需要删除原来的数据，再使用新的建表语句，指定列的新压缩算法。修改单列数据的压缩方式将在后续版本中推出。

示例：

```
CREATE TABLE t14 (
  col1 INT COMMENT 'COLUMNAR=1 codec_opt={ZSTD}',
  col2 DATETIME COMMENT 'COLUMNAR=1 codec_opt={ZSTD}',
  col3 VARCHAR(200)
) ENGINE InnoDB;
-- 修改列索引的压缩算法
ALTER TABLE t14 COMMENT 'COLUMNAR=1', MODIFY COLUMN col2 DATETIME COMMENT 'codec_opt={LZ4}';
```

在以上示例中，表 `t14` 的 col2 列上已经建立列索引，压缩算法为 ZSTD。若执行示例中的命令尝试将其修改为 LZ4，不会立即生效，需要重建数据时才会生效。由于该命令不会触发数据重建，因此数据（包括后续新增数据）仍使用之前的指定的压缩方式。

18.多机并行（MPP）

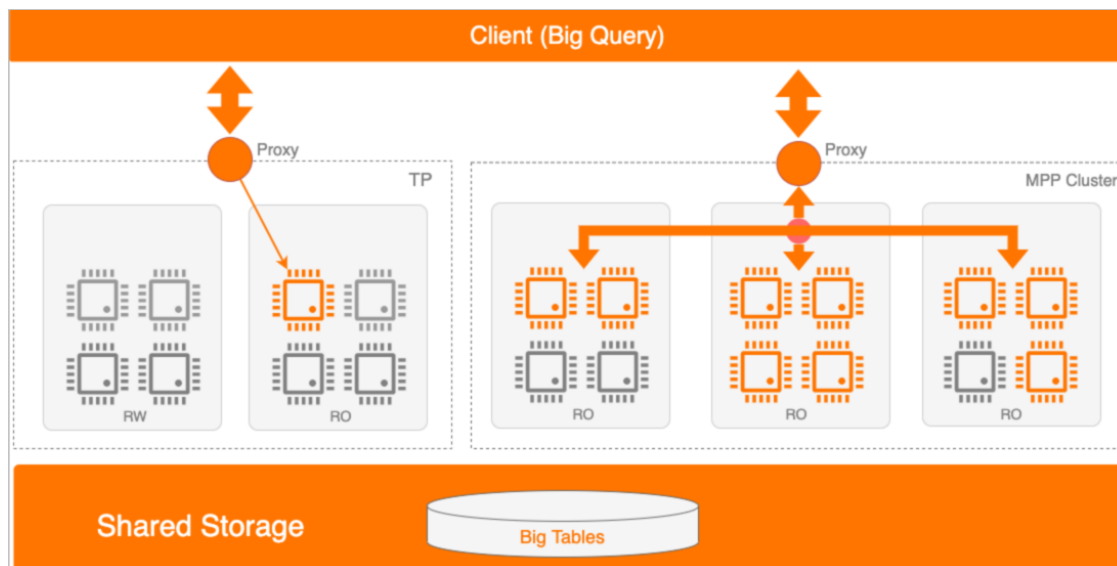
18.1. 概述

PolarDB MySQL引擎重磅推出了多机并行（MPP）计算引擎，可以将分析型查询的计算任务分发到集群内的任意节点执行，提升集群资源的利用率，从而大幅提升数据库的整体查询性能。本文介绍了多机并行（MPP）计算引擎的技术原理、核心优势和适用场景等内容。

简介

对于复杂分析型查询场景，PolarDB MySQL引擎已支持的**并行查询（Parallel Query）**特性提供了对大查询进行并行加速的能力，但它仅能利用单机多核CPU进行并行加速。节点规格（CPU核数）将成为制约加速能力的瓶颈，而且集群内其它节点即便有空闲计算的资源，也没有被有效利用而造成了浪费。

PolarDB MySQL引擎重磅推出了多机并行（MPP）计算引擎，可以将分析型查询的计算任务分发到集群内的任意节点执行。一是能够突破单机并行加速的瓶颈；二是能够有效利用集群内的空闲计算资源，提升集群资源的利用率，从而大幅提升数据库的整体查询性能，更好的满足复杂分析型查询场景的低时延需求。如下图所示：



版本要求

PolarDB集群版本需为PolarDB MySQL引擎8.0.2版本且修订版本需为8.0.2.2.3或以上。

如何查看集群版本，请参见[查询版本号](#)。

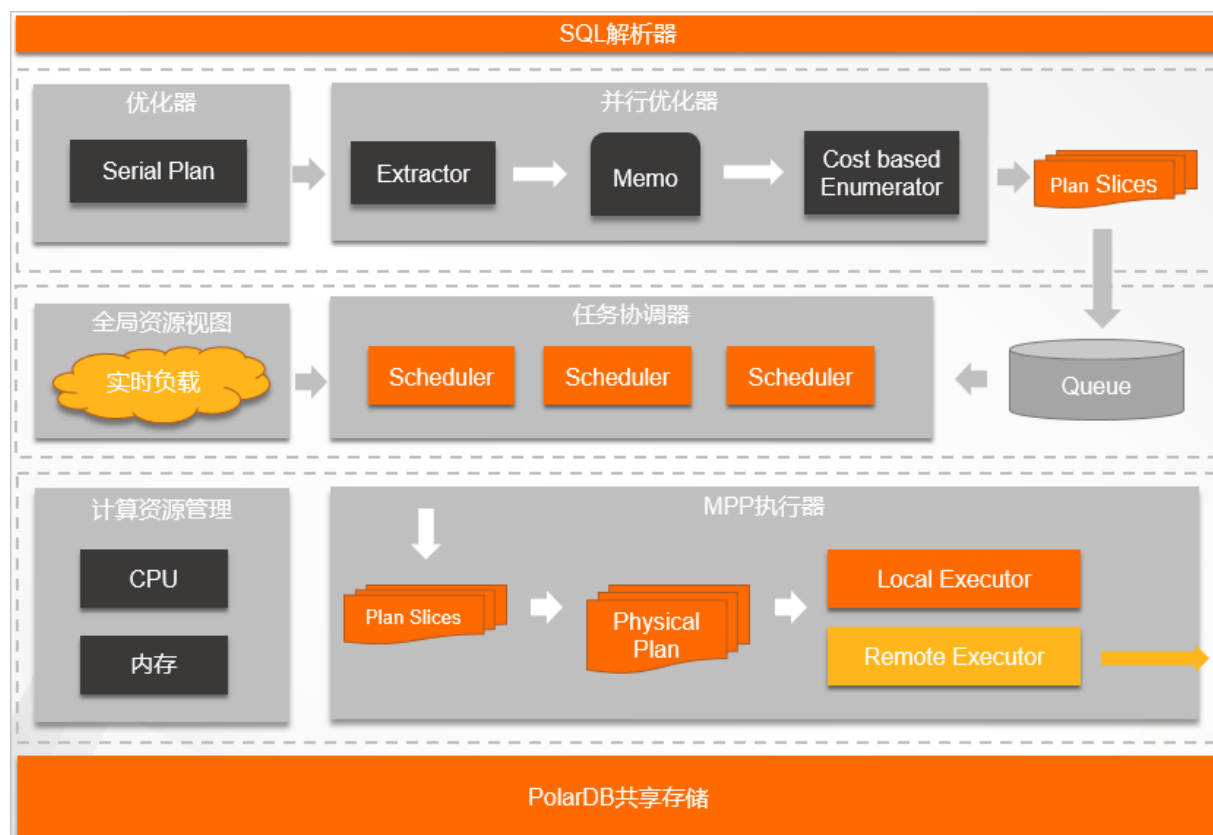
说明 多机并行（MPP）已支持2022年6月28日后新购的集群，存量集群需要[提交工单](#)联系技术支持进行处理。

费用说明

该功能可免费使用。您只需要支付每个PolarDB集群自身的费用，关于PolarDB集群自身费用，请参见[计费项概览](#)。

技术原理

多机并行（MPP）的目标是打通节点间的计算资源，基本原理是将一个复杂查询任务拆分为多个子任务，子任务可以被派发到同集群内的任意节点来完成计算，从而有效利用集群内其它节点的空闲计算资源（CPU、内存等）来加速查询。基本原理如下图所示：



- 并行优化器：在串行计划的基础上提取信息，然后基于代价的枚举并拆分出最优的子计划切片（Plan Slices）。
- 全局资源视图：维护集群内所有节点的实时负载信息。
- 任务协调器：根据实时资源负载情况，调度执行任务队列。
- MPP执行器：负责将被调度的子计划切片（Plan Slices）生成对应的Physical Plan，然后交给Executor模块执行。如果是远程节点执行Physical Plan，还需要将Physical Plan序列化为二进制协议，通过内部网络通道传输到远程节点上执行。
- PolarDB基于共享存储的MPP架构，在跨节点一致性视图机制的保证下，子任务在任意节点执行都能读取到正确的数据。

核心优势

多机并行（MPP）能够有效利用集群内所有节点的计算资源进行并行加速。具体优势如下：

- 实时性分析，统一的底层存储，数据实时可见；
- 零附加成本和运维成本，随集群部署，开箱即用；
- 打通节点间的计算资源，突破单机硬件性能瓶颈，优异的性能表现；
- 空闲计算资源得到充分利用，提升集群整体资源利用率；
- 按需扩容，更灵活的弹性计算能力。

适用场景

- 所有适用于单机并行（并行查询）的应用场景。详情请参见[应用场景](#)。

多机并行（MPP）是单机并行（并行查询）的下一代演进版本。所以，所有单机并行（并行查询）的适用场景，同样适用于多机并行（MPP）。

- 海量数据分析查询

当面对大查询或者复杂查询时，单节点并行加速由于资源受限可能无法满足数据实时延迟要求。此时，您可以选择开启多机并行（MPP）引擎，突破单节点CPU、内存的资源瓶颈，并借助多节点的IO吞吐能力进一步利用共享存储的高带宽，获得更好的执行效率。

- 弹性计算场景

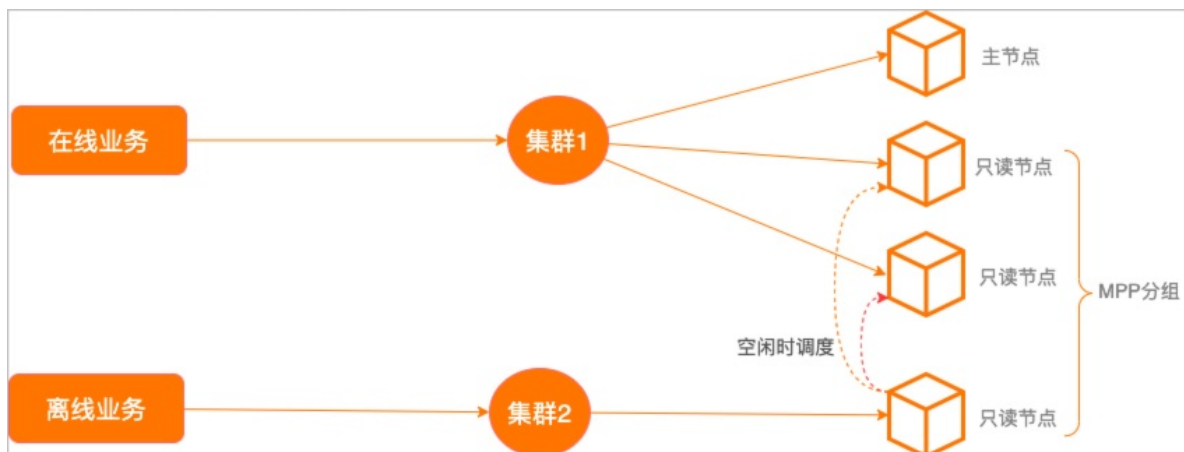
弹性计算是PolarDB MySQL引擎的核心能力，自动扩容功能提供了对短查询业务类型非常友好的弹性能力，但并不适用于离线分析类业务。因为对于大查询业务场景，无法通过增加节点获得更多的计算能力。开启多机并行（MPP）引擎的集群，新增加的节点也会自动加入到MPP集群分组中，正好弥补了PolarDB弹性能力的这一块短板。

- 集群内资源负载不均衡场景

集群内的多个只读节点，借助数据库代理的负载均衡能力可以使每个节点的QPS大致相同。但是，由于不同节点上的查询计算量是有差异的，不同节点间依然会出现负载不均衡的现象。此时，开启多机并行（MPP）引擎后，当节点计算资源不足时，可以将查询的部分任务或者全部任务调度到有空闲计算资源的节点上执行。从而提升集群的整体资源利用率，也降低了查询的时延。

- 在离线业务混合场景

一般在线业务和离线业务会连接不同的只读集群地址，避免相互影响。开启多机并行（MPP）引擎后，离线业务还可以有效利用在线业务低峰时段的空闲计算资源。



性能提升情况

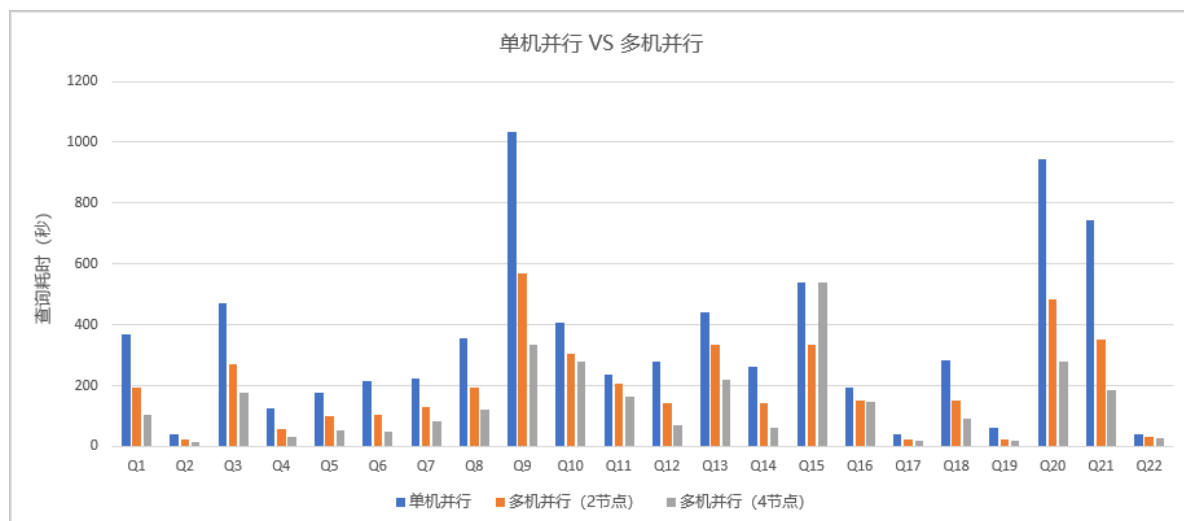
相对于串行查询模式，开启单机并行（并行查询）后，TPC-H的查询性能加速比已有约19倍，具体请参见[性能指标](#)。多机并行（MPP）在单机并行（并行查询）性能提升的基础上，随着节点数的扩展查询性能得到进一步的线性提升。

- **测试方法：**TPC-H是业界常用的一套基准，由TPC委员会制定发布，用于评测数据库的分析型查询能力。TPC-H查询包含8张数据表、22条复杂的SQL查询，大多数查询包含若干表Join、子查询和Group by聚合等。详细的测试流程请参见[性能测试方法（OLAP）](#)。

② 说明 本文的TPC-H的实现基于TPC-H的基准测试，并不能与已发布的TPC-H基准测试结果相比较，本文中的测试并不符合TPC-H基准测试的所有要求。

- **测试数据量：**1 TB
- **测试结果：**

下图展示了在32核256 GB独享规格的集群中，分别开启单机并行（并行查询）、多机并行（2节点）和多机并行（4节点）后执行TPC-H的22条查询语句的性能对比情况。



18.2. 使用说明

多机并行（MPP）计算引擎，可以将分析型查询的计算任务分发到集群内的任意节点执行，提升集群资源的利用率，从而大幅提升数据库的整体查询性能。本文介绍如何使用多机并行（MPP）计算引擎。

版本要求

PolarDB集群版本需为PolarDB MySQL引擎8.0.2版本且修订版本需为8.0.2.2.3或以上。

如何查看集群版本，请参见[查询版本号](#)。

说明 多机并行（MPP）已支持2022年6月28日后新购的集群，存量集群需要[提交工单](#)联系技术支持进行处理。

使用限制

- 如下场景将无法使用多机并行（MPP）：
 - 查询隔离级别为READ UNCOMMITTED；
 - 查询语句中包含Blob列；
 - 查询语句中存在非InnoDB引擎的用户表。
- 多机并行（MPP）的使用限制以及与串行执行结果的兼容性问题，与单机并行（并行查询）一致，详情请参见[并行查询使用限制与串行执行结果兼容问题](#)。

参数说明

参数名	级别	描述
polar_cluster_group	Session	MPP集群分组。每个集群地址对应一个MPP集群分组，MPP集群分组中的成员就是集群地址里面的数据库节点。多机并行（MPP）会在指定MPP集群分组内的数据库节点做任务调度。MPP集群分组的取值就是集群地址的ID，例如 <code>pe-2zejv3fb4063*****</code> 。

参数名	级别	描述
parallel_workers_policy	Session	多机并行Workers调度策略。 LOCAL：单机并行。仅在单个节点内进行任务调度。 AUTO：开启多机并行。根据集群分组内节点的实时负载情况做任务调度，当本节点的计算资源不足时，会尝试选择其它有空闲资源的节点进行任务调度。 ROUND_ROBIN：开启多机并行。将并行Workers均分到集群分组内的所有节点进行任务调度，适用于需要设置更高并行度的大查询。

 **说明** 上述参数仅支持通过Session变量和Hint语法来设置。

如何设置多机并行（MPP）

您可以通过控制台、Session变量、Hint语法三种方式来设置多机并行（MPP）。

● 通过控制台来设置多机并行（MPP）

在控制台的集群地址编辑页面，可开启或关闭并行查询，设置并行度，并选择并行引擎（MPP）。具体操作请参见[配置数据库代理](#)中的并行查询说明。

 **说明** 控制台开启多机并行（MPP）后，parallel_workers_policy默认为**AUTO**，关闭多机并行（MPP）后parallel_workers_policy为**LOCAL**。

控制台中设置的并行度对当前MPP集群分组内的数据库节点生效，并行度的大小建议设置为单机CPU核数的1/4。开启多机并行（MPP）后，并行度可根据实际查询业务量适当调大。对于复杂分析型查询场景，您也可以单独通过Hint指定并行度。

● 通过Session变量来设置多机并行（MPP）

客户端连接到集群地址后，在Session中可以通过设置parallel_workers_policy来控制多机并行（MPP）。

```
/* 使用AUTO模式开启多机并行 */
set parallel_workers_policy=AUTO;
SELECT ... FROM ...;
/* 使用ROUND_ROBIN模式开启多机并行 */
set parallel_workers_policy=ROUND_ROBIN;
SELECT ... FROM ...;
/* 关闭多机并行 */
set parallel_workers_policy=LOCAL;
SELECT ... FROM ...;
/* 设置并行度 */
set max_parallel_degree=n;
```

● 通过Hint语法来设置多机并行（MPP）

客户端连接到集群地址后，在SQL语句中增加Hint语法来控制多机并行。

```
/* 使用AUTO模式开启多机并行 */
SELECT /*+ SET_VAR(parallel_workers_policy=AUTO) */ ... FROM ...;
/* 使用ROUND_ROBIN模式开启多机并行，并设置并行度大于单机CPU核数 */
SELECT /*+ SET_VAR(parallel_workers_policy=LOCAL) PARALLEL(64) */ ... FROM ...;
/* 关闭多机并行 */
SELECT /*+ SET_VAR(parallel_workers_policy=LOCAL) */ ... FROM ...;
```

通过Hint语法设置并行度的方法，详情请参见[通过Hint指定并行表](#)。

如何选择MPP集群分组

MPP集群分组中的数据库节点配置非常灵活，一个节点允许被加入到多个MPP集群分组中。系统默认以客户端连接的集群地址为MPP集群分组进行任务调度，也可以通过Session变量或Hint语法设置polar_cluster_group参数来指定另一个MPP集群分组进行任务调度。

```
/* 在Session建立连接后指定集群分组，影响当前Session的所有后续查询 */
set polar_cluster_group=pe-xxxxxx;
/* 在SQL中通过Hint指定另一个集群分组，只影响当前SQL */
SELECT /*+ SET_VAR(polar_cluster_group=pe-xxxxxx) */ ... FROM ...;
```

 **说明** 如果通过Session变量或Hint语法设置的polar_cluster_group不存在，就会回退为单机并行，而不会回退到系统默认的MPP集群分组。

19.PolarDB for AI

19.1. 概述

PolarDB for AI功能通过一系列MLOps和内置的模型解决了数据、特征和模型的割裂状态，实现了基于数据库的数据智能的一站式服务。本文介绍了PolarDB for AI功能的相关特性。

发布时间

2022年7月8日。

 **说明** PolarDB for AI当前处于灰度发布阶段。如有需求，请[点击此处](#)申请试用。

背景信息

随着数据的累积，数据驱动的智能应用（例如：搜索、推荐和问答）已经越来越广泛地被用于各种场景中。在过去几年，数据驱动的智能应用已经逐渐从基于简单的数据分析和统计转型为基于机器学习和深度学习的特征和模型应用。转型的速度远远超出预期，部分原因是机器学习和人工智能方面的技术突破。深度神经网络在图像分析和自然语言处理等学科中取得了前所未有的进步。强化学习成为补充传统监督学习的强大范式。

然而，在目前的数据驱动的智能应用中，数据、特征和模型仍处于割裂状态。首先，数据工程师通过手工编写流程进行数据清洗和数据集成；然后，算法工程师通过自定义的特征工程流程、模型训练脚本以及定时任务脚本进行周期性的生产特征和模型；最后，开发工程师负责模型的上线、稳定性保证和监控运维。这就导致了不同系统之间会进行数据迁移，同一份数据可能在不同源之间冗余，进而出现数据不一致的问题，以及特征难管理、模型难升级等困境。除此之外，数据工程师、算法工程师和开发工程师的人力成本也是当前数据驱动的智能决策应用难以大规模落地的一个阻碍。

