

ALIBABA CLOUD

阿里云

图数据库
快速入门

文档版本：20220707

 阿里云

法律声明

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

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

通用约定

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

目录

1.入门概览	05
2.使用限制	07
3.RAM用户（子账号）使用GDB的准备工作	08
4.创建实例	10
4.1. 创建主实例	10
4.2. 创建只读实例	14
4.2.1. 简介	14
4.2.2. 创建只读实例	15
4.2.3. 使用只读实例	16
5.初始化实例	20
5.1. 创建账号	20
5.2. 设置白名单	20
6.连接实例	23
6.1. 通过DMS登录GDB数据库	23
6.2. 通过Gremlin Console连接实例	23
6.3. 通过Cypher Shell连接实例	25
6.4. 通过开源组件GDB Console登录图数据库	25
7.查询和分析图	27

1. 入门概览

本文旨在介绍如何购买实例、初始化设置、数据导入和连接实例，使您能够了解从购买图数据库GDB到开始使用的流程。

前提条件

- 如果您是初次使用图数据库GDB，请先了解相关使用限制。具体信息，请参见[使用限制](#)。
- 如果您需要使用RAM用户（子账号）管理图数据库GDB，需要提前对RAM用户进行授权。具体操作，请参见[授权RAM用户访问控制台](#)。
- 如果您需要将数据从OSS导入到图数据库GDB中，需要提前创建服务关联角色。具体操作，请参见[授权GDB访问OSS](#)。

操作流程

从购买实例到连接实例，您需要完成以下操作：



1. 创建图数据库GDB实例。具体操作，请参见[创建主实例](#)。

实例购买成功后，您可以在[实例列表](#)页面查看实例相关信息。通常，实例创建成功需要3~5分钟。

2. 初始化设置。

创建图数据库GDB实例后，您需要创建账号和密码、设置安全组，保证您具有图数据库GDB实例的访问权限，可以根据如下步骤进行操作。

- 创建账号和密码。具体操作，请参见[创建账号](#)。
- 设置实例访问权限，您可以通过设置白名单或者设置安全组的方式进行。具体操作，请参见[设置白名单](#)。
- 图数据库GDB支持申请外网连接地址，用于通过外网连接实例的场景。具体操作，请参见[申请外网地址](#)。

说明 如果您需要构造测试数据文件时，您可以从以下地址下载点文件和边文件。

- [下载点文件](#)。
- [下载边文件](#)。

3. 数据导入。

如果您需要进行数据导入，可以使用以下两种方式进行数据导入：

- [从阿里云对象存储OSS导入数据至图数据库GDB实例](#)。
- [使用DataWorks导入数据至图数据库GDB实例](#)。

4. 连接实例。

完成初始化设置和数据导入，您可以通过以下五种方式连接实例，进行后续查询操作。

- 通过GDB控制台直接登录数据管理服务DMS，更加方便快捷地远程访问、在线管理您的GDB数据库。具体操作，请参见[通过DMS登录GDB数据库](#)。
- 通过开源组件GDB Console可视化控制台登录图数据库，可视化界面，操作简单，并可根据业务需求对可视化界面进行二次开发。具体操作，请参见[通过开源组件GDB Console登录图数据库](#)。
- 通过Gremlin Console连接实例，命令行模式，适合Gremlin内核版本，适合用于查询语句性能优化。具体操作，请参见[通过Gremlin Console连接实例](#)。
- 通过Cypher Shell连接实例，命令行模式，适合Cypher内核版本。具体操作，请参见[通过Cypher Shell连接实例](#)。
- 通过SDK连接，支持Java、Python、.Net、Go、Node.js五种SDK。具体操作，请参见[SDK参考](#)。

2.使用限制

为保障图数据库GDB实例的稳定性及安全性，在使用图数据库GDB时，您需要注意图数据库GDB自身存在的约束。

功能	使用约束
实例部署	<p>仅支持在以下地域创建GDB实例。</p> <ul style="list-style-type: none">• 中国：华南1（深圳）、华北2（北京）、华北1（青岛）、华东2（上海）、中国（香港）、华北3（张家口）、华东1（杭州）。• 亚太：印度（孟买）、新加坡、印度尼西亚（雅加达）。• 欧洲与美洲：美国（弗吉尼亚）、美国（硅谷）。
数据备份	<ul style="list-style-type: none">• 仅支持通过图数据库控制台进行实例备份。• 备份最多可以保留730天。 <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin-top: 10px;"><p> 说明 保留天数根据您实际设置的备份保留天数确定，备份保留天支持7~730天中自定义设置。</p></div>
数据库恢复	<ul style="list-style-type: none">• 仅支持通过图数据库控制台进行数据恢复。• 公测期间，只支持实例覆盖性恢复。更多信息，请参见备份与恢复。

