云数据库 PostgreSQL

操作指南

产品文档

腾讯云

版权所有：腾讯云计算（北京）有限责任公司

【版权声明】

©2013-2024 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

【商标声明】

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

文档目录

操作指南
- 实例管理
  - 实例生命周期
  - 设置实例维护时间
  - 调整实例配置
  - 设置实例可用区容灾或者地域容灾
  - 变更可用区设置
  - 销毁实例
  - 恢复实例
  - 下线实例
  - 重启实例
  - 修改数据复制方式
  - 切换实例主备
- 升级实例
  - 升级内核小版本
  - 升级数据库大版本
- 只读实例
  - 只读实例概述
  - 管理只读实例 RO 组
  - 剔除策略和负载均衡
- 帐号管理
  - 数据库权限概述
  - 用户与权限操作
  - 控制台操作指引
- 数据库优化
  - 慢查询分析
  - 错误日志
- 参数管理
  - 设置实例参数
  - 参数值限制
  - 参数模板
- 日志管理及分析
  - 运行日志管理
- 备份与恢复
  - 备份原理和方案

版权所有：腾讯云计算（北京）有限责任公司

- 备份数据
- 下载备份
- 克隆实例
- 自动备份设置
- 在云服务器上恢复 PostgreSQL 数据
- 删除备份
- 查看备份空间
- 设置备份下载规则
- 使用云函数转存 PostgreSQL 历史备份
- 回档至原实例

物理迁移
- 配置物理迁移任务
- 物理迁移检查项

数据库审计
- 审计服务说明
- 查看审计日志
- 修改审计服务
- 审计性能说明

插件管理
- 插件概述
- 支持插件
  - 支持插件版本概览
  - PostgreSQL 9.3 支持插件
  - PostgreSQL 9.5 支持插件
  - PostgreSQL 10 支持插件
  - PostgreSQL 11 支持插件
  - PostgreSQL 12 支持插件
  - PostgreSQL 13 支持插件
  - PostgreSQL 14 支持插件
  - PostgreSQL 15支持的插件
  - PostgreSQL 16支持的插件
- pgAgent 插件
- 跨库访问
- 位图计算 pg_roaringbitmap 插件
- 定时任务 pg_cron 插件

网络管理
- 网络管理概述
- 修改网络

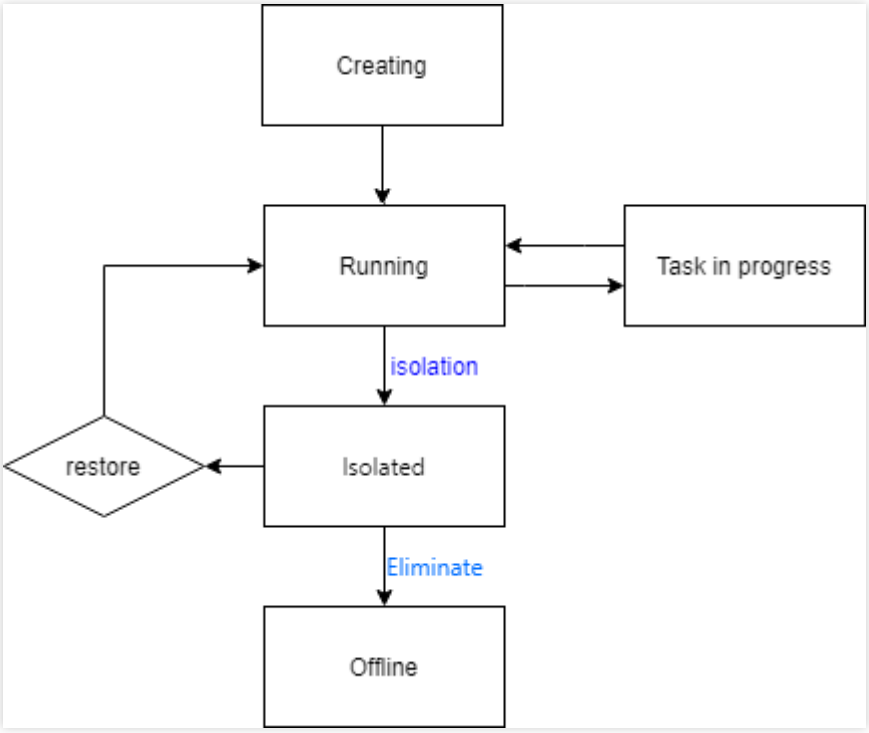
- 开启外网地址
- 访问管理
  - 访问管理概述
  - 授权策略语法
  - 可授权的资源类型
  - 控制台示例
- 数据加密
  - 透明数据加密概述
  - 开启透明数据加密
- 租户及资源隔离
  - DataBase 资源隔离
- 安全组
  - 管理安全组
  - 关联实例至安全组
- 监控与告警
  - 监控功能
  - 告警功能
- 标签
  - 标签概述
  - 编辑标签

版权所有：腾讯云计算（北京）有限责任公司

操作指南
实例管理
实例生命周期

最近更新时间：2024-01-22 16:22:57

云数据库 PostgreSQL 实例有诸多状态，不同状态下实例可执行的操作不同。本文主要是实例生命周期的相关介绍。
云数据库 PostgreSQL 包含如下实例状态，如图：



其中**创建中**状态为实例的初始状态，创建完成后的实例可以正常使用。
运行中和**任务执行中**两个状态为实例正常运行状态，任务执行中代表当前实例正在执行一些操作，如配置变更等。
当包年包月实例到期、按量计费实例的用户欠费、用户主动销毁实例这三种场景时，实例将进入隔离状态，状态将变为**已隔离**，此时可以在回收站中查看到实例。
在回收站中可对实例进行**恢复**，恢复后实例将变为运行中状态。
在回收站中的按量计费实例保存3天。到期后，实例将自动下线。下线后实例将彻底删除，无法被恢复，也无法从控制台中看到此实例。

设置实例维护时间

最近更新时间：2024-01-22 16:22:57

操作场景

维护时间对于云数据库 PostgreSQL 而言非常重要，为保证您的云数据库 PostgreSQL 实例的稳定性，后台系统会不定期在维护时间内对实例进行维护操作。建议您对业务实例设置自己可接受的维护时间，一般设置在业务低峰期，将对业务的影响降到最低。

另外，建议实例规格调整或涉及数据搬迁的操作也放置在维护时间内（目前主实例、只读实例都支持维护时间）。以数据库实例规格升级为例，实例规格升级若涉及到数据搬迁，那么在升级完成时会发生秒级数据库连接闪断。在发起升级时选择**切换时间**为**维护时间内**，实例规格切换将会在实例升级完成后的下一个**维护时间**内发起。需要注意的是，选择切换时间为**维护时间内**时，数据库规格升级完成时不会立即切换，会保持同步直到实例的**维护时间**内发起切换，因此可能会延长整个实例升级所需时间。

说明：

云数据库 PostgreSQL 在进行维护前，会向腾讯云账户内设置的联系人发送短信和邮件，请注意查收。

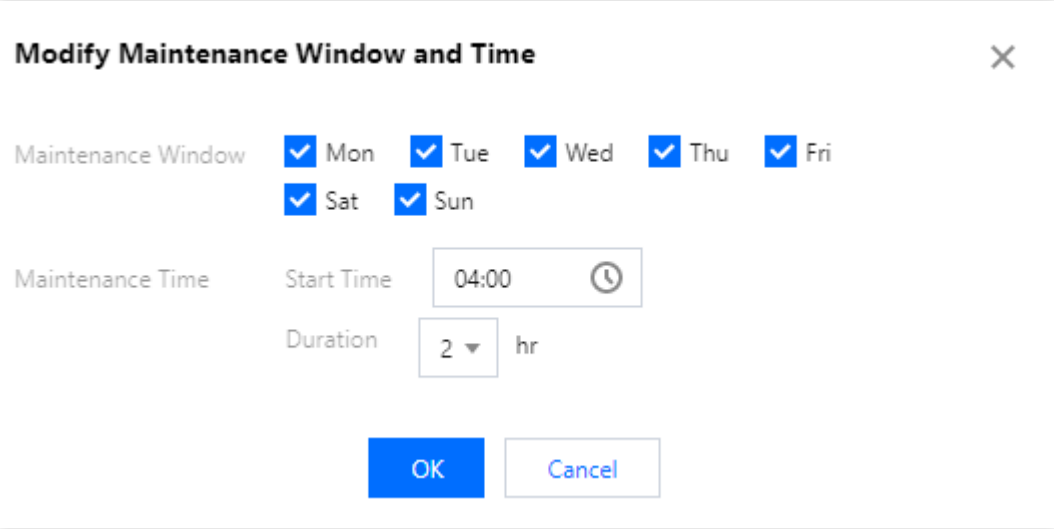
实例切换时会发生秒级数据库连接闪断，请确保业务具备重连机制。

操作步骤

设置维护时间

1. 登录 PostgreSQL 控制台，在实例列表，单击实例 ID 或**操作**列的**管理**，进入实例详情页面。
2. 在详情页的**维护信息**处，单击**修改**。

3. 在弹出的对话框，选择您所需的**维护周期**和**维护时间**，单击**确定**。



立即切换

若某任务选择在维护时间内切换，但因特殊情况需在未到维护时间内做切换，可单击**操作**列的**立即切换**。

说明：

立即切换适用于实例规格升级等涉及数据搬迁的操作。

调整实例配置

最近更新时间：2024-01-22 16:22:57

本文为您介绍如何修改云数据库 PostgreSQL 实例的计算规格与存储。

操作场景

当实例当前性能或者存储容量无法满足业务变化时，可对实例进行变配操作，以满足业务增长的述求。

注意：

目前仅支持实例升配，暂不支持实例降配。

操作步骤

1. 登录 PostgreSQL 控制台，在实例列表，找到待变配的实例，单击**操作**列的**调整配置**。
2. 在弹出的页面，选择想要变更的实例配置，与存储容量大小。完成选择之后，下方将提示具体的变配价格，确认费用之后，单击**提交**。

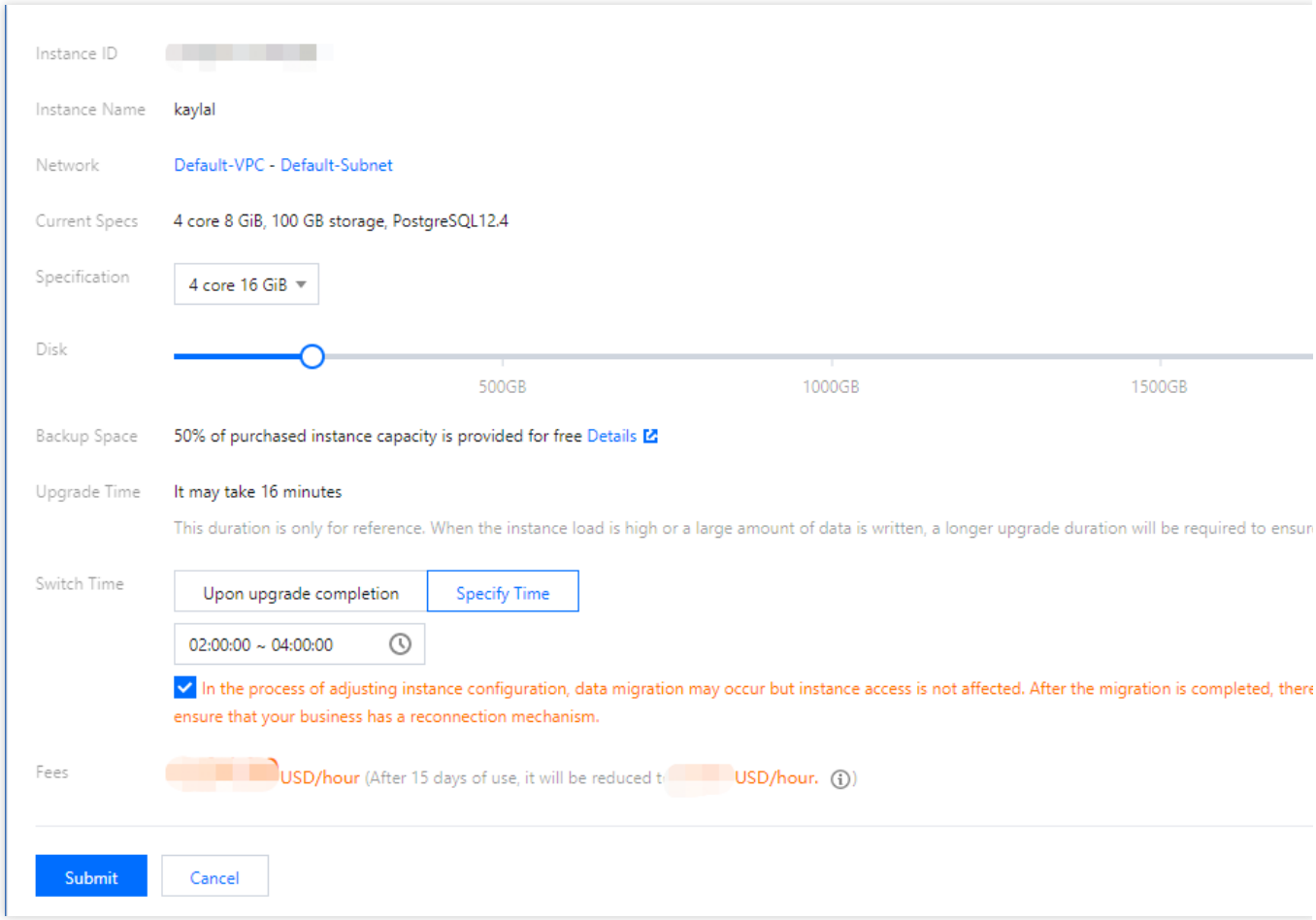
说明：

因实例变配会涉及到实例数据迁移，最后会以一次实例主从切换而完成实例的变配。实例主从切换会导致一次闪断，闪断会一定程度的影响业务对数据库实例的访问，所以一般会将实例切换时间设置在业务低峰时进行。

当需要控制切换时间时，请在**切换时间**选择**指定时间**。指定时间将选择一个时间范围，当实例数据完成搬迁后，将自动检查是否在切换时间范围内，若没有在这个时间范围内，实例处于“等待切换”状态，直到到达最近的这个时间范围中进行切换完成。

时间范围是按天为粒度进行计算，当错过当天的时间窗口。只有在第二天的此时间窗口中进行切换。

当实例处于**等待切换**状态时，可通过实例列表中**立即切换**功能使实例立即完成变配。


Instance ID
Instance Name
kaylal
Network
Default-VPC - Default-Subnet
Current Specs
4 core 8 GiB, 100 GB storage, PostgreSQL12.4
Specification
4 core 16 GiB
Disk
500GB
1000GB
1500GB
Backup Space
50% of purchased instance capacity is provided for free Details
Upgrade Time
It may take 16 minutes
This duration is only for reference. When the instance load is high or a large amount of data is written, a longer upgrade duration will be required to ensur
Switch Time
Upon upgrade completion
Specify Time
02:00:00 ~ 04:00:00
In the process of adjusting instance configuration, data migration may occur but instance access is not affected. After the migration is completed, there
ensure that your business has a reconnection mechanism.
Fees
USD/hour (After 15 days of use, it will be reduced t
USD/hour.
Submit
Cancel

设置实例可用区容灾或者地域容灾

最近更新时间：2024-07-23 11:36:03

云数据库 PostgreSQL 可以实现跨可用区容灾，同时可以配置跨地域容灾，下面分别进行说明。

设置跨可用区容灾

您可以更改云数据库 PostgreSQL 实例的主备可用区部署方式。更改后，实例的所有属性、配置和连接地址都不会改变。更改可用区所需时间跟实例的数据量有关。本文为您介绍如何通过 PostgreSQL 控制台实现更改可用区操作。

背景信息

相比单可用区部署，多可用区部署实例的方式具备更高的容灾能力，可以保护数据库，以防数据库实例发生故障或可用区中断，可以抵御机房级别的故障。
多可用区部署为数据库实例提供高可用性和故障转移支持。多可用区是在单可用区的级别上，将同一地域的多个单可用区组合成的物理区域。

注意事项

实例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。
只读实例为单节点，不支持多可用区部署方式，若其主实例可用区变更，该只读实例的所在区域不会变更。

费用

暂时不需要支付额外费用。

操作步骤

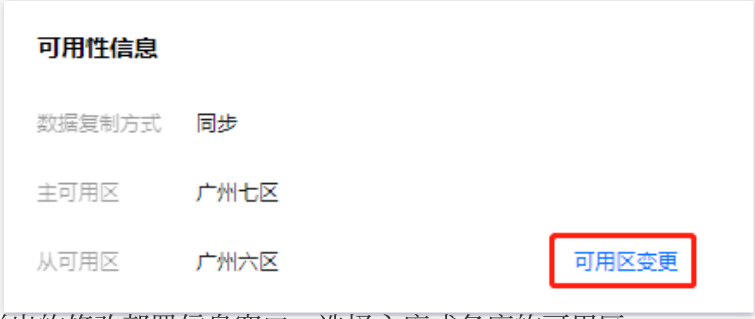
创建实例时在购买页选择可用区

1. 登录 PostgreSQL 控制台，单击**新建**。
2. 在购买页选择对应地域，并在该地域下设置**主可用区**和**备可用区**区域。

3. 购买完成后，可在**实例详情**页的**可用性信息**处查询主备可用区。

控制台更改可用区

1. 登录 PostgreSQL 控制台，选择地域，单击实例 ID 进入实例管理页。
2. 在**实例详情**页的**可用性信息**下，单击**可用区变更**。



3. 在弹出的修改部署信息窗口，选择主库或备库的可用区。

注意：

数据复制方式默认为同步复制，同步复制是数据完整性优先的复制模式，实例性能将受到日志传输效率的影响。

4. 选择切换时间，单击**确定**。

指定时间：切换时间将会在您自定义的时间范围内进行切换。

变更完成时：将在变更完成时立即切换。

说明：

变更实例的主库可用区会导致实例切换，切换过程中会有秒级闪断，请确保业务具备重连机制。变更备库的可用区不会对实例访问产生影响，请放心操作。

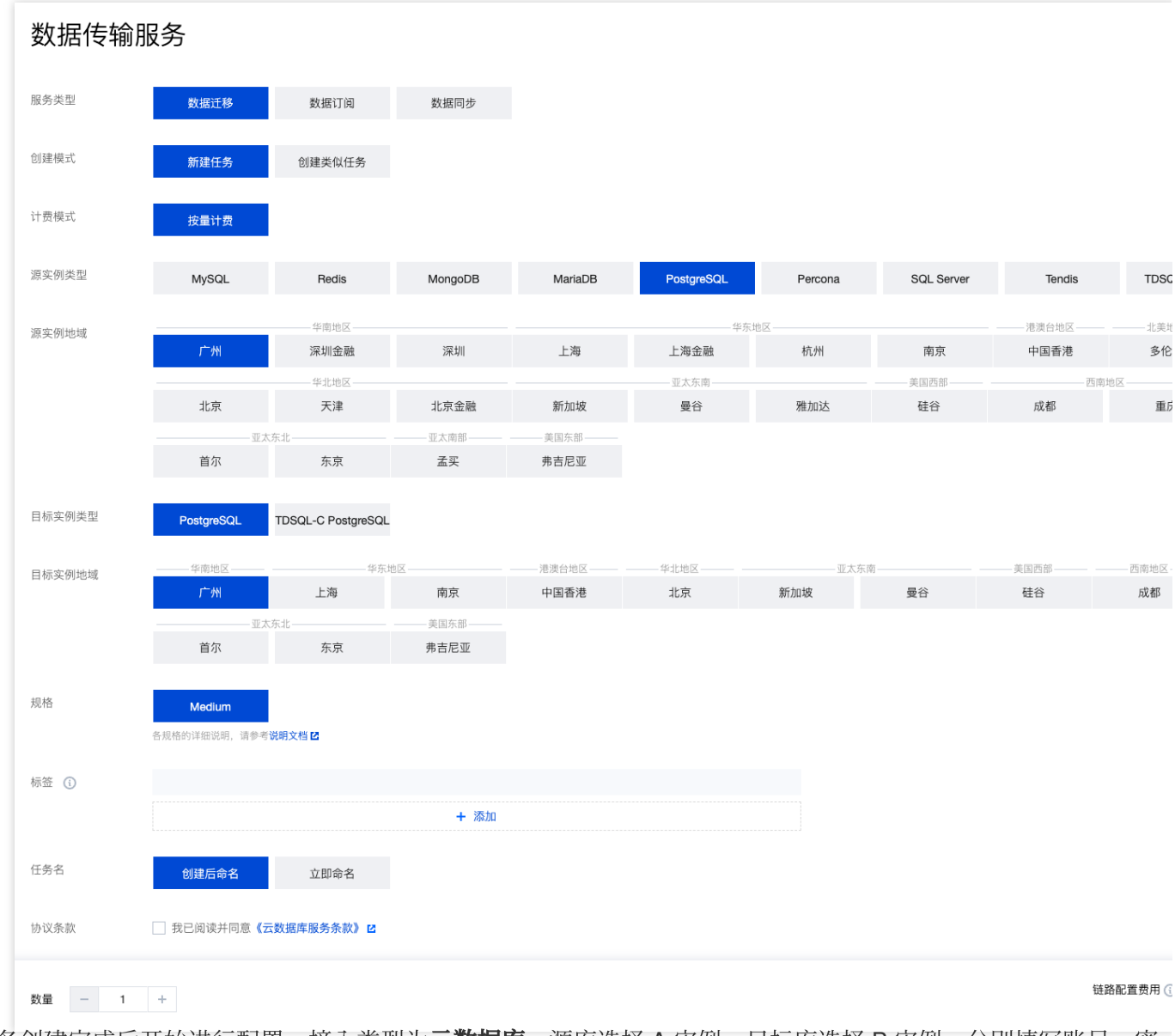
5. 待实例状态由**可用区变更中**变为**运行中**时，完成可用区更改操作。

设置跨地域容灾

操作步骤

1. 登录 PostgreSQL 控制台，单击**新建**。购买两个 PostgreSQL 实例 A 和 B。
2. 在 PostgreSQL 控制台，单击**数据迁移**，进入数据迁移控制台。

3. 在数据迁移控制台购买数据迁移任务。原端和目标端实例类型都为 PostgreSQL。



4. 任务创建完成后开始进行配置，接入类型为**云数据库**，源库选择 A 实例，目标库选择 B 实例，分别填写账号、密码进行连通性测试。

5. 设置迁移类型为‘全量 + 增量迁移’，迁移对象为‘整个实例’。

基于以上步骤能实现实例 A 到不同地域实例B的数据同步，最终实现 A 实例的跨地域容灾。

变更可用区设置

最近更新时间：2024-01-22 16:22:57

您可以更改云数据库 PostgreSQL 实例的主备可用区部署方式。更改后，实例的所有属性、配置和连接地址都不会改变。更改可用区所需时间跟实例的数据量有关。本文为您介绍如何通过 PostgreSQL 控制台实现更改可用区操作。

背景信息

相比单可用区部署，多可用区部署实例的方式具备更高的容灾能力，可以保护数据库，以防数据库实例发生故障或可用区中断，可以抵御机房级别的故障。
多可用区部署为数据库实例提供高可用性和故障转移支持。多可用区是在单可用区的级别上，将同一地域的多个单可用区组合成的物理区域。

注意事项

实例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。
只读实例为单节点，不支持多可用区部署方式，若其主实例可用区变更，该只读实例的所在区域不会变更。

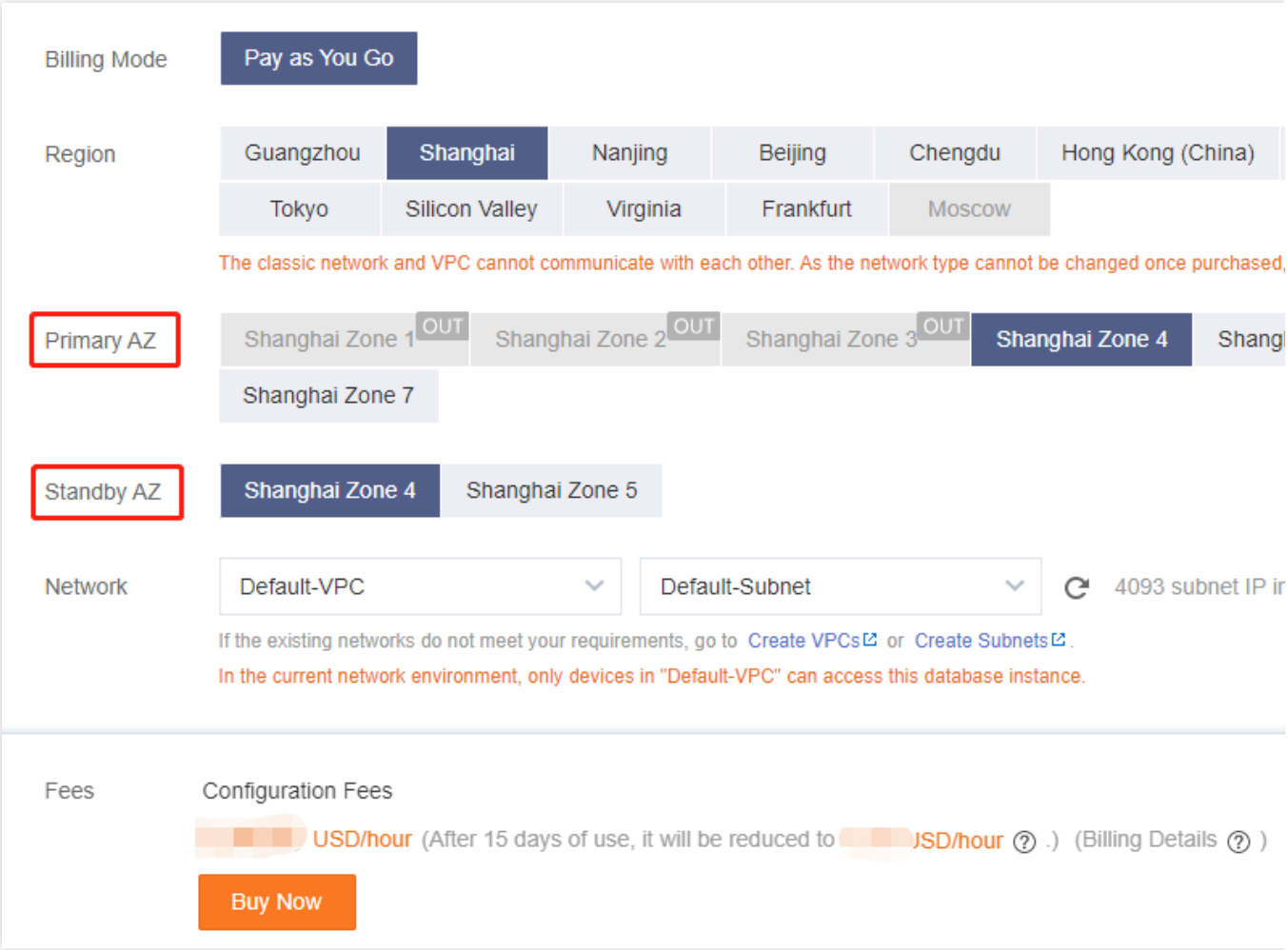
费用

暂时不需要支付额外费用。

操作步骤

创建实例时在购买页选择可用区

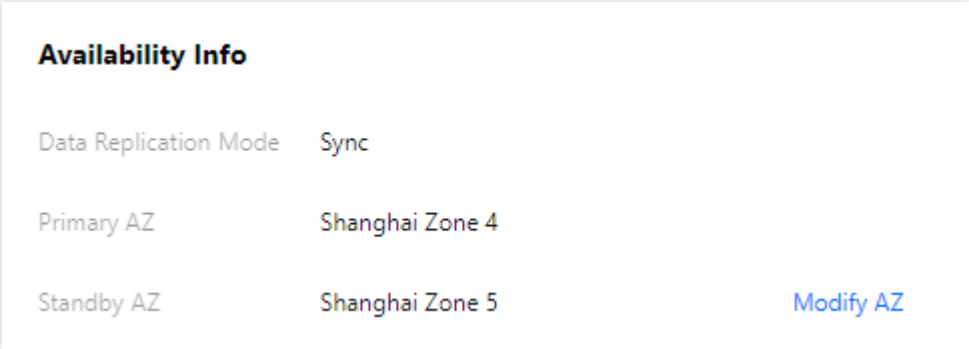
1. 登录 PostgreSQL 控制台，单击**新建**。
2. 在购买页选择对应地域，并在该地域下设置**主可用区**和**备可用区**区域。



3. 购买完成后，可在**实例详情**页的**可用性信息**处查询主备可用区。

控制台更改可用区

1. 登录 PostgreSQL 控制台，选择地域，单击实例 ID 进入实例管理页。
2. 在**实例详情**页的**可用性信息**下，单击**可用区变更**。



3. 在弹出的修改部署信息窗口，选择主库或备库的可用区。

注意：

数据复制方式默认为同步复制，同步复制是数据完整性优先的复制模式，实例性能将受到日志传输效率的影响。

4. 选择切换时间，单击**确定**。

指定时间：切换时间将会在您自定义的时间范围内进行切换。

变更完成时：将在变更完成时立即切换。

说明：

变更实例的主库可用区会导致实例切换，切换过程中会有秒级闪断，请确保业务具备重连机制。变更备库的可用区不会对实例访问产生影响，请放心操作。

5. 待实例状态由**可用区变更中**变为**运行中**时，完成可用区更改操作。

销毁实例

最近更新时间：2024-01-22 16:22:57

本文为您介绍如何通过云数据库 PostgreSQL 控制台销毁实例。

操作场景

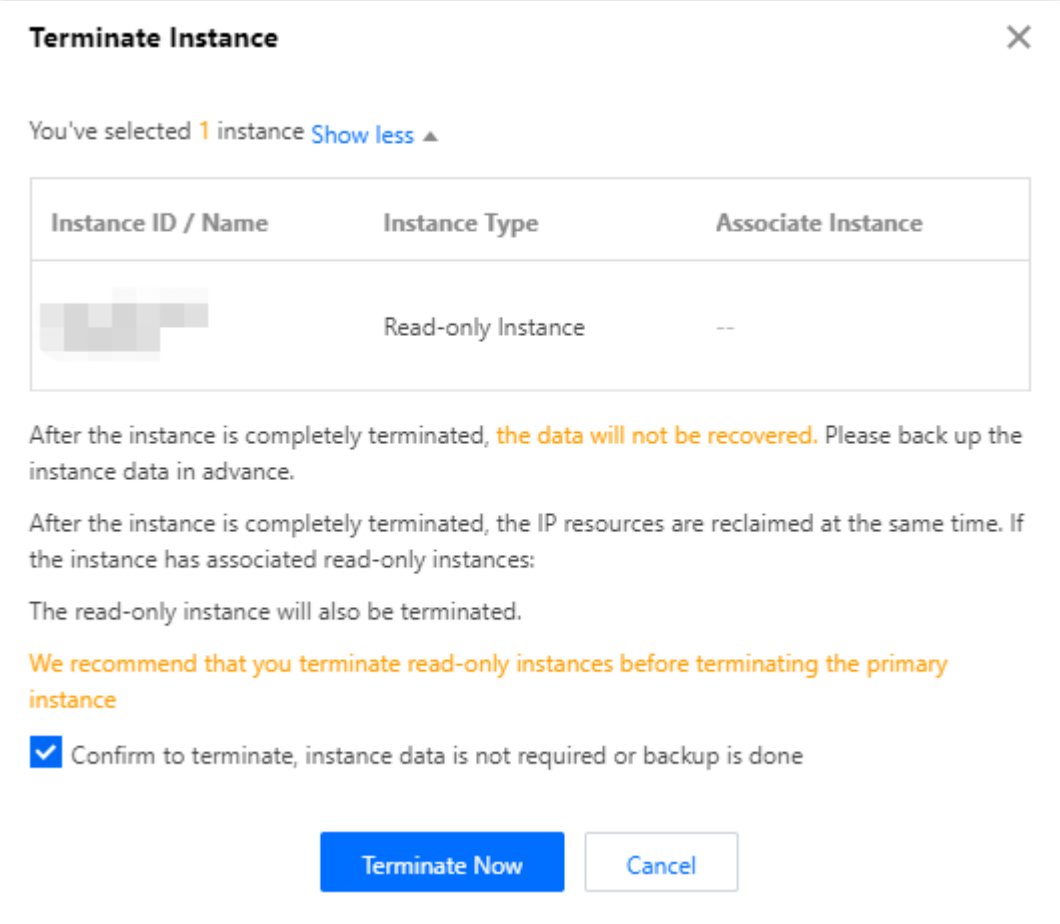
根据业务需求，您可以随时选择将实例进行销毁，销毁实例可以将实例放入回收站中。

注意事项

按量计费实例销毁后将自动进入回收站中暂存3天，实例此时状态变更为“已隔离”，实例将无法访问。
如需要恢复使用此实例，可以在回收站中对实例进行恢复操作。
如无需保留此实例，可以在回收站中对实例进行立即下线操作。
实例下线后数据将无法找回，备份文件会同步销毁，无法在云上进行数据恢复，请提前做好备份文件的转存。
当主实例存在只读实例时，对主实例进行销毁操作，对只读实例没有影响，但主实例下线会导致只读实例立即下线，若实例需要继续使用，请及时关注实例到期信息。避免因为欠费导致实例下线而无法使用。
按量计费实例销毁后，将停止计费。

操作步骤

1. 登录 PostgreSQL 控制台，在实例列表找到待销毁的实例，在**操作**列选择**更多>销毁实例**。
2. 在弹出的对话框，勾选确认后，单击**立即销毁**。


Terminate Instance
You've selected 1 instance Show less
Instance ID / Name
Instance Type
Associate Instance
Read-only Instance
--
After the instance is completely terminated, the data will not be recovered. Please back up the instance data in advance.
After the instance is completely terminated, the IP resources are reclaimed at the same time. If the instance has associated read-only instances:
The read-only instance will also be terminated.
We recommend that you terminate read-only instances before terminating the primary instance
Confirm to terminate, instance data is not required or backup is done
Terminate Now
Cancel

恢复实例

最近更新时间：2024-01-22 16:22:57

本文为您介绍如何通过云数据库 PostgreSQL 控制台恢复已隔离状态的实例。

操作场景

若因为误操作、实例到期、欠费导致的实例被销毁暂存在回收站当中，可以在回收站中将实例进行恢复使用。

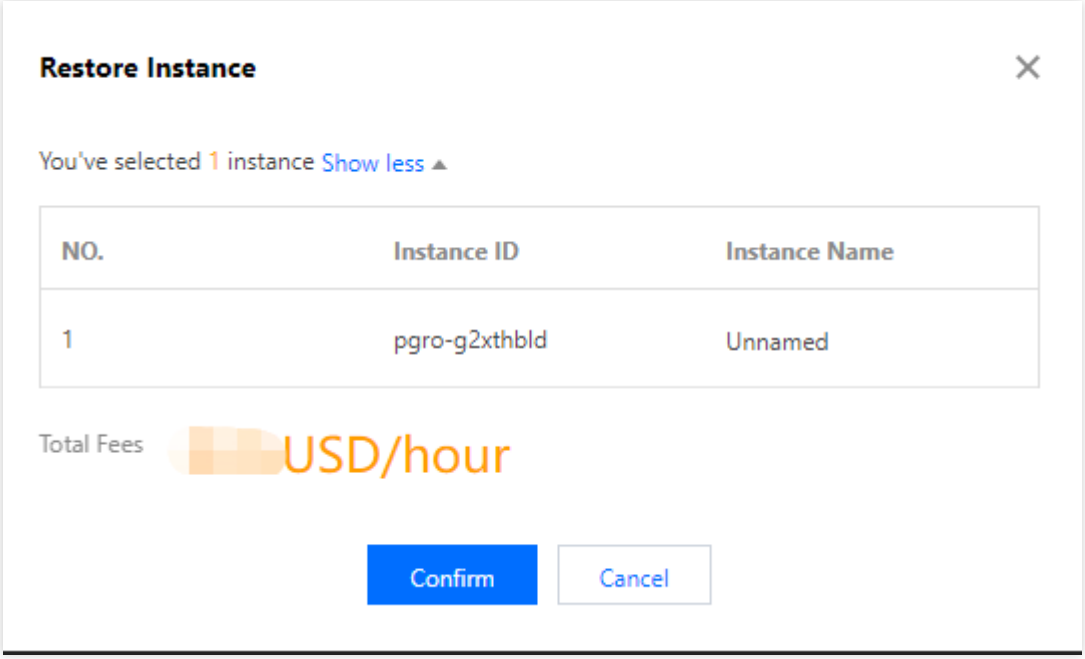
注意：

实例恢复后，所有的配置均不会发生变化。

实例短时间内无法连续被销毁，恢复。

操作步骤

1. 登录 PostgreSQL 控制台，在回收站列表找到待恢复的实例，在**操作**列选择**恢复**。
2. 在弹出的对话框，确认续费信息以及重新扣费信息后，单击**确定**。



3. 恢复完成后，实例将正常在实例列表中查看到。

下线实例

最近更新时间：2024-01-22 16:22:57

本文为您介绍如何通过云数据库 PostgreSQL 控制台立即下线已隔离状态的实例。

操作场景

当确认实例的确不需要时，您可以随时选择将实例进行立即下线。

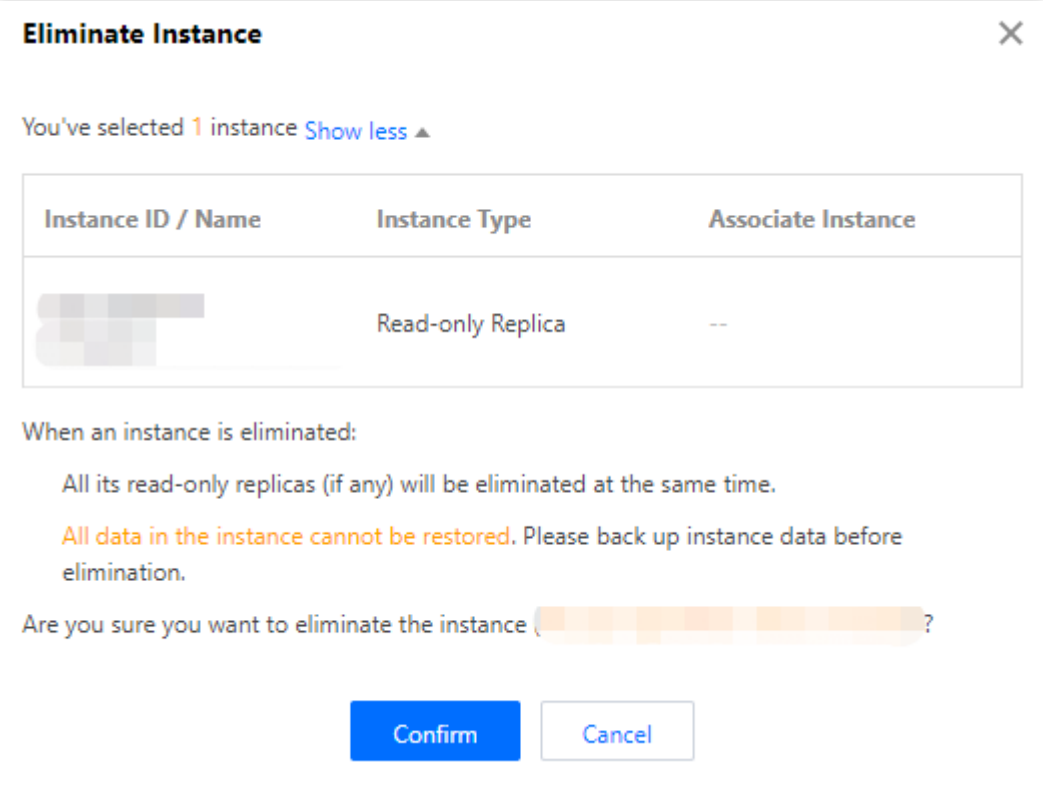
注意：

实例下线后数据将无法找回，备份文件会同步销毁，无法在云上进行数据恢复，请提前做好备份文件的转存。

当主实例存在只读实例时，且只读实例正常运行时，主实例下线会导致只读实例立即下线，若实例需要继续使用，请及时关注实例到期信息。避免因为欠费导致实例下线而无法使用。

操作步骤

1. 登录 PostgreSQL 控制台，在回收站列表找到待下线的实例，在**操作**列选择**下线**。
2. 在弹出的对话框，确认无误后，单击**确定**。



重启实例

最近更新时间：2024-01-22 16:22:57

重启操作经常是维护云数据库的一种必要手段，重启 PostgreSQL 实例相当于本地服务器的重启数据库（服务和进程）操作。

注意事项

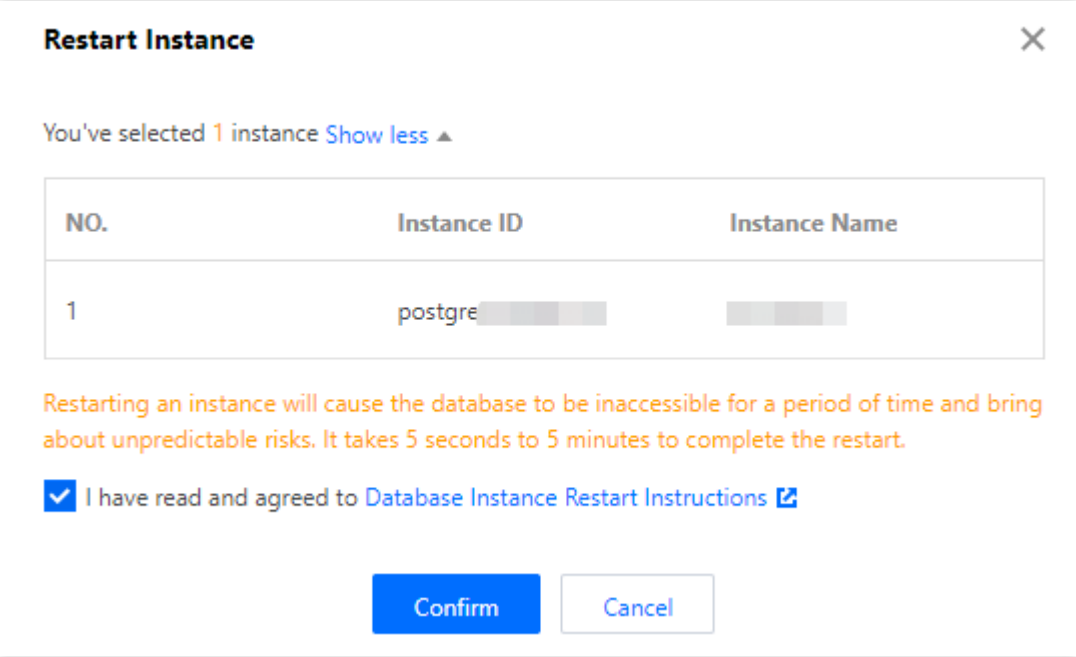
由于数据库本身对于业务的重要性，我们建议您谨慎重启，重启前尽量提前断开服务器与数据库连接，并停写数据。
重启实例不会改变实例的物理特性，因此实例的公网 IP、内网 IP、存储的任何数据都不会改变。
重启后，业务需要重连数据库，请确保您的业务有重连机制。
请确保在业务低峰期重启，保证重启成功率，降低对业务的影响。

操作步骤

1. 登录 PostgreSQL 控制台，在实例列表选择所需实例，在**操作**列选择**更多>重启**。

说明：
一般运行重启操作后需要几秒到几分钟时间，实例将无法正常访问，已有的连接会断掉，请您做好准备，以免造成影响。
重启期间，如果业务写入量过大，脏页过多，会导致重启失败。重启失败后，实例回到重启之前状态，实例仍可访问。
重启数据库有几率失败，这是正常现象，如果重启时间超过10分钟建议 提交工单 咨询。

2. 在弹出的对话框，勾选同意后，单击**确定**。


Restart Instance
You've selected 1 instance Show less
NO.
Instance ID
Instance Name
1
postgre
Restarting an instance will cause the database to be inaccessible for a period of time and bring about unpredictable risks. It takes 5 seconds to 5 minutes to complete the restart.
I have read and agreed to Database Instance Restart Instructions
Confirm
Cancel

修改数据复制方式

最近更新时间：2024-03-20 16:19:07

支持的数据复制方式

数据库实例复制是指通过为服务器配置一个或多个备份数据库的方式来进行数据同步，将 PostgreSQL 的数据分布到多个系统上去。云数据库 PostgreSQL 支持以下两种数据复制方式：

异步复制

腾讯云数据库 PostgreSQL 异步复制采用一主一备的架构。
应用发起数据更新（含 insert、update、delete 操作）请求，Master 在执行完更新操作后立即向应用程序返回响应，然后 Master 再向 Slave 复制数据。
数据更新过程中 Master 不需要等待 Slave 的响应，因此异步复制的数据库实例通常具有较高的性能（具体性能请参见 测试结果），且 Slave 不可用并不影响 Master 对外提供服务。但因数据并非实时同步到 Slave，而 Master 在 Slave 有延迟的情况下发生故障后发生切换则有较小概率会引起数据不一致。
说明：
腾讯云数据库 PostgreSQL 默认采用异步复制的数据复制方式。

半同步复制

腾讯云数据库 PostgreSQL 半同步复制采用一主一备的架构。
应用发起数据更新（含 insert、update、delete 操作）请求，Master 在执行完更新操作后立即向 Slave 复制数据，Slave **接收到数据并写到 WAL 中（无需执行）** 后才向 Master 返回成功信息，Master 必须在接受到 Slave 的成功信息后再向应用程序返回响应。
仅在数据复制发生异常（Slave 节点不可用或者数据复制所用网络发生异常）的情况下，Master 会暂停（PostgreSQL 默认10秒左右）对应用的响应，将复制方式降为异步复制。当数据复制恢复正常，将恢复为半同步复制。

退化说明

故障退化

如果当前 PostgreSQL 的主备数据复制方式为半同步，则在数据复制发生异常（Slave 节点不可用或者数据复制所用网络发生异常）时，Master 会暂停对应用的响应（云数据库 PostgreSQL 默认时10秒左右），并将主备复制方式降为异步，保证系统可用性。当高可用系统检测到数据复制恢复正常后会恢复主备复制方式为半同步复制。
注意：
故障退化为云数据库 PostgreSQL 高可用系统为默认行为，为保证系统高可用性，当前不开放设置。

延迟退化

当您有特殊需要时，可以开启半同步复制下的延迟退化，开启延迟退化之后，云数据库 PostgreSQL 高可用系统会基于您设置的条件来判断主备复制延迟，超过延迟将触发半同步退化为异步。建议只对延迟极其敏感型业务开启该能力。
延迟退化的条件为主备同步的大小或者时间，可参考监控指标为备库日志落盘延迟 (Bytes)、备库日志落盘时间延迟 (秒)，具体可参考 主备延迟监控指标。

故障切换说明

当实例的主备复制方式为**异步复制**或者**半同步降级为异步复制**时，Master 发生故障无法恢复时会触发主备切换。因数据并非实时同步到 Slave，有较小概率会引起数据不一致。腾讯云数据库 PostgreSQL 开放了故障切换条件方便您可以进行灵活设置。当前系统默认为，同时满足**主备同步延迟10240MB**和**主备延迟10秒**两个条件时允许切换。建议您有特殊业务需求时再进行设置。

修改数据复制方式

1. 登录 PostgreSQL 控制台，在实例列表，单击实例 ID 或**操作**列的**管理**，进入实例详情页面。
2. 在实例详情的**可用性信息**部分显示了详细的实例的可用性信息。
2.1 当数据复制方式为异步时，具体展示信息如下：

信息	说明
数据复制方式	主备之间数据同步方式，当前双机高可用（一主一从）架构支持**异步复制**和**半同步复制**两种方式。
实例可用状态	显示实例当前可访问状态，当状态为正常时，正常接收用户请求，如果显示异常，则说明实例当前已经无法接收应用请求。
故障切换条件	当 Master 节点发生故障无法恢复时，需要自动进行故障切换，由 Slave 提供服务，此时系统定义了故障切换条件，条件为主备延迟大小和主备延迟时间，系统默认条件为10240MB和10秒。具体切换条件请参考 故障切换说明。
主可用区	Master 节点所在的可用区。
备可用区	Slave 节点所在的可用区。

2.2 当数据复制方式为半同步时具体展示信息如下：

信息	说明
数据复制方式	主备之间数据同步方式，当前双机高可用（一主一从）架构支持**异步复制**和**半同步复制**

	两种方式。
实例可用状态	显示实例当前可访问状态，当状态为正常时，正常接收用户请求，如果显示异常，则说明实例当前已经无法接收应用请求。
故障退化条件	当实例数据复制方式为半同步复制时，在用户设定的条件范围外，系统会自动退化主备复制方式为异步，以保证系统的可用性。此退化条件为主备延迟大小或者延迟时间。其中，PostgreSQL 大版本为9的实例仅支持主备延迟大小的条件。具体请参考 退化说明。
故障切换条件	当 Master 节点发生故障无法恢复时，需要自动进行故障切换，由 Slave 提供服务，此时系统定义了故障切换条件，条件为主备延迟大小或者时间。应用可基于特殊需求来修改切换条件。具体请参考 故障切换说明。
主可用区	Master 节点所在的可用区。
备可用区	Slave 节点所在的可用区。

3. 单击**修改**，可修改当前实例的数据复制方式。

说明：

数据复制方式修改后立即生效，修改数据方式可能引起主备切换，主备切换时会有瞬间闪断，请确保应用有重连。

修改复制方式及配置
数据复制方式修改后立即生效，可能会产生主备自动切换，请核实业务详情后操作。
实例 ID / 名称
pc
数据复制方式
异步
半同步
了解更多
数据采用半同步复制模式，支持异常情况下数据库可切换
延迟退化条件
延迟 >
0
MB 或者 延迟时间 >
0
秒时，发生退化
同步退化为异步时，主备节点同步延迟大小或者延迟时间设置
故障切换条件
延迟 ≤
10240
MB 并且 延迟时间 ≤
10
秒时，允许切换
切换说明
切换为异步模式后，允许系统自动进行主备切换时允许的主备同步延迟大小或者延迟时间设置
确定
取消

切换实例主备

最近更新时间：2024-03-20 13:20:01

主备切换原因

实例的 Master 和 Slave 节点角色互换，称之为主备切换。切换后实例地址保持不变，应用自动连接到新的 Master 节点，从而保证了实例高可用性。主备切换的原因主要为以下几点：

故障切换

系统检测到实例发生异常，无法正常使用时自动发起的主备切换。具体切换条件请查看 故障切换条件说明。

人工切换

应用运维人员或者被授权的腾讯云技术专家手动发起的主备切换。人工切换包含正常主备延迟的切换和超过主备延迟的强制切换。

强制切换

当实例的主备复制方式为异步复制或者半同步降级为异步复制时，Master 发生故障无法恢复时会触发主备切换。因数据并非实时同步到 Slave，有较小概率会引起数据不一致。当前系统默认设置了允许切换条件，您也可以基于业务需要进行特殊设置。因此只有满足切换条件才允许切换。为满足紧急情况下的切换，系统提供了强制切换能力。

注意：

为了防止随时间推移而切换条件发生变化，当执行强制切换时，需要立即执行切换。

主备切换影响

主备切换过程中会有闪断，请确保应用程序有重连机制。
如果主实例挂载的有只读实例，主备切换后只读实例会有分钟级延迟。

手动切换实例

1. 登录 PostgreSQL 控制台，在实例列表，单击**实例 ID** 或**操作**列的**管理**，进入实例详情页面。
2. 在实例详情的**可用性信息**部分单击**主备切换**。

可用性信息
主备切换
切换记录
数据复制方式
半同步
修改
了解更多
实例可用状态
正常
延迟退化条件
512MB (或者5秒)
故障切换条件
10240MB (并且10秒)
切换说明
主可用区
曼谷一区
备可用区
曼谷一区
可用区变更

查看切换记录

1. 登录 PostgreSQL 控制台，在实例列表，单击**实例 ID** 或**操作**列的**管理**，进入实例详情页面。
2. 在实例详情的**可用性信息**部分，单击**切换记录**。系统将保留1年内的切换记录。

主备延迟监控指标

腾讯云数据库 PostgreSQL 提供详细的监控信息来帮助您查看主备节点同步延迟。具体为如下几个监控指标：

指标中文名	指标中文描述
备库日志落盘延迟（Bytes）	备库落盘 LSN 与主实例当前 LSN 之间的大小差异，对于主实例而言，此指标可以体现出故障切换的数据丢失大小。
备库日志落盘时间延迟（秒）	日志从主库发送至备库与备库接收到日志并落盘之间的时间差异，对于主实例而言，此指标可以体现出故障切换的数据丢失大小。实例版本在10.x以上版本才有此指标。
主备数据同步延迟（Bytes）	备库回放 LSN 与主实例当前 LSN 之间的大小差异，对于主实例而言，此指标可以体现出故障切换的 RTO，对于只读实例而言，此指标可以体现出数据延迟大小。
主备数据同步延迟时间（秒）	日志从主库发送至备库与备库接收到日志并回放之间的时间差异。实例版本在10.x以上版本才有此指标。
备库日志发送与回放位置差异（Bytes）	日志从主库发送至备库与备库回放完成之间的大小差异，主要反映了备库日志应用的速度，主要能够通过此指标查看出备库的性能、网络传输的速度。只读实例无此指标。

升级实例
升级内核小版本

最近更新时间：2024-01-22 16:22:57

云数据库 PostgreSQL 支持升级内核小版本，升级内核小版本可实现实例新功能使用、性能提升或者问题修复等能力。
云数据库 PostgreSQL 支持的内核版本的功能详情，请参考内核版本更新动态。

操作场景

用户自助在腾讯云控制台手动升级内核小版本。

升级规则

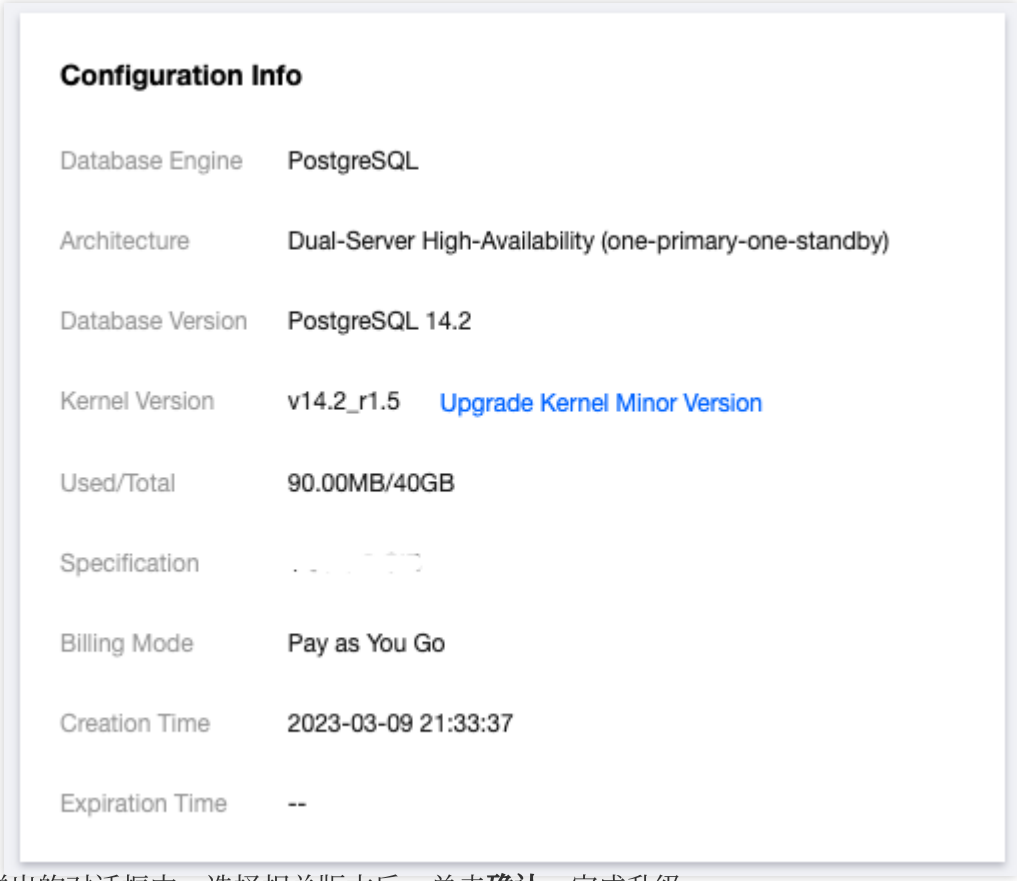
若待升级的内核版本的实例有关联其他实例（只读实例），为保证数据复制一致性，关联实例会一起进行内核小版本升级。
云数据库 PostgreSQL 版本升级可能会涉及数据迁移，通常迁移速度跟数据量大小有关系，请您耐心等待，升级期间您业务不受影响，可正常访问。

注意事项

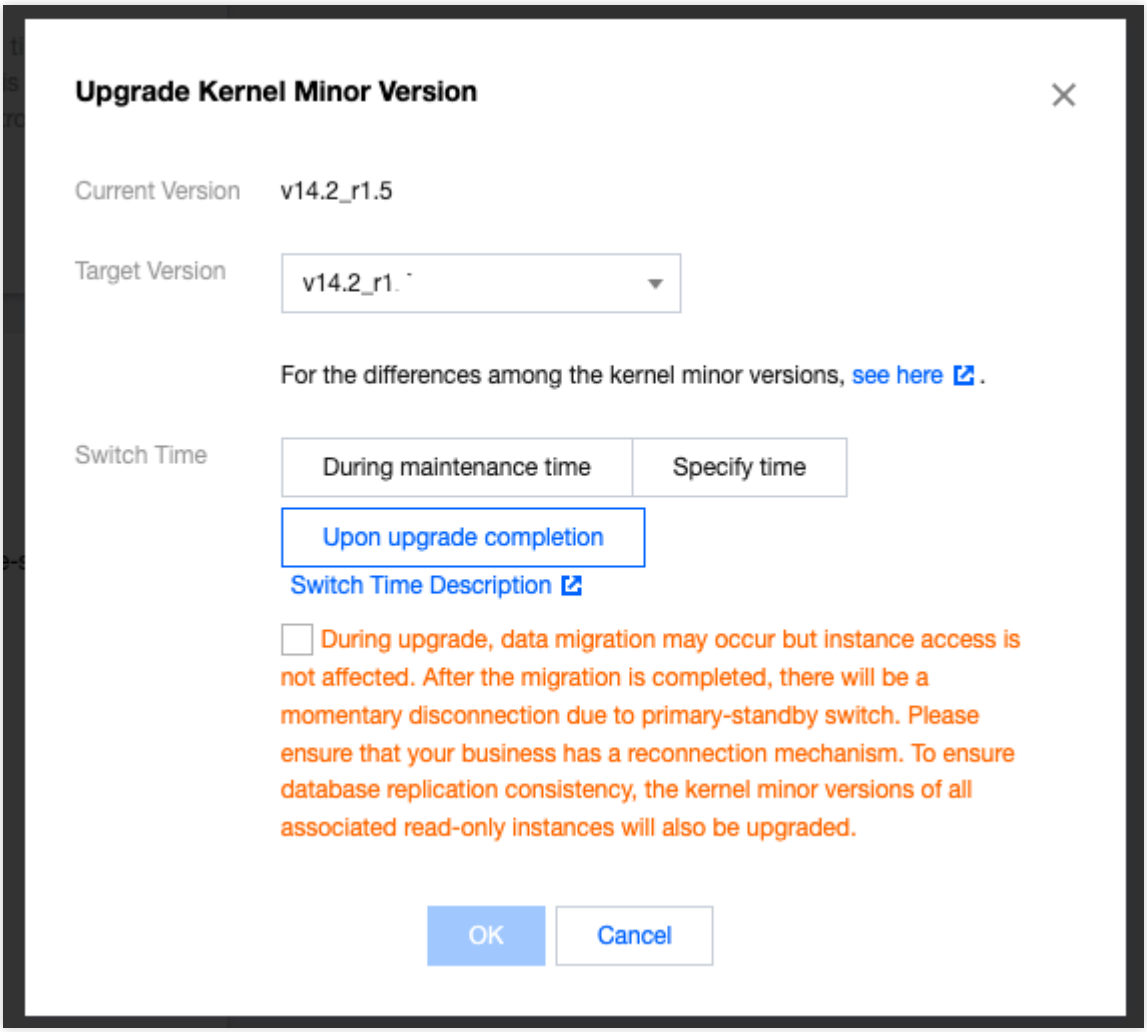
小版本升级完成时会涉及实例切换，数据库会有秒级闪断，建议应用程序有自动重连功能，并且建议选择实例可维护时间内做切换。
内核小版本升级后无法降级。

操作步骤

1. 登录 PostgreSQL 控制台，在实例列表，单击实例 ID 或**操作**列的**管理**，进入实例详情页面。
2. 在实例详情页的配置信息处，单击**升级内核小版本**。



3. 在弹出的对话框中，选择相关版本后，单击**确认**，完成升级。



注意：

数据库内核小版本升级可能会涉及到数据迁移，在升级完成时会发生秒级闪断，建议您选择切换时间为**维护时间内**，实例会在升级完成后的下一个维护时间内发起切换。

升级数据库大版本

最近更新时间：2024-03-20 14:25:11

场景及优势

PostgreSQL 不断在提供新版本给客户使用，新版本会引入更多的特性、更好的性能和更高的稳定性和安全性，因此建议您基于业务需要来合理规划，尽早升级数据库大版本。云数据库 PostgreSQL 支持数据库大版本升级，该工具有如下优势：

支持 PostgreSQL 9～15之间的跨版本升级。支持升级前演练。

在原实例上拉取只读实例进行升级操作，原实例不受任何影响。

升级流程对业务几乎只有秒级闪断和短暂只读影响。

只读时间取决于原实例对象数量，与数据规模无关。

原实例升级完成后，连接地址、标签、监控、备份集等信息完整保留。

说明：

暂不支持对开启数据透明加密实例进行大版本升级。

操作步骤

正式大版本升级

1. 登录 PostgreSQL 控制台，在实例列表，单击**实例 ID** 或**操作**列的**管理**，进入实例详情页面。
2. 单击**版本升级**，进入版本升级操作页面。

3. 单击**升级大版本**，进入版本升级页面。

3.1 目标版本

当前系统会返回可以升级的所有版本供用户选择，每个大版本会返回一个最新的版本号。

3.2 升级时间

升级过程中会有闪断和短暂的只读，因此业务需要评估可操作时间窗口，可选择立即执行、指定时间或者维护时间内。维护时间设置请参考 设置实例维护时间。

当升级完成还不到指定时间或者维护时间窗口时，实例的状态为**等待切换**，您可以在控制台**实例列表** > **操作** > **立即切换**来完成升级流程。

3.3 统计信息搜集说明

准确的统计信息可确保 PostgreSQL 查询规划器以最优方式处理查询。缺少统计信息可能会导致查询计划错误，进而可能会降低性能并占用过多内存。该操作，主要是对主实例运行 ANALYZE 并在升级后更新系统统计信息。统计信息生成时长取决于实例数据量大小。