事实上，鉴于数据库研究在数据发现、数据管理、版本控制、数据清理和数据集成方面积累的专业知识，PolarDB for AI可以为数据驱动的智能应用提供一站式的数据（包括：数据、特征和模型）服务来解决这种割裂状态，大大减少数据驱动的智能决策开发过程中的人力成本，从而走出当前的困境。

简介

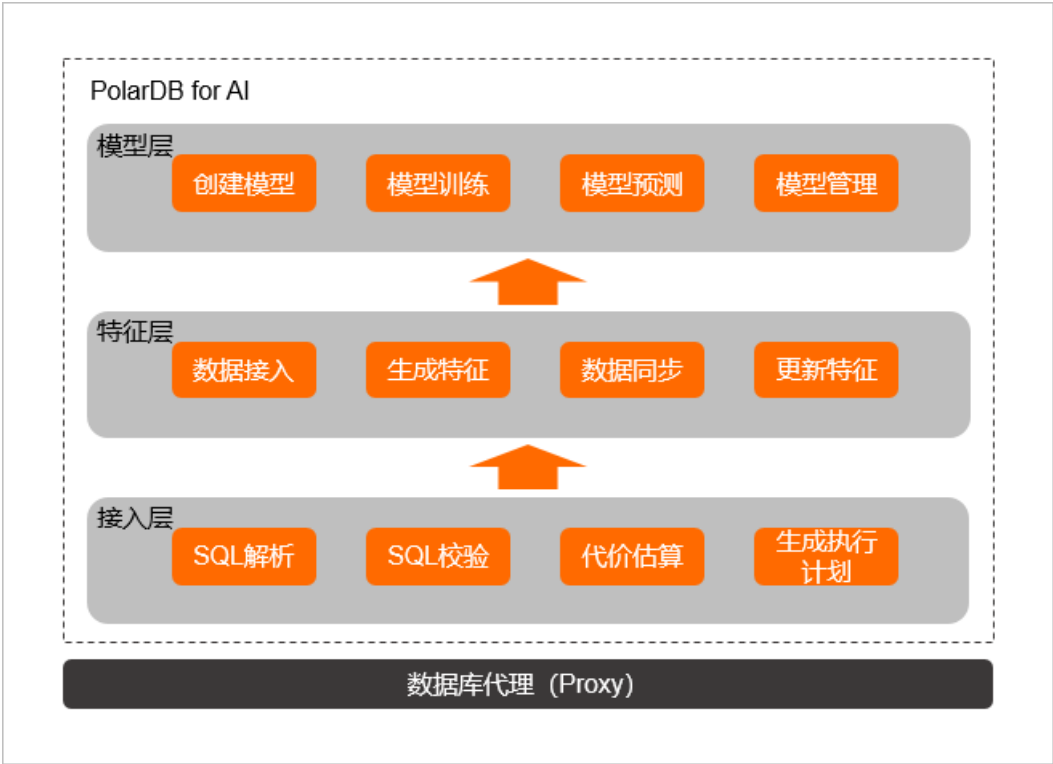
PolarDB for AI是基于PolarDB MySQL的一个数据库内的分布式机器学习组件。其基于云原生的体系架构，通过SQL语句的方式提供了支持机器学习的一系列MLOps，包括：创建模型、训练模型、查看模型状态、查看模型列表和模型预测等能力；同时PolarDB for AI提供了一系列内置的机器学习和人工智能算法，包括：分类算法、回归算法和聚类算法等。基于MLOps和内置的模型，PolarDB for AI为数据驱动的智能应用提供了高效、可靠、方便的数据智能能力，打破了数据库和业务应用之间的系统墙，提供了基于数据库数据智能的一站式服务。

费用

当前PolarDB for AI功能可免费试用。

技术架构

PolarDB for AI通过数据库代理进行SQL路由，含有 `/*polar4ai*/` 的SQL语句会被分发到具有AI能力的集群上进行计算，计算的返回结果和普通的SQL返回结果遵循同样的协议。所以，您可以[连接数据库集群](#)并执行带有 `/*polar4ai*/` 的SQL语句来使用该功能。其架构图如下所示：



说明 PolarDB数据库代理版本（Proxy）需为2.7.1及以上，如何查看和升级当前数据库代理版本，请参见[版本管理](#)。

除普通实例具有的存储功能和计算功能之外，PolarDB for AI还具有三层结构，分别是接入层、特征层及模型层。其中：

- 接入层：负责处理SQL语句，包括SQL解析、SQL校验、代价估算和生成执行计划等，并优化为可以执行的SQL语句节点树。
- 特征层：负责处理从数据转换为特征的工作，包括数据接入、特征生成、数据同步和特征更新。
- 模型层：负责处理模型相关的MLOps能力，包括模型创建、模型训练、模型预测和模型管理。

适用场景

如下图所示，PolarDB for AI将传统的 DB+AI 的数据智能应用转变为一站式数据智能应用，从数据库服务于传统的数据工程师和算法工程师转变为 DB+AI 服务于业务工程师，直接与业务系统对接。



PolarDB for AI可用于如下行业场景：

● ID-Mapping

ID-Mapping通常适用于平台性的客户，如游戏和电商等。以游戏平台为例，某些游戏平台中的同一个用户可能会有多个游戏账号，这些账号散落在各种游戏或平台本身的数据库中，并没有对账号进行融合。基础数据建设不完善会带来各种问题，例如做精准营销时不能精准定位同一个用户，导致流量在跨渠道使用时效率较低。采用机器学习模型则会将账号统一，不但提高了基础数据质量，而且还为上游业务提供了优质的数据。

● 问答机器人

问答机器人基于数据库中的内容。根据用户的业务场景，结合AI能力（对话控制、机器学习、自然语言理解等），打造适合企业的对话服务。问答机器人可以实现 7×24 小时在线服务，能帮助企业接待更多客户、提升客户满意度、提高工作效率和降低运营成本。是企业进行在线咨询、在线营销和在线服务的好帮手。

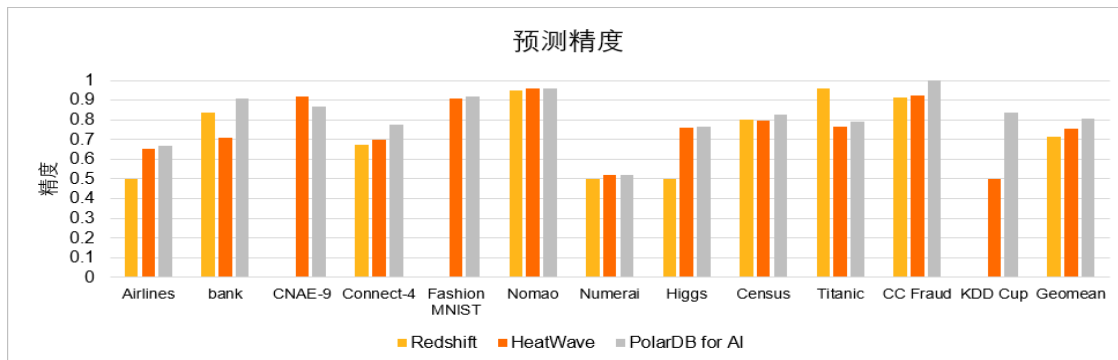
● 搜索推荐

在传统的数据库中，用户的搜索能力通常基于数据库固有的全文检索能力，不支持自然语言类的检索需求（如语义检索、同义词匹配等）。采用PolarDB for AI中成熟的搜索解决方案，可以大幅度提升搜索的精确性。

基于PolarDB for AI的智能推荐算法和知识图谱技术，再结合阿里巴巴电商策略，为企业提供贯穿推荐能力的一站式服务，助力企业快速过渡冷启动过程。面向不同的业务场景定制个性化解决方案，持续提升核心业务能力，以实现业务营收增长。

优势

- 功能优势：PolarDB for AI支持PolarDB内置的模型训练和模型预测等MLOps，无缝兼容MySQL语句，提供达摩院加持的行业智能算法服务，避免数据在两个甚至多个系统间转换和迁移，从而降低数据智能应用的开发成本，加速数据价值业务变现。
- 性能优势：在80%以上的公开数据集之下，PolarDB for AI的性能优势尤为明显，其中：PolarDB for AI的预测精度高于Oracle HeatWave和AWS RedShift。



使用说明

开启PolarDB for AI功能后，您可以创建AI模型、查看模型当前状态、查看模型列表、进行模型预测以及删除模型等操作，具体的操作步骤请参见[使用说明](#)。

19.2. 开启PolarDB for AI功能

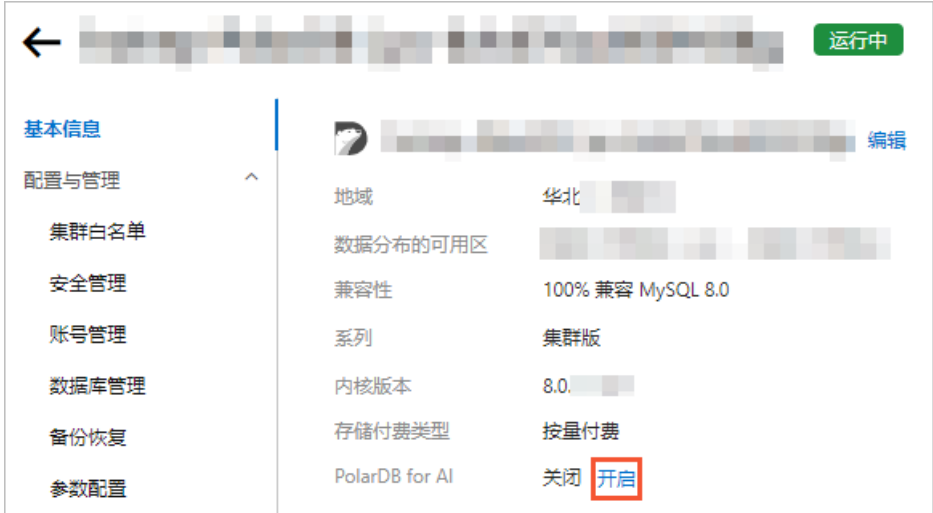
本文介绍了开启或关闭PolarDB for AI功能，以及查看该功能当前状态的操作步骤。

前提条件

- 已创建普通账号。
- 已通过[申请链接](#)申请试用PolarDB for AI功能，且申请已审批通过。

开启PolarDB for AI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. 在集群列表页面，单击目标集群。
4. 在基本信息页面，单击PolarDB for AI右侧的开启。



5. 在您确认开启PolarDB for AI吗？对话框中，选择需要开启该功能的账号并输入账号密码。

说明

- 该账号必须为普通账号。
- 若无账号，则需要创建新账号，请参见[创建普通账号](#)。

6. 勾选我已知晓以上内容。单击确认。

关闭PolarDB for AI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. 在集群列表页面，单击目标集群。
4. 在基本信息页面，单击PolarDB for AI右侧的关闭。



5. 在弹出的对话框中，单击确认。

查看PolarDB for AI功能状态

- 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，单击目标集群。
- 4. 在基本信息页面，查看PolarDB for AI的状态。



19.3. 使用说明

本文介绍了PolarDB for AI功能使用说明及其对应的语法。

前提条件

已在控制台开启PolarDB for AI功能。

注意事项

执行SQL语句时，需要在SQL语句前添加 `/*polar4ai*/` 。

创建模型

创建一个机器学习模型。机器学习模型的训练过程是异步的，需要通过查询模型状态来判断该模型是否训练完成。

语法

```
CREATE MODEL model_name WITH ( model_class = '', x_cols = '', y_cols='',model_parameter=())
AS (SELECT select_expr [, select_expr] ... FROM table_reference)
```

参数说明

参数	说明
model_name	模型名称。

参数	说明
model_class	模型的类型。取值如下： - lightgbm：详情请参见LightGBM算法。 - deepfm：详情请参见DeepFM算法。 - kmeans：详情请参见K均值聚类算法（K-Means）。 - randomforestreg：详情请参见随机森林回归算法（Random Forest Regression）。 - gbrt：详情请参见梯度提升回归树算法（GBRT）。 - linearreg：详情请参见线性回归算法（Linear Regression）。 - svr：详情请参见支持向量回归算法（SVR）。
x_cols	模型训练输入列。 ❓ 说明 多个列名之间使用英文逗号（,）分隔。
y_cols	模型训练输出列。
model_parameter	模型训练参数。 - 当 model_class 值为lightgbm时，model_parameter 取值请参见参数说明。 - 当 model_class 值为deepfm时，model_parameter 取值请参见参数说明。 - 当 model_class 值为kmeans时，model_parameter 取值请参见参数说明。 - 当 model_class 值为randomforestreg时，model_parameter 取值请参见参数说明。 - 当 model_class 值为gbrt时，model_parameter 取值请参见参数说明。 - 当 model_class 值为linearreg时，model_parameter 取值请参见参数说明。 - 当 model_class 值为svr时，model_parameter 取值请参见参数说明。
select_expr	列名。
table_reference	表名。

示例
创建 siku_kmeans 模型。

```
/*polar4ai*/CREATE MODEL siku_kmeans WITH ( model_class = 'kmeans', x_cols = 'dx1,dx2', y_cols='',model_parameter=(n_clusters=2)) AS (select * from db4ai.testdata1)
```

执行结果显示类似如下内容，表示 siku_kmeans 模型创建成功：

```
Query OK, 0 rows affected (0.79 sec)
```

查看模型列表

查看当前集群下的所有模型。

语法

```
SHOW MODELS
```

示例

```
/*polar4ai*/SHOW MODELS
```

执行结果如下：

```
+-----+-----+-----+
| model_name | model_class | model_status |
+-----+-----+-----+
| airline_gbm | lightgbm    | saved        |
| siku_kmeans | kmeans      | saved        |
| airline_deepfm | deepfm     | saved        |
| test_kmeans1 | kmeans      | saved        |
| test_kmeans  | kmeans      | saved        |
+-----+-----+-----+
5 rows in set (0.23 sec)
```

查看模型状态

查看模型的当前状态。创建模型之后使用该语法来查看模型是否训练完成。模型状态主要有以下几种：

- training：模型训练中。
- loading_data：模型在加载数据中。
- trained：模型已经完成训练。
- saved：模型已经被保存。
- serving：模型正在服务中。
- deleted：模型已经被删除。

语法

```
SHOW MODEL model_name
```

参数说明

参数	说明
model_name	模型名称。

示例

查看 `siku_kmeans` 模型的当前状态。

```
/*polar4ai*/SHOW MODEL siku_kmeans
```

执行结果如下：

```
+-----+
| modelStatus |
+-----+
| saved      |
+-----+
1 row in set (0.24 sec)
```

模型预测

使用已经训练完成的模型进行模型预测。

语法

```
SELECT select_expr [, select_expr] ... FROM PREDICT (MODEL model_name, SELECT select_expr_f
or_prediction [, select_expr_for_prediction] ... FROM table_reference LIMIT row_count) WITH
(x_cols = '', y_cols='')
```

参数说明

参数	说明
select_expr	列名。
model_name	模型名称。
select_expr_for_prediction	列名。
table_reference	表名。
row_count	预测的记录数量。
x_cols	模型训练输入列。多个列名之间使用英文逗号 (,) 分隔。
y_cols	模型训练输出列。

示例

使用已经训练完成的 `siku_kmeans` 模型进行模型预测。

```
/*polar4ai*/SELECT dx1 FROM PREDICT(MODEL test_kmeans, select * from db4ai.testdata1 limit
10) WITH (x_cols = 'dx1,dx2', y_cols='');
```

执行结果如下：

```
+-----+-----+
| dx1 | predicted_results |
+-----+-----+
| 1    | 0                  |
| 2    | 3                  |
| 20   | 1                  |
| 21   | 2                  |
| 22   | 1                  |
| 2    | 0                  |
+-----+-----+
6 rows in set (0.90 sec)
```

删除模型

当模型不再使用时，使用该语法删除已经创建的模型。

语法

```
DROP MODEL model_name
```

参数说明

参数	说明
model_name	模型名称。

示例

删除 `siku_kmeans` 模型。

```
/*polar4ai*/DROP MODEL siku_kmeans
```

执行结果显示类似如下内容，表示已成功删除 `siku_kmeans` 模型：

```
Query OK, 0 rows affected (0.57 sec)
```

19.4. 算法介绍

19.4.1. LightGBM算法

本文介绍了LightGBM算法相关内容。

简介

LightGBM (Light Gradient Boosting Machine) 是一个基于决策树算法的分布式梯度提升框架。设计初衷是提供一个快速、高效、低内存、高准确度、支持并行和大规模数据处理的工具。LightGBM可以减小数据对内存的使用、减小通信代价以及提升多机并行时的效率，在数据计算上实现线性加速。

使用场景

LightGBM是一个算法框架，包括GBDT模型、随机森林和逻辑回归等模型。通常应用于二分类、多分类和排序等场景。

例如：在个性化商品推荐场景中，通常需要做点击预估模型。使用用户过往的行为（点击、曝光未点击、购买等）作为训练数据，来预测用户点击或购买的概率。根据用户行为和用户属性提取一些特征，包括：

- 类别特征（Categorical Feature）：字符串类型，如性别（男/女）。
- 物品类型：服饰、玩具和电子等。
- 数值特征（Numrical Feature）：整型或浮点型，如用户活跃度或商品价格等。

参数说明

下表中的参数为创建模型 `CREATE MODEL` 语法中 `model_parameter` 参数的取值，您可以根据当前需求选择对应的参数。

参数	说明
boosting_type	弱学习器的类型。取值如下： • gbdt（默认）：使用基于树的模型进行计算。 • gblinear：使用线性模型进行计算。 • rf：使用随机森林模型进行计算。 • dart：使用dropout技术删除部分树，防止过拟合。 • goss：使用单边梯度抽象算法进行计算。速度块，但是可能欠拟合。  说明 设置该参数值时，需要使用单引号将参数值包裹起来。例如：boosting_type='gbdt'
objective	学习任务及相应的学习目标。取值如下： • binary（默认）：二分类。 • regression：使用L2正则项回归模型。 • regression_l1：使用L1正则项回归模型。 • multiclass：多分类。
num_leaves	叶子的个数。默认为128个。
max_depth	树的最大深度。默认值为7。  说明 如果设置为-1，则表示不对树的深度做限制。但为防止过拟合，建议合理设置该值。

示例

创建模型与模型离线训练

```
/*polar4ai*/
CREATE MODEL airline_gbm WITH
(model_class = 'lightgbm',
x_cols = 'Airline,Flight,AirportFrom,AirportTo,DayOfWeek,Time,Length',
y_cols='Delay',model_parameter=(boosting_type='gbdt'))
AS (select * from db4ai.airlines)
```

模型预测

```
/*polar4ai*/
select Airline FROM PREDICT(MODEL airline_gbm,
select * from db4ai.airlines limit 20) WITH
(x_cols = 'Airline,Flight,AirportFrom,AirportTo,DayOfWeek,Time,Length',
y_cols='Delay')
```

19.4.2. DeepFM算法

本文介绍了DeepFM算法相关内容。

简介

DeepFM将深度学习模型（DNN）和因子分解机（FM）模型结合，同时支持学习低阶显式特征组合和高阶隐式特征组合，不需要人工做特征工程，常用于推荐系统或广告系统。

- 输入通常有以下两类特征：
 - 类别特征（Categorical Feature）：字符串类型。如性别（男/女）或物品类型（服饰、玩具、电子等）。
 - 数值特征（Numrical Feature）：整型或浮点型。如用户活跃度或商品价格等。
- 输出通常是0~1之间的浮点数，表示取值为1的概率。可以用于排序，也可以用于二分类。

使用场景

DeepFM通常用于分类或排序场景。在人工构建的特征但不能直接反映结果的场景上效果更好。在各种各样的推荐场景中，低阶特征组合或者高阶的特征组合都会对用户最终的行为产生影响。但人为可能无法明确特征组合，DeepFM可以自动学习这些组合。

例如：在个性化商品推荐场景中，通常需要做点击预估模型。使用用户过往的行为（点击、曝光未点击、购买等）作为训练数据，来预测用户点击或购买的概率。如果用户过往行为较多，且不能直接反映出在未来某时刻的点击或购买情况。DeepFM会对用户行为进行组合，同时将一些稀疏特征转化为高维的稠密特征。

参数说明

下表中的参数为创建模型 `CREATE MODEL` 语法中 `model_parameter` 参数的取值，您可以根据当前需求选择对应的参数。

参数	说明
metrics	评估模型的指标参数。取值如下： ● accuracy（默认）：精确度，用于评估分类模型。 ● binary_crossentropy：交叉熵，用于评估二分类问题。 ● mse：均方误差，用于评估回归模型。
loss	学习任务及对应的学习目标。取值如下： ● binary_crossentropy（默认）：交叉熵，用于二分类问题。 ● mean_squared_error：均方误差，用于回归模型。

参数	说明
optimizer	优化器。取值如下： - adam（默认）：吸收了AdaGrad（自适应学习率的梯度下降算法）和动量梯度下降算法的优点，既能适应稀疏梯度（即自然语言和计算机视觉问题），又能缓解梯度震荡的问题。 - sgd：随机梯度下降。 - rmsprop：对AdaGrad算法进行改进，引入了权重参数，其对应的步长分量累计修改为加权和。
validation_split	交叉验证数据比例。默认值为0.2。
epochs	迭代次数。默认6次。
batch_size	batch长度，batch越短越容易过拟合。默认64。
task	任务类型。取值如下： - binary（默认）：分类模型。 - regression：回归模型。

示例

创建模型与离线训练

```
/*polar4ai*/
CREATE MODEL airline_deepfm WITH
(model_class = 'deepfm',
x_cols = 'Airline,Flight,AirportFrom,AirportTo,DayOfWeek,Time,Length',
y_cols='Delay',model_parameter=(epochs=6))
AS (select * from db4ai.airlines)
```

模型预测

```
/*polar4ai*/
select Airline FROM PREDICT(MODEL airline_deepfm,
select * from db4ai.airlines limit 20) WITH
(x_cols = 'Airline,Flight,AirportFrom,AirportTo,DayOfWeek,Time,Length',
y_cols='Delay')
```

19.4.3. K均值聚类算法（K-Means）

本文介绍了K均值聚类算法（K-Means Clustering Algorithm，以下简称K-Means）相关内容。

简介

K-Means算法是一种迭代求解的聚类分析算法。该算法原理为：先将数据分为K组，随机选取K个对象作为初始的聚类中心，然后计算每个对象与各个种子聚类中心之间的距离，将每一个对象分配给距离它最近的聚类中心，聚类中心以及分配给它们的对象就代表一个聚类。即K-Means算法将输入表的某些列作为特征，根据用户指定的相似度计算方式，将原始数据聚成若干类。