3.RAM用户（子账号）使用GDB的准备工作

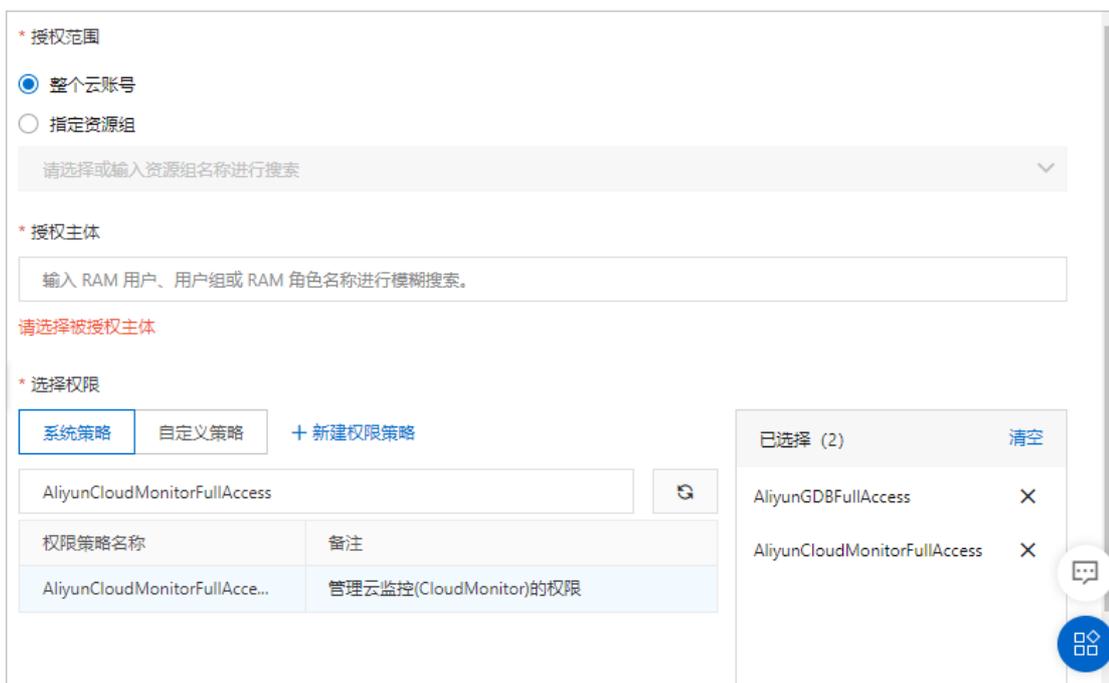
当您使用RAM用户（子账号）管理图数据库GDB时，需要先对RAM用户进行授权。本文介绍如何授权RAM用户访问图数据库GDB控制台，以及如何授权GDB导入OSS中的数据。

授权RAM用户访问控制台

 说明 请使用阿里云账号（主账号）进行以下操作。

1. 登录[访问控制控制台](#)。
2. 选择左侧导航栏中的[权限管理](#) > [授权](#)。
3. 在授权页面，单击[新增授权](#)。
4. 在[新增授权](#)页面，进行如下操作：
 - i. [授权范围](#)选择整个云账号。
 - ii. 在[授权主体](#)下方文本框中输入需要授权的RAM用户。
 - iii. [选择权限](#)，GDB相关权限及说明如下：

权限策略名称	说明
AliyunGDBFullAccess	该权限提供了GDB控制台的所有操作权限，建议优先授予该权限。
AliyunGDBReadOnlyAccess	该权限仅提供访问GDB控制台的权限。
AliyunCloudMonitorFullAccess	该权限提供了查看监控信息的权限，建议优先授予该权限。



* 授权范围

整个云账号

指定资源组

请选择或输入资源组名称进行搜索

* 授权主体

输入 RAM 用户、用户组或 RAM 角色名称进行模糊搜索。

请选择被授权主体

* 选择权限

系统策略 自定义策略 + 新建权限策略

AliyunCloudMonitorFullAccess

权限策略名称	备注
AliyunCloudMonitorFullAcce...	管理云监控(CloudMonitor)的权限

已选择 (2) 清空

- AliyunGDBFullAccess
- AliyunCloudMonitorFullAccess

5. 单击确定。
6. 单击完成。

授权GDB访问OSS

 说明 请使用阿里云账号（主账号）或拥有AliyunGDBFullAccess权限的RAM用户进行以下操作。

1. 登录图数据库GDB控制台。
2. 单击数据导入-OSS授权。
3. 系统提示关联角色创建成功，GDB可以正常访问您的OSS即可。

授权GDB访问OSS后，系统会自动创建服务关联角色AliyunServiceRoleForGDB，后续使用OSS导入数据时需要使用该角色的ARN（arn:info）信息，获取ARN信息方式如下：

- 登录访问控制控制台查看。



- RAM用户如果无法登录访问控制控制台，可以将以下示例中的 `<Aliuid>` 替换为阿里云账号UID，即为ARN信息：

```
acs:ram::<Aliuid>:role/aliyunserviceroleforgdb
```

例如阿里云账号UID为 `123456789`，那么ARN信息为 `acs:ram::123456789:role/aliyunserviceroleforgdb`。

 说明 您可以使用阿里云账号（主账号）登录阿里云控制台，将鼠标移动到右上角头像处获取阿里云账号UID。



4. 创建实例

4.1. 创建主实例

图数据库GDB是一种用于处理高度互连数据集场景（例如社交网络、实施图谱）的在线数据库服务。本文介绍创建GDB实例的方法。

前提条件

- 已注册阿里云账号。注册流程请参见[注册阿里云账号](#)。
- 如果您使用RAM账号购买GDB实例，您需要确保已拥有阿里云账号授予的AliyunGDBFullAccess与AliyunVPCFullAccess权限。授权方法，请参见[为RAM用户授权](#)。

注意事项

- 如果应用部署在ECS实例，您需要确保图数据库GDB实例和ECS实例满足以下条件，以保证网络的连通性。
 - 所在地域相同。
 - （可选）所属可用区相同，以减少网络延时。
 - ECS实例的网络类型为**专有网络**并且和GDB实例的专有网络ID相同。

说明

- 您可以通过[查看实例信息](#)获取ECS实例所在地域、可用区和专有网络ID。
- 当ECS实例的网络类型为**经典网络**时，您可以将[ECS实例从经典网络迁移到专有网络](#)。

- 仅支持在以下地域创建GDB实例。
 - 中国：华南1（深圳）、华北2（北京）、华北1（青岛）、华东2（上海）、中国（香港）、华北3（张家口）、华东1（杭州）。
 - 亚太：印度（孟买）、新加坡、印度尼西亚（雅加达）。
 - 欧洲与美洲：美国（弗吉尼亚）、美国（硅谷）。