3.4 插件升级设置

该操作主要是在升级完成后在创建过插件的数据库上执行"ALTER EXTENSION UPDATE"。有三个选项：

升级完成前之前升级插件： 由于需要检查所有的数据库的插件列表并升级到对应的版本，执行时间跟数据库的数量和插件数量正相关，会延长整个大版本升级的执行时间，请您评估后选择。

升级完成后升级插件：升级完成后，立即恢复读写。用户需要评估插件升级完成前对业务是否有影响。

不升级插件版本：升级完成后，立即恢复读写。用户自己对插件进行升级，需要评估插件升级完成前对业务是否有影响。

3.5 升级开始前是否备份

执行大版本升级时，为保证数据可恢复性，系统默认会自动进行两次备份，备份类型为**升级备份**：

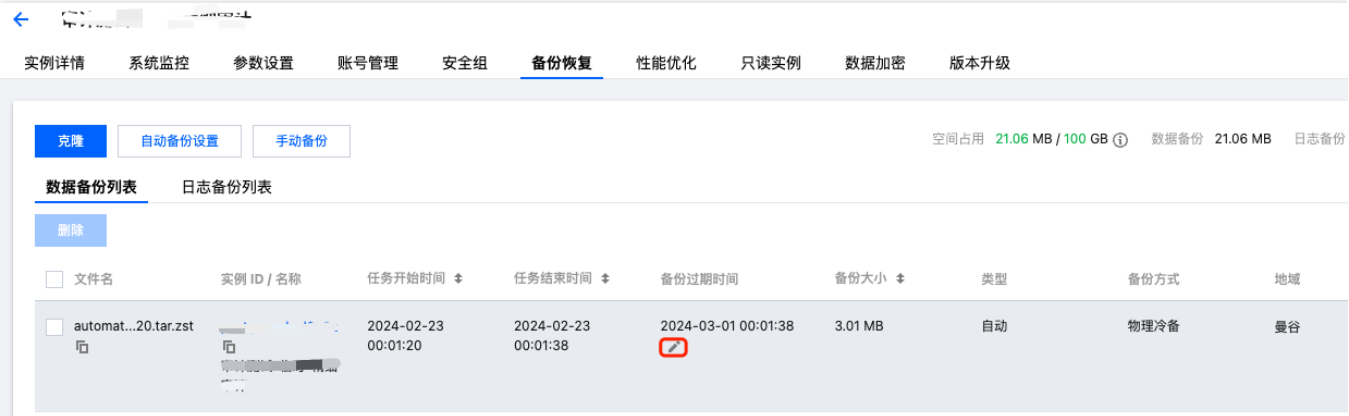
升级开始前会进行一次全量备份，在开始升级之前立即进行。用户可以使用此备份将数据库实例恢复为先前版本的状态。

升级完成后进行一次全量备份，该备份在允许对升级后的数据库实例执行新写入后立即创建。

升级前备份支持选项：

否：当实例已有备份可以恢复到升级前的状态时，可选择此选项，否则不建议选择。

是：默认选项。备份商业化后可能会产生费用，具体请参考 备份空间收费说明，用户可以在升级验证完成后选择将其删除。此备份文件的删除策略也受用户设置的备份集保留规则管理。如果您想延长备份集保留时间，请参考下图：

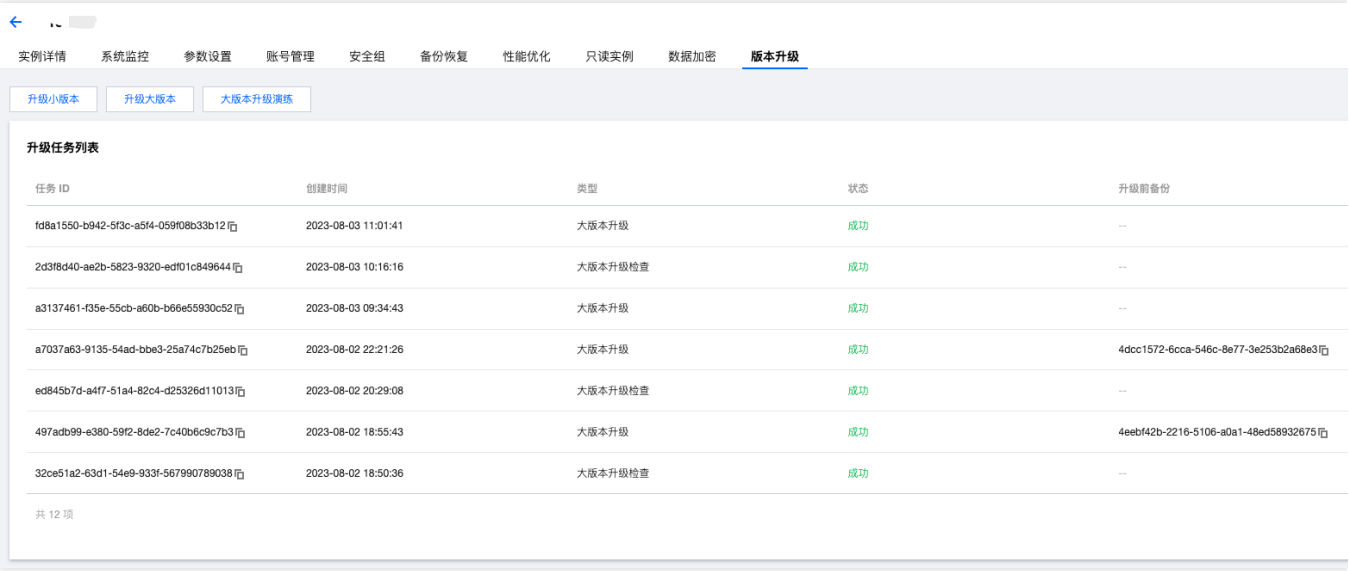


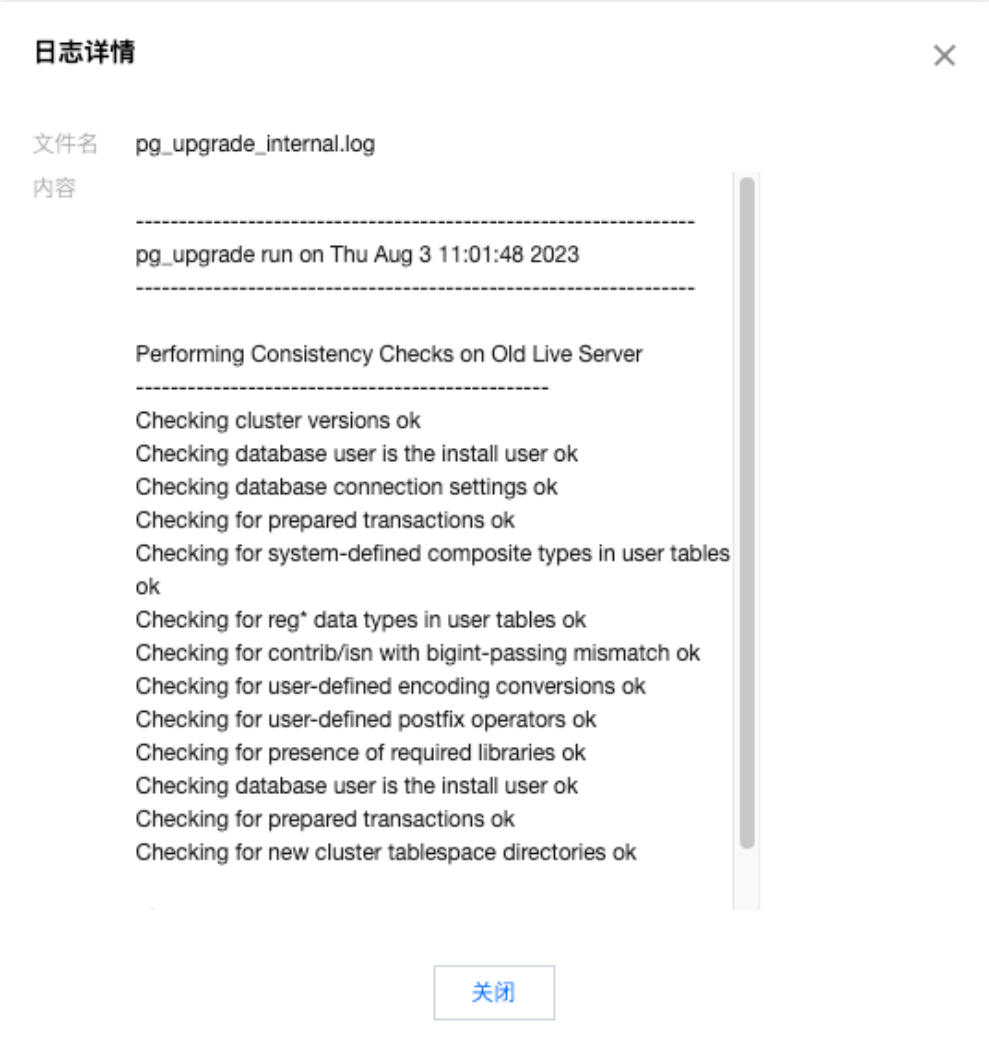
3.6 任务启动设置

只检查不发起任务：只进行升级前的检查，包含实例运行状态、实例参数设置合法性、数据库连接检查等，但不发起任务。用户可以通过该操作来预检查升级可行性。

检查并发起任务：进行升级前的检查，包含实例运行状态、实例参数设置合法性、数据库连接检查等。如果检查通过则直接发起任务。

4. 单击**提交**会产生一个升级任务，实例的状态为**内核版本升级中**，用户可以在任务列表来查看任务状态和系统日志。





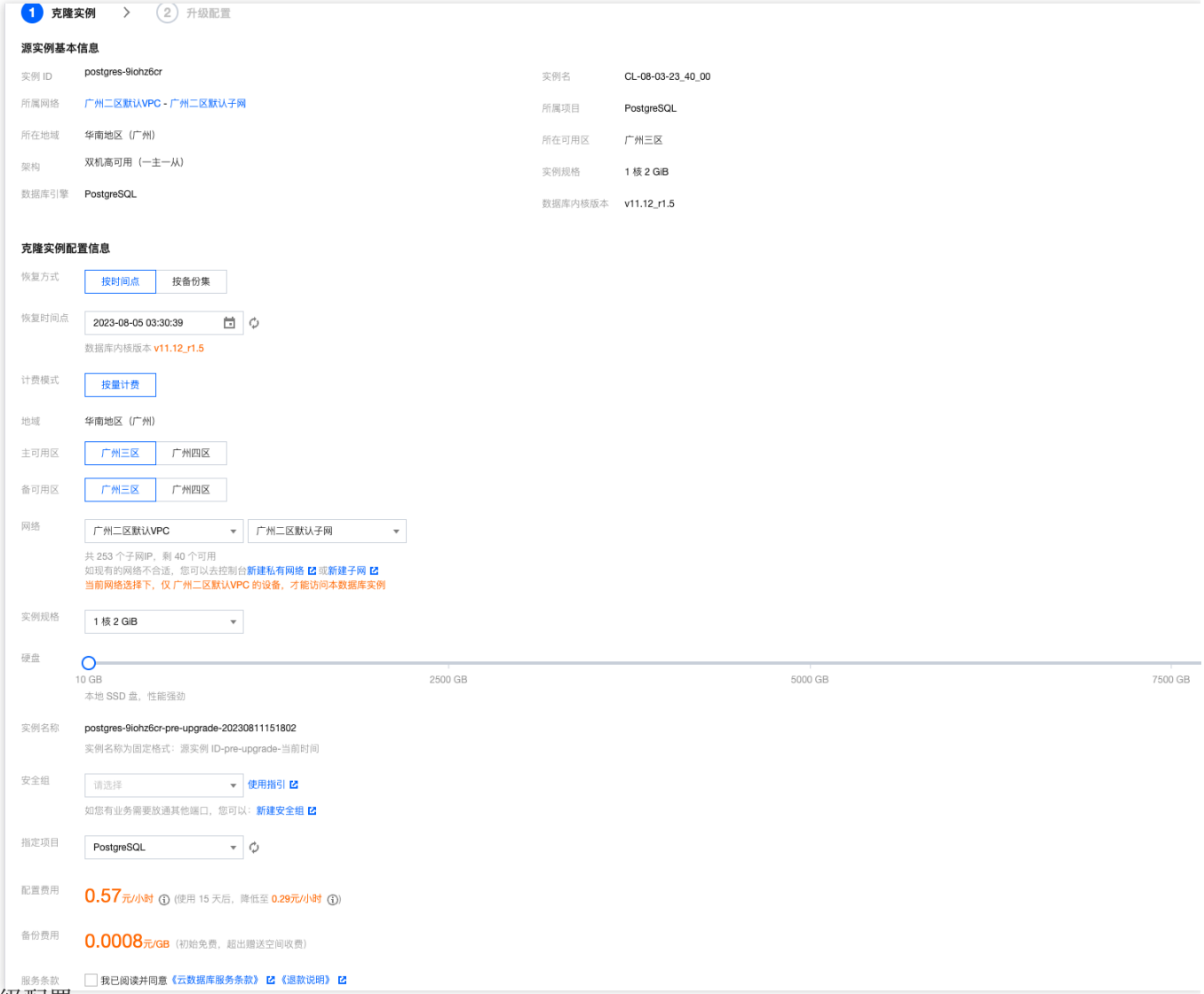
演练升级

为了保证升级的成功率，业务可以选择先演练升级。演练升级是基于当前实例的备份集先克隆出新实例，在新实例上进行升级，升级完成后产生新实例，用户可以选择直接使用该实例，也可以删除实例。克隆产生实例会正常计费，计费模式为按量计费，具体价格请参考 产品价格。具体操作如下：

1. 登录 PostgreSQL 控制台，在实例列表，单击**实例 ID** 或**操作**列的**管理**，进入实例详情页面。
2. 点击**版本升级**，进入版本升级操作页面。
3. 点击**大版本升级演练**，进入演练操作界面。

3.1 克隆实例配置

具体克隆实例配置请参考 克隆实例。



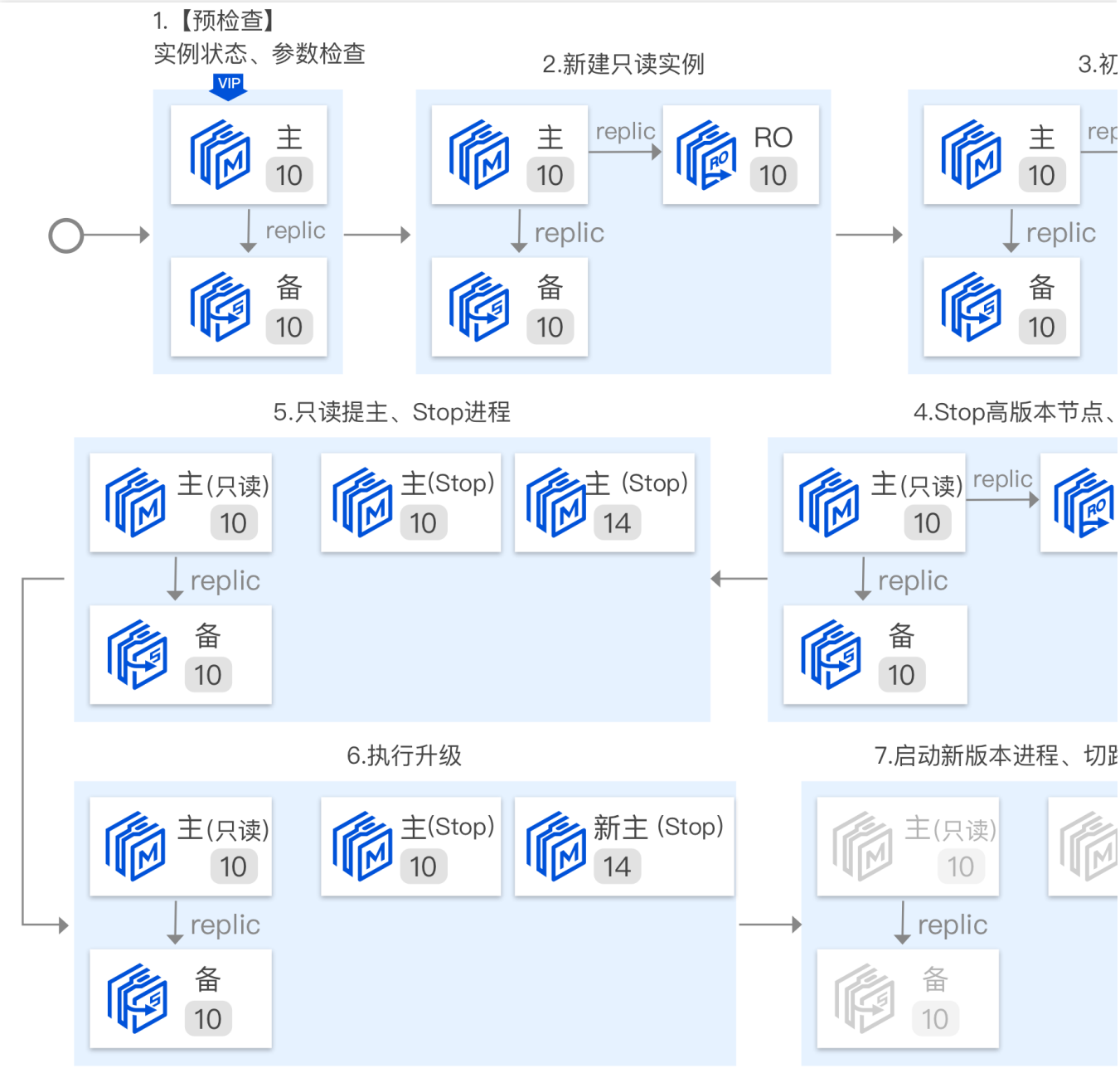
3.2 升级配置

升级配置项请参考 正式大版本升级。

单击**提交**，发起演练任务，如果原实例为 postgres-3e3ug2nj ，则系统会生产一个实例名为 postgres-3e3ug2nj-pre-upgrade-20230809154431 的新实例，并完成后续的升级。

原理解析

为了方便您了解系统原理，合理规划大版本升级时间段，本文为您展示了在大版本升级过程中，后台系统所做的操作，具体如下图所示：



上图以 PostgreSQL 10升级到 PostgreSQL 14为例，详细说明了后台系统操作步骤如下：

1. 预检查

检查实例状态是否为运行中；

检查实例参数合法性；

检查升级目标版本是否合法。

2. 新建只读实例

新建源实例的只读实例，为减少升级操作对原实例的影响，后续升级操作会在该只读实例上进行。

3. 初始化新版本数据库

新建空白目录，初始化一个版本为目标版本的新实例，该新实例只有单节点。

4. 停止高版本节点、原实例主节点只读。

停止该新版本的实例，同时将源实例设为只读。

5. 只读实例提升为主实例、停止进程。

将只读实例提升为主实例，并停止该实例。

6. 执行升级

执行升级，导出并导入元数据，处理数据。

7. 启动新版本进程、切换路由、新建备机并回收历史节点。

启动升级完成的新版本实例，切换原实例路由信息，并给新实例搭建从节点。最后清理环境，任务结束。

注意：

系统升级完成后，请业务务必做好相关验证，以保证业务顺利运行，具体如下：

请在升级之后的数据库上进行业务负载测试。

验证插件兼容性。

验证参数兼容性。

系统限制

如果实例有关联只读实例，则需要先删除只读实例再发起升级。

如果实例存在磁盘空间超用则不能发起大版本升级。

热点问题

是否支持大版本降级？

不支持大版本降级，当前云数据库 PostgreSQL 大版本向前兼容，建议优先使用高版本。

只读实例
只读实例概述

最近更新时间：2024-01-22 16:22:57

云数据库 PostgreSQL 支持用户创建一个或多个只读实例，以支持用户的读写分离和一主多从应用场景，可显著提高用户数据库的读负载能力。
目前暂不支持统一读写分离地址，即自动分离读请求与写请求。只读实例可以通过单独的 IP、PORT 来访问，也可以将只读实例分配至只读实例组中进行统一的负载均衡。
说明：
只读实例费用请参见 产品定价。

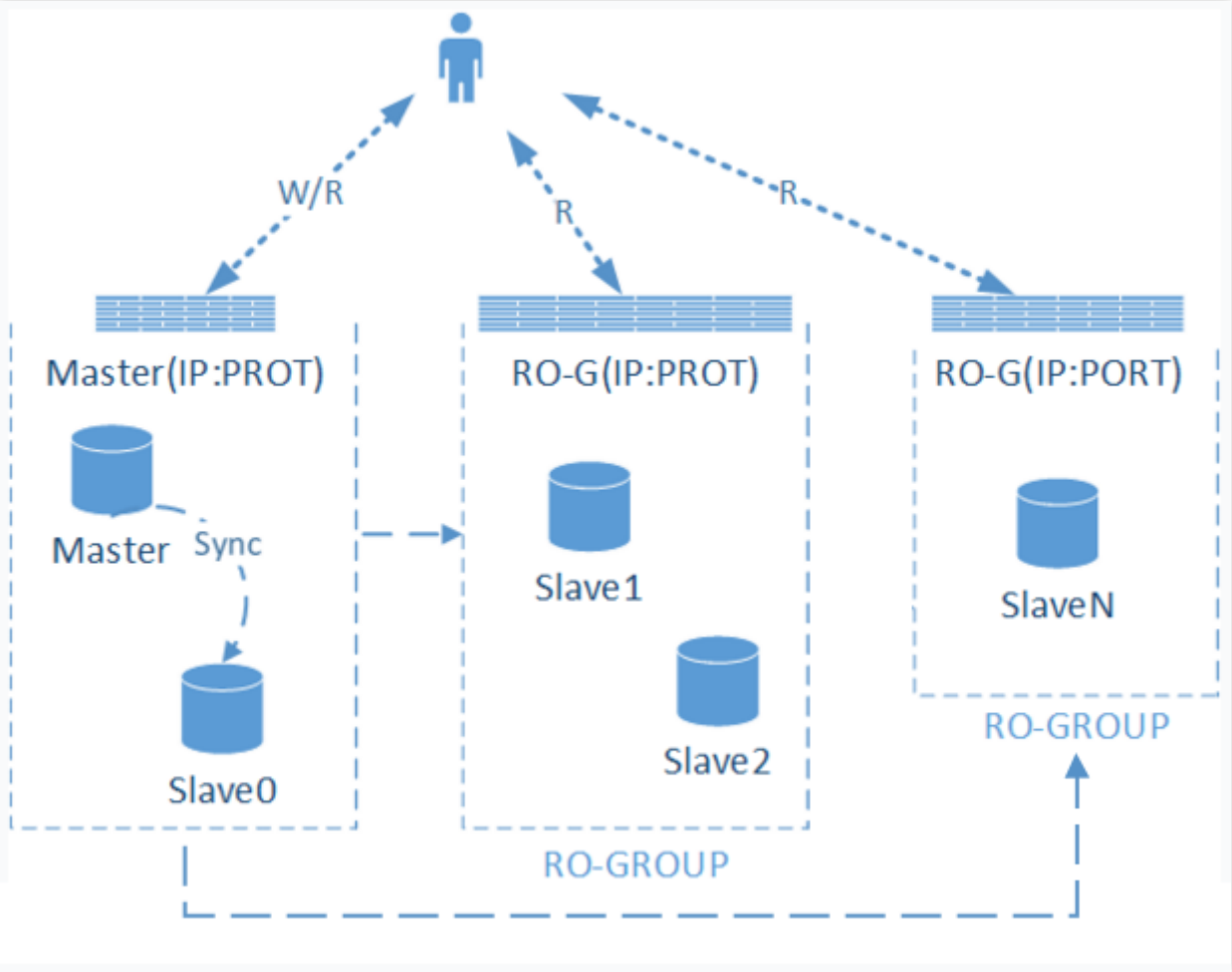
基本概念

只读实例组：带有负载均衡功能的只读实例（RO）组。若只读实例组内有多个只读实例，可将用户读请求量均匀分配到组内的每个只读实例上。且 RO 组会对外提供 IP、PORT 供访问数据库。
只读实例：可以支持读请求的单节点（无从机）实例。只读实例无法单独存在，每个只读实例都属于某个主实例。

基础架构

只读实例采用 PostgreSQL 流复制机制，将主实例（源数据库）的更改同步到所有只读实例中。只读实例均采用单节点（不含备机）架构，若只读实例宕机，我们将不断的尝试恢复，若需要更高的可用性，您可以选择 RO 组。

版权所有：腾讯云计算（北京）有限责任公司



功能限制

只读实例最低硬盘规格必须大于或等于主实例已使用存储规格。
一个主实例最多可以创建6个只读实例。
不支持备份以及回档功能。
不支持数据迁移至只读实例。
不支持创建/删除数据库。
不支持创建/删除帐号，不支持为帐号授权以及修改帐号及密码。

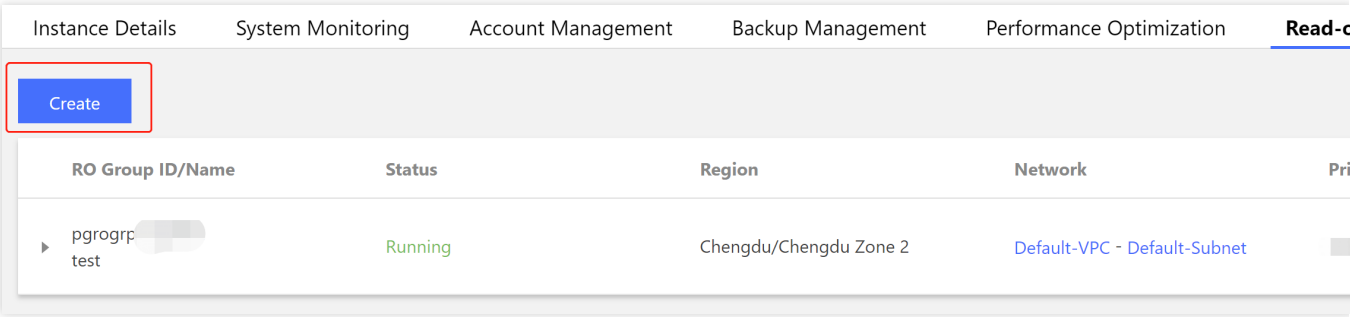
注意事项

只读实例无需维护帐号与数据库，均从主实例同步。
由于数据同步有延迟，多个只读实例之间可能存在少量数据不一致的情况。各只读实例与主实例之间的同步时延可在控制台查看，并可设置云监控告警。

只读实例的实例规格可以与主实例不一致，方便用户根据负载情况升级。建议同一 RO 组内的只读实例的实例规格保持一致。
当主库实例写入太过频繁，超过了日志自动清理阈值时，日志将自动进行删除，此时从库若未获取到已经删除的日志，主从复制将会断开，同时也会自动重建只读实例，只读实例将无法访问。
只读实例无高可用能力，建议使用只读实例组，配置两个以上的只读实例，避免只读实例单点故障后导致业务访问失败。

操作步骤

1. 登录 PostgreSQL 控制台，在实例列表，单击实例 ID 或**操作**列的**管理**进入实例管理页面。
2. 在实例详情页，单击**实例架构图**中的**添加只读实例**，或在只读实例页，单击**新建**，进入购买页。



3. 在购买页，选择只读实例的相应配置，确认无误后，单击**立即购买**。
指定RO组：支持不分配 RO 组、新建 RO 组、已有 RO 组。
新建 RO 组：若一次购买多个实例，将都分配至这个 RO 组。RO 组仅支持系统自动分配读权重，RO 组中的所有只读实例流量分配将根据读权重值来自动分配。请参见 创建只读实例 RO 组。
已有 RO 组：指定一个已有 RO 组，若一次购买多个实例，将都分配至这个 RO 组。
延迟超限剔除：开启延迟剔除功能后，可根据配置的延迟大小来决定只读实例是否从 RO 组中剔除。延迟大小是指主实例与只读实例之间数据同步的日志大小差异，配置粒度为 MB。
可用区：可选择与主实例同地域下的所有可用区。
4. 购买完成后，返回实例列表，待实例状态变为**运行中**，即可正常使用。

管理只读实例 RO 组

最近更新时间：2024-01-22 16:22:57

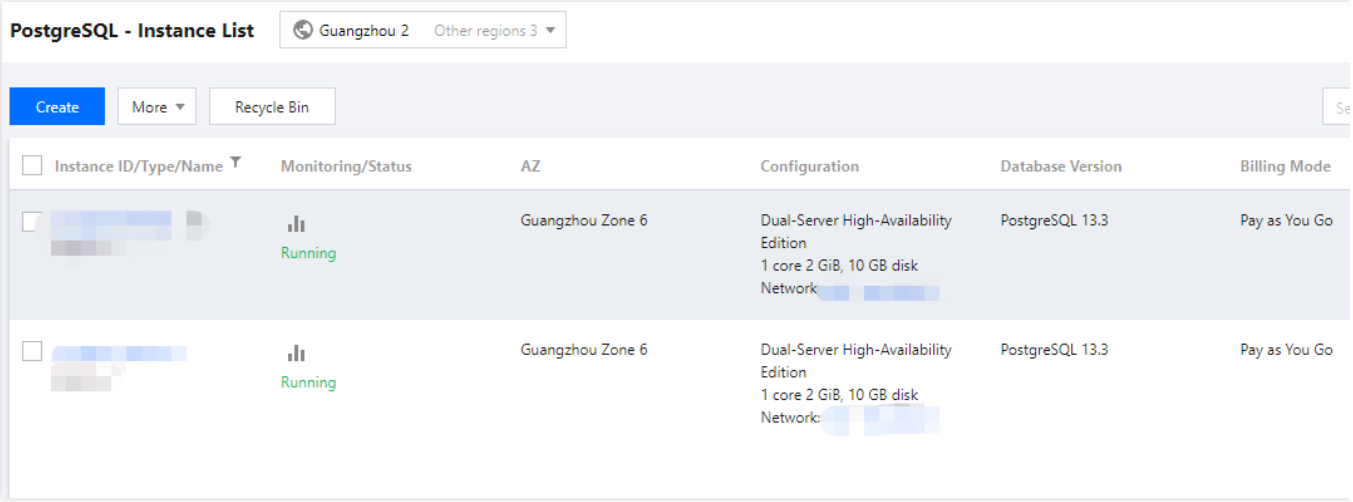
云数据库 PostgreSQL 支持用户创建多个只读实例并组成只读实例 RO 组，适用于读写分离和一主多从应用场景，可显著提高用户数据库的读负载能力。

前提条件

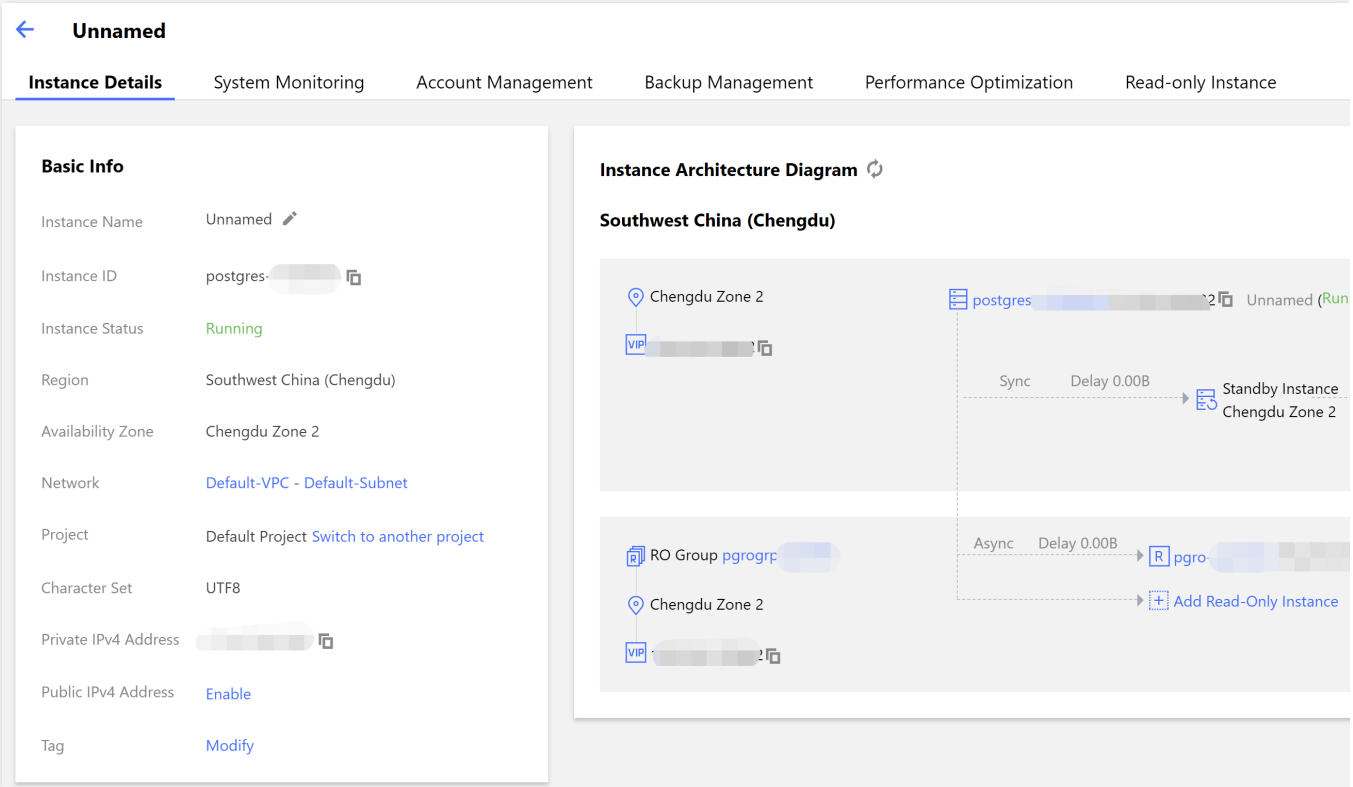
已创建主实例，请参见 创建 PostgreSQL 实例。
已创建只读实例，请参见 创建只读实例。

创建只读实例 RO 组

1. 登录 PostgreSQL 控制台，在实例列表对应实例的**操作**列，选择**更多** > **新建只读实例**，进入只读实例购买页。



说明：
您也可单击实例 ID 或**操作**列的**管理**，进入实例管理页面。
在实例详情页，单击**实例架构图**中的**添加只读实例**，进入只读实例购买页。
或在只读实例页，单击**新建**，进入只读实例购买页。



2. 在购买页，选择只读实例的相应配置，确认无误后，单击**立即购买**。

指定 RO 组：选择**新建 RO 组**，若一次购买多个实例，将都分配至这个 RO 组。RO 组仅支持系统自动分配读权重，RO 组中的所有只读实例流量分配将根据读权重值来自动分配。

设置 RO 组名称：RO 组名称不要求唯一。支持长度小于60的中文、英文、数字、 `-` 、 `_` 。

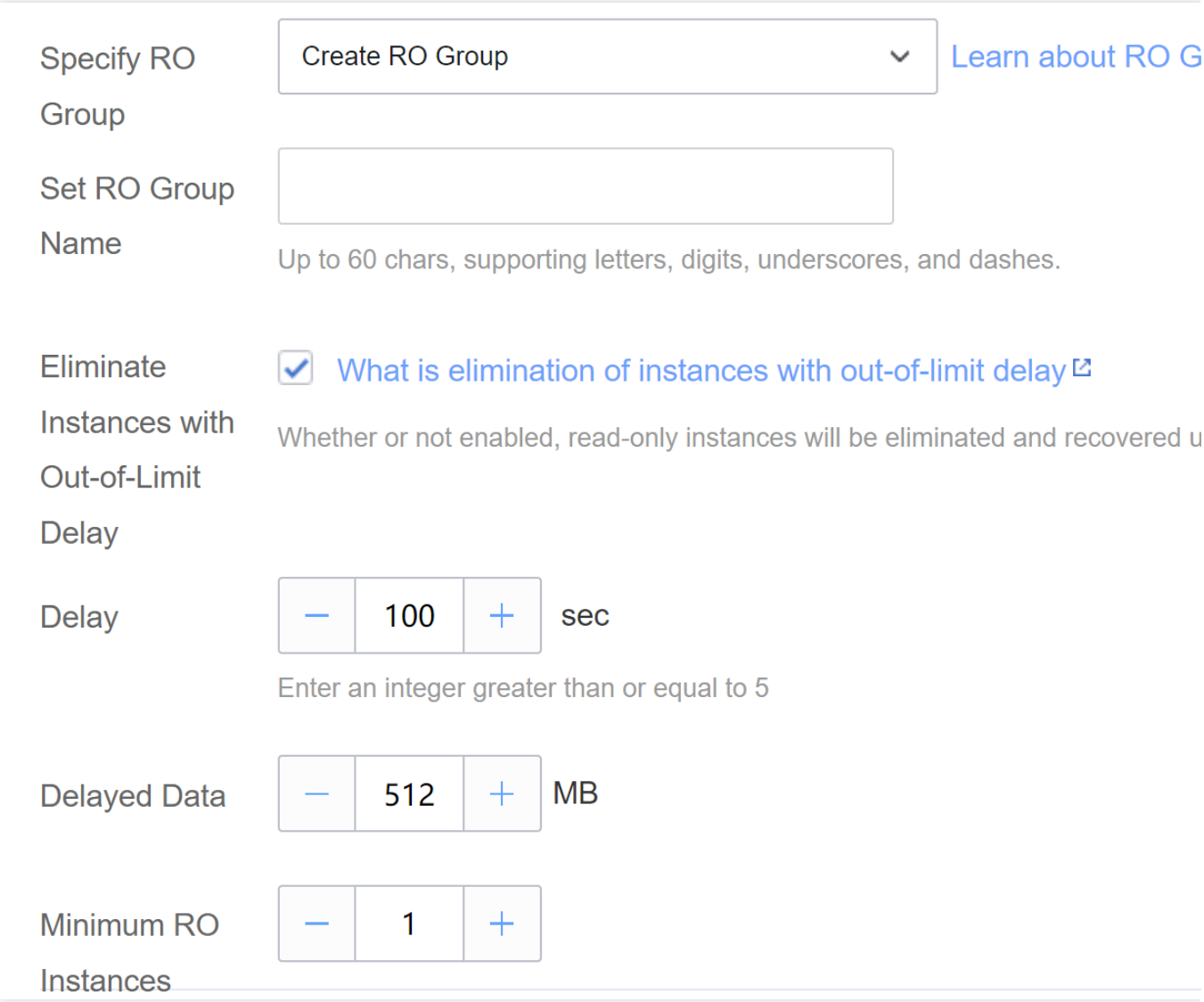
延时超限剔除：是否启动剔除策略。被剔除的实例权重自动设置为0，且实例状态为剔除中、权重为0，当只读实例延迟时间小于阈值时会重新加入到 RO 组。若只读实例被剔除会向用户发出告警，配置只读实例剔除告警和接收对象请参见 告警功能。

不管实例是否启用延时超限剔除功能，只读实例故障都会被剔除，待实例修复也会重新加入到 RO 组。

延迟时间：为只读实例设置延迟的时间，超过阈值的只读实例会被剔除 RO 组。

延迟大小：为只读实例设置延迟的日志大小阈值，超过阈值的只读实例会被剔除 RO 组。

最少保留实例数：组内需要保证的实例下限数，若现有只读实例数小于等于此下限且延迟时间超过阈值，现有只读实例均不被剔除。



3. 返回实例列表，创建的实例状态为**发货中**，待状态切换成**运行中**，则表示只读实例创建成功。

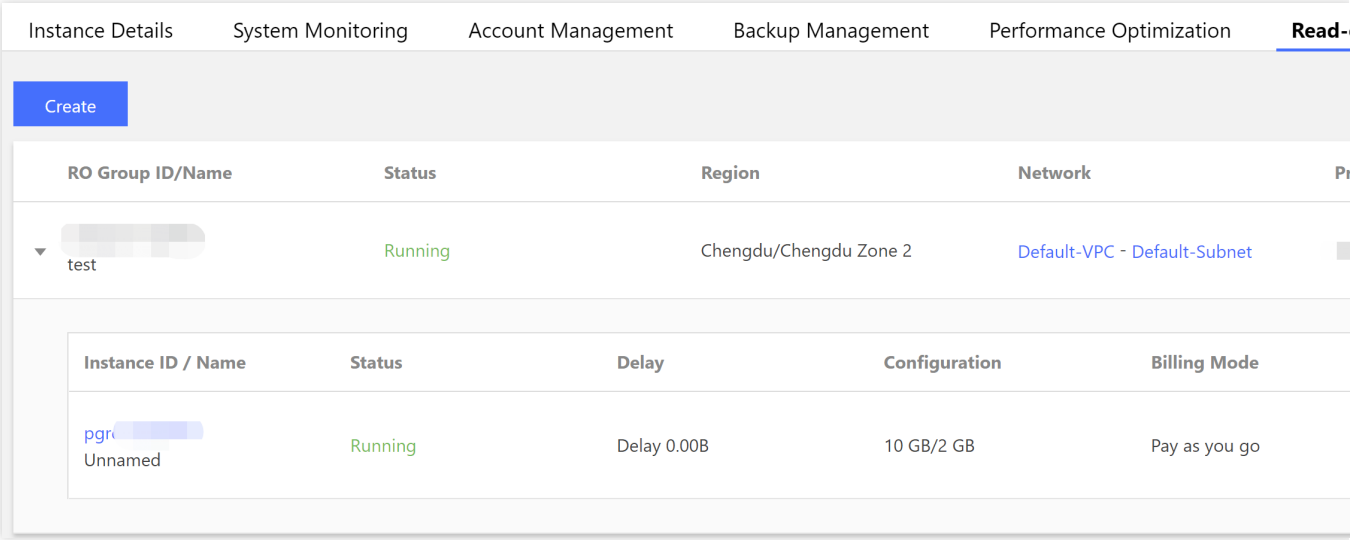
配置只读实例 RO 组

在只读实例 RO 组的配置界面，可以配置只读实例 RO 组的名称、延时超限策略、延时阈值、最少保留实例等基本信息。

说明：

RO 组内只读实例可使用不同规格，读写权重由系统根据实例规格进行自动分配。

1. 登录 PostgreSQL 控制台，在实例列表，单击主实例 ID，进入实例管理页面。
2. 在实例管理页面，选择**只读实例**页，在 RO 组列单击**配置**，进入 RO 组配置页。



3. 在 RO 组配置页，配置 RO 组信息，单击**提交**。

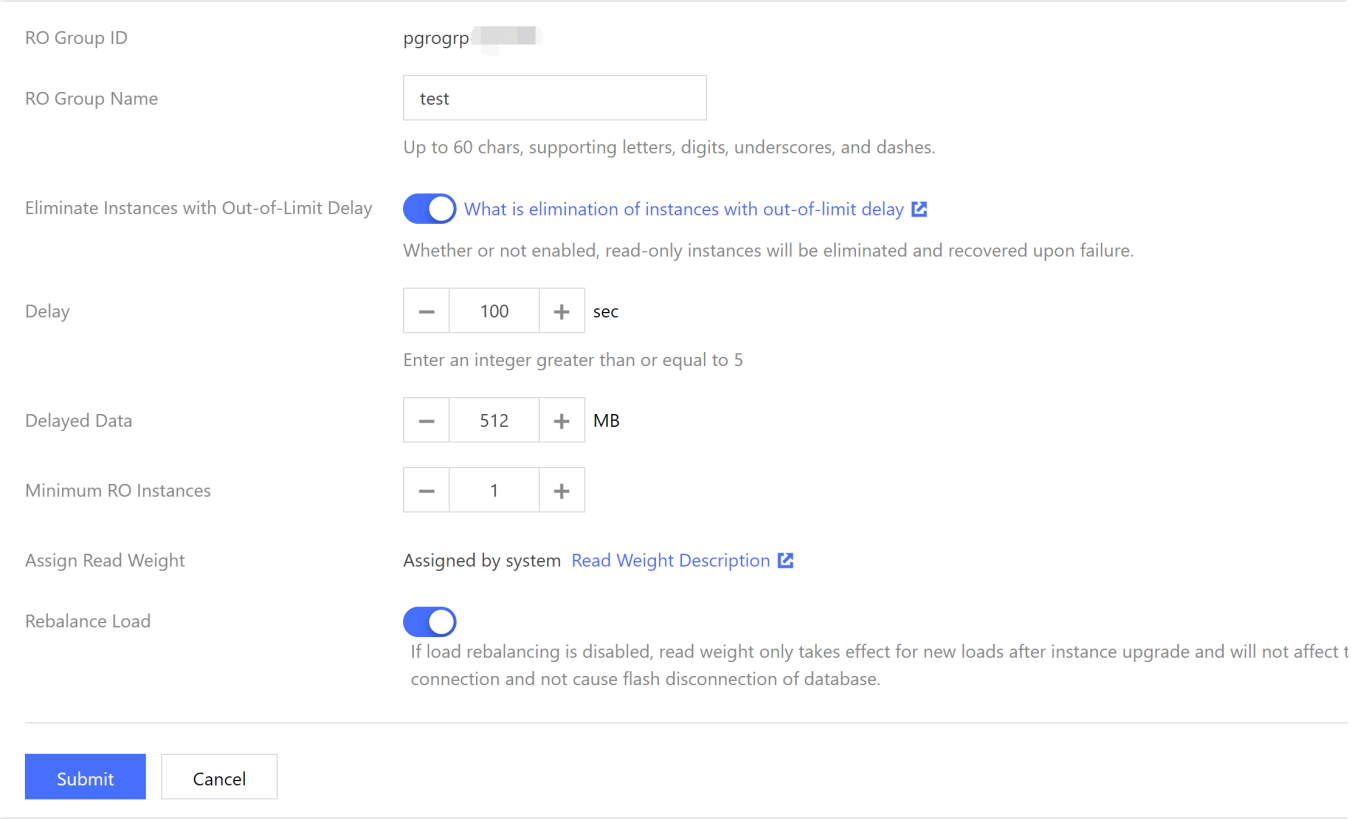
读权重分配：只读实例 RO 组中的所有只读实例流量分配将根据读权重值来自动分配，可实现负载均衡能力，避免因管理多个只读实例地址而增加 IP 地址的管理难度。RO 组仅支持系统自动分配权重。关于不同规格的只读实例的读权重值列表如下：

实例规格	权重
内存2GB	1
内存4GB	2
内存8GB	2
内存12GB	4
内存16GB	4
内存24GB	8
内存32GB	8
内存48GB	10
内存64GB	12
内存96GB	14
内存128GB	16
内存240GB	26
内存480GB	50

重新负载均衡：

关闭重新负载均衡时，修改权重时仅对新增负载生效，不改变原长连接所访问的只读实例，不会引起数据库闪断。

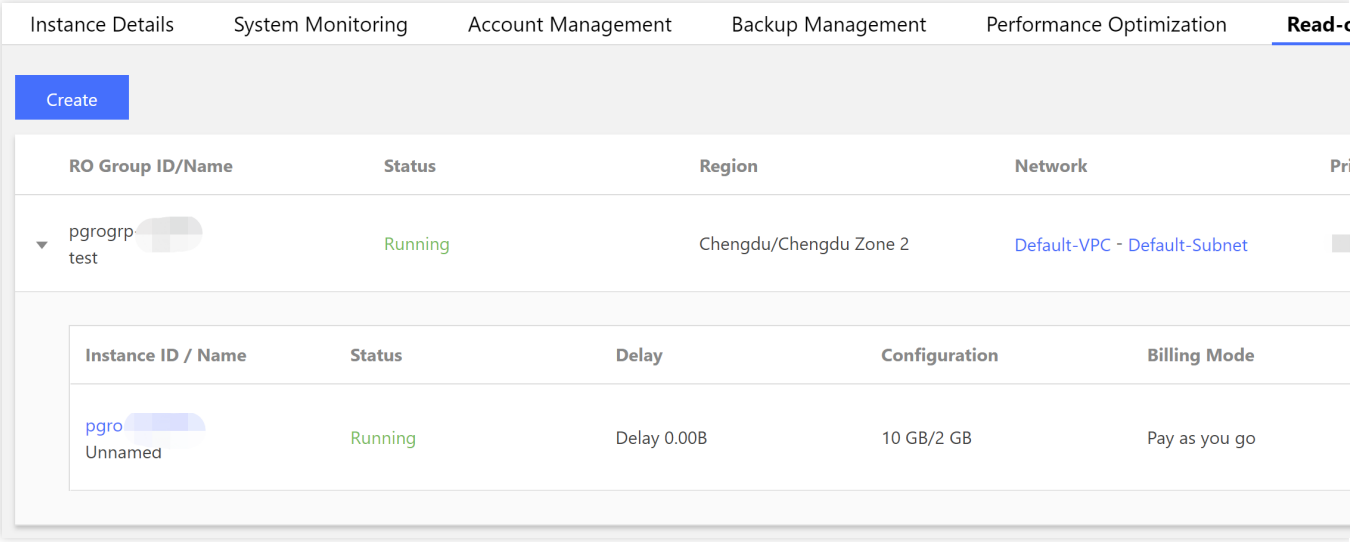
开启重新负载均衡时，当 RO 组中的只读实例进行了变更配置操作，RO 组会自动闪断所有连接，所有新建连接会按照 RO 组中的实例权重重新分配连接。



删除只读实例 RO 组

RO 组随着组内最后一个只读实例被销毁后，即可删除。

1. 登录 PostgreSQL 控制台，在实例列表，单击主实例 ID，进入实例管理页面。
2. 在实例管理页面，选择**只读实例**页，可查看所有 RO 组列表，当 RO 组中无任何 RO 实例后即可删除。


Instance Details
System Monitoring
Account Management
Backup Management
Performance Optimization
Read-
Create
RO Group ID/Name
Status
Region
Network
pgrogrp
test
Running
Chengdu/Chengdu Zone 2
Default-VPC - Default-Subnet
Instance ID / Name
Status
Delay
Configuration
Billing Mode
pgro
Unnamed
Running
Delay 0.00B
10 GB/2 GB
Pay as you go

剔除策略和负载均衡

最近更新时间：2024-01-22 16:22:57

云数据库 PostgreSQL 支持用户创建多个只读实例并组成只读实例 RO 组，适用于读写分离和一主多从应用场景，可显著提高用户数据库的读负载能力。本文主要介绍 RO 组的部分管理功能。

流量重均衡

关闭重新负载均衡时，修改权重仅对新增负载生效，不改变原长连接所访问的只读实例，不会引起数据库闪断。
开启重新负载均衡时，当 RO 组中的只读实例升级后，其读权重值发生了变化，当只读实例升级完成后， RO 组中的所有连接将重新被断开。所有新增连接将按照设置的权重均衡负载。
如果您对当前 RO 组中各实例的连接数分配不太满意，也可以手动重新均衡连接。登录 控制台，单击实例 ID 进入管理页，在【只读实例】页中对实例【重新均衡】。
注意：
请保证您的业务具有自动重连机制。若无自动重连机制，请勿开启自动重均衡和手工重均衡。

故障剔除

当只读实例 RO 组中的只读实例因意外导致无法访问时，RO 组将自动将此只读实例进行剔除，此规则为默认规则。

延迟剔除

开启延迟剔除功能后，可根据用户配置的延迟大小来决定只读实例是否从 RO 组中剔除。延迟大小是指主实例与只读实例之间数据同步的日志大小差异，配置粒度为MB。

读权重分配

只读实例 RO 组中的所有只读实例流量分配将根据读权重值来自动分配，可实现负载均衡能力，避免因管理多个只读实例地址而增加 IP 地址的管理难度。RO 组仅支持系统自动分配权重。关于不同规格的只读实例的读权重值列表如下：

实例规格	权重
内存2GB	1
内存4GB	2

内存8GB	2
内存12GB	4
内存16GB	4
内存24GB	8
内存32GB	8
内存48GB	10
内存64GB	12
内存96GB	14
内存128GB	16
内存240GB	26
内存480GB	50

相关操作

云数据库 PostgreSQL 支持用户创建一个或多个只读实例，请参见 创建只读实例。
云数据库 PostgreSQL 支持用户创建多个只读实例并组成只读实例 RO 组，请参见 管理只读实例 RO 组。

帐号管理
数据库权限概述

最近更新时间：2024-01-22 16:22:57

账号权限体系

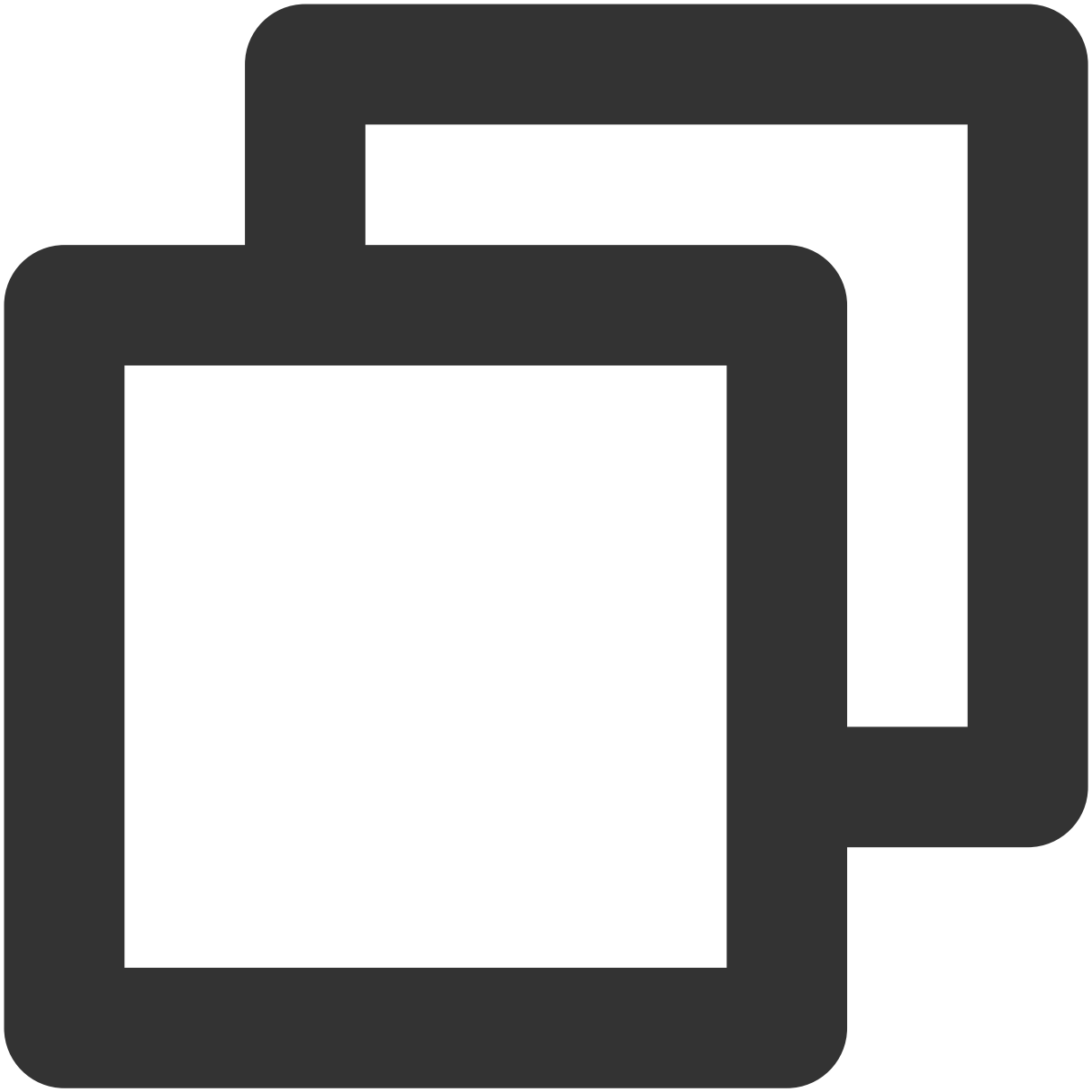
PostgreSQL 的权限管理模型，是一种典型的 RBAC（基于角色的权限控制）的实现。其将用户、角色、权限通过此模型管理起来。
PostgreSQL 中的“用户”概念和“角色”概念实际是等同的，唯一的区别在于“用户”拥有 login 权限，而“角色”拥有 nologin 权限。
PostgreSQL 的权限可以分为两类，“系统权限”和“数据库对象权限”。PostgreSQL 通过角色来进行权限管理，角色可以拥有系统权限和数据库对象权限。并且也可以作为一组权限集合，授予给其他的角色或者用户。
可以通过授予角色/用户相应的系统权限和对象权限来对数据库进行管理。

系统权限

系统权限是指可以执行特定数据库操作的权限。PostgreSQL 中使用“角色属性（Role Attributes）” 和“默认角色（Default Roles）”两种方式来管理系统权限。

角色属性

角色属性可以在 CREATE ROLE 时指定，或者通过 ALTER ROLE 来修改。角色属性保存在 pg_authid 系统表中。
CREATE ROLE 语法如下：



```
CREATE ROLE name [ [ WITH ] option [ ... ] ]
where option can be:
      SUPERUSER | NOSUPERUSER
    | CREATEDB | NOCREATEDB
    | CREATEROLE | NOCREATEROLE
    | INHERIT | NOINHERIT
    | LOGIN | NOLOGIN
    | REPLICATION | NOREPLICATION
    | BYPASSRLS | NOBYPASSRLS
    | CONNECTION LIMIT connlimit
    | [ ENCRYPTED ] PASSWORD 'password' | PASSWORD NULL
```

```
      | VALID UNTIL 'timestamp'
      | IN ROLE role_name [, ...]
      | IN GROUP role_name [, ...]
      | ROLE role_name [, ...]
      | ADMIN role_name [, ...]
      | USER role_name [, ...]
      | SYSID uid
```

其中 superuser 角色属性可以绕过所有的权限检查，执行数据库的任何操作，是数据库中的最高权限。superuser 权限类似于 Linux 操作系统 root 权限。

注意：

应安全要求，腾讯云数据库 PostgreSQL 关闭了 superuser 权限的使用。但因为部分操作必须要求 superuser 进行，所以云数据库 PostgreSQL 提供了 tencentdb_superuser 角色，请参见 用户与权限操作。

默认角色

PostgreSQL 提供了一组默认角色，这些角色提供对某些通常需要特权功能和信息的访问。管理员可以将这些角色授予其他用户或角色，使这些用户可以访问指定的功能和信息。下表列举了 PostgreSQL 11 中支持的 默认角色。

角色	允许的操作
pg_execute_server_program	允许通过部分能够执行操作语句的函数或者功能去执行相应的操作系统语句。
pg_monitor	读取和执行与监控相关的函数或视图的权限，这个角色包含 pg_read_all_settings、 pg_read_all_stats 和 pg_stat_scan_tables。
pg_read_all_settings	读取所有配置信息，包括部分 superuser 才能看到的信息。
pg_read_all_stats	阅读所有 pg_stat_* 视图并使用各种与统计相关的扩展，包括部分 superuser 才可见的信息。
pg_read_server_files	允许通过 copy 或者文件访问的函数读取数据库所在服务器的一些拥有权限的文件。
pg_signal_backend	可以通过向后端发出指令使某一个会话启动或者停止。
pg_stat_scan_tables	执行可能需要长时间锁定表（ACCESS SHARE 锁）的监控功能。
pg_write_server_files	允许通过 copy 或者文件访问的函数写入数据库所在服务器的一些拥有权限的文件。
public	隐藏角色，代表所有人，如果赋予一个权限给 public，那么所有角色就拥有了此权限。有一些对象是有赋予给 public 角色默认权限的。

数据库对象权限

数据库对象权限使用 ACL（Access Control List）进行控制。下表给出了 PostgreSQL 中所有的数据库对象权限及其缩写。

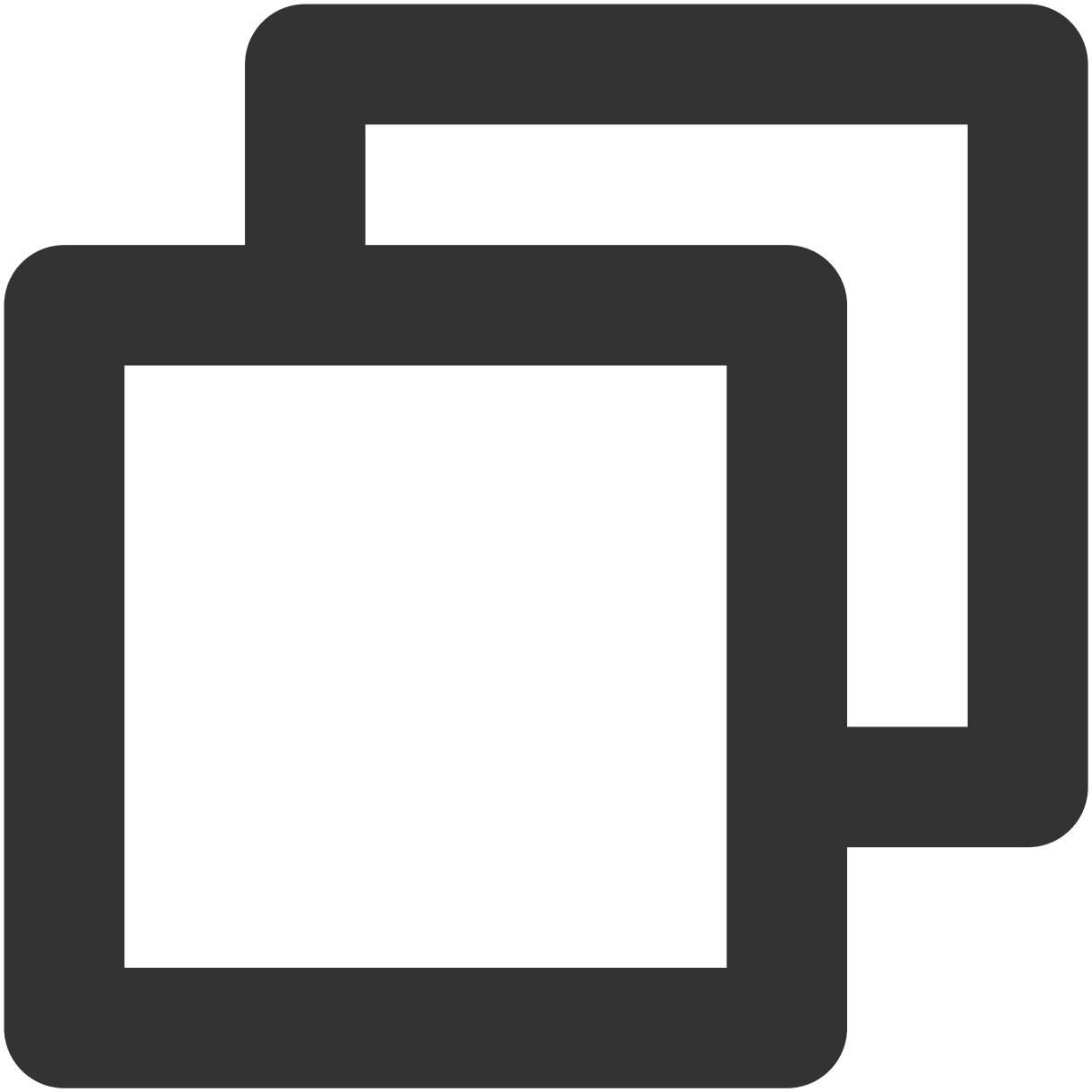
权限	缩写	支持的对象
SELECT	r (“read”)	LARGE OBJECT、SEQUENCE、TABLE (and table-like objects)、table column
INSERT	a (“append”)	TABLE、table column
UPDATE	w (“write”)	LARGE OBJECT、SEQUENCE、TABLE、table column
DELETE	d	TABLE
TRUNCATE	D	TABLE
REFERENCES	x	TABLE、table column
TRIGGER	t	TABLE
CREATE	C	DATABASE、SCHEMA、TABLESPACE
CONNECT	c	DATABASE
TEMPORARY	T	DATABASE
EXECUTE	X	FUNCTION、PROCEDURE
USAGE	U	DOMAIN、FOREIGN DATA WRAPPER、FOREIGN SERVER、LANGUAGE、SCHEMA、SEQUENCE、TYPE