使用场景

K-Means算法被广泛应用于需要对数据记录进行聚类的场景。以下列出了六类使用场景：

- **文档分类**
根据标签、主题和文档内容将文档分为多个不同的类别。
首先，需要对文档进行初始化处理，将每个文档都使用矢量来表示，并采用术语频率识别常用术语的方式来进行文档分类，然后对文档向量进行聚类来识别文档组中的相似性。
- **客户分类**
K-Means算法能帮助营销人员改善他们的客户群（在其目标区域内工作）。并根据客户的购买历史、兴趣或活动监控来对客户类别做进一步分析。
例如：使用K-Means算法可以将电信预付费客户的付费方式分为充值模式、发送短信和浏览网站几个类别。对客户进行分类有助于公司针对特定客户群制定特定的广告。
- **欺诈检测**
K-Means算法在欺诈检测中也扮演着一个至关重要的角色，被广泛应用于汽车、医疗保险和保险欺诈检测领域。利用以往欺诈性索赔的历史数据，根据它和欺诈性模式聚类的相似性来识别新的欺诈。
- **警报的自动化聚类**
大型企业IT基础架构技术组件（如网络、存储或数据库）会生成大量的警报消息，由于警报消息可以指向具体的操作，因此必须对警报信息进行手动筛选，确保后续操作的优先级。对数据进行K-Means聚类可以对警报类别和平均修复时间做深入分析，有助于对未来故障进行预测。
- **通话记录分析**
通话记录（CDR）是电信公司对用户的通话、短信和网络活动信息的记录。将通话详细记录与客户个人资料结合在一起，能够帮助电信公司对客户需求做更多的预测。
- **识别犯罪地点**
K-Means算法可以对城市中特定地区的相关犯罪数据进行分析。分析内容包括犯罪类别、犯罪地点以及两者之间的关联等，可以对城市或区域中容易犯罪的地区做高质量的勘察。

参数说明

下表中的参数为创建模型 `CREATE MODEL` 语法中 `model_parameter` 参数的取值，您可以根据当前需求选择对应的参数。

参数	说明
<code>n_clusters</code>	类数量。默认值为4。

示例

创建模型与模型离线训练：

```
/*polar4ai*/
CREATE MODEL test_kmeans WITH
(model_class = 'kmeans', x_cols = 'dx1,dx2',
 y_cols='',model_parameter=(n_clusters=2))
AS (select * from db4ai.testdata1)
```

模型预测：

```
/*polar4ai*/
select dx1,dx2 FROM
PREDICT(MODEL test_kmeans,
select * from db4ai.testdata1 limit 10)
WITH (x_cols = 'dx1,dx2',
 y_cols='')
```

 说明 `x_cols` 中的列数据类型必须是浮点型或整型。

19.4.4. 随机森林回归算法（Random Forest Regression）

本文介绍了随机森林回归算法（Random Forest Regression）相关内容。

简介

随机森林回归算法（Random Forest Regression）是随机森林（Random Forest）的重要应用分支。随机森林回归模型通过随机抽取样本和特征，建立多棵相互不关联的决策树，通过并行的方式获得预测结果。每棵决策树都能通过抽取的样本和特征得出一个预测结果，通过综合所有树的结果取平均值，得到整个森林的回归预测结果。

使用场景

随机森林回归算法在对数据维度要求相对较低（几十维），同时对准确性要求较高的场景下使用。

例如，预测Twitter的某一话题的热度，可以使用SVR回归模型来处理。模型的输入可以是话题的特征，比如话题在某时刻产生的讨论组数量、参与该话题讨论的人数和话题关注度等。模型的输出是平均每小时的活跃讨论组的数量，是一个正的浮点数，值越大热度越高。

参数说明

下表中的参数为创建模型 `CREATE MODEL` 语法中 `model_parameter` 参数的取值，您可以根据当前需求选择对应的参数。

参数	说明
<code>n_estimators</code>	迭代次数。次数越多拟合越好，通常为正整数，默认值为100。
<code>objective</code>	学习任务及相应的学习目标。取值如下： <code>mse</code>（默认）：使用均方误差。 <code>mae</code>：使用平均绝对误差。
<code>max_features</code>	决定划分时考虑的最大特征数量。取值如下： <code>sqrt</code>（默认）：表示最大特征数量为 <code>sqrt(n_features)</code>。 - 整数：表示最大特征数量为 <code>max_features</code>，取值必须在 <code>0~n_features</code> 之间，包括 <code>n_features</code>，<code>n_features</code> 为建模时使用的特征字段的数量。 - 浮点数：表示最大特征数量为 <code>max_features*n_features</code>。 <code>log2</code>：表示最大特征数量为 <code>log2(n_features)</code>。
<code>random_state</code>	随机状态。通常为正整数，默认值为1。
<code>n_jobs</code>	并行线程数。数量越多训练速度越快，通常为正整数，默认值为4。

参数	说明
max_depth	每颗树的最大深度。通常为正整数，默认值为None。 ? 说明 当设置为None时，表示对树的深度没有限制。

示例

创建模型与模型离线训练

```
/*polar4ai*/
CREATE MODEL randomforestreg1 WITH
( model_class = 'randomforestreg', x_cols = 'dx1,dx2', y_cols='y',
  model_parameter=(objective='mse')) AS (select * from db4ai.testdata1)
```

模型预测

```
/*polar4ai*/
select dx1,dx2 FROM
PREDICT(MODEL randomforestreg1, select * from db4ai.testdata1 limit 10)
WITH (x_cols = 'dx1,dx2', y_cols='')
```

? 说明 其中，`x_cols` 和 `y_cols` 中的列数据类型必须是浮点型或整型。

19.4.5. 梯度提升回归树算法（GBRT）

本文介绍了梯度提升回归树算法（Gradient Boosting Regression Tree，下文简称GBRT）相关内容。

简介

GBRT算法是集成学习Boosting家族的成员，使用了前向分布算法，但是弱学习器限定了只能使用CART回归树模型。前向分布算法的思想是基于当前模型和拟合函数来选择合适的决策树函数，从而最小化损失函数。

GBRT主要有以下两部分组成：

- 回归树（Regression Tree（RT））：回归树是决策树类别之一，用来预测实际值。GBRT算法是一种迭代的回归树算法，由多棵回归树组成，所有回归树的结论累加起来为最终结果。
- 梯度提升（Gradient Boosting（GB））：梯度提升通过迭代多棵树来共同决策最终结果。每一棵树都是学习之前的树的结论和残差。

使用场景

GBRT是一个回归模型，主要用于拟合数值。

GBRT算法可以应用于流行病学。例如，有关于人的死亡率和发病率早期证据来自于回归分析的观察性研究。假设有一个回归模型，在回归模型中死亡率（或发病率）是需要拟合的变量 `y_cols`，则社会经济地位、教育或收入等可以做为其因变量。

参数说明

下表中的参数为创建模型 `CREATE MODEL` 语法中 `model_parameter` 参数的取值，您可以根据当前需求选择对应的参数。

参数	说明
n_estimators	迭代次数。次数越多拟合越好，通常为正整数，默认值为100。
objective	学习任务及相应的学习目标。取值如下： - ls（默认）：最小二乘法。 - lad：最小一乘法。 - huber：最小二乘法和最小一乘法结合使用。
max_depth	树的最大深度。默认值为7。 ? 说明 如果设置为-1，则表示不对树的深度做限制。但为防止过拟合，建议合理设置该值。
random_state	随机状态。通常为正整数，默认值为1。

示例

模型创建与模型离线训练

```
/*polar4ai*/
CREATE MODEL gbrt1 WITH
( model_class = 'gbrt', x_cols = 'dx1,dx2', y_cols='y',
  model_parameter=(objective='ls')) AS (select * from db4ai.testdata1)
```

模型预测

```
/*polar4ai*/select dx1,dx2 FROM
PREDICT(MODEL gbrt1, select * from db4ai.testdata1 limit 10)
WITH (x_cols = 'dx1,dx2', y_cols='')
```

? 说明

`x_cols` 和 `y_cols` 中的列数据类型必须是浮点型或整型。

19.4.6. 线性回归算法（Linear Regression）

本文介绍了线性回归算法（Linear Regression，以下简称LR）相关内容。

简介

LR是利用线性回归方程的最小平方函数对一个或多个自变量和因变量之间关系进行建模的一种回归分析。

使用场景

LR是一个回归模型，主要用于拟合数值。模型简单，可解释性强。

LR适用于拟合趋势线，一条趋势线代表着时间序列数据的长期走势，它表示一组特定数据（如股票价格、GMV和销量）是否在一段时期内增长或下降。虽然可以根据肉眼观察数据点在坐标系的位置来画出趋势线，但是更恰当的方法是利用LR计算出趋势线的位置和倾斜率。

参数说明

下表中的参数为创建模型 `CREATE MODEL` 语法中 `model_parameter` 参数的取值，您可以根据当前需求选择对应的参数。

参数	说明
epoch	迭代次数。通常为正整数，默认值为-1。 ? 说明 迭代次数设置为-1时，表示持续迭代直至收敛。
normalize	是否需要正则化。取值如下： - False（默认）：在模型训练前不将数据进行正则化处理。 - True：在模型训练前将数据进行正则化处理。

示例
创建模型与模型离线训练

```
/*polar4ai*/
CREATE MODEL linearreg1 WITH
( model_class = 'linearreg', x_cols = 'dx1,dx2', y_cols='y',
  model_parameter=(epoch=3)) AS (select * from db4ai.testdata1)
```

模型预测

```
/*polar4ai*/select dx1,dx2 FROM
PREDICT(MODEL linearreg1, select * from db4ai.testdata1 limit 10)
WITH (x_cols = 'dx1,dx2', y_cols='')
```

? 说明 `x_cols` 和 `y_cols` 中的列数据类型必须是浮点型或整型。

19.4.7. 支持向量回归算法（SVR）

本文介绍了支持向量回归算法（Support Vector Regression，以下简称SVR）回归算法相关内容。

简介

SVR是支持向量机（SVM）的重要应用分支。通过SVR算法，可以找到一个回归平面并使得一个集合中的所有数据距离该平面的距离最短。

使用场景

SVR是一个回归模型，主要是用于拟合数值，一般应用于特征较为稀疏且特征数较少的场景。

例如，可以使用SVR回归模型来预测某个城市的温度。输入特征有很多，例如这个城市某个时期的平均温度、绿化程度、湖泊数量以及日期等。训练数据可以是一段时间内的城市温度。

参数说明

下表中的参数为创建模型 `CREATE MODEL` 语法中 `model_parameter` 参数的取值，您可以根据当前需求选择对应的参数。

参数	说明
kernel	核函数，用于将低维数据映射到高维空间。取值如下： - rbf（默认）：高斯径向基核函数。可以将一个样本映射到一个更高维的空间内。 - linear：线性核。主要用于线性可分的情况。特征空间到输入空间的维度一样，其参数少、速度快。 - poly：多项式核函数。可以实现将低维的输入空间映射到高维的特征空间，其参数较多。 - sigmoid：使用该函数作为核函数时，效果类似多层神经网络。
c	松弛系数的惩罚项系数。取值为大于0的浮点数，可不填。默认值为1。 ❓ 说明 如果数据质量较差，可以适当降低惩罚项 <code>c</code> 的值。
epsilon	SVR损失函数的阈值。当预测值与实际值的差值等于该阈值时，需要计算该样本的损失。默认值为0.1。
max_iter	最大迭代次数。取值为正整数或-1，默认值为-1。 ❓ 说明 如果设置为-1，则表示不限制迭代次数，直到收敛到 <code>epsilon</code> 内为止。

示例

创建模型与模型离线训练

```
/*polar4ai*/
CREATE MODEL svr1 WITH
( model_class = 'svr', x_cols = 'dx1,dx2', y_cols='y',
  model_parameter=(kernel='rbf')) AS (select * from db4ai.testdata1)
```

模型预测

```
/*polar4ai*/select dx1,dx2 FROM
PREDICT(MODEL svr1, select * from db4ai.testdata1 limit 10)
WITH (x_cols = 'dx1,dx2', y_cols='')
```

❓ 说明 其中，`x_cols` 和 `y_cols` 中的列数据类型必须是浮点型或整型。

20. 集群回收站

20.1. 费用说明

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。

本文介绍集群回收站的费用说明。

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（大陆）	0.00021
中国香港及海外	0.000294

20.2. 恢复已释放的集群

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（即已释放的集群）恢复至新集群，或删除已释放集群的备份集。本文以PolarDB MySQL引擎集群为例介绍如何恢复回收站中的集群。

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150 MB/s左右。更多关于备份的介绍请参见[数据备份](#)。

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/名称		地域	主节点配置	兼容性	创建时间	删除时间	状态
+ 		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
+ 		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

5. 选择商品类型为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。
- **按量付费**：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。

 **说明** PolarDB MySQL引擎已推出计算包。推荐购买按量付费的集群，因为按量付费的集群支持配合计算包使用，比包年包月付费方式更划算、更灵活。详情请参见[计费方式1：按量付费+计算包（推荐）](#)。

6. 设置以下参数。

参数	说明
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。
创建方式	选择从回收站恢复，即从回收站恢复已经删除的数据库集群。
原版本	选择已删除集群的版本。
已删除集群	选择已删除的集群ID。
历史备份	选择需要恢复的备份。  说明 历史备份显示的备份时间为UTC时间，数据备份列表页签中显示的备份时间已转换成您当前本地时区的时间，请确保所选择的历史备份无误。 例如，您当前位于东八区（北京时间），控制台的备份列表里显示某个备份的创建时间是 2021年05月28日 11:19:30 （北京时间），那么请在此处的历史备份列表里选择 2021-05-28T03:19:30Z （UTC时间）。
主可用区	选择集群的主可用区。  说明 在有两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	固定为专有网络，无需选择。  说明 如需使用经典网络请先任选一个专有网络（VPC），待创建完成后再进行配置，详情请参见私网、公网和经典网络地址。
VPC网络	选择集群所在的 VPC网络 和 VPC交换机 ，建议与原集群一致。
VPC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。
兼容性	PolarDB集群的数据库引擎版本，默认与已删除集群的版本保持一致，不可变更。
系列	固定为 集群版（2-16个节点）【推荐】 ，无需选择。

参数	说明
小版本号	您可以选择小版本号为8.0.1（与MySQL 8.0.13完全兼容）【推荐】或8.0.2（与MySQL 8.0.18完全兼容）。关于各小版本的具体信息，请参见内核发布记录。  注意 - 仅当PolarDB集群的兼容性选择为MySQL 8.0时，支持该配置。 8.0.2版本目前正在公测中，可能存在若干稳定性风险，不承诺服务等级协议SLA。
子系列	PolarDB集群版支持通用规格和独享规格两种子系列，其中： 独享规格：每个集群会独占所分配到的计算资源（如CPU），而不会与同一服务器上的其他集群共享资源，性能更加稳定可靠。 通用规格：同一服务器上的不同集群，会互相充分利用彼此空闲的计算资源（如CPU），通过复用计算资源享受规模红利，性价比更高。 关于两种类型的详细对比，请参见如何选择通用规格和独享规格。
节点规格	选择节点规格，不同规格有不同的最大存储容量和性能。更多详情，请参见计算节点规格。  说明 为了保障恢复后的集群运行正常，建议选择高于原集群的节点规格。
节点个数	- 若原集群系列为集群版（2-16个节点）【推荐】，节点个数默认为2，无需选择。  说明 集群版集群默认包含1主1只读两个节点，创建成功后最多可添加至1主15只读节点。关于如何添加节点，请参见增加或删除节点。 - 若原集群系列为单节点（入门级）或历史库单节点版，节点个数默认为1，无需选择。
存储费用	您购买时无需选择容量，PolarDB会根据实际使用量按小时计费，您也可以预购存储包，如何购买存储包请参见购买存储包。
时区	设置集群时区，默认时区为UTC+08:00。
表名大小写	设置集群表名是否区分大小写，建议选择不区分大小写（默认）或当本地数据库区分大小写时，建议选择区分大小写。  说明 集群创建后该参数无法修改，请谨慎选择。

参数	说明
删除（释放）集群时	设置删除（释放）集群时的备份保留策略，默认保留最后一个备份（释放前自动备份）。 - 保留最后一个备份（释放前自动备份）：删除集群时保留最后一个备份。 - 保留全部备份：删除集群时保留所有备份。 - 不保留备份（释放后无法恢复）：删除集群时不保留任何备份。  说明 删除（释放）集群时保留备份可能会产生少量费用，您可以随时删除备份来节省成本，详情请参见数据备份计费规则。
集群名称	输入集群名称，集群名称需满足如下要求： - 不能以 <code>http://</code> 或 <code>https://</code> 开头。 - 长度为2~256个字符。 如果留空，系统将为自动生成一个集群名称，创建集群后还可以修改。
资源组	从已创建的资源组中选择一个目标资源组，创建资源组的操作步骤，请参见创建资源组。  说明 资源组是在单个云账号下将一组相关资源进行统一管理的容器，一个资源只能归属于一个资源组，详情请参见RAM资源分组与授权。
购买时长	选择集群的购买时长。  说明 仅当付费模式为包年包月时支持设置该参数。
购买数量	选择集群的购买数量。

7. 根据集群的付费模式完成后续购买操作。

o 按量付费

a. 单击立即购买

b. 在**确认订单**页面确认订单信息，阅读并选中服务协议，单击**立即开通**即可

o 包年包月

a. 单击立即购买。

b. 在**确认订单**页面确认订单信息，阅读并选中服务协议，单击**去支付**。

c. 在**支付**页面，确认未支付订单信息和支付方式，单击**订购**。

开通成功后，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 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长。集群创建成功后，您可以返回[PolarDB控制台](#)并在**集群列表**查看新创建的集群。

相关API

API	描述
CreateDBCluster	从回收站恢复PolarDB的集群需要通过CreateDBCluster来实现。 说明 参数CreationOption取值需要为CloneFromPolarDB。

20.3. 彻底删除已释放的集群

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至新集群，或删除已释放集群的备份集。本文以PolarDB MySQL引擎集群为例介绍如何删除已释放集群的备份集。

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150 MB/s左右。更多关于备份的介绍请参见[数据备份](#)。

删除备份

- 登录[PolarDB控制台](#)。
- 在控制台左上角，选择集群所在地域。
- 在左侧导航栏中单击[集群回收站](#)。
- 找到目标集群，单击目标集群名称左侧的图标，展开备份集列表。
- 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				
pc- xxxxxx -hq2c polardbfortest	华东1（杭州）	2核 8GB	100% 兼容 MySQL 5.6	2021年10月14日 14:39:30	2021年11月1日 11:29:29	已释放	恢复数据到新集群				
2021年9月1日 - 2021年11月1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
1150880031	2021年11月1日 11:29:39	2021年11月1日 11:29:50	备份完成	2021年11月1日 11:29:41	2.31 GB	一级备份	有效	快照备份	全量备份	手动备份	删除备份
1150238569	2021年10月31日 19:57:02	2021年10月31日 19:57:13	备份完成	2021年10月31日 19:57:05	2.31 GB	一级备份	有效	快照备份	全量备份	自动备份	删除备份
1148605078	2021年10月30日 19:57:08	2021年10月30日 19:57:23	备份完成	2021年10月30日 19:57:11	2.31 GB	一级备份	有效	快照备份	全量备份	自动备份	删除备份

- 在弹出的对话框中，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。

21. 监控与优化

21.1. 一键诊断

PolarDB MySQL引擎的一键诊断融合了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。

功能介绍

- **自治中心**：您可以通过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）。详情请参见[自治中心](#)。
- **会话管理**：您可以通过会话管理功能查看目标实例的会话详情和会话统计等信息。详情请参见[会话管理](#)。
- **实时性能**：您可以通过实时性能功能了解目标集群的QPS、TPS和网络流量等信息。详情请参见[实时性能](#)。
- **空间分析**：您可以通过空间分析功能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。详情请参见[空间分析](#)。
- **锁分析**：您可以通过锁分析功能直观地查看和分析数据库最近一次发生的死锁。详情请参见[锁分析](#)。
- **性能洞察**：您可以通过性能洞察功能，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。详情请参见[性能洞察](#)。
- **诊断报告**：您可以通过诊断报告功能自定义诊断报告创建条件并查看诊断报告。详情请参见[诊断报告](#)。

21.2. 自治中心

PolarDB MySQL引擎的一键诊断融合了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

产品系列需为PolarDB MySQL引擎集群版或历史库。单节点不支持该功能。更多关于3个系列的介绍，请参见[概述](#)。

注意事项

- 仅商品类型为**按量付费**的PolarDB MySQL引擎集群才支持自动扩容和自动回缩，**包年包月**的集群暂不支持自动扩容和回缩，更多关于商品类型的说明，请参见[购买按量付费集群](#)。
- 若您的业务量波动较大且频繁，推荐您购买**PolarDB计算包**并配合DAS的自动扩容和回缩功能一起使用，当集群配置发生调整时，计算包能根据当前规格自动进行抵扣。

❓ **说明** 若您使用的PolarDB MySQL引擎集群购买方式为**包年包月**，则需要将其转为**按量付费**后才能使用计算包进行抵扣，详情请参见[包年包月转按量付费](#)。

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，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**自治中心**页签。



6. 单击右上角自治功能开关。



7. 在设置对话框中，单击自治功能开关页签，打开开启自治服务。

❓ 说明 打开开启自治服务后，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）。



8. 开启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。

- **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。您可通过单击开启自动索引创建实现创建自动索引功能的开启；通过单击开启自动索引删除实现删除自动索引功能的开启。
- **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。

❓ 说明 例如在可限流时间段（默认可限流时间段为00:00~23:59）内，自定义触发条件为当CPU使用率大于70%，同时活跃会话数大于16，且该现象持续时间超过2分钟时触发自动限流，同时自动开始跟踪，如果发现故障未消除，则自动回滚该限流操作。自动限流触发后，限流操作持续的时间不会超过最大限流时间，详情请参见[自动SQL限流](#)。