说明

如果您需要在其他未开通（当前不支持）地域创建图数据库GDB实例，可以[提交工单](#)申请。

费用说明

图数据库GDB实例的付费类型如下，您可以根据实际业务需要选择其中一种创建实例：

- 包年包月：预付费，在创建实例时就需要支付费用。
- 按量后付费：后付费，每小时根据实例配置生成1个收费订单，并扣除相应的账户余额。

关于付费类型的更多信息，请参见[计费方式](#)。

操作步骤

1. 登录[图数据库控制台](#)。
2. 在页面左上角，选择实例所在地域。
3. 在左侧导航栏，单击实例列表。

- 在实例列表页面，单击**创建GDB实例**。
- 在商品类型区域，请根据业务需求，选择**图数据库GDB（包年包月）**或**图数据库GDB（按量后付费）**。
- 设置以下参数，完成GDB实例配置。

参数	参数说明
地域和可用区	<p>选择实例所属地域。图数据库GDB支持地域请参见注意事项。</p> <p>说明</p> <ul style="list-style-type: none"> 实例创建成功后将无法更换地域，请您谨慎选择。 如果应用部署在ECS实例，请您选择与ECS实例相同的地域，否则无法直接通过内网通信。
可用区	<p>选择实例所属可用区。</p> <p>说明</p> <ul style="list-style-type: none"> 实例创建成功后将无法更换可用区，请您谨慎选择。 如果应用部署在ECS实例，建议您选择与ECS实例相同的可用区，以减少网络延时。
系列	<p>选择实例所属的产品系列。</p> <ul style="list-style-type: none"> 高可用版：采用一主一备的经典高可用架构，主备都采用独立的图数据库节点，计算与存储分离，相对单机测试版能够提供更高的可用性。更多信息请参见高可用版。 单机测试版：仅用于测试，没有SLA保障，存在丢失数据的风险，不建议用在生产系统。更多信息请参见单节点版。
节点数	<p>选择实例的节点数。</p> <ul style="list-style-type: none"> 高可用版：固定为2，分别作为主备节点。 单机测试版：固定为1。
内核版本	<p>选择实例的内核版本。</p> <ul style="list-style-type: none"> Gremlin OpenCypher 自动机器学习 <p>更多内核版本信息，请参见内核版本。</p>
存储类型	<p>选择实例的存储类型。</p> <ul style="list-style-type: none"> ESSD云盘：具有比SSD云盘更高的性能，能够满足高性能需求版本或规格功能。 SSD云盘：具有高稳定性、高随机读写性能和高可靠性。 <p>更多存储类型信息，请参见存储类型</p>

参数	参数说明
网络类型	<p>固定为专有网络。</p> <p>? 说明</p> <ul style="list-style-type: none"> 专有网络VPC (Virtual Private Cloud) 是一种隔离的网络环境，安全性和性能均高于传统的经典网络。 如果应用部署在ECS实例，您需要确保ECS实例的网络类型为专有网络，以保证网络的连通性。
VPC网络	<p>选择实例所属专有网络。</p> <p>? 说明</p> <ul style="list-style-type: none"> 如果没有可以使用的专有网络，您可以在专有网络控制台创建专有网络。创建方法，请参见创建和管理专有网络。 如果应用部署在ECS实例，ECS实例和GDB实例需要使用相同的专有网络。
VPC交换机	<p>选择实例所属交换机。</p> <p>? 说明</p> <ul style="list-style-type: none"> 如果没有可以使用的交换机，您可以在专有网络控制台创建交换机。创建方法，请参见创建和管理交换机。 如果应用部署在ECS实例，ECS实例与GDB实例需要使用同一专有网络下的同一个交换机。
规格	选择实例的CPU和内存规格。
存储空间	<p>选择实例的存储空间的大小。</p> <p>? 说明 存储空间包含您的数据文件、系统文件、日志文件空间。</p>
购买数量	选择相同配置实例的购买数量。取值范围1~50。
购买时长	<p>如果您创建的是包年包月实例，请您选择购买时长。包月可选择1~9个月，包年可选择1、2、3或5年。</p> <p>? 说明 如果您希望实例到期后自动续费，勾选到期自动续费，并确保实例在即将到期前您的账户中有足够的余额。</p>

7. 购买实例。

付费类型	操作步骤
包年包月	i. 配置完成后，单击 立即购买 。 ii. 在 确认订单 页面，查看配置详情，阅读并勾选服务协议。 iii. 单击 去支付 ，并在 支付 页面，根据提示完成支付流程。 ? 说明 您也可以单击 加入购物车 ，与其他产品一起进行结算。结算方法请参见 购物车结算 。
按量付费	i. 配置完成后，单击 立即购买 。 ii. 在 确认订单 页面，查看配置详情，阅读并勾选服务协议。 iii. 单击 立即开通 ，系统将在下一小时自动扣取对应金额。

8. 查看实例是否创建成功。

- i. 支付完成后，单击**管理控制台**，进入图数据库GDB控制台。
- ii. 在页面左上角，选择实例所在的地域。
- iii. 在左侧导航栏，单击**实例列表**。
- iv. 查看创建的实例是否在实例列表中。
 - 是，说明实例已创建，当实例状态为**运行中**时，说明实例创建成功。
 - 否，请您等待10~15分钟后刷新页面重新查看，如果实例列表中仍然没有，请联系阿里云技术支持。

购物车结算

如果您购买的是包年包月实例，在实例配置完成后，您也可以单击**加入购物车**，与其他产品一起进行结算。结算方法如下：

1. 单击页面右侧的**购物车**。
2. 在**购物车**面板，勾选您需要购买的产品，并单击**去购物车结算**。
3. 在**购物车**页面，确认您需要购买的产品选择是否正确。