下表中列出了每一种对象拥有的权限以及通过 psql 查询权限命令：

对象类型	所有权限	默认角色 Pubilc 的权限	psql 查看权限的命令
DATABASE	CTc	Tc	\\l
DOMAIN	U	U	\\dD+
FUNCTION or PROCEDURE	X	X	\\df+
FOREIGN DATA WRAPPER	U	none	\\dew+
FOREIGN SERVER	U	none	\\des+
LANGUAGE	U	U	\\dL+
LARGE OBJECT	rw	none	-

SCHEMA	UC	none	\\dn+
SEQUENCE	rwU	none	\\dp
TABLE (and table-like objects)	arwdDxt	none	\\dp
Table column	arwx	none	\\dp
TABLESPACE	C	none	\\db+
TYPE	U	U	\\dT+

PostgreSQL 中使用 aclitem 来表示一个具体的数据库对象上的权限。对于 database 和 schema，aclitem 存储在 pg_database.datacl 和 pg_namespace.nspacl 中，对于 table，view 等其他数据库对象，pg_class.relacl 中保存了 aclitem 的一个 list。对于列级别的权限，aclitem 将保存在 pg_attribute.attacl 中。
例如：normal_user=a*r/test1 表示 normal_user 这个用户对于当前数据库对象拥有 INSERT和 SELECT 权限，其中 INSERT 权限带有 with grant option，表示可以授权其他用户该权限。最后的 /test1 表示这条 aclitem 权限是 test1 授予的。



```
postgres=# \\dp
                             Access privileges
 Schema | Name | Type  | Access privileges | Column privileges | Policies
--------+------+-------+-------------------+-------------------+----------
 public | t1   | table | test1=arwdDxt/test1 |                   |
(1 rows)
postgres=# grant select on t1 to normal_user;
GRANT
postgres=# grant insert on t1 to normal_user with grant option;
GRANT
postgres=# grant update on t1 to public;
```

```
GRANT
postgres=# grant select (a) on t1 to test2;
GRANT
postgres=# \\dp
                              Access privileges
 Schema | Name | Type  |  Access privileges    | Column privileges | Policies
--------+------+-------+-----------------------+-------------------+----------
 public | t1   | table | test1=arwdDxt/test1  +| a:               +|
        |      |       | normal_user=a*r/test1+|   test2=r/test1   |
        |      |       | =w/test1              |                   |
(1 rows)
-- =w/test1 表示 test1 授予了 public 角色 UPDATE 权限
```

用户与权限操作

最近更新时间：2024-01-22 16:22:57

云数据库默认角色

腾讯云数据库 PostgreSQL 未开放 superuser 角色属性和 pg_execute_server_program、pg_read_server_files、pg_write_server_files 角色供用户使用。但因为部分操作必须要求 superuser 进行，所以云数据库 PostgreSQL 提供了 pg_tencentdb_superuser 角色用以替代 superuser。

pg_tencentdb_superuser 角色

此角色支持的权限包括：系统权限和数据库对象权限。详细权限如下表：

系统权限

权限	描述
CREATEDB	可以执行 database 的创建。
BYPASSRLS	可以绕过所有行级安全策略检查 。
REPLICATION	默认拥有 replication 权限，也可以授权 replication 权限给其他用户。
CREATEROLE	拥有如社区版中 CREATEROLE 的权限，但是已将 pg_read_server_files, pg_write_server_files 和 pg_execute_server_program 从 createrole 权限中移除。

对象权限

对象	描述
database	默认拥有所有 owner 为非 superuser 的 database 的权限。
schema	默认拥有所有 owner 为非 superuser 的 schema 的权限。
table/sequence	默认拥有所有 owner 为非 superuser 的 table/sequence 的权限。
function	默认拥有所有 owner 为非 superuser 的 function 的权限。
language	无权限。
tablespace	无权限。

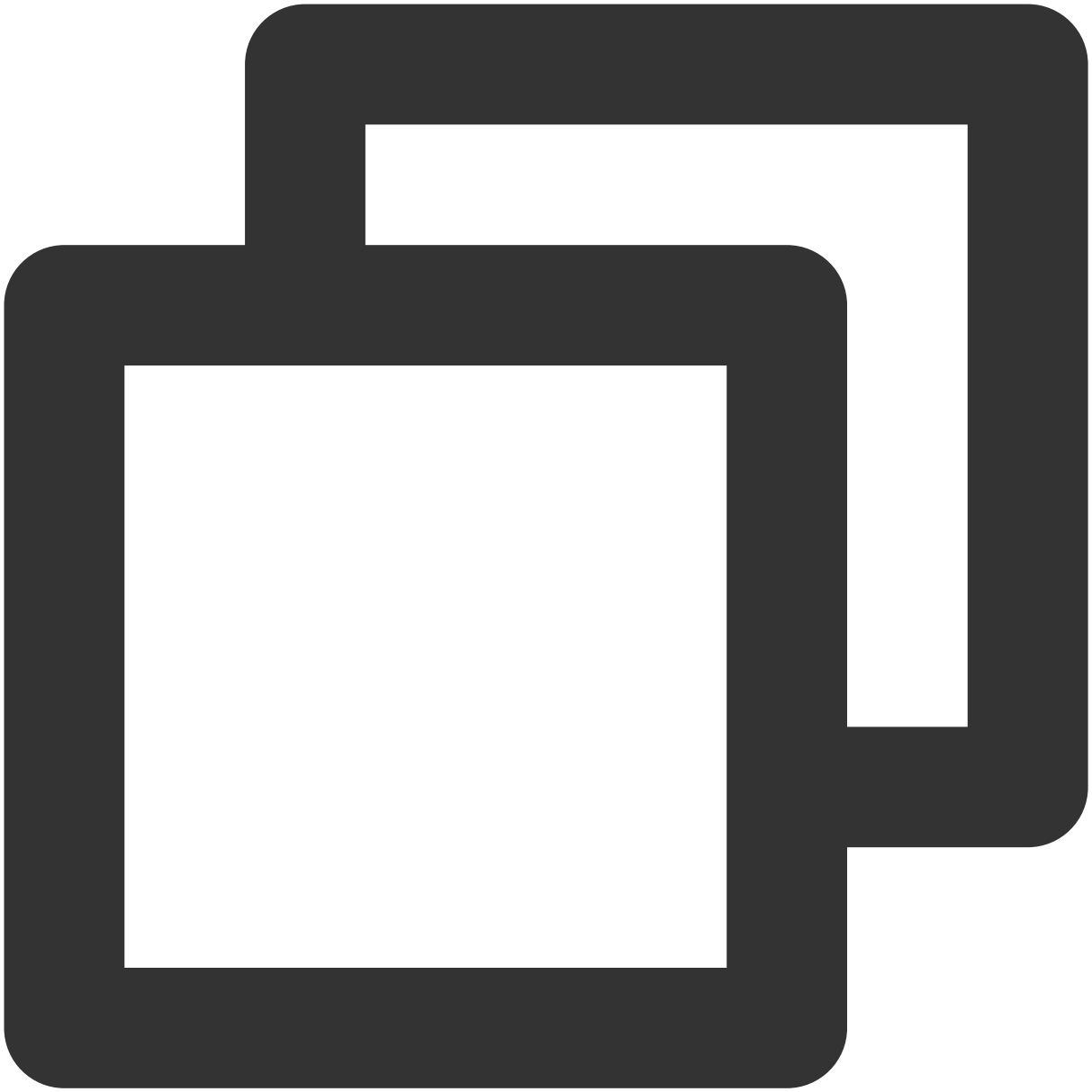
FDW / foreign server	默认拥有所有 owner 为非 superuser 的 FDW / foreign server 的权限。
TYPE	默认拥有所有 owner 为非 superuser 的 type 的权限。

其他操作

订阅/发布：可以创建订阅/发布；可以创建 publication for all tables；可以创建 slot。
插件：可创建所有支持的插件。创建插件时，当前 pg_tencentdb_superuser 会临时提权为 superuser，通过所有的权限检查。
load_file 权限保持默认行为，仅可以加载支持的插件库。
开放 pgstat_get_backend_current_activity 函数中查看死锁的详细信息的权限，方便用户自己排查死锁。
限制 pg_signal_backend 函数，只有 pg_tencentdb_superuser 才能 kill pg_tencentdb_superuser 的进程。

权限操作

详细权限操作可以参考 PostgreSQL 社区官方文档：
创建用户，详细可参考 官方文档：



```
CREATE USER name [ [ WITH ] option [ ... ] ]

where option can be:

    SUPERUSER | NOSUPERUSER
  | CREATEDB | NOCREATEDB
  | CREATEROLE | NOCREATEROLE
  | INHERIT | NOINHERIT
  | LOGIN | NOLOGIN
  | REPLICATION | NOREPLICATION
  | BYPASSRLS | NOBYPASSRLS
```

```
    | CONNECTION LIMIT connlimit
    | [ ENCRYPTED ] PASSWORD 'password' | PASSWORD NULL
    | VALID UNTIL 'timestamp'
    | IN ROLE role_name [, ...]
    | IN GROUP role_name [, ...]
    | ROLE role_name [, ...]
    | ADMIN role_name [, ...]
    | USER role_name [, ...]
    | SYSID uid
```

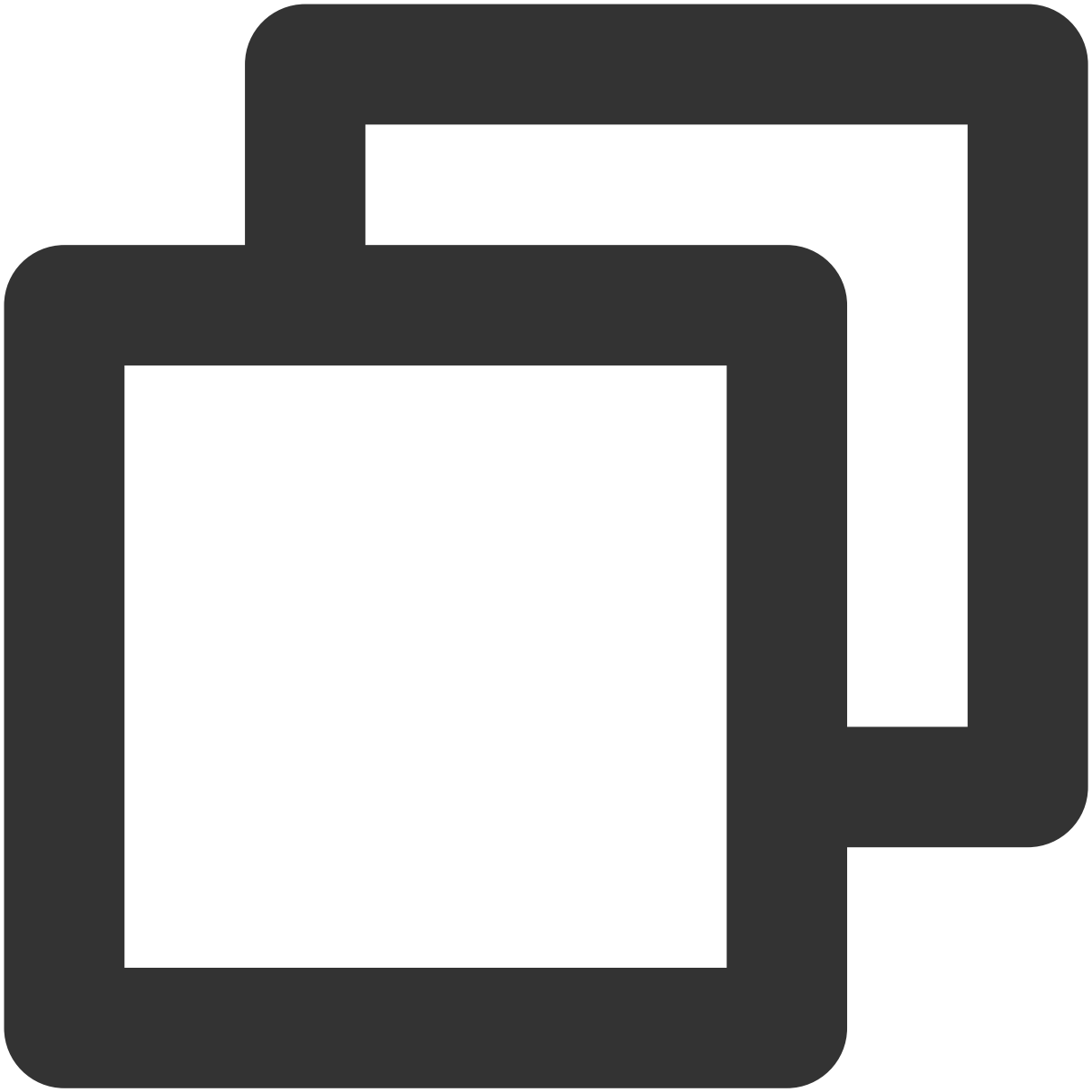
创建角色，详细可参考 官方文档：

```
CREATE ROLE name [ [ WITH ] option [ ... ] ]

where option can be:

    SUPERUSER | NOSUPERUSER
  | CREATEDB | NOCREATEDB
  | CREATEROLE | NOCREATEROLE
  | INHERIT | NOINHERIT
  | LOGIN | NOLOGIN
  | REPLICATION | NOREPLICATION
  | BYPASSRLS | NOBYPASSRLS
  | CONNECTION LIMIT connlimit
  | [ ENCRYPTED ] PASSWORD 'password' | PASSWORD NULL
  | VALID UNTIL 'timestamp'
  | IN ROLE role_name [, ...]
  | IN GROUP role_name [, ...]
  | ROLE role_name [, ...]
  | ADMIN role_name [, ...]
  | USER role_name [, ...]
  | SYSID uid
```

修改角色属性，详细可参考 官方文档：



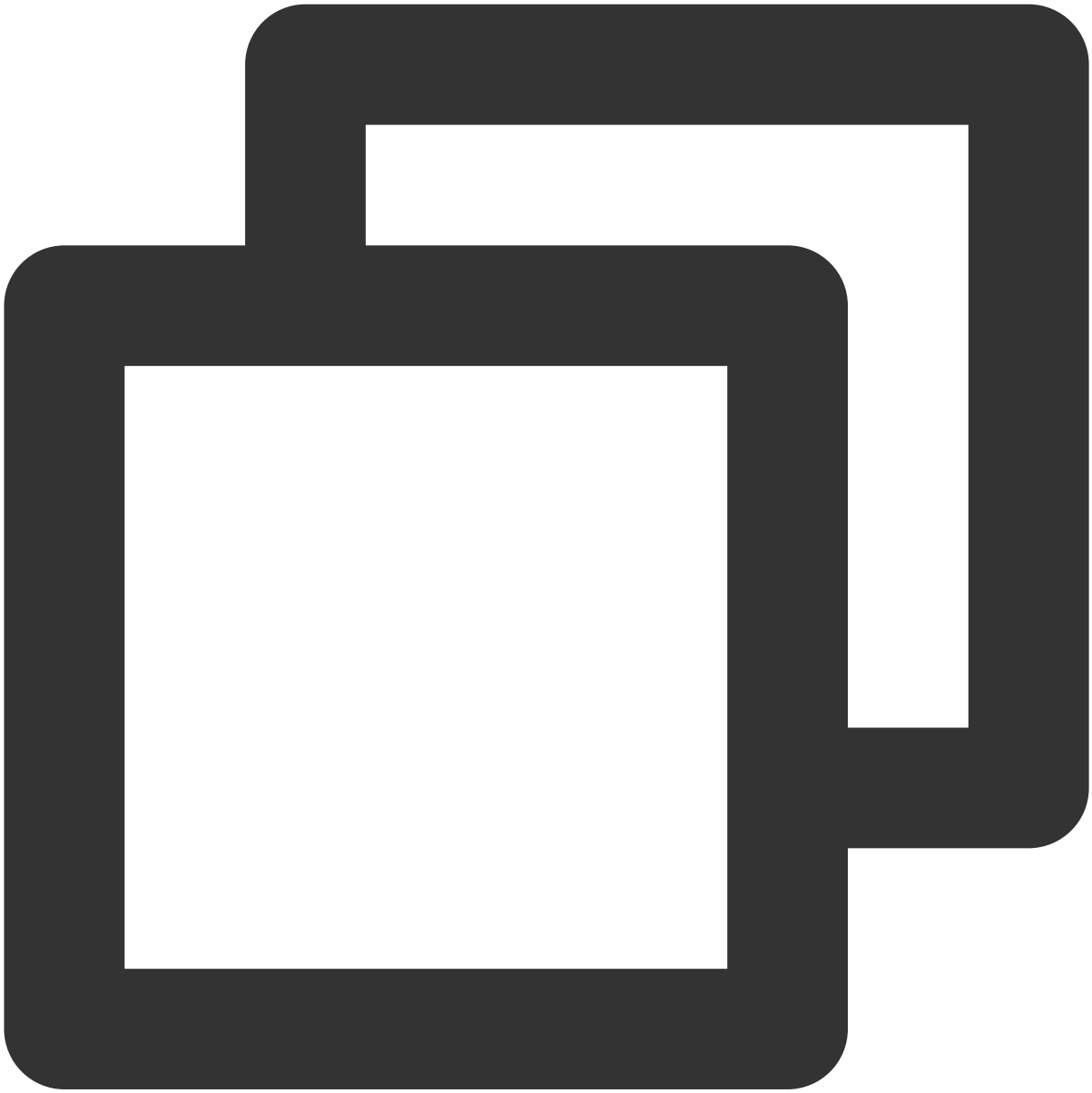
```
ALTER ROLE role_specification [ WITH ] option [ ... ]

where option can be:

    SUPERUSER | NOSUPERUSER
  | CREATEDB | NOCREATEDB
  | CREATEROLE | NOCREATEROLE
  | INHERIT | NOINHERIT
  | LOGIN | NOLOGIN
  | REPLICATION | NOREPLICATION
  | BYPASSRLS | NOBYPASSRLS
```

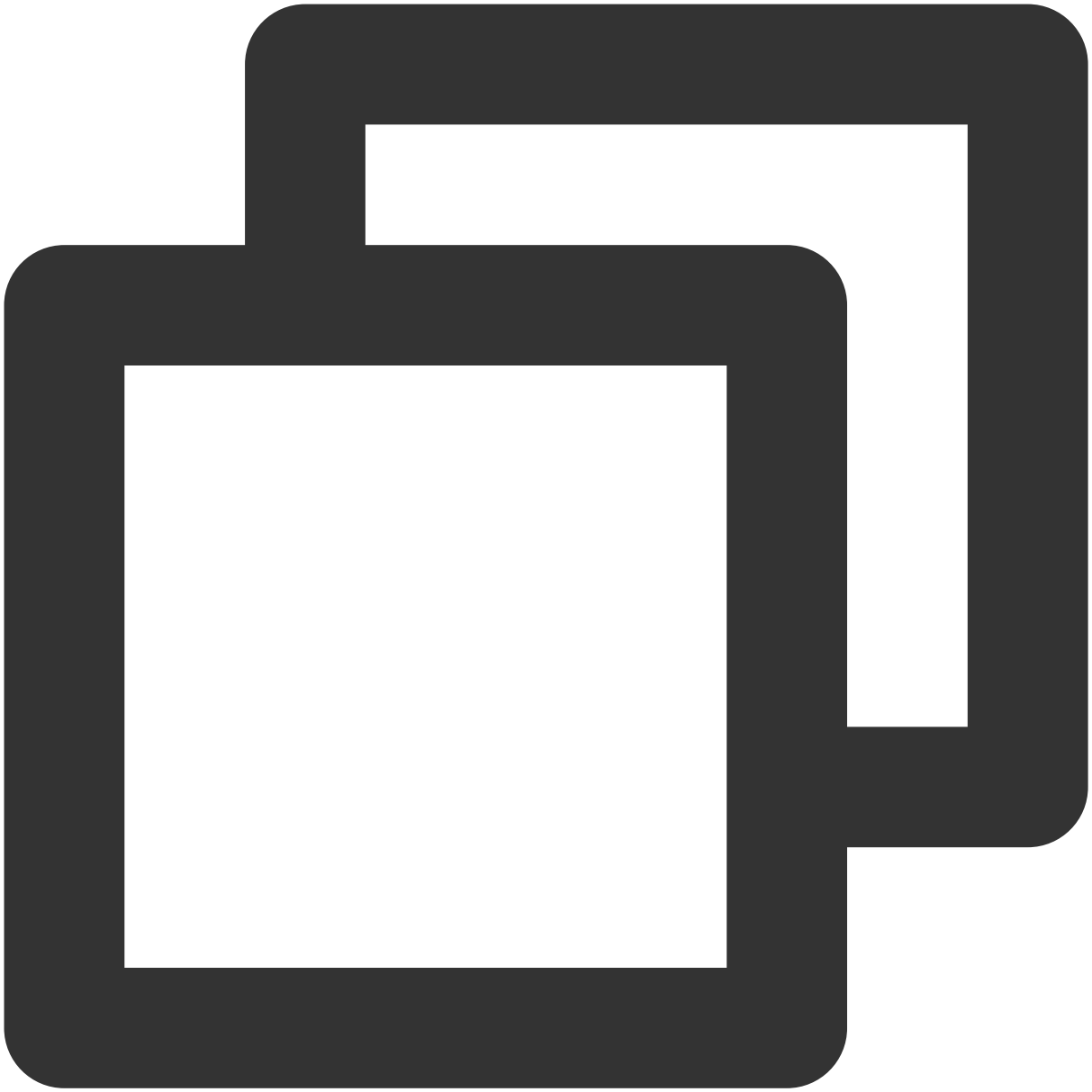
```
    | CONNECTION LIMIT connlimit
    | [ ENCRYPTED ] PASSWORD 'password' | PASSWORD NULL
    | VALID UNTIL 'timestamp'
```

赋予角色对象权限，详细可参考 官方文档：



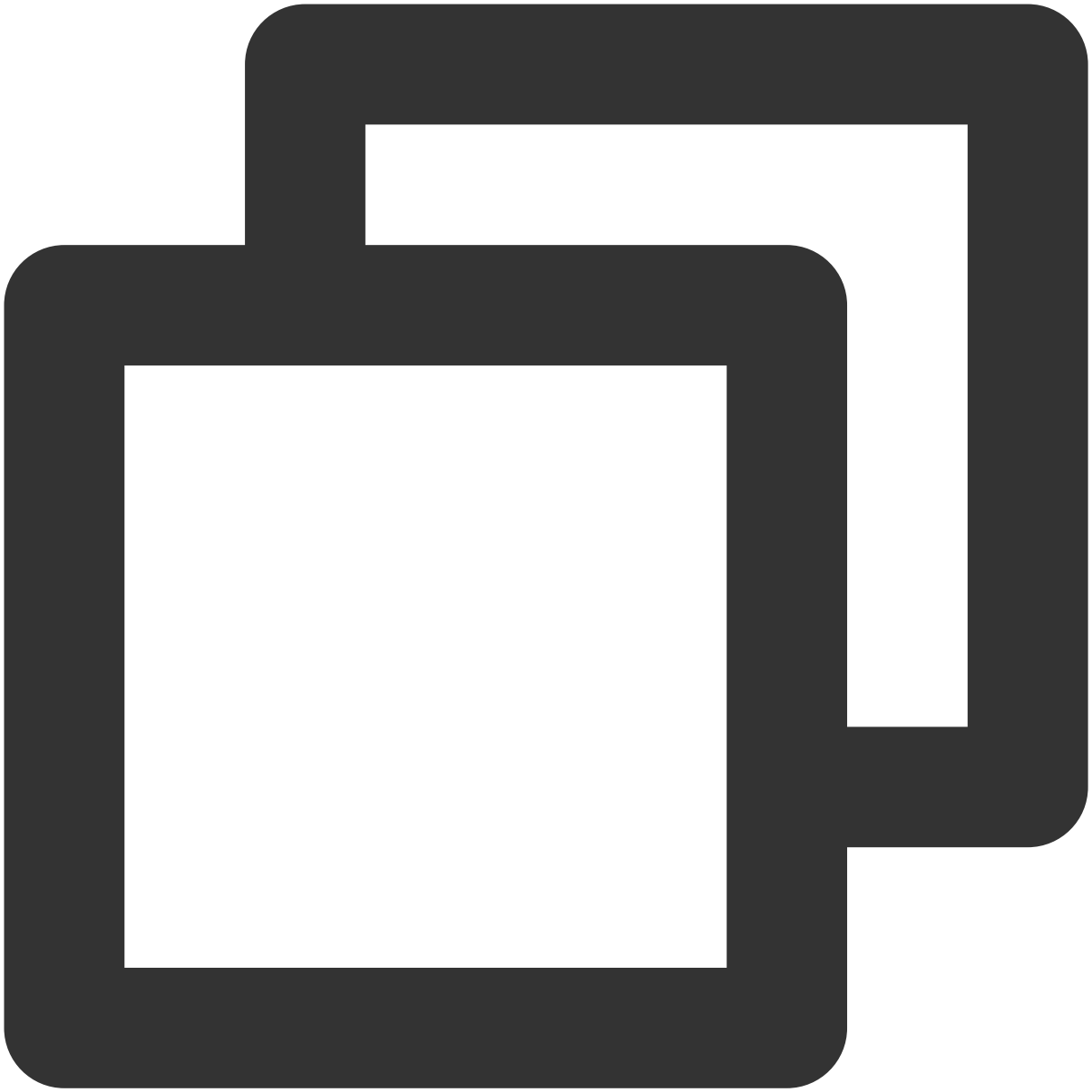
```
#简单写法
GRANT <权限> on <对象> to <角色>;
```

回收角色对象权限，详细可参考 官方文档：



```
#简单写法
REVOKE <权限> ON <对象> FROM <角色>;
```

将角色赋予其他角色：



```
#简单写法
GRANT <角色名> to <其他角色>;
```

控制台操作指引

最近更新时间：2024-08-09 15:13:16

本文主要内容为在控制台进行界面化的数据库账号操作。

创建账号

腾讯云数据库 PostgreSQL 未开放 superuser 角色属性和 pg_execute_server_program、pg_read_server_files、pg_write_server_files 角色供用户使用。但因为部分操作必须要求 superuser 进行，所以云数据库 PostgreSQL 提供了 pg_tencentdb_superuser 角色用以替代 superuser。更多 pg_tencentdb_superuser 介绍请参考 用户与权限操作。
控制台账号类型分为两类：pg_tencentdb_superuser 账号和普通账号。一个账号，只要是 pg_tencentdb_superuser 角色的成员，则为 pg_tencentdb_superuser 账号，否则为普通账号。
您可以登录 PostgreSQL 控制台，在实例列表，单击**实例 ID** 或**操作**列的**管理**，进入实例详情页面。单击**账号管理** > **创建账号**来进行新建操作。具体如下：

说明：

在控制台创建账号时，其账号名称需满足如下条件：1-16个字符，只能由字母、数字或下划线组成；不能为postgres；不能由数字、pg_ 或 tencentdb_ 开头；所有规则均不区分大小写。
控制台只支持大版本为9.5及以上的实例的账号的创建。低于该版本的实例请先升级实例大版本。

账号展示

说明：

1. 控制台账号列表可以展示数据库实例中所有状态正常和锁定中的账号。这些账号可以是控制台创建，也可以是使用其它客户端连上数据库创建。
2. 控制台只支持 user 的展示，不支持 role 的展示。
3. 只有在控制台创建的账号，管理系统才会记录其创建时间。请知悉。
4. 当一个账号的系统权限有 nologin 或者 CONNECTION 为0，则该账号的状态为锁定中。
5. 控制台仅能支持符合账号名称要求的数据库账号的展示，请知悉。

您通过控制台、其它客户端或者程序创建的数据库账号，均可以展示在账号列表。具体如下图所示：

修改账号

重置密码

您可以在控制台中重置已存在账号的登录密码。如果该账号当前已经登录数据库，则重置密码后对后续该账号的链接有效。单击**操作 > 重置密码**，具体如下图所示：

弹框如下所示：

修改备注

您可以在控制台修改账号的备注，单击**操作 > 更多 > 修改备注**，具体如下图所示：

弹框如下所示：

修改类型

控制台账号类型分为两类：pg_tencentdb_superuser 账号和普通账号。一个账号，只要是 pg_tencentdb_superuser 角色的成员，则为pg_tencentdb_superuser 账号，否则为普通账号。更多 pg_tencentdb_superuser 介绍请参考 用户与权限操作。

您可以在控制台修改账号类型，单击**操作** > **更多** > **修改类型**，具体如下图所示：

弹框如下图所示：

修改权限

账号对象权限说明

一个账号的对象权限由三部分组成：

从所属角色继承的权限：在 PostgreSQL 中，一个用户可以属于一个或多个角色，这些角色可以具有特定的权限。例如，一个角色可能有权限访问某个数据库或修改某个表。如果一个用户属于这个角色，那么他将继承这个角色的所有权限。

从 PUBLIC 继承的权限：如前面所述，PUBLIC 是一个特殊的预定义组，所有的用户都自动属于这个组。如果为 PUBLIC 组授予了某个权限，那么所有用户都将拥有这个权限。

被直接授予的权限：除了从角色和 PUBLIC 继承的权限外，一个用户还可以直接被授予权限。例如，数据库管理员可以直接授予一个用户权限，让他可以访问某个数据库或修改某个表。

这三个方面的权限可以相互覆盖。要想 revoke 掉一个 user 的某个权限，必须三个地方都必须 revoke 掉才行。而对象权限的授予与回收对于 OWNER 不生效，OWNER 对一个对象具备任何权限。

注意：

控制台的修改权限能力针对直接授予的权限。如果一个账号在回收权限之后该权限还在，您需要进一步排查下该用户是否是从 PUBLIC 组继承了权限或者是该对象的 OWNER。

您可以在控制台对一个账号同时赋予或者回收多个对象的多个不同权限，也可以批量对一个账号授予或者回收多个同类对象的多个不同权限。

对一个账号赋予或者回收多个对象的多个不同权限

单击**操作** > **修改权限**，弹框如下：

弹框如下：

注意：

由于账号的权限可以通过多个客户端或者程序来修改，因强烈建议您每次在控制台操作权限之前单击**刷新**按钮，拉取最新的权限。

注意：

PUBLIC 是一个特殊的预定义组，所有的用户都自动属于这个组。当为 PUBLIC 组授予权限时，实际上是为所有用户授予了权限，包括现有的和将来创建的用户。

如果某种对象的某种权限已经被 PUBLIC 组所拥有，要想 REVOKE 该权限，只有从 PUBLIC 组中去除该权限才能真正生效。

一旦某权限从 PUBLIC 组中 REVOKE ，则所有用户都将失去该权限，请谨慎操作。

为了帮助您在错误单击之后能迅速恢复至初始化状态，单击**重置**按钮，则本次界面操作将全部恢复至当前的账号权限状态，您可以在此基础上重新操作。

单击确定，系统将汇总您本次对当前选择的账号的即将发起的所有权限操作的汇总，便于您二次确认。具体如下图所示：

如上图所示，本次账号的权限修改操作涉及 nomal_1和 db_priv 两个 database，分别如下：

1. nomal_1数据库的操作涉及如下三个：

1.1 nomal_1数据库进行 REVOKE TEMPORARY 操作。

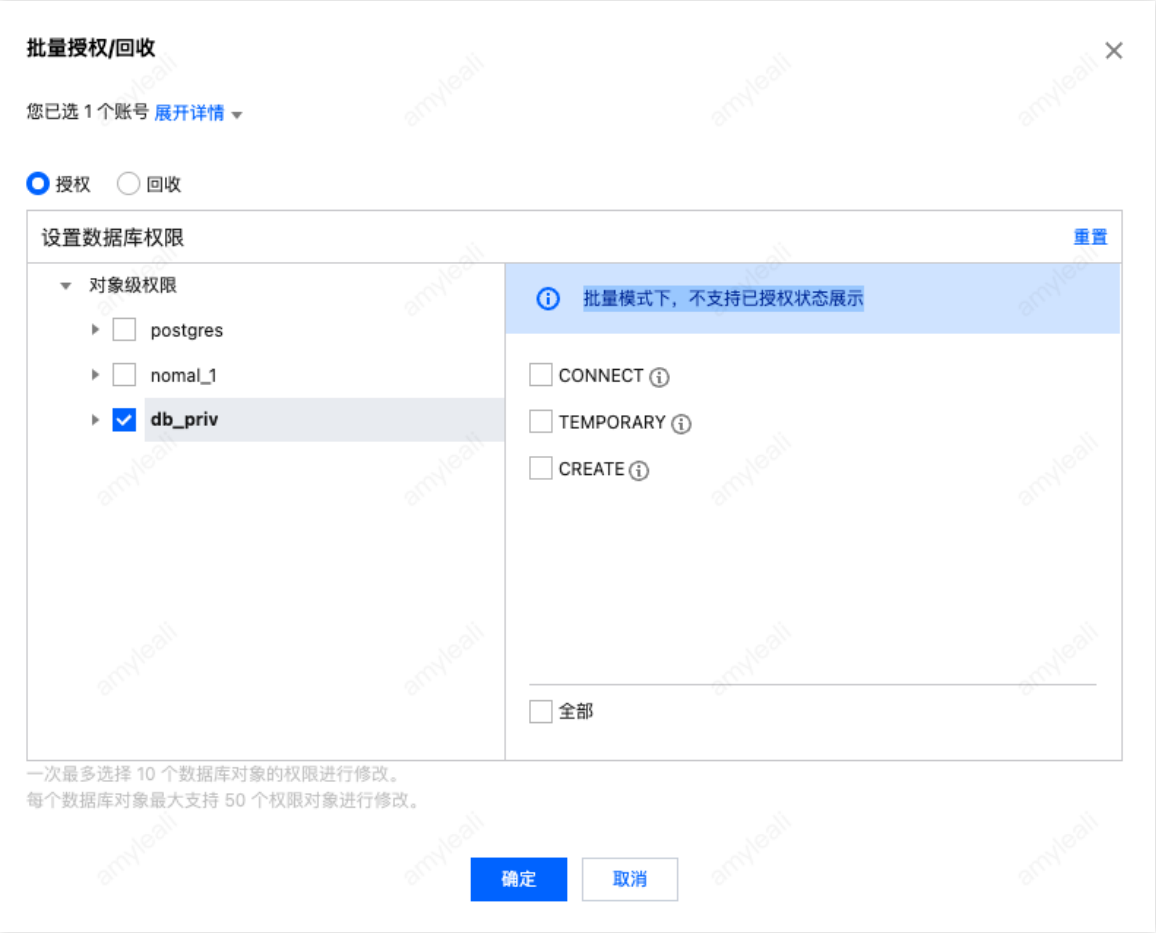
1.2 sche_priv 模式的 amyttt 表格的 id 字段进行 REVOKE INSERT 操作。

1.3 sche_priv 模式的 am_s 序列进行 REVOKE UPDATE 操作。

2. db_priv 数据库进行 REVOKE CREATE 操作。

对一个账号赋予或者回收多个同一类对象的多个不同权限

当您需要对一个账号同时操作同一类对象的多个权限时，您可以使用**批量授权/回收**功能。弹框如下：



注意：

批量授权/回收功能针对一个账号对于同一类对象的批量操作。不支持某个批量操作及操作了 database，又操作了 schema。

在批量模式下，因为权限可能各不一致，因此不支持已授权状态的展示。请谨慎操作。

删除账号

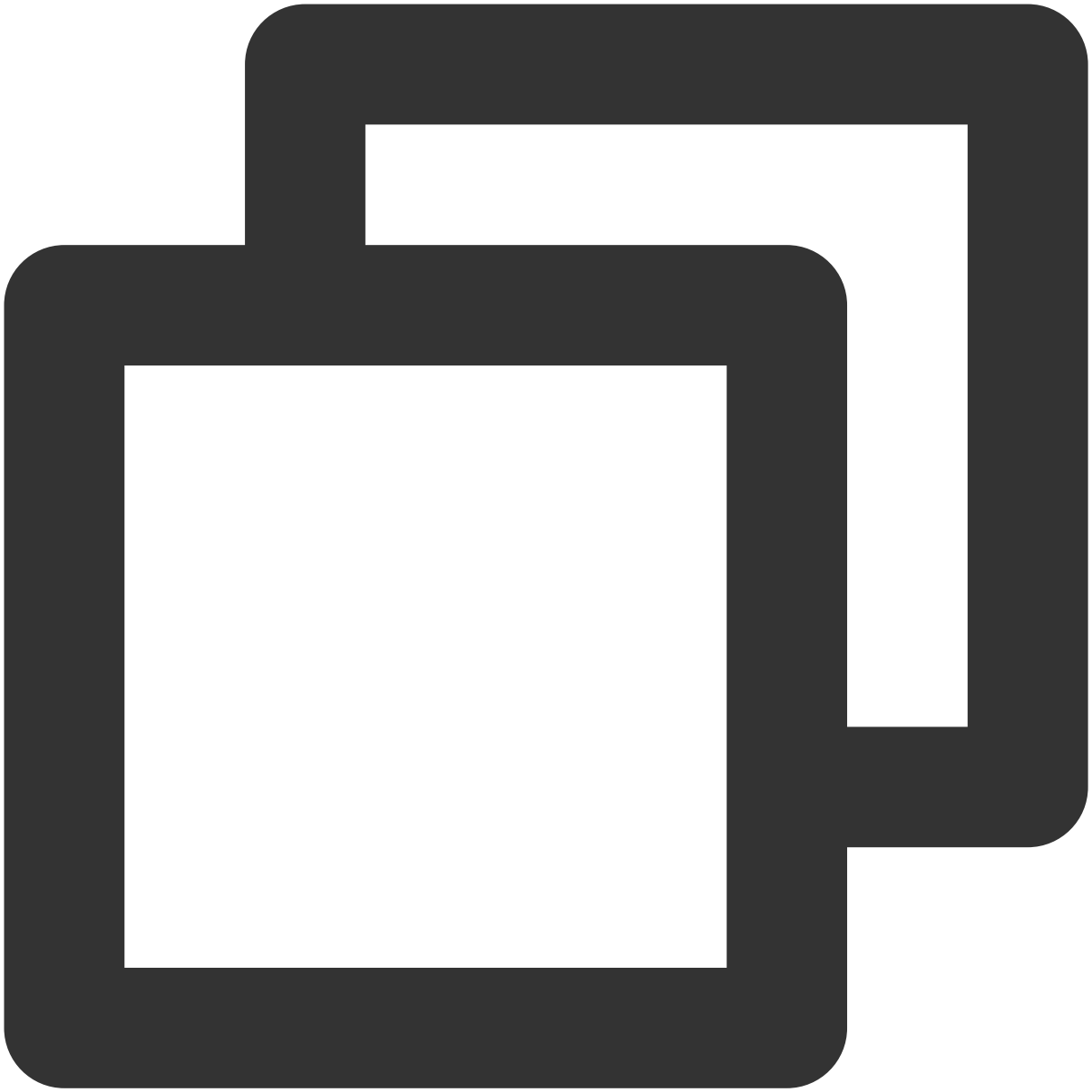
当您不再需要某个数据库账号时，控制台支持删除操作，请单击**操作** > **更多** > **删除账号**，具体如下图所示：

只有在账号没有任何对象权限时，该账号才允许被删除，否则删除账号报错，具体如下图所示：

如果数据库账号删除报错，我们可以查询该账号是否在某些对象上有相应的权限，或者是某些对象的 OWNER。如果是，则回收相应的权限或者替换对象的 OWNER 后再进行该账号的授权即可。PostgreSQL 中使用 aclitem 来表示一个具体的数据库对象上的权限。具体解释请参考 数据库权限概述。

查询某个账号是否具有某些对象的权限方法如下：

（1）查询该账号是否对某些 database 有权限，命令参考如下：



```
 nomal_1=> select datname,datacl from pg_database;
 datname   |                                  datacl
-----------+-----------------------------------------------------------------------
 template1 | {=c/postgres,postgres=CTc/postgres}
 template0 | {=c/postgres,postgres=CTc/postgres}
 postgres  | {=Tc/postgres,postgres=CTc/postgres,pg_tencentdb_superuser=C/postgres}
 nomal_1   | {=Tc/nomal_usr1,nomal_usr1=CTc/nomal_usr1}
 db_priv   | {=Tc/dbadmin,dbadmin=Tc/dbadmin}
(5 rows)
```

如上图所示，datacl 字段清晰的描述了账号对于 database 的相应权限。我们找到对应的账号的权限，然后回收相应的权限后再删除账号。

（2）查询该账号是否对某些 schema 有权限，您可以在每个 database 下查询，命令参考如下：

```
nomal_1=> select nspname,nspacl from pg_namespace;
      nspname        |                           nspacl
---------------------+-------------------------------------------------------------
 pg_toast            |
 pg_temp_1           |
 pg_toast_temp_1     |
 pg_catalog          | {postgres=UC/postgres,=U/postgres}
```

```
 public              | {postgres=UC/postgres,=UC/postgres}
 information_schema | {postgres=UC/postgres,=U/postgres,nomal_usr1=C/postgres}
 sche_priv           | {dbadmin=UC/dbadmin}
(7 rows)
```

如上图所示，datacl 字段清晰的描述了账号对于 schema 的相应权限。我们找到对应的账号的权限，然后回收相应的权限后再删除账号。

（3）查询该账号是否对某些对象有权限，您可以在每个 database 下查询，命令参考如下：

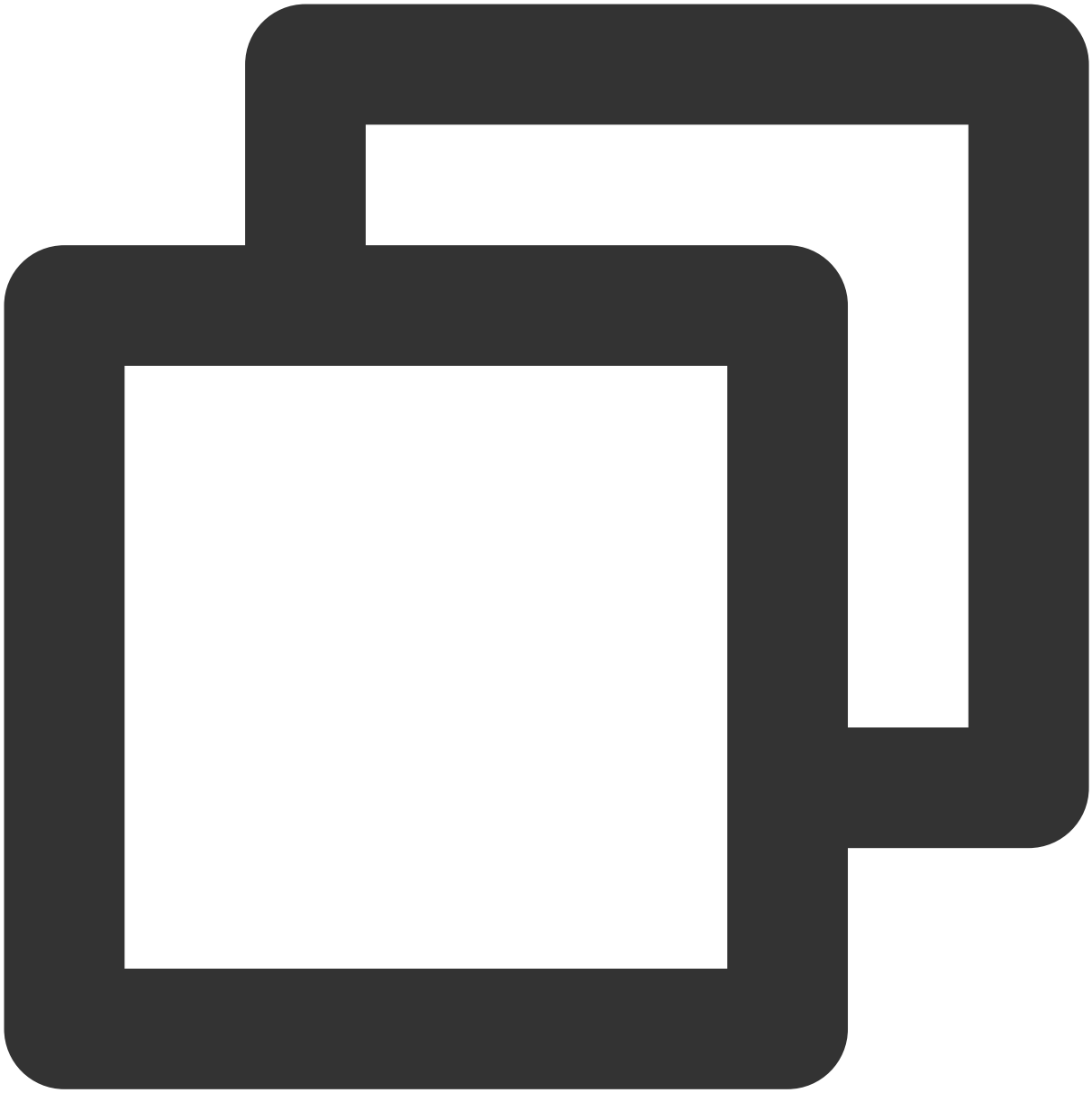
```
nomal_1=> select relname,a.typname as reltype ,relacl from pg_class  join pg_type a
```

查询某个账号是否是某个对象的 OWNER，示例如下：

（1）查询账号是否为某个 database 的 OWNER，示例如下：

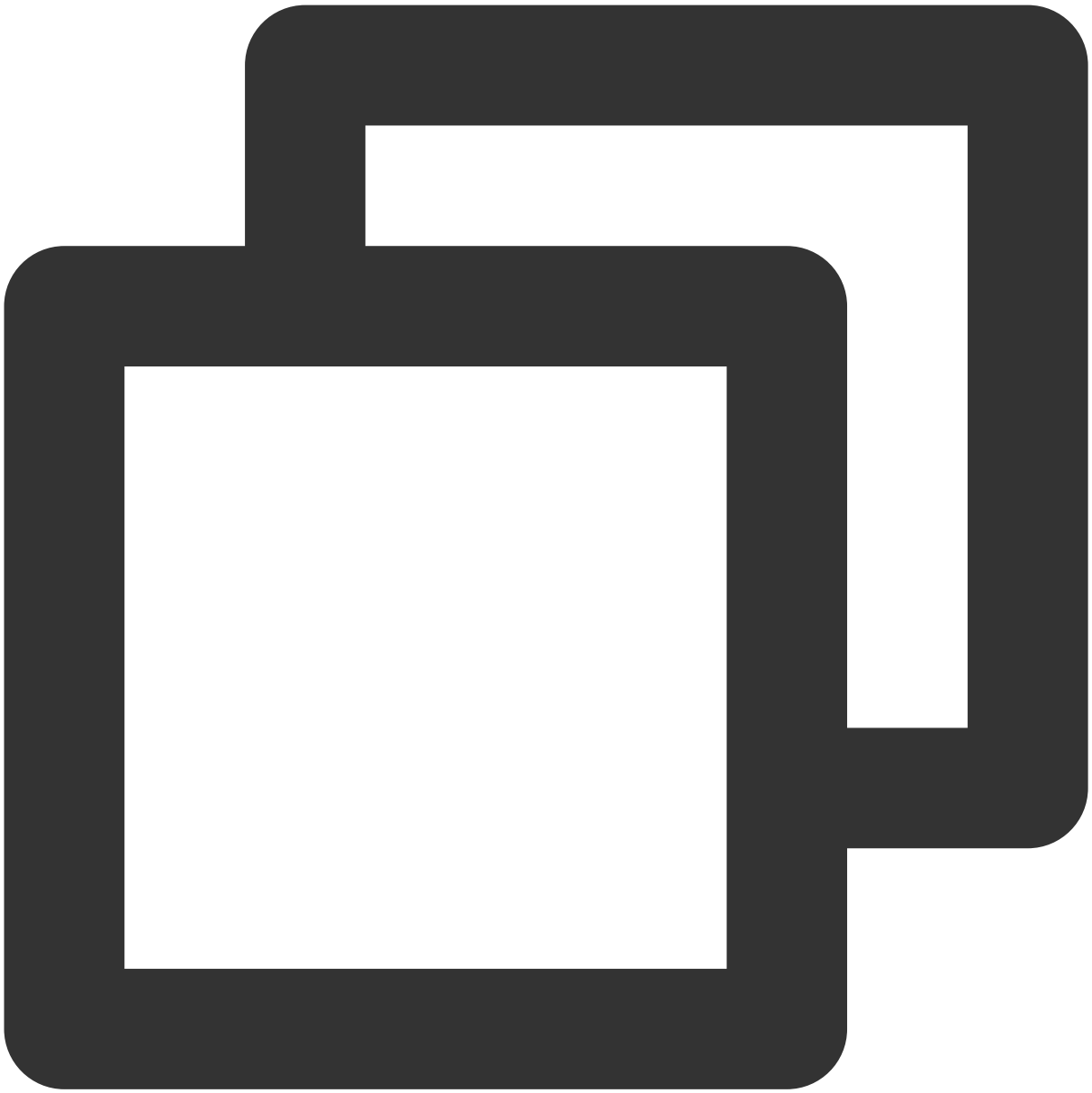
```
nomal_1=> SELECT r.rolname,d.datdba,datname AS database_name FROM pg_database d  le
  rolname    | datdba | database_name
------------+--------+---------------
 dbadmin     |  16398 | db_priv
 nomal_usr1 |  16401 | nomal_1
```

（2）查询账号是否为某个 schema 的 OWNER，示例如下：



```
nomal_1=>  SELECT n.nspname as schema_name, pg_catalog.pg_get_userbyid(n.nspowner)
    schema_name      | schema_owner
--------------------+--------------
 information_schema | postgres
 pg_catalog         | postgres
 pg_temp_1          | postgres
 pg_toast           | postgres
 pg_toast_temp_1    | postgres
 public             | postgres
 sche_priv          | dbadmin
```

（3）查询账号是否为某个对象的 OWNER，示例如下：



```
nomal_1=> select * from (select relname,relnamespace as schema_name ,a.typname as r
    relname      | schema_name |    reltype      |  owner
----------------+-------------+-----------------+---------
 pg_toast_16488 |          99 | pg_toast_16488 | dbadmin
 amyttt         |       16480 | amyttt         | dbadmin
 am             |        2200 | am             | dbadmin
 am_s           |       16480 | am_s           | dbadmin
 bug_id_seq     |        2200 | bug_id_seq     | dbadmin
 bug            |        2200 | bug            | dbadmin
```

```
 pg_toast_16507 |          99 | pg_toast_16507 | dbadmin
```

锁定账号

说明：

一个账号被执行锁定操作，管理系统会修改系统权限为 NOLOGIN，同时修改 CONNECTION 为0。

一个账号被执行锁定操作，则当前已经登录的链接会立即生效。

您可以锁定当前已经存在的账号，单击**操作** > **锁定账号**，具体如下图所示：

弹框如下所示：

一个账号被锁定之后，其状态会变为锁定中，具体如下图所示：

当一个账号被锁定之后，可以在控制台执行解锁操作，单击**操作** > **更多** > **解锁账号**，具体如下图所示：

说明：

当一个账号被执行解锁操作，则管理系统会修改账号的系统权限为 LOGIN，同时修改 CONNECTION 为-1。

导出列表

控制台支持两种账号列表导出，分别为下载当前页数据，下载所有数据。如下图所示：

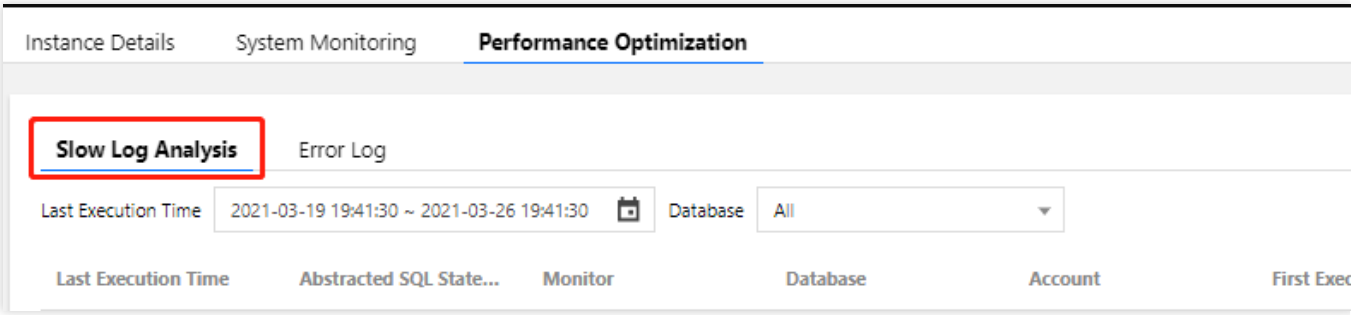
实例详情
系统监控
参数设置
账号管理
安全组
备份恢复
性能优化
只读实例
数据加密
版本升级
创建账号
账号名
类型
创建时间
修改时间
状态
备注
pg_tencentdb_superuser
2024-06-27 19:28:07
2024-06-27 19:28:07
正常
普通用户
0000-00-00 00:00:00
2024-06-26 21:19:20
锁定中
pg_tencentdb_superuser
2024-06-26 20:17:29
2024-06-26 20:17:29
正常
dfd
共 3 项
20

数据库优化
慢查询分析

最近更新时间：2024-01-22 16:22:57

功能说明

默认数据库对超过1s的 SQL 语句查询判断为“慢查询”，对应语句称为“慢查询语句（俗称慢 SQL）”，而数据库管理员（DBA）对慢查询语句进行分析并找到慢查询出现原因的过程叫做慢查询分析。
云数据库 PostgreSQL 控制台 在实例管理页的**性能优化**模块下，提供慢查询分析能力。如下图：



监控视图

控制台中的两个图表为监控视图，可以直观方便地查看数据库慢 SQL 的相关信息。
慢查询与其他监控组合视图：支持通过图表化形式查看慢查询指标以及与其他指标联合对比查看的能力。可以支持的其他指标有：CPU 利用率，QPS，请求数，读请求数，写请求数，其他请求数，缓冲区缓存命中率，平均执行时延。
慢 SQL 耗时分布： 可以通过不同时间段的慢 SQL 进行查看慢查询主要分布在什么时段区间内。

慢 SQL 列表

慢 SQL 列表可以实时查看数据库中所产生的慢 SQL 语句，列表根据时间降序排列，最新产生的慢 SQL 会自动生成显示在第一行。
支持查看的慢 SQL 字段：执行时间，慢 SQL 语句，总耗时，客户端 IP 地址，数据库名，执行用户。
注意：
慢 SQL 列表默认保存最近7天的慢 SQL 数据与最大50GiB的日志总量，当哪一个条件率先满足，则会自动删除超出要求的日志。

当单条慢 SQL 大于20KB，将无法从控制台上查看，请 提交工单，联系腾讯云为您提供。

慢 SQL 统计分析

慢 SQL 统计分析可以根据指定时间范围内的所有慢 SQL 经过系统的参数抽象后，对同类 SQL 进行聚合分析得到的慢 SQL 分析信息。其中包含多种字段信息。

最后执行时间：在统计范围内，该抽象语句最后一次出现的时间，由于某些预计执行较长，我们统一按语句执行的 begin_time 记录。

抽象后的 SQL 语句：去掉慢 SQL 中常数后的语句，抽象后的语句可以将同类同性质的语句进行汇总统计，以方便您更好的分析。

数据库：该语句调用了哪个数据库。

帐号：该语句是使用哪个帐号来运行的。

客户端 IP 地址：语句在哪些客户端上执行过。

首次执行时间：在统计范围内，该慢 SQL 第一次出现的时间（抽象后汇总后，可能有很多条记录）。

执行总时间：在统计范围内，慢查询语句查询总耗时。

平均执行时间：慢查询语句总时间除以总次数的平均时间。

最小执行时间：所有该条抽象语句中，慢查询语句出现的最小时间；用于帮助判断该语句是否偶发性事件。

最大执行时间：所有该条抽象语句中，慢查询语句出现的最大时间；用于帮助判断该语句是否偶发性事件。

总时间占比：在统计范围内，慢查询语句占所有慢查询语句时间的占比。

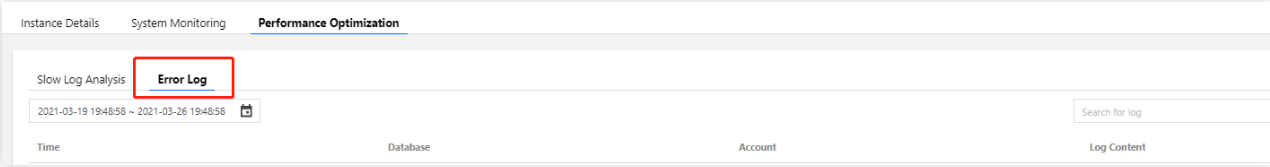
版权所有：腾讯云计算（北京）有限责任公司

错误日志

最近更新时间：2024-01-22 16:22:57

功能说明

数据库运行过程中，因操作、SQL、系统运行错误等原因产生的日志叫做错误日志，错误日志常用于开发者找出业务系统或数据库出现问题的原因。
云数据库 PostgreSQL 在实例管理页的**性能优化**模块下，提供错误日志查看能力。如下图：



错误日志默认设置

错误日志功能：默认开启
错误日志记录级别：log_min_error_statement=ERROR
分析数据输出延迟：1分钟 - 5分钟
日志记录时长：7天（最大最近10000条）

参数管理
设置实例参数

最近更新时间：2024-01-22 16:22:57

您可以通过 PostgreSQL 控制台 查看和修改部分参数，并可以在控制台查询参数修改记录。

注意事项

为保证实例的稳定，控制台仅开放部分参数的修改，控制台的参数配置页面展示的参数即为用户可以修改的参数。如果修改的参数需要重启实例才生效，系统会提示您是否重启，建议您在业务低峰期操作，并确保应用程序具有重连机制。

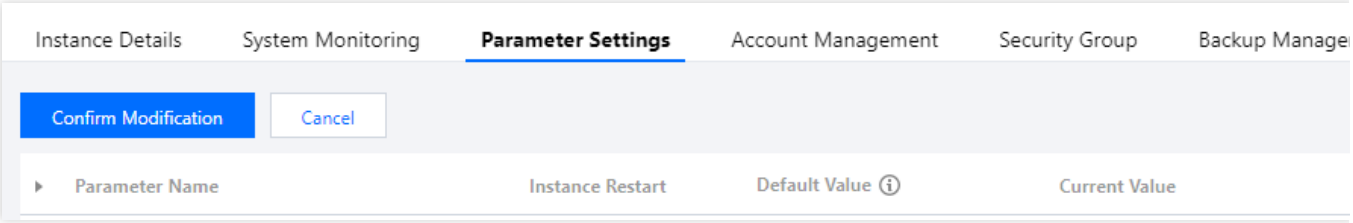
注意：

部分参数属于重点参数，修改会影响数据库实例使用行为或者影响数据库正常运行，请谨慎修改。

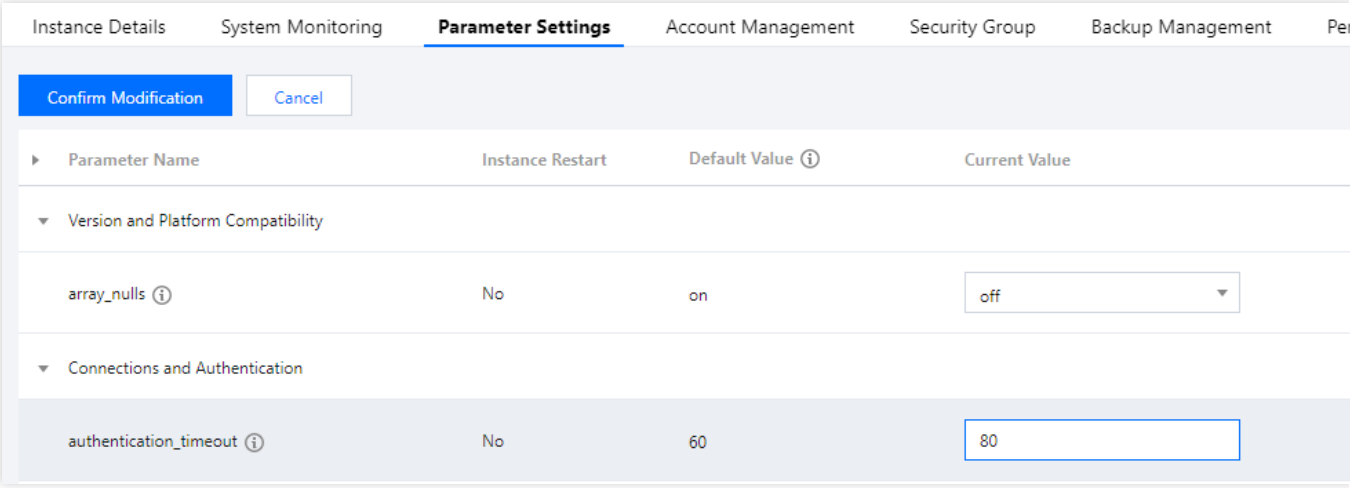
通过参数列表修改参数

批量修改参数

1. 登录 PostgreSQL 控制台，在实例列表，单击实例 ID 或**操作**列的**管理**，进入实例管理页面。
2. 在实例管理页面，选择**参数设置**页，单击**批量修改参数**。



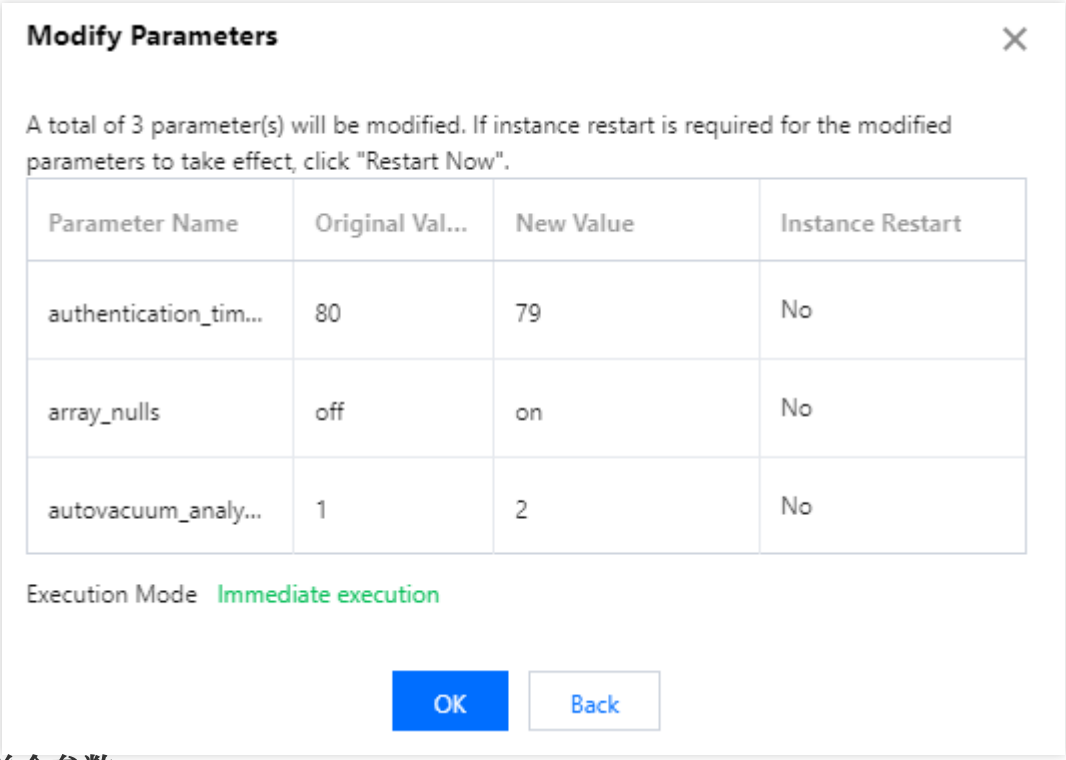
3. 在**参数运行值**列，选择需要修改的参数进行修改，确认无误后，单击**确认修改**。