- **自动扩容和自动回缩**：您可以开启自动扩、缩容的开关，并自定义扩容触发条件，当条件满足时会自动触发扩、缩容。

参数名称	参数说明
自动扩容	打开自动扩容的开关。
观测窗口	在整个观测窗口期内，若CPU平均使用率大于等于设定值，则在观测窗口结束后，PolarDB将根据集群的实时读写流量情况选择增加节点或升配的扩容方式。例如观测窗口期是5分钟，自动扩容的时间是10分钟，所以您需要等待15分钟左右才能看到自动扩容的效果。
CPU平均使用率	自动扩容的触发阈值。当CPU平均使用率大于或等于设置的值时，就会触发自动扩容。
扩容规格上限	自动扩容的规格上限。触发自动扩容后，会逐级扩容PolarDB的集群规格，例如从4核到8核再到16核，直到升级至扩容规格的上限。
只读节点数量上限	自动扩容只读节点的数量上限。触发自动扩容后，会逐个增加PolarDB的只读节点，直到增加至上限。 ❓ 说明 自动扩容的节点会加入集群默认地址，自定义地址需要配置新节点自动加入。关于如何配置新节点自动加入，详情请参见配置数据库代理。
自动回缩	打开自动回缩的开关。 ❓ 说明 开启自动回缩后，在静默期内如果CPU平均使用率小于30%的时间超过静默期时间的99%，则在静默期结束后，将触发自动逐级回缩PolarDB集群规格，直到回缩至原始规格。
静默期	两次自动扩容或自动回缩的最小间隔时间。在静默期内，PolarDB会持续进行观测，但不会触发自动扩容或缩容。若在静默期与观测窗口同时结束，且该观测窗口内CPU使用率达到调整阈值，PolarDB会在静默期和观测窗口同时结束时触发自动规格调整。

9. 单击**确定**。

21.3. 会话管理

PolarDB MySQL引擎的一键诊断融合了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. 在[集群列表页](#)，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[会话管理](#)页签。

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. 在会话管理页面，您可以查看实例会话和会话统计的详细信息。

- 实例会话：支持查看异常会话、活跃会话及CPU使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		按用户统计(1)			按访问来源统计(1)			按数据库统计(1)	
用户	统计项	用户	活跃数	总数	来源	活跃数	总数	DB	活跃数
会话总数	3	root	0	3		0	3		0
运行中会话总数	0								
运行中会话最长时间	0								

21.4. 实时性能

PolarDB MySQL引擎的一键诊断融合了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

- 登录[PolarDB控制台](#)。
- 在控制台左上角，选择集群所在地域。
- 在集群列表页，单击目标集群ID。
- 在左侧导航栏中，选择[诊断与优化](#) > [一键诊断](#)。
- 在[实时性能](#)页面，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。



6. 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。



21.5. 空间分析

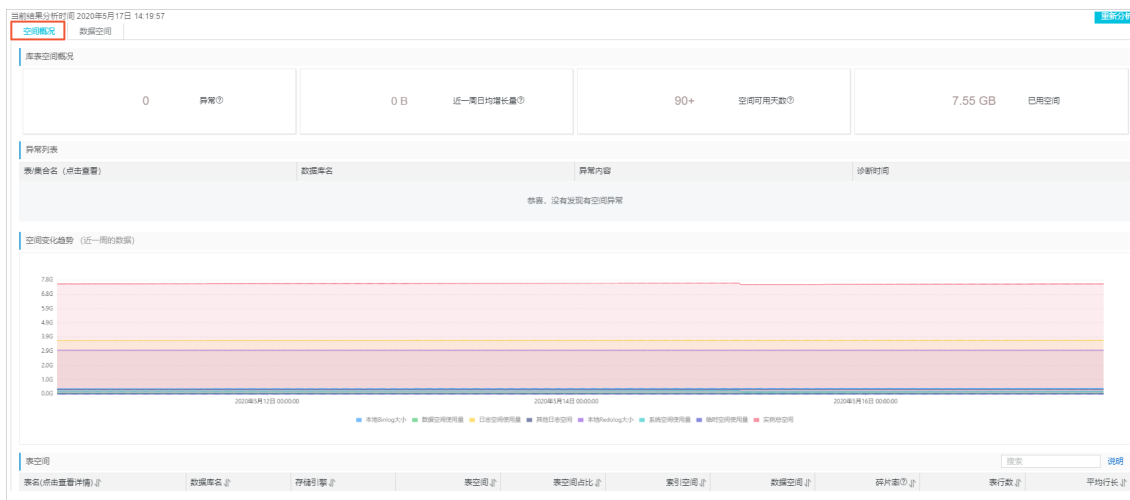
PolarDB MySQL引擎的一键诊断融合了DAS部分功能，您可以通过空间分析，来查看目标实例的空间使用概况、空间剩余可用天数，以及数据库中某个表的空间使用情况、空间碎片、空间异常诊断等。

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. 在集群列表页，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。



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空间使用情况。单击表名可查看对应表的字段和索引。

5.6三节点 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

表空间 (576.00 KB)

表名(点击查看详情)

表名	数据库名	存储引擎
order	ImoDB	
dislectdata	ImoDB	

表名: order, 库名: ImoDB

字段

字段名	类型	描述	可否	自增	缺省值
orderid	int(11)		NO	NO	
username	char(32)		YES	NO	*
ordertime	datetime		YES	NO	
commodity	varchar(32)		YES	NO	*
phonenum	int(11)		YES	NO	
address	text		YES	NO	

21.6. 锁分析

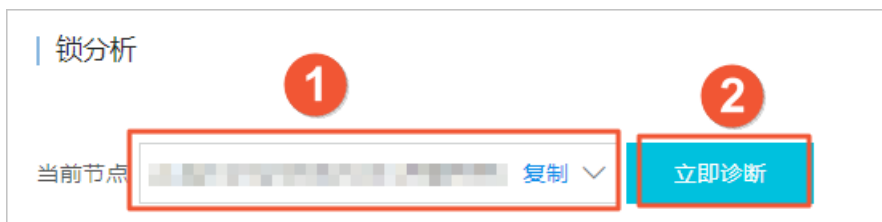
PolarDB MySQL引擎的一键诊断融合了DAS部分功能，您可以通过其中的锁分析直观地查看和分析数据库最近一次发生的死锁。

操作步骤

- 登录PolarDB控制台。
- 在控制台左上角，选择集群所在地域。
- 在集群列表页，单击目标集群ID。
- 在左侧导航栏中，选择诊断与优化 > 一键诊断。
- 单击锁分析页签。



- 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。



- 在新生成的死锁诊断列表右侧，单击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

 说明 仅当发现死锁为是时，才可以单击查看详情。

- 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		

21.7. 诊断报告

PolarDB MySQL引擎的一键诊断融合了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

- 登录PolarDB控制台。
- 在控制台左上角，选择集群所在地域。
- 在集群列表页，单击目标集群ID。
- 在左侧导航栏中，选择诊断与优化 > 一键诊断。
- 单击诊断报告页签。

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
------	------	------	------	-----	----------------------	------

- 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：

- 单击实例诊断报告列表旁边的自动生成报告设置按钮。
- 在弹出的对话框内，选中定时触发复选框。
- 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。
- 单击确认。

自动生成诊断报告设置

☐ 通过事件触发自动诊断（正在支持中）
 ☒ 定时触发

☒ 全部
 ☒ 星期一
 ☒ 星期二
 ☒ 星期三
 ☒ 星期四
 ☒ 星期五
 ☒ 星期六
 ☒ 星期日

时间:

由于诊断报告生成任务的调度和负载等原因，可能会延迟生成
(例如设置了0点生成诊断报告，可能在00:10诊断报告才最终生成)

确认
取消

- 手动生成诊断报告：单击发起诊断生成即时诊断报告。

实例诊断报告列表 (自动生成报告设置)

发起诊断
近1天
近3天
近1周
2020年5月16日 14:31:55 - 2020年5月17日 14:31:55
查看

- 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

- 在实例诊断报告详情中，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 [扣分详情](#)

[<< 返回列表](#)
[返回顶部](#)
[健康状况](#)
[会话列表](#)
[慢SQL](#)
[表空间](#)
[死锁](#)
[性能趋势](#)
[下载](#)

实例基本信息

实例ID: 772761df1cad9018c5269b119...	实例来源: 自建实例
数据库类型: MySQL	库名: 库名

节点	角色	规格	最大连接数
只读节点		规格: 5核16GB 磁盘: 160GB	8000

21.8. 性能洞察

PolarDB MySQL引擎的一键诊断融合了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。

前提条件

产品系列需为PolarDB MySQL引擎集群版或历史库。单节点不支持该功能。更多关于3个系列的介绍，请参见[概述](#)。

背景信息

用于性能洞察的数据主要来源如下：

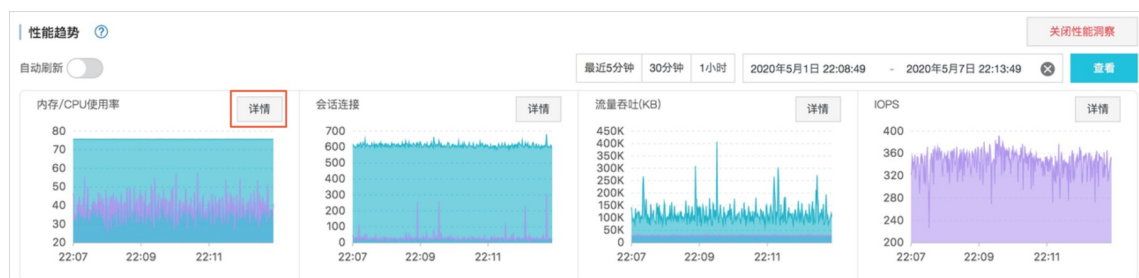
- 如果目标实例已经开启了performance_schema，直接采集和分析performance_schema中的数据。
- 如果目标实例未开启performance_schema，则采集和分析活跃会话数据。

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. 在集群列表页，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**。

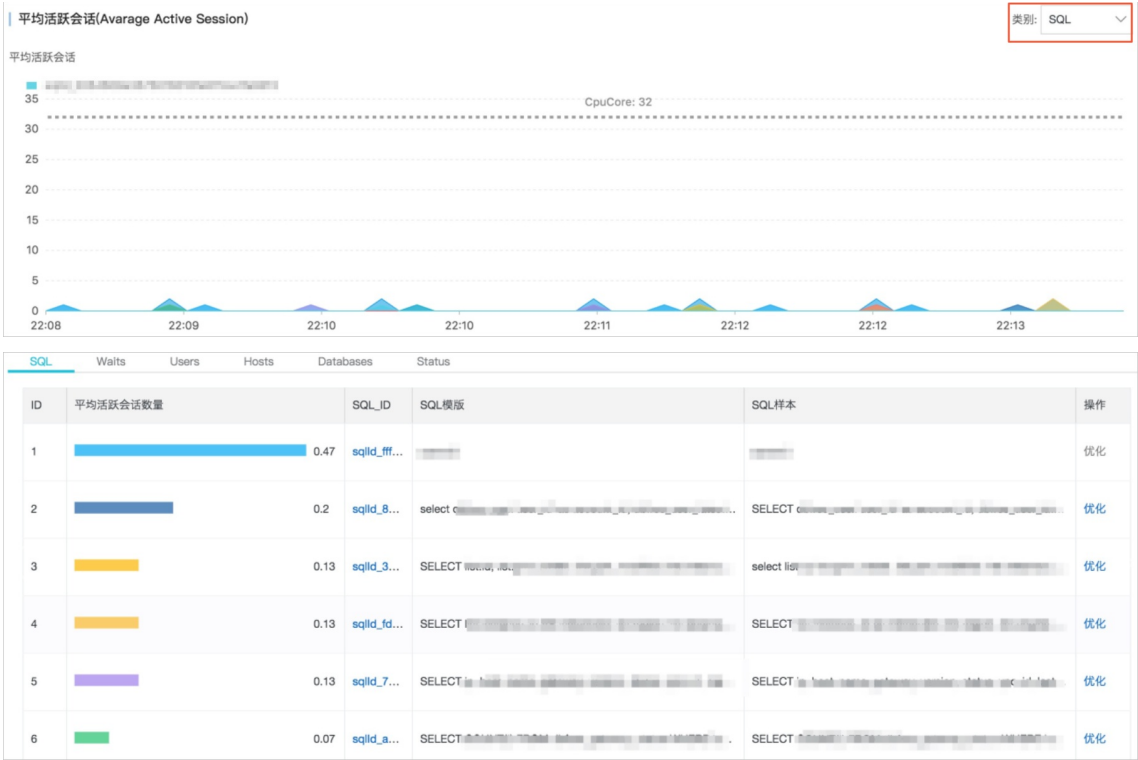


7. 在弹出的对话框内，单击**确定**。
8. 您可以在性能洞察页面查看和管理如下信息：
 - 在**性能趋势**区域，您可以查看特定时段的数据库性能情况。若您需要查看某个具体性能（如CPU使用率），可以单击该性能名称右侧的**详情**按钮进行查看。



说明 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。



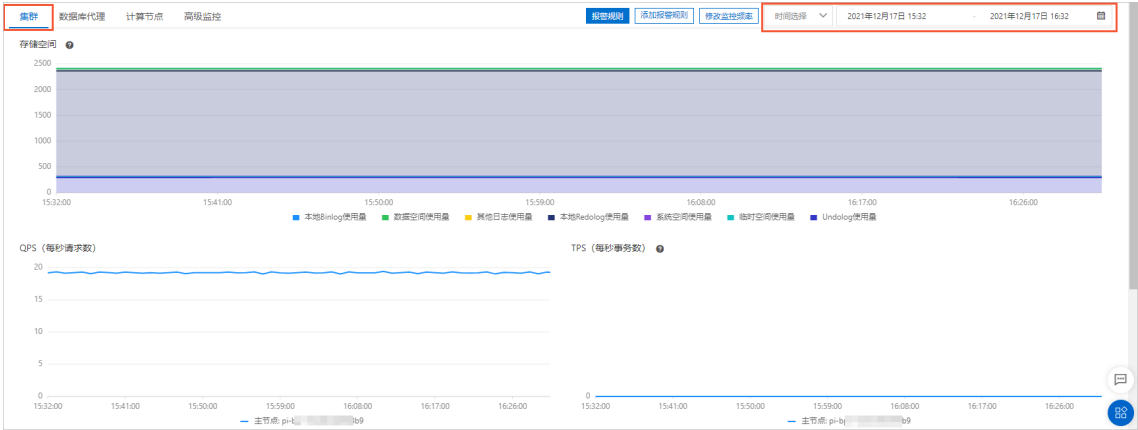
21.9. 性能监控

21.9.1. 查看性能监控指标

PolarDB控制台提供了丰富的性能监控项和秒级监控频率，方便您掌握集群的运行状态并通过细粒度的监控数据快速定位运维问题。

操作步骤

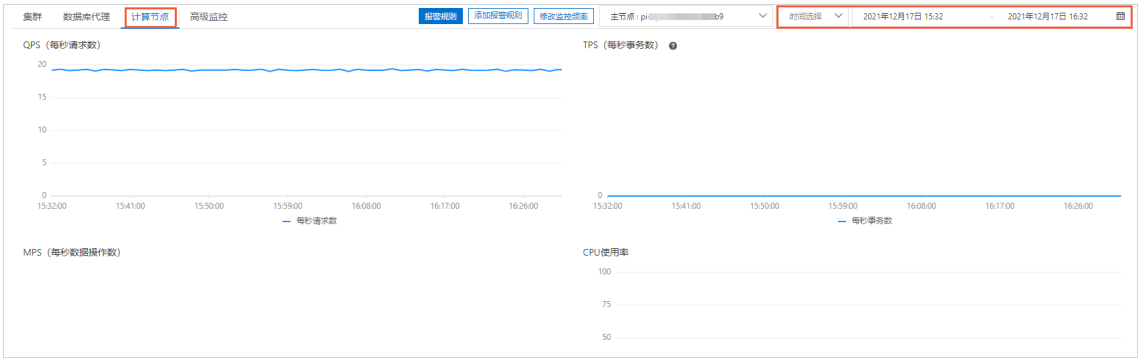
1. 登录[PolarDB控制台](#)。
2. 在控制台左上角，选择集群所在地域。
3. 找到目标集群，单击集群ID。
4. 在左侧导航栏中，选择[诊断与优化](#) > [性能监控](#)。
5. 您可以根据业务需求选择查看[集群](#)、[数据库代理](#)、[计算节点](#)或[高级监控](#)的监控信息，详情请参见[监控项说明](#)。
 - [集群性能监控](#)：单击[集群](#)页签，在右侧设置时间段后单击[确定](#)。



○ 数据库代理性能监控：单击数据库代理页签，在右侧设置时间段后单击确定。



○ 计算节点性能监控：单击计算节点页签，在右侧选择节点并设置时间段后单击确定。



② 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引擎，可以帮助您快速识别异常集群，进行进一步的优化分析。

监控项说明

类别	监控项	说明
	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。该指标请联系阿里云技术支持获取。
	QPS	展示各节点的每秒请求数。来源于 <code>show global status</code> 命令查询结果中的 <code>Questions</code> 指标。

类别	监控项	说明
集群	TPS	展示各节点的每秒事务数。来源于 <code>show global status</code> 命令查询结果中的 <code>Com_commit</code> 和 <code>Com_rollback</code> 指标。
	MPS	展示各节点的每秒数据操作数。来源于 <code>show global status</code> 命令查询结果中的 <code>Com_insert</code> 、 <code>Com_insert_select</code> 、 <code>Com_update</code> 、 <code>Com_update_multi</code> 、 <code>Com_delete</code> 、 <code>Com_delete_multi</code> 、 <code>Com_replace</code> 、 <code>Com_replace_select</code> 指标。
	CPU使用率	展示各节点的CPU使用率。该指标请联系阿里云技术支持获取。
	内存使用率	展示各节点的内存使用率。该指标请联系阿里云技术支持获取。
	连接	展示各节点的每秒活跃连接数。来源于 <code>show processlist</code> 命令查询结果中的统计信息。
计算节点	QPS	展示所选择节点的每秒请求数。来源于 <code>show global status</code> 命令查询结果中的 <code>Questions</code> 指标。
	TPS	展示所选择节点的每秒事务数。来源于 <code>show global status</code> 命令查询结果中的 <code>Com_commit</code> 和 <code>Com_rollback</code> 指标。
	MPS	展示各节点的每秒数据操作数。来源于 <code>show global status</code> 命令查询结果中的 <code>Com_insert</code> 、 <code>Com_insert_select</code> 、 <code>Com_update</code> 、 <code>Com_update_multi</code> 、 <code>Com_delete</code> 、 <code>Com_delete_multi</code> 、 <code>Com_replace</code> 、 <code>Com_replace_select</code> 指标。
	CPU使用率	展示所选择节点的CPU使用率。该指标请联系阿里云技术支持获取。
	内存使用率	展示所选择节点的内存使用率。该指标请联系阿里云技术支持获取。
	连接	展示所选择节点的当前总连接数和活跃连接数。来源于 <code>show processlist</code> 命令查询结果中的统计信息。
	操作	展示所选择节点的每秒操作数，包括DELETE、INSERT、UPDATE、REPLACE等等。来源于 <code>show global status</code> 命令查询结果中的对应的以 <code>Com_</code> 开头的指标。
	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。来源于 <code>show global status</code> 命令查询结果中的 <code>Innodb_buffer_pool_pages_dirty</code> 、 <code>Innodb_buffer_pool_pages_total</code> 、 <code>Innodb_buffer_pool_reads</code> 、 <code>Innodb_buffer_pool_read_requests</code> 等指标。
	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。该指标请联系阿里云技术支持获取。
	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。该指标请联系阿里云技术支持获取。

类别	监控项	说明
	网络	展示所选择节点的每秒输入输出流量。来源于 <code>show global status</code> 命令查询结果中的 <code>Bytes_received</code> 、 <code>Bytes_sent</code> 指标。
	扫描行数	展示所选择节点每秒插入、读取、更新、删除的行数。来源于 <code>show global status</code> 命令查询结果中的 <code>Innodb_rows_deleted</code> 、 <code>Innodb_rows_inserted</code> 、 <code>Innodb_rows_read</code> 、 <code>Innodb_rows_updated</code> 指标。
	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。来源于 <code>show global status</code> 命令查询结果中的 <code>Innodb_data_read</code> 、 <code>Innodb_data_written</code> 指标。
	InnoDB缓冲池请求数	展示所选择节点缓冲池的每秒读写次数。来源于 <code>show global status</code> 命令查询结果中的 <code>Innodb_buffer_pool_read_requests</code> 、 <code>Innodb_buffer_pool_write_requests</code> 指标。
	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。来源于 <code>show global status</code> 命令查询结果中的 <code>Innodb_log_write_requests</code> 、 <code>Innodb_os_log_fsyncs</code> 指标。
	临时表	展示所选择节点每秒创建的临时表数量。来源于 <code>show global status</code> 命令查询结果中的 <code>Created_tmp_disk_tables</code> 指标。

常见问题

- Q: 当集群无业务流量时，为何性能监控中的QPS还显示为大约10次/秒？
A: 因为系统后台存在监控、日志采集以及管控任务，大概每秒会产生10个左右的查询请求，对集群运行基本无影响。
- Q: CPU使用率过高时，应该如何解决？
A: 建议您按如下步骤进行排查：
 - 确认是否有大量慢请求，建议优化慢SQL后再进行测试。如何查看和优化慢SQL，请参见[慢SQL](#)。
 - 确认CPU曲线是否与QPS或TPS曲线走向一致，若一致，说明该问题是事务高并发导致，可以考虑升级集群配置。如何升配，请参见[手动变配](#)。
 - 若没有慢请求，CPU曲线与QPS或TPS曲线走向也不一致，请[提交工单](#)联系技术支持解决。
- Q: 当前连接数远大于活跃连接数时，应该如何解决？
A: 您可以尝试将`wait_timeout`和`interactive_timeout`参数值设置的小一些以加速空闲连接的释放。但建议最好在业务端使用完连接后及时关闭回收以减少空闲连接的存在。

相关API

API	描述
DescribeDBClusterPerformance	查询PolarDB集群的性能数据。
DescribeDBNodePerformance	查询PolarDB集群节点的性能数据。