? 说明 您也可以重新选择图数据库GDB实例的**购买时长**和**购买数量**。

- o 是：单击**立即购买**。
- o 否：勾选正确的购买产品，单击**立即购买**。
4. 在**确认订单**页面，阅读并勾选服务协议，并单击**确认购买**。
5. 在**支付**页面，根据提示完成支付流程。

后续步骤

实例创建完成后，您还需要执行以下操作：

- 创建账号并设置密码，创建方法请参见**创建账号**。
- 设置实例的白名单，以允许外部设备访问该实例，设置方法请参见**设置白名单**。
- （可选）如果您需要通过外网连接实例，请您先申请外网连接地址。申请方法，请参见**申请外网地址**。

4.2. 创建只读实例

4.2.1. 简介

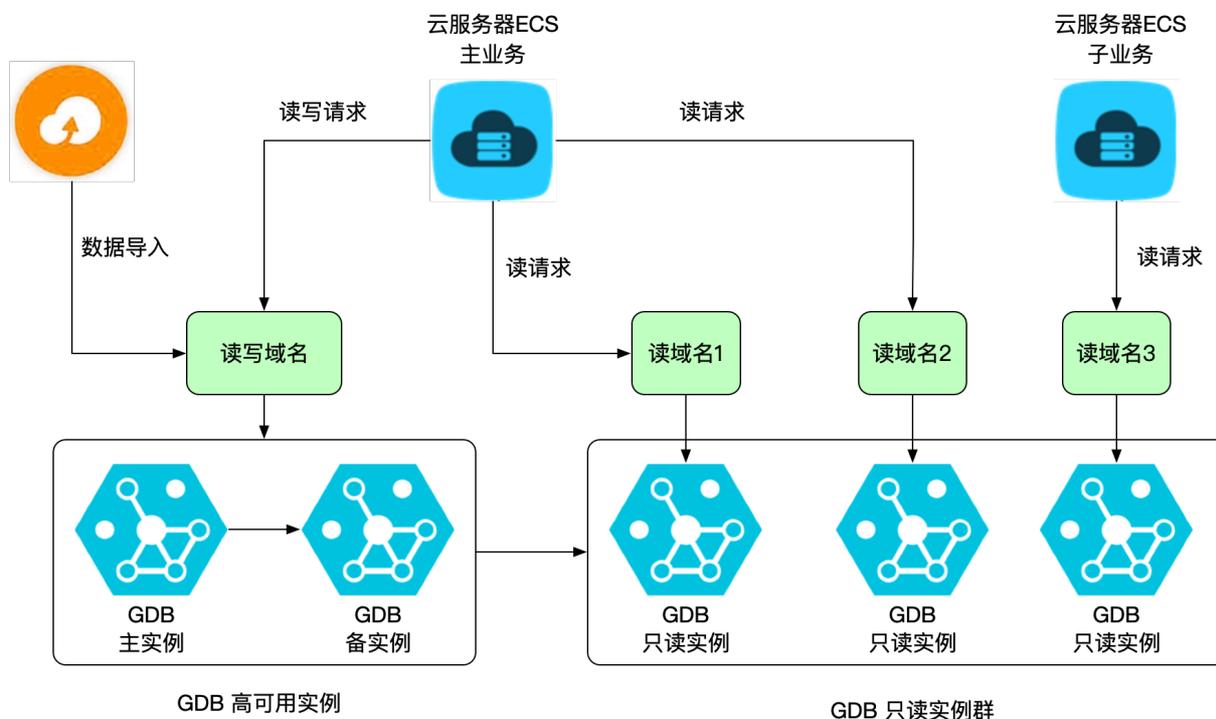
只读实例可以满足大量的数据库读取需求，增加应用的吞吐量，实现读取能力的弹性扩展，分担数据库压力。

使用场景

数据库有少量写请求，但有大量读请求。

总览

GDB创建只读实例时会利用云盘快照功能，发起最新的数据快照，数据与主实例一致，主实例的数据更新也会在主实例完成操作后立即自动同步到所有只读实例。



说明 只读实例的主实例必须是高可用实例。

常见问题

主实例上创建的账号在只读实例上可以用吗？

主实例创建的账号会同步到只读实例，只读实例无法管理账号。账号在只读实例上只能进行读操作，不能进行写操作。

只读实例能否暂停扣费？将读权重调为0可以不扣费吗？

只读实例无法暂停扣费。如果不需要使用只读实例，请及时释放实例。

4.2.2. 创建只读实例

本文介绍创建图数据库GDB只读实例的方法。

前提条件

- 主实例为高可用版，且已生成备份数据。
- 只读实例目前仅支持内核版本为Gremlin和OpenCypher的主实例。
- 如果当前实例规格为gdb.r5.* 机型系列，请先进行一次升级操作，选择相同的规格与容量，此操作不会产生费用。

使用说明

- 只读实例仅支持按量付费。
- 只读实例的账号和主实例的账号相同，无需单独创建。
- 只读实例创建时会自动复制主实例的白名单信息，但只读实例和主实例的白名单是相互独立的。
- 最多可以创建10个只读实例。
- 因为主实例已有备份，因此暂不支持备份只读实例数据。

操作步骤

1. 登录[图数据库控制台](#)。
2. 在页面左上角，选择实例所在资源组和地域。
3. 在实例列表中，单击目标实例ID或单击目标实例所在行操作列的管理。
4. 在基本信息页面的只读节点分布区域，单击右侧的增加只读。



5. 在克隆实例页面，查看当前配置并设置以下参数。

参数	参数说明
克隆源集群	固定为主实例ID。
克隆源类型	固定为备份集。
克隆源备份集	在下拉列表中选择备份集的时间。
实例类型	固定为只读。
地域和可用区	固定为主实例所在地域。
可用区	选择实例所属可用区。
系列	固定为单节点版本。
节点数	固定为1。
内核版本	与主实例相同的内核版本。