4. 在弹出的对话框，请确认参数修改的值是否正确，确认无误后单击**确定**。

说明：

若修改的参数需要重启，则按钮名为**立即重启**。单击**立即重启**，实例将会进行重启。完成重启后，修改的参数才会生效。



修改单个参数

1. 登录 PostgreSQL 控制台，在实例列表，单击实例 ID，进入实例管理页面。
2. 在实例管理页面，选择**参数设置**页，选择目标参数所在行，在**参数运行值**列，单击

修改参数值。

Parameter Name	Instance Restart	Default Value
Version and Platform Compatibility		
Connections and Authentication		
authentication_timeout	No	60

3. 根据**参数可修改值**列的提示，输入目标参数值，单击 ✓ 保存，单击 ✕ 可取消操作。
4. 在弹出的对话框，请确认参数修改的值是否正确，确认无误后单击**确定**。

说明：

若修改的参数需要重启，则按钮名为**立即重启**。单击**立即重启**，实例将会进行重启。完成重启后，修改的参数才会生效。

查看参数修改记录

1. 登录 PostgreSQL 控制台，在实例列表，单击实例 ID，进入实例管理页面。
2. 在实例管理页面，选择**参数设置**页，单击右侧的**最近修改记录**。
3. 在最近参数修改记录页，可查看近期参数修改记录。

Recently Modified Parameters

Parameter Name	Original Value	Modified to	Modification Status	Modificati
authentication_timeout	60	80	Successful	2021-10-14
autovacuum_analyze_scale_factor	0.1	1	Successful	2021-10-14
autovacuum_analyze_threshold	50	51	Successful	2021-10-14
array_nulls	on	off	Successful	2021-10-14

参数值限制

最近更新时间：2024-01-22 16:22:57

因资源规格等原因，部分实例参数修改会受到限制。本文主要介绍受到实例规格限制的参数与其限制值。

注意事项

以下参数值设置会占用系统资源，为避免数据库使用受到参数值的影响，请按需配置。

注意：

部分参数属于重点参数，修改会影响数据库实例使用行为或者影响数据库正常运行，请谨慎修改。

与规格相关的参数值限制

规格\\参数	max_replication_slots	max_wal_senders	max_worker_processes	max_logical_replicati
1C2GiB	[10-100]	[27-150]	[4-300]	[4-150]
2C4GiB	[10-100]	[27-150]	[4-300]	[4-150]
2C6GiB	[10-150]	[27-200]	[4-400]	[4-200]
4C8GiB	[10-150]	[27-200]	[4-400]	[4-200]
4C16GiB	[10-150]	[27-200]	[4-400]	[4-200]
6C24GiB	[10-200]	[27-250]	[4-500]	[4-250]
8C32GiB	[10-200]	[27-250]	[4-500]	[4-250]
8C48GiB	[10-200]	[27-250]	[4-500]	[4-250]
12C64GiB	[10-400]	[27-450]	[4-900]	[4--450]
16C96GiB	[10-400]	[27-450]	[4-900]	[4--450]
20C128GiB	[10-500]	[27-450]	[4-900]	[4--450]
28C240GiB	[10-600]	[27-650]	[4-1300]	[4--650]
48C480GiB	[10-600]	[27-650]	[4-1300]	[4--650]

版权所有：腾讯云计算（北京）有限责任公司

版权所有：腾讯云计算（北京）有限责任公司

参数模板

最近更新时间：2024-05-11 14:24:51

通过云数据库 PostgreSQL 控制台您可以创建自定义参数模板，根据自己的业务场景，实现批量参数设置。
您可以使用数据库参数模板管理数据库引擎的参数配置。数据库参数组就像是引擎配置值的容器，这些值可应用于一个或多个数据库实例。
参数模板支持如下功能，用户可登录 PostgreSQL 控制台，在左侧导航选择**参数模板**页面查看参数：
支持新建模板，生成自定义的参数优化方案。
支持从 PostgreSQL 配置文件 `**.conf` 导入生成模板。
支持从参数设置保存为模板。
单个或多个实例进行参数设置时，支持从模板导入。
支持两个参数模板之间进行对比。

注意事项

已使用参数模板的数据库实例并不会随着参数模板更新而同步更新，需要手动批量更新数据库实例。
如果您需要将新的参数应用到批量实例上，可通过批量参数设置时导入模板来重新应用。

新建参数模板

当您想使用自己的数据库参数模板时，只需创建一个新的数据库参数模板，修改所需的参数并应用于您的数据库实例，就可以使用新的数据库参数模板。
1. 登录 PostgreSQL 控制台，在左侧导航选择**参数模板**页面，单击**创建模板**。
2. 在弹出的对话框，配置如下参数，单击**创建并设置参数**。
模板名称：输入参数模板的名称，参数模板名称需具有唯一性。
引擎：选择对应的数据库引擎，如 PostgreSQL 或者 SQL Server 兼容。
数据库版本：选择需要的数据库版本。
模板描述：输入参数模板的简要说明。

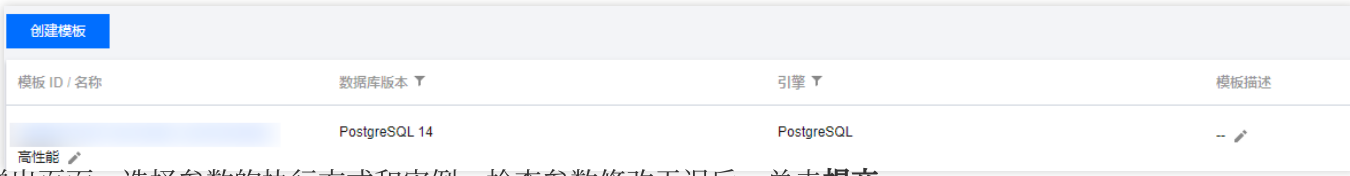
3. 创建完成后，您可以在模板详情页修改、导入、导出参数等。

应用参数模板于实例

注意：

应用参数模板到多个实例上时，请确认参数是否适用这些实例，请您谨慎操作。

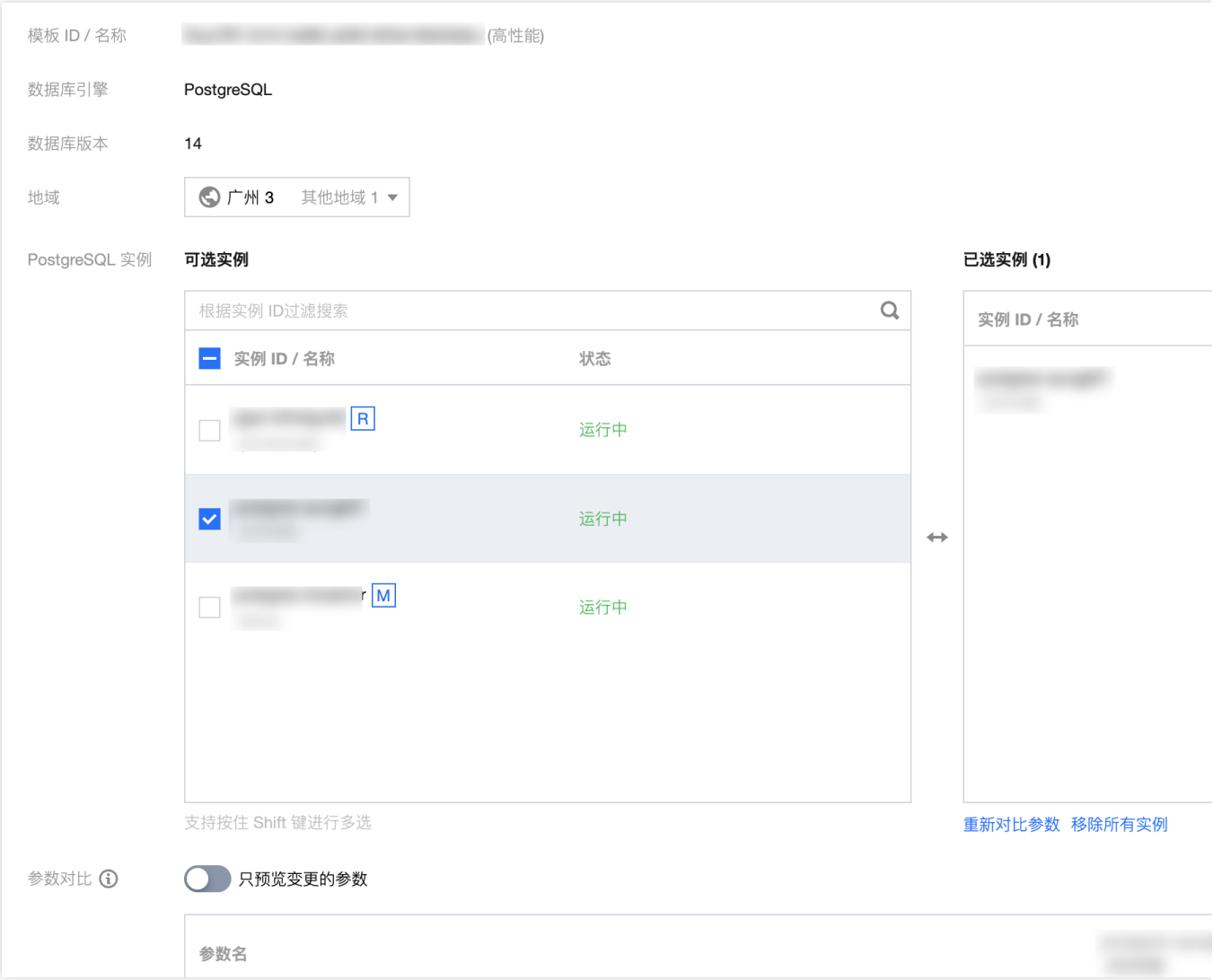
1. 登录 PostgreSQL 控制台，在左侧导航选择**参数模板**页面。
2. 在参数模板列表中，找到需要应用的模板，单击**应用到实例**。



3. 在弹出页面，选择参数的执行方式和实例，检查参数修改无误后，单击**提交**。

PostgreSQL 实例：选择对应地域下，需要应用参数模板的实例。

参数对比：可查看所选实例参数修改前、后的配置。



复制参数模板

当您已创建一个数据库参数模板，并且想在新的数据库参数模板中包含该组中的大部分自定义参数和值时，复制参数模板是一个方便的解决方案。

方式一：通过已有参数模板复制参数模板

1. 登录 PostgreSQL 控制台，在左侧导航选择**参数模板**页面，单击模板 ID 或**操作**列的**查看详情**，进入模板详情页。
2. 在模板详情页，单击**另存为模板**。

3. 在弹出的对话框，配置如下参数：

模板名称：输入参数模板的名称，在参数模板名称需具有唯一性。

模板描述：输入参数模板的简要说明。

4. 确认无误后，单击**确定**即可将当前参数模板另存为新的参数模板，完成复制操作。

方式二：通过实例的参数设置功能复制参数模板

1. 登录 PostgreSQL 控制台，在左侧导航选择**实例列表**页面，单击实例 ID 进入管理页。
2. 选择**参数设置**页，单击**另存为模板**。
3. 在弹出的对话框，配置如下参数：

模板名称：输入参数模板的名称，在参数模板名称需具有唯一性。

模板描述：输入参数模板的简要说明。

4. 确认无误后，单击**确定**即可将当前参数模板另存为新的参数模板，完成复制操作。

修改参数模板的参数值

1. 登录 PostgreSQL 控制台，在左侧导航选择**参数模板**页面，单击模板 ID，进入模板详情页。
2. 在模板详情页，单击**批量修改参数**、或在参数运行值列，单击“修改参数值”图标。

导入参数模板

1. 登录 PostgreSQL 控制台，在左侧导航选择**参数模板**页面，单击模板 ID，进入模板详情页。
2. 在模板详情页，单击**导入参数**。

注意：

选择导入参数时，在从本地文件导入页中，选择参数配置文件时，需要注意配置文件的格式与 PostgreSQL 数据库服务器的配置文件格式一致或者使用导出参数的文件模板，否则提示导入不成功。

3. 在弹出的对话框，选择文件，单击**导入并覆盖原有参数**。

导出参数模板

方式一

1. 登录 PostgreSQL 控制台，在左侧导航选择**参数模板**页面。
2. 在参数模板列表，在所需模板的**操作**列，单击**导出**。

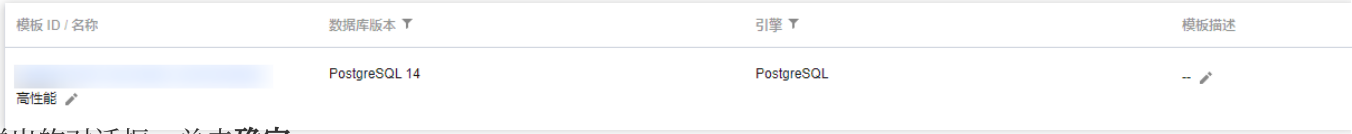
方式二

1. 登录 PostgreSQL 控制台，在左侧导航选择**参数模板**页面，单击模板 ID，进入模板详情页。
2. 在模板详情页，在上方单击**导出参数**。

删除参数模板

当您创建了多余的参数模板或者确定该参数模板已无使用价值，可通过删除参数模板功能删除不再使用的参数模板。

1. 登录 PostgreSQL 控制台，在左侧导航选择**参数模板**页面。
2. 在参数模板列表，在所需模板的**操作**列，单击**删除**。



3. 在弹出的对话框，单击**确定**。

日志管理及分析
运行日志管理

最近更新时间：2024-03-20 14:27:39

本文为您介绍如何管理 pg_log 保留时长。

PG_LOG 介绍

pg_log 一般是记录数据库的状态信息，例如：错误信息、慢查询 SQL、数据库的启动关闭信息等。该日志会按大小和时间自动切割，当前云数据库 PostgreSQL 的 pg_log 默认保留30天。pg_log 会占用数据库实例的存储空间，您可以基于实际需要来修改保留时长。

说明：

数据库实例的慢日志和错误日志默认保留7天，修改 pg_log 保留时长不影响慢日志和错误日志保留时长。

修改 PG_LOG 保留时长

您可以基于实际需要修改 pg_log 保留时长，系统当前支持7天和30天两种。具体操作如下：

1. 登录 云数据库 PostgreSQL 控制台。
2. 在实例列表找到需要修改的实例，单击**操作** > **管理**，进入实例详情。
3. 在实例详情页找到**参数设置**，在右上方搜索框中搜索需要设置的参数 log_filename 并更新。

log_filename 参数说明如下：

参数值	说明
postgresql_%a_%H.log	选择该值，日志保留7天。
postgresql_%d_%H.log	选择该值，日志保留30天，系统默认保留30天。

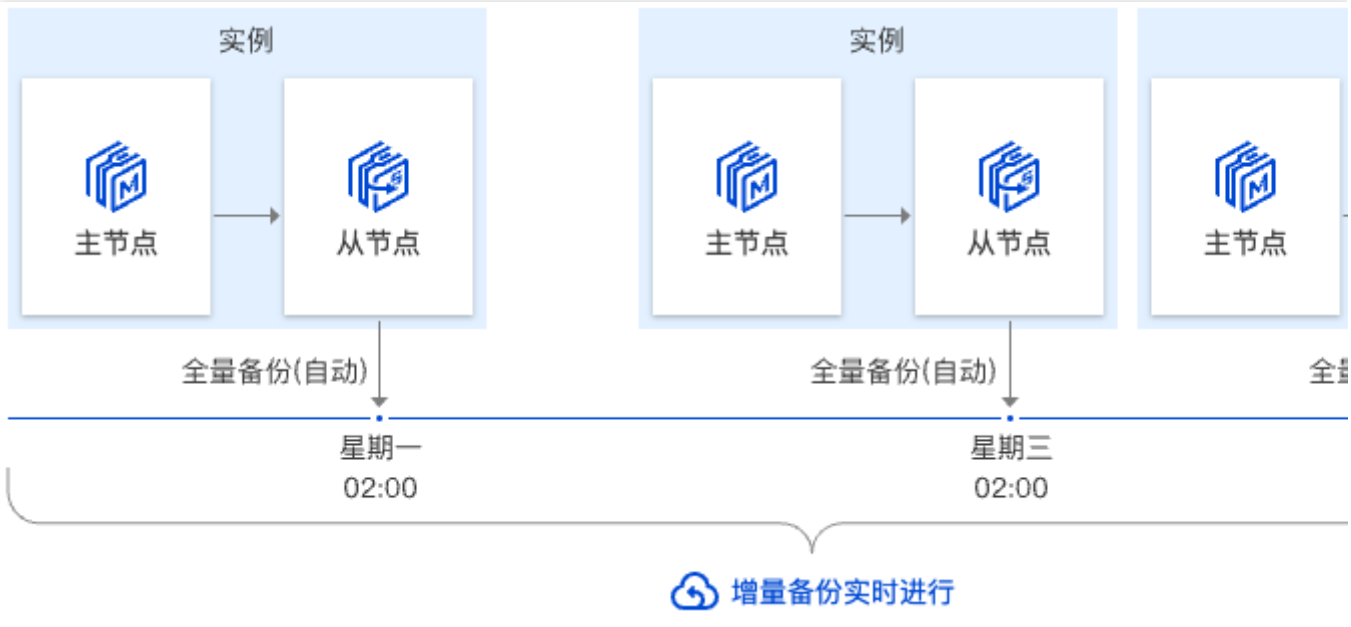
备份与恢复
备份原理和方案

最近更新时间：2024-03-20 14:30:47

云数据库 PostgreSQL 支持**自动全量备份**、**手动全量备份**和**日志备份**，您可以通过自动备份设置来定期执行数据库备份，当发生数据库故障或者误删除时，您可以通过已存储的备份文件恢复数据库。云数据库 PostgreSQL 支持以压缩的方式进行存储数据库备份，压缩后备份文件约为压缩前文件的30%（具体压缩比跟实例存储数据强相关，重复数据越多，压缩比越大）。

备份原理

对于双机高可用（一主一从）架构，当备份任务被触发后，系统会从实例的备节点拉取数据，并压缩上传到对象存储服务上。备份空间不会占用实例的磁盘空间。当发生数据库故障或者误删除时，您可以通过已存储的备份文件恢复数据库。恢复数据时可以采用克隆的方式，具体请参考 克隆实例，也可以自行下载备份文件进行恢复，具体请参考 在云服务器上恢复 PostgreSQL 数据。原理如下图所示：



备份方案

操作类型	备份类型	操作详情
数据备份	自动全量备份	系统会根据您的自动备份设置在指定的时间内触发全量的数据备份，备份方式为物理备份，物理备份方式备份速度快，恢复效率更高。

		产生的全量自动备份数据基于您设定的数据备份保留时间而保留，该备份数据不会随着实例销毁而删除，过期后将自动删除，如此可以满足您延期保存备份数据以防止因误删实例造成严重影响的需求。因全量备份占用备份存储空间，如您需要，可以及时删除。 一周之内自动备份数据不允许删除，一周之外的自动备份数据可根据需求灵活删除。备份数据删除后将无法恢复，请谨慎操作。
	手动全量备份	您可以在控制台基于应用需要进行手工备份，手工备份的任务发起后，系统将在1分钟内使用物理备份的方式进行全实例备份。 手工备份的过期时间为发起后的一周，因手工备份占用备份存储空间，如您需要，可以及时删除手工备份。
	增量备份	增量备份为 WAL 日志备份，系统默认自动开启增量备份，不可关闭。增量备份基于您设定的数据备份保留时间而保留。系统会基于数据库产生的 WAL 日志进行实时备份。因增量备份占用备份存储空间，如您需要，可以及时删除。 一周之内增量备份数据不允许删除，一周之外的增量备份数据可根据需求灵活删除。增量备份数据删除后无法进行任意时间点恢复，请谨慎操作。
备份文件下载	下载全量备份	支持本地浏览器下载和按地址下载。
	下载增量备份	支持本地浏览器下载和按地址下载。
通过云函数转存历史备份	通过云函数转存	支持通过云函数来转存历史备份数据，具体请参考 使用云函数转存 PostgreSQL 历史备份。

备份费用

云数据库 PostgreSQL 备份文件以压缩包形式进行存储，压缩后备份文件约为压缩前文件的30%左右。单实例备份空间以购买容量进行赠送，大多数实例不需要付费。具体收费规则请参考 备份空间收费说明。

备份数据

最近更新时间：2024-04-09 10:50:43

本文为您介绍通过 PostgreSQL 控制台备份设置及下载备份文件的操作。

操作场景

云数据库 PostgreSQL 高可用版目前支持物理备份的方式：

全量备份：每天备份一次，备份时间为01:00。

增量备份：当产生了 xlog，15分钟会备份一次；或者当产生 xlog，并且已经积压到60个 xlog 时备份一次。

数据文件保留时间：支持设置保存7～1830天的备份数据。

手工备份：用户自助发起备份任务，备份任务在您点击后1分钟内发起。

操作步骤

1. 登录 PostgreSQL 控制台，单击实例 ID，进入实例管理页面。
2. 在实例管理页面，选择**备份恢复**页，选择**备份列表**或**日志备份列表**，选择需要下载的备份，在**操作**列单击**下载**。

说明：

备份列表的备份数据即全量备份；**日志备份列表**的备份数据即增量备份。

3. 在弹出的对话框，提供内网 VPC 网络地址和本地下载两种下载方式。

说明：

为保证数据安全，地址有效期为12小时，过期后请刷新页面重新获取，VPC 网络地址请在 VPC 网络进行访问。

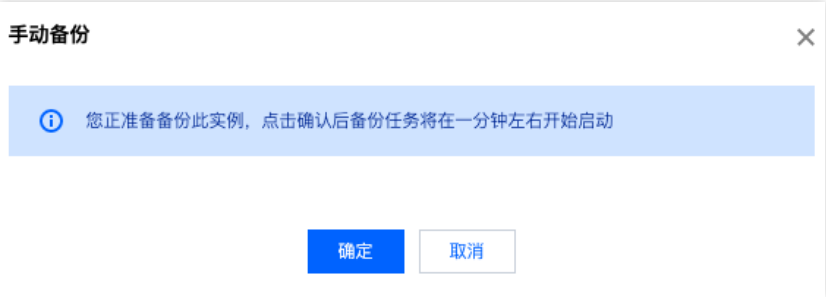
4. 用户可以自助按需发起备份任务。单击**备份恢复**页面的**手工备份**按钮发起手工备份。

说明：

实例在执行每天的自动备份任务期间，无法发起手工备份。

手工备份可在备份列表删除，释放掉备份空间，避免空间浪费和占用，如果没有手动删除则会一直保留至数据库实例下线。

5. 在**手动备份**的对话框中，单击**确定**。



说明：

手动备份创建时间跟实例实际容量大小相关，如果容量大，则手动备份产生的时间更长。

热点问题

超出备份保留时间的备份还可以下载或者还原吗？

到期后的自动备份集会自动删除，无法进行下载还原。您可以在控制台通过手动备份实例数据，手动备份会一直保存。

备份可以手动删除吗？

7天内的自动备份不允许删除，7天外的自动备份可以按需删除，系统也会基于自动备份保留时长自动删除。手动备份可在备份列表手动删除，若没有删除手动备份将会一直保留。

数据和日志备份是否可以关闭？

不可以关闭。但您可以通过减少备份频率和删除不必要的手工备份数据来降低备份空间占用量。

如何减少备份空间开销？

删除不必要的手动备份数据。
降低非核心业务的数据自动备份频率（一周至少备份2次）。

下载备份

最近更新时间：2024-04-09 10:57:17

本文为您介绍通过 PostgreSQL 控制台下载备份的操作。

注意事项

内网地址与本地下载地址有效期均为12小时，过期后请主动刷新页面重新获取。
使用 wget 下载时需要对 URL 添加英文引号。
下载文件后需要使用 zstd 进行解压，如没有该工具，需要进行安装。

操作步骤

数据备份下载

1. 登录 PostgreSQL 控制台，选择地域，单击**实例 ID**，进入实例管理页面。
2. 在实例管理页面，选择**备份恢复**页，单击**备份列表**。

3. 在备份列表下选择需要下载的备份，在其**操作**列单击**下载**。
4. 在下载窗口，提供内网 VPC 网络地址和本地下载两种下载方式。

说明：
推荐您复制下载地址，并 登录到云数据库所在 VPC 下的 CVM（Linux 系统） 中，运用 wget 命令进行内网高速下载，更高效。

5. 您也可以单击控制台左侧的**数据库备份**栏，进入**备份列表**，单击**数据备份列表**进行数据备份下载。

日志备份下载

1. 登录 PostgreSQL 控制台，选择地域，单击**实例 ID**，进入实例管理页面。
2. 在实例管理页面，选择**备份恢复**页，单击**日志备份列表**。

3. 在日志备份列表下选择需要下载的备份，在其**操作**列单击**下载**。
4. 在下载窗口，提供内网 VPC 网络地址和本地下载两种下载方式。

5. 您也可以单击控制台左侧的**数据库备份**栏，进入**备份列表**，单击日志**备份列表**进行数据备份下载。

克隆实例

最近更新时间：2024-01-22 16:22:57

如您需要进行数据恢复，可使用克隆 PostgreSQL 实例功能，从备份中快速恢复实例数据至新购的 PostgreSQL 实例。本文为您介绍如何通过云数据库 PostgreSQL 控制台实现克隆操作。

操作场景

云数据库 PostgreSQL 提供克隆实例功能，支持通过克隆将实例恢复到日志备份保留时间内的任意时间点，并且支持恢复到指定物理备份的备份集。克隆会先根据用户所选的时间点并通过备份数据创建一个新实例，新实例经过验证后，用户可再将数据通过 DTS 数据传输服务迁回源实例，或者直接使用克隆的新实例。

克隆方式

按时间点克隆：将实例恢复到任意时间点，时间点的选择范围由您设置的日志保留时长决定。
按备份集克隆：将实例恢复到任一物理备份的备份集，备份集选择范围由您设置的数据备份保留时长决定。

新实例费用

新的克隆实例计费模式可根据克隆时自行选择，与购买实例保持一致。
新的克隆实例会在完全克隆成功后，才开始进行计费。

前提条件

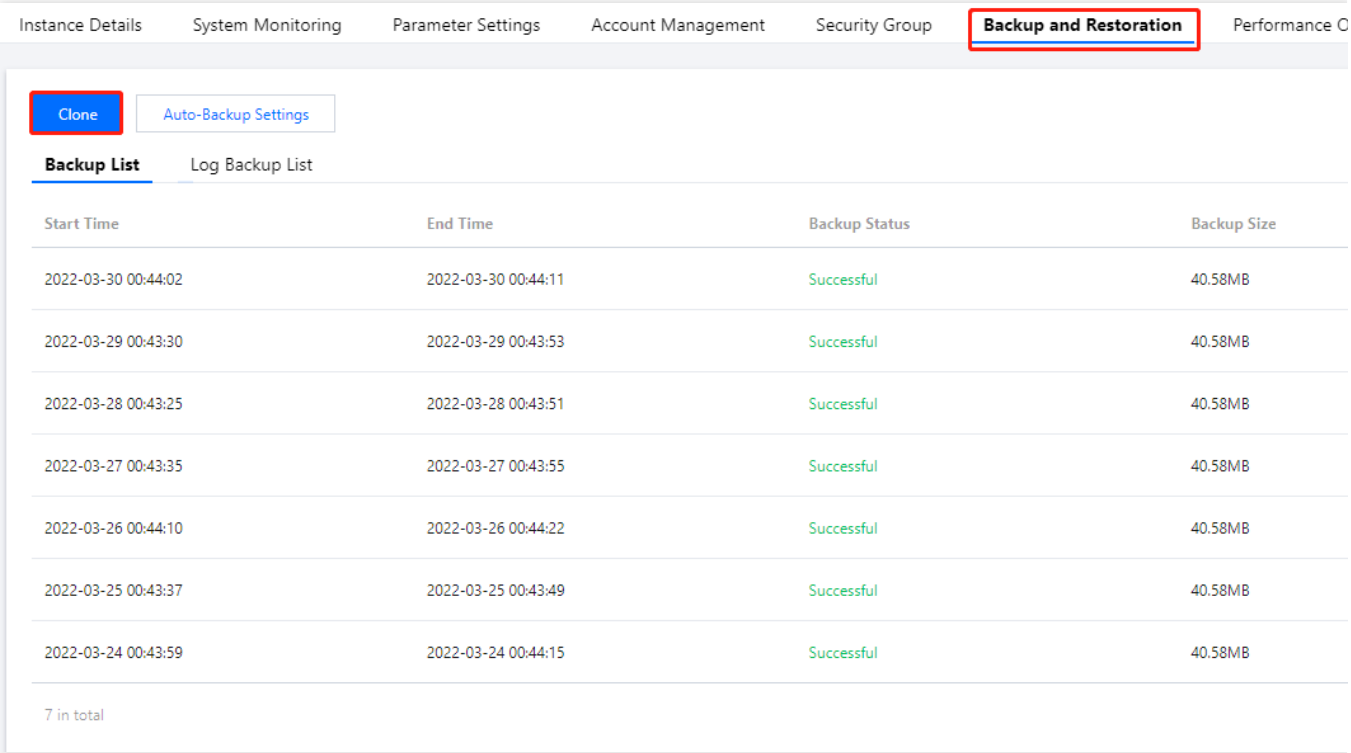
源实例状态须为**运行中**。
若克隆方式选择**按备份集**克隆，源实例必须已经完成物理备份，备份状态可至 控制台 的备份列表查看。
账户内余额须大于0。

注意事项

新克隆实例硬盘空间大小须大于要克隆的数据大小，否则可能导致克隆失败。
新克隆实例的数据库版本须与源实例相同。
对于已使用容量超过6TB的实例，请 提交工单 进行数据恢复。

操作步骤

1. 登录 PostgreSQL 控制台，在实例列表，单击实例 ID 或**操作**列的**管理**，进入实例管理页面。
2. 在实例管理页面，选择**备份恢复** > **备份列表**页， 在左上角单击**克隆**，或在需要克隆的备份**操作**列单击**克隆**。



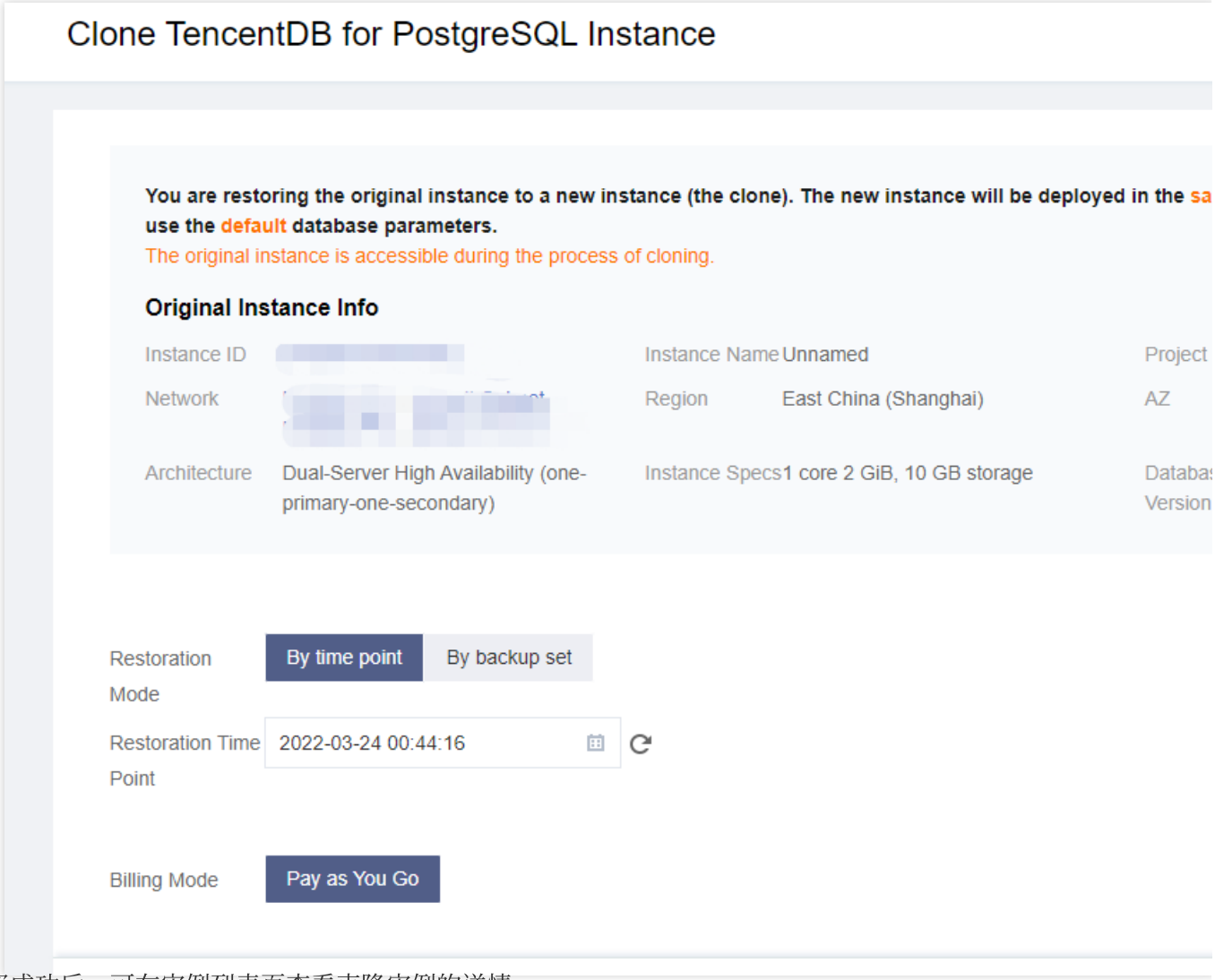
3. 进入购买页，选择克隆实例的克隆方式和相应配置，确认无误后，单击**立即购买**。

按时间点恢复：可从源数据库实例上恢复出具体到某个时间点的数据库实例，可选择的时间范围为最近7天。

按备份集恢复：可从指定的备份集中恢复出新实例，备份集选择范围由您设置的数据备份保留时长所决定。

说明：

备份保留时长可至 控制台 的备份列表查看。



4. 购买成功后，可在实例列表页查看克隆实例的详情。

热点问题

克隆过程中是否会影响源实例的访问？

克隆过程使用原有备份集以及上传至对象存储 COS 中的日志文件进行恢复，不会影响源实例的访问。

自动备份设置

最近更新时间：2024-04-09 11:05:11

云数据库 PostgreSQL 会按照默认的备份设置自动备份数据，您也可以修改自动备份设置，本文为您介绍通过 PostgreSQL 控制台进行自动备份设置。

注意事项

尽量选择业务低峰期进行备份。
若数据量较大，花费的时间可能较长，请耐心等待。
备份文件有保留时间，请及时下载需要保留的备份文件到本地，或者 使用云函数转存 PostgreSQL 历史备份。
备份形式为物理备份，暂时不支持逻辑备份方式。

操作步骤

1. 登录 PostgreSQL 控制台，选择地域，单击实例 ID，进入实例管理页面。
2. 在实例管理页面，选择**备份恢复**页，单击**自动备份设置**。

3. 在弹出的备份设置窗口，完成数据备份设置，单击**确认**。

备份开始时间：支持选择默认时间（默认时间将选中全天中资源空闲时进行备份）或自定义开始备份时间。备份在此时间范围内进行发起。如因意外导致此时间窗未能发起备份，本次备份将不再启动，待到下一次备份开始时间窗发起备份。
数据备份保留时间：可输入7 - 1830天，到期后备份集自动删除。在数据恢复时，将无法恢复到保留的时间段外。
备份周期：支持周一到周日连续或间隔选择。

4. 待右上角提示备份设置成功即完成自动备份设置。

备份设置说明

参数	说明
备份开始时间	默认时间为系统自动分配的备份发起时间。 支持自定义选择备份开始时间，建议设置为业务低峰期。备份发起时间只是备份开始启动时间，并不代表备份结束时间。 例如，选择02:00～06:00开启备份，系统会在02:00～06:00时间范围内的某个时间点发起备份，具体取决于后端的备份策略和备份系统状况。
数据备份保留时间	备份文件可以保留7～1830天，默认为7天，到期后备份集自动删除。
备份周期	默认选中星期一～星期日7天，支持自定义选择备份时间。但为了保障您的数据安全，请设置一周至少备份两次。

在云服务器上恢复 PostgreSQL 数据

最近更新时间：2024-01-22 16:22:57

当用户出现数据丢失或者损坏时，可通过 克隆实例 功能对数据进行按时间点恢复，可恢复的时间根据日志保留时间来决定。

通过控制台下载备份进行恢复

1. 安装 PostgreSQL 数据库

在待恢复数据的云服务器 CVM 中，安装与备份数据相同版本的 PostgreSQL 数据库，如已安装可跳过此步骤。

说明：

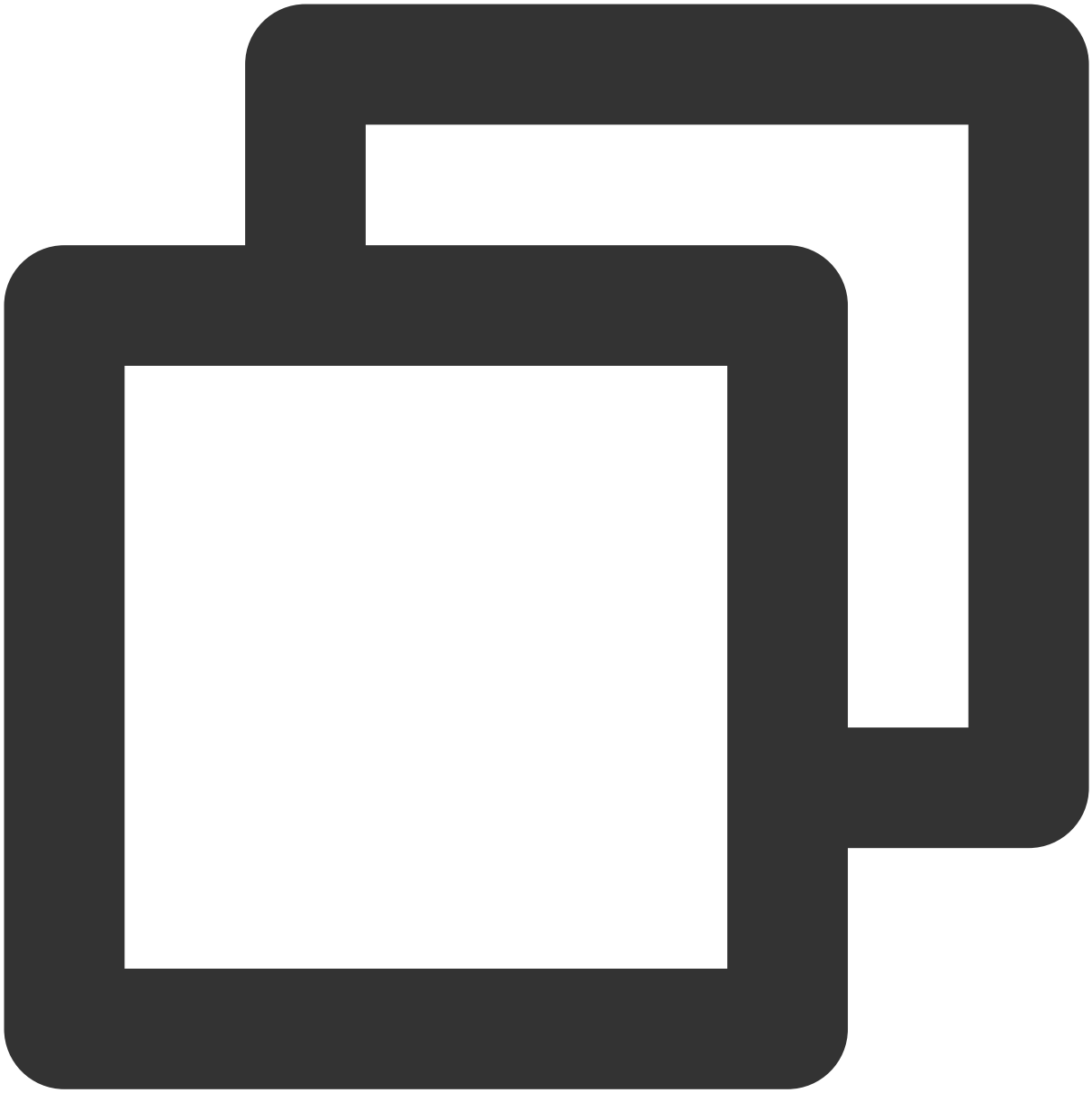
本文以在 CentOS 7 操作系统的云服务器中，安装 PostgreSQL 10 版本数据库并恢复数据为例。

1. 登录 Linux 系统的云服务器，请参见 快速配置 Linux 云服务器。
2. 安装 PostgreSQL 数据库，本文采用 yum 源的安装方式，yum 源可至 该地址 查找所需版本。

说明：

如果您需要恢复 11.8 或者 12.4 版本，建议安装同版本的数据库，请更改安装包中的版本号，如 `postgresql10-server` 修改为 `postgresql11-server` 或者 `postgresql12-server` 。

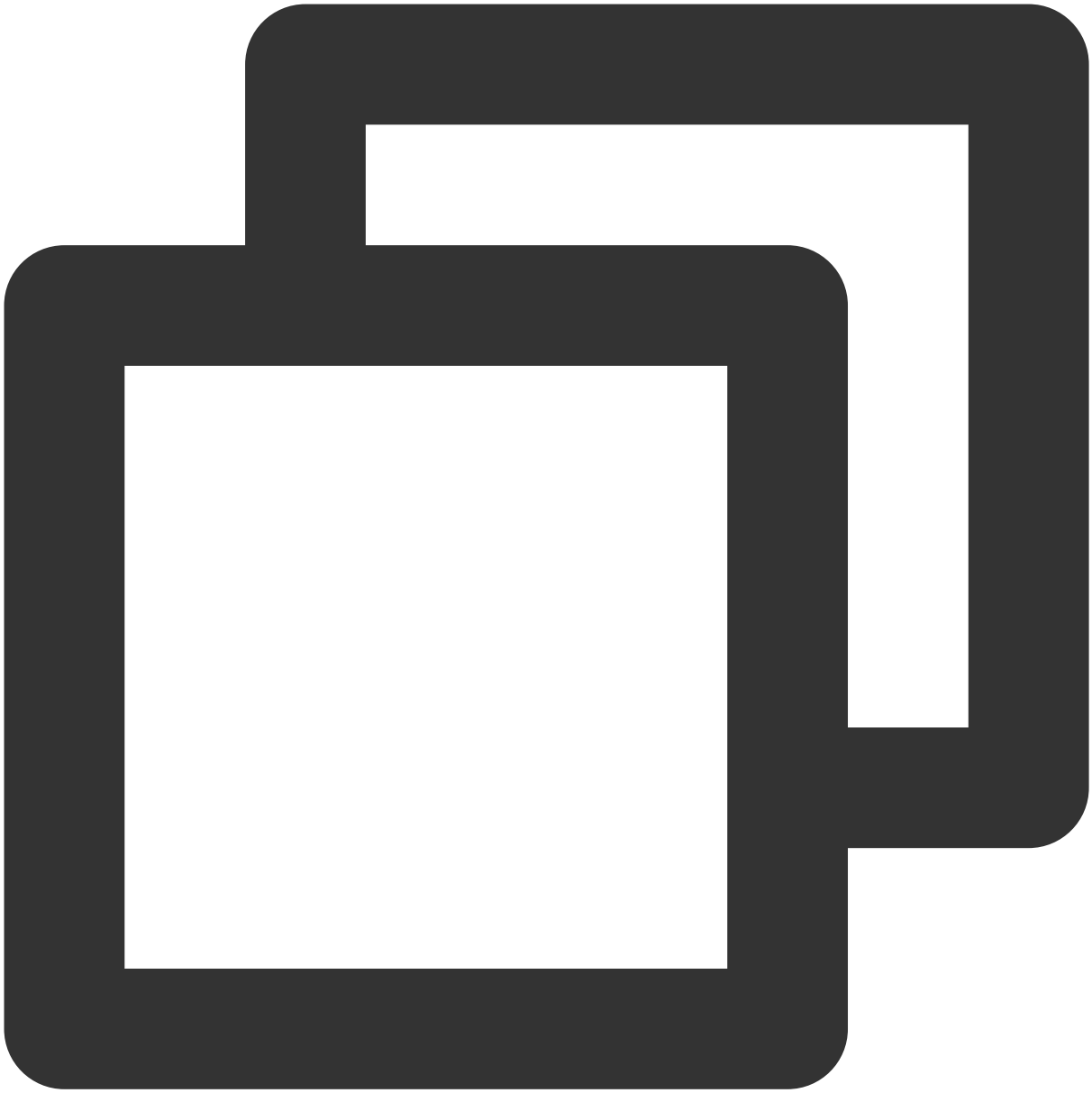
执行如下命令，安装 PostgreSQL 10 版本。



```
yum install https://download.postgresql.org/pub/repos/yum/reporpms/EL-7-x86_64/pgdg
yum install postgresql10-server postgresql10-contrib postgresql10 postgresql10.x86_
```

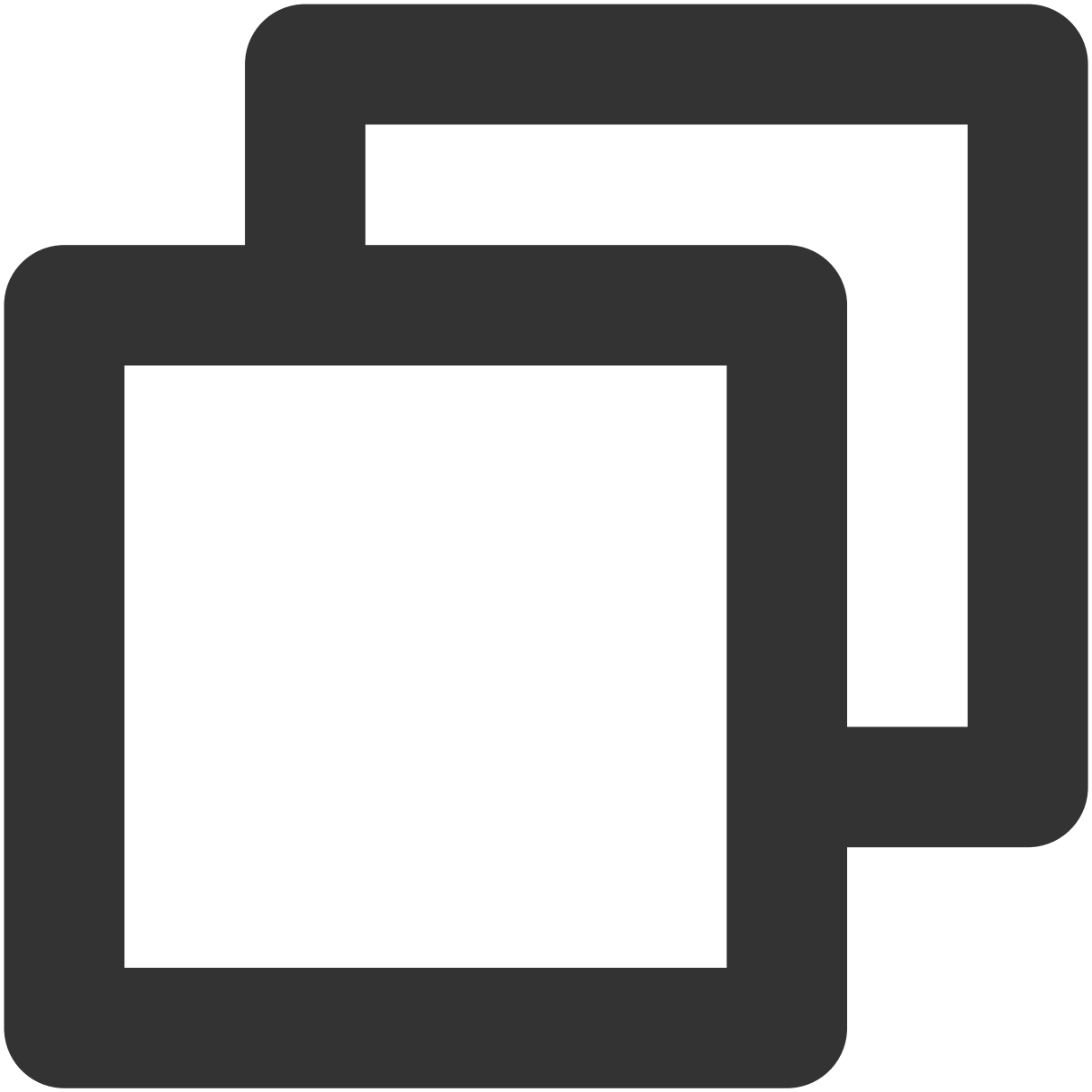
说明：

PostgreSQL 9.5 版本安装命令如下：



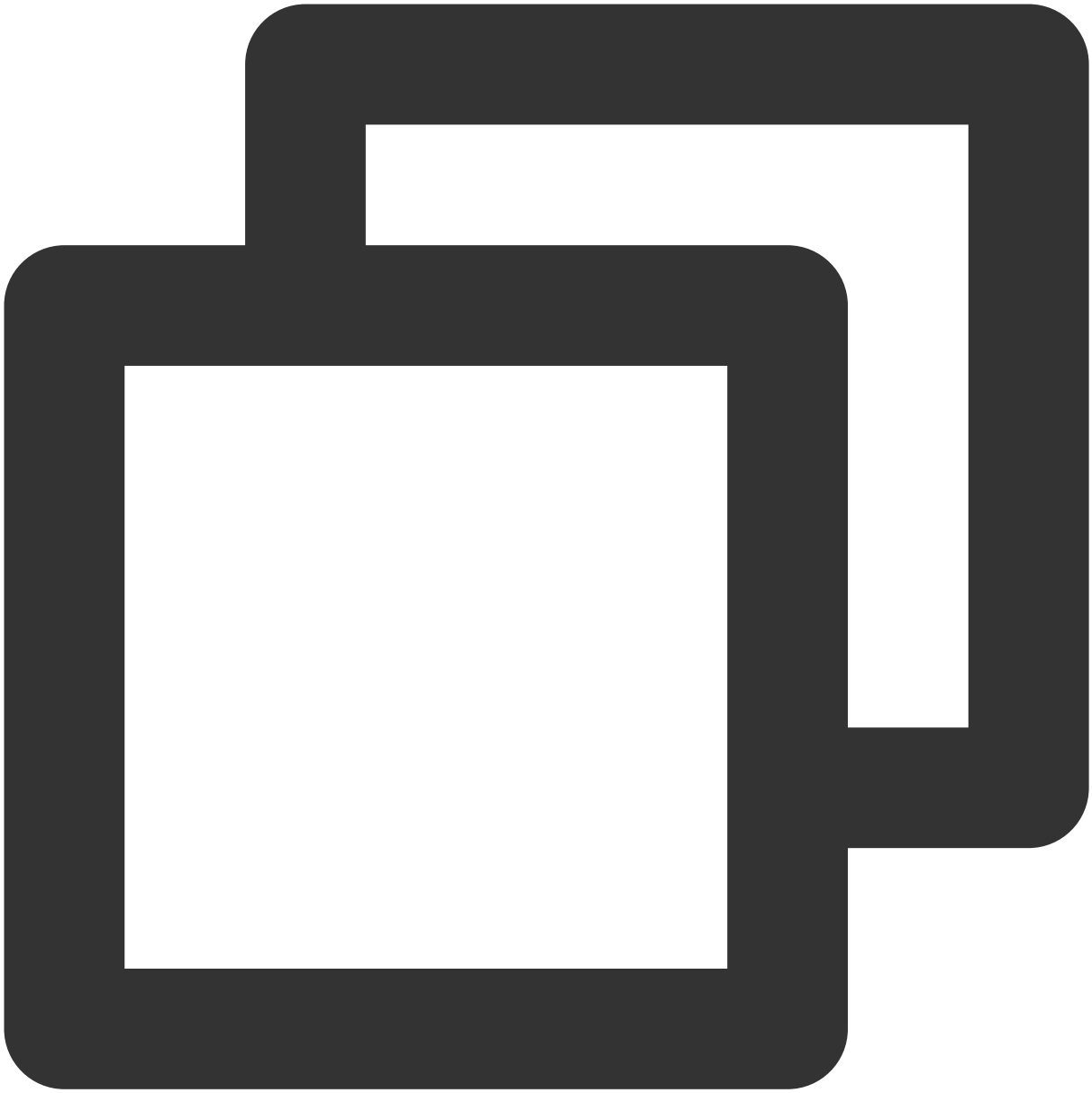
```
yum install https://yum.postgresql.org/9.5/redhat/rhel-7.6-x86_64/pgdg-centos95-9.5
yum install postgresql95-server postgresql95-contrib postgresql95
```

3. 执行如下命令，查看安装结果。



```
rpm -aq| grep postgres
```

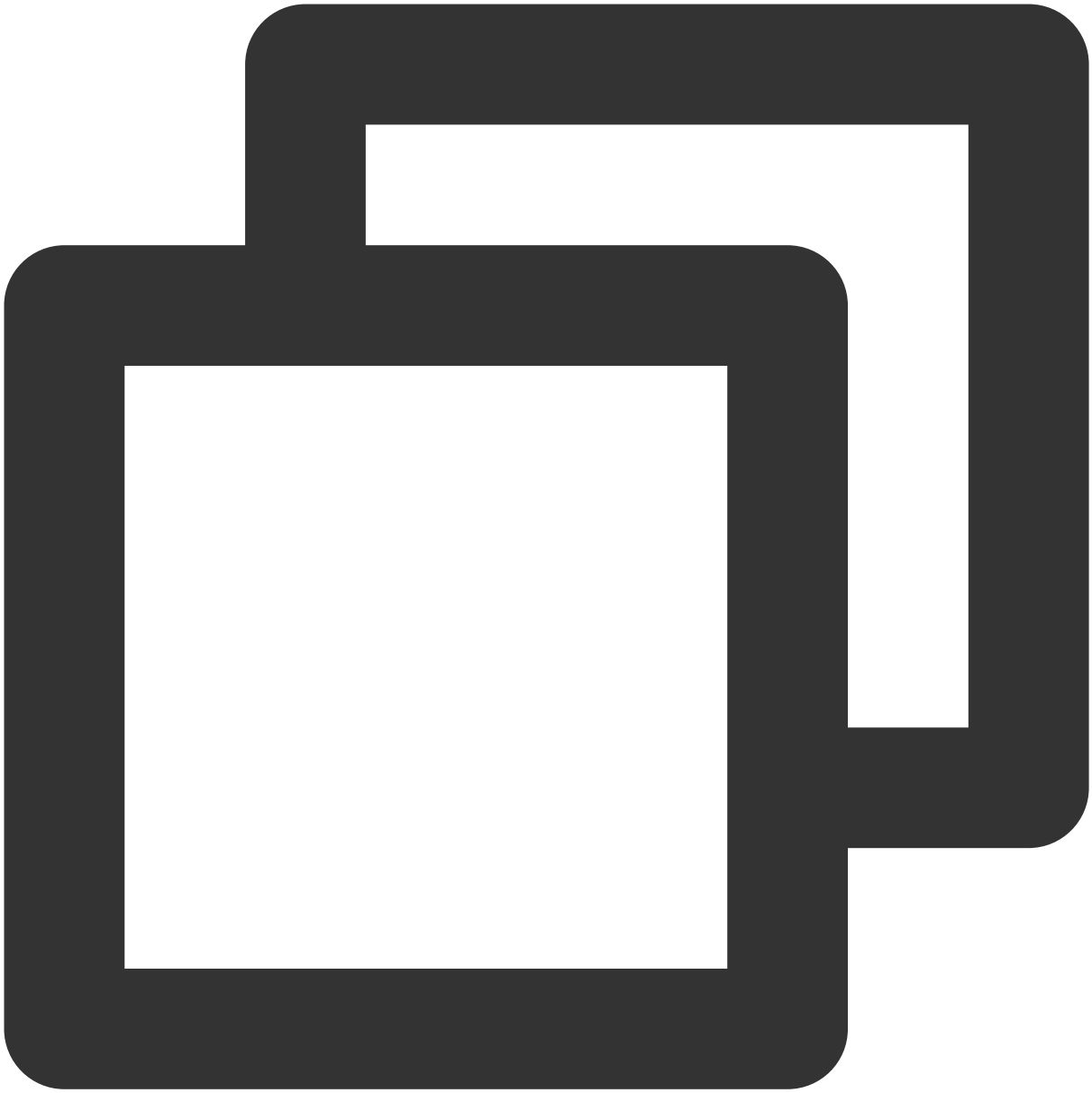
可返回如下类似信息：



```
[root@i-87-575-VM vmuser]# rpm -aq| grep postgres
postgresql10-libs-10.11-2PGDG.rhel7.x86_64
postgresql10-server-10.11-2PGDG.rhel7.x86_64
postgresql10-contrib-10.11-2PGDG.rhel7.x86_64
postgresql10-10.11-2PGDG.rhel7.x86_64
```

2. 使用 postgres 用户创建恢复目录

切换为 postgres 用户，在云服务器中创建恢复目录。

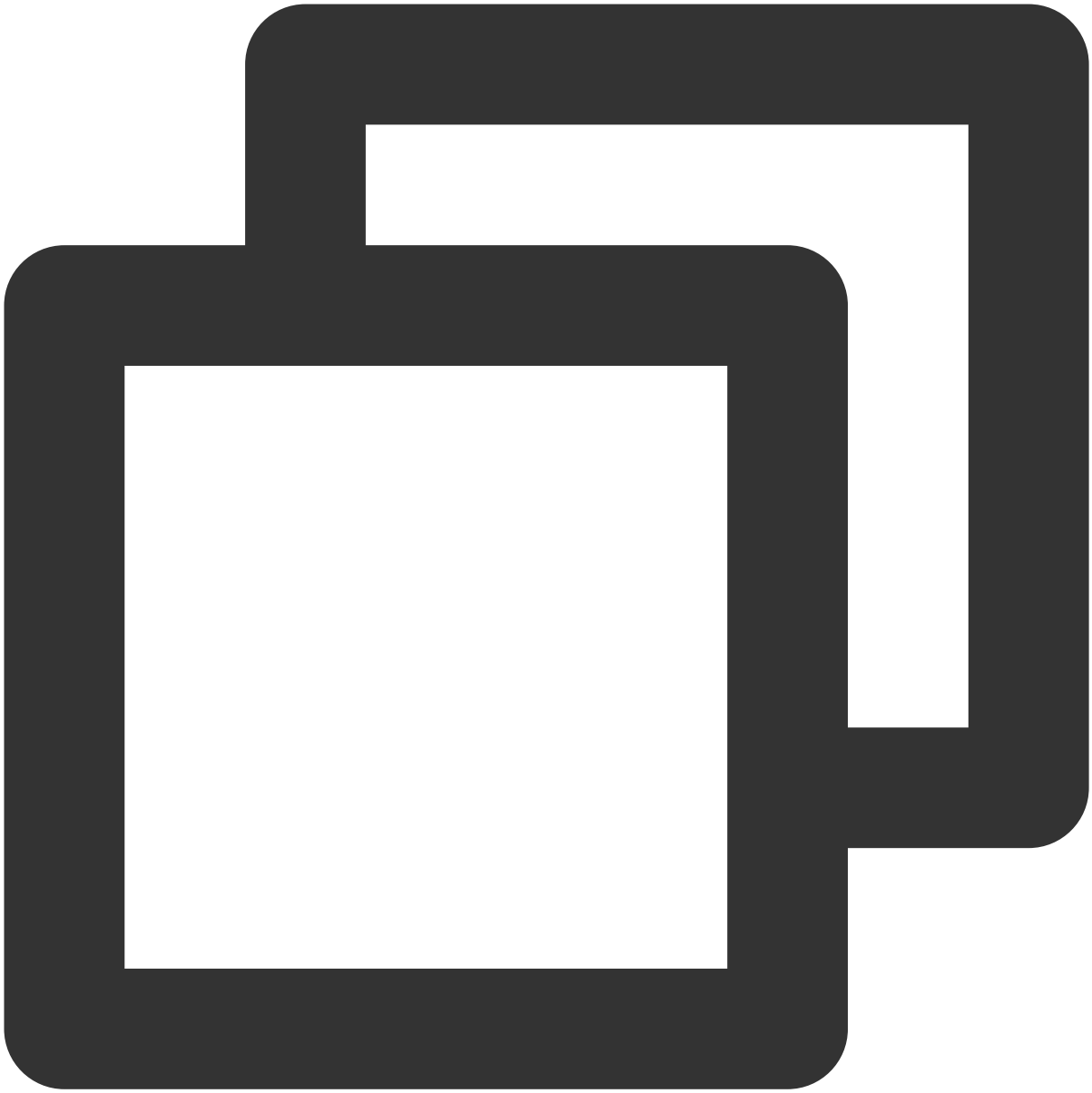


```
mkdir /var/lib/pgsql/10/recovery
```

其中，recovery 为示例目录，用户可自行修改恢复目录。后续示例中不同版本目录名将不再区分描述，请以实际为准，如PostgreSQL 10.x 均为 `/var/lib/pgsql/10` ，PostgreSQL 9.5.x 均为 `/var/lib/pgsql/9.5` 。

说明：

PostgreSQL 9.5 版本命令如下：



```
mkdir /var/lib/pgsql/9.5/recovery
```

3. 下载全量备份文件

1. 登录 PostgreSQL 控制台，在实例列表，单击**操作**列的**管理**进入管理页面。
2. 选择**备份管理**页，在备份列表中，根据备份时间选择需要恢复的备份版本，单击**操作**列的**下载**。
3. 根据提供的 VPC 网络地址或外网地址链接下载备份文件。