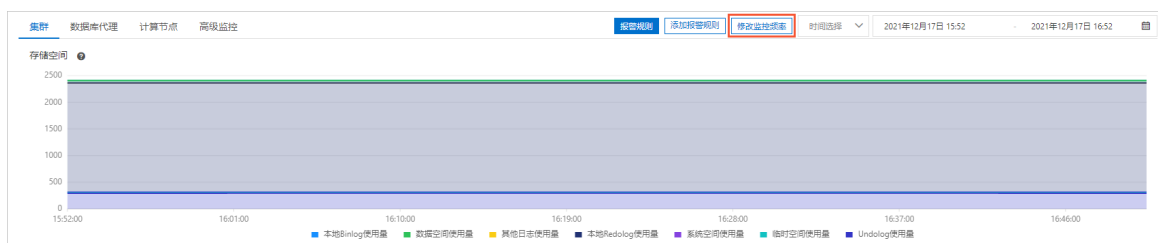
API	描述
<code>DescribeDBClusterMonitor</code>	查询PolarDB集群监控数据的采集频率。

21.9.2. 修改监控频率

本文介绍了修改性能监控频率的操作步骤。

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，您可以根据业务需求选择监控频率为**5秒/次**或**60秒/次**（默认值）。

监控频率设置

监控频率：

☐ 5秒/次
 ☒ 60秒/次

确定

取消

- 若设置监控频率为**5秒/次**，则：
 - 当查询时间范围小于等于1小时，监控数据的显示粒度为5秒。
 - 当查询时间范围小于等于1天，监控数据的显示粒度为1分钟。
 - 当查询时间范围小于等于7天，监控数据的显示粒度为10分钟。
 - 当查询时间范围小于等于30天，监控数据的显示粒度为1小时。
 - 当查询范围大于30天，监控数据的显示粒度为1天。
- 若设置监控频率为**60秒/次**，则：
 - 当查询范围小于等于1天，监控数据的显示粒度为1分钟。
 - 当查询范围小于等于7天，监控数据的显示粒度为10分钟。
 - 当查询范围小于等于30天，监控数据的显示粒度为1小时。
 - 当查询范围大于30天，监控数据的显示粒度为1天。

7. 单击**确定**。

相关API

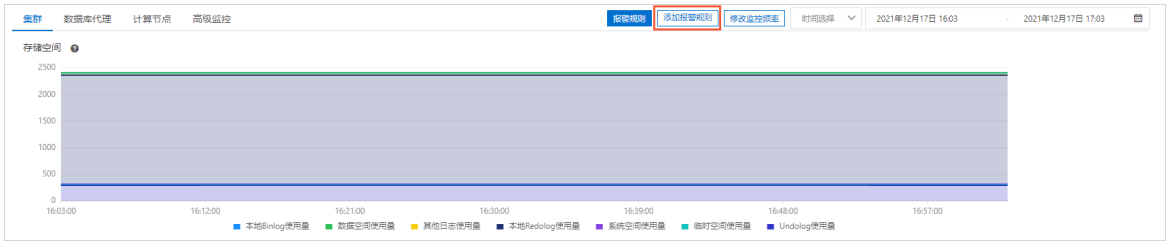
API	描述
ModifyDBClusterMonitor	修改PolarDB集群监控数据的采集频率。

21.9.3. 创建报警规则

您可以通过PolarDB控制台创建阈值报警规则，帮助您及时了解PolarDB集群或节点的监控数据异常并快速进行处理。

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**添加报警规则**。



6. 在创建报警规则页面，设置以下参数。

步骤	参数	说明
关联资源	产品	保持默认值云数据库POLARDB-MySQL(新版)即可。
	资源范围	报警规则的作用范围，取值范围为全部资源或集群。 ❓ 说明 - 若选择资源范围为全部资源，则当产品下任何集群满足报警规则描述时，都会发送报警通知。 - 若选择资源范围为集群，则仅当目标集群满足报警规则描述时，才会发送报警通知。
	规则名称	输入报警规则的名称。

步骤	参数	说明
设置报警规则	规则描述	报警规则的主体，定义在监控数据满足指定条件时，触发报警规则。 ? 说明 更多关于报警规则的举例说明，请参见创建阈值报警规则。
	通道沉默期	指报警发生后如果关联资源未恢复正常，间隔多久重复发送一次报警通知，最短为5分钟，最长为24小时。
	生效时间	报警规则的生效时间。 ? 说明 报警规则只在生效时间内发送报警通知，非生效时间内产生的报警只记录报警历史。
通知方式	关于通知方式步骤的参数详情，请参见 创建阈值报警规则 。	

7. 单击**确认**。

21.9.4. 管理报警规则

您可以通过PolarDB控制台管理阈值报警规则，帮助您及时了解PolarDB集群或节点的监控数据异常并快速进行处理。

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**报警规则**，进入报警规则列表页面。

报警规则列表 ← 返回 刷新									
阈值报警 事件报警 创建报警规则 请输入进行查询 搜索									
规则名称	状态 (全部)	启用	监控项 (全部)	维度 (全部)	报警规则	产品名称 (云数据库 PolarDB)	通知对象	操作	
☐ 测试		已禁用	连接连接数	resource_ALL	连接连接数 >= 5个 Info 连续1次就报警	云数据库PolarDB- PostgreSQL/Oracle	POLARDB报警联系人组 查看	查看	报警历史
☐ 旧版测试		已禁用	连接数使用率	resource_ALL	连接数使用率 >= 30% Info 连续1次就报警	云数据库 POLARDB(旧版)	测试 查看	查看	报警历史
☐ CPU负载报警		已禁用	CPU使用率	clusterId: 云数据库POLARDB(旧版)	CPU使用率 >= 50% Warn 连续3次就报警	云数据库POLARDB- MySQL(新版)	POLARDB报警联系人组 查看	查看	报警历史
☐ 内存使用报警		已禁用	内存使用率	instanceId: 云数据库POLARDB(旧版)	内存使用率 >= 5% Warn 连续1次就报警	云数据库 POLARDB(旧版)	shukun 云账号报警联系人 查看	查看	报警历史
☐ 启用 禁用 删除 共 4 条 10 1									

6. 在**阈值报警**页签下，您可以对已有的报警规则执行以下操作。
 - 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。
- 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。

21.10. 慢SQL

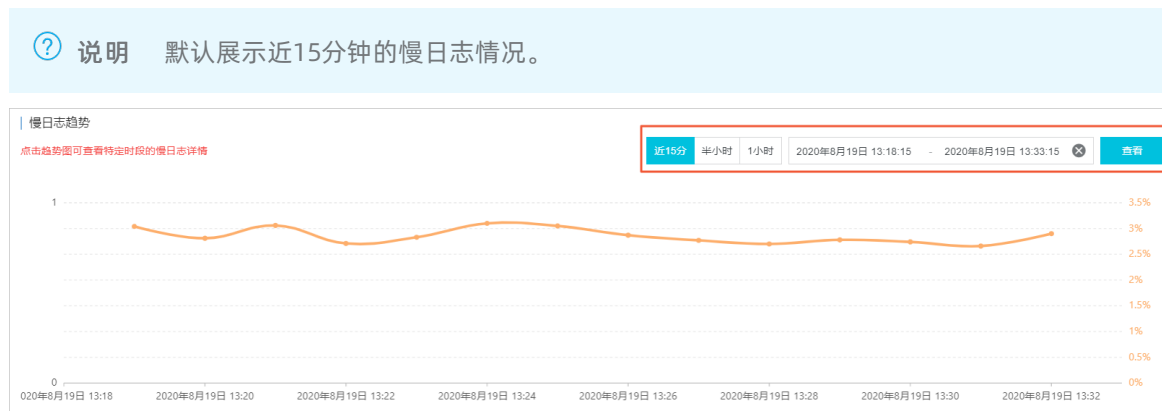
PolarDB MySQL引擎提供慢SQL分析功能，能够查看慢日志趋势和统计信息，并且提供SQL建议和诊断分析。

前提条件

产品系列需为PolarDB MySQL引擎集群版或历史库。单节点不支持该功能。更多关于3个系列的介绍，请参见[概述](#)。

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。
- 找到目标集群，单击集群ID。
- 在左侧导航栏中，选择**诊断与优化 > 慢SQL**。
- 您可以通过以下两种方式查看慢日志趋势：
 - 单击近15分、半小时或1小时，查看最近15分钟、最近半小时或最近1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30天之内的慢日志信息，且起始时间间隔需在24小时之内。



- (可选) 如果您只需要查看某个节点的慢SQL情况，可以在节点列表中，单击目标节点ID，即可查看目标节点慢SQL情况。

节点列表 (时间范围: 2020年8月19日 13:40:54 - 2020年8月19日 13:55:54)

节点	角色	慢请求数量
pl-xxxxxxxxxxxxxx	主节点	0
pl-xxxxxxxxxxxxxx	只读节点	0

- 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。

慢日志统计 (时间范围: 2020年4月27日 15:00 - 2020年4月27日 16:00)

SQL模板	库名	执行次数	平均执行耗时 (秒)	平均阻塞耗时 (秒)	平均等待耗时 (秒)	平均扫描行数	平均返回行数	平均变更行数	平均逻辑读	操作
SE-xxxxxxxxxxxxxx	xxxxxxxxxxxxxx	437	2.340	0.000	0.00	60.6K	30.3K	0.0	119.7K	样本 优化

- 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。

慢日志统计 (时间范围: 2020年4月27日 15:00 - 2020年4月27日 16:00)

导出慢日志

SQL模板	库名	执行次数	平均执行耗时 (秒)	平均阻塞耗时 (秒)	平均排队耗时 (秒)	平均扫描行	平均返回行	平均变更行	平均逻辑读	操作
SE		437	2.340	0.000	0.00	60.6K	30.3K	0.0	119.7K	样本 优化

- 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: 2020年4月27日 15:00 - 2020年4月27日 16:00)

导出慢日志

SQL模板	库名	执行次数	平均执行耗时 (秒)	平均阻塞耗时 (秒)	平均排队耗时 (秒)	平均扫描行	平均返回行	平均变更行	平均逻辑读	操作
SE		437	2.340	0.000	0.00	60.6K	30.3K	0.0	119.7K	样本 优化

相关API

API	说明
DescribeSlowLogRecords	查看PolarDB MySQL引擎集群的慢日志明细。
DescribeDBClusterAuditLogCollector	查看PolarDB MySQL引擎集群的SQL采集功能（例如审计日志、SQL洞察等）是否开启。
ModifyDBClusterAuditLogCollector	开启或关闭PolarDB MySQL引擎集群的SQL采集功能（例如审计日志、SQL洞察等）。

21.11. SQL洞察

PolarDB MySQL引擎的SQL洞察融合了DAS部分功能，您可以通过SQL洞察一键开启DAS专业版的搜索、SQL洞察、安全审计、流量回放和压测功能，帮助您更好地获取SQL语句的具体信息、性能问题排查、识别高危风险来源，并且提供流量回放和压测功能，帮助您验证您的实例规格是否需要扩容，有效应对业务流量高峰。

费用说明

SQL洞察功能按照审计日志存储的容量收取费用：

- 中国内地地域：0.008元/GB/小时。
- 中国香港及海外地域：0.0122元/GB/小时。

 **说明** 审计日志的存储容量按量扣费，不支持包年包月。更多详情请参见[SQL洞察计费规则（可选）](#)。

功能说明

- 增强搜索
查询并导出SQL语句及对应的各种信息，如数据库、状态、执行时间等，详情请参见[搜索](#)。
- SQL洞察。
对SQL进行健康情况诊断、性能问题排查、业务流量分析等，详情请参见[SQL洞察](#)。
- 安全审计
可自动识别高危SQL、SQL注入、新增访问来源等风险，详情请参见[安全审计](#)。
- 流量回放和压测
提供流量回放和压测功能，帮助您验证您的实例规格是否需要扩容，有效应对业务流量高峰，详情请参见[流量回放和压测](#)。

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. 找到目标集群，单击集群ID。
4. 在左侧导航栏中，选择日志与审计 > SQL洞察。
5. 单击一键开启。



- 如果您尚未购买DAS专业版，单击前往[购买](#)并确认订单进行支付，开通DAS专业版。
- 如果您已购买DAS专业版，直接单击开启专业版。

DAS专业版开通成功后，您就可以开始使用DAS专业版的搜索、SQL洞察、安全审计、流量回放和压测功能。

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. 找到目标集群，单击集群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. 找到目标集群，单击集群ID。
4. 在左侧导航栏中，选择日志与审计 > SQL洞察。
5. 单击日志列表右侧的导出。

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(毫秒) ↑	更新行数 ↑	扫描行数 ↑
-------	-----	------	----	-------	----	----	----------	------------	--------	--------

6. 在导出SQL记录对话框中，选中需要导出的选项并确认导出的时间范围，单击**确认**。

导出SQL记录

导出字段

☒ 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

导出时间范围

2021年1月13日 14:02:31 - 2021年1月13日 14:17:31

确认

取消

7. 导出完成后，您可以在导出SQL日志记录面板中，单击下载，SQL日志记录将以已 `.csv` 文件形式导出。

SQL洞察

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

时间范围

2018-12-21 13:04:15 - 2018-12-21 13:19:15 自定义

数据库

可组合查询，如：DB1 DB2 DB3

用户

可组合查询，如：user1 user2

操作类型

☐ SELECT ☐ INSERT ☐ UPDATE

开启高级查询

查询

日志列表

SQL语句	数据库	线程ID	用户	客户端IP	操作	状态
SELECT * FROM table1	DB1	12345	user1	192.168.1.1	SELECT	成功
INSERT INTO table2	DB1	12346	user1	192.168.1.1	INSERT	成功
UPDATE table3	DB1	12347	user1	192.168.1.1	UPDATE	成功

导出SQL日志记录

100%

下载

最近导出任务

创建时间	记录条数	操作
2018-12-21 13:05:33	556	下载

确定

取消

8. 单击**确定**即可。

关闭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. 找到目标集群，单击集群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. 在弹出的对话框中，关闭**开启SQL洞察**开关后单击**确定**。



说明 SQL洞察功能关闭后，SQL审计日志会被清空，建议先导出SQL记录后，再关闭SQL洞察功能。如何导出SQL记录，请参见[导出SQL记录](#)。

7. 在弹出的对话框中，单击确定。

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，选择账单管理 > 账单详情。
4. 在账单详情页签，单击明细账单页签，设置搜索实例ID并输入目标集群ID进行搜索。

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。

常见问题

Q: SQL洞察的全量请求统计区域中的 `logout!` 语句是什么？

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(毫秒)	更新行数	扫描行数
logout!		1144944104			LOGOUT	失败(1158)	14624729.38	2022-05-05 11:42:04.401	0	0

A: `logout!` 是指连接断开。`logout!` 的耗时，是指上一次交互时间到 `logout!` 发生时间的差值，可以理解为连接空闲的时长。状态列的1158是指网络链接断开，其可能的原因是：

- 客户端连接超时。

- 服务端异常断开。
- 服务端连接Reset（超过interactive_timeout或wait_timeout时长）。

22. 版本管理

PolarDB集群架构共三层：数据库代理Proxy、数据库内核引擎DB和数据库分布式存储Store。您可以根据实际情况单独升级Proxy或内核引擎，也可以绑定一起升级。

注意事项

- 版本升级一般不超过30分钟，升级过程中会重启数据库代理Proxy或内核引擎DB，可能会导致数据库连接闪断。请您尽量在业务低峰期执行升级操作，并且确保您的应用有自动重连机制。
- 同时升级数据库代理（Proxy）和内核引擎期间，主地址和集群地址均会有30~90秒的连接闪断，请确保应用具备重连机制。
- 仅升级数据库代理（Proxy）期间，集群地址和自定义地址会有30秒的连接闪断，主地址的连接不受影响，请确保应用具备重连机制。
- 仅升级内核引擎期间，数据库代理Proxy的版本高于2.4.7（包含）的PolarDB集群可以通过Connection Preserving技术保护95%的数据库连接不中断。
- 升级过程中无法使用PolarDB控制台的部分变更类功能（如升降配置、增删节点、修改参数、重启节点），但查询类功能（如性能监控）不受影响。
- 版本升级后无法降级。

查看版本信息

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，选择配置与管理 > 版本管理。
5. 在版本信息区域，查看数据库代理Proxy和内核引擎DB的版本信息。

升级版本

若集群当前数据库代理Proxy或内核引擎DB的版本不是最新版本，则可以根据实际需要进行升级操作。

1. 进入目标集群的配置与管理 > 版本管理菜单，在升级版本区域，您可以根据需要选择同时升级数据库代理（Proxy）和内核引擎、仅升级内核引擎或仅升级数据库代理（Proxy）。



说明

- 若集群当前数据库代理Proxy或内核引擎DB的版本已经是最新版本，则同时升级数据库代理（Proxy）和内核引擎、仅升级内核引擎或仅升级数据库代理（Proxy）选项将置灰不可选。
- 若选择仅升级数据库代理（Proxy），将只升级读写分离相关功能，例如一致性级别（全局一致性）、事务拆分、主库是否接受读等。

2. 单击立即升级或可维护窗口升级。

若选择在可维护窗口升级，您还可以在计划任务页查看该任务的具体信息或取消该任务，详情请参见[查看或取消计划任务](#)。

 注意

- 同时升级数据库代理（Proxy）和内核引擎期间，主地址和集群地址均会有30 ~ 90秒的连接闪断，请确保应用具备重连机制。
- 仅升级数据库代理（Proxy）期间，集群地址和自定义地址会有30秒的连接闪断，主地址的连接不受影响，请确保应用具备重连机制。

3. 在弹出的对话框中，单击确定即可。

相关API

API	描述
DescribeDBClusterVersion	查看PolarDB MySQL引擎集群当前内核版本的详细信息。
UpgradeDBClusterVersion	将PolarDB MySQL引擎集群更新至最新版本。

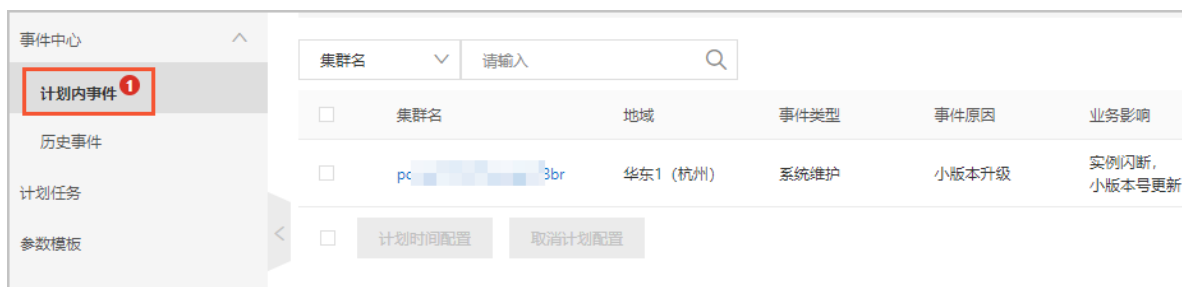
23. 计划内运维事件

23.1. 查看并管理计划内事件

PolarDB计划内的运维事件（例如数据库软件升级、硬件维护与升级）除了会通过短信、语音、邮件或站内信通知您，还会在控制台上进行通知。您可以在计划内事件中，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。

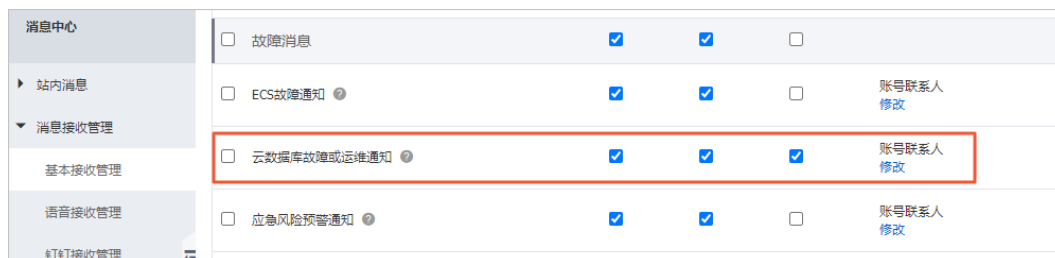
注意事项

- 当您有未处理的运维事件时，可以在控制台左侧导航栏的事件中心 > 计划内事件中看到事件提醒。



- 云数据库的计划内事件（如数据库软件升级、硬件维护与升级等）通常会至少在执行前的3天通知您，通知方式为短信、语音、邮件、站内信或控制台等。您需要登录消息中心，确保云数据库故障或运维通知的通知开关处于开启状态并设置消息接收人（推荐设置为数据库运维人员），否则您将无法收到相应的通知信息。

消息中心通知设置



操作步骤

- 登录PolarDB控制台。
- 在控制台左上角，选择集群所在地域。
- 在左侧导航栏中，单击事件中心 > 计划内事件。

② 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。

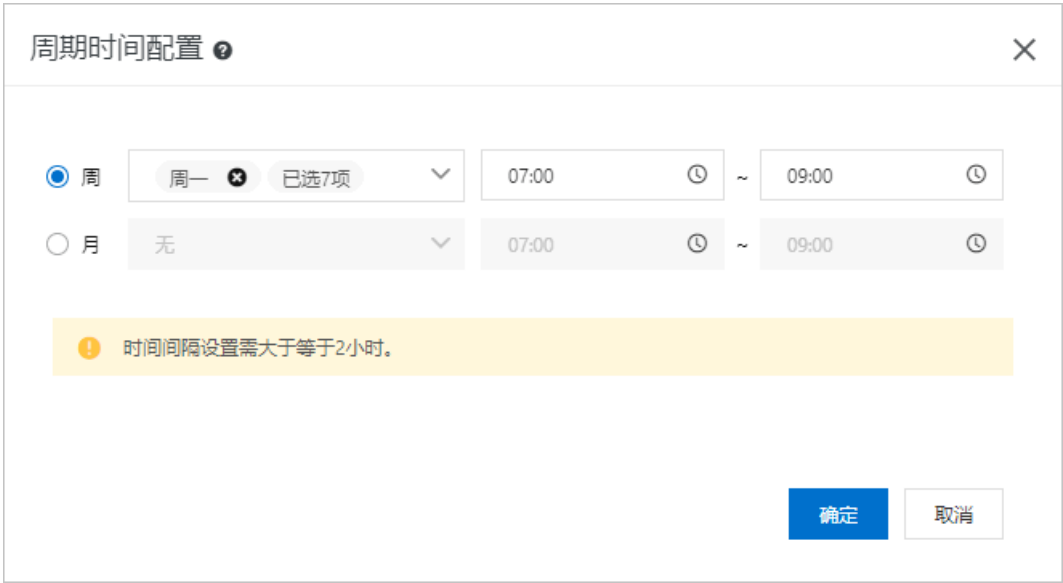
- （可选）在计划内事件页面，可以进行周期时间配置。

i. 在页面右侧，单击周期时间配置。



说明 周期时间配置是数据库主动运维事件的全局配置项（不包含紧急风险修复类事件）。周期时间设置后，新生成的主动运维事件的计划切换时间会自动命中周期时间；如果不设置周期时间，新生成的主动运维事件的计划切换时间会自动命中集群的可维护窗口时间。

ii. 在弹出的对话框中，设置周期时间并单击确定。



5. 在计划内事件页面，可以查看事件的详细信息，如需修改事件的切换时间，请选中目标集群，单击计划时间配置。



6. 在计划时间配置对话框中，设置计划切换时间并单击确定。

计划时间配置

可以配置最晚操作时间2021年9月5日 23:59:59前任意时间点切换

2021年8月4日 00:06:04

☐ 设置最早执行时间

⚠ 您所选的实例将在指定时间进行切换，请再次确认

集群名	最晚操作时间	计划切换时间
pc-xxxxx78br	2021年9月5日 23:59:59	2021年8月3日 00:03:01

确定
取消

说明

- 您可以勾选设置最早执行时间，系统将自动填充最近的预约切换日期和预约切换时间。确定后集群开始切换准备，进入待处理状态；取消勾选后，可自定义修改预约切换日期和时间。
- 计划切换时间不能晚于最晚开始时间。