参数	参数说明
存储类型	选择实例的存储类型。更多信息请参见 存储类型 。 <ul style="list-style-type: none"> ESSD云盘：具有比SSD云盘更高的性能，能够满足高性能需求版本或规格功能。 SSD云盘：具有高稳定性、高随机读写性能和高可靠性。
网络类型	固定为专有网络。
VPC网络	与主实例相同的专有网络。
VPC交换机	与主实例相同的交换机。
规格	与主实例规格相同。 <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> <p>? 说明 实例创建成功后，可以通过变更实例配置修改实例规格（建议只读实例规格不小于主实例规格，否则易导致只读实例延迟高、负载高等现象）。</p> </div>
存储空间	与主实例存储空间相同。

6. 阅读并勾选[图数据库按量付费服务协议](#)。

7. 单击立即购买。

查看只读实例

您可以在实例列表中，单击主实例前的查看只读实例。

? **说明** 只读实例的实例名为gdr开头。



实例名称	状态	创建时间	付费类型	实例类型	内核引擎	地域	可用区	VPC ID	操作
gdr-xxxxxx	运行中	2022年03月09日 15:36:37	包年包月	高可用版	Gremlin	华东1（深圳）	可用区E	vpc-xxxxxx	管理
gdr-xxxxxx	运行中	2022年03月15日 15:59:16	按量付费	基础版	Gremlin	华东1（深圳）	可用区E	vpc-xxxxxx	管理

4.2.3. 使用只读实例

本文主要通过样例代码介绍如何使用只读实例。

注意事项

- 数据导入只能将数据导入主实例。
- 通过Gremlin DSL写数据只能将数据写入主实例。
- 查询请求可以发送给主实例或只读实例。

样例代码

样例模拟社交场景，进行新增查询、建立关系、查询关系等操作。

- 依赖包

```
<dependency>
  <groupId>org.apache.tinkerpop</groupId>
  <artifactId>gremlin-driver</artifactId>
  <version>3.4.3</version>
</dependency>
```

- 实例信息

```
主实例:
host: $master_host
只读实例1:
host: $read_host1
只读实例2:
host: $read_host2
账号信息:
username: $username
password: $password
```

- 实例链接YAML文件

- 主实例YAML文件 remote-master.yaml，用来初始化写客户端

```
# remote-master.yaml
hosts: [$master_host]
port: $port
username: $username
password: $password
connectionPool: {
  maxSize: 16,
  maxInProcessPerConnection: 2,
  maxContentLength: 81928192
}
serializer: {
  className: org.apache.tinkerpop.gremlin.driver.ser.GraphBinaryMessageSerializerV1
}
```

- 查询实例YAML文件 remote-read.yaml，可配置一批GDB实例来进行查询任务，客户端有负载均衡功能。

```
# remote-read.yaml
hosts: [$master_host,$read_host1,$read_host2]
port: $port
username: $username
password: $password
connectionPool: {
  maxSize: 48,
  maxInProcessPerConnection: 2,
  maxContentLength: 81928192
}
serializer: {
  className: org.apache.tinkerpop.gremlin.driver.ser.GraphBinaryMessageSerializerV1
}
```

- 参考代码

```
package com.alibaba.gdb.test;
```

```
import org.apache.tinkerpop.gremlin.driver.*;
import java.io.File;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
/**
 * @author xxx
 * @date 2021/01/19
 */
public class ReadInstance {
    private static Cluster masterCluster = null;
    private static Client masterClient = null;
    private static Cluster readCluster = null;
    private static Client readClient = null;
    private static final String masterYamlFile = "remote-master.yaml";
    private static final String readYamlFile = "remote-read.yaml";
    // 返回写client
    public Client getMasterClient() throws Exception {
        if (masterClient == null) {
            File file = new File(masterYamlFile);
            masterCluster = Cluster.build(file).create();
            masterClient = masterCluster.connect().init();
        }
        return masterClient;
    }
    // 返回查询client
    public Client getReadClient() throws Exception {
        if (readClient == null) {
            File file = new File(readYamlFile);
            readCluster = Cluster.build(file).create();
            readClient = readCluster.connect().init();
        }
        return readClient;
    }
    // 新写入一个用户
    public boolean addUser(String userId, String name, int age, double height, String city) throws Exception {
        String dsl = "g.addV('user').property(id, userId).property('name', name).property('age', age).property('height', height).property('city', city)";
        Map<String, Object> map = new HashMap<>();
        map.put("userId", userId);
        map.put("name", name);
        map.put("age", age);
        map.put("height", height);
        map.put("city", city);
        List<Result> resultList = getMasterClient().submit(dsl, map).all().join();
        return true;
    }
    // 新建立两个用户的关系
    public boolean addRelationship(String userId1, String userId2, long timeStamp) throws Exception {
        String dsl = "g.addE('friend').from(V(userId1)).to(V(userId2)).property('timeStamp', timeStamp)";
        Map<String, Object> map = new HashMap<>();
    }
```

```
map.put("userId1", userId1);
map.put("userId2", userId2);
map.put("timeStamp", timeStamp);
List<Result> resultList = getMasterClient().submit(dsl, map).all().join();
return true;
}
// 查询某个用户所有的好友中和自己年龄一样的好友,返回好友的信息
public List<Result> querySameAgeFriend(String userId) throws Exception {
    String dsl = "g.V().hasId(userId).as('a').both('friend').where(eq('a')).by('age')
.valueMap()";
    Map<String, Object> map = new HashMap<>();
    map.put("userId", userId);
    List<Result> resultList = getReadClient().submit(dsl, map).all().join();
    return resultList;
}
// 查询用户好友的好友, 哪些还不是自己的好友,返回他们的信息
public List<Result> query2HopFriend(String userId) throws Exception {
    String dsl = "g.V().hasId(userId).both('friend').aggregate('my_friend').both('fri
end').where(without('my_friend')).valueMap()";
    Map<String, Object> map = new HashMap<>();
    map.put("userId", userId);
    List<Result> resultList = getReadClient().submit(dsl, map).all().join();
    return resultList;
}
}
```