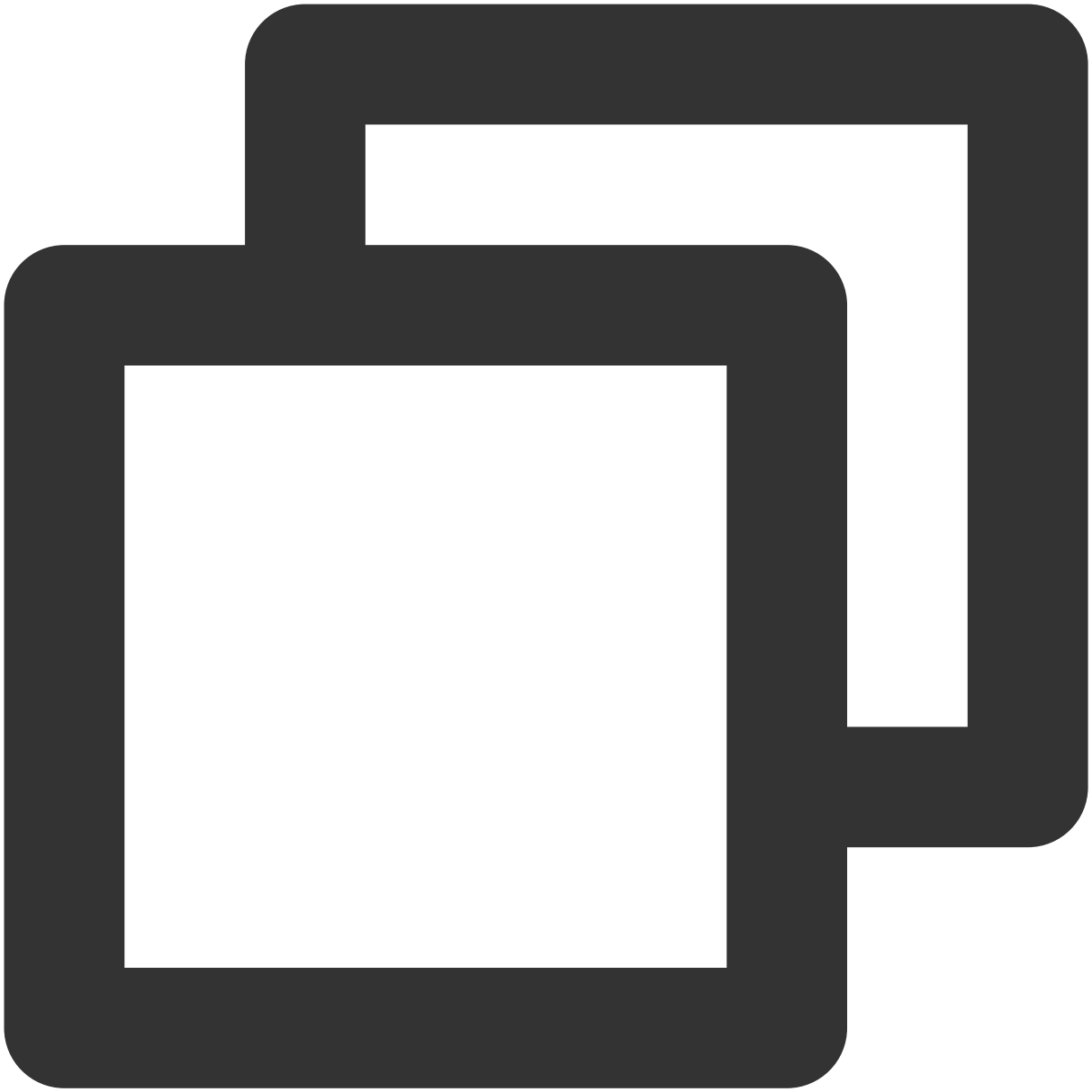
说明：

使用 VPC 网络地址下载备份时，云数据库须与云服务器处于同一 VPC，备份需下载至 `/var/lib/pgsql/10/recovery` 目录。
使用外网地址下载备份时，下载后需将备份文件上传至云服务器中的 `/var/lib/pgsql/10/recovery` 目录，请参见 如何将本地文件拷贝到云服务器。
上传完示例如下：

```
[root@VM_0_12_centos recovery]# ls -lh
total 6.3M
-rw-r--r-- 1 root root 6.3M Dec 23 11:45 20191221010146.tar.gz
```

4. 解压全量备份文件

执行如下命令，解压全量备份文件。



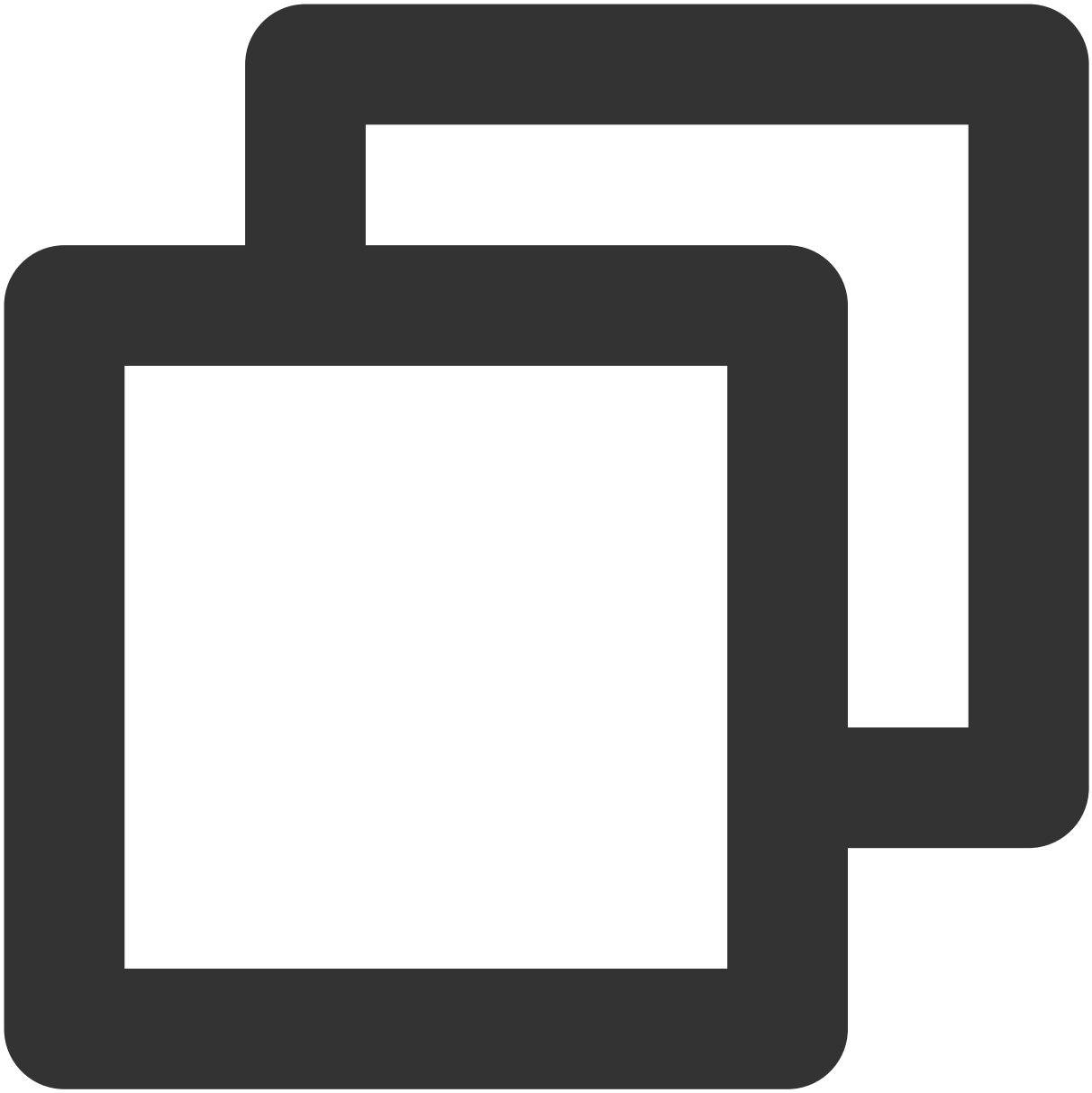
```
cd /var/lib/pgsql/10/recovery
tar -xf 20191221010146.tar.gz
```

解压后如下图：

```
[root@VM_0_12_centos recovery]# ls -lh
total 6.4M
-rw-r--r-- 1 root root  6.3M Dec 23 11:45 20191221010146.tar.gz
-rw------- 1 1003 users  215 Dec 21 01:01 backup_label
drwx------ 7 1003 users 4.0K Dec 13 17:37 base
-rw------- 1 1003 users   35 Dec 21 00:00 current_logfiles
drwx------ 2 1003 users 4.0K Dec  5 20:12 global
drwx------ 2 1003 users 4.0K Dec  2 21:59 pg_commit_ts
drwx------ 2 1003 users 4.0K Dec  2 21:59 pg_dynshmem
-rw------- 1 1003 users 4.8K Dec  2 21:59 pg_hba.conf
-rw------- 1 1003 users 1.6K Dec  2 21:59 pg_ident.conf
drwx------ 4 1003 users 4.0K Dec 21 01:01 pg_logical
drwx------ 4 1003 users 4.0K Dec  2 21:59 pg_multixact
drwx------ 2 1003 users 4.0K Dec  2 21:59 pg_notify
drwx------ 2 1003 users 4.0K Dec  2 21:59 pg_replslot
drwx------ 2 1003 users 4.0K Dec  2 21:59 pg_serial
drwx------ 2 1003 users 4.0K Dec  2 21:59 pg_snapshots
drwx------ 2 1003 users 4.0K Dec  2 21:59 pg_stat
drwx------ 2 1003 users 4.0K Dec 21 01:01 pg_stat_tmp
drwx------ 2 1003 users 4.0K Dec  2 21:59 pg_subtrans
drwx------ 2 1003 users 4.0K Dec  2 21:59 pg_tblspc
drwx------ 2 1003 users 4.0K Dec  2 21:59 pg_twophase
-rw------- 1 1003 users    3 Dec  2 21:59 PG_VERSION
drwx------ 2 1003 users 4.0K Dec  2 21:59 pg_xact
drwxr-xr-x 2 1003 users 4.0K Dec  3 01:01 pg_xlog
-rw------- 1 1003 users   11 Dec 20 12:02 pg_xlog_archive.tmp
-rw------- 1 1003 users   88 Dec  2 21:59 postgresql.auto.conf
-rw------- 1 1003 users  24K Dec  2 21:59 postgresql.conf
```

5. 删除多余的临时文件

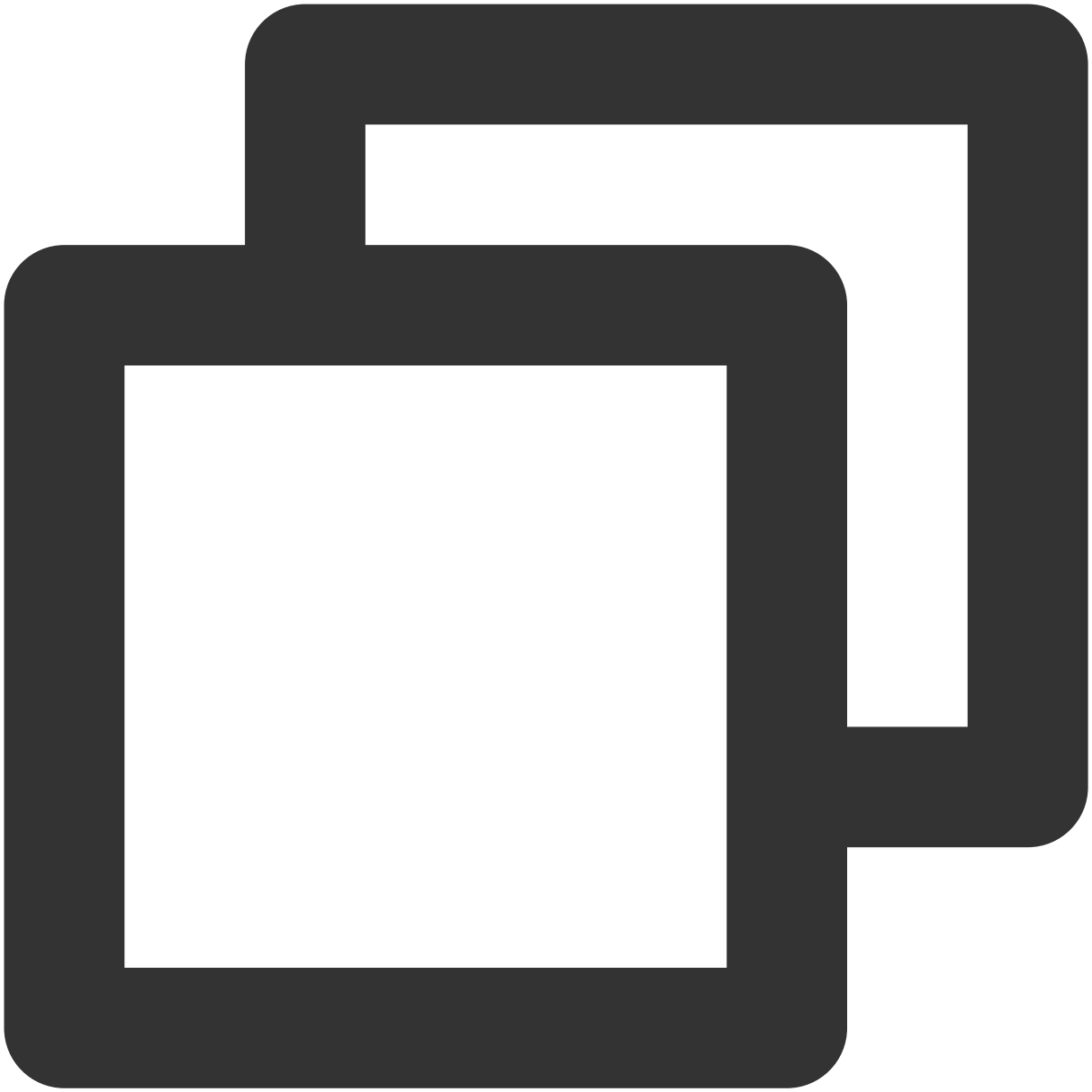
执行如下命令，删除多余的临时文件。



```
rm -rf backup_label
```

6. 修改配置文件

6.1 将配置文件 `postgresql.conf` 中的以下选项注释掉，注释方法：在行首使用#。
如有多个该选项，则全部注释掉。

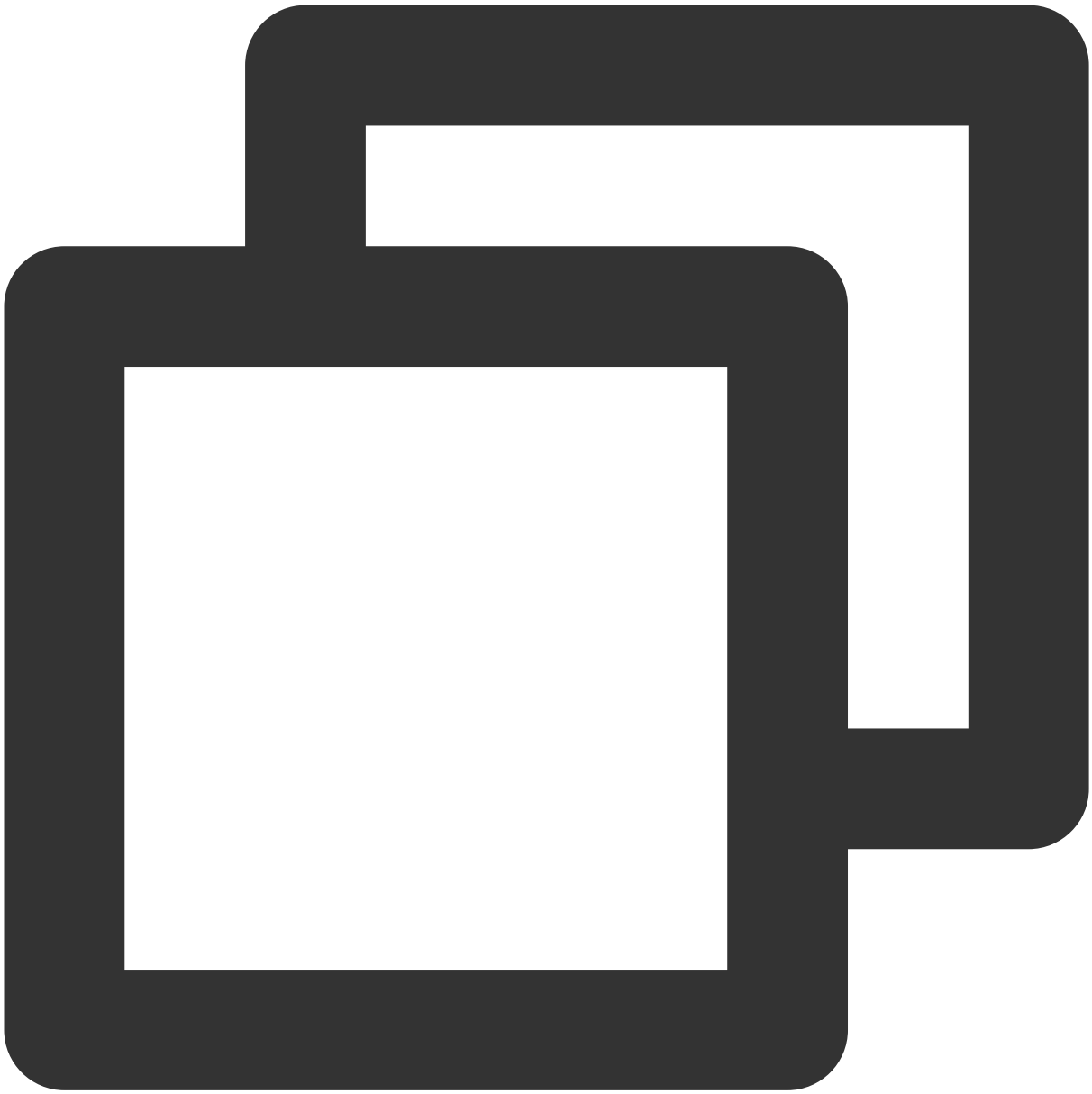


```
shared_preload_libraries
local_preload_libraries
pg_stat_statements.max
pg_stat_statements.track
archive_mode
archive_command
synchronous_commit
synchronous_standby_names
```

说明：

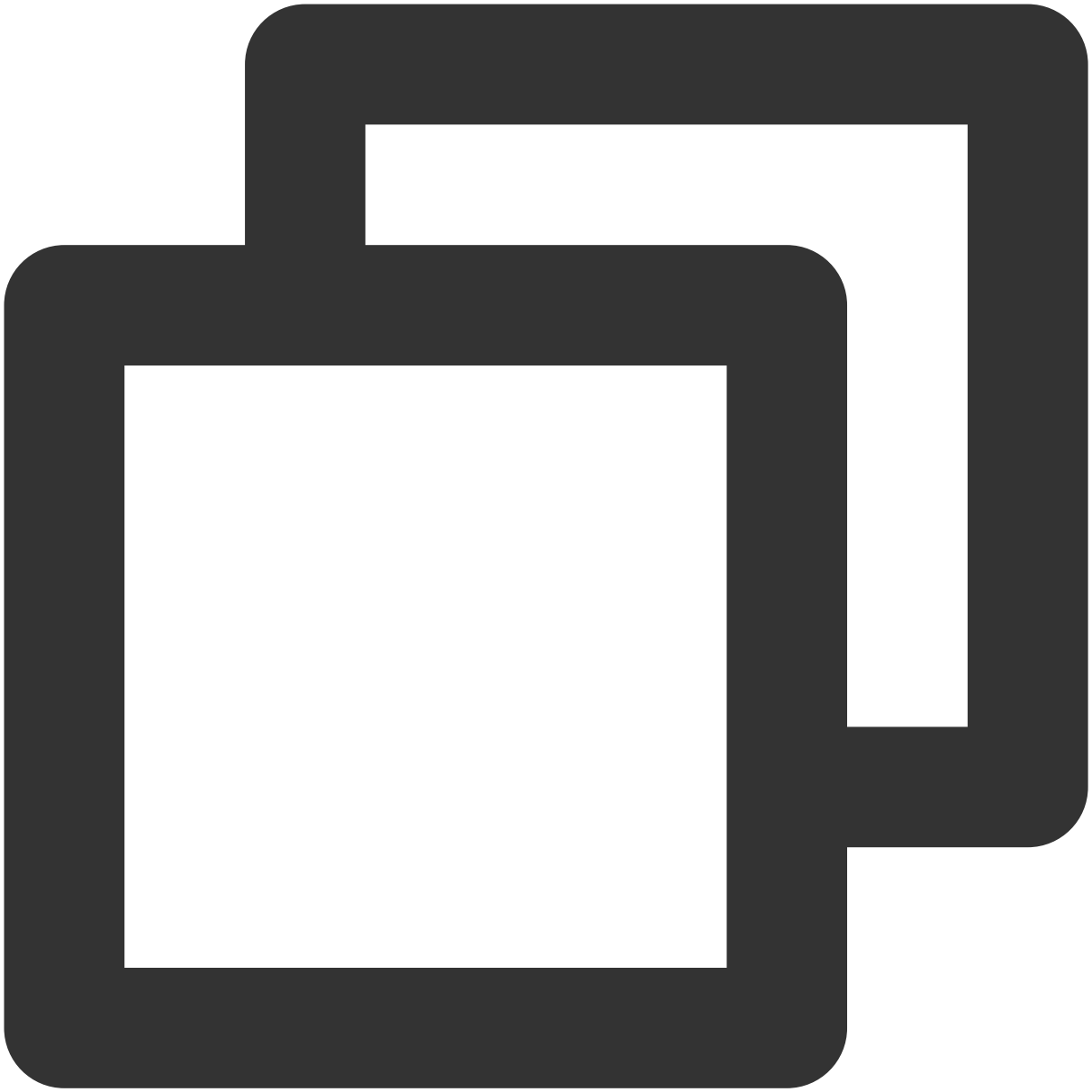
如果恢复版本为 PostgreSQL 12.4，还需要注释 `include = 'standby.conf'` 这一行。

2. 修改配置文件 `postgresql.conf` 。



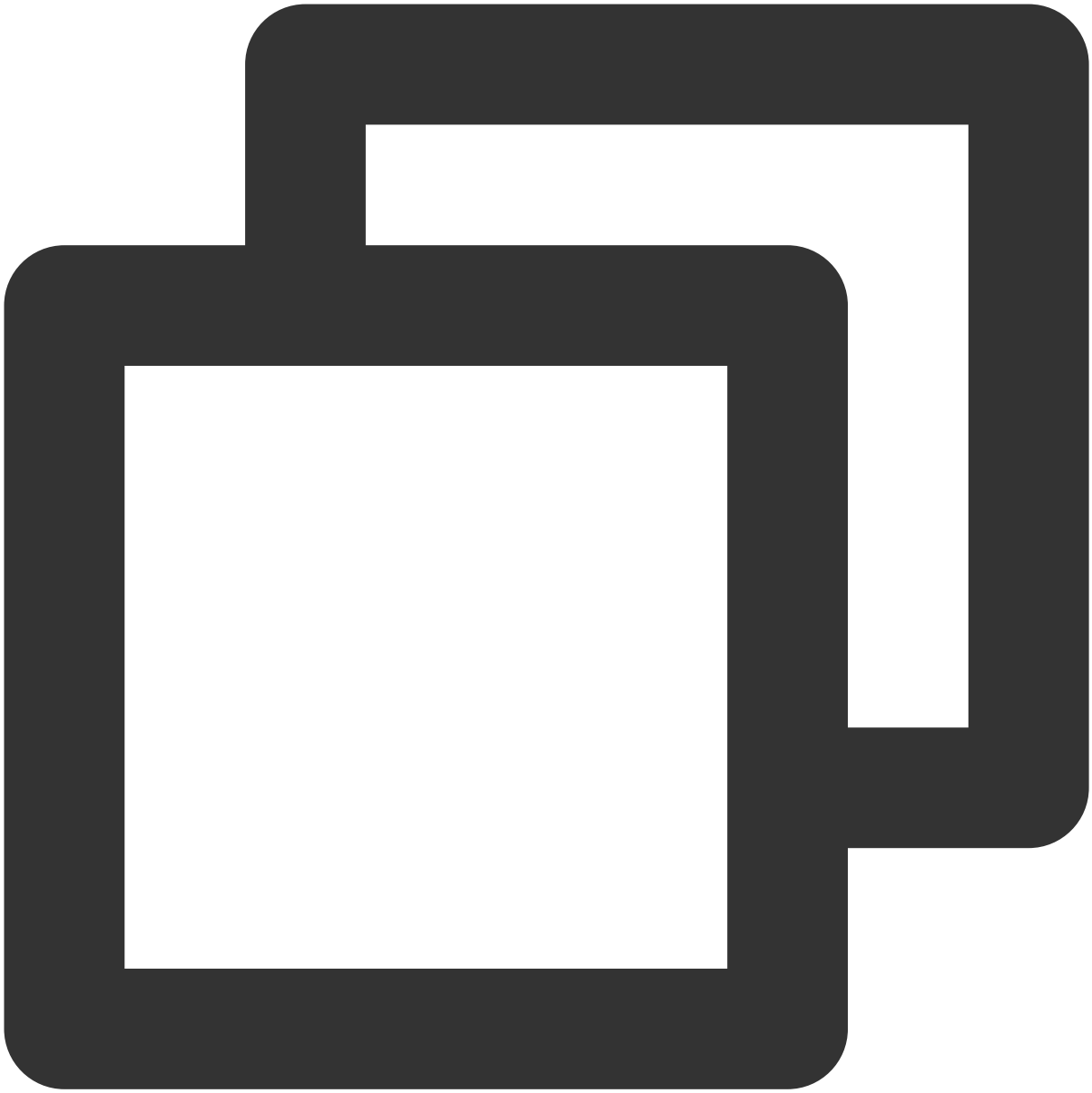
```
port = '5432'     ##将port参数的值修改为5432
unix_socket_directories = '/var/run/postgresql/'   ##将unix_socket_directories的值修改
```

6.2 在 `postgresql.conf` 文件末尾追加配置，表示不再使用强同步模式。



```
synchronous_commit = local
synchronous_standby_names = ''
```

7. 使用 root 用户更改文件夹权限



```
chmod 0700 /var/lib/pgsql/10/recovery
chown postgres:postgres /var/lib/pgsql/10/recovery -R
```

修改后如下图：

```
[root@VM_0_12_centos recovery]# ls -al
total 6528
drwx------ 19 postgres postgres    4096 Dec 23 11:50 .
drwx------  6 postgres postgres    4096 Dec 23 11:44 ..
-rw-r--r--  1 postgres postgres 6546935 Dec 23 11:45 20191221010146.tar.
-rw-------  1 postgres postgres     215 Dec 21 01:01 backup_label
drwx------  7 postgres postgres    4096 Dec 13 17:37 base
-rw-------  1 postgres postgres      35 Dec 21 00:00 current_logfiles
drwx------  2 postgres postgres    4096 Dec  5 20:12 global
drwx------  2 postgres postgres    4096 Dec  2 21:59 pg_commit_ts
drwx------  2 postgres postgres    4096 Dec  2 21:59 pg_dynshmem
-rw-------  1 postgres postgres    4858 Dec  2 21:59 pg_hba.conf
```

8.（可选）应用增量备份文件

如跳过该步骤，则数据库的内容为开始做全量备份时数据库的内容。

将 xlog 文件放入 `/var/lib/pgsql/10/recovery/pg_wal` 文件夹下，如下载的备份中不包含 `pg_wal` 目录，请将 `pg_xlog` 目录修改为 `pg_wal` ，pg 会自动重放 xlog 日志。

例如12:00时做的全量备份，如果在该全量备份的基础上，在 `pg_wal` 文件夹下放置12:00至13:00的所有 xlog，则数据库能恢复到13:00时的数据内容。

说明：

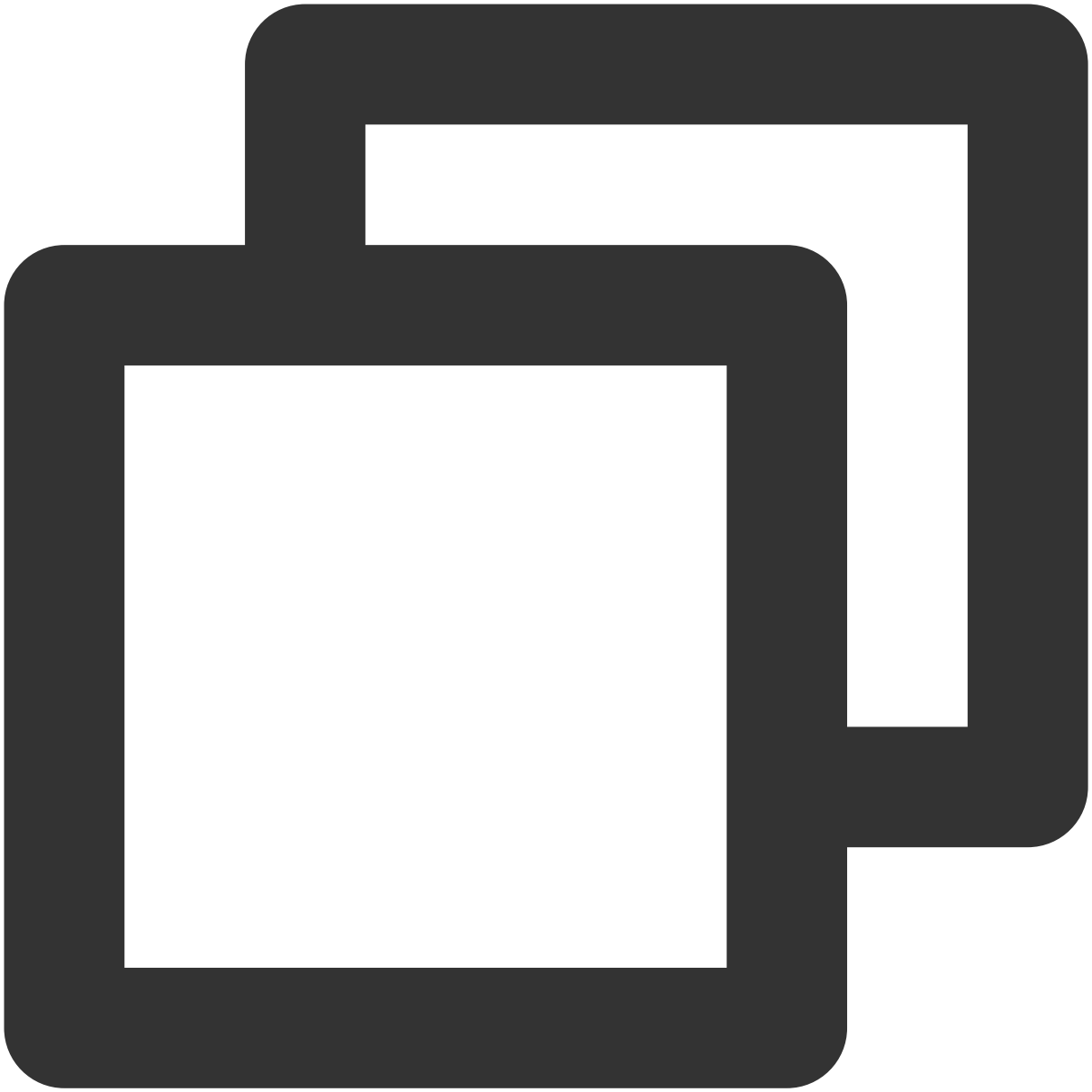
PostgreSQL 版本为 9.x 时，则为 `/var/lib/pgsql/9.x/recovery/pg_xlog` 文件夹。

1. 在控制台【备份管理】页，获取 xlog 下载地址，下载增量备份文件（xlog）。

下载后如下图：

```
[root@VM_0_12_centos recovery]# mv pg_xlog pg_wal
[root@VM_0_12_centos recovery]# cd pg_wal/
[root@VM_0_12_centos pg_wal]# ls -lh
total 64K
-rw-r--r-- 1 root root 31K Dec 23 11:44 20191221010157_20191221010157.ta
-rw-r--r-- 1 root root 31K Dec 23 11:44 20191221013157_20191221013157.ta
```

2. 解压日志至 `pg_wal` 文件夹。



```
tar -xf 20170904010214_20170905010205.tar.gz
```

```
[root@VM_0_12_centos pg_wal]# ll
total 32836
-rw------- 1 postgres postgres 16777216 Dec 21 01:01 000000010000000000
-rw------- 1 postgres postgres      312 Dec 21 01:01 000000010000000000
-rw------- 1 postgres postgres 16777216 Dec 21 01:31 000000010000000000
-rw-r--r-- 1 postgres postgres    31319 Dec 23 11:44 20191221010157_201
-rw-r--r-- 1 postgres postgres    30966 Dec 23 11:44 20191221013157_201
```

9. 使用 postgres 用户启动数据库

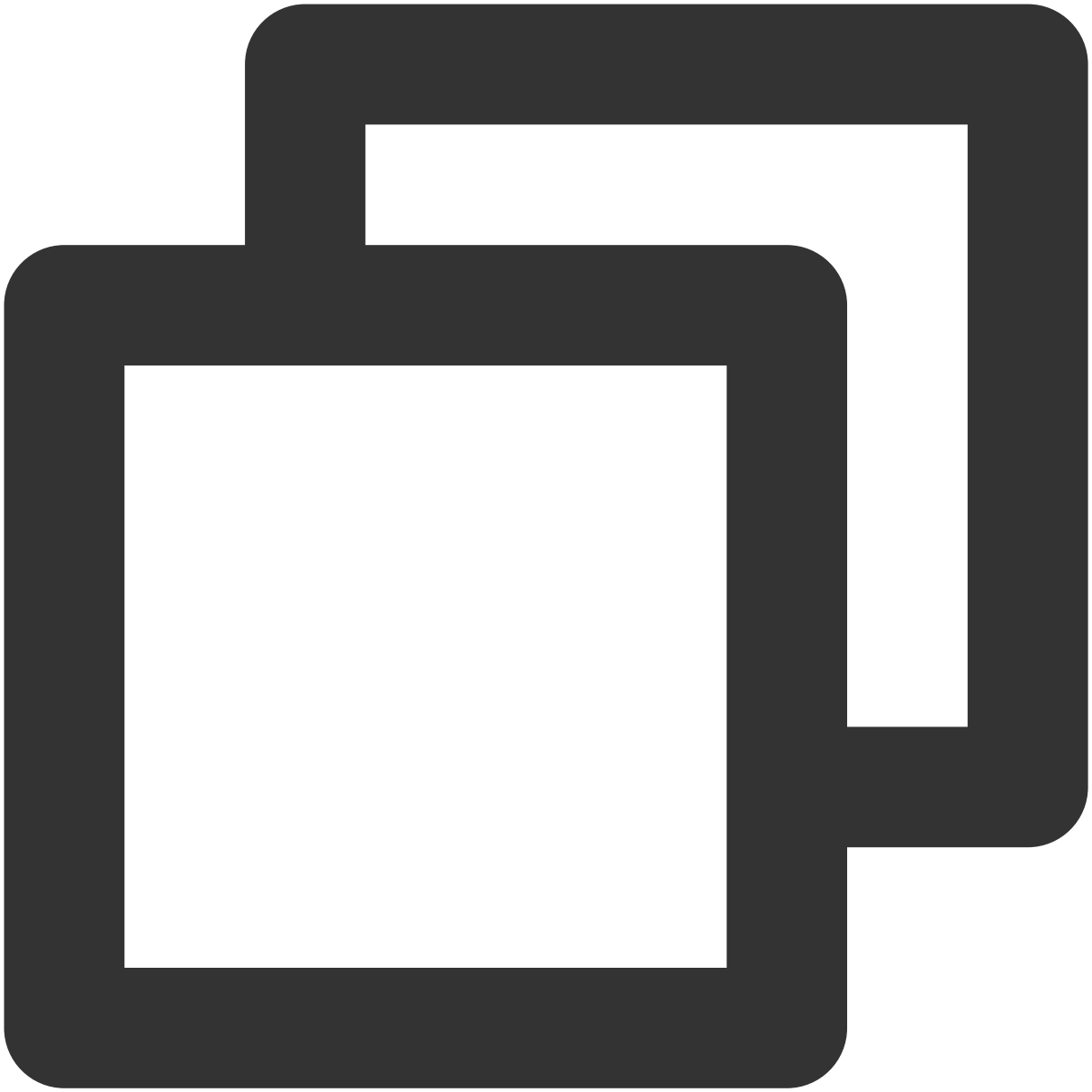
```
/usr/pgsql-10/bin/pg_ctl start -D /var/lib/pgsql/10/recovery
```

```
-bash-4.2$ /usr/pgsql-10/bin/pg_ctl start -D /var/lib/pgsql/10/recovery
waiting for server to start....2019-12-23 11:59:42.654 CST [14061] LOG:
.0.0", port 5432
2019-12-23 11:59:42.654 CST [14061] LOG:  listening on IPv6 address "::
2019-12-23 11:59:42.664 CST [14061] LOG:  listening on Unix socket "/va
2019-12-23 11:59:42.686 CST [14061] LOG:  listening on Unix socket "/tm

2019-12-23 11:59:42.840 CST [14061] LOG:  redirecting log output to log
2019-12-23 11:59:42.840 CST [14061] HINT:  Future log output will appea
 done
server started
```

10. 登录数据库验证

1. 登录 PostgreSQL 数据库。



```
export PGDATA=/var/lib/pgsql/10/recovery
psql
```

```
-bash-4.2$ export PGDATA=/var/lib/pgsql/10/recovery
-bash-4.2$ psql
psql (9.2.24, server 10.11)
WARNING: psql version 9.2, server version 10.0.
         Some psql features might not work.
Type "help" for help.
```

2. 验证数据库是否运行。

```
/usr/pgsql-10/bin/pg_ctl status -D /var/lib/pgsql/10/recovery
```

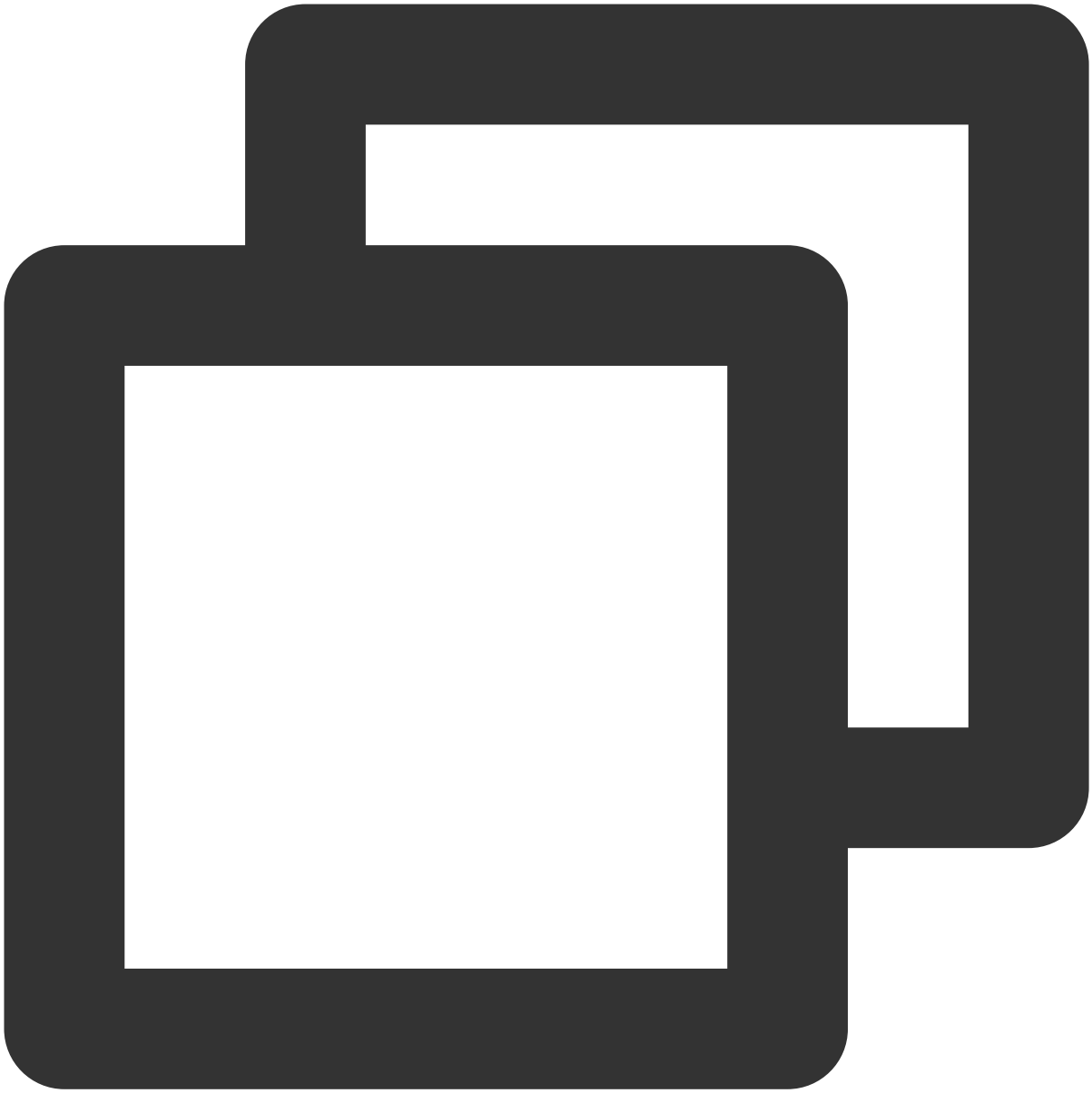
如提示"server is running"，则代表数据库正在运行。

```
-bash-4.2$ /usr/pgsql-10/bin/pg_ctl status -D /var/lib/pgsql/10/recovery
pg_ctl: server is running (PID: 14061)
/usr/pgsql-10/bin/postgres "-D" "/var/lib/pgsql/10/recovery"
```

通过手动导出数据进行恢复

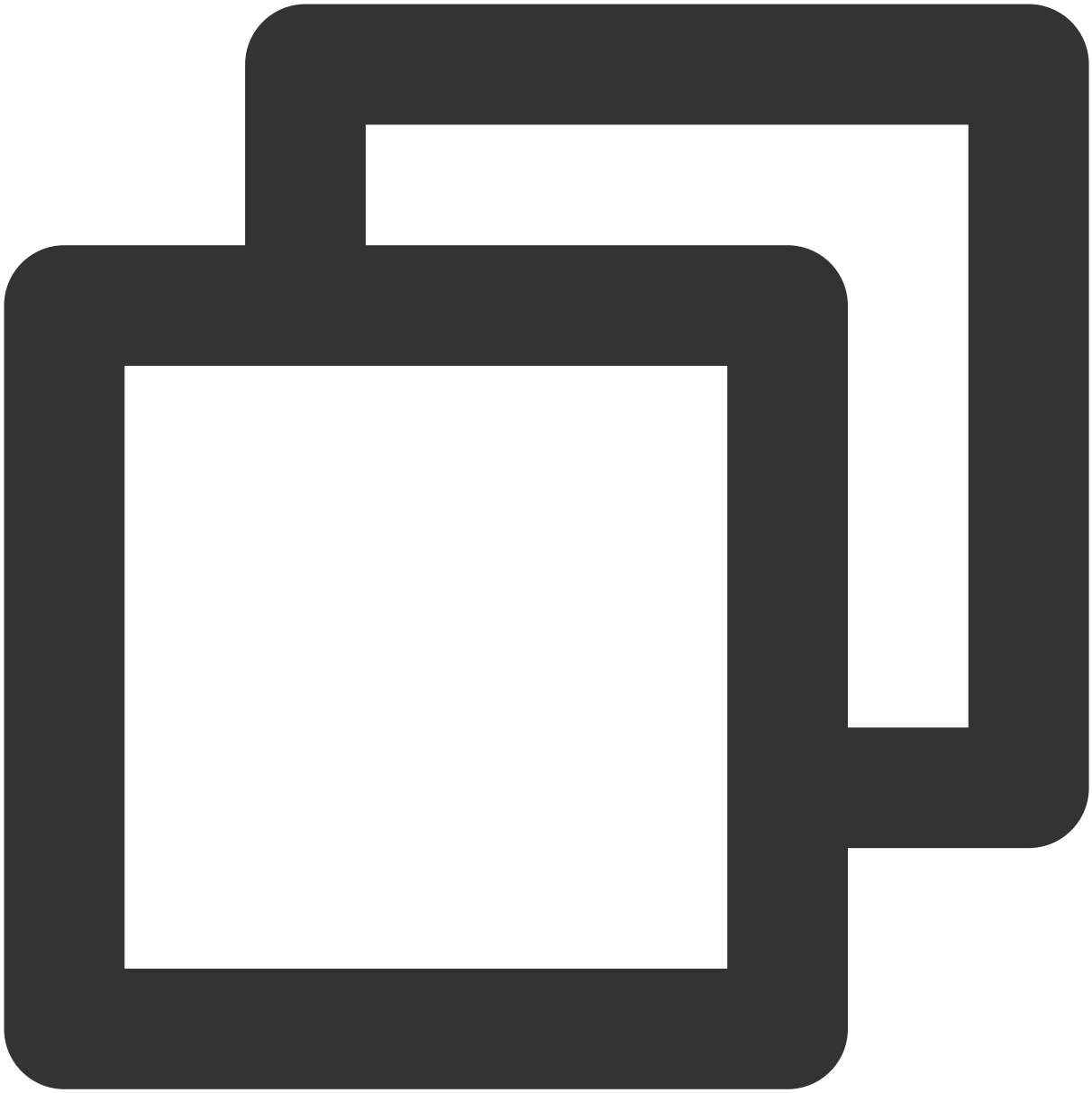
您也可以手动导出备份数据，然后在腾讯云云服务器上进行恢复操作，该方案在 Windows 和 Linux 下同样适用，与物理文件所在的文件系统无关。

1. 在云服务器下 dump 出数据，示例如下：



```
命令格式为：pg_dump -h <访问IP> -U <访问用户> -f <备份文件全路径> -c -C <导出的数据库名>
Example:
/usr/pgsql-10/bin/pg_dump -h 192.168.0.16 -U testroot -f backup.sql -c -C postgres
```

不指定文件格式时，默认导出的文件为文本文件，文件形式如下：



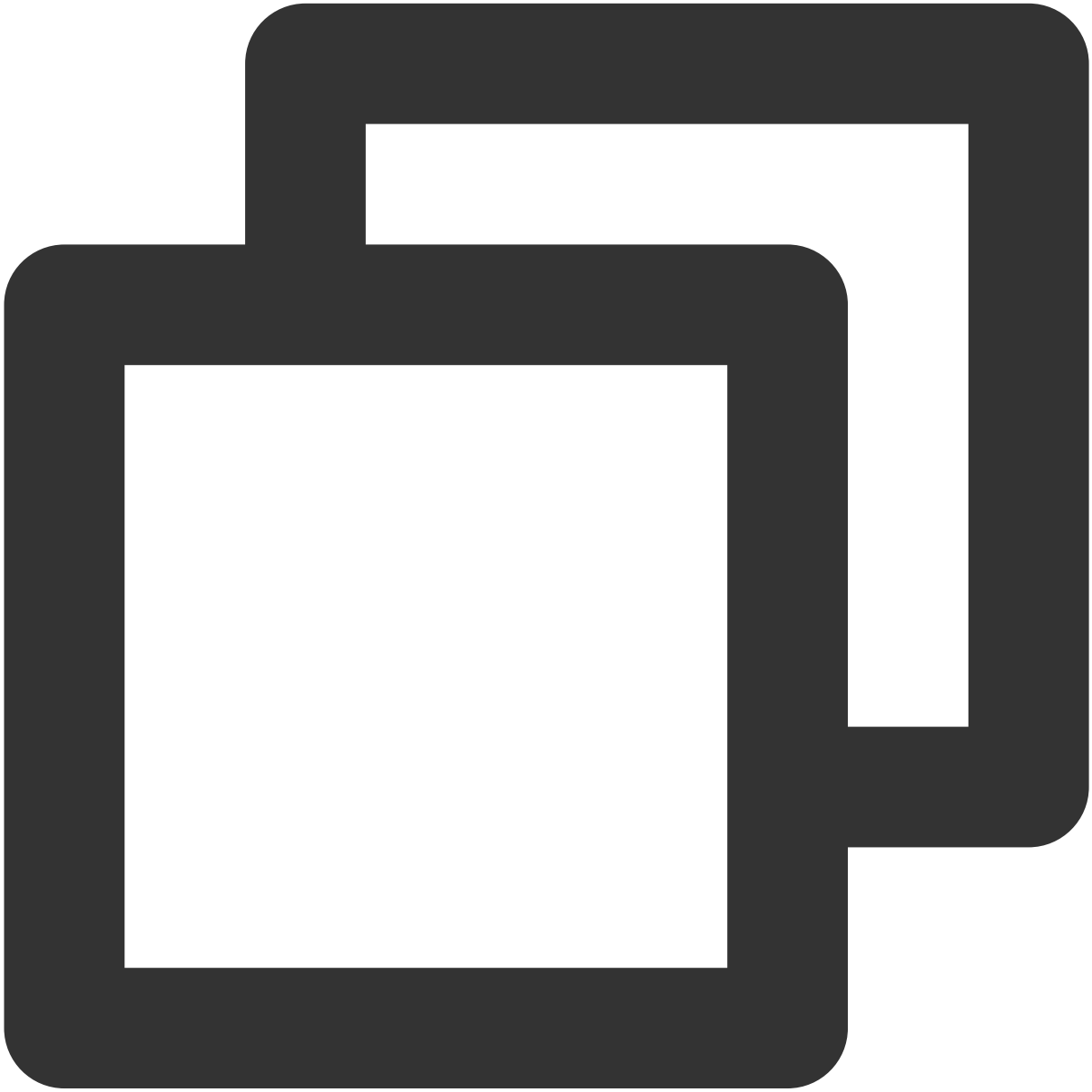
```
-- PostgreSQL database dump
--
-- Dumped from database version 9.5.4
-- Dumped by pg_dump version 9.5.19
SET statement_timeout = 0;
SET lock_timeout = 0;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = on;
SELECT pg_catalog.set_config('search_path', '', false);
SET check_function_bodies = false;
SET xmloption = content;
```

```
SET client_min_messages = warning;
SET row_security = off;
```

若数据较大，可通过 -Fc 指定为二进制文件。

2. 在云服务器上恢复数据。

文本文件，可直接通过执行 sql 语句恢复，示例如下：



```
psql -U postgres <./backup.sql
```

说明：

因为有 pg_stat_error 等插件，可能会导致报错，但不影响数据导入。

二进制文件，需要用 pg_restore 还原。

版权所有：腾讯云计算（北京）有限责任公司

删除备份

最近更新时间：2024-01-22 16:22:57

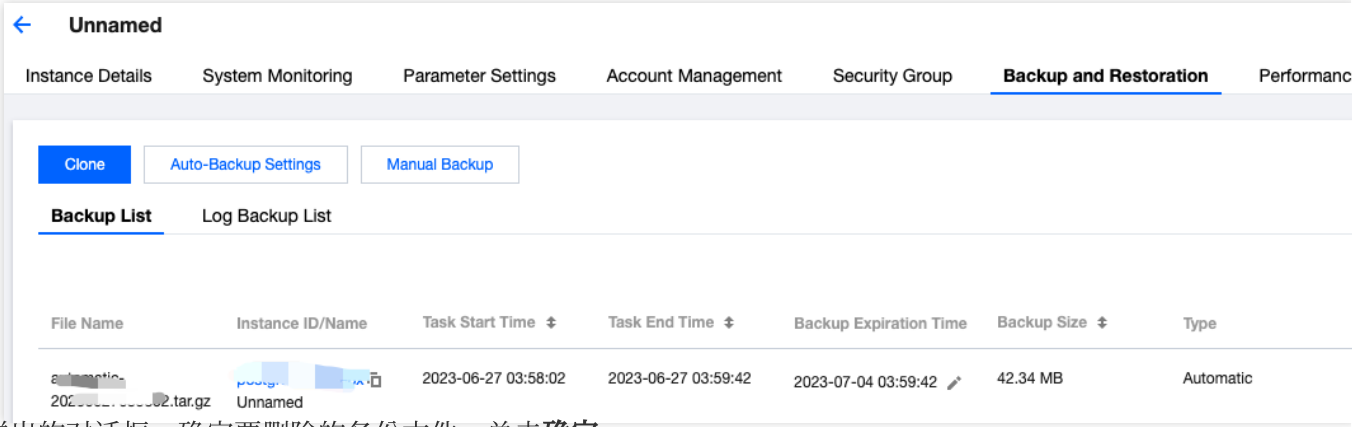
您可以删除数据库备份以节省备份空间开销。

说明：

手动备份、自动备份和日志备份均可以删除。备份删除后数据不可恢复，特别的是自动备份和日志备份被删除后备份数据不连续导致无法进行 PITR 恢复。

操作步骤

1. 登录 PostgreSQL 控制台，选择地域，单击实例 ID，进入实例管理页面。
2. 在实例管理页面，选择**备份恢复**页，在备份列表的**操作**列，单击**删除**。



3. 在弹出的对话框，确定要删除的备份文件，单击**确定**。

删除
删除后备份文件无法恢复，
确定删除备份文件[redacted].tar.gz？
确定
取消

查看备份空间

最近更新时间：2024-01-22 16:22:57

操作场景

对于双机高可用实例，云数据库 PostgreSQL 实例备份文件所占用的备份空间按地域分配，备份空间相当于该地域中所有 PostgreSQL 数据库备份的存储空间之和，包括自动数据备份、手动数据备份和日志备份。增加手动备份频次会相应增加数据库备份的存储空间。

查看备份空间

1. 登录 PostgreSQL 控制台，在左侧导航栏选择**数据库备份**页。
2. 在页面上方选择地域，**概览**页可查看对应地域的备份信息：备份总计、备份统计。

备份总计：展示总备份、数据备份、日志备份的空间大小和数量，以及总备份的免费空间和收费空间占用情况。

说明：

绿色：表示总备份使用量没有超过赠送空间。

橙色：表示总备份使用量已经超过赠送空间，产生费用，请参见 备份空间收费说明。

备份统计：展示该地域下实例的实例 ID/名称，实例状态，备份空间，数据备份大小和个数，日志备份大小和个数，自动备份大小和个数，手动备份大小和个数。

3. 在页面上方选择**备份列表页**，备份列表分为数据备份列表和日志备份列表。在列表单击实例名可以跳转到实例详情页。备份列表支持时间段筛选，及实例名和实例 ID 模糊搜索，支持实例 IP 的精准搜索。

数据备份列表

支持按备份时间点、任务开始时间、任务结束时间、备份大小进行排序。

在操作列单击**详情**可跳转到实例备份恢复页，单击**下载**可进行备份下载。

说明：

包年包月实例和按量计费实例在销毁时，为避免误操作导致系统不可恢复的情况发生，系统额外**赠送**一次类型为“最终”的全量物理备份（“最终”备份容量信息不计入备份空间统计，不参与计费），用户可在数据库备份的数据备份列表中进行下载使用。实例销毁7天后，“最终”类型的物理备份将自动删除。

日志备份列表

支持按日志数据开始时间、日志数据结束时间、备份大小字段进行排序。

在操作列单击详情可跳转到实例备份恢复页，单击**下载**可进行日志下载，单击**删除**可进行日志删除。

说明：

日志删除后将无法进行 PITR 恢复，请谨慎操作。

免费额度

云数据库 PostgreSQL 备份即将收费，在计费灰度期间，免费备份空间大小为您在对应地域下所有的主实例的存储空间之和乘以700%，在备份计费正式运行后，免费备份空间大小为您在对应地域下所有的主实例的存储空间之和乘以100%。具体请参见 备份空间即将收费说明。

热点问题

超出免费额度的备份空间如何收费？如何减少备份空间开销？

请参见 备份空间即将收费说明。

设置备份下载规则

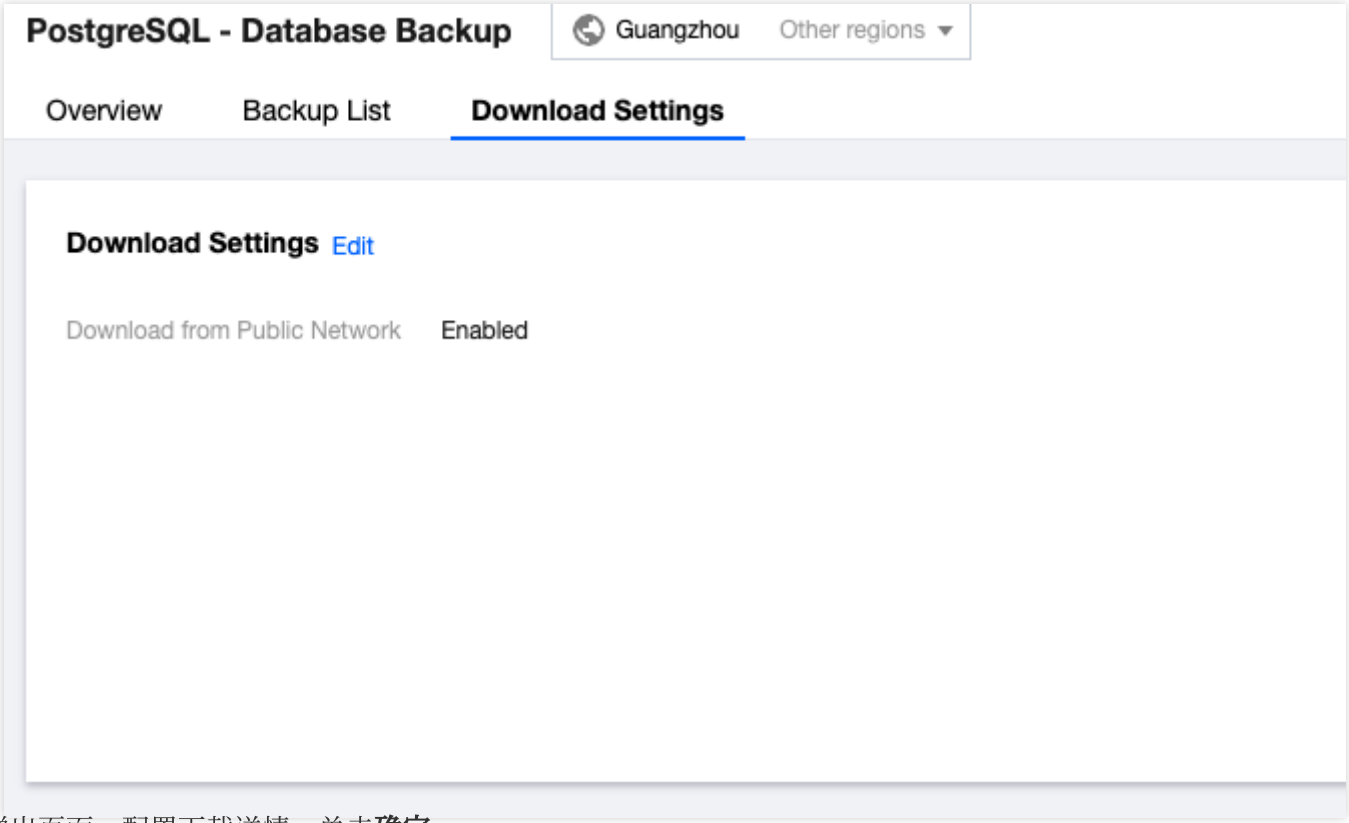
最近更新时间：2024-01-22 16:22:57

云数据库 PostgreSQL 实例备份文件默认可以通过公网或者内网进行下载，当您需要对下载进行限制时，可以通过下载配置进行相应的调整。下面详细说明。

1. 登录 PostgreSQL 控制台，在左侧导航选择**数据库备份**页，在上方选择地域。
2. 选择**下载配置**页，可查看对应地域的备份下载配置信息，单击**编辑**，对下载限制条件进行配置。