事件的原因与影响

升级类型	事件原因	影响类型	影响说明
	实例迁移	实例闪断	进入后，将产生下述影响： - 一般情况下，实例小版本升级采用热升级模式。实例或实例中涉及切换的分片将发生连接闪断及30秒以内的只读状态（用于等待数据完全同步），请在业务低峰期执行，并确保应用程序具备重连机制。 - 短暂影响该实例在DMS和DTS中的使用，操作完成后自动恢复正常。 计划切换时间
	主备切换		
	实例参数调整		
	主机风险修复		
	SSL证书更新		
	备份模式升级		
	小版本升级	实例闪断	进入后，将产生下述影响： - 一般情况下，实例小版本升级采用热升级模式。实例或实例中涉及切换的分片将发生连接闪断及30秒以内的只读状态（用于等待数据完全同步），请在业务低峰期执行，并确保应用程序具备重连机制。 - 短暂影响该实例在DMS和DTS中的使用，操作完成后自动恢复正常。

升级类型	事件原因	影响类型	影响说明
<u>热升级</u>		小版本号间的差异	不同的小版本号（内核版本号）更新的内容有所区别，您需要关注升级后的小版本和当前小版本的差异，具体请参见 PolarDB MySQL小版本更新日志 。
	<u>代理小版本升级</u>	实例闪断	进入后，将产生下述影响： - 一般情况下，实例小版本升级采用热升级模式。实例或实例中涉及切换的分片将发生连接闪断及30秒以内的只读状态（用于等待数据完全同步），请在业务低峰期执行，并确保应用程序具备重连机制。 - 短暂影响该实例在DMS和DTS中的使用，操作完成后自动恢复正常。
		小版本号间的差异	不同的小版本号更新的内容有所区别，您需要关注升级后的小版本和当前小版本的差异。
	<u>网络升级</u>	实例闪断	进入后，将产生下述影响： - 一般情况下，实例小版本升级采用热升级模式。实例或实例中涉及切换的分片将发生连接闪断及30秒以内的只读状态（用于等待数据完全同步），请在业务低峰期执行，并确保应用程序具备重连机制。 - 短暂影响该实例在DMS和DTS中的使用，操作完成后自动恢复正常。
		VIP直连影响	部分网络升级过程中可能涉及跨可用区迁移，实例的虚拟IP（VIP）地址会发生改变，如果客户端使用VIP连接云数据库将会引起连接中断。  说明 为避免影响，您应当使用实例提供的域名形式的连接地址，同时关闭应用及其所属服务器的DNS缓存。
	<u>存储网关升级</u>	I/O 抖动	可能出现短暂的I/O抖动或SQL时延增加，影响的时间不超过3秒。
<u>冷升级</u>	大版本实例升级	实例闪断	进入后，将产生下述影响： - 实例或实例中涉及切换的分片通常会有2分钟以内的连接闪断。对于表文件较多的实例、升级前有正在执行的大事务实例或升级前CPU使用率较高的实例，冷升级过程中的闪断时间可能会超过2分钟。请在业务低峰期执行，并确保应用程序具备重连机制。 - 如遇到需要冷升级的情况，请关注版本间的差异，并根据业务情况选择合适的升级时间。 计划切换时间
	公测版本升级		

相关API

API	描述
-----	----

API	描述
<code>DescribePendingMaintenanceActions</code>	查看不同任务类型下待处理事件的数量。
<code>ModifyPendingMaintenanceAction</code>	修改待处理事件的任务切换时间。
<code>DescribePendingMaintenanceAction</code>	查询待处理事件的详情。

23.2. 查询历史事件

您可以在PolarDB控制台查看已完成的计划内的运维事件，例如数据库软件升级、硬件维护与升级。

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. 在左侧导航栏，单击事件中心 > 历史事件。



4. 在历史事件页面，选择不同的页签查看不同事件类型下具体的历史事件详情，例如集群名称和切换时间等。

24. 参数配置

24.1. 设置集群参数和节点参数

PolarDB集群创建完成后，您可以通过控制台修改集群参数和节点参数，并可以将修改后的集群参数导出为模板，同时同一地域的集群可以通过应用模板功能快速修改集群参数。本节介绍修改集群参数和节点参数，将修改后集群参数导出为模板并应用模板的方法。

背景信息

PolarDB提供节点参数可配置功能，用户可以根据实际需求对任意节点参数进行配置。

- 当集群中新增节点后，新增节点默认使用集群参数。
- 当主节点发生故障时，主节点切换为只读节点后：
 - 原主节点角色变为只读节点，原主节点参数保持不变。
 - 原只读节点角色变为主节点，原只读节点参数保持不变。
- 当主可用区故障时，备可用区的节点切换为主节点后，备可用区的节点将使用集群参数。

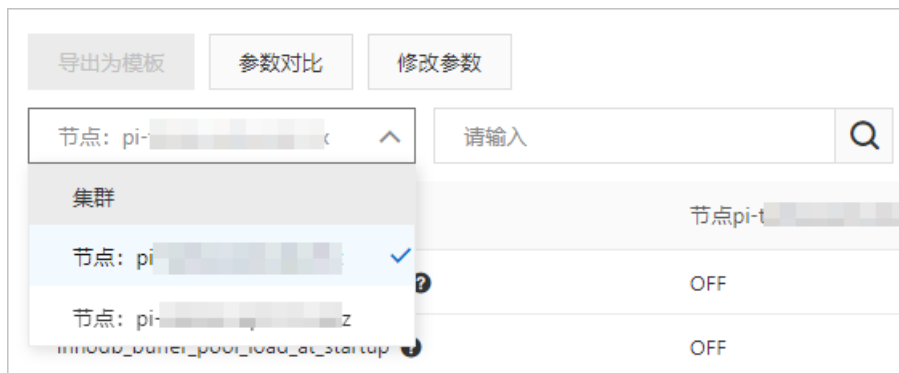
注意事项

您当前仅能修改PolarDB MySQL引擎8.0版本的节点参数 `innodb_buffer_pool_size`，其余节点参数如有修改需求请[提交工单](#)联系技术支持。

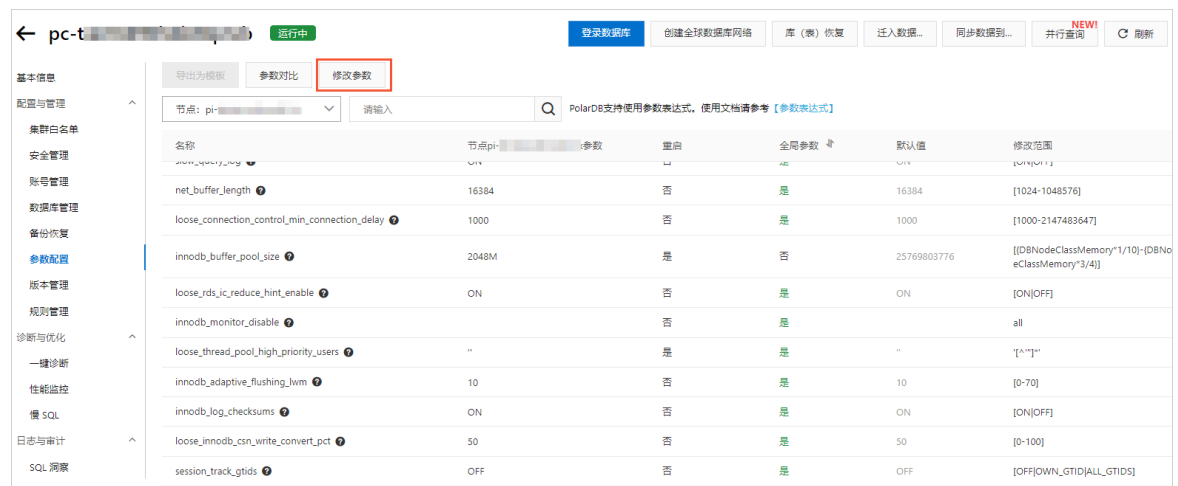
修改参数值

以下操作以修改节点参数为例进行说明。

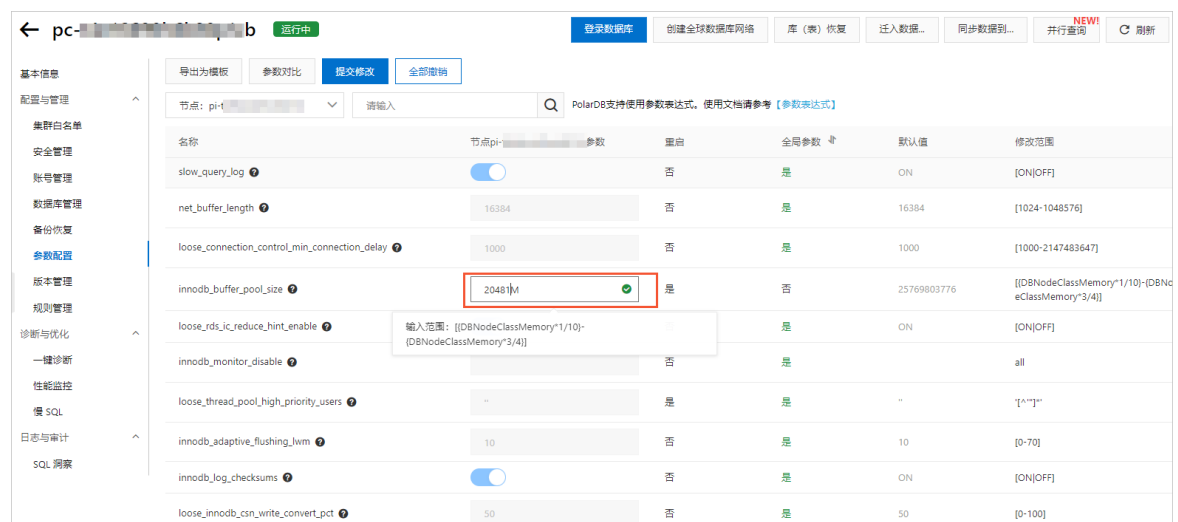
- 登录[PolarDB控制台](#)。
- 在控制台左上角，选择集群所在地域。
- 找到目标集群，单击集群ID。
- 在左侧导航栏中，选择配置与管理 > 参数配置。
- 在左上角下拉列表选择需要修改参数的节点。



- 在页面左上方，单击修改参数。



7. 找到目标参数后在集群参数列输入新的参数值。



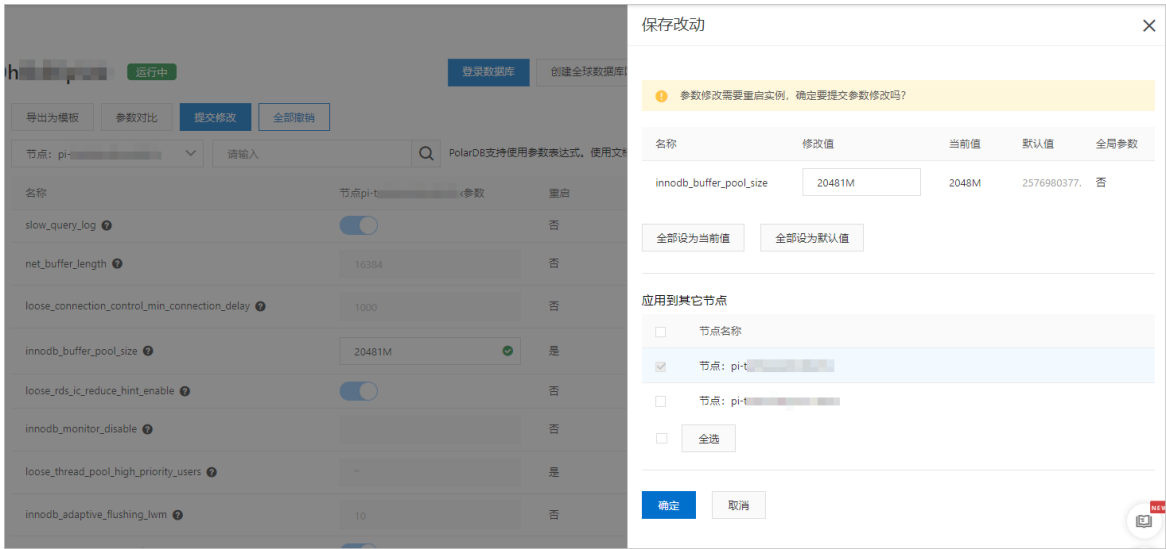
说明

- 请输入目标参数右侧修改范围栏内规定的参数值，否则当您提交修改时会出现错误提示。
- 您可以单击目标参数名称后的图标查看参数的详细说明。

8. 单击左上角提交修改，在弹出的保存改动对话框中，单击确定。



9. 在保存改动页选择需要应用到的节点，单击确定。



说明

- 对于重启栏显示为是的参数，提交修改后，需要重启集群内所有节点新参数值才会生效。重启前，请做好业务安排，谨慎操作。
- 若您的PolarDB MySQL引擎集群版本为8.0，您还可以修改参数innodb_fast_startup的值为0来加速重启。

应用参数模板

参数配置完成后，您可以进行如下操作：

- 将实例的参数配置导出为参数模板
 - i. 登录PolarDB控制台。
 - ii. 在控制台左上角，选择目标集群所在地域。
 - iii. 在左侧导航栏中，单击集群列表。
 - iv. 在集群列表页面，单击目标集群实例。
 - v. 在左侧导航栏中，单击配置与管理 > 参数配置。
 - vi. 单击导出为模板。

说明

- 如果您需要修改当前参数列表中的参数信息，请您在完成修改后，单击导出为模板。
- 导出为模板生成的集群参数由您手动修改的参数和集群在运行过程中自动修改的参数组成。

- vii. 在弹出的应用模板对话框中，设置以下参数。

参数	说明
模板名称	模板的名称。名称需要满足以下规则： - 由英文、数字、和下划线（_）组成，以字母开头，不能包含中文。 - 长度为8~64个字符。

参数	说明
描述	备注该模板的相关信息，便于后续管理。长度为0~200个字符。

viii. 单击确定。

● 将参数模板应用至目标集群实例

i. 单击应用模板。

ii. 在弹出的应用模板对话框中，选择需要使用的参数模板名称，并单击确定。

② 说明

- 您可以查看到该参数模板的参数个数、是否需要重启和更新时间。
- 如果参数模板中存在需要重启实例的参数，建议您在业务低峰期操作，并确保应用程序具有重连机制。
- 您也可以通过在参数模板页面，将参数模板应用至集群实例，详情请参见[应用参数模板](#)。

对比节点参数

由于当前不同节点的参数可以不同，您可以通过参数对比功能，对比不同的节点的参数配置情况。

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 > 参数配置。
5. 在页面左上方，单击参数对比。



6. 选择需要对比的节点，查看节点对比结果。

相关API

API	描述
DescribeDBClusterParameters	查看集群的参数。
ModifyDBClusterParameters	修改集群的参数。
ModifyDBClusterAndNodesParameters	修改集群参数并应用到指定节点。
ModifyDBNodesParameters	修改单个节点并选择应用到指定节点。

API	描述
DescribeParameterTemplates	查询默认参数列表。

24.2. 参数表达式

PolarDB新增支持将集群规格相关的参数设置为表达式的形式，当集群规格发生变更时，使用表达式设置的参数值会适应规格的变更而自动变化，使集群始终保持最优或最稳定的状态。

注意事项

- 参数值为数值类型的参数才支持使用参数表达式，参数值为字符类型的参数不支持。
例如innodb_sort_buffer_size（取值范围为65536~67108864）参数值可以使用表达式设置为 `{DBNodeClassMemory*3/4}`，但loose_recycle_bin（取值为ON或OFF）的参数值不支持使用表达式。
某些参数值也是数值类型，但该参数值并不具备数字含义，例如ssl参数的取值为0或1，其中0表示SSL加密关闭，1表示SSL加密开启，此时不建议使用表达式的方式设置该参数。
- 集群变配后，使用了参数表达式设置的参数值可能会超出该参数允许的取值范围，此时最终取值将以该参数的取值范围为准。
例如，设置了innodb_max_undo_log_size（取值范围为 `[10485760-107374182400]`）参数值为 `{DBNodeClassMemory*1/2}`，那么：
 - 当集群规格的内存升配至256 GB及以上时，`{DBNodeClassMemory*1/2}` 的计算值（`256 GB×1/2=128 GB=137438953472 Byte`）大于107374182400，此时innodb_max_undo_log_size最终取值即为107374182400。
 - 当集群规格的内存存在256 GB以下时，`{DBNodeClassMemory*1/2}` 的计算值符合该参数的取值范围，innodb_max_undo_log_size最终取值即为 `{DBNodeClassMemory*1/2}` 的计算值。

 **说明** 建议您使用参数表达式设置参数前先确认对应参数的取值范围，并要考虑到集群规格升降导致的参数值变动问题。

参数表达式支持类别

PolarDB参数表达式语法的相关支持详见下表。

支持类别	说明	示例
变量	- DBNodeClassIOPS：计算节点规格的IOPS大小，整数型。 - DBNodeClassMemory：计算节点规格的内存大小，整数型。默认单位为Byte。 - DBNodeClassCPU：计算节点规格的CPU核数，整数型。 - DBNodeClassConnections：实例规格的最大连接数，整数型。  说明 关于计算节点规格列表以及各规格的IOPS、内存、CPU核数以及最大连接数，请参见计算节点规格。	

支持类别	说明	示例 $\{DBNodeClassMemory*3/4\}$ ：表示目标参数的取值需等于当前计算节点规格内存的75%。
运算符	- 表达式语法：使用 <code>{ }</code> 包裹。 - 除法运算符 (<code>/</code>)：用被除数除以除数，返回整数型商。如果商是小数，则不四舍五入，只取整数部分。 语法： <code>dividend / divisor</code> - 乘法运算符 (<code>*</code>)：两个乘数相乘，返回整数型积。如果积是小数，则不四舍五入，只取整数部分。 语法： <code>expression * expression</code> 说明 被除数、除数和乘数均不支持小数运算。例如支持 <code>{DBNodeClassMemory*3/4}</code>，但不支持 <code>{DBNodeClassMemory*0.75}</code>。	
函数	- 函数 <code>GREATEST()</code>：返回整数型或者参数公式列表中最大的值。 语法： <code>GREATEST(argument1, argument2,...argumentn)</code> - 函数 <code>LEAST()</code>：返回整数型或者参数公式列表中最小的值。 语法： <code>LEAST(argument1, argument2,...argumentn)</code> - 函数 <code>SUM()</code>：添加指定整数型或者参数公式的值。 语法： <code>SUM(argument1, argument2,...argumentn)</code>	<code>LEAST({DBNodeClassMemory*1/2},10485760)</code> ：表示目标参数需选取当前计算节点规格内存的50%和10485760 Byte两个值中的较小值。

使用方法

使用表达式来设置参数值的操作步骤与设置集群参数的步骤一致，具体操作步骤，请参见[设置集群参数和节点参数](#)。

您只需在修改当前值时，输入对应的表达式即可。例如您可以将 `innodb_max_undo_log_size` 的当前值修改为 `{DBNodeClassMemory*1/2}`。

名称	当前值	重启	默认值
innodb_max_undo_log_size	1073741824	否	1073741824

`{DBNodeClassMemory*1/2}`

输入范围: [10485760-107374182400]

确定
取消

24.3. 使用参数模板

在配置PolarDB集群时，如果多个集群中包含一个或多个参数的配置相同，您可以使用参数模板功能，对参数进行集中管理，并快速应用至集群，以提升参数管理效率和集群配置效率。

前提条件

集群版本如下：

- MySQL 8.0
- MySQL 5.7
- MySQL 5.6

创建参数模板

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，选择**参数模板**。
4. 在**参数模板**页面，单击**创建参数模板**。
5. 设置以下参数。

参数	说明
模板名称	模板的名称。名称需要满足以下规则： ◦ 由英文、数字和下划线（_）组成，以字母开头，不能包含中文。 ◦ 长度为8~64个字符。
数据库引擎	数据库引擎类型，当前仅支持MySQL。
数据库版本	数据库引擎版本，取值范围如下： ◦ 8.0 ◦ 5.7 ◦ 5.6
描述	备注该模板的相关信息，便于后续管理。长度为0~200个字符。
添加配置	单击添加配置会增加一行参数供您选择参数名，并通过参考描述和可修改参数值设置配置值。您还可以看到是否重启集群、参数默认值等信息。  说明 ◦ 需要配置下一个参数，请再次单击添加配置。 ◦ 需要删除某个参数，请单击该参数右侧的删除。

参数	说明
导入	单击导入，输入参数及参数值。输入格式需要按照 <code>key=value</code> 格式输入，多组数据之间请换行，例如： <pre>wait_timeout=60 thread_stack=262144</pre>