5. 初始化实例

5.1. 创建账号

图数据库GDB实例创建成功后，您需要先创建账号和密码，保证您具有图数据库GDB实例的访问权限。

操作步骤

1. 登录[图数据库控制台](#)。
2. 在页面左上角，选择实例所在地域。
3. 单击目标实例的实例ID或目标实例所在行操作列的管理。
4. 在左侧导航栏，单击账号管理。
5. 在用户账号页面，单击页面右上角的创建账号。
6. 在弹出的创建账号对话框中，设置以下参数。

参数	参数说明
账号名	请按照以下规则输入账号名： <ul style="list-style-type: none"> ◦ 必须以字母开头。 ◦ 由小写字母、数字或下划线（_）组成。 ◦ 长度为2~16个字符。 ◦ 不能使用保留关键字。
密码	请按照以下规则设置账号名对应的密码： <ul style="list-style-type: none"> ◦ 必须同时包含大写、小写、数字、特殊字符四项中的三项，其中特殊字符为 <code>!@#\$%^&*()_+*=</code> ◦ 长度为8~32个字符。
确认密码	请再次输入密码。
备注说明	建议输入具有业务意义的说明（无唯一性要求），便于后续识别。

7. 单击确定。

5.2. 设置白名单

图数据库GDB实例创建成功后，您需要为图数据库GDB实例设置白名单，以允许外部设备访问图数据库GDB实例。默认在白名单只包含默认的IP地址（127.0.0.1），表示任何设备均无法访问该实例。

背景信息

- 在初次使用目标实例前，需要先设置实例的白名单。添加白名单后，实例的基本信息页面和数据库连接页面才可以显示实例的连接地址。
- 正确使用白名单可以让图数据库GDB得到高级别的安全保护，建议定期维护白名单。

操作步骤

1. 登录[图数据库控制台](#)。
2. 在页面左上角，选择实例所在地域。
3. 单击目标实例的实例ID或目标实例所在行操作列的管理。
4. 在左侧导航栏，选择[数据安全性](#)。
5. 在白名单设置页面，单击修改。
6. 在弹出的设置白名单分组对话框组内白名单右侧的输入框中，填写允许访问图数据库GDB实例的IP地址。

② 说明

- 支持如下格式的IP地址或IP地址段。
 - 单个IP地址，例如10.23.12.24。
 - 0.0.0.0/0
 - CIDR，即无类域间路由。例如10.23.12.24/24，其中/24表示了地址中前缀的长度，范围[1, 32]。
- 多个IP地址或IP地址段之间以英文逗号(,)分隔。
- 设置为0.0.0.0/0或空表示对访问的IP地址来源不作任何限制，此时图数据库GDB将面临高风险，请谨慎使用。

7. 单击确定。

 **注意** 设置白名单一般需要2~3分钟生效。

添加或删除白名单分组

图数据库GDB支持添加除default以外的白名单分组，也支持删除default以外的已有白名单分组。

- 添加白名单分组

② 说明 最多支持10个白名单分组。

- i. 登录[图数据库控制台](#)。
- ii. 在左侧导航栏，选择[数据安全性](#)。
- iii. 在白名单设置页面，单击添加分组。
- iv. 在弹出的修改白名单分组对话框中，填写分组名称与组内白名单。

② 说明

- 支持如下格式的IP地址或IP地址段。
 - 单个IP地址，例如10.23.12.24。
 - 0.0.0.0/0
 - CIDR，即无类域间路由。例如10.23.12.24/24，其中/24表示了地址中前缀的长度，范围[1, 32]。
- 多个IP地址或IP地址段之间以英文逗号（,）分隔。
- 设置为0.0.0.0/0或空表示对访问的IP地址来源不作任何限制，此时图数据库GDB将面临高风险，请谨慎使用。

v. 单击**确定**。

● 删除白名单分组

② 说明 default白名单为系统默认白名单，不支持删除。

i. 登录[图数据库控制台](#)。

ii. 在左侧导航栏，选择**数据安全性**。

iii. 在白名单设置页面，单击需要删除的白名单分组右侧的**删除**。

6. 连接实例

6.1. 通过DMS登录GDB数据库

数据管理服务DMS (Data Management Service) 是一种集数据管理、结构管理、用户授权、安全审计、数据趋势、数据追踪、BI图表、性能优化和服务器管理于一体的可视化、图形化数据管理服务。通过DMS获取GDB数据库列表后，您可以更加方便快捷地远程访问、在线管理您的GDB数据库。

背景信息

实例的管控模式为自由操作或稳定变更。安全协同模式的实例无需登录，通过授权使用。详情请参见[权限管理](#)。

前提条件

已申请相应数据库权限的子账号。详情请参见[管控模式](#)。

操作步骤

1. 登录[GDB控制台](#)。
2. 在页面左上角，选择实例所在资源组和地域。
3. 在实例列表中，单击目标实例ID或单击目标实例所在行操作列的**管理**。
4. 单击页面右上角的**登录数据库**。
5. 在RDS数据库登录页面，输入数据库账号和密码，并单击**登录**。

说明

- 如果您希望浏览器记住该账号的密码，可以先勾选记住密码，然后单击登录。
- 当数据量超过64 MB时，建议您不要执行全量删除操作，例如g.V().drop()或g.E().drop()。
- 建议您在进行单次操作时，点或边的操作此处均在1000以内，避免数据达到系统上线。