说明：

外网下载默认开启，开启外网下载时，外网和内网均可以下载备份。



3. 在弹出页面，配置下载详情，单击**确定**。

外网下载：

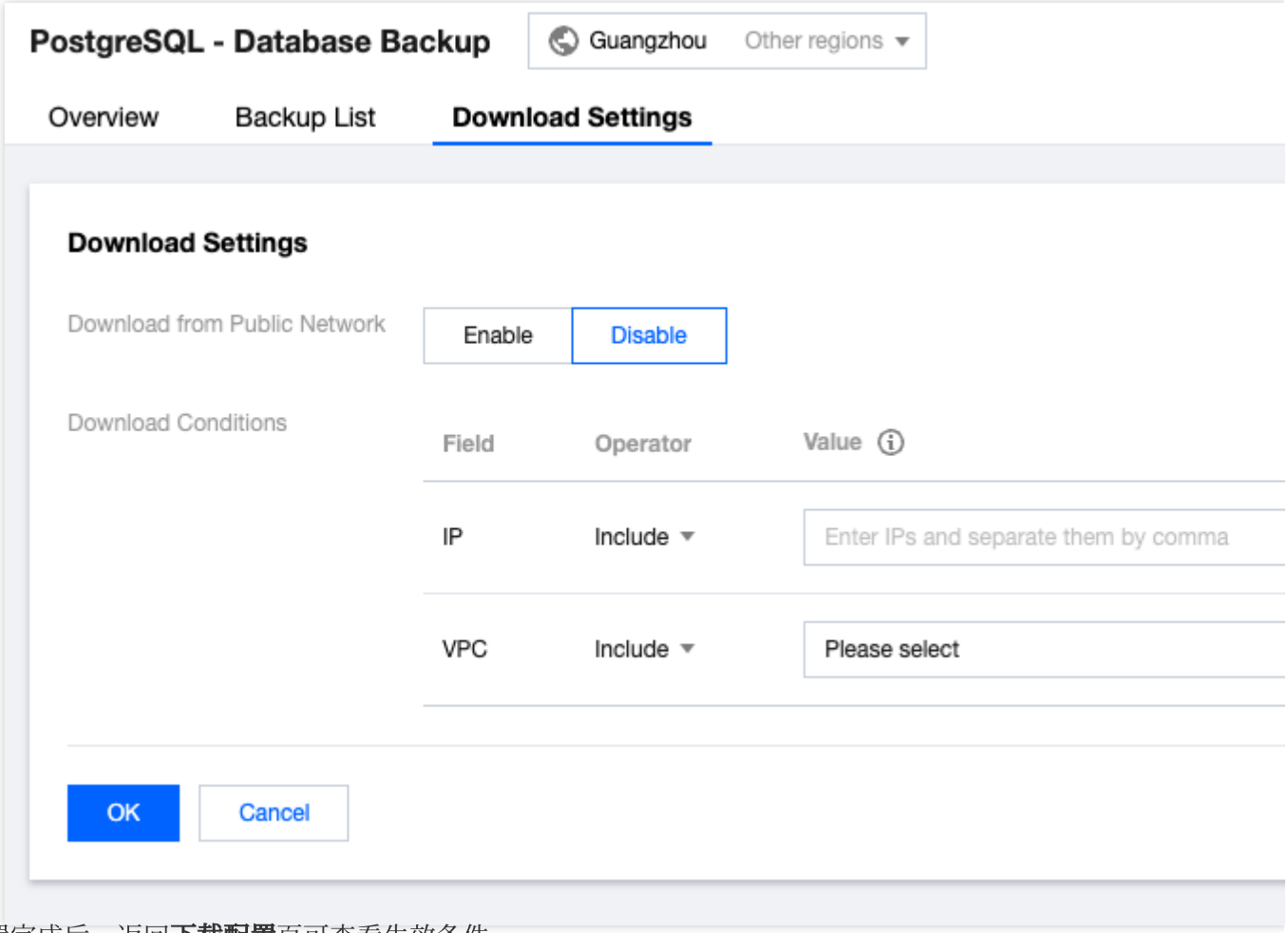
开启：外网下载开启时，无法对下载条件进行设置。

关闭：关闭外网下载时，可以对下载条件进行设置，设置条件包含 IP、VPC。

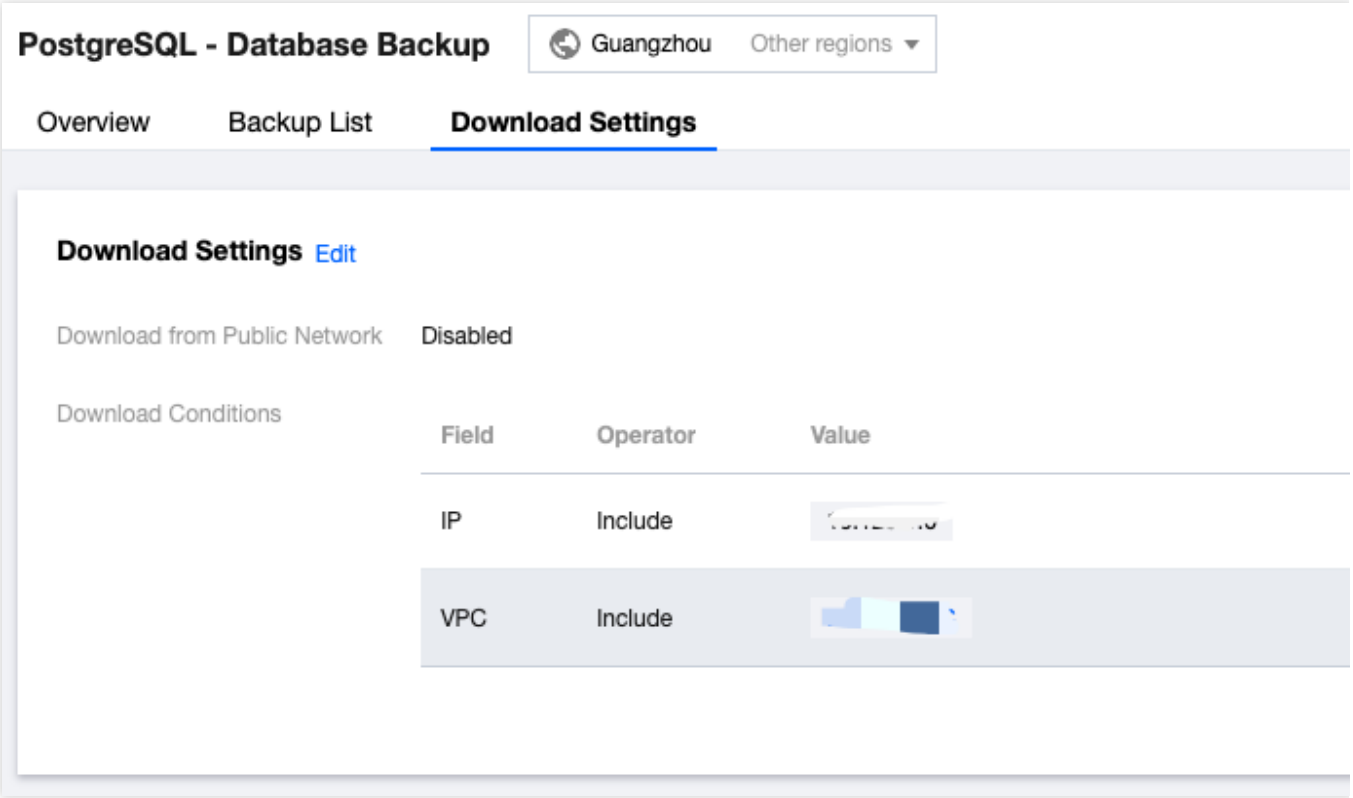
下载条件设置：

当条件对应的值为空时，代表不使用该条件进行限制。当所有条件均为空时，内网下载不受限制。

IP 地址条件值使用“,”进行分隔。IP 条件支持 IP 地址以及 IP 地址段作为条件值。



4. 配置完成后，返回**下载配置**页可查看生效条件。


PostgreSQL - Database Backup
Guangzhou
Other regions
Overview
Backup List
Download Settings
Download Settings
Edit
Download from Public Network
Disabled
Download Conditions
Field
Operator
Value
IP
Include
VPC
Include

使用云函数转存 PostgreSQL 历史备份

最近更新时间：2024-03-20 16:21:06

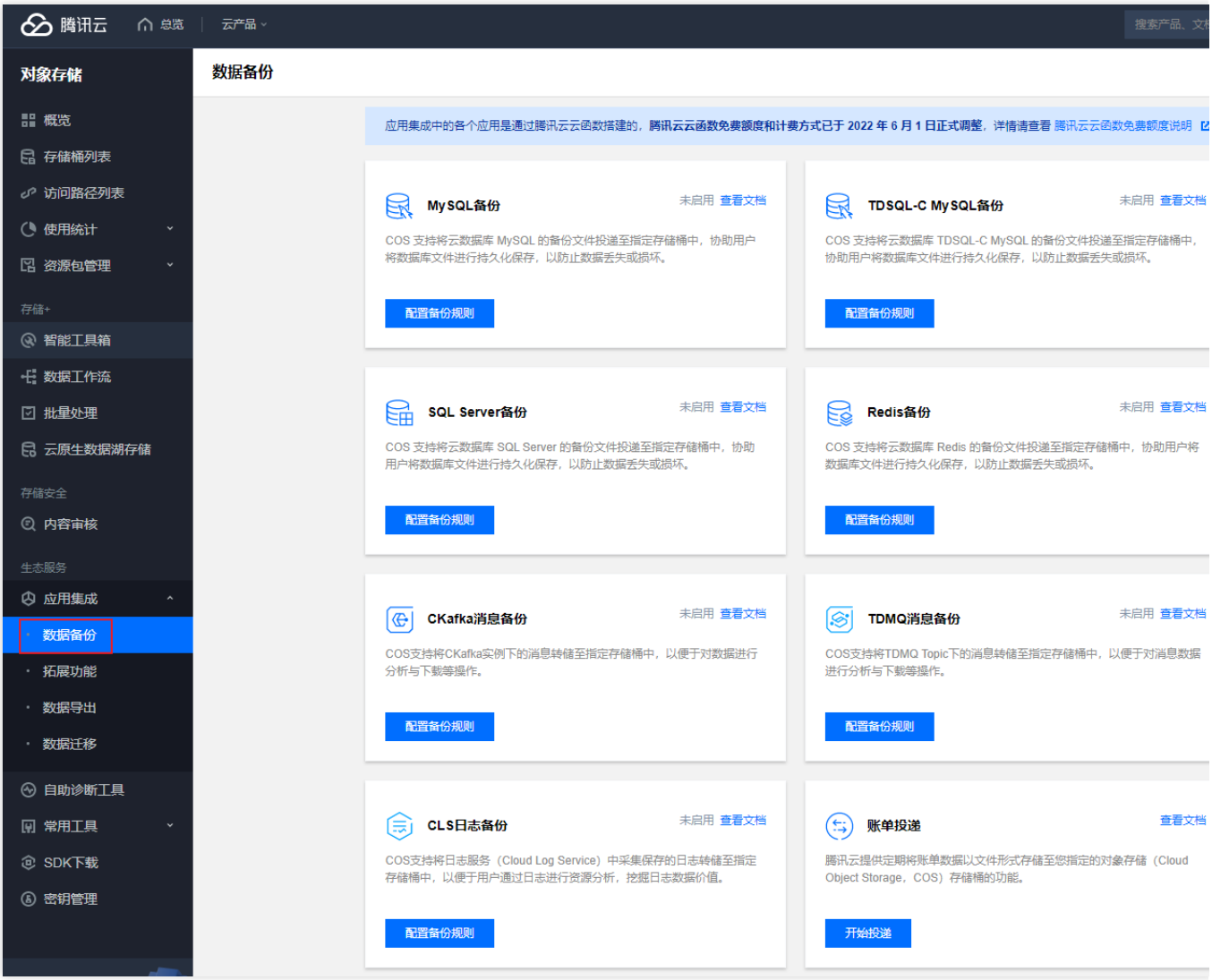
当前云数据库 PostgreSQL 支持备份数据保留时长3～7天，您也可以通过云函数将云数据库 PostgreSQL 的备份文件转存至对象存储进行长期的持久化的保存。

说明：

请在 对象存储控制台 通过严格的配置和监控确保数据拉取成功，否则在数据库侧的数据备份过期被自动删除后，可能会造成备份数据丢失。

操作步骤

1. 登录 对象存储 控制台。找到**应用集成** > **数据备份** > **PostgreSQL 备份**，如下图所示：



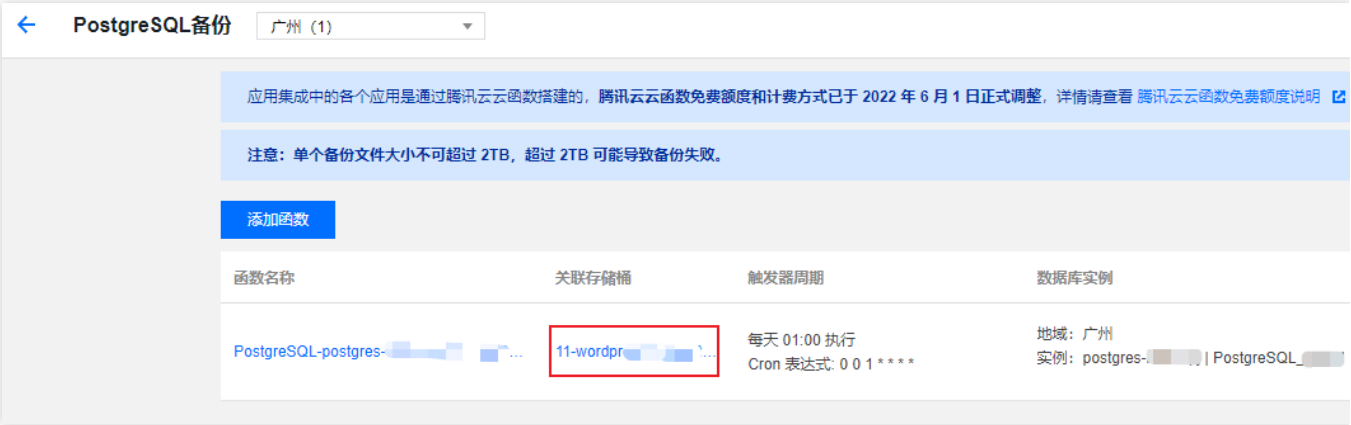
2. 单击**配置备份规则**，进入 PostgreSQL 备份配置页。单击**添加函数**，配置云函数拉取指定时间内 PostgreSQL 实例的数据备份和日志备份文件。

3. 配置完成后，在 PostgreSQL 备份函数页可以查看当前已经配置的备份函数列表，如下图所示：

4. 函数创建后可以单击**更多** > **触发** > **确认**来触发函数运行，该函数可以拉取云数据库 PostgreSQL 的备份数据，最长可获取最近7天的数据。

也可以使用默认的设置，由系统自动触发函数执行。

5. 备份数据拉取成功之后，单击**关联存储桶**列的对应存储桶地址，即可看到被保存的数据。



说明：

云函数的运行日志可以到 云函数控制台 进行查看。不建议在云函数控制台直接修改拉取备份的函数。

拉取云数据库 PostgreSQL 历史备份可能会产生相关费用，具体收费细节请查看 对象存储费用说明、云函数费用说明 和 日志服务费用说明。

回档至原实例

最近更新时间：2024-08-09 15:15:45

说明：

回档至原实例支持 database 级别对象的操作。

对于 SQL Server 兼容引擎的实例，暂不支持 mssql_compatible 库的回档。

单次回档任务最多选择100个 database 。

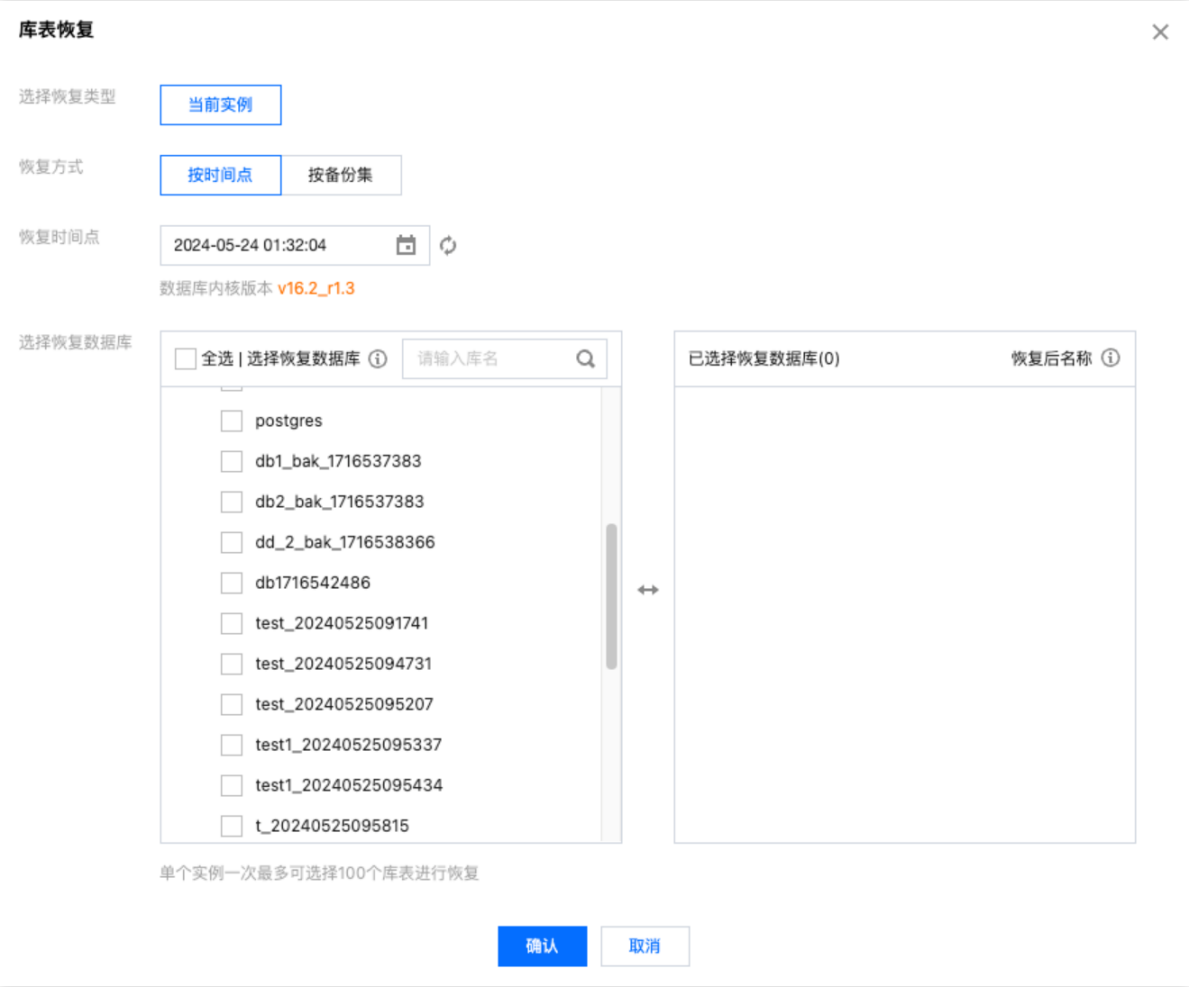
回档至原实例可以将当前实例中已经存在的 database 对象按照时间点或者备份集还原至当前实例。同时也能解决误删 database 的恢复问题。下面分别为您介绍。

选择恢复方式与时间点

注意：

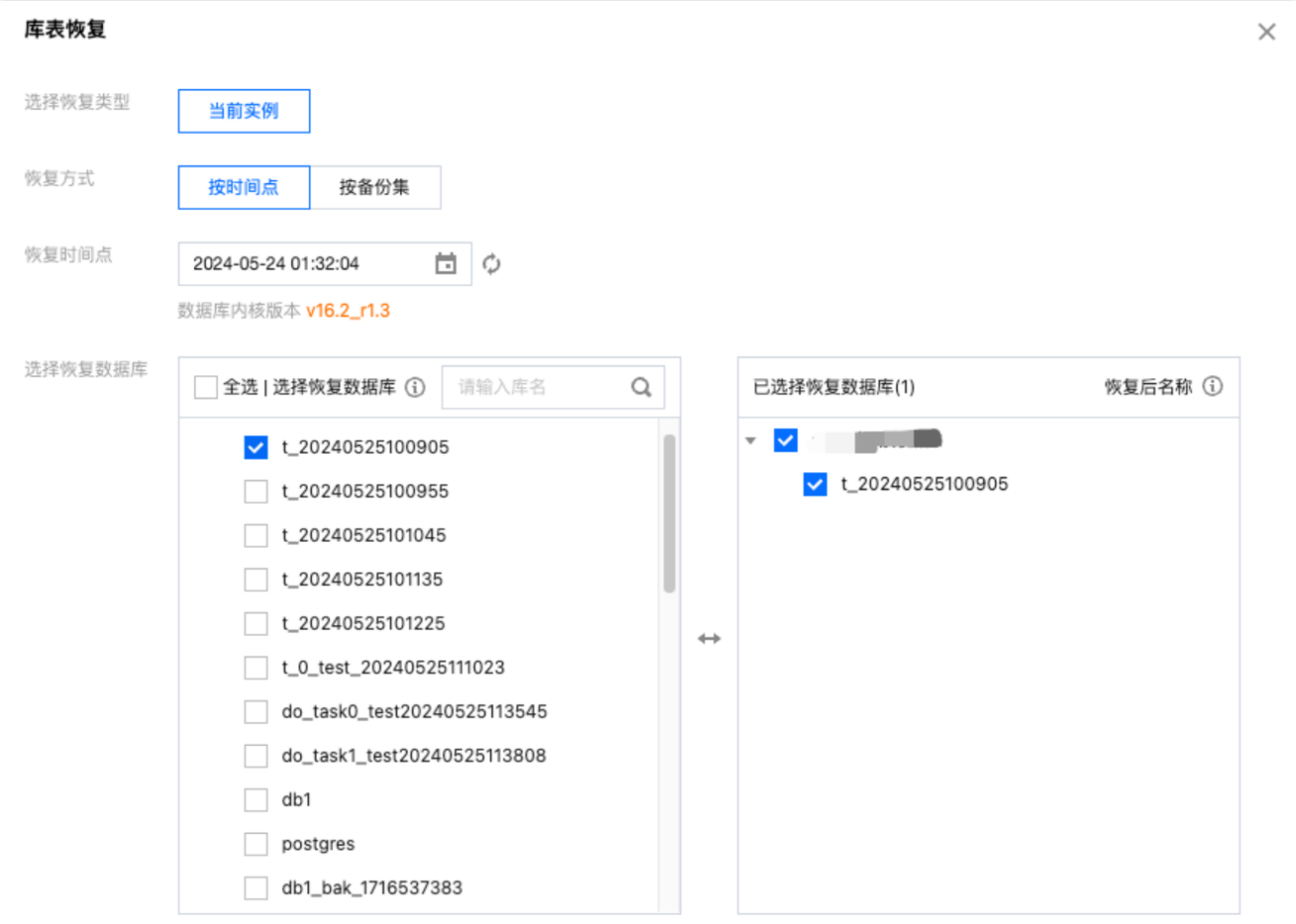
库表恢复的可选择时间跟您当前备份的保留策略强相关。备份设置请参考 指引。

您可以选择按照恢复到某个备份集或者选择恢复至系统检测到有数据的任意时间点。



选择恢复数据库

您可以选择当前系统拉取到的已存在 database 进行恢复。具体如下图所示：

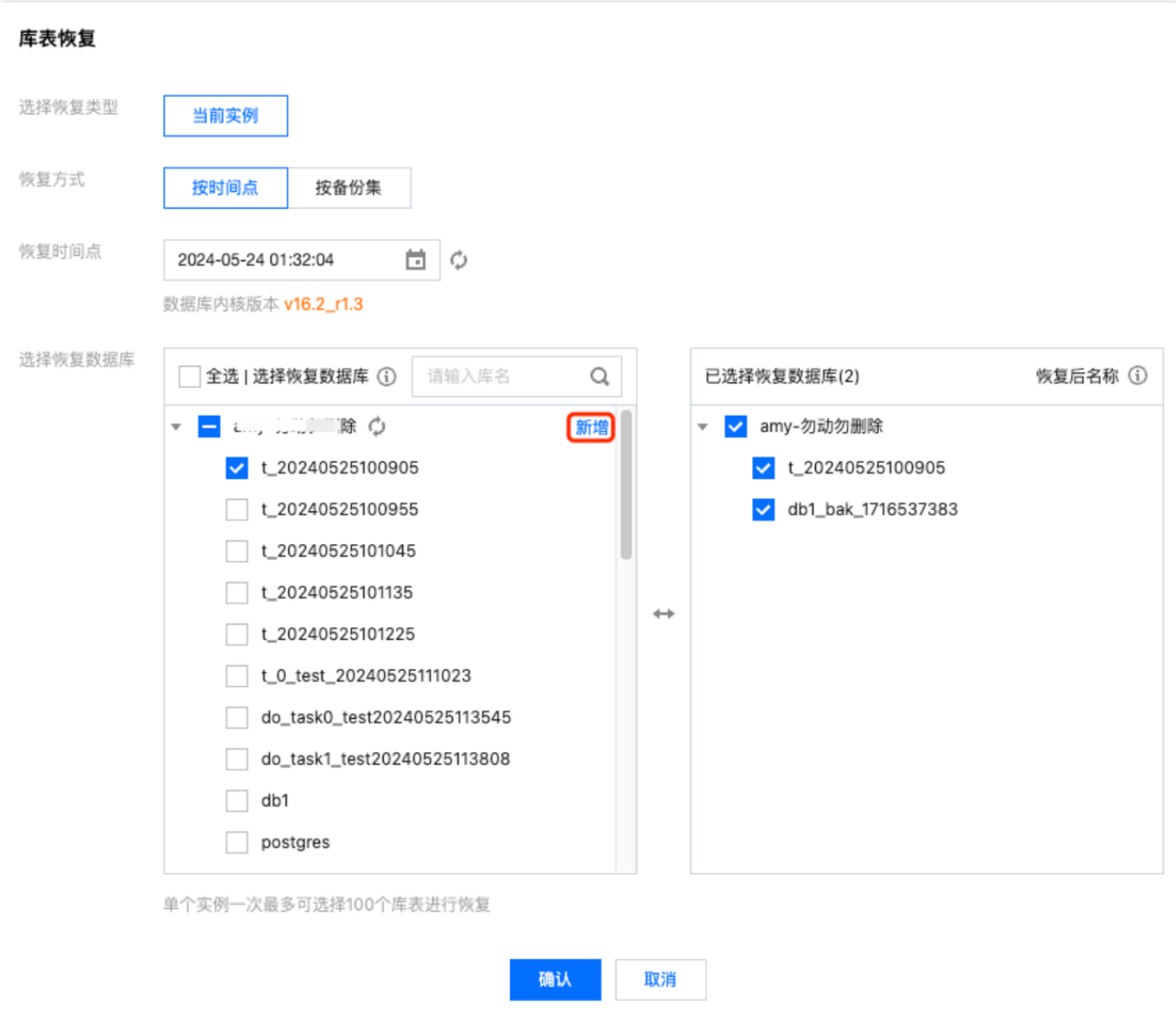


恢复后名称

恢复后的 database 名称将使用 *_bak_timestamp 后缀标识。timestamp 为一个后台任务发起后的一个 Unix 时间戳。当您在控制台或者 API 提交库表回档任务后，后台将会在5分钟内发起回档任务。如您选择的 database 名称为 dbone，后台发起任务的时间为2024-05-30 11:26:25，则恢复任务完成后，恢复至原实例的新数据库名为 dbone_bak_1717039585。

误删数据库的处理

当您由于误操作导致某个 database 被删除，则您也能通过库表回档功能解决此问题。由于 database 被误删除，因此您发起任务时拉取不到该 database ，因此，您可通过新增 database 的方式来发起任务。单击下图中的**新增**。



注意：

如果您新增 database 名称在所选备份集或者 PITR 时间点中不存在，则会恢复出对应的空 database。

物理迁移
配置物理迁移任务

最近更新时间：2024-07-23 11:54:41

物理迁移采用 PostgreSQL 物理流复制方式实现，速度更快、稳定性更好。本文为您介绍如何配置物理迁移任务及相关的注意事项。

配置任务

1. 登录 云数据库 PostgreSQL 控制台。
2. 在左侧导航栏选择**数据迁移**。
3. 进入数据迁移任务列表。单击**新建迁移任务**，建立一个源实例和目标实例类型为 PostgreSQL 的迁移任务。

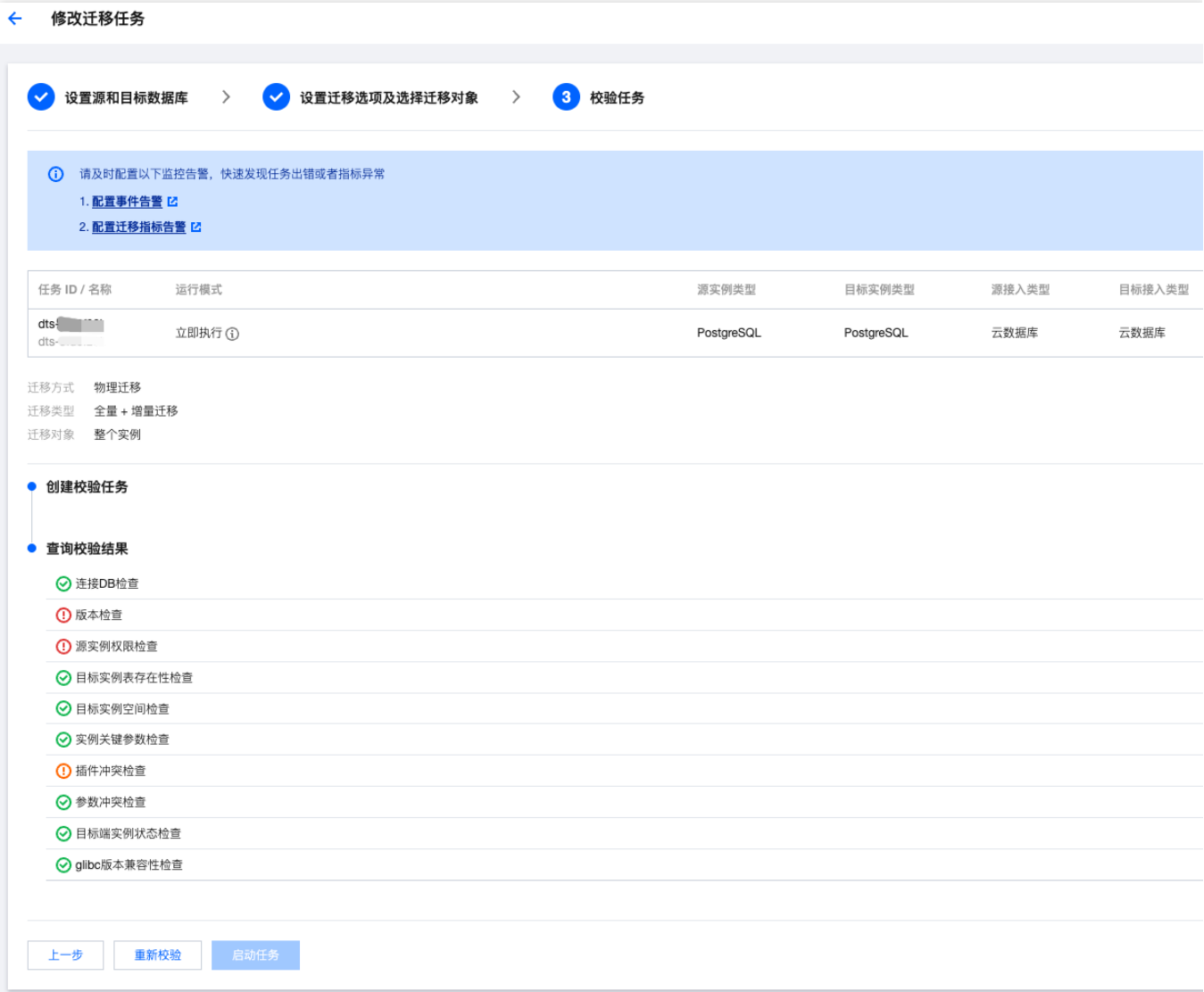
4. 建立迁移任务之后，进入迁移任务列表，单击**操作** > **配置**，对新建的任务进行配置。

5. 配置任务分为三步，需要注意，在第二步时，选择迁移方式为**物理迁移**。迁移类型和迁移对象选择默认。随后进行校验任务。

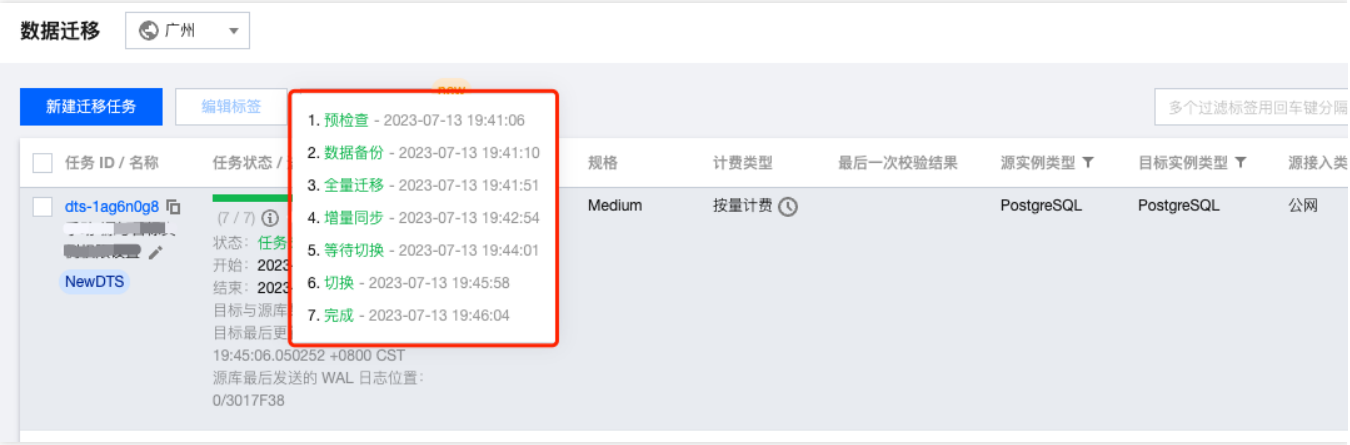
6. 系统校验细则及修复方法请参考 物理迁移检查项。

注意：

如果源实例为自建 PostgreSQL ，请务必保证 PostgreSQL 数据目录中不包含其他任何自建的文件或目录，否则会导致流程失败。



7. 全部校验通过之后，任务正式启动，您可以到任务列表查看任务详情。整个任务总共分为七个步骤，如下图所示：



迁移步骤详情描述请参考 迁移步骤说明。

迁移步骤说明

1. 预检查

系统在发起迁移任务之前，对源和目标端实例进行多项环境检查，包含如下几项：
目标实例会话检查：检查目标实例是否存在用户会话，如果有，则系统报错。用户会话主要是指如下三类数据库账号以外的账号发起的会话。
postgers
repluser
以 tencentdb_开头的账号
目标实例库表检查：检查目标实例是否存在用户的 database 或者对象，若存在，则系统报错。
目标实例关联实例检查：检查目标实例是否有关联的 RO 实例，如果存在则报错。

2. 数据备份

为保证目标实例数据可恢复性，系统会先对目标实例进行一次全量自动备份，再利用 PostgreSQL 原生工具 pg_basebackup 在腾讯云 PostgreSQL 环境上搭建源实例的备机，并准备开始进行主备数据同步。云数据库 PostgreSQL 备份空间已经计费，超过实例赠送空间部分将产生费用，具体费用详情请参考 备份空间收费说明。

3. 全量迁移

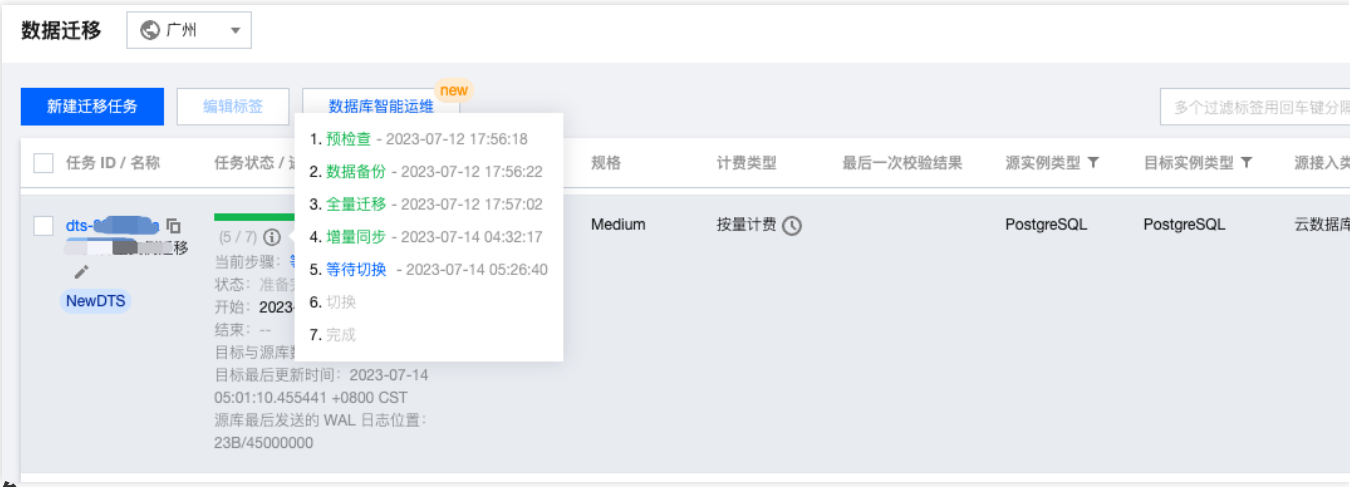
利用 PostgreSQL 物理流复制方式实现，实现源实例和腾讯云 PostgreSQL 侧备机之间的主备全量同步。

4. 增量同步

利用 PostgreSQL 物理流复制方式实现，实现源实例和腾讯云 PostgreSQL 侧备机之间的主备增量同步。业务需要及时观测任务列表上主备同步进度，当同步一致后，请业务确保源实例需要停止写入。

5. 等待切换

当源实例与腾讯云 PostgreSQL 侧备机之间的主备增量同步差异为0时，系统处于等待切换状态。用户需要手工单击**完成**来发起切换。



6. 切换

用户单击**完成**后，系统会将腾讯云 PostgreSQL 侧备机提升为主机。

注意：

系统检测到瞬时源和目标 LSN 一致后可以进行切换，但并不保证源实例已经停写，请在进行切换前务必确保源实例业务已停止写入。

7. 完成

在切换完成后，备机升主。腾讯云 PostgreSQL 管理系统将纳管新的主机，并适配管控操作。

注意事项

实例需要为主实例，只读实例不支持物理迁移。

物理复制不支持对开启数据透明加密的源端或者目标端进行数据传输。

物理迁移检查项

最近更新时间：2024-04-09 10:10:21

云数据库 PostgreSQL 物理迁移发起任务前检查项具体如下：

连接 DB 检查

具体检查详情、原因及修复方法请参考 源实例或目标实例连通性检查。

版本检查

源实例和目标实例的大版本需要保持一致，且目标实例的内核版本不能低于以下版本。

大版本	内核版本
10	v10.17_r1.5
11	v11.12_r1.5
12	v12.7_r1.5
13	v13.3_r1.4
14	v14.2_r1.11
15	v15.1_r1.4

源实例权限检查

检查详情

源实例迁移账号需要拥有 LOGIN、REPLICATION 权限。
源实例 pg_hba.conf 文件配置需要满足系统要求。

修复方法

用户可能不具备操作权限，请按照检查要求中的对应权限要求对用户进行授权，然后重新执行校验任务。
如果报错类似 `connect source with replication failed: pq: no pg_hba.conf entry for replication connection from host "xx.xx.xxx.xx", user "xxx"` ，说明源实例 pg_hba.conf 文件配

置不满足迁移要求，可以在源实例 pg_hba.conf 文件中增加 `host replication all 0.0.0.0/0 md5` 配置后重启实例或者执行 `select pg_reload_conf();` 重新加载配置即可。

注意：

当源和目标实例数据库账号存在权限和密码冲突时，系统默认使用源实例账号配置。

如果源实例为其他云厂商的 PostgreSQL 实例，请使用最高权限账号，方式迁移任务校验失败。

目标实例表存在性检查

检查详情

检查目标实例是否为空。

说明：

目标实例可以有用户新建的数据库账号，但是不能有创建好的 database 或者对象。

修复方法

需要删除目标实例上已经存在的 database 或者对象。

目标实例空间检查

检查详情

目标实例磁盘空间需要大于源端实例的磁盘占用空间大小110%，否则会报错。

修复方法

对目标端进行磁盘空间扩容，具体请参考 调整实例配置。

实例关键参数检查

检查详情

源实例 wal_level 必须为 logical 。

源和目标实例的 block_size 必须相同。

PostgreSQL13以下版本实例，源实例 wal_keep_segments 必须大于或等于256。

PostgreSQL13及以上版本，源实例 wal_keep_size/wal_segment_size 的值必须大于或等于256。

源和目标实例的 wal_block_size 必须相同。

源和目标实例的 segment_size 必须相同。

目标实例的 max_connections 必须大于或等于源实例的 max_connections 。

目标实例的 max_wal_senders 必须大于或等于源端的 max_wal_senders 。

目标实例的 max_worker_processes 必须大于或等于源端的 max_worker_processes 。

修复方法

需要按照检查详情中的要求来对应修改源或者目标实例的参数。

插件冲突检查

检查详情

检查源实例安装的插件是否在目标实例支持的插件列表，如果不在，则系统报错。同时，如果源实例插件版本跟目标实例插件版本不一致，则系统发布警告。

修复方法

通过安装插件或者升级插件版本来保证源和目标实例的插件保持一致。

参数冲突检查

检查详情

原则上源和目标实例的参数值需要保持一致，当系统检测到源和目标端参数值不一致时，通常的处理方式是**使用目标实例的参数值**并**不提醒用户**。不同处理方式的为如下几个参数：

参数名	迁移默认处理方式	校验动作
data_checksums	使用源实例参数值	参数值不一致时不提醒
enable_partitionwise_aggregate	使用源实例参数值	参数值不一致时不提醒
enable_partitionwise_join	使用源实例参数值	参数值不一致时不提醒
lc_ctype	使用源实例参数值	参数值不一致时不提醒
max_locks_per_transaction	使用源实例参数值	参数值不一致时不提醒
max_prepared_transactions	使用源实例参数值	参数值不一致时不提醒

random_page_cost	使用源实例参数值	参数值不一致时不提醒
max_connections	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
max_wal_senders	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
max_worker_processes	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
array_nulls	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
authentication_timeout	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
autovacuum_analyze_scale_factor	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
autovacuum_analyze_threshold	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
autovacuum_freeze_max_age	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
autovacuum_multixact_freeze_max_age	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
autovacuum_naptime	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
autovacuum_vacuum_cost_delay	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
autovacuum_vacuum_cost_limit	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
autovacuum_vacuum_insert_scale_factor	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
autovacuum_vacuum_insert_threshold	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
autovacuum_vacuum_scale_factor	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式

版权所有：腾讯云计算（北京）有限责任公司

autovacuum_vacuum_threshold	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
bytea_output	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
check_function_bodies	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
constraint_exclusion	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
cursor_tuple_fraction	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
DateStyle	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
deadlock_timeout	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
default_statistics_target	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
default_transaction_isolation	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
exit_on_error	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
extra_float_digits	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
from_collapse_limit	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
geqo	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
geqo_effort	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
geqo_generations	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
geqo_pool_size	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式

版权所有：腾讯云计算（北京）有限责任公司

geqo_seed	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
geqo_selection_bias	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
geqo_threshold	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
idle_in_transaction_session_timeout	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
idle_session_timeout	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
IntervalStyle	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
jit	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
jit_above_cost	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
jit_inline_above_cost	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
jit_optimize_above_cost	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
join_collapse_limit	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
lc_monetary	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
lc_numeric	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
lc_time	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
local_preload_libraries	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
log_filename	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式

版权所有：腾讯云计算（北京）有限责任公司

max_logical_replication_workers	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
max_parallel_workers	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
max_replication_slots	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
max_standby_archive_delay	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
max_standby_streaming_delay	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
recursive_worktable_factor	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
search_path	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
statement_timeout	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
stats_fetch_consistency	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
TimeZone	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
vacuum_cost_delay	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
vacuum_cost_limit	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
vacuum_cost_page_dirty	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
vacuum_cost_page_hit	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
vacuum_cost_page_miss	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
vacuum_freeze_min_age	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式

版权所有：腾讯云计算（北京）有限责任公司

vacuum_freeze_table_age	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
vacuum_multixact_freeze_min_age	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
vacuum_multixact_freeze_table_age	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式
wal_level	使用目标实例参数值	参数值不一致时提醒用户修改，如未修改，则采用迁移默认处理方式

修复方法

请按照实际需要修改参数值。

目标端实例状态检查

检查详情

目标实例不能开启 TDE 数据加密。
目标实例可以有只读组但不能有关联的只读实例。
目标实例当前不允许有用户连接。
注意：
进行物理迁移时，需要保证源实例不是加密实例。

修复方法

暂不支持 TDE 实例和只读实例。

GLIBC版本兼容性检查

检查详情

检查源和目标实例 GLIBC 版本兼容问题。
说明：
GLIBC 2.28版本 UTF8 部分字符排序规则发生变化，版本不兼容时，数据排序规则不同，存在排序结果不符合预期的风险。

修复方法

升级 GLIBC 版本。

数据库审计
审计服务说明

最近更新时间：2024-07-23 12:29:48

审计分类

云数据库 PostgreSQL 支持审计类型为**审计极速版**和**审计精细版**。具体说明如下：

1. **审计极速版，**对性能的影响最小，与原生社区 PostgreSQL 开启 log_statement=all 效果一致，除此之外审计极速版还额外记录了影响行数、执行时长。

2. **审计精细版**使用 pgaudit 插件开启全审计，审计日志会记录的更详细，可以审计 SQL 类型、对象名。但相较于极速版理解成本更高，适合有特定需求的开发人员使用。

说明：

单条 SQL，如果存在子查询或函数等的对象调用关系，审计精细版会记录多条日志，分别为多个对象被调用的信息，为了避免同样的语句打印多次，第二次及以后的语句显示为 `previously logged` ，SQL类型为 `???` 。

下面将基于几种不同的场景来对比**审计极速版**和**审计精细版**记录的日志。

函数调用

具体的 SQL 如下：

```
CREATE FUNCTION a_t(integer, integer) RETURNS integer
AS 'select $1 + $2;'
LANGUAGE SQL;
select a_t(2,3);
```

精细审计日志如下图所示：

时间	客户端 IP	账户名称	SQL 类型	数据库名	执行语句	对象类型	会话 ID	对
2023-10-13 11:14:28		n	SELECT	amyt	SELECT a_t(2, 3) LIMIT 11 OFFSET 0	--		
2023-10-13 11:14:24		n	SELECT	amyt	<previously logged>	TABLE		
2023-10-13 11:14:24		n	SELECT	amyt	<previously logged>	TABLE		

极速审计日志如下图所示：

时间	客户端 IP	账户名称	数据库名	执行语句	会话 ID
2023-10-13 11:43:04			amyt	SELECT a_t(2, 3) LIMIT 11 OFFSET 0	

表关联

具体的 SQL 如下：

```
create table a(id integer,name varchar);
```

```
create table b(id integer,age int);
insert into a(id,name)values(1,'anne'),(2,'bob');
insert into b(id,age)values(2,30);
select a.id,name,age from a,b where a.id=b.id;
```

精细审计日志如下图所示：

时间	客户端 IP	账户名称	SQL 类型	数据库名	执行语句	对象类型	会话 ID	对象名
2023-10-13 14:53:38			SELECT	amyt	<previously logged>	TABLE		public
2023-10-13 14:53:38		n	SELECT	amyt	SELECT a.id, name, age FROM a, b WHERE a.id = b.id ...	TABLE		public

极速审计日志如下图所示：

时间	客户端 IP	账户名称	数据库名	执行语句	会话 ID
2023-10-13 14:53:34			amyt	SELECT a.id, name, age FROM a, b WHERE a.id = b.id ...	

子查询

精细审计日志如下图所示：

时间	客户端 IP	账户名称	SQL 类型	数据库名	执行语句	对象类型	会话 ID	对
2023-10-13 15:11:08			SELECT	amyt	<previously logged>	TABLE		p
2023-10-13 15:11:08			SELECT	amyt	SELECT c.id FROM (SELECT a.id, name, age FROM a, b ...	TABLE		p

极速审计日志如下图所示：

时间	客户端 IP	账户名称	数据库名	执行语句	会话 ID
2023-10-13 15:11:18			amyt	SELECT c.id FROM (SELECT a.id, name, age FROM a, b ...	

存储过程

存储过程的定义及调用如下所示：

```
CREATE OR REPLACE PROCEDURE update_m(
p_city in integer,
p_ldate in date,
p_id in integer)
AS $$
BEGIN
update m
set city = p_city,
ldate = p_ldate
where city = p_id;
```

```
END; $$
LANGUAGE plpgsql;


call update_m(4,'2023-02-05',4);
```

审计精细版日志如下图所示：

时间	客户端 IP	账户名称	SQL 类型	数据库名	执行语句	对象类型	会话 ID
2023-10-13 15:48:58			???	amyt	<previously logged>	TABLE	
2023-10-13 15:48:58			CALL	amyt	call update_m(4,'2023-02-05',4)	--	
2023-10-13 15:48:58			EXECUTE	amyt	call update_m(4,'2023-02-05',4)	FUNCTION	
2023-10-13 15:48:58			UPDATE	amyt	update m set city = p_city, ldate = p_lda ...	TABLE	
2023-10-13 15:48:58			???	amyt	<previously logged>	TABLE	
2023-10-13 15:48:41			CREATE PROCEDURE	amyt	CREATE OR REPLACE PROCEDURE update_m(p_city in ...	--	

审计极速版日志如下图所示：

时间	客户端 IP	账户名称	数据库名	执行语句	会话 ID
2023-10-13 15:49:24			amyt	call update_m(4,'2023-02-05',4)	
2023-10-13 15:49:17			amyt	CREATE OR REPLACE PROCEDURE update_m(p_city in ...	

记录日志说明

1. **审计极速版**和**审计精细版**记录的 SQL 语句类型与使用 log_statement = all 设置所记录的 SQL 类型一致，即会记录所有 simple query 协议和 execute 协议的 SQL。在扩展查询协议（即 extened query）的情况下，在执行阶段之前（即在解析分析或规划期间）出错的语句不会被记录。
2. **审计极速版**和**审计精细版**记录的 SQL 语句的长度默认为8192字节，超过限制长度的语句将被截断，同时该 SQL 语句的对象类型、对象名称、执行时间、影响行数也会无法展示。如您需要自定义，请在控制台修改数据库参数 tencentdb_audit_message_truncate_length。当 SQL 语句被截断后，如果该 SQL 语句为慢 SQL 或者执行失败的 SQL，您可到 云数据库 PostgreSQL 控制台 的**性能优化** > **慢查询分析**或者**错误日志**中查看完整 SQL 详情。
3. 当前大版本为11的云数据库 PostgreSQL 的实例不支持影响行数的统计。
4. 因审计和慢查询使用的计时系统的差异，二者记录的 SQL 执行时间会有毫秒级差异。

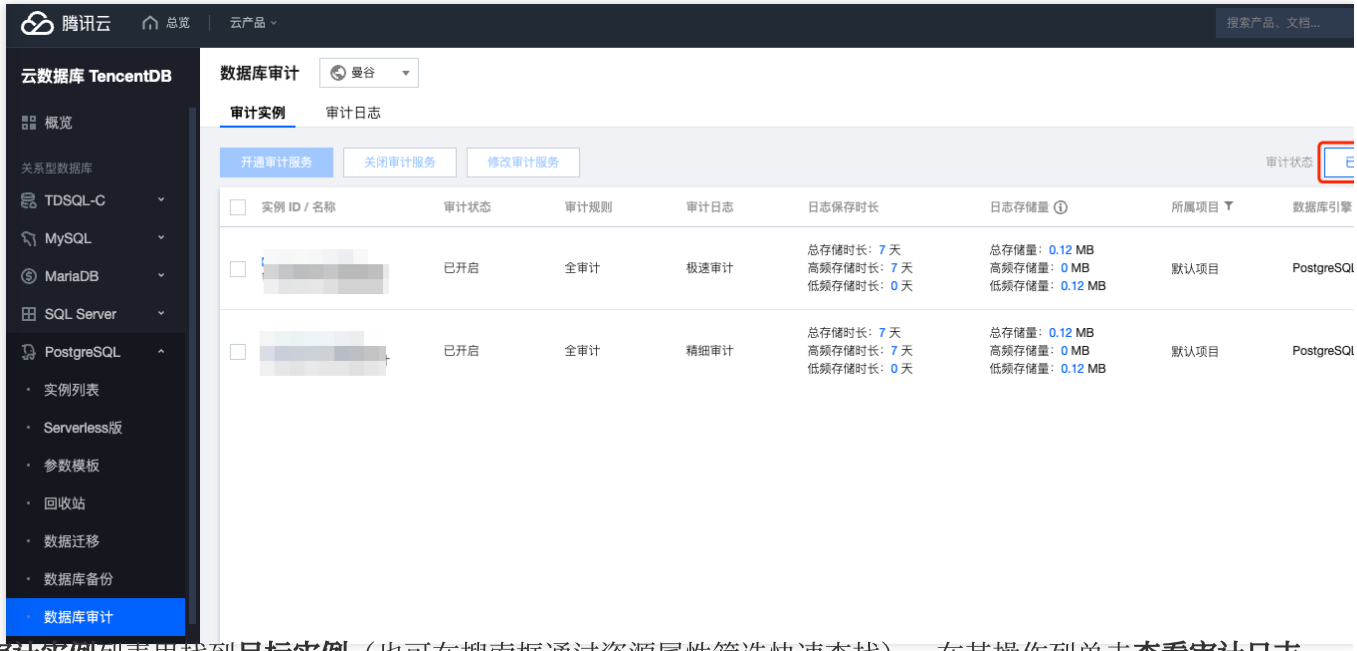
查看审计日志

最近更新时间：2024-07-23 12:31:38

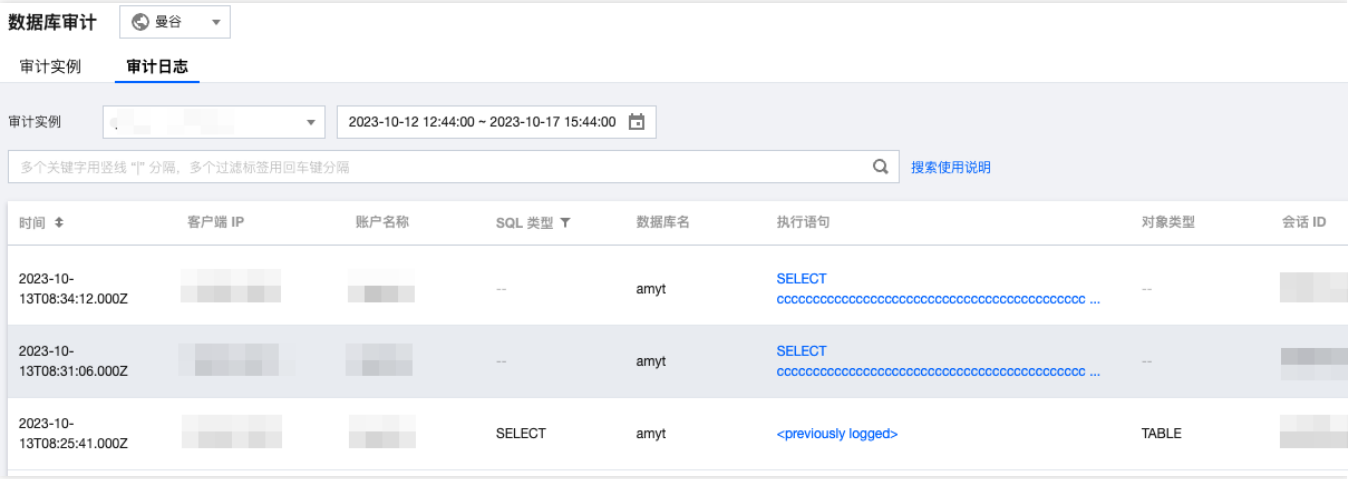
本文为您介绍如何查看数据库审计日志及相关审计日志列表的字段。

查看审计日志

1. 登录 云数据库 PostgreSQL 控制台。
2. 在左侧导航栏选择**数据库审计**。
3. 在上方选择**地域**后，在**审计实例**页，单击**审计状态**选择**已开启**选项过滤已开启审计的实例。



4. 在**审计实例**列表里找到**目标实例**（也可在搜索框通过资源属性筛选快速查找），在其操作列单击**查看审计日志**，跳转至**审计日志**页查看对应日志。审计极速版与审计精细版对应审计日志的区别，请查看 审计服务说明。



搜索工具说明

在**审计实例筛选框**，可选择切换已开启审计服务的其他审计实例。

在**时间框**，选择时间段（日志保留时间），可查看所选时间段内相关审计日志。

说明：

搜索时间段支持选取存在数据的任意时间段进行搜索，最多展示符合条件的前60000条记录。

在**搜索框**，选择搜索项（SQL 命令详情、客户端 IP、账号名称、数据库名、SQL 类型、执行时间（毫秒）、影响行数）进行搜索，可查看相关审计结果，多个搜索项使用回车键进行分隔。

所有的搜索项均使用包含的方式进行匹配。

搜索项	说明
SQL 命令详情	输入 SQL 命令详情，多个关键字使用换行符进行分隔。 SQL 命令详情搜索不区分大小写。 当匹配类型为包含/不包含时，仅支持分词维度模糊搜索，不支持通配符模糊搜索。
客户端 IP	输入客户端 IP，多个关键字使用换行符进行分隔；IP 地址支持使用 * 作为条件进行筛选。
账号名称	输入用户账号，多个关键字使用换行符进行分隔。
数据库名	输入数据库名，多个关键字使用换行符进行分隔。
SQL 类型	下拉选择 SQL 类型（ALTER、ANALYZE、BEGIN、CALL、CHECKPOINT、CLOSE、COMMENT、COMMIT、COPY、CREATE、DEALLOCATE、DECLARE、DISCARD、DO、DROP、EXECUTE、EXPLAIN、FETCH、GRANT、IMPORT、LISTEN、LOAD、LOCK、MOVE、NOTIFY、PREPARE、REASSIGN、REFRESH、REINDEX、RELEASE、RESET、REVOKE、ROLLBACK、SAVEPOINT、SECURITY、SELECT、SET、SHOW、START TRANSACTION、TRUNCATE、UNLISTEN、UPDATE、VACUUM），支持多选。
执行时间（毫秒）	输入执行时间，格式为数字 N，支持筛选 N 毫秒以上的匹配。

影响行数	输入影响行数，格式为数字 N，支持筛选 N 行以上的匹配。

日志下载

您可以先生成审计日志文件后再进行下载。具体操作如下：

1. 在审计日志页面，单击右侧的

，生成审计日志文件。

2. 生成审计日志文件后，可在文件列表中进行下载。

说明：

审计日志文件最多保存搜索结果中的前6000000条记录。

目前日志文件下载仅提供腾讯云内网地址，请通过同一地域的腾讯云服务器进行下载（例如：北京区的 PostgreSQL 实例审计日志请通过北京区的 CVM 下载）。

日志文件有效期为24小时，请及时下载。

wget 命令格式：wget -c '<日志文件下载地址>' -O <自定义文件名>.tar.gz。

每一个数据库实例的日志文件不得超过30个，请下载后及时删除清理。

版权所有：腾讯云计算（北京）有限责任公司

修改审计服务

最近更新时间：2024-04-09 10:26:31

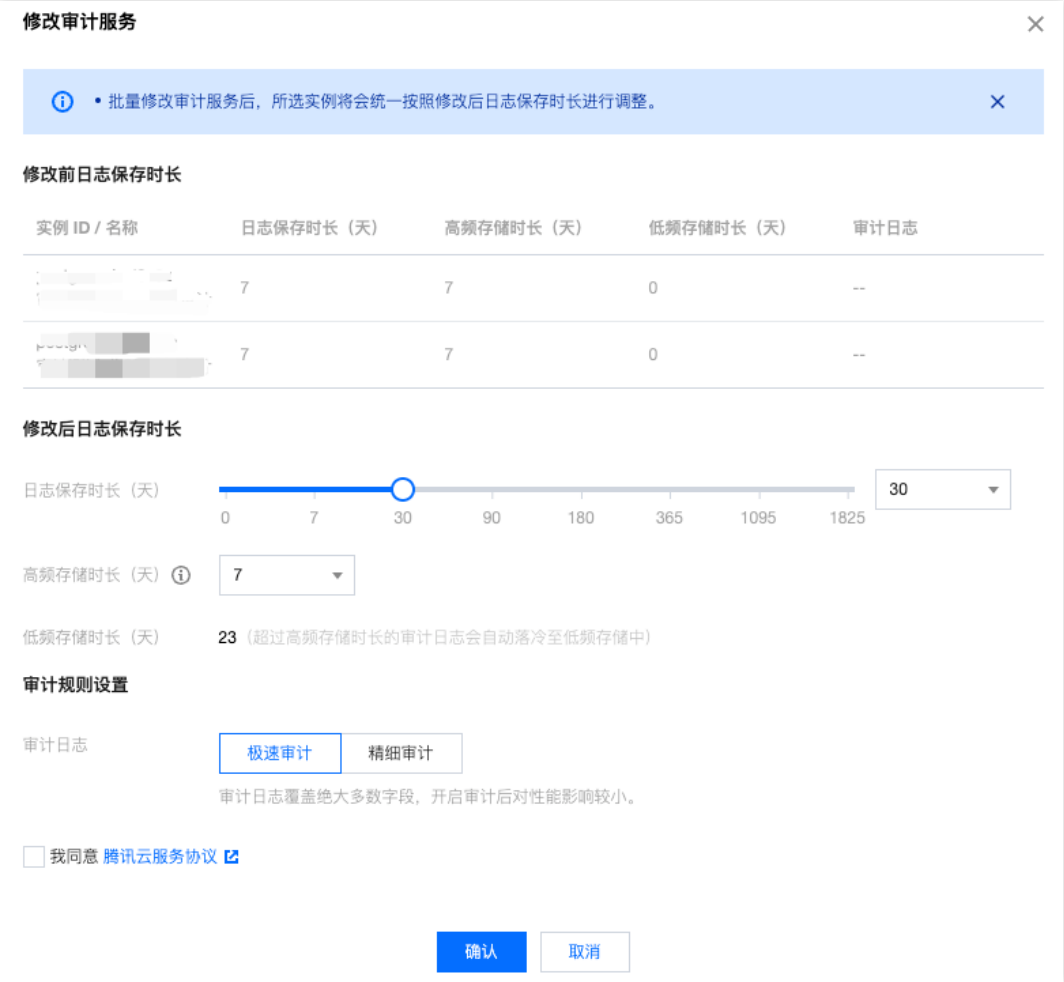
本文为您介绍审计服务设置和关闭功能。

审计服务设置

具体操作，可参见如下两种方式设置审计服务。

方式一：

1. 登录 云数据库 PostgreSQL 控制台。
2. 在左侧导航栏选择**数据库审计**。
3. 在上方选择**地域**后，在**审计实例**页，单击**审计状态，**选择**已开启**选项过滤已开启审计的实例。
4. 您可以在审计实例列表进行单次或批量的审计服务修改，具体如下图所示：



方式二：

1. 登录 云数据库 PostgreSQL 控制台。
2. 在左侧导航栏选择**数据库审计**。
3. 在上方选择**地域**后，在**审计实例**页，单击**审计状态**选择**已开启**选项过滤已开启审计的实例。
4. 单击**实例 ID**，进入该实例的审计日志列表，单击**服务设置**，进行该实例的审计服务设置。

关闭审计

注意：

1. 审计服务关闭后，将会停止对实例进行审计且历史审计日志将被清空。请谨慎操作。
2. 切换极速审计或者精细审计不会清空历史审计日志。

您可以在审计实例列表单次或批量关闭审计服务，也可以在审计日志列表也进行单个实例的审计设置。

1. 登录 云数据库 PostgreSQL 控制台。
2. 在左侧导航栏选择**数据库审计**。
3. 在上方选择**地域**后，在**审计实例**页，单击**审计状态**，选择**已开启**选项过滤已开启审计的实例。
4. 您可以在审计实例列表进行单次或批量的审计服务修改，具体如下图所示：

5. 您也可以单击**实例 ID**，进入审计日志详情，进行单个实例的审计服务设置或者关闭审计服务。

审计服务设置
审计实例
postgres
日志保存时长
7 天
30 天
3 个月（90天）
6 个月（180天）
1 年（365天）
3 年（1095天）
5 年（1825天）
关闭服务
提交
取消

审计性能说明

最近更新时间：2024-04-09 10:28:59

测试工具

Sysbench 是一款开源的、模块化的、跨平台的多线程基准测试工具，针对在线事务处理（OLTP）场景，Sysbench 可以评估测试数据库在运行高负载时相关核心参数的性能。
SysBench 标准 OLTP 读写混合场景中一个事务包含18个读写 SQL。
SysBench 标准 OLTP 只读场景中一个事务包含14个读 SQL（10条主键点查询、4条范围查询）。
SysBench 标准 OLTP 只写场景中一个事务包含4个写 SQL（2条 UPDATE、1条 DETELE、1条 INSERT）。
本压测使用 SysBench1.1.0版本。更多信息，请参见 Sysbench 官方文档。

测试环境

本文介绍云数据库 PostgreSQL 性能测试所使用的环境。
地域/可用区：北京 - 北京七区
客户端：云服务器标准型 S6（16核32GB，SSD 云硬盘）5Mbps
客户端操作系统：TencentOS Server2.6（Final） 64位
网络：云服务器 CVM 和云数据库 PostgreSQL 实例网络类型均为私有网络（VPC）且在同一子网下
测试的云数据库 PostgreSQL 实例信息如下：
存储类型：通用型 - 本地高性能 SSD 盘，8核32GB
实例架构：双机高可用（一主一从）- 读写实例
实例版本：V14.2
主备复制方式：异步复制

测试指标

本文为您介绍云数据库 PostgreSQL 性能测试的测试指标。

指标	定义
QPS（Queries Per Second）	数据库每秒执行的 SQL 数，包含 INSERT、SELECT、UPDATE、DELETE、COMMIT 等。
并发	性能测试时客户端发起的并发数。