6. 单击**确认创建**。

应用参数模板

② 说明

- 如果参数模板中存在需要重启集群的参数，建议您在业务低峰期操作，并确保应用程序具有重连机制。
- 参数模板创建后，您可以通过以下两种方法将参数模板应用至集群：
 - 在PolarDB集群页面应用模板，详情请参见[应用模板](#)。
 - 直接将模板应用至集群，详情见以下操作步骤。

- 登录[PolarDB控制台](#)。
- 在控制台左上角，选择集群所在地域。
- 在左侧导航栏中，选择**参数模板**。
- 在**参数模板**页面，单击**自定义参数模板**页签。
- 在自定义参数模板列表中，单击目标模板所在行**操作列**的**应用到实例**。
- 在**应用到实例**页面，选择需要应用参数模板的集群，单击  移动到右侧，同时您可以通过**参数对比**查看参数模板中参数配置值与集群中参数配置值的差异。

③ 说明 应用参数模板到多个集群上时，请注意以下问题：

- 请确认参数是否适用这些集群，请谨慎操作。
- 单次操作只能将参数模板应用到10个集群。如果需要将参数模板应用至10个以上的集群，您可以分批操作。
- 参数模板所在地域需与需要应用该模板的集群所在地域一致。如果不在同一地域，请您在集群所在地域创建参数模板，然后应用。

应用到实例

基本信息

模版名称:

test_

数据库版本:

MySQL 8.0

是否需要重启:

1

模板类型:

用户自定义

全部实例

请输入

Q

☒ pc-

☐ pc-

☐ pc-

☐ 3 项

已选实例

请输入

Q

☒ pc-

☐ 1 项

>

<

注: 最多可应用10个集群。

参数对比:

参数项	pcpg-	pc-
back_log	2000	3000
binlog_checksum	NONE	CRC32
binlog_expire_logs_seconds	100000	1209600
default_storage_engine	InnoDB	InnoDB

确定

取消

7. 单击确定。

复制参数模板

- 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，选择参数模板。
- 4. 在参数模板页面，单击自定义参数模板页签。
- 5. 在自定义参数模板列表中，单击目标模板所在行操作列的克隆。
- 6. 设置以下参数。

参数	说明
----	----

参数	说明
模板名称	模板的名称，默认为 <code>原参数模板名称_clone</code> ，您也可以重新命名。名称需满足以下规则： - 由英文、数字、和下划线（_）组成，以字母开头，不能包含中文。 - 长度为8~64个字符。
数据库引擎	当前仅支持MySQL。
数据库版本	当前支持MySQL 5.6、5.7和8.0，默认为原参数模板选择的版本，您也可以根据需要重新选择版本。
描述	备注该模板的相关信息，便于后续管理。长度为0~200个字符。
添加配置	默认为原参数模板中的参数配置项，您也可以根据需要重新进行配置。  说明 - 如果您需要添加某个参数，请单击添加配置。 - 如果您需要修改某个参数的配置值，请在该参数右侧的配置值输入框中重新输入。 - 如果您需要删除某个参数，请单击该参数右侧的删除。
导入	单击导入，输入参数及参数值。输入格式需要按照 <code>key=value</code> 格式输入，多组数据之间请换行，例如： <pre>wait_timeout=60 thread_stack=262144</pre>

7. 单击**确认创建**。

删除参数模板

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，选择**参数模板**。
4. 在参数模板页面，单击**自定义参数模板**页签。
5. 在自定义参数模板列表中，单击目标模板所在行**操作列**的**删除**。
6. 单击**确定**。

 **说明** 如果需要同时删除多个参数模板，请选中目标参数模板，然后单击**批量删除**即可。

查看参数模板差异

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，选择参数模板。
4. 在参数模板页面，单击自定义参数模板页签。
5. 在自定义参数模板列表中，选择需要对比的参数模板，单击比较差异。
6. 在比较差异页面，查看各参数模板中参数的差异。

相关API

API	描述
CreateParameterGroup	创建参数模板。
DescribeParameterGroups	查询参数模板列表。
DescribeParameterGroup	查询参数模板详情。
ModifyDBClusterParameters	修改或应用参数模板。
DeleteParameterGroup	删除参数模板。

24.4. 高性能参数模板（8.0）

PolarDB MySQL引擎支持高性能参数模板功能。本章节详细介绍PolarDB MySQL引擎8.0版本支持的高性能参数模板中的参数设置情况，以及如何开启高性能参数模板和开启高性能模板前后的性能提升情况。

高性能参数模板简介

由于PolarDB的参数比较多，针对特定场景进行参数适配的学习成本比较高。为了满足客户对数据库性能的需求，PolarDB推出了高性能参数模板，您可以将高性能参数模板应用于目标集群，或者基于该模板进一步配置集群参数。在通常情况下，高性能参数模板能够提升数据库的性能。

在PolarDB MySQL引擎8.0版支持的高性能参数模板中，设置或开启的功能如下：

8.0高性能参数模板参数说明

参数	参数说明	高性能参数模板中取值	参数默认值
innodb_flush_log_at_trx_commit	设置为0后，每秒写入并刷新一次，从而提升数据库的持久性，提升性能。	0	1
loose_innodb_lock_sys_rec_partition	事务锁的分片数量。在InnoDB内，可采用分片的方式，对事务锁（包括行锁和表锁）进行管理，从而减少事务锁管理所引入的竞争开销。	64	1
loose_query_cache_type	是否开启Fast Query Cache功能。开启后，可有效提高数据库的查询性能。	ON	OFF

使用限制

PolarDB MySQL引擎8.0.1版本支持高性能参数模板，且Revision Version需为8.0.1.1.21及以上。

适用场景与潜在风险

在通常情况下，高性能参数模板能够提升数据库的性能。但应用高性能模板后，由于该模板以追求性能为目标，会导致数据库使用过程中存在一定风险：

- 数据库可能会损失一部分稳定性。例如，可能会减弱日志持久性等。
- 事务可能会损失一部分持久性。例如，当数据库出现异常crash时，未刷盘的事务修改将会丢失。

因此，若您的业务接受损失部分数据库稳定性作为代价，追求更高的数据库性能，则您可以为您的集群应用该高性能参数模板。

 **注意** 在应用高性能参数模板前，建议您先[提交工单](#)向阿里云技术团队咨询确认。

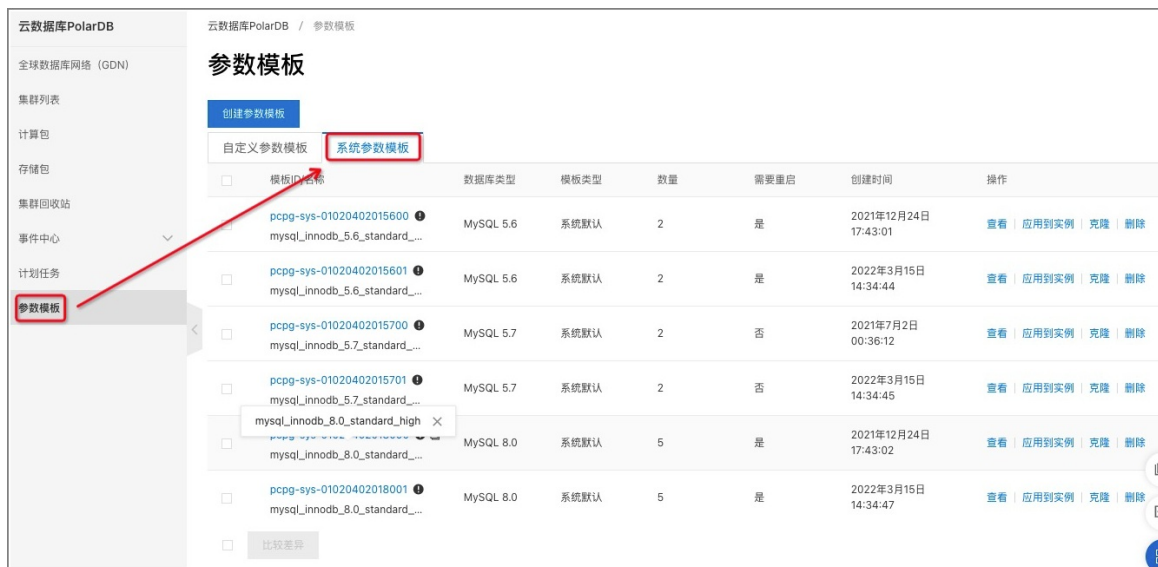
应用高性能参数模板

您可以通过以下两种方式为您的集群应用高性能参数模板。

 **警告** 在为集群应用高性能参数模板后，数据库需要重新启动。重启期间，您的服务会有闪断。因此建议您在业务低峰期操作，并确保应用程序具有重连机制。

方式一：

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**参数模板**。
4. 在参数模板页面，单击**系统参数模板**。



5. 选择名称为mysql_innodb_8.0_standard_high的模板，单击操作列的应用到实例。

参数模板

创建参数模板

自定义参数模板

系统参数模板

☐	模板ID/名称	数据库类型	模板类型	数量	需要重启	创建时间	操作
☐	pcpg-sys-01020402015600 mysql_innodb_5.6_standard_...	MySQL 5.6	系统默认	2	是	2021年12月24日 17:43:01	查看 应用到实例 克隆 删除
☐	pcpg-sys-01020402015601 mysql_innodb_5.6_standard_...	MySQL 5.6	系统默认	2	是	2022年3月15日 14:34:44	查看 应用到实例 克隆 删除
☐	pcpg-sys-01020402015700 mysql_innodb_5.7_standard_...	MySQL 5.7	系统默认	2	否	2021年7月2日 00:36:12	查看 应用到实例 克隆 删除
☐	pcpg-sys-01020402015701 mysql_innodb_5.7_standard_...	MySQL 5.7	系统默认	2	否	2022年3月15日 14:34:45	查看 应用到实例 克隆 删除
☐	mysql_innodb_8.0_standard_high mysql_innodb_8.0_standard_...	MySQL 8.0	系统默认	5	是	2021年12月24日 17:43:02	查看 应用到实例 克隆 删除
☐	pcpg-sys-01020402018001 mysql_innodb_8.0_standard_...	MySQL 8.0	系统默认	5	是	2022年3月15日 14:34:47	查看 应用到实例 克隆 删除

6. 在应用到实例页面中，选择要应用高性能参数的集群，并单击>将其添加至已选实例列表中。

应用到实例

基本信息

模板名称: mysql_innodb_8.0_standard_high (mysql_innodb_8.0_standard_high)

数据库版本: MySQL 8.0

是否需要重启: 1

模板类型: 系统默认

全部实例

请输入

☐ pc-xxxxxx

☐ pc-xxxxxx

☐ pc-xxxxxx

☐ pc-xxxxxx

☐ pc-xxxxxx

15 项

已选实例

请输入

☐ pc-xxxxxx

1 项

注: 最多可创建10个集群。

参数对比:

参数项	pcpg-sys-01020402015600(mysql_innodb_5.6_standard_high)	pc-xxxxxx(mysql_innodb_8.0_standard_high)
innodb_flush_log_at_trx_commit	0	1
loose_innodb_lock_sys_rec_partition	64	1

确定

取消

选择集群后，您可以在参数对比区域查看集群参数的当前参数值与高性能参数模板中参数值的差异。

7. 单击确定。

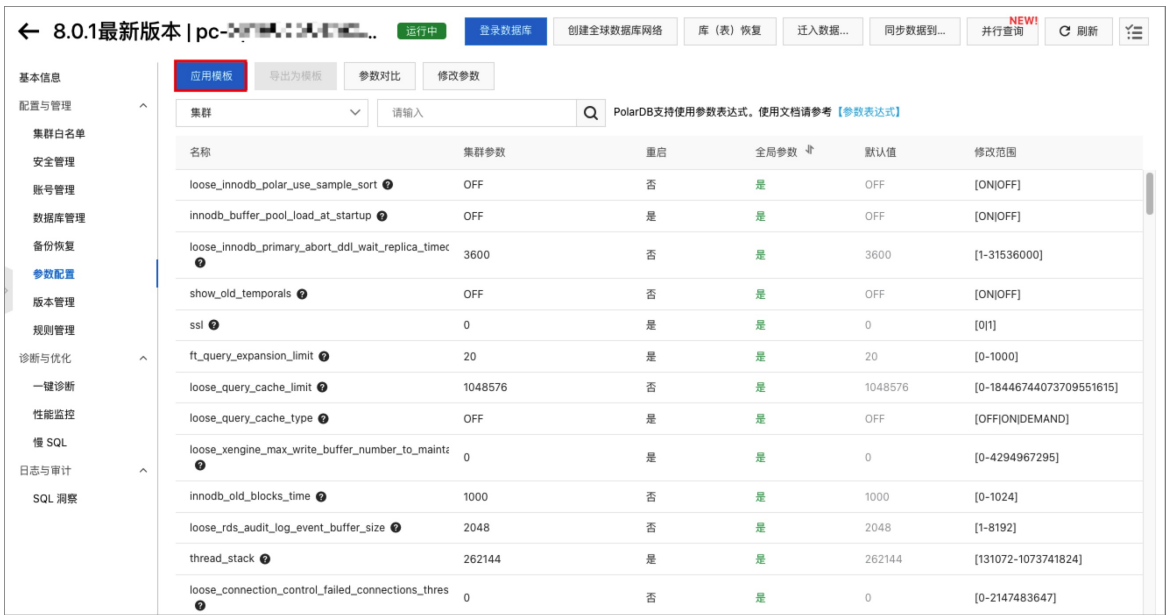
说明

为集群应用高性能参数模板后，数据库需要重新启动。

方式二：

- 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 > 参数配置。

5. 单击应用模板。



6. 在应用模板页面的模板名称下拉列表中选择mysql_innodb_8.0_standard_high。



选择模板后，您可以在参数对比区域查看集群参数的当前参数值与高性能参数模板中参数值的差异。

7. 单击确定。

说明

为集群应用高性能参数模板后，数据库需要重新启动。

性能对比

本节以PolarDB MySQL引擎8.0版本支持的高性能参数模板为例，在SysBench和TPC-C场景下，分别测试默认参数和应用高性能参数模板后的集群性能。

说明

- SysBench是一个跨平台且支持多线程的模块化基准测试工具，用于评估系统在运行高负载的数据库时相关核心参数的性能表现。
- TPC-C是业界常用的一套Benchmark，由TPC委员会制定发布，用于评测数据库的联机交易处理（偏向OLTP）能力。
- 本文中TPC-C的实现基于TPC-C的基准测试，并不能与已发布的TPC-C基准测试结果相比较，本文中的测试并不符合TPC-C基准测试的所有要求。

测试环境：

- PolarDB MySQL引擎集群：
 - 集群规格：88c720g。
 - 内核版本：8.0.1.1.21及以上。
- 压力机环境：
 - 压力机与数据库ping延迟为1ms左右。
 - 压力机的计算资源和网络资源充足。

SysBench测试过程：

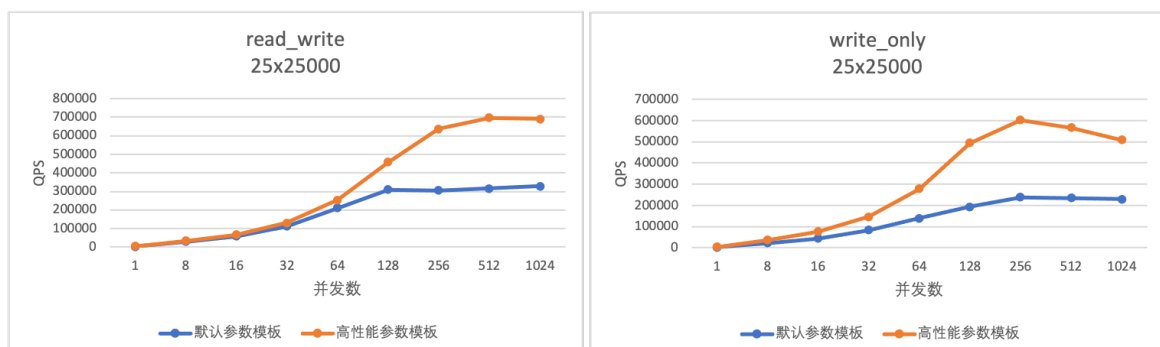
- 测试集：read_write、write_only。
- 起始数据量：25张表 × 25000行数据。
- 性能指标：分别测试在1、8、16、32、64、128、256、512、1024并发数下的QPS（Queries per Second），即数据库每秒执行的SQL数（含INSERT、SELECT、UPDATE、DELETE等）。

TPC-C测试过程：

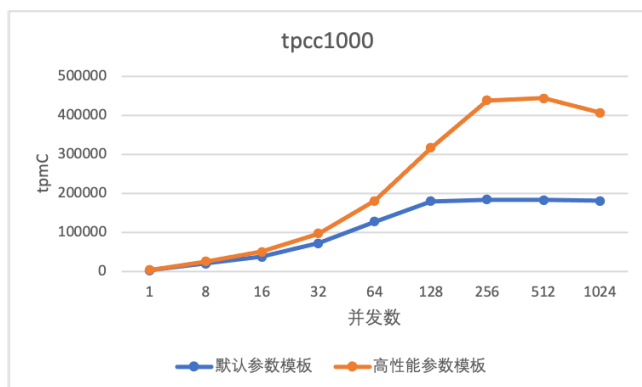
- 测试工具：TPCC-MYSQL。
- 起始数据量：1000 warehouse。
- 性能指标：分别测试在1、8、16、32、64、128、256、512、1024并发数下的tpmC（Transactions per Minute），其可以衡量数据库最大有效吞吐量（MQTh, Max Qualified Throughput）。

测试结果：

- SysBench测试：



○ TPC-C测试：



- **结论：**从以上SysBench和TPC-C的测试结果可以看出，应用了高性能参数模板后，相比默认参数值，集群的性能随着并发数的增加，有显著提升。在达到256并发数后，高性能参数模板的性能是默认参数模板的2倍甚至更高。

25.更多操作

25.1. 克隆集群

您可以根据源PolarDB集群的数据克隆一个新的PolarDB集群，本文介绍克隆PolarDB集群的详细操作步骤。

使用场景

在业务正式上线前，通常需要模拟一个和正式集群一样的环境进行测试（如压力测试），此时您可以根据源PolarDB集群克隆一个新的PolarDB集群，并在克隆集群上进行测试，从而既能确保测试的真实性，又不会影响正常业务的运行。

注意事项

- 支持被克隆的数据包括：
 - 源集群的账号信息。
 - 若源集群在克隆前已开启TDE，则相关TDE配置也支持被克隆到新集群。
- 不支持被克隆的数据包括：
 - 源集群的参数配置。
 - 源集群的白名单配置。
 - 源集群的SSL配置。
- 仅克隆集群开始创建前的数据支持被克隆，开始克隆后才写入源集群的数据将不会被克隆。

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. 找到目标集群，单击操作列的[更多 > 克隆数据到新集群](#)。
4. 在[克隆实例](#)页面，选择新集群的付费模式。
 - **包年包月**：在创建集群时支付计算节点的费用，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。
 - **按量付费**：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。

 **说明** PolarDB MySQL引擎已推出计算包。推荐购买按量付费的集群，因为按量付费的集群支持配合计算包使用，比包年包月付费方式更划算、更灵活。详情请参见[计费方式1：按量付费+计算包（推荐）](#)。

5. 设置以下参数。

参数	说明
克隆类型	默认选择克隆一个独立集群，无需修改。
源集群	需要克隆的源集群ID，无需选择。
地域	默认与源集群的地域一致，无需选择。

参数	说明
主可用区	选择克隆集群的主可用区。  说明 在有两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	固定为专有网络，无需选择。  说明 如需使用经典网络请先任选一个VPC专有网络，待集群创建完成后，再申请经典网络连接地址。申请经典网络连接地址的步骤，请参见申请集群地址和主地址。
VPC网络	选择集群所在的 VPC网络 和 VPC交换机 ，建议与原集群一致。
VPC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。
兼容性	克隆集群和源集群的兼容性默认保持一致。例如源集群的兼容性为 MySQL 8.0 ，则此处克隆集群的兼容性也为 MySQL 8.0 ，无需选择。
系列	克隆集群和源集群的系列默认保持一致。例如源集群的系列为 集群版（2-16个节点）【推荐】 ，则此处克隆集群的系列也为 集群版（2-16个节点）【推荐】 ，无需选择。
子系列	PolarDB MySQL引擎集群版支持通用规格和独享规格两种子系列，其中： 关于两种类型的详细对比，请参见如何选择通用规格和独享规格。  说明 仅当原集群系列为集群版（2-16个节点）【推荐】时，支持该配置。系列为单节点（入门级）或历史库单节点版时，不支持该配置。
节点规格	选择节点规格，不同规格有不同的最大存储容量和性能。更多详情，请参见计算节点规格。  说明 为了保障恢复后的集群运行正常，建议选择等于或高于源原集群的节点规格。
节点个数	◦ 若原集群系列为集群版（2-16个节点）【推荐】，节点个数默认为2，无需选择。  说明 集群版集群默认包含1主1只读两个节点，创建成功后最多可添加至1主15只读节点。关于如何添加节点，请参见增加或删除节点。 ◦ 若原集群系列为单节点（入门级）或历史库单节点版，节点个数默认为1，无需选择。
存储费用	您购买时无需选择容量，PolarDB会根据实际使用量按小时计费，您也可以预购存储包，如何购买存储包请参见 购买存储包 。

参数	说明
集群名称	输入集群名称，集群名称需满足如下要求： - 不能以 <code>http://</code> 或 <code>https://</code> 开头。 - 长度为2~256个字符。 如果留空，系统将为自动生成一个集群名称，创建集群后还可以修改。
购买时长	选择集群的购买时长。  说明 仅当付费模式为包年包月时支持设置该参数。
购买数量	选择集群的购买数量。

6. 阅读并选中服务协议，并根据集群的付费模式完成后续购买操作。

- 按量付费
单击立即购买即可。
- 包年包月
 - 单击立即购买。
 - 在支付页面，确认未支付订单信息和支付方式，单击订购。

 说明 购买成功后，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。

25.2. 开启Binlog

本文介绍如何为PolarDB集群开启Binlog功能。

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支持手动开启Binlog，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。