6.2. 通过Gremlin Console连接实例

本文介绍如何基于Gremlin控制台连接和操作图数据库GDB。

准备工作

进行以下操作时，请确保图数据库GDB的实例与您的ECS虚拟机处于同一个VPC网络环境。

1. Gremlin控制台需要在Java8环境中使用，如未安装请执行以下命令。

```
sudo yum install java-1.8.0-devel
```

2. 如果您的ECS机器上有多个Java版本，请将Java8设置为默认运行。示例如下：

```
sudo /usr/sbin/alternatives --config java
```

共有4个提供“java”的程序。

```
+ 1      java-1.8.0-openjdk.x86_64 (/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.191.b12-1
        .e17_6.x86_64/jre/bin/java)
      2      java-1.8.0-openjdk.x86_64 (/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.191.b12-0
        .e17_5.x86_64-debug/jre/bin/java)
      3      java-1.7.0-openjdk.x86_64 (/usr/lib/jvm/java-1.7.0-openjdk-1.7.0.191-2.6.1
        5.4.e17_5.x86_64/jre/bin/java)
      4      /usr/lib/jvm/jre-1.6.0-openjdk.x86_64/bin/java
```

3. 从[Apache Tinkerpop官方网站](#)下载最新版本Gremlin控制台。

```
wget https://archive.apache.org/dist/tinkerpop/3.4.4/apache-tinkerpop-gremlin-console-3
.4.4-bin.zip
```

4. 解压并进入 `apache-tinkerpop-gremlin-console-3.4.4` 目录。

```
unzip apache-tinkerpop-gremlin-console-3.4.4-bin.zip
cd apache-tinkerpop-gremlin-console-3.4.4
```

5. 创建 `conf/gdb-remote.yaml` 文件，该文件为Gremlin控制台与GDB图数据库建立连接的配置文件。并且

- 将 `${your-gdb-endpoint}` 改为您的GDB实例的域名。
- 将 `${username}` 改为您的GDB实例的用户名。
- 将 `${password}` 改为您的GDB实例的密码。

```
hosts: [ ${your_gdb_endpoint} ]
port: 8182
username: ${username}
password: ${password}
serializer: {
  className: org.apache.tinkerpop.gremlin.driver.ser.GraphSONMessageSerializerV3d0,
  config: { serializeResultToString: true }
}
```

建立连接

1. 在Gremlin控制台主目录中执行 `./bin/gremlin.sh` 命令启动控制台。返回结果如下：

```
\,,,/
      (o o)
-----o00o- (3) -o00o-----
plugin activated: tinkerpop.server
plugin activated: tinkerpop.utilities
plugin activated: tinkerpop.tinkergraph
gremlin>
```

2. 执行以下命令连接到图数据库GDB实例。

```
:remote connect tinkerpop.server conf/gdb-remote.yaml
```

返回结果显示当前Gremlin控制台所连接的GDB实例的Host和Port：

```
==>Configured gds-uf6rge6e35594khg150010.graphdb.rds.aliyuncs.com/192.X.X.X:XXXX
```

3. 执行以下命令切换到远程模式，此后您在Gremlin控制台输入的所有DSL均将发向远端的GDB实例。

```
:remote console
```

返回结果如下：

```
==>All scripts will now be sent to Gremlin Server - [iz2ze30qqhwn3ba5mu2qgiz/172.X.X.X:XXXX]
- type ':remote console' to return to local mode
```

6.3. 通过Cypher Shell连接实例

本章节介绍如何通过Cypher Shell连接和操作图数据库GDB。

前提条件

进行实例连接操作前，请确保以下前提条件：

- 如从外网连接，请确保已经申请外网地址；如使用ECS虚拟机连接，请确保您的ECS与图数据库GDB实例处于同一个VPC网络环境。
- 请确保您的IP地址已经加入图数据库GDB的白名单中。

操作步骤

1. Cypher Shell需要Java运行环境，使用前请安装Java 8。

```
sudo yum install java-1.8.0-devel
```

2. 在[Neo4j官网](#)下载Cypher Shell。请使用4.0.5及以下版本。
3. 用如下命令启动Cypher Shell并连接到GDB。
 - 将\${your_gdb_endpoint}改为您的图数据库GDB实例地址，形式为 `域名:端口`。内网和外网的地址、端口请在[图数据库控制台](#)的基本信息页面查看。
 - 将\${username}改为您的图数据库GDB实例的用户名。
 - 将\${password}改为您的图数据库GDB实例的密码。

```
./cypher-shell -a bolt://${your_gdb_endpoint} -u ${username} -p ${password} --encryption false
```

4. 在Cypher Shell提示符后，您可以输入并测试Cypher查询语句，每条查询语句应以英文分号（;）结尾。
5. 您可以执行如下命令退出Cypher Shell。

```
:exit
```

6.4. 通过开源组件GDB Console登录图数据库

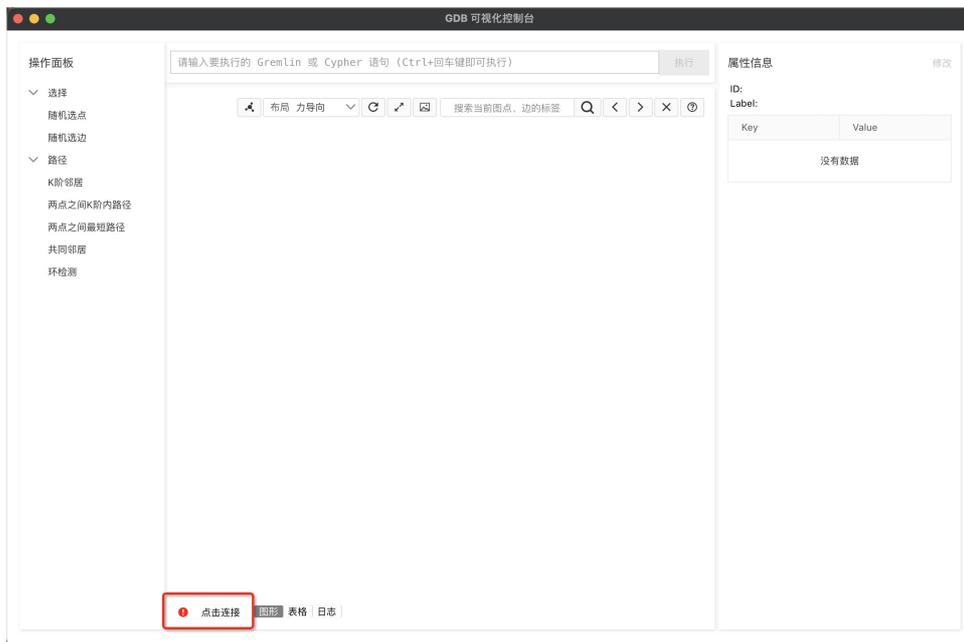
本文介绍如何基于GDB Console可视化控制台连接和操作图数据库GDB。GDB Console是阿里云图数据库GDB自有开源组件，您可以根据自有业务需求对可视化界面进行二次开发。