参数说明

pgsql-host：云数据库 PostgreSQL 实例的内网地址
pgsql-port：云数据库 PostgreSQL 实例的端口号
pgsql-user：云数据库 PostgreSQL 实例的用户名
pgsql-password：上述用户名对应的密码
pgsql-db：数据库名
table_size：单表数据量
tables：表总数
threads：并发线程数
time：运行时间

测试方法

说明：
请将以下命令中的 XXX 替换为云数据库 PostgreSQL 测试实例的内网地址、端口号、用户名、用户密码、数据库名，以及对应测试场景的单表数据量和表总数。
1. 准备数据。
`sysbench /usr/local/share/sysbench/oltp_write_only.lua --db-driver=pgsql --pgsql-host=XXX --pgsql-port=XXX --pgsql-user=XXX --pgsql-password=XXX --pgsql-db=XXX --table-size=XXX --tables=XXX prepare`
2. 运行命令，进行压测。
`sysbench /usr/local/shareping`
`sysbench/oltp_read_write.lua --db-driver=pgsql --pgsql-host=XXX --pgsql-port=XXX --pgsql-user=XXX --pgsql-password=XXX --pgsql-db=XXX --table-size=XXX --tables=XXX --threads=XXX --time=XXX --report-interval=1 run`
3. 清理数据。
`sysbench /usr/local/share/sysbench/oltp_write_only.lua --db-driver=pgsql --pgsql-host=XXX --pgsql-port=XXX --pgsql-user=XXX --pgsql-password=XXX --pgsql-db=XXX --table-size=XXX --tables=XXX cleanup`

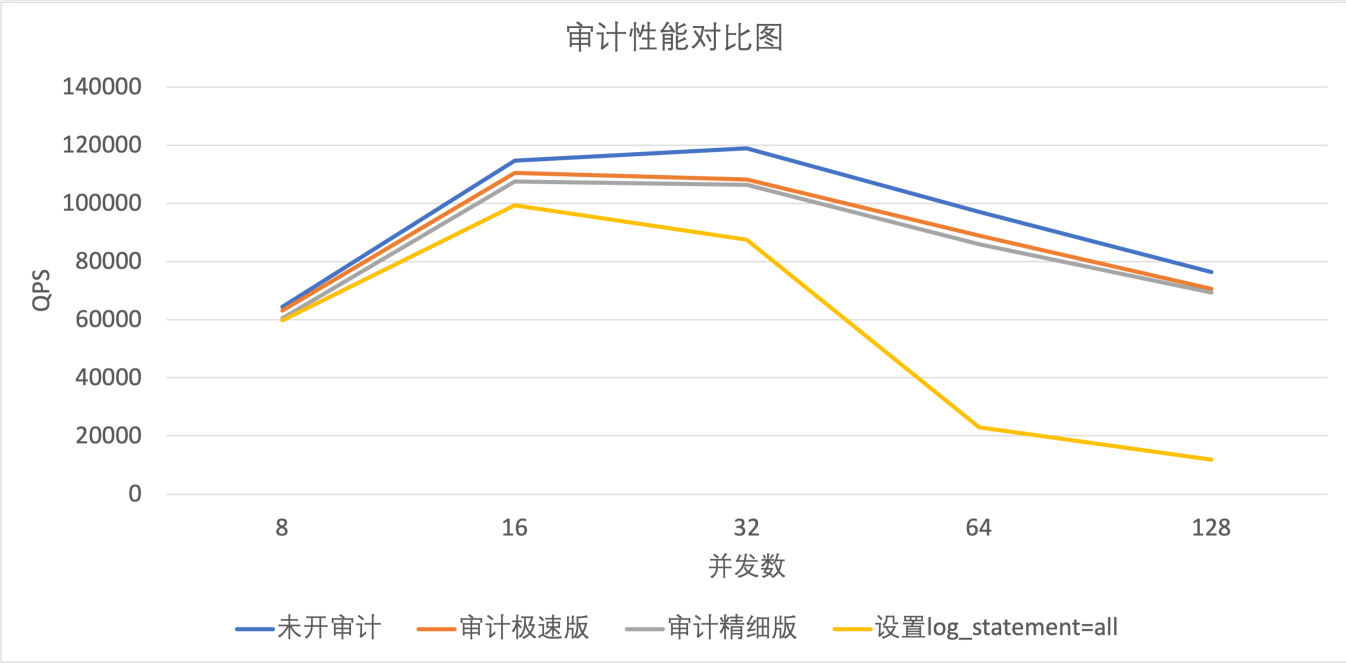
测试结果

下表为单表数据量（table_size)为25000，表总数（tables）为64，测试条件下各个场景的 QPS 性能对比。

并发	未开审计	审计极速版	审计精细版	设置 log_statement=all

8	64435.35	63150.59	60371.58	59686.77
16	114649.43	110425.06	107427.92	99278.01
32	118850.71	108233.97	106368.01	87540.78
64	97012.02	88828.3	85892.76	23017.53
128	76381.21	70592.04	69241.45	11799.92

对应下图所示。



插件管理
插件概述

最近更新时间：2024-01-22 16:22:57

本文为您介绍云数据库 PostgreSQL 支持的插件相关信息。

插件概述

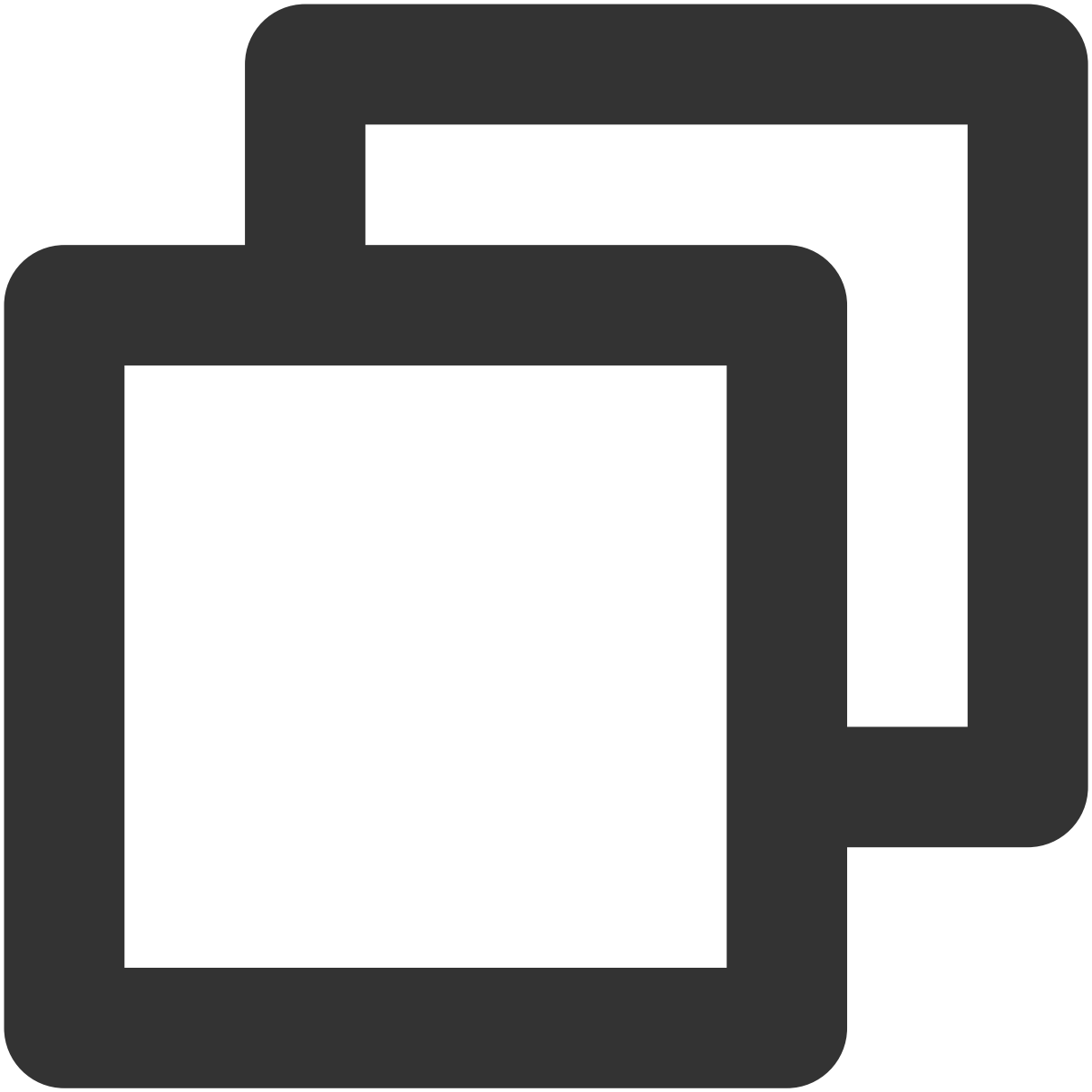
云数据库 PostgreSQL 支持多款开源和自研插件，通过插件能够提升实例运维易用性、查询和写入性能、分词查询、数据检索、数据增量迁移等各方面的能力。

使用插件

云数据库 PostgreSQL 已支持大部分常用插件，可直接使用。而部分插件开启需要指定版本或特殊权限支持，请 提交工单 联系工作人员说明实例 ID 和插件名称开启。

创建插件

创建插件时，当前 pg_tencentdb_superuser 会临时提权为 superuser，通过所有的权限检查。
PostgreSQL 的插件是在 database 之下进行管理的，不同 database 之间可创建不同的插件，并且无法互相使用。
如需创建插件，请通过客户端工具访问至数据库中，执行如下语句创建指定插件：



```
CREATE EXTENSION [ IF NOT EXISTS ] extension_name
[ WITH ]
[ SCHEMA schema_name ]
[ VERSION version ]
[ FROM old_version ]
```

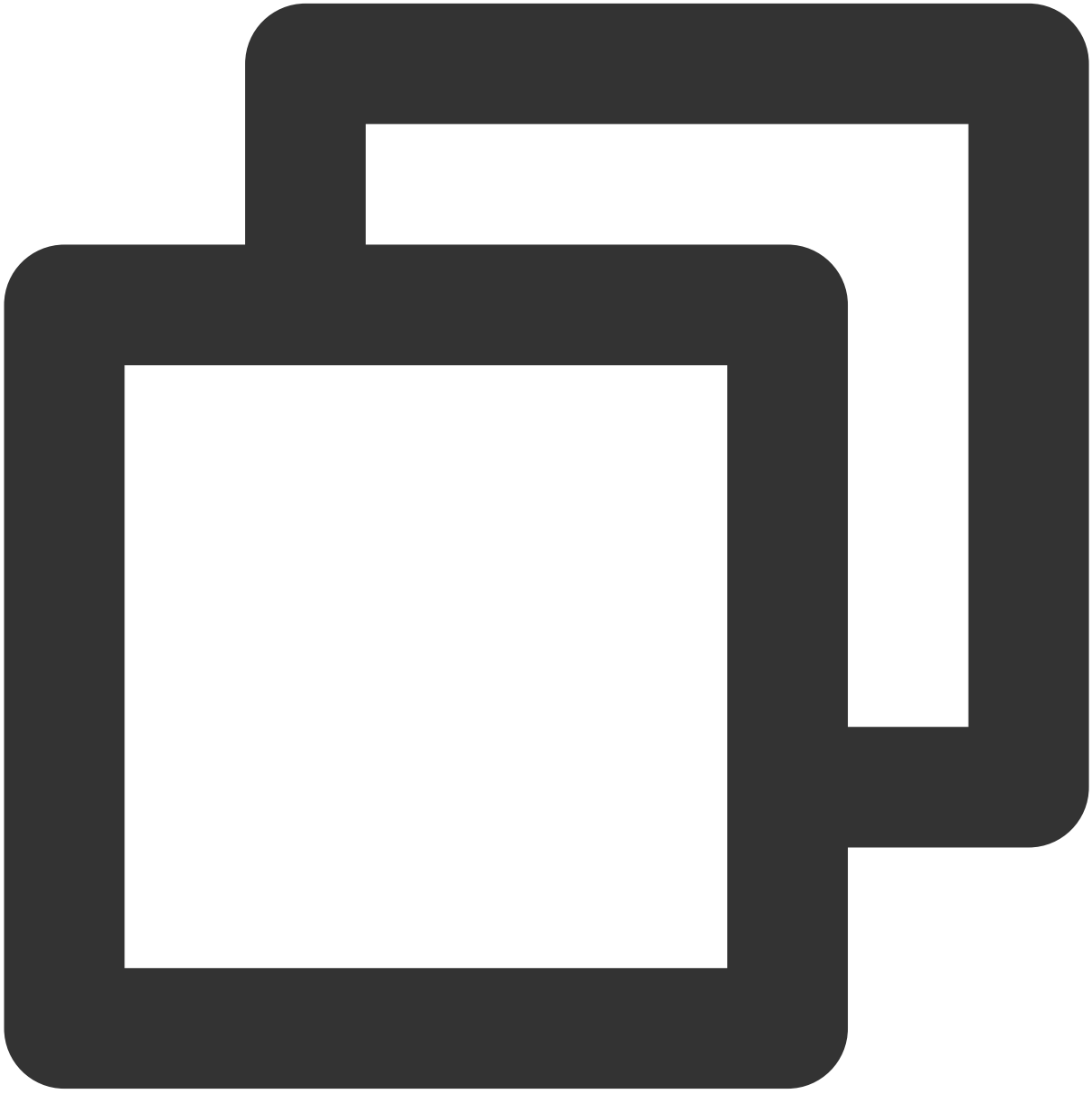
查看已创建的插件

当已经安装了部分插件，需要查看当前 database 安装的插件列表，可通过以下命令查看：
如使用 psql 客户端，可执行 \\dx 命令。

```
\\dx
                                       List of installed extensions
     Name       | Version |   Schema   |                              Description
----------------+---------+------------+------------------------------------------
 amcheck        | 1.2     | public     | functions for verifying relation integrity
 bloom          | 1.0     | public     | bloom access method - signature file based
 hstore         | 1.6     | public     | data type for storing sets of (key, value)
 hstore_plperl  | 1.0     | public     | transform between hstore and plperl
```

```
 jsonb_plperl  | 1.0     | public     | transform between jsonb and plperl
 plperl        | 1.0     | pg_catalog | PL/Perl procedural language
 plpgsql       | 1.0     | pg_catalog | PL/pgSQL procedural language
 postgis       | 3.0.2   | public     | PostGIS geometry, geography, and raster spa
(8 rows)
```

如需使用 SQL 查看，则可执行 `select * from pg_available_extensions where installed_version is not null;` 语句查看已安装的插件列表。



```
        name       | default_version | installed_version |
----------------+-----------------+-------------------+-----------------------------
```

```
 plperl        | 1.0            | 1.0              | PL/Perl procedural language
 amcheck       | 1.2            | 1.2              | functions for verifying rela
 hstore_plperl | 1.0            | 1.0              | transform between hstore and
 plpgsql       | 1.0            | 1.0              | PL/pgSQL procedural language
 jsonb_plperl  | 1.0            | 1.0              | transform between jsonb and
 hstore        | 1.6            | 1.6              | data type for storing sets o
 bloom         | 1.0            | 1.0              | bloom access method - signat
 postgis       | 3.0.2          | 3.0.2            | PostGIS geometry, geography,
(8 rows)
```

支持插件列表

云数据库 PostgreSQL 已支持多款功能强大、性能优秀的插件，各数据库版本详细支持插件列表请参见 支持插件列表。

支持插件

支持插件版本概览

最近更新时间：2024-01-22 16:22:57

本文为您介绍云数据库 PostgreSQL 各个内核版本支持的插件列表。

云数据库 PostgreSQL 已经支持自主创建插件（CREATEEXTENSION<插件名>），具体您可以单击以下对应链接了解各版本支持插件的情况：

云数据库 PostgreSQL 9.3

云数据库 PostgreSQL 9.5

云数据库 PostgreSQL 10

云数据库 PostgreSQL 11

云数据库 PostgreSQL 12

云数据库 PostgreSQL 13

云数据库 PostgreSQL 14

说明：

timescaledb、pipelinedb、wal2json、decoder_raw、decoderbufs 插件无法直接创建使用，如需使用或者您对插件有其他需求或建议，请 提交工单。

PostgreSQL 9.3 支持插件

最近更新时间：2024-01-22 16:22:57

本文为您介绍云数据库 PostgreSQL 9.3 各内核版本支持的插件情况。

插件名	v9.3.25_r1.3	v9.3.25_r1.2	v9.3.25_r1.1	v9.3.5_r1.0
pg_hint_plan	1.1.4	1.1.4	1.1.4	1.1.4
pg_prewarm	不支持	不支持	不支持	不支持
pg_stat_error	1	1	1	1
pg_stat_log	1	1	1	1
pg_stat_statements	1.1	1.1	1.1	1.1
pgrowlocks	1.1	1.1	1.1	1.1
sslinfo	1	1	1	1
tablefunc	1	1	1	1
tcn	1	1	1	1
unaccent	1	1	1	1
uuid-ossp	1	1	1	1
pg_cron	不支持	不支持	不支持	不支持
pgagent	4	4	4	4
pg_partman	不支持	不支持	不支持	不支持
tsearch2	1	1	1	1
postgis	2.3.0	2.3.0	2.3.0	2.3.0
postgis_raster	不支持	不支持	不支持	不支持
postgis_sfcgal	2.3.0	2.3.0	2.3.0	不支持
postgis_tiger_geocoder	2.3.0	2.3.0	2.3.0	2.3.0
postgis_topology	2.3.0	2.3.0	2.3.0	2.3.0
pgrouting	2.4.1	2.4.1	2.4.1	2.4.1

版权所有：腾讯云计算（北京）有限责任公司

address_standardizer	2.3.0	2.3.0	2.3.0	2.3.0
address_standardizer_data_us	2.3.0	2.3.0	2.3.0	2.3.0
earthdistance	1	1	1	1
plperl	1	1	1	1
plpgsql	1	1	1	1
pltcl	1	1	1	1
plv8	2.0.0	2.0.0	2.0.0	2.0.0
bool_plperl	不支持	不支持	不支持	不支持
jsonb_plperl	不支持	不支持	不支持	不支持
hstore	1.2	1.2	1.2	1.2
hstore_plperl	不支持	不支持	不支持	不支持
plcoffee	2.0.0	2.0.0	2.0.0	2.0.0
plls	2.0.0	2.0.0	2.0.0	2.0.0
timescaledb	不支持	不支持	不支持	不支持
pipelinedb	不支持	不支持	不支持	不支持
rdkit	不支持	不支持	不支持	不支持
imgsmlr	1	1	1	1
zhparser	1	1	1	1
intagg	1	1	1	1
intarray	1	1	1	1
isn	1	1	1	1
xml2	1	1	1	1
jsonbx	不支持	不支持	不支持	不支持
dict_int	1	1	1	1
dict_xsyn	1	1	1	1

版权所有：腾讯云计算（北京）有限责任公司

citext	1	1	1	1
ltree	1	1	1	1
postgres_fdw	1	1	1	1
orafce	3.3	3.3	3.3	3.3
chkpass	1	1	1	1
bloom	不支持	不支持	不支持	不支持
btree_gin	1	1	1	1
btree_gist	1	1	1	1
roaringbitmap	不支持	不支持	不支持	不支持
rum	不支持	不支持	不支持	不支持
cube	1	1	1	1
decoderbufs	不支持	不支持	不支持	不支持
pg_bigm	不支持	不支持	不支持	不支持
fuzzystrmatch	1	1	1	1
hll	不支持	不支持	不支持	不支持
pg_trgm	1.1	1.1	1.1	1.1
pg_hashids	1.2.1	1.2.1	1.2.1	1.2.1
pgcrypto	1	1	1	1
cos_fdw	不支持	不支持	不支持	不支持
topn	不支持	不支持	不支持	不支持

版权所有：腾讯云计算（北京）有限责任公司

PostgreSQL 9.5 支持插件

最近更新时间：2024-01-22 16:22:57

本文为您介绍云数据库 PostgreSQL 9.5 各内核版本支持的插件情况。

插件名	v9.5.25_r1.3	v9.5.25_r1.2	v9.5.25_r1.1	v9.5.4_r1.0
pg_hint_plan	1.1.5	1.1.5	1.1.5	1.1.5
pg_prewarm	1	1	1	不支持
pg_stat_error	1	1	1	1
pg_stat_log	1	1	1	1
pg_stat_statements	1.3	1.3	1.3	1.3
pgrowlocks	1.1	1.1	1.1	1.1
sslinfo	1	1	1	1
tablefunc	1	1	1	1
tcn	1	1	1	1
unaccent	1	1	1	1
uuid-ossp	1	1	1	1
pg_cron	1.1	1.1	1.1	1.1
pgagent	1.2	1.2	1.2	4
pg_partman	2.6.4、1.4	2.6.4、1.4	2.6.4、1.4	2.6.4、1.4、1.0
tsearch2	1	1	1	1
postgis	2.3.0	2.3.0	2.3.0	2.3.0
postgis_raster	不支持	不支持	不支持	不支持
postgis_sfcgal	2.3.0	2.3.0	2.3.0	不支持
postgis_tiger_geocoder	2.3.0	2.3.0	2.3.0	2.3.0
postgis_topology	2.3.0	2.3.0	2.3.0	2.3.0
pgrouting	2.4.1	2.4.1	2.4.1	2.4.1

版权所有：腾讯云计算（北京）有限责任公司

address_standardizer	2.3.0	2.3.0	2.3.0	2.3.0
address_standardizer_data_us	2.3.0	2.3.0	2.3.0	2.3.0
earthdistance	1	1	1	1
plperl	1	1	1	1
plpgsql	1	1	1	1
pltcl	1	1	1	1
plv8	2.0.0	2.0.0	2.0.0	2.0.0
bool_plperl	不支持	不支持	不支持	不支持
jsonb_plperl	不支持	不支持	不支持	不支持
hstore	1.3	1.3	1.3	1.3
hstore_plperl	1	1	1	1
plcoffee	2.0.0	2.0.0	2.0.0	2.0.0
plls	2.0.0	2.0.0	2.0.0	2.0.0
timescaledb	不支持	不支持	不支持	不支持
pipelinedb	不支持	不支持	不支持	不支持
rdkit	不支持	不支持	不支持	不支持
imgsmlr	1	1	1	1
zhparser	1	1	1	1
intagg	1	1	1	1
intarray	1	1	1	1
isn	1	1	1	1
xml2	1	1	1	1
jsonbx	1	1	1	1
dict_int	1	1	1	1
dict_xsyn	1	1	1	1

版权所有：腾讯云计算（北京）有限责任公司

citext	1.1	1.1	1.1	1.1
ltree	1	1	1	1
postgres_fdw	1	1	1	1
orafce	3.3	3.3	3.3	3.3
chkpass	1	1	1	1
bloom	不支持	不支持	不支持	不支持
btree_gin	1	1	1	1
btree_gist	1.1	1.1	1.1	1.1
roaringbitmap	不支持	不支持	不支持	不支持
rum	不支持	不支持	不支持	不支持
cube	1	1	1	1
decoderbufs	不支持	不支持	不支持	不支持
pg_bigm	1.2	1.2	1.2	1.2
fuzzystrmatch	1	1	1	1
hll	2.14	2.14	2.14	2.14
pg_trgm	1.1	1.1	1.1	1.1
pg_hashids	1.2.1	1.2.1	1.2.1	1.2.1
pgcrypto	1.2	1.2	1.2	1.2
cos_fdw	不支持	不支持	不支持	不支持
topn	不支持	不支持	不支持	不支持

版权所有：腾讯云计算（北京）有限责任公司

PostgreSQL 10 支持插件

最近更新时间：2024-01-22 16:22:57

本文为您介绍云数据库 PostgreSQL 10 各内核版本支持的插件情况。

插件名	v10.17_r1.4	v10.17_r1.3	v10.17_r1.2	v10.17_r1.1	v10.4_r1.1
pg_hint_plan	1.3.6	1.3.6	1.3.6	1.3.6	1.3.6
pg_prewarm	1.1	1.1	1.1	1.1	1.1
pg_stat_error	1	1	1	1	1
pg_stat_log	1	1	1	1	1
pg_stat_statements	1.6	1.6	1.6	1.6	1.5
pgrowlocks	1.2	1.2	1.2	1.2	1.2
sslinfo	1.2	1.2	1.2	1.2	1.2
tablefunc	1	1	1	1	1
tcn	1	1	1	1	1
unaccent	1.1	1.1	1.1	1.1	1.1
uuid-ossp	1.1	1.1	1.1	1.1	1.1
pg_cron	1.4	1.4	1.1	1.1	1.1
pgagent	4	4	4	4	4
pg_partman	不支持	不支持	不支持	不支持	不支持
tsearch2	不支持	不支持	不支持	不支持	不支持
postgis	3.0.1	3.0.1	3.0.1	3.0.1	2.3.7
postgis_raster	3.0.1	3.0.1	3.0.1	3.0.1	不支持
postgis_sfcgal	3.0.1	3.0.1	3.0.1	3.0.1	不支持
postgis_tiger_geocoder	3.0.1	3.0.1	3.0.1	3.0.1	2.4.1
postgis_topology	3.0.1	3.0.1	3.0.1	3.0.1	2.4.1
pgrouting	2.6.0	2.6.0	2.6.0	2.6.0	2.6.0

版权所有：腾讯云计算（北京）有限责任公司

address_standardizer	3.0.1	3.0.1	3.0.1	3.0.1	2.4.1
address_standardizer_data_us	3.0.1	3.0.1	3.0.1	3.0.1	2.4.1
earthdistance	1.1	1.1	1.1	1.1	1.1
plperl	1	1	1	1	1
plpgsql	1	1	1	1	1
pltcl	1	1	1	1	1
plv8	2.3.4	2.3.4	2.3.4	2.3.4	2.3.4
bool_plperl	不支持	不支持	不支持	不支持	不支持
jsonb_plperl	不支持	不支持	不支持	不支持	不支持
hstore	1.4	1.4	1.4	1.4	1.4
hstore_plperl	1	1	1	1	1
plcoffee	2.3.4	2.3.4	2.3.4	2.3.4	2.3.4
plls	2.3.4	2.3.4	2.3.4	2.3.4	2.3.4
timescaledb	1.7.5	1.7.5	1.7.5	1.7.5	1.7.5
pipelinedb	1.0.0	1.0.0	1.0.0	1.0.0	1.0.0
rdkit	3.8	3.8	3.8	3.8	不支持
imgsmlr	1	1	1	1	1
zhparser	1	1	1	1	1
intagg	1.1	1.1	1.1	1.1	1.1
intarray	1.2	1.2	1.2	1.2	1.2
isn	1.1	1.1	1.1	1.1	1.1
xml2	1.1	1.1	1.1	1.1	1.1
jsonbx	不支持	不支持	不支持	不支持	不支持
dict_int	1	1	1	1	1
dict_xsyn	1	1	1	1	1

版权所有：腾讯云计算（北京）有限责任公司

citext	1.4	1.4	1.4	1.4	1.4
ltree	1.1	1.1	1.1	1.1	1.1
postgres_fdw	1	1	1	1	1
orafce	不支持	不支持	不支持	不支持	不支持
chkpass	1	1	1	1	1
bloom	1	1	1	1	1
btree_gin	1.2	1.2	1.2	1.2	1.2
btree_gist	1.5	1.5	1.5	1.5	1.5
roaringbitmap	0.5	0.5	0.5	0.5	不支持
rum	1.3	1.3	1.3	1.3	1.3
cube	1.2	1.2	1.2	1.2	1.2
decoderbufs	0.1.0	0.1.0	0.1.0	0.1.0	0.1.0
pg_bigm	1.2	1.2	1.2	1.2	1.2
fuzzystrmatch	1.1	1.1	1.1	1.1	1.1
hll	2.14	2.14	2.14	2.14	2.14
pg_trgm	1.3	1.3	1.3	1.3	1.3
pg_hashids	1.2.1	1.2.1	1.2.1	1.2.1	1.2.1
pgcrypto	1.3	1.3	1.3	1.3	1.3
cos_fdw	1	不支持	不支持	不支持	不支持
topn	2.4.0	2.4.0	不支持	不支持	不支持

版权所有：腾讯云计算（北京）有限责任公司

PostgreSQL 11 支持插件

最近更新时间：2024-01-22 16:22:57

本文为您介绍云数据库 PostgreSQL 11 各内核版本支持的插件情况。

插件名	v11.12_r1.4	v11.12_r1.3	v11.12_r1.2	v11.12_r1.1	v11.8_r1.0
pg_hint_plan	1.3.6	1.3.6	1.3.6	1.3.6	1.3.6
pg_prewarm	1.2	1.2	1.2	1.2	1.2
pg_stat_error	1	1	1	1	1
pg_stat_log	1	1	1	1	1
pg_stat_statements	1.6	1.6	1.6	1.6	1.6
pgrowlocks	1.2	1.2	1.2	1.2	1.2
sslinfo	1.2	1.2	1.2	1.2	1.2
tablefunc	1	1	1	1	1
tcn	1	1	1	1	1
unaccent	1.1	1.1	1.1	1.1	1.1
uuid-ossp	1.1	1.1	1.1	1.1	1.1
pg_cron	1.4	1.4	不支持	不支持	不支持
pgagent	4	4	4	4	4
pg_partman	不支持	不支持	不支持	不支持	不支持
tsearch2	不支持	不支持	不支持	不支持	不支持
postgis	3.0.1	3.0.1	3.0.1	3.0.1	3.0.1
postgis_raster	3.0.1	3.0.1	3.0.1	3.0.1	3.0.1
postgis_sfcgal	3.0.1	3.0.1	3.0.1	3.0.1	不支持
postgis_tiger_geocoder	3.0.1	3.0.1	3.0.1	3.0.1	3.0.1
postgis_topology	3.0.1	3.0.1	3.0.1	3.0.1	3.0.1
pgrouting	2.6.0	2.6.0	2.6.0	2.6.0	2.6.0

版权所有：腾讯云计算（北京）有限责任公司

address_standardizer	3.0.1	3.0.1	3.0.1	3.0.1	3.0.1
address_standardizer_data_us	3.0.1	3.0.1	3.0.1	3.0.1	3.0.1
earthdistance	1.1	1.1	1.1	1.1	1.1
plperl	1	1	1	1	1
plpgsql	1	1	1	1	1
pltcl	1	1	1	1	1
plv8	2.3.15	2.3.15	2.3.15	2.3.15	2.3.15
bool_plperl	不支持	不支持	不支持	不支持	不支持
jsonb_plperl	1	1	1	1	1
hstore	1.5	1.5	1.5	1.5	1.5
hstore_plperl	1	1	1	1	1
plcoffee	2.3.15	2.3.15	2.3.15	2.3.15	2.3.15
plls	2.3.15	2.3.15	2.3.15	2.3.15	2.3.15
timescaledb	1.7.5	1.7.5	1.7.5	1.7.5	1.7.5
pipelinedb	1.0.0	1.0.0	1.0.0	1.0.0	1.0.0
rdkit	3.8	3.8	3.8	3.8	不支持
imgsmlr	1	1	1	1	1
zhparser	1	1	1	1	1
intagg	1.1	1.1	1.1	1.1	1.1
intarray	1.2	1.2	1.2	1.2	1.2
isn	1.2	1.2	1.2	1.2	1.2
xml2	1.1	1.1	1.1	1.1	1.1
jsonbx	不支持	不支持	不支持	不支持	不支持
dict_int	1	1	1	1	1
dict_xsyn	1	1	1	1	1

版权所有：腾讯云计算（北京）有限责任公司

citext	1.5	1.5	1.5	1.5	1.5
ltree	1.1	1.1	1.1	1.1	1.1
postgres_fdw	1	1	1	1	1
orafce	不支持	不支持	不支持	不支持	不支持
chkpass	1	1	1	1	1
bloom	1	1	1	1	1
btree_gin	1.3	1.3	1.3	1.3	1.3
btree_gist	1.5	1.5	1.5	1.5	1.5
roaringbitmap	0.5	0.5	0.5	0.5	不支持
rum	1.3	1.3	1.3	1.3	1.3
cube	1.4	1.4	1.4	1.4	1.4
decoderbufs	0.1.0	0.1.0	0.1.0	0.1.0	0.1.0
pg_bigm	1.2	1.2	1.2	1.2	1.2
fuzzystrmatch	1.1	1.1	1.1	1.1	1.1
hll	2.14	2.14	2.14	2.14	2.14
pg_trgm	1.4	1.4	1.4	1.4	1.4
pg_hashids	1.2.1	1.2.1	1.2.1	1.2.1	1.2.1
pgcrypto	1.3	1.3	1.3	1.3	1.3
cos_fdw	1	不支持	不支持	不支持	不支持
topn	2.4.0	2.4.0	不支持	不支持	不支持

版权所有：腾讯云计算（北京）有限责任公司

PostgreSQL 12 支持插件

最近更新时间：2024-01-22 16:22:57

本文为您介绍云数据库 PostgreSQL 12 各内核版本支持的插件情况。

插件名	v12.7_r1.4	v12.7_r1.3	v12.7_r1.2	v12.7_r1.1	v12.4_r1.0
pg_hint_plan	1.3.6	1.3.6	1.3.6	1.3.6	1.3.6
pg_prewarm	1.2	1.2	1.2	1.2	1.2
pg_stat_error	1	1	1	1	1
pg_stat_log	1	1	1	1	1
pg_stat_statements	1.7	1.7	1.7	1.7	1.7
pgrowlocks	1.2	1.2	1.2	1.2	1.2
sslinfo	1.2	1.2	1.2	1.2	1.2
tablefunc	1	1	1	1	1
tcn	1	1	1	1	1
unaccent	1.1	1.1	1.1	1.1	1.1
uuid-ossp	1.1	1.1	1.1	1.1	1.1
pg_cron	1.4	1.4	不支持	不支持	不支持
pgagent	4	4	4	4	4
pg_partman	不支持	不支持	不支持	不支持	不支持
tsearch2	不支持	不支持	不支持	不支持	不支持
postgis	3.0.2	3.0.2	3.0.2	3.0.2	3.0.2
postgis_raster	3.0.2	3.0.2	3.0.2	3.0.2	3.0.2
postgis_sfcgal	3.0.2	3.0.2	3.0.2	3.0.2	3.0.2
postgis_tiger_geocoder	3.0.2	3.0.2	3.0.2	3.0.2	3.0.2
postgis_topology	3.0.2	3.0.2	3.0.2	3.0.2	3.0.2
pgrouting	3.1.0	3.1.0	3.1.0	3.1.0	3.1.0

版权所有：腾讯云计算（北京）有限责任公司

address_standardizer	3.0.2	3.0.2	3.0.2	3.0.2	3.0.2
address_standardizer_data_us	3.0.2	3.0.2	3.0.2	3.0.2	3.0.2
earthdistance	1.1	1.1	1.1	1.1	1.1
plperl	1	1	1	1	1
plpgsql	1	1	1	1	1
pltcl	1	1	1	1	1
plv8	2.3.15	2.3.15	2.3.15	2.3.15	2.3.15
bool_plperl	不支持	不支持	不支持	不支持	不支持
jsonb_plperl	1	1	1	1	1
hstore	1.6	1.6	1.6	1.6	1.6
hstore_plperl	1	1	1	1	1
plcoffee	2.3.15	2.3.15	2.3.15	2.3.15	2.3.15
plls	2.3.15	2.3.15	2.3.15	2.3.15	2.3.15
timescaledb	1.7.4	1.7.4	1.7.4	1.7.4	1.7.4
pipelinedb	不支持	不支持	不支持	不支持	不支持
rdkit	3.8	3.8	3.8	3.8	3.8
imgsmlr	1	1	1	1	1
zhparser	1	1	1	1	1
intagg	1.1	1.1	1.1	1.1	1.1
intarray	1.2	1.2	1.2	1.2	1.2
isn	1.2	1.2	1.2	1.2	1.2
xml2	1.1	1.1	1.1	1.1	1.1
jsonbx	不支持	不支持	不支持	不支持	不支持
dict_int	1	1	1	1	1
dict_xsyn	1	1	1	1	1

版权所有：腾讯云计算（北京）有限责任公司

citext	1.6	1.6	1.6	1.6	1.6
ltree	1.1	1.1	1.1	1.1	1.1
postgres_fdw	1	1	1	1	1
orafce	不支持	不支持	不支持	不支持	不支持
chkpass	1	1	1	1	1
bloom	1	1	1	1	1
btree_gin	1.3	1.3	1.3	1.3	1.3
btree_gist	1.5	1.5	1.5	1.5	1.5
roaringbitmap	0.5	0.5	0.5	0.5	不支持
rum	1.3	1.3	1.3	1.3	1.3
cube	1.4	1.4	1.4	1.4	1.4
decoderbufs	0.1.0	0.1.0	0.1.0	0.1.0	0.1.0
pg_bigm	1.2	1.2	1.2	1.2	1.2
fuzzystrmatch	1.1	1.1	1.1	1.1	1.1
hll	2.14	2.14	2.14	2.14	2.14
pg_trgm	1.4	1.4	1.4	1.4	1.4
pg_hashids	1.2.1	1.2.1	1.2.1	1.2.1	1.2.1
pgcrypto	1.3	1.3	1.3	1.3	1.3
cos_fdw	1	不支持	不支持	不支持	不支持
topn	2.4.0	2.4.0	不支持	不支持	不支持

版权所有：腾讯云计算（北京）有限责任公司

PostgreSQL 13 支持插件

最近更新时间：2024-01-22 16:22:57

本文为您介绍云数据库 PostgreSQL 13 各内核版本支持的插件情况。

插件名	v13.3_r1.3	v13.3_r1.2	v13.3_r1.1	v13.3_r1.0
pg_hint_plan	1.3.7	1.3.7	1.3.7	1.3.7
pg_prewarm	1.2	1.2	1.2	1.2
pg_stat_error	1	1	1	1
pg_stat_log	1	1	1	1
pg_stat_statements	1.8	1.8	1.8	1.8
pgrowlocks	1.2	1.2	1.2	1.2
sslinfo	1.2	1.2	1.2	1.2
tablefunc	1	1	1	1
tcn	1	1	1	1
unaccent	1.1	1.1	1.1	1.1
uuid-ossp	1.1	1.1	1.1	1.1
pg_cron	1.4	1.4	不支持	不支持
pgagent	4	4	4	4
pg_partman	不支持	不支持	不支持	不支持
tsearch2	不支持	不支持	不支持	不支持
postgis	3.0.2	3.0.2	3.0.2	3.0.2
postgis_raster	3.0.2	3.0.2	3.0.2	3.0.2
postgis_sfcgal	3.0.2	3.0.2	3.0.2	3.0.2
postgis_tiger_geocoder	3.0.2	3.0.2	3.0.2	3.0.2
postgis_topology	3.0.2	3.0.2	3.0.2	3.0.2
pgrouting	3.1.0	3.1.0	3.1.0	3.1.0

版权所有：腾讯云计算（北京）有限责任公司

address_standardizer	3.0.2	3.0.2	3.0.2	3.0.2
address_standardizer_data_us	3.0.2	3.0.2	3.0.2	3.0.2
earthdistance	1.1	1.1	1.1	1.1
plperl	1	1	1	1
plpgsql	1	1	1	1
pltcl	1	1	1	1
plv8	2.3.15	2.3.15	2.3.15	2.3.15
bool_plperl	1	1	1	1
jsonb_plperl	1	1	1	1
hstore	1.7	1.7	1.7	1.7
hstore_plperl	1	1	1	1
plcoffee	2.3.15	2.3.15	2.3.15	2.3.15
plls	2.3.15	2.3.15	2.3.15	2.3.15
timescaledb	2.1.1	2.1.1	2.1.1	2.1.1
pipelinedb	不支持	不支持	不支持	不支持
rdkit	3.8	3.8	3.8	3.8
imgsmlr	1	1	1	1
zhparser	1	1	1	1
intagg	1.1	1.1	1.1	1.1
intarray	1.3	1.3	1.3	1.3
isn	1.2	1.2	1.2	1.2
xml2	1.1	1.1	1.1	1.1
jsonbx	不支持	不支持	不支持	不支持
dict_int	1	1	1	1
dict_xsyn	1	1	1	1

版权所有：腾讯云计算（北京）有限责任公司

citext	1.6	1.6	1.6	1.6
ltree	1.2	1.2	1.2	1.2
postgres_fdw	1	1	1	1
orafce	不支持	不支持	不支持	不支持
chkpass	1	1	1	1
bloom	1	1	1	1
btree_gin	1.3	1.3	1.3	1.3
btree_gist	1.5	1.5	1.5	1.5
roaringbitmap	0.5	0.5	0.5	0.5
rum	1.3	1.3	1.3	1.3
cube	1.4	1.4	1.4	1.4
decoderbufs	0.1.0	0.1.0	0.1.0	0.1.0
pg_bigm	1.2	1.2	1.2	1.2
fuzzystrmatch	1.1	1.1	1.1	1.1
hll	2.15	2.15	2.15	2.15
pg_trgm	1.5	1.5	1.5	1.5
pg_hashids	1.2.1	1.2.1	1.2.1	1.2.1
pgcrypto	1.3	1.3	1.3	1.3
cos_fdw	1	不支持	不支持	不支持
topn	2.4.0	2.4.0	不支持	不支持

版权所有：腾讯云计算（北京）有限责任公司

PostgreSQL 14 支持插件

最近更新时间：2024-01-22 16:22:57

本文为您介绍云数据库 PostgreSQL 14 各内核版本支持的插件情况。

插件名	v14.2_r1.1	v14.2_r1.0
pg_hint_plan	1.4	1.4
pg_prewarm	1.2	1.2
pg_stat_error	不支持	不支持
pg_stat_log	1	1
pg_stat_statements	1.9	1.9
pgrowlocks	1.2	1.2
sslinfo	1.2	1.2
tablefunc	1	1
tcn	1	1
unaccent	1.1	1.1
uuid-ossp	1.1	1.1
pg_cron	1.4	1.4
pgagent	4	4
pg_partman	不支持	不支持
tsearch2	不支持	不支持
postgis	3.2.1	3.2.1
postgis_raster	3.2.1	3.2.1
postgis_sfcgal	3.2.1	3.2.1
postgis_tiger_geocoder	3.2.1	3.2.1
postgis_topology	3.2.1	3.2.1
pgrouting	3.2.2	3.2.2

版权所有：腾讯云计算（北京）有限责任公司

address_standardizer	3.2.1	3.2.1
address_standardizer_data_us	3.2.1	3.2.1
earthdistance	1.1	1.1
plperl	1	1
plpgsql	1	1
pltcl	1	1
plv8	2.3.15	2.3.15
bool_plperl	1	1
jsonb_plperl	1	1
hstore	1.8	1.8
hstore_plperl	1	1
plcoffee	2.3.15	2.3.15
plls	2.3.15	2.3.15
timescaledb	2.6.0	2.6.0
pipelinedb	不支持	不支持
rdkit	4.0.1	4.0.1
imgsmlr	1	1
zhparser	2.2	2.2
intagg	1.1	1.1
intarray	1.5	1.5
isn	1.2	1.2
xml2	1.1	1.1
jsonbx	不支持	不支持
dict_int	1	1
dict_xsyn	1	1

版权所有：腾讯云计算（北京）有限责任公司

citext	1.6	1.6
ltree	1.2	1.2
postgres_fdw	1.1	1.1
orafce	不支持	不支持
chkpass	1	1
bloom	1	1
btree_gin	1.3	1.3
btree_gist	1.6	1.6
roaringbitmap	0.5	0.5
rum	1.3	1.3
cube	1.5	1.5
decoderbufs	0.1.0	0.1.0
pg_bigm	1.2	1.2
fuzzystrmatch	1.1	1.1
hll	2.16	2.16
pg_trgm	1.6	1.6
pg_hashids	1.3	1.3
pgcrypto	1.3	1.3
cos_fdw	1	不支持
topn	2.4.0	2.4.0

版权所有：腾讯云计算（北京）有限责任公司

PostgreSQL 15支持的插件

最近更新时间：2024-03-20 14:48:39

本文为您介绍云数据库 PostgreSQL 15 最新版本支持的插件情况，如您需要，请 升级到最新内核小版本 使用。

插件名	插件版本
address_standardizer	3.3.2
address_standardizer_data_us	3.3.2
autoinc	1.0
bloom	1.0
bool_plperl	1.0
btree_gin	1.3
btree_gist	1.7
chkpass	1.0
citext	1.6
cos_fdw	1.0
cube	1.5
dblink	1.2
decoderbufs	0.1.0
dict_int	1.0
dict_xsyn	1.0
earthdistance	1.1
fuzzystrmatch	1.1
hll	2.16
hstore	1.8
hstore_plperl	1.0
imgsmlr	1.0

版权所有：腾讯云计算（北京）有限责任公司

insert_username	1.0
intagg	1.1
intarray	1.5
isn	1.2
jsonb_plperl	1.0
lo	1.1
ltree	1.2
moddatetime	1.0
mysql_fdw	1.1
old_snapshot	1.0
pg_bigm	1.2
pg_buffercache	1.3
pg_cron	1.4
pg_freespacemap	1.2
pg_hashids	1.3
pg_hint_plan	1.4
pg_prewarm	1.2
pg_squeeze	1.5.2
pg_stat_log	1.0
pg_stat_statements	1.10
pg_surgery	1.0
pg_trgm	1.6
pgagent	4.0
pgcrypto	1.3
pgrouting	3.2.2

版权所有：腾讯云计算（北京）有限责任公司

pgrowlocks	1.2
pgstattuple	1.5
pgvector	0.4.2
plcoffee	2.3.15
plperl	1.0
plpgsql	1.0
pltcl	1.0
postgis	3.3.2
postgis_raster	3.3.2
postgis_sfcgal	3.3.2
postgis_tiger_geocoder	3.3.2
postgis_topology	3.3.2
postgres_fdw	1.1
rdkit	4.0.1
refint	1.0
roaringbitmap	0.5
rum	1.3
seg	1.4
sslinfo	1.2
tablefunc	1.0
tcn	1.0
tencentdb_failover_slot	1.0
tencentdb_system_stat	1.0
topn	2.4.0
tsm_system_rows	1.0

版权所有：腾讯云计算（北京）有限责任公司

tsm_system_time	1.0
unaccent	1.1
uuid-ossp	1.1
wal2json	2.5
xml2	1.1
zhparser	2.2

PostgreSQL 16支持的插件

最近更新时间：2024-03-20 14:50:09

本文为您介绍云数据库 PostgreSQL 16 最新版本支持的插件情况，如您需要，请 升级到最新内核小版本 使用。

插件名	插件版本
address_standardizer	3.4.0
address_standardizer_data_us	3.4.0
autoinc	1.0
bloom	1.0
bool_plperl	1.0
btree_gin	1.3
btree_gist	1.7
chkpass	1.0
citext	1.6
cos_fdw	1.0
cube	1.5
dblink	1.2
decoderbufs	0.1.0
dict_int	1.0
dict_xsyn	1.0
earthdistance	1.1
fuzzystrmatch	1.1
hll	2.18
hstore	1.8
hstore_plperl	1.0
imgsmlr	1.0

版权所有：腾讯云计算（北京）有限责任公司

insert_username	1.0
intagg	1.1
intarray	1.5
isn	1.2
jsonb_plperl	1.0
lo	1.1
ltree	1.2
moddatetime	1.0
mysql_fdw	1.1
old_snapshot	1.0
pg_bigm	1.2
pg_buffercache	1.4
pg_cron	1.6
pg_freespacemap	1.2
pg_hashids	1.3
pg_hint_plan	1.6.0
pg_prewarm	1.2
pg_squeeze	1.6
pg_stat_log	1.0
pg_stat_statements	1.10
pg_surgery	1.0
pg_similarity	1.0
pg_trgm	1.6
pg_walinspect	1.0
pgagent	4.0

版权所有：腾讯云计算（北京）有限责任公司

pgcrypto	1.3
pgrouting	3.2.2
pgrowlocks	1.2
pgstattuple	1.5
pgvector	0.5.0
plcoffee	2.3.15
plperl	1.0
plpgsql	1.0
pltcl	1.0
postgis	3.4.0
postgis_raster	3.4.0
postgis_sfcgal	3.4.0
postgis_tiger_geocoder	3.4.0
postgis_topology	3.4.0
postgres_fdw	1.1
rdkit	4.0.1
refint	1.0
roaringbitmap	0.5
rum	1.3
seg	1.4
sslinfo	1.2
tablefunc	1.0
tcn	1.0
tencentdb_failover_slot	1.0
tencentdb_system_stat	1.0

版权所有：腾讯云计算（北京）有限责任公司

topn	2.4.0
tsm_system_rows	1.0
tsm_system_time	1.0
unaccent	1.1
uuid-ossp	1.1
wal2json	2.5
xml2	1.1
zhparser	2.2

版权所有：腾讯云计算（北京）有限责任公司

pgAgent 插件

最近更新时间：2024-03-21 11:30:27

本文介绍如何通过 pgAgent 功能来实现云数据库 PostgreSQL 的自动任务执行。推荐使用 pg_cron 插件实现定时的功能。

简介

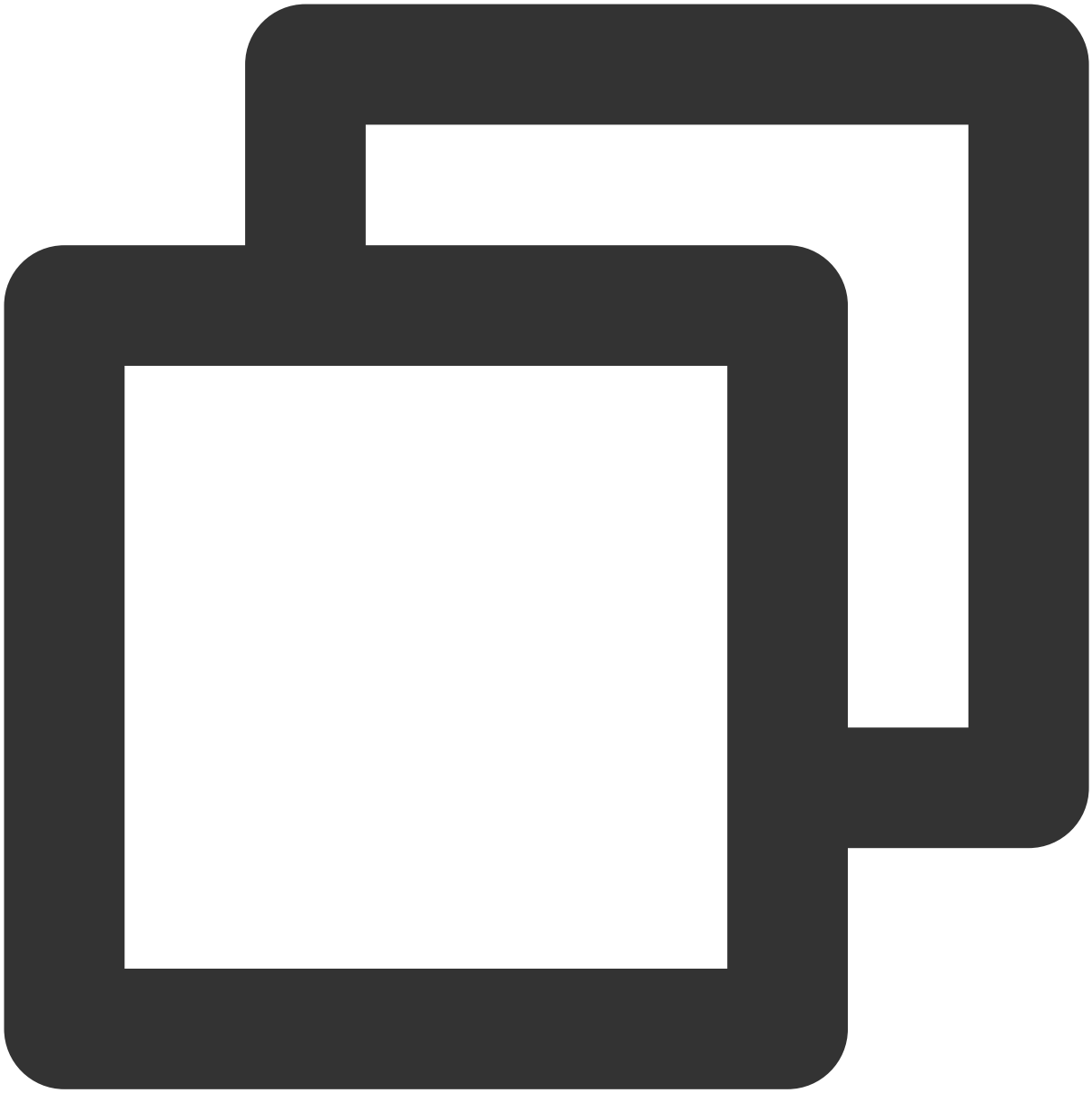
如果业务需要定期清理数据库中的多余数据，定时做物化视图更新，定时 vacuum full，定时执行 DML 等数据库中需要定时做的事情，在 PostgreSQL 中可以通过以下功能来实现：
Linux 的 crontab 功能
pgadmin 的 pgAgent 功能
pgAgent 是 pgAdmin 工具中的一个插件，在 pgAdmin III v1.4 版本中引用。主要用于 PostgreSQL 的作业调度代理，能够在复杂的时间表上运行多步批处理 shell 和 SQL 任务。
需要注意的是，pgAgent 需要一些数据库表和其他对象的支持，因此需要先安装 pgAgent 插件。

操作方法

配置 pgAgent

1. 登录 PostgreSQL 实例，登录完成后，创建您的业务数据库。
2. 在需要开通 pgAgent 功能的 database 以及 postgres 库中分别执行以下语句：

注意：
一定要在 postgres 库中也创建 pgagent。



```
psql > create extension pgagent;
CREATE EXTENSION
```

3. 配置反馈完成后，您需要通过 pgAgent 工具开启任务调度程序。

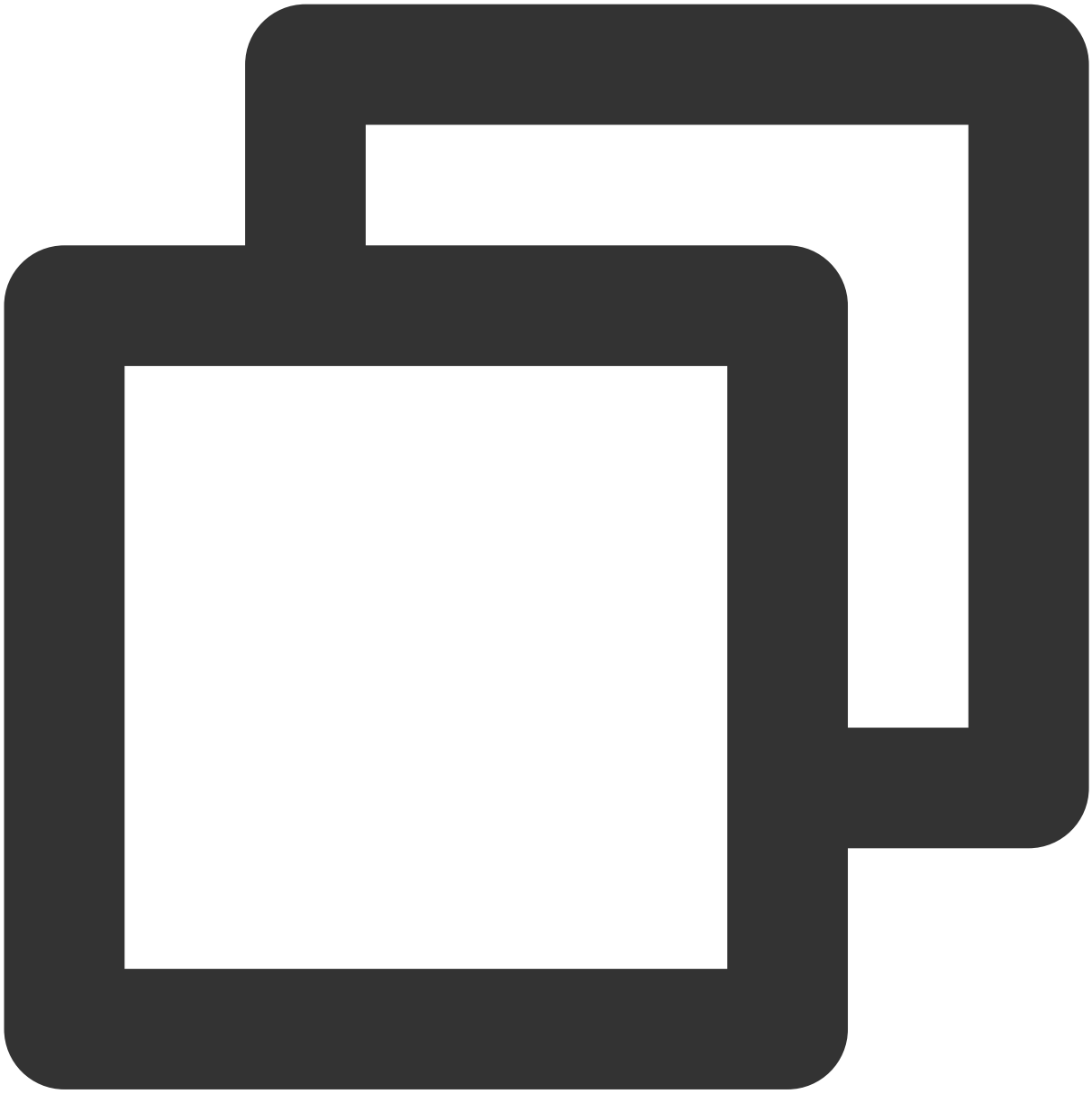
登录 CVM（建议 CVM 与 PostgreSQL 同一 VPC），根据实际数据库版本选择 pgAgent 版本，本文以11.8版本为例，安装 下载地址 中的 pgagent。

4. pgAgent 工具安装完成后，执行以下语句启动任务调度程序：

说明：

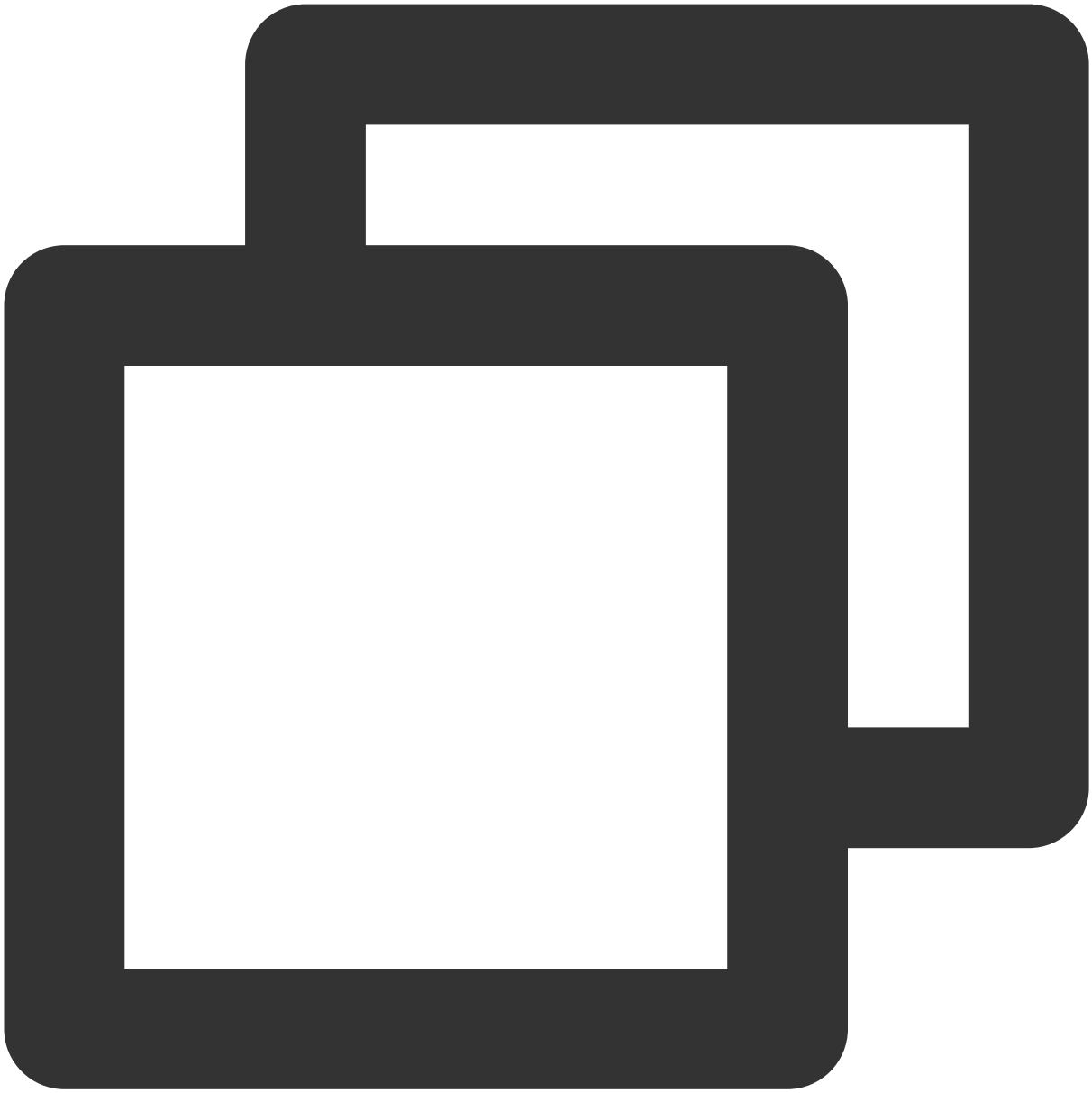
请根据实际安装的 pgAgent 版本来使用命令，如果安装的是10版本，则为 pgagent_10。

请注意 dbname 一定需要填写 postgres，不能填写需要执行定时调度程序的 database，否则 pgAdmin 的页面中无法显示 job 配置项。
当连接执行时候，报错 ERROR: Unsupported schema version，请 提交工单 进行处理。



```
pgagent_11 hostaddr=IP地址 dbname=postgres user=用户名 port=端口 password=密码
```