使用限制

- 若PolarDB集群创建于2019年04月05日之后，您可以直接开启Binlog；若PolarDB集群创建于2019年04月05日之前，请您先将集群小版本升级至最新版本再开启Binlog，关于如何升级小版本，请参见[版本管理](#)。
- GDN中的从集群不支持开启Binlog。

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，详情请参见[存储空间计费规则](#)。

注意事项

- Binlog功能默认关闭，开启Binlog后集群会自动重启，重启任务会在5分钟之内完成。重启时服务闪断时间在40秒左右，具体时间与数据量和表数量相关，建议您在业务低谷期进行操作并确保应用程序具备重连机制。
- 开启后Binlog默认保存2周，超出两周的Binlog文件会被自动删除，详情请参见[常见问题](#)。

- 开启Binlog后会导致写性能下降，读性能不受影响。一般情况下，开启Binlog后对性能影响不超过10%。详情请参见[常见问题](#)。
- 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。关于如何查看主地址，请参见[查看连接地址和端口](#)。
- 本文介绍的loose_polar_log_bin参数为全局（Global）级别参数。若需要使用会话（Session）级别的Binlog功能，需要开启sql_log_bin参数。

 **说明** sql_log_bin参数用于控制是否开启会话级别的Binlog。默认情况下，该参数处于关闭状态。如需开启，请[提交工单](#)联系阿里云技术支持。

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，单击集群ID。
4. 在左侧导航栏中选择[配置与管理](#) > [参数配置](#)。
5. 找到目标参数loose_polar_log_bin，单击当前值栏的图标，在弹出的对话框中，输入新的参数值，单击确定。



说明

- 若您的PolarDB MySQL引擎集群版本为MySQL 5.6，当前值修改为 *ON_WITH_GTID*；
- 若您的PolarDB MySQL引擎集群版本为MySQL 5.7或MySQL 8.0，当前值修改为 *ON*。

6. 单击左上角提交修改，在弹出的保存改动对话框中，单击确定。



说明

- 开启Binlog后集群会自动重启，重启任务会在5分钟之内完成，重启完成后新参数配置才会生效。重启时服务闪断时间在40秒左右。具体时间与数据量和表数量相关，建议您在业务低谷期进行操作并确保应用程序具备重连机制。
- 若您的PolarDB MySQL引擎集群版本为8.0和5.6，您还可以修改参数`innodb_fast_startup`的值为`ON`来加速重启。
- 如果报错提示`Cust ins minor version does not support current action`，请[提交工单](#)开启。

常见问题

● Q: Binlog文件能保存多久？

A: Binlog文件有如下2种保存策略：

- 开启Binlog后，文件默认保存2周，超出两周的Binlog文件会被自动删除。
 - 若您的集群版本为PolarDB MySQL引擎5.6，您可以通过修改`loose_expire_logs_hours`（取值范围为0~2376，单位为小时）的参数值来设置Binlog的保存时长。0表示不自动删除Binlog文件。
 - 若您的集群版本为PolarDB MySQL引擎5.7或8.0，您可以通过修改`binlog_expire_logs_seconds`（取值范围为0~4294967295，单位为秒）的参数值来设置Binlog的保存时长。0表示不自动删除Binlog文件。
 - 您通过修改`loose_expire_logs_hours`或`binlog_expire_logs_seconds`的参数值来设置Binlog的保存时长后，集群中历史Binlog文件不会被自动删除。您可以重启集群，或者当集群中最后一个Binlog文件达到`max_binlog_size`，切换到新的Binlog文件后，这些历史Binlog文件将会被清除。
- 关闭Binlog后，已有的Binlog文件会一直保留，不会自动删除。

说明

如需删除，您可以重新打开Binlog，将上述Binlog的保存时长参数（`loose_expire_logs_hours`或`binlog_expire_logs_seconds`）设置为一个较小的值，等文件超过保存时长自动删除后再关闭Binlog。

● Q: 开启Binlog后可以关闭吗？

A: 将参数`loose_polar_log_bin`修改为`OFF`并提交即可关闭。

说明

Binlog关闭后，已有的Binlog文件会一直保留。您可以先缩短Binlog文件的保存时长，等待不需要的文件自动删除后，再关闭Binlog。

● Q: 如何减少Binlog占用的存储空间？

A: 您可以将Binlog的保存时长参数（`loose_expire_logs_hours`或`binlog_expire_logs_seconds`）设置为一个较小的值，来减少Binlog文件对存储空间的占用。

● Q: 开启Binlog对性能影响有多大？

A: 开启Binlog不会影响查询（SELECT）性能，只会影响写入更新（如INSERT、UPDATE、DELETE等）性能。一般情况下，开启Binlog后对性能影响不超过10%。

● Q: 开启Binlog后集群会自动重启，重启任务多久才能完成？

A: 一般来说，重启任务会在5分钟之内完成，重启时服务闪断时间在40秒左右。具体时间与数据量和表数量相关，建议您在业务低谷期进行操作并确保应用程序具备重连机制。

● Q: 如何远程获取并查看Binlog日志内容？

A: 详情请参见[远程获取并解析PolarDB MySQL引擎Binlog日志](#)。

- Q: 为什么无法使用DMS的DDL无锁变更功能对PolarDB MySQL引擎集群进行添加索引等表结构变更操作?
A: PolarDB MySQL引擎集群Binlog默认关闭, 因此若需要使用DMS的无锁结构变更功能, 请先开启PolarDB MySQL引擎集群的Binlog功能。若不想开启Binlog, 您也可以使用online DDL命令来变更表结构。

25.3. 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

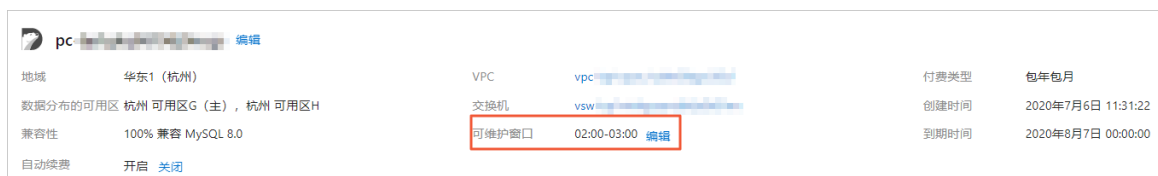
在阿里云平台上, 为保障云原生关系型数据库PolarDB的稳定性, 后端系统会不定期对集群进行维护操作, 确保集群平稳运行。您可以根据业务规律, 将可维护窗口设置在业务低峰期, 以免维护过程中对业务造成影响。

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, 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. 找到目标集群, 单击集群ID。
4. 在基本信息中的可维护窗口后, 单击编辑。



5. 在弹出的对话框中, 选中集群的可维护窗口时间, 单击确定。

说明

- 为保障云原生关系型数据库PolarDB的稳定性, 后端系统会不定期对集群进行维护操作。建议将可维护窗口设置在业务低峰期, 以免维护过程中可能对业务造成影响。
- 在可维护窗口内, 集群可能会发生1到2次连接闪断, 请确保应用程序具有自动重连机制。

相关API

API	描述
CreateDBCluster	创建数据库集群。

API	描述
ModifyDBClusterMaintainTime	修改集群可运维时间。

25.4. 重启节点

PolarDB提供了重启节点功能，您可以在控制台上手动重启节点解决数据库连接和性能问题。

注意事项

- 重启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。重启只读节点之前建立的读写分离连接不会转发请求到重启后的只读节点，您可以重启应用断开该读写分离连接并重新建立连接，请求将转发到重启后的只读节点。
- 重启节点的过程中，可能会出现1分钟以内的连接闪断。建议您在业务低谷期执行重启节点操作并确保应用具备重连机制。
- 重启节点的时间长短跟您业务的数据量有关，可能需要几个小时，请谨慎操作。

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. 找到目标集群，单击集群ID。
4. 在基本信息页的数据库节点区域，单击右上角图标切换视图。
5. 找到目标节点，单击右侧操作栏中的重启。

数据库节点							
添加节点 升降配 主备切换		自动扩容 已关闭 设置 扩容历史 					
节点名称	可用区	状态	当前角色	规格	最大IOPS	Failover 优先级	操作
 p-xxxxxx	杭州 可用区1	运行中	主节点	4核 16GB	32000	1	重启
 p-xxxxxx	杭州 可用区1	运行中	只读节点	4核 16GB	32000	1	重启

 **说明** 若您的PolarDB MySQL引擎集群版本为8.0，您还可以修改参数innodb_fast_startup的值为ON来加速重启。关于如何修改参数，请参见[设置集群参数和节点参数](#)。

6. 在弹出的对话框中，单击确定。

相关API

API	描述
RestartDBNode	重启数据库节点。

25.5. 查看或取消计划任务

您可以在执行部分运维操作（如集群升配、增加节点、升级版本和更换主可用区等）任务时，自定义任务的执行时间。任务创建成功后，您可以在控制台上查看或取消该任务。

注意事项

- 当前仅支持查看以下计划任务的详情：

- 集群升配。操作步骤请参见[操作步骤](#)。
- 增加节点。操作步骤请参见[增加只读节点](#)。
- 升级版本。操作步骤请参见[升级版本](#)。
- 更换主可用区。操作步骤请参见[更换主可用区和交换机](#)。
- 仅支持取消任务状态为等待执行中的任务。降配类（如删除节点、自动和手动降配）的计划任务不支持取消。

查看计划任务

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，单击计划任务。
4. 在计划任务页，您可以查看该地域下所有计划任务的详情，包括任务ID、任务状态、任务动作、计划起始时间、计划结束时间和计划执行时间等信息。

任务ID	集群ID	任务状态	任务动作	计划起始时间	计划结束时间	计划执行时间	订单ID	操作
20e0c...	c067e7ab	完成	UpgradeDBClusterVersion	2022年4月2日 10:00:00 (UTC+08:00)	2022年4月2日 11:00:00 (UTC+08:00)	2022年4月2日 10:00:00 (UTC+08:00)	-	取消
978c...	acc6ab	取消	RefreshProxyLevel	2022年2月12日 02:00:00 (UTC+08:00)	2022年2月12日 03:00:00 (UTC+08:00)	2022年2月12日 02:00:00 (UTC+08:00)	-	取消

说明

- 任务动作即该任务对应的API接口，当前仅支持以下任务动作：
 - ModifyDBClusterPrimaryZone（更换主可用区）
 - ModifyDBNodeClass（集群升配）
 - CreateDBNodes（增加节点）
 - UpgradeDBClusterVersion（升级版本）
- 仅当任务动作为ModifyDBNodeClass或CreateDBNodes时，支持查看订单ID信息。

取消计划任务

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，单击计划任务。
4. 在计划任务页找到目标计划任务，单击右侧操作栏中的取消。

任务ID	集群ID	任务状态	任务动作	计划起始时间	计划结束时间	计划执行时间	订单ID	操作
...	...	等待执行中	ModifyDBClusterPrimaryZone	2021年4月7日 02:00:00 (UTC+08:00)	2021年4月7日 03:00:00 (UTC+08:00)	2021年4月7日 02:00:00 (UTC+08:00)	-	取消

说明 仅支持取消任务状态为等待执行中的任务。降配类（如删除节点、自动和手动降配）的计划任务不支持取消。

5. 在弹出的对话框中，单击确定即可。

相关API

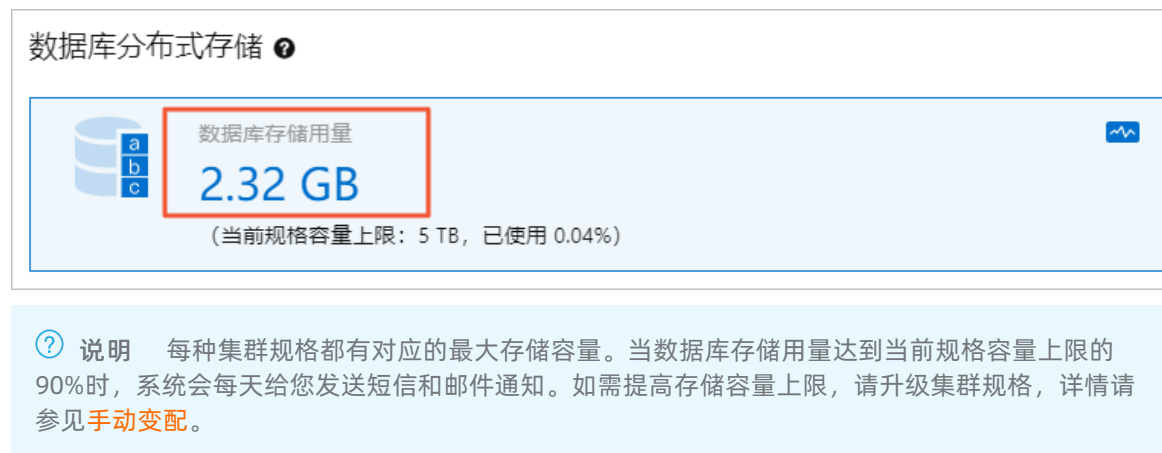
API	描述
DescribeScheduleTasks	查看当前账号下所有或指定的计划任务详情。
CancelScheduleTasks	取消目标计划任务。

25.6. 查看数据库存储用量

您可以在PolarDB控制台查看当前集群的数据库存储用量。本文将介绍如何查看数据库存储用量。

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，单击集群ID。
4. 在基本信息页面的数据库分布式存储区域，查看数据库存储用量。



25.7. 释放集群

您可以根据业务需求手动释放后付费（按小时付费）的集群，本文介绍如何手动释放PolarDB MySQL引擎集群。

注意事项

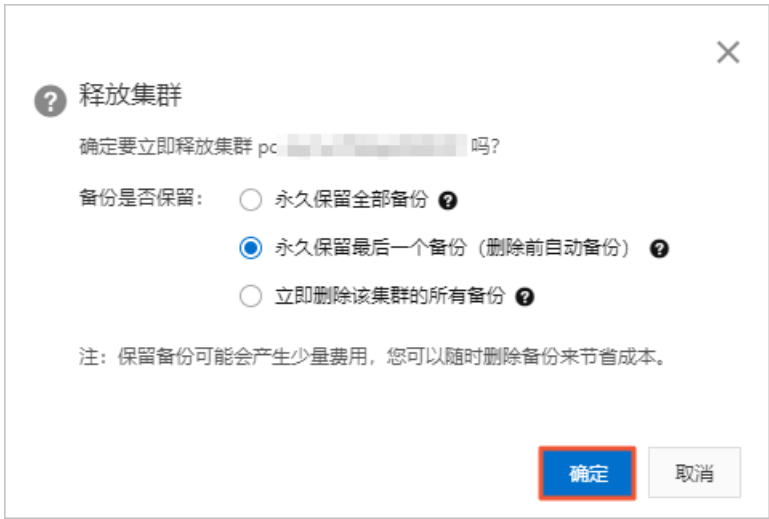
- 包年包月（也称预付费）集群不支持手动释放，集群到期后会自动被释放。
- 只有在运行状态为运行中的集群才能被手动释放。
- 本功能用于释放整个集群，包括集群中的所有节点。如要释放单个只读节点，请参见[增加或删除节点](#)。

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，单击右侧操作栏的更多 > 释放。



4. 在弹出的释放集群对话框中，选择备份保留策略后，单击确定。



备份是否保留	说明
删除集群时会自动备份，并永久保留该集群的所有备份集。	删除集群时保留所有备份。
删除集群时会自动备份，永久保留该备份集。	删除集群时保留最后一个备份。
删除集群时，立即删除该集群的所有备份集。	删除集群时不保留任何备份。  警告 选择该策略会导致集群删除后将无法恢复。

说明

- 如果您选择了删除集群时会自动备份，并永久保留该集群的所有备份集。或删除集群时会自动备份，永久保留该备份集。策略，删除PolarDB集群时，系统会主动发起一次备份，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[恢复已释放的集群](#)。

相关API

API	描述
DescribeDBClusters	查看PolarDB集群列表。
DeleteDBCluster	删除PolarDB集群。

25.8. 集群保护锁

如果您的集群承载了关键业务，建议为按量付费集群开启集群保护锁，防止手动释放按量付费集群，可以有效避免因操作疏忽、团队成员沟通不及时等原因造成不可挽回的后果。本文为您介绍如何开启或关闭集群保护锁。

前提条件

集群的付费类型为按量付费。

注意事项

- 开启集群保护锁的集群不允许转为包年包月的集群。
- 集群保护锁不能阻止因合理原因自动执行的释放行为，包括但不限于：
 - 账号欠费导致集群停机超过8天，集群被自动释放的行为。
 - 集群存在安全合规风险，被停止或释放的行为。

开启集群保护锁

- 登录[PolarDB控制台](#)。
- 在控制台左上角，选择集群所在地域。
- 您可以按照以下两种方式中的任意一种开启集群保护锁。
 - 方式一：
在[集群列表](#)页找到目标集群，单击右侧操作栏的更多 > 添加集群保护锁。



- 方式二：
 - 在[集群列表](#)页面，单击目标集群。

b. 在基本信息页面，单击**集群保护锁**右侧的**开启**。



4. 在弹出的对话框中，单击**确定**。

关闭集群保护锁

1. 登录PolarDB控制台。
2. 在控制台左上角，选择集群所在地域。
3. 您可以按照以下两种方式中的任意一种关闭集群保护锁。
 - 方式一：
在**集群列表**页找到目标集群，单击右侧操作栏的**更多 > 释放集群保护锁**。



- 方式二：
 - a. 在**集群列表**页面，单击目标集群。
 - b. 在**基本信息**页面，单击**集群保护锁**右侧的**关闭**。



4. 在弹出的对话框中，单击**确定**。

查看集群保护锁状态

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，单击目标集群。

4. 在基本信息页面，查看集群保护锁的状态。



相关API

API	描述
ModifyDBClusterDeletion	开启或关闭集群保护锁。

25.9. 标签

25.9.1. 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。
- 每个集群最多可绑定 20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。



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20个标签，单次操作绑定/解绑标签的数量分别不能超过20个。

绑定：

已有标签

键：

key

值：

value

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20个标签，单次操作绑定/解绑标签的数量分别不能超过20个。

绑定：

已有标签

新建标签

标签键

test:polardb

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
TagResources	该接口用于为PolarDB集群绑定标签。

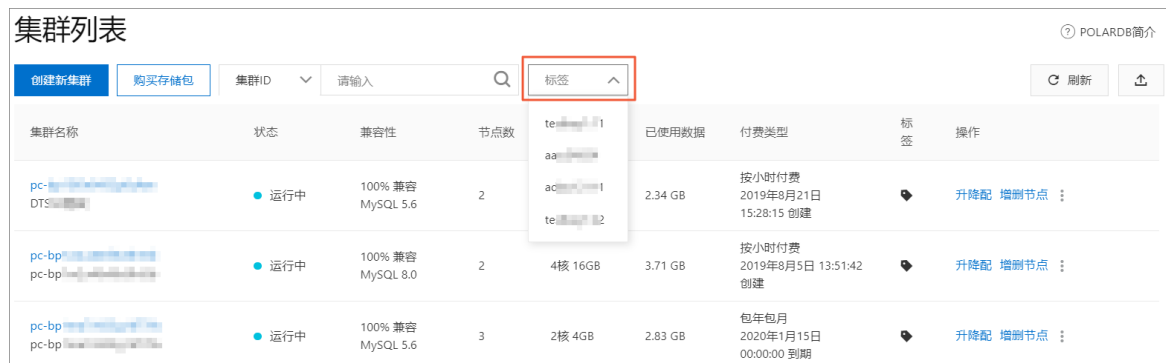
25.9.2. 根据标签筛选集群

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。

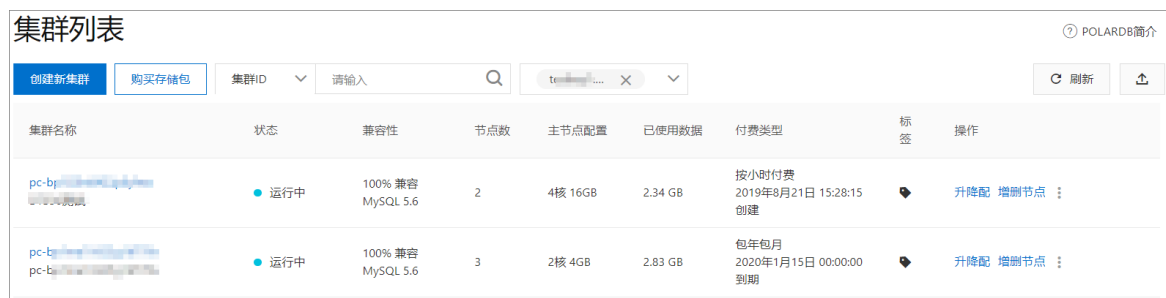
操作步骤

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

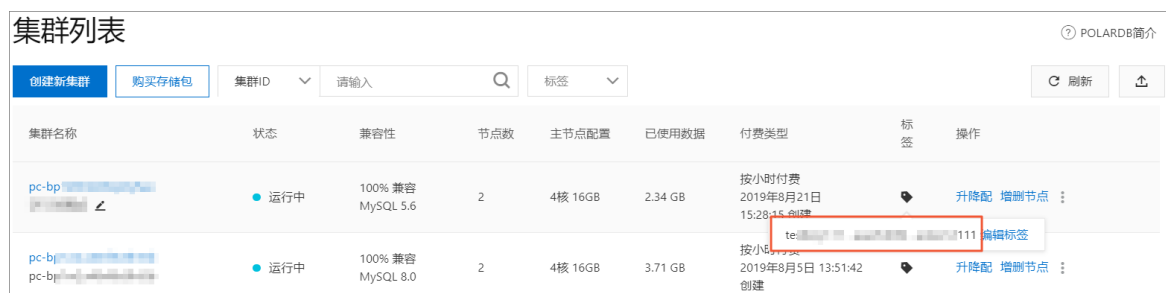
API	说明
ListTagResources	该接口用于查询一个或多个PolarDB集群已绑定的标签列表，或者查询一个或多个标签绑定的PolarDB集群列表。

25.9.3. 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. 在集群列表页面，将鼠标移动到目标集群标签栏的图标上。
4. 查看目标集群绑定的标签，如下图所示。



相关API

API	说明
ListTagResources	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。

25.9.4. 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. 在集群列表页面，将鼠标移动到目标集群标签栏的图标上，单击编辑标签。



4. 在编辑标签对话框，单击目标标签右侧的图标。



5. 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
UntagResources	该接口用于将标签从PolarDB集群上解绑。