操作步骤

1. 下载并安装GDB Console。
 - i. 下载Release版本，可以直接安装使用。更多信息，请参见[Release版本下载](#)。

ii. 下载源码版本，可以进行二次开发。更多信息，请参见[源码下载](#)。

2. 在GDB可视化控制台页面下方，单击[点击连接](#)。



3. 输入GDB实例地址、端口、用户名和密码，单击[连接](#)。

7. 查询和分析图

连接图数据库GDB实例后，您可以通过数据管理DMS或GDB开源可视化控制台GDB Console对创建好的图数据库进行查询和分析。

目前图数据库GDB支持Gremlin和OpenCypher两种查询语言，您可以根据购买的内核版本选择不同的查询语言进行图查询和分析。

Gremlin

以下示例语句适用于内核版本为Gremlin的图数据库GDB实例。

点查询

- 统计点的数量。

```
g.V().count()
```

- 统计每种类型点的数量。

```
g.V().group().by(label).by(count())
g.V().groupCount().by(label)
```

- 查询所有点。

```
g.V().limit(100)
```

- 查询指定标签的点。

```
g.V().hasLabel('person')
```

- 根据点ID查询点。

```
g.V('123')
```

- 按属性过滤点。

```
g.V('name','张三').valueMap(true)
```

边查询

- 统计边的数量。

```
g.E().count()
```

- 统计每种类型边的数量。

```
g.E().group().by(label).by(count())
```

- 查询所有边。

```
g.E()
```

- 查询指定标签的边。

```
g.E().hasLabel('connect')
```

- 根据边ID查询边。

```
g.E('12')
```

- 查询点所关联的边。

```
g.V('123').outE('connect')
```

- 查询路径。

```
g.V('123').outE('connect').path()
```

属性查询

- 查询点的属性。

```
g.V('123').valueMap(true)
```

排序查询

- 按照属性字段进行排序。

```
g.V().has('name', '张三').outE('connect').order().by('weight', decr).valueMap(true)
```

新增点

- 新增点。

```
g.addV('person').property(id, '23').property('name', '李四')
```

删除点

- 删除指定点。

```
g.V('123').drop()
```

- 根据属性删除点。

```
g.V().has('person').property('age', values('18')).drop()
```

- 分批删除所有点。

```
g.V().drop().limit(1024)
```

新增边

- 新增边。

```
g.addEdge('connect').property('weight', '4').from(g.V('12')).to(g.V('16'))
```

删除边

- 删除边。

```
g.E('23').drop()
```

OpenCypher

以下示例语句适用于内核版本为OpenCypher的图数据库GDB实例。

点查询

- 统计点的数量。

```
MATCH (n) return count(n)
```

- 统计每种类型点的数量。

```
MATCH (n) return labels(n), count(*)
```

- 查询所有点（不建议使用）。

```
MATCH (n) return n limit 10
```

- 查询指定标签的点。

```
MATCH (n:person) return n
```

- 按属性过滤点。

```
MATCH (n:person {firstName: "Carlos"}) return n;
```

边查询

- 统计边的数量。

```
MATCH (n)-[r]->(m) return count(r)
```

- 统计每种类型边的数量。

```
MATCH (n)-[r]->(m) return type(r), count(*)
```

- 查询所有边。

```
MATCH (n)-[r]->(m) return r
```

- 查询指定标签的边。

```
MATCH (n)-[r:knows]->(m) return r
```

- 根据边属性查询边。

```
MATCH (n)-[r:knows {creationDate:1274809046847}]->(m) return r
```

- 查询点所关联的边。

```
MATCH (n:person {firstName: "Carlos"})-[r]->(m) return r
```

- 查询路径。

```
MATCH p=(n:person {firstName: "Carlos"})-[r]->(m) return p
```

属性查询

- 查询点的属性。

```
MATCH (n:person {firstName: "Carlos"}) return properties(n) limit 2;
```

排序查询

- 按照属性字段进行排序。

```
MATCH (n:person {firstName: "Mahinda"})-[r]-(m) return m order by m.locationIP
```

新增点

- 新增点。

```
CREATE (n:Person {name: 'Andy', title: 'Developer'})
```

删除点

- 删除指定点。

```
MATCH (n:Person {name: 'tom'})
DELETE n
```

- 根据属性删除点。

```
MATCH (a {name: 'Andy'})
REMOVE a.age
RETURN a
```

- 分批删除所有点。

```
MATCH (n:Person )
DELETE n
LIMIT 10
```

新增边

- 新增边。

```
MATCH
  (a:Person),
  (b:Person)
WHERE a.name = 'A' AND b.name = 'B'
CREATE (a)-[r:RELTYPE]->(b)
RETURN type(r)
```

删除边

- 删除边。

```
MATCH (n {name: 'Andy'})-[r:KNOWS]->()
DELETE rg.E('23').drop()
```