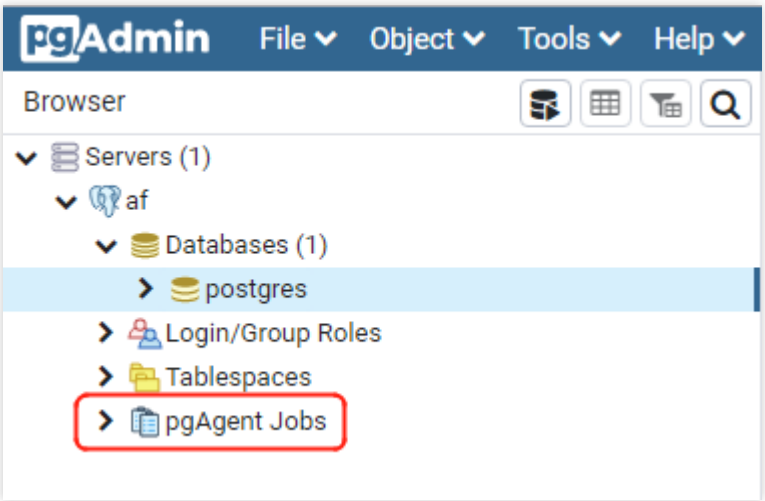
5. 执行成功后，无任何回显提示，您可以使用以下命令来判断进程是否启动成功：



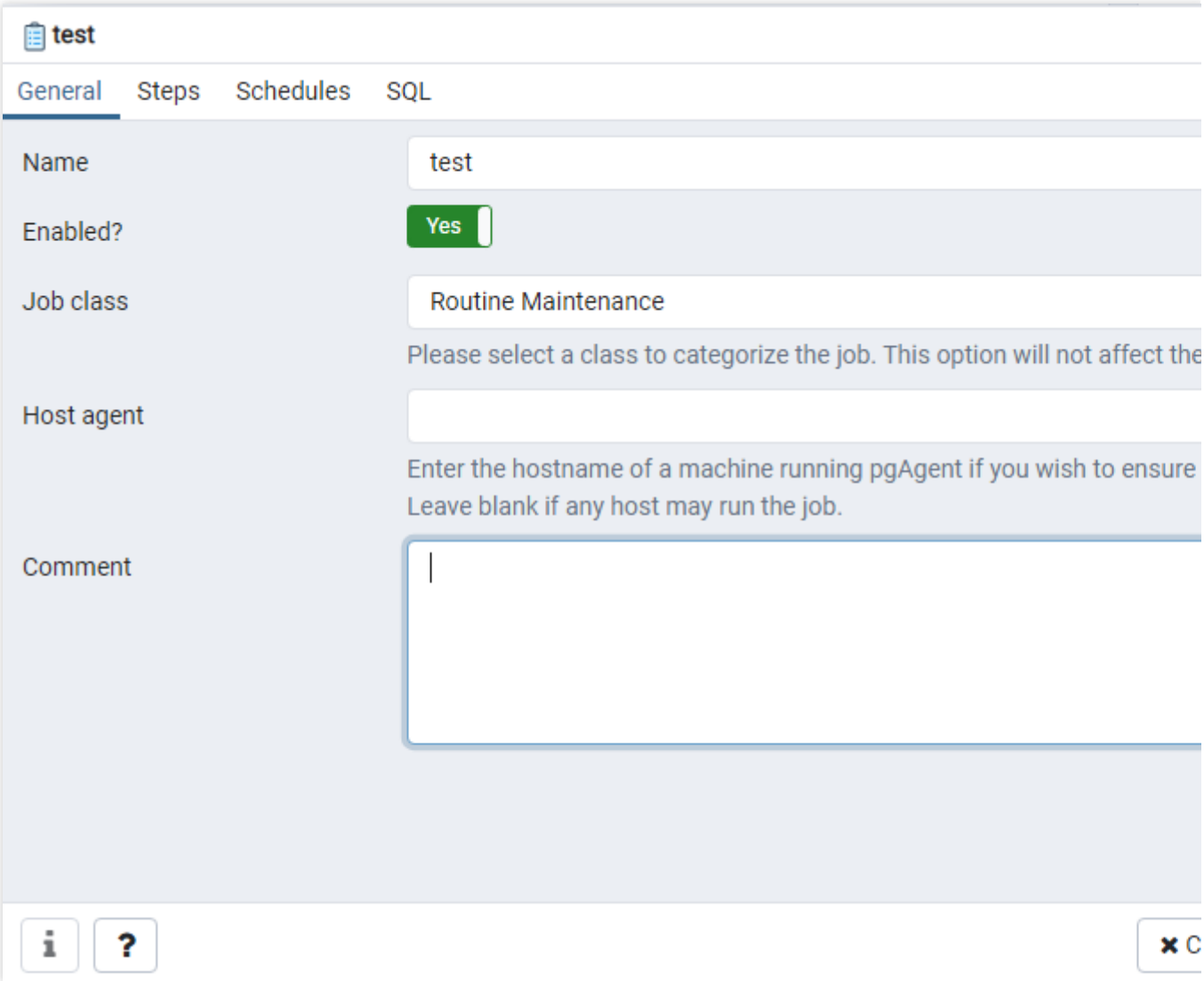
```
执行此语句，如果存在 pgagent 进程，则表示已经启动成功。
# ps -ef |grep pgagent
root      158553       1  0 Oct30 ?        00:00:15 pgagent_11 hostaddr=IP地址 dbnam
```

通过 pgAdmin 配置 pgAgent Jobs

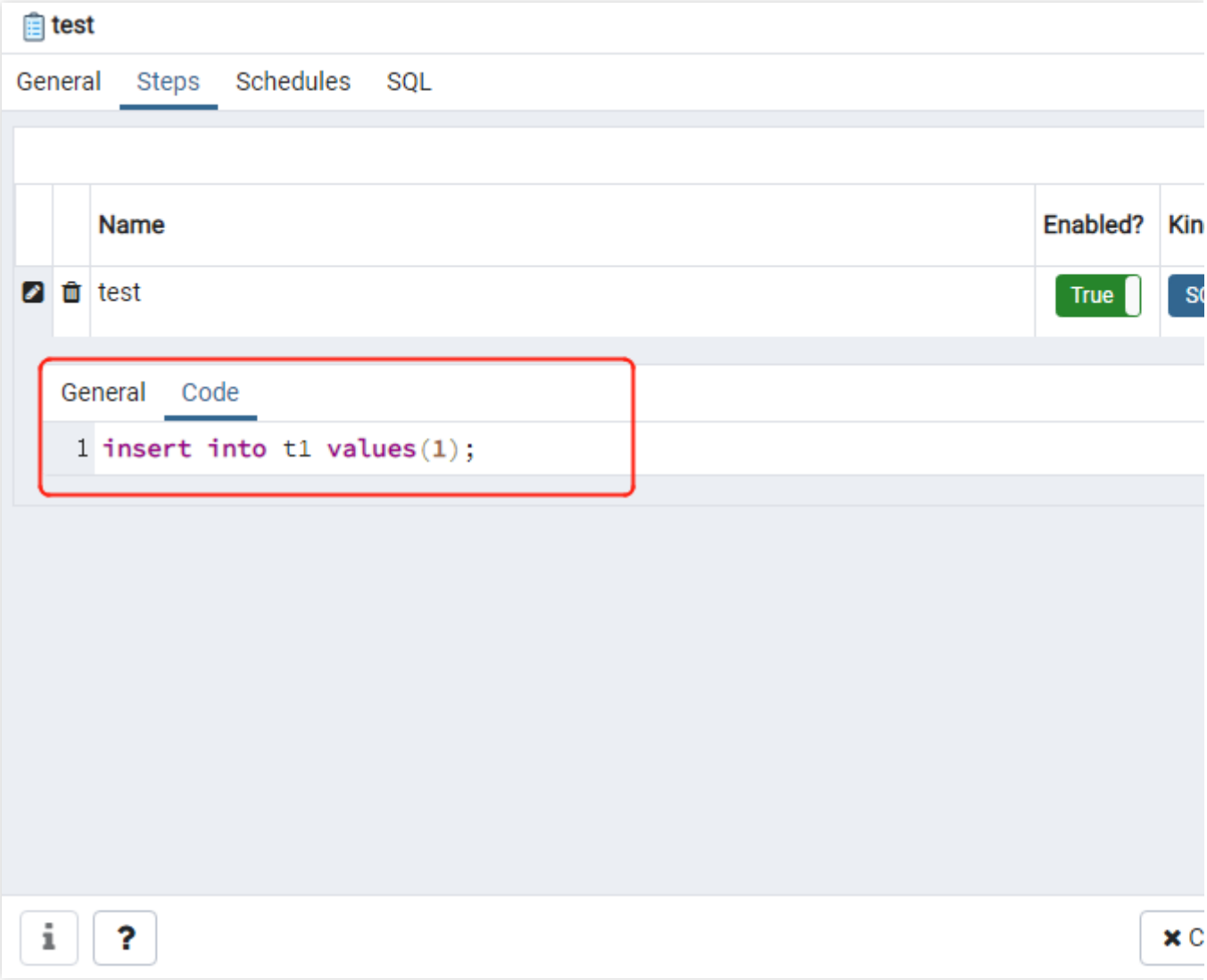
1. 登录 PostgreSQL 控制台，在实例列表单击实例 ID，进入实例详情页，开启外网地址。
2. 打开 pgAdmin 4，通过外网地址访问您的 PostgreSQL，此时可以在界面中看到 pgAgent Jobs。



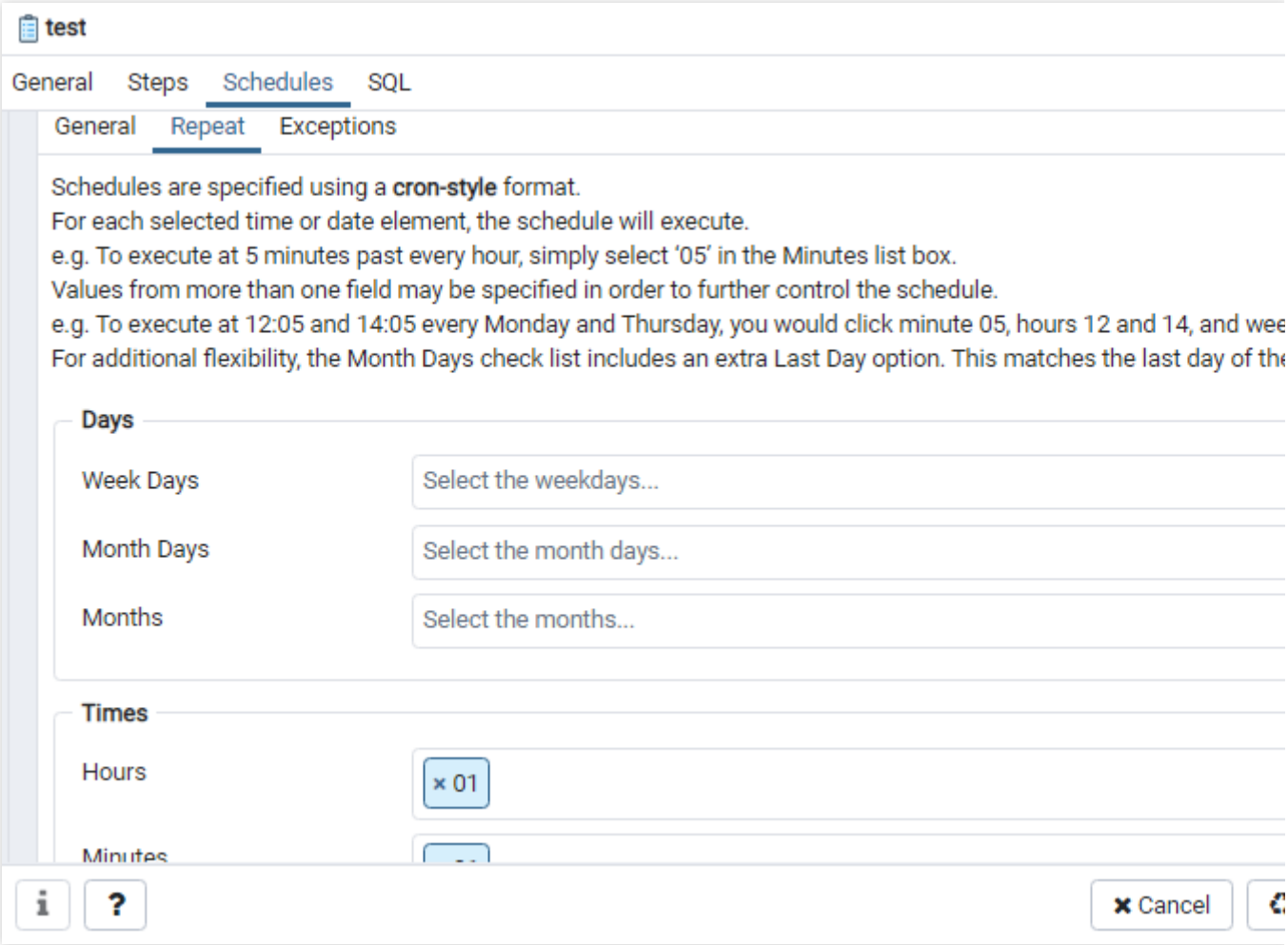
3. 在 pgAdmin 界面，右键选择 **pgAgent Jobs** > **Create** > **Create Jobs**，创建定时任务。
4. 在 General 界面，配置基础 Job 信息。



5. 进入 Step 界面，配置关于需要定时执行的内容，单击右上角的**+**添加一个 Step，为此 Step 配置一个名字，然后在 Code 子标签中配置需要执行的 SQL 语句等内容。



6. 进入 Schedules 界面，配置任务执行的调度信息：
7. 在下方 General 子标签中，主要配置任务生效时间。
8. 在下方 Repeat 子标签中，配置 crontab 风格的执行计划。



9. 完成时间配置之后，也可以在 Exceptions 子标签中配置不执行任务的时间。
10. 最后单击 **Save** 保存，此任务将根据配置的内容进行自动执行。

跨库访问

最近更新时间：2024-01-22 16:22:57

云数据库 PostgreSQL 提供用于访问外部数据源的一类插件，外部数据源包括本实例其他库中数据或者其他实例的数据。跨库访问插件包含同构的跨库访问插件 dblink、postgresql_fdw，异构的跨库访问插件 mysql_fdw、cos_fdw。
跨库访问使用步骤如下：
1. 使用 “CREATE EXTENSION” 语句安装插件。
2. 为每个需要连接的远程数据库创建一个外部服务器对象并创建链接映射。
3. 使用对应的命令访问外部表以获取数据。
由于跨库访问插件可以直接跨实例访问或在同实例中进行跨 database 访问。云数据库 PostgreSQL 对创建外部服务器对象时进行了权限控制优化，根据目标实例所在环境进行分类管理。在开源版本基础上增加了额外辅助参数，来验证用户身份和调整网络策略。具体请参考下文 插件辅助参数。

插件辅助参数

host
跨实例访问时候为必须项。目标实例的 IP 地址。
port
跨实例访问时候为必须项。目标实例的 port。
instanceid
实例 ID
在云数据库 PostgreSQL 间跨实例访问时使用，当跨实例访问时为必选项。格式类似 postgres-xxxxxx、pgro-xxxxxx，可在 控制台 查看。
如果目标实例在腾讯云 CVM 上，则为 CVM 机器的实例 ID，格式类似 ins-xxxxx。
dbname
database 名，填写需要访问的远端 PostgreSQL 服务的 database 名字。若不跨实例访问，仅在同实例中进行跨库访问，则只需要配置此参数即可，其他参数都可为空。
access_type
非必须项。目标实例所属类型：
目标实例为 TencentDB 实例，包括云数据库 PostgreSQL、云数据库 MySQL 等，如果不显示指定，则默认该项。
目标实例在腾讯云 CVM 机器上。
目标实例为腾讯云外网自建。
目标实例为云 VPN 接入的实例。
目标实例为自建 VPN 接入的实例。
目标实例为专线接入的实例。

uin

非必须项。实例所属的账号 ID，通过该信息鉴定用户权限，可参见 查询 uin。

own_uin

非必须项。实例所属的主账号 ID，同样需要该信息鉴定用户权限。

vpcid

非必须项。私有网络 ID，目标实例如果在腾讯云 CVM 的 VPC 网络中，则需要提供该参数，可在 VPC 控制台 中查看。

subnetid

非必须项。私有网络子网 ID，目标实例如果在腾讯云 CVM 的 VPC 网络中，则需要提供该参数，可在 VPC 控制台 的子网中查看。

dcgid

非必须项。专线 ID，目标实例如果需要通过专线网络连接，则需要提供该参数值。

vpngwid

非必须项。VPN 网关 ID，目标实例如果需要通过 VPN 进行网络连接，则需要提供该参数值。

region

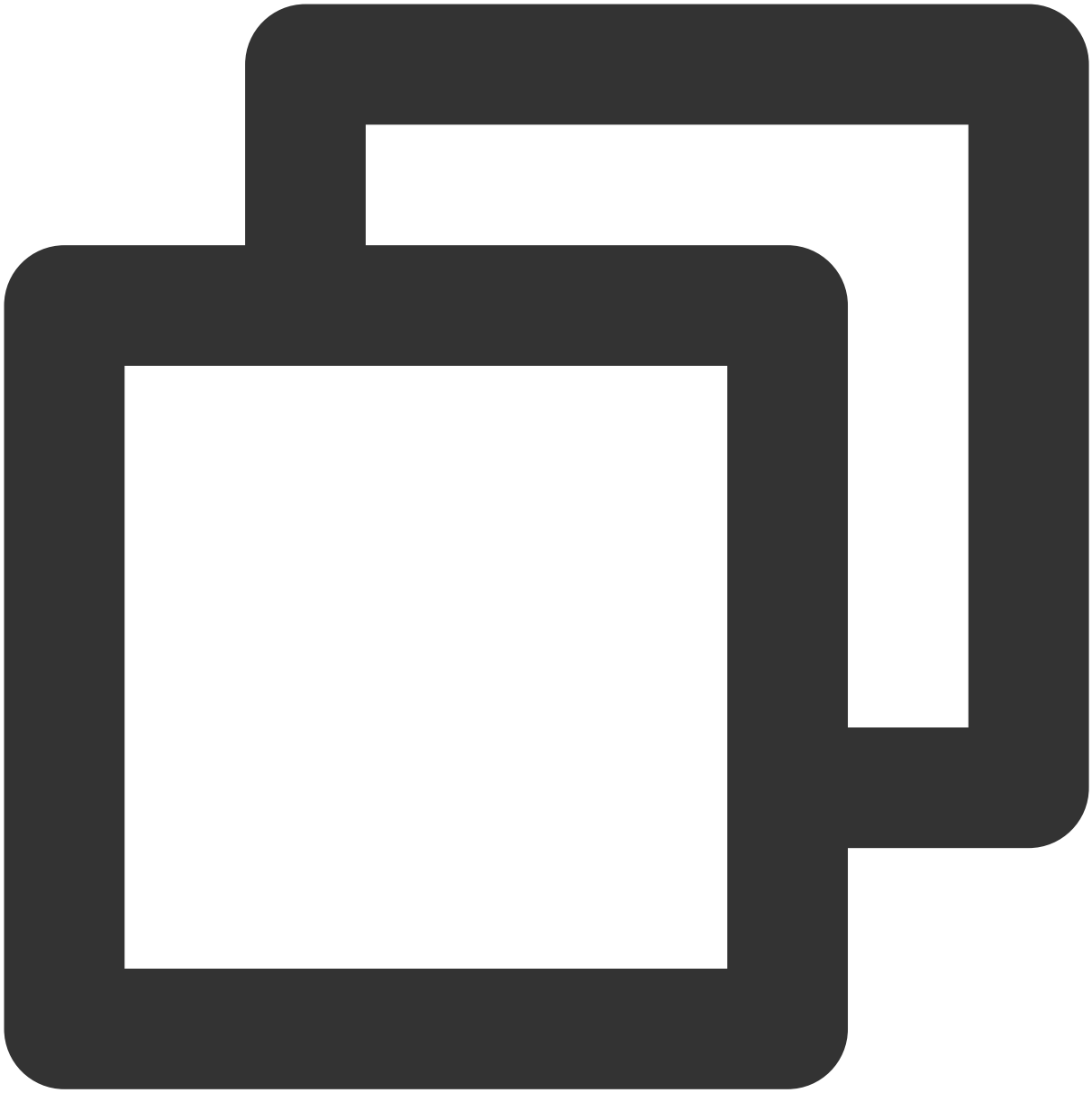
非必须项。目标实例所在地域，如 “ap-guangzhou” 表示广州。如果需要跨地域访问数据，则需要提供该参数值。

使用 postgres_fdw 示例

使用 postgres_fdw 插件可以访问本实例其他库或者其他 postgres 实例的数据。

步骤1：前置条件

1. 在本实例中创建测试数据。

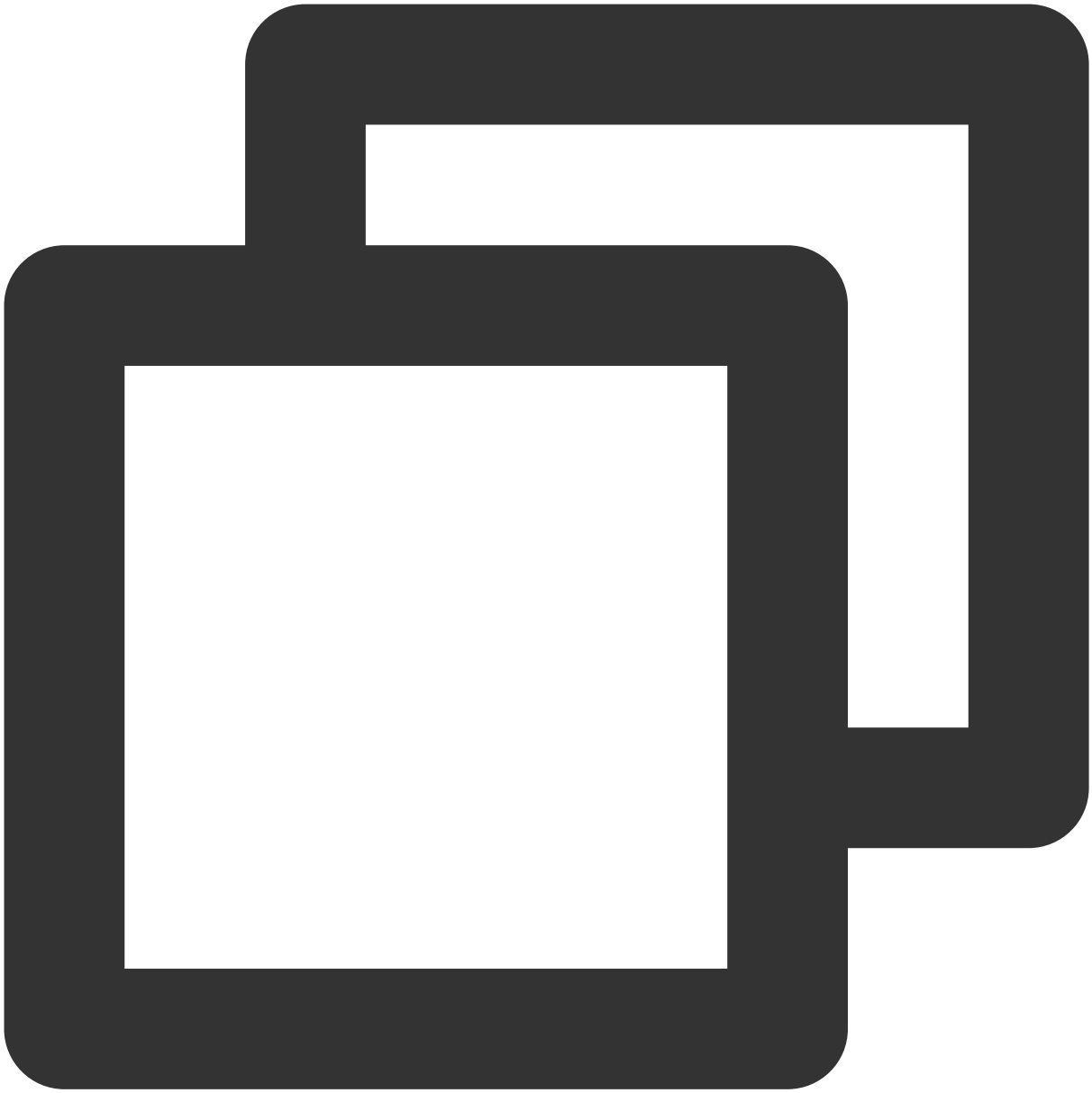


```
postgres=>create role user1 with LOGIN  CREATEDB PASSWORD 'password1';
postgres=>create database testdb1;
CREATE DATABASE
```

注意：

若创建插件报错，请 提交工单 联系腾讯云售后协助处理。

2. 在目标实例中创建测试数据。



```
postgres=>create role user2 with LOGIN  CREATEDB PASSWORD 'password2';
postgres=> create database testdb2;
CREATE DATABASE
postgres=> \\c testdb2 user2
You are now connected to database "testdb2" as user "user2".
testdb2=> create table test_table2(id integer);
CREATE TABLE
testdb2=> insert into test_table2 values (1);
INSERT 0 1
```

版权所有：腾讯云计算（北京）有限责任公司

步骤2：创建 postgres_fdw 插件

说明：

若创建插件时，提示插件不存在或权限不足，请 提交工单 处理。

```
#创建
postgres=> \\c testdb1
You are now connected to database "testdb1" as user "user1".
testdb1=> create extension postgres_fdw;
CREATE EXTENSION
#查看
testdb1=> \\dx
```

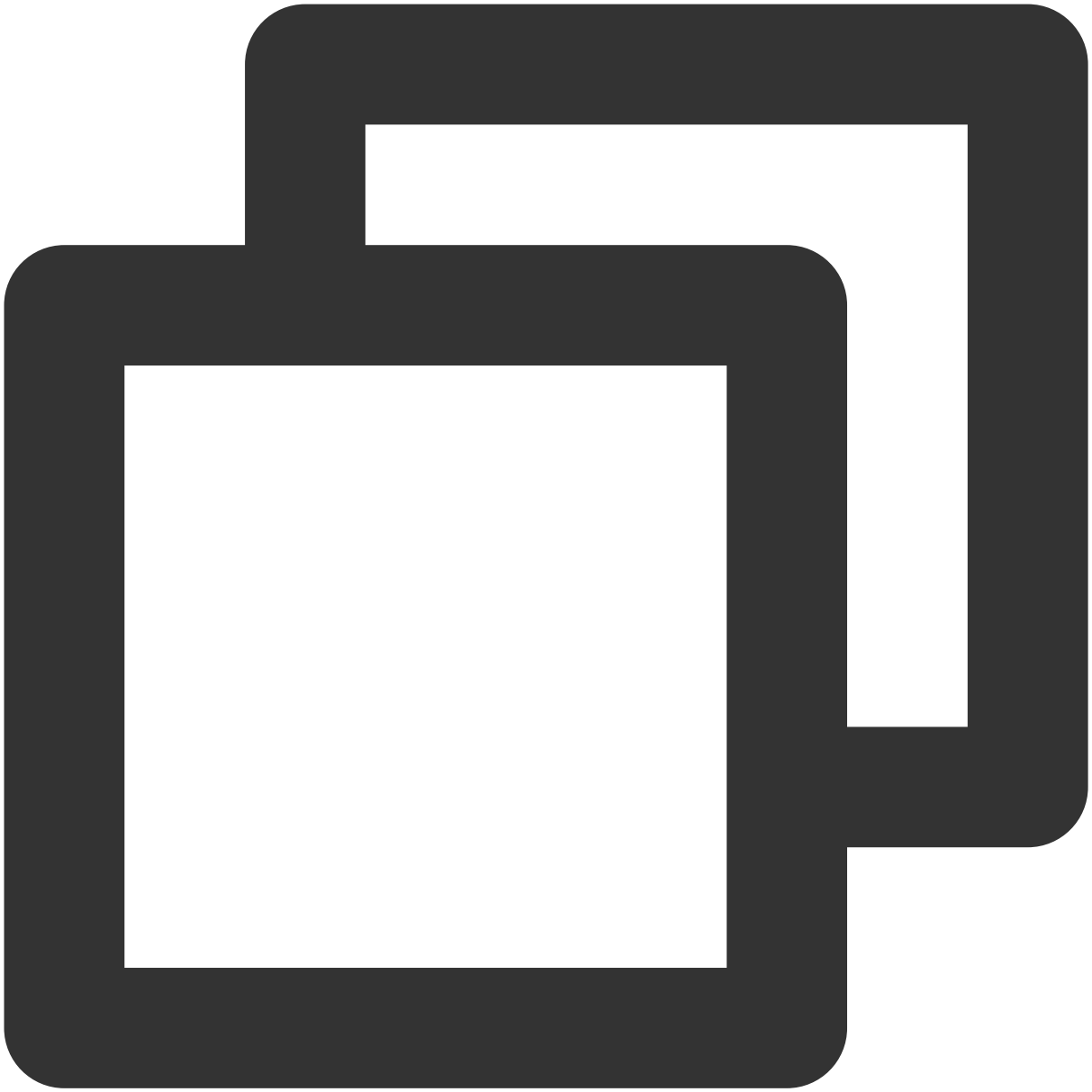
```
                              List of installed extensions
      Name      | Version |   Schema   |                     Description
--------------+---------+------------+---------------------------------------------
  plpgsql      | 1.0     | pg_catalog | PL/pgSQL procedural language
  postgres_fdw | 1.0     | public     | foreign-data wrapper for remote PostgreSQL
(2 rows)
```

步骤3：创建 SERVER

注意：

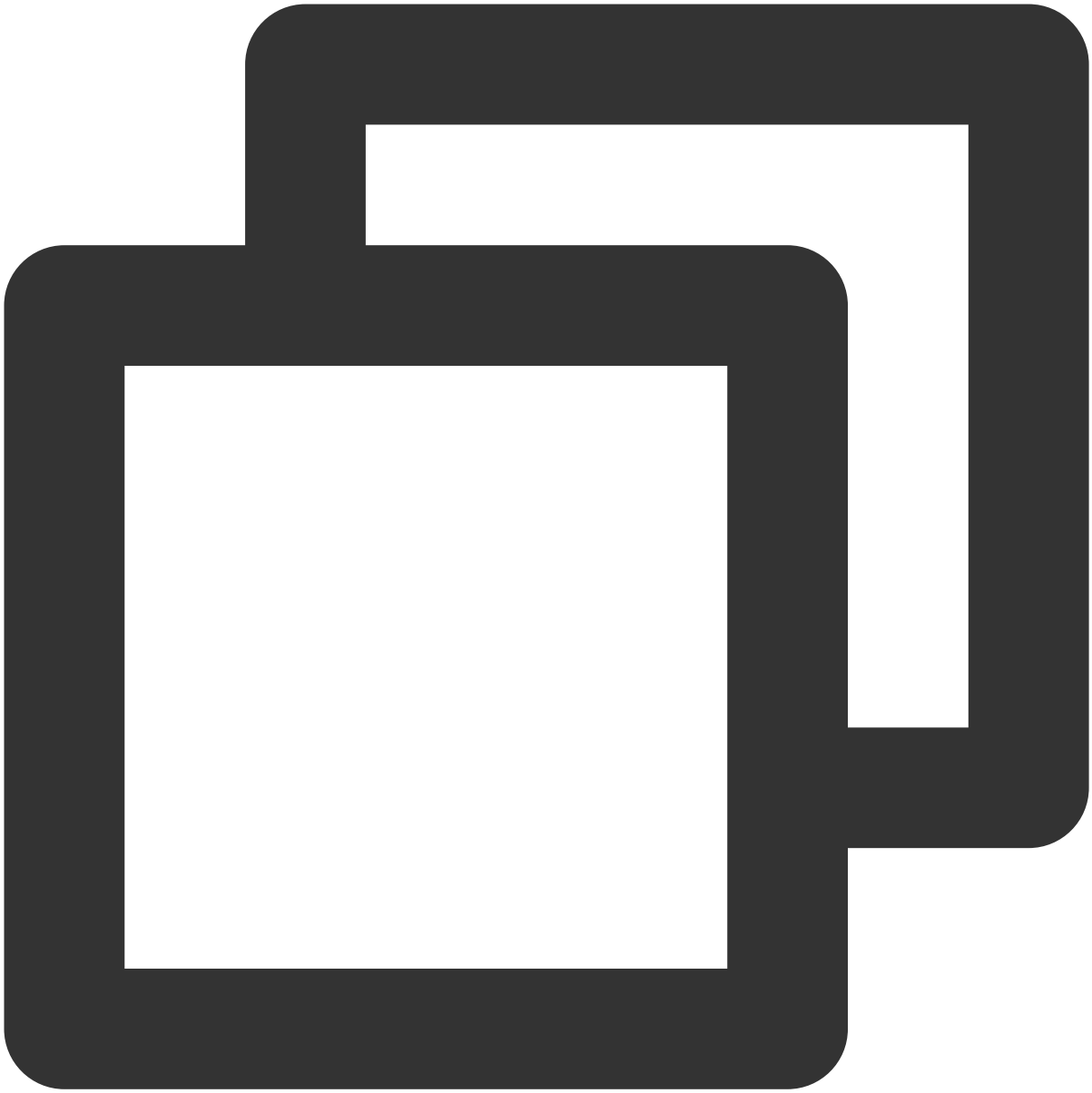
仅 v10.17_r1.2、v11.12_r1.2、v12.7_r1.2、v13.3_r1.2、v14.2_r1.0 及之后的内核版本支持跨实例访问。

跨实例访问。



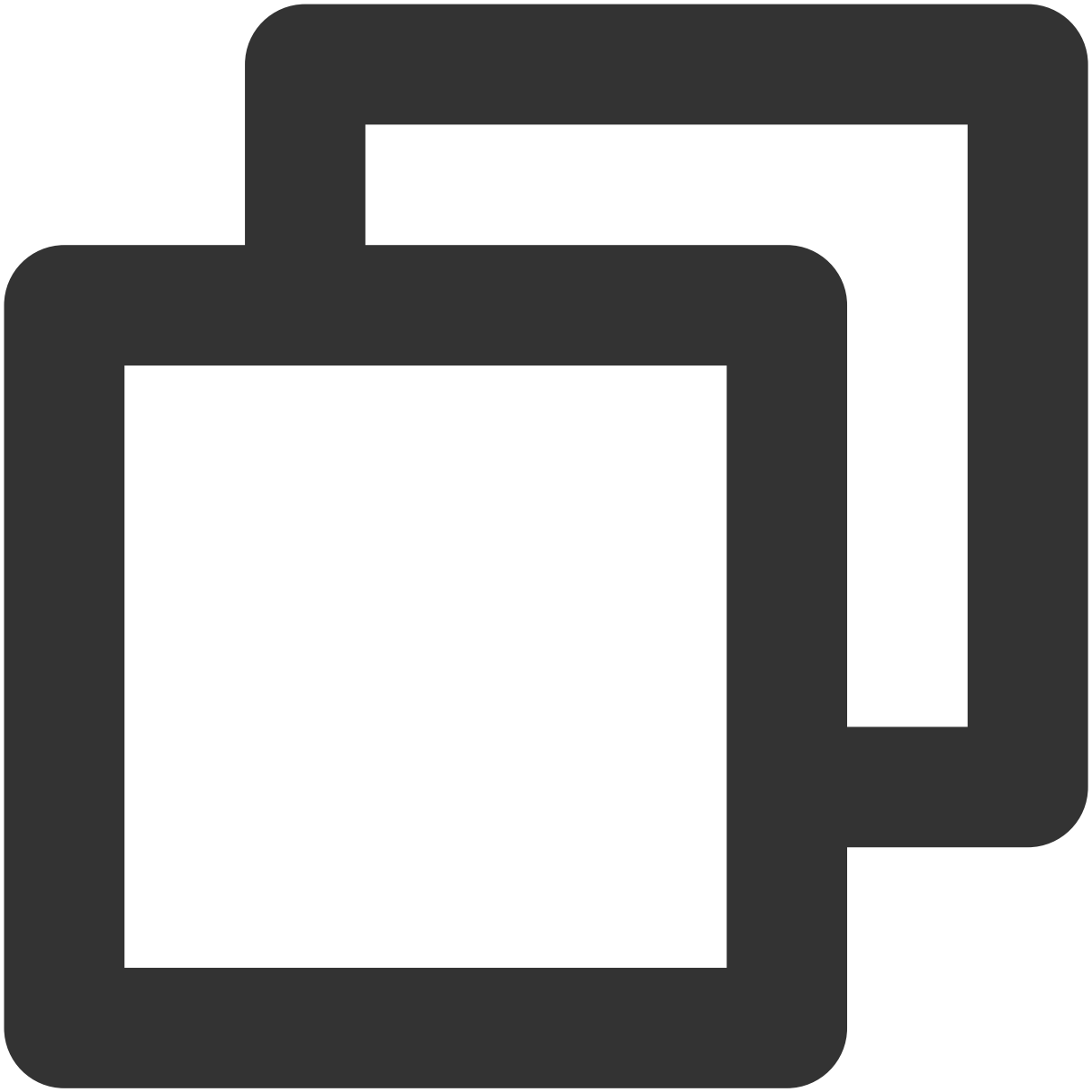
```
#从本实例的 testdb1 访问目标实例 testdb2 的数据
testdb1=>create server srv_test1 foreign data wrapper postgres_fdw options (host 'x
CREATE SERVER
```

不跨实例，仅跨 database 访问，仅需要填写 dbname 参数即可。



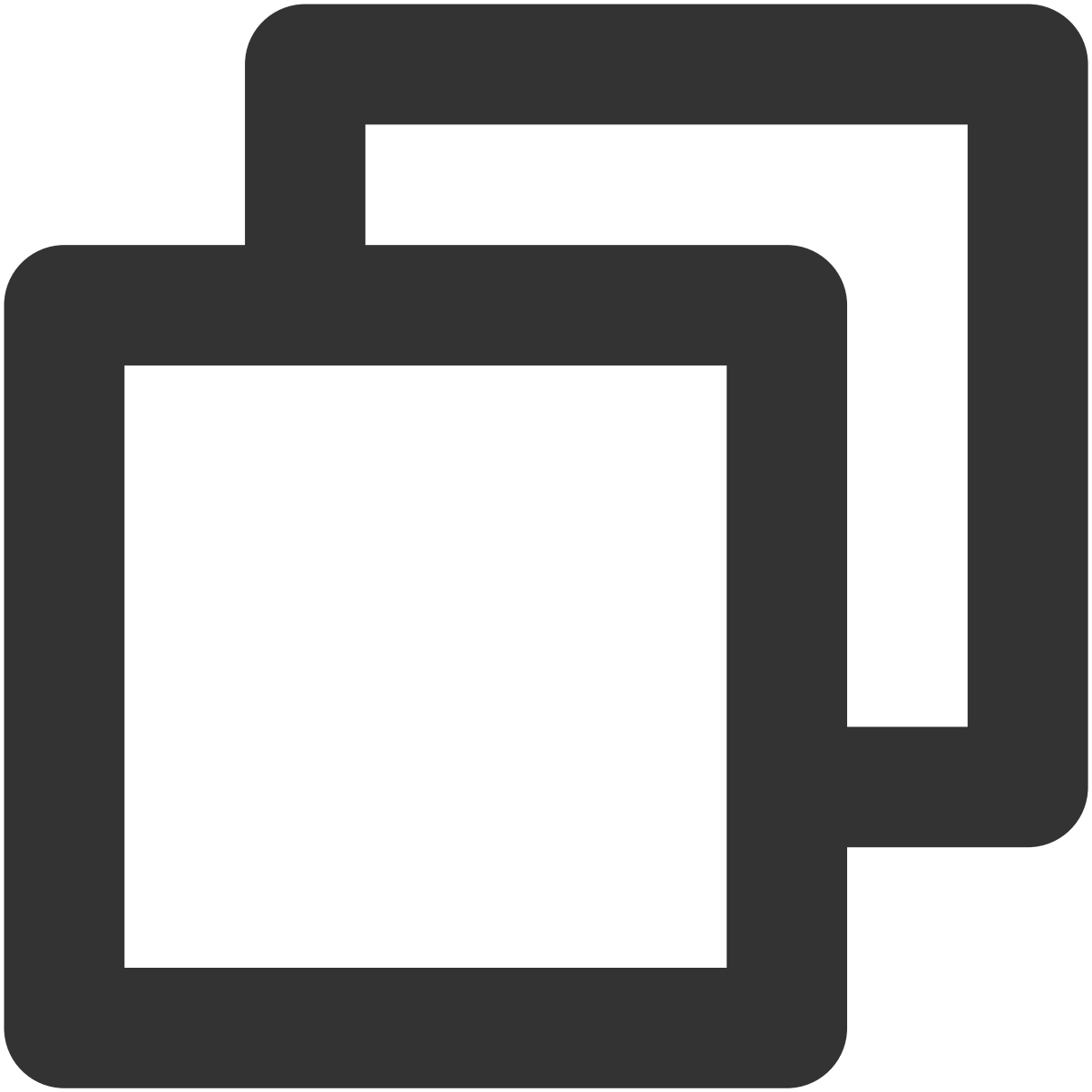
```
#从本实例的 testdb1 访问本实例 testdb2 的数据
create server srv_test1 foreign data wrapper postgres_fdw options (dbname 'testdb2'
```

目标实例在腾讯云 CVM 上，且网络类型为基础网络。



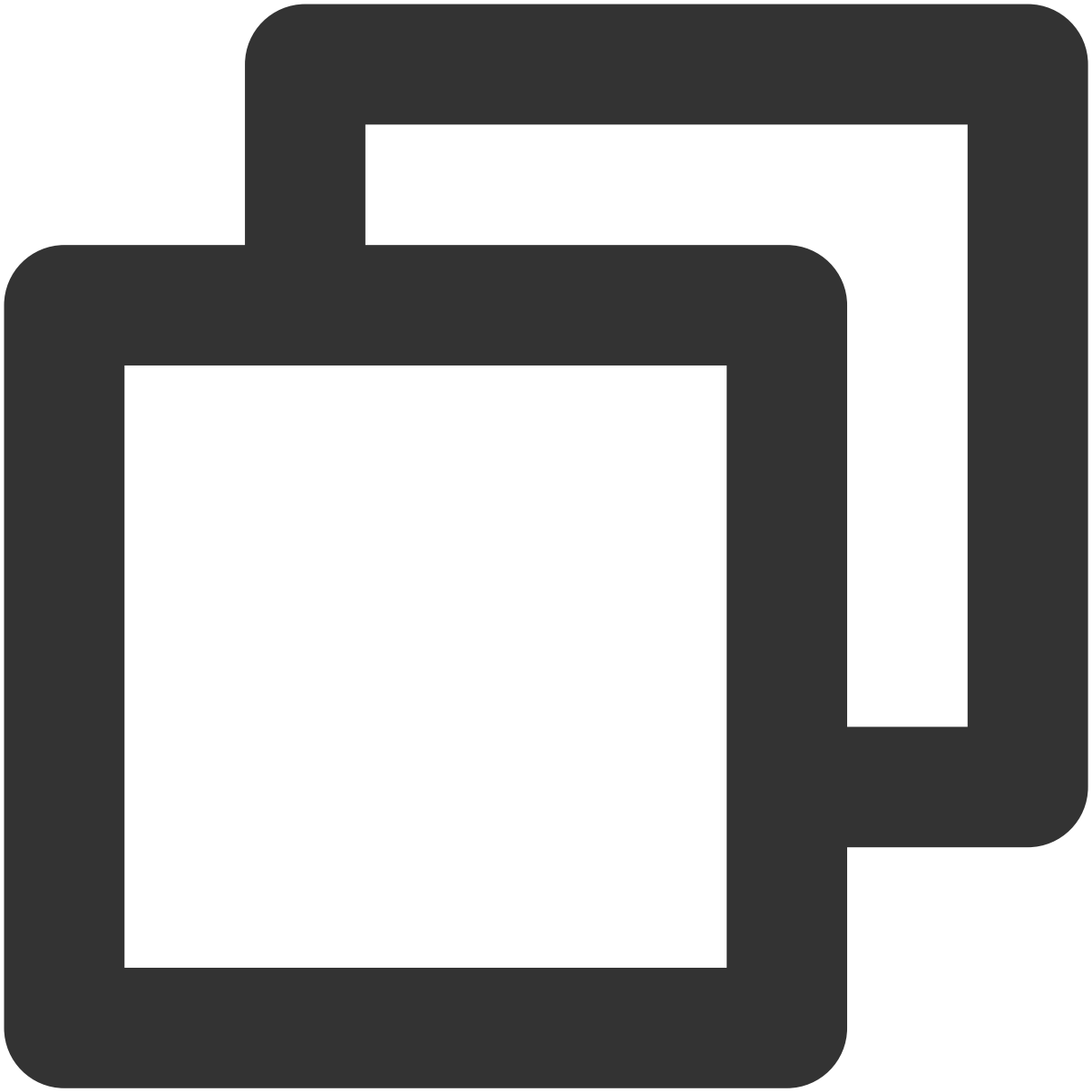
```
testdb1=>create server srv_test foreign data wrapper postgres_fdw options (host '
CREATE SERVER
```

目标实例在腾讯云 CVM 上，且网络类型为私有网络。



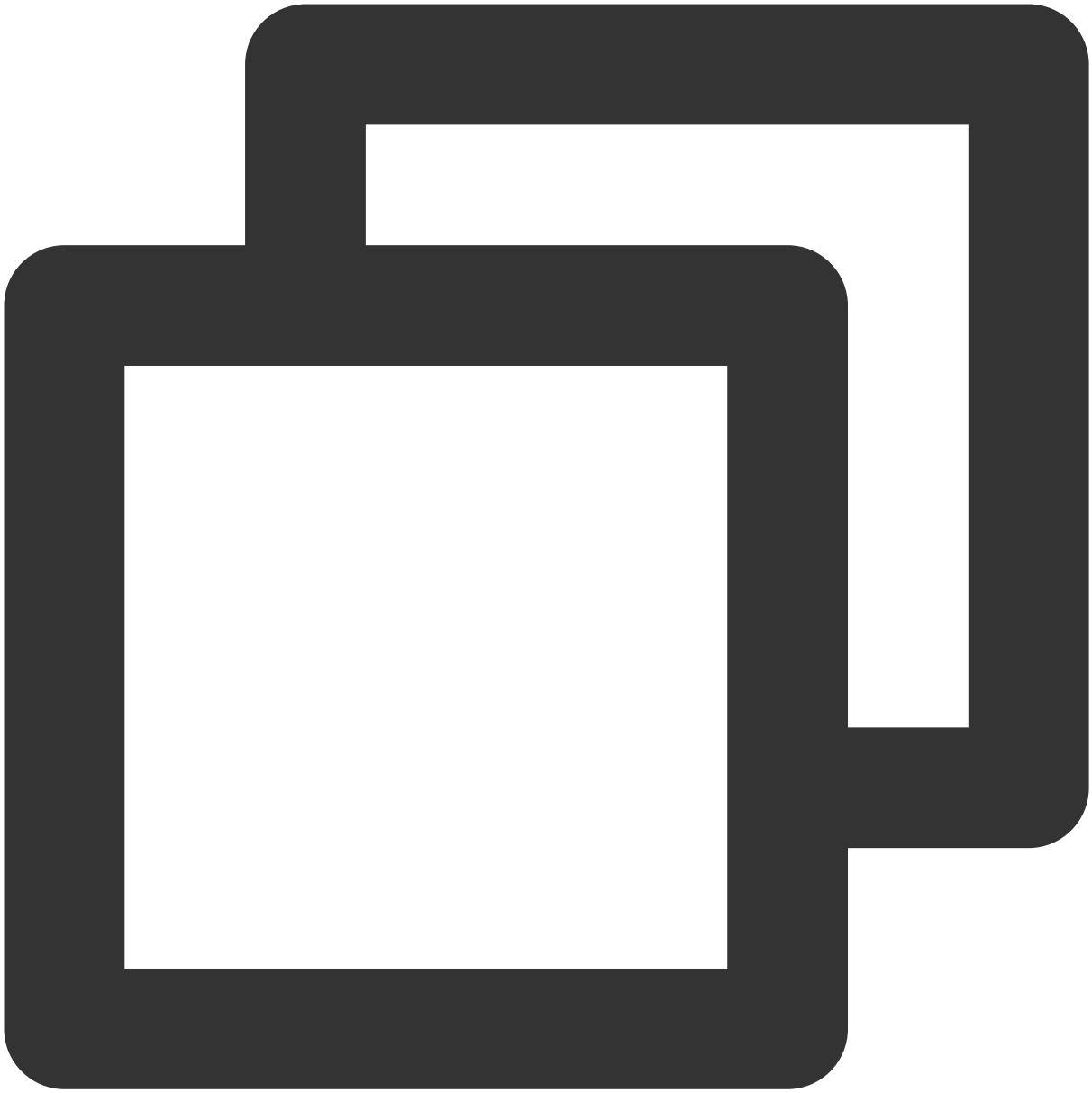
```
testdb1=>create server srv_test1 foreign data wrapper postgres_fdw options (host
CREATE SERVER
```

目标实例在腾讯云外网自建。



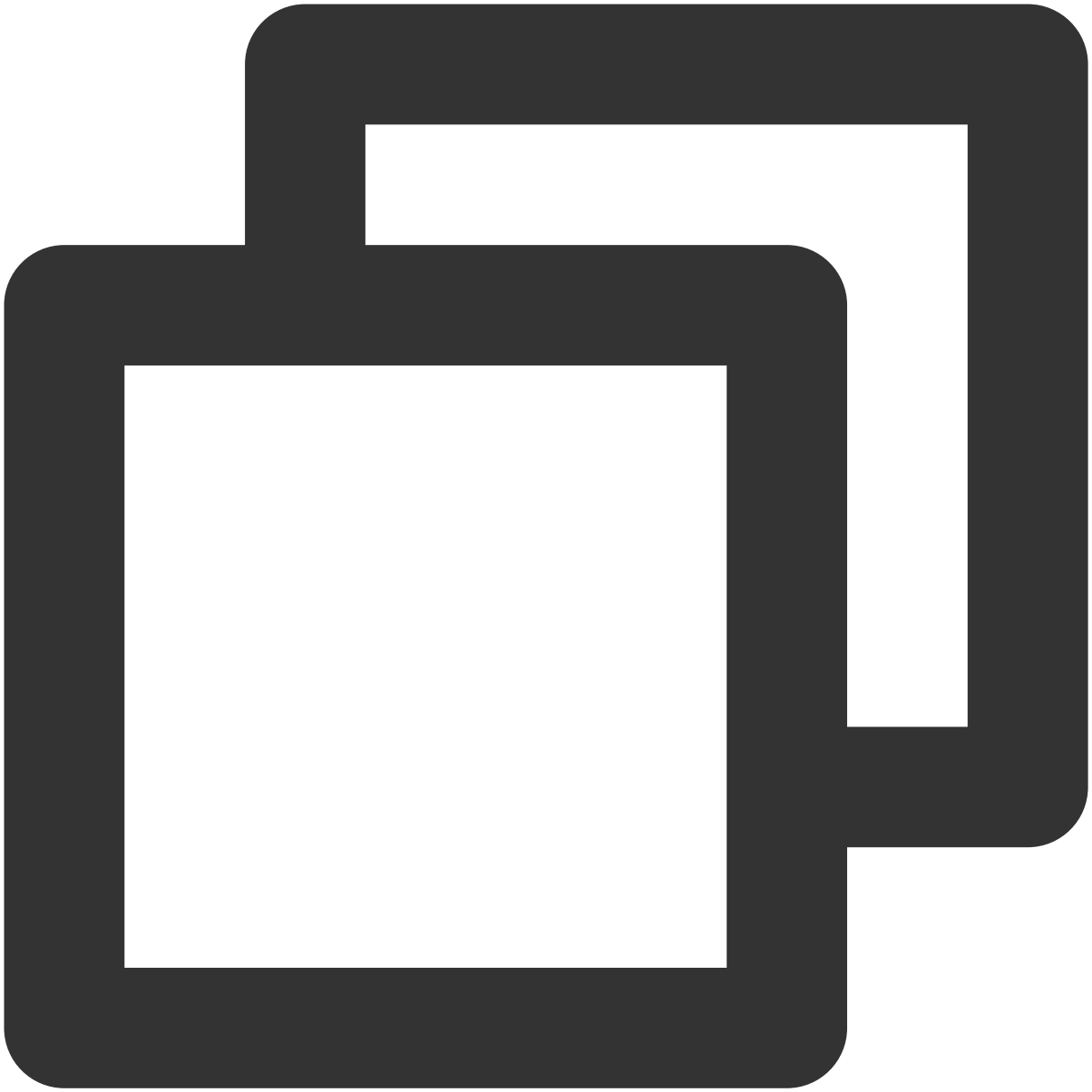
```
testdb1=>create server srv_test1 foreign data wrapper postgres_fdw options (host
CREATE SERVER
```

目标实例在腾讯云 VPN 接入的实例。



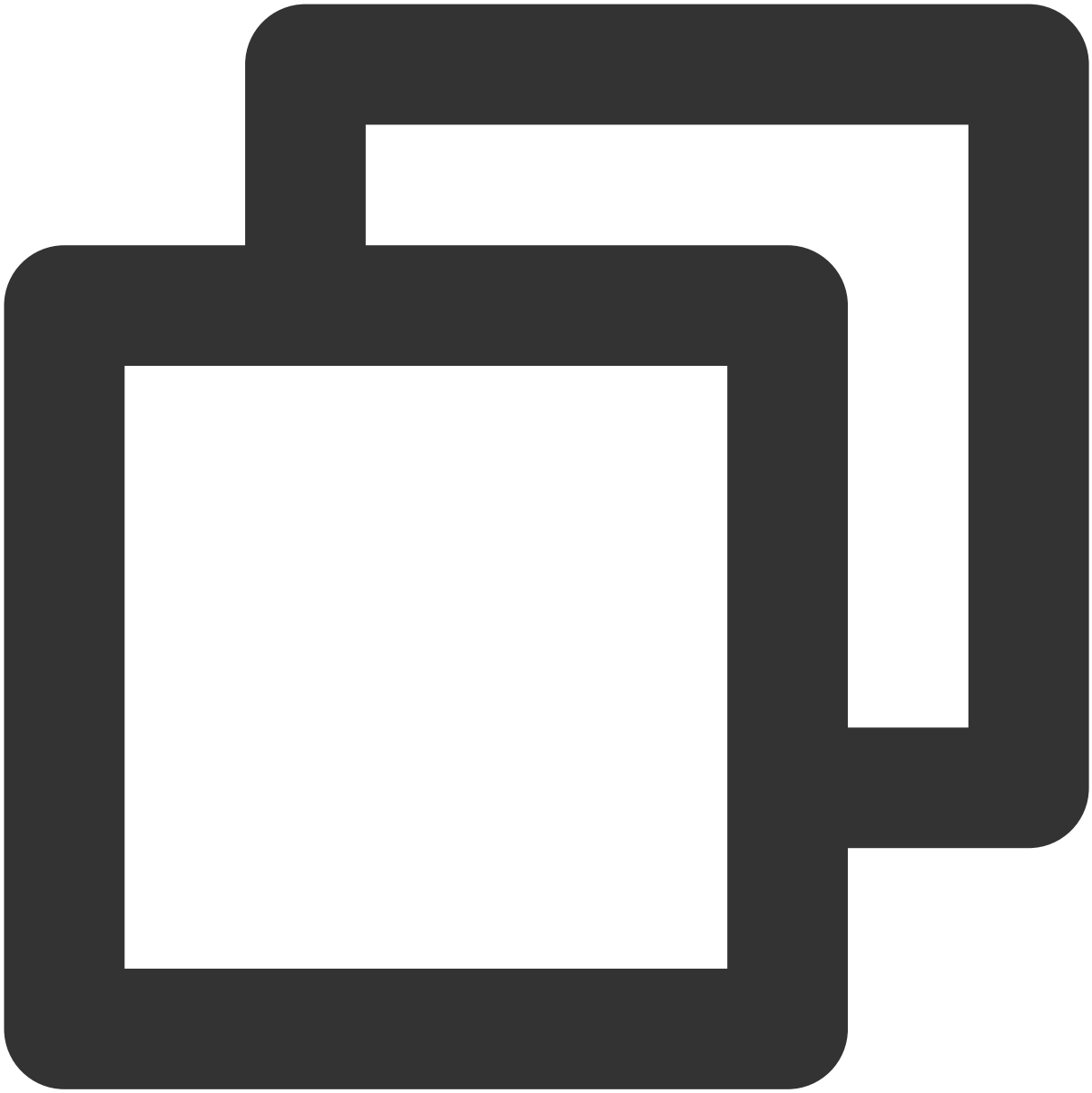
```
testdb1=>create server srv_test1 foreign data wrapper postgres_fdw options (host
```

目标实例在自建 VPN 接入的实例。



```
testdb1=>create server srv_test1 foreign data wrapper postgres_fdw options (host
```

目标实例在腾讯云专线接入的实例。

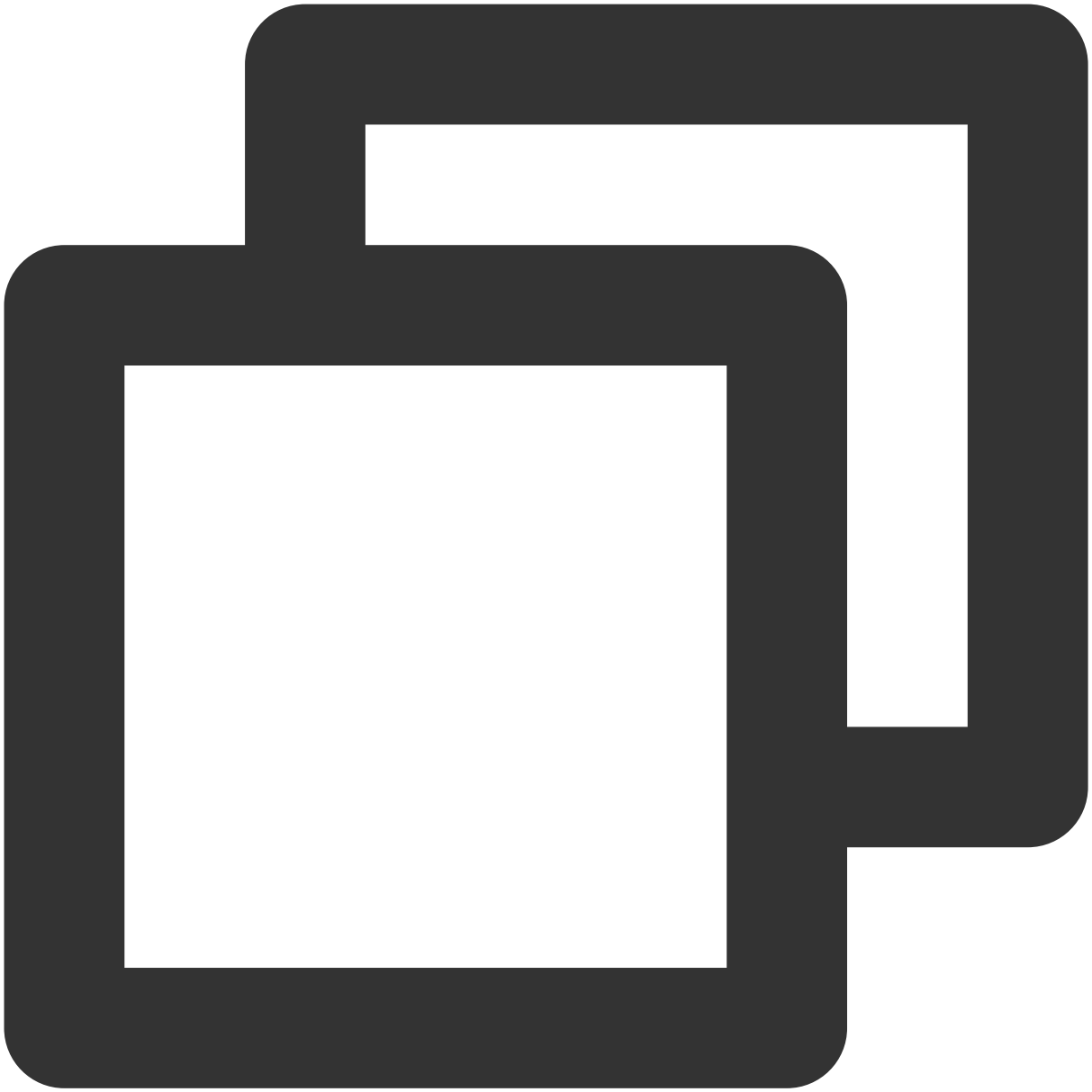


```
testdb1=>create server srv_test1 foreign data wrapper postgres_fdw options (host
CREATE SERVER
```

步骤4：创建用户映射

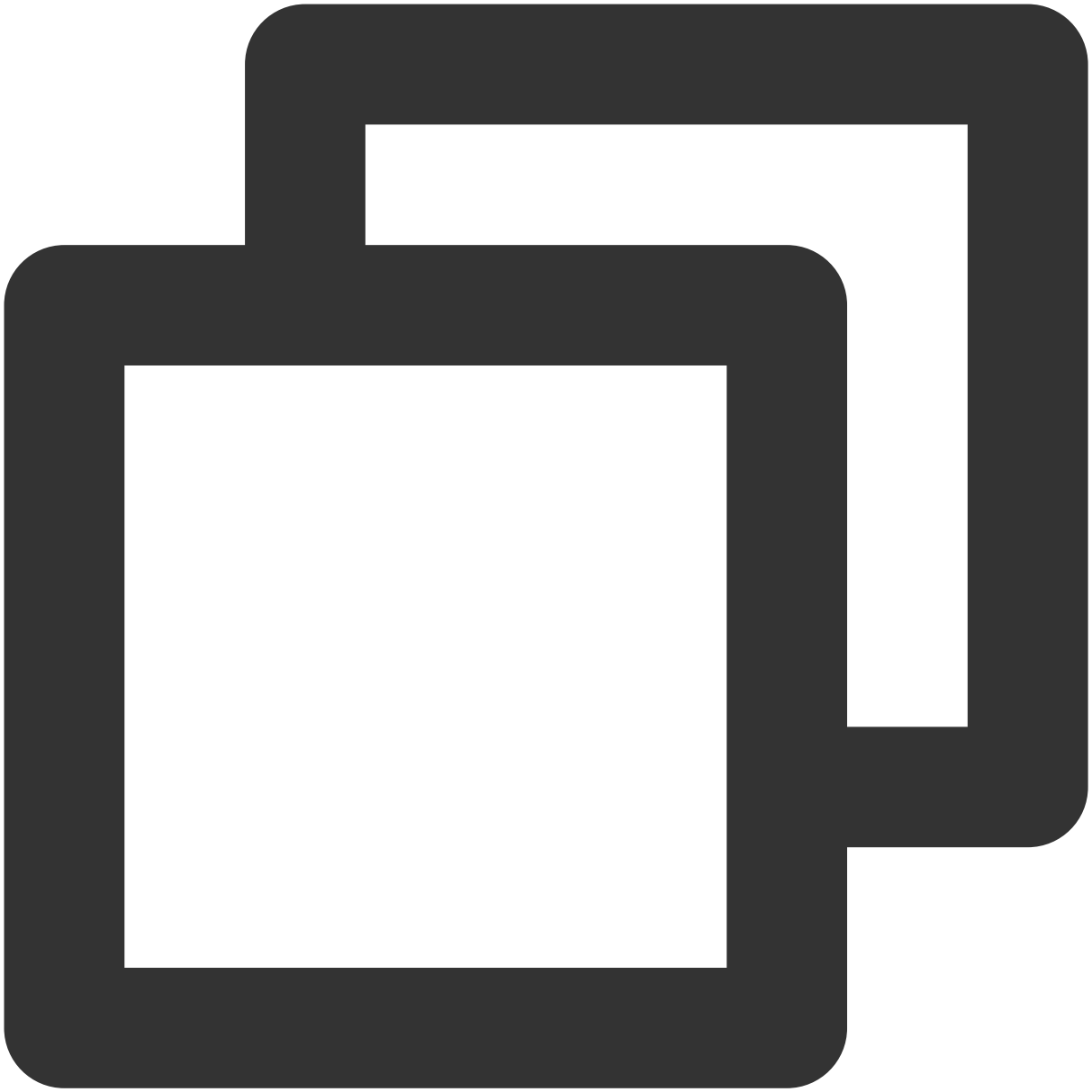
说明：

同实例的跨 database 访问则可跳过此步骤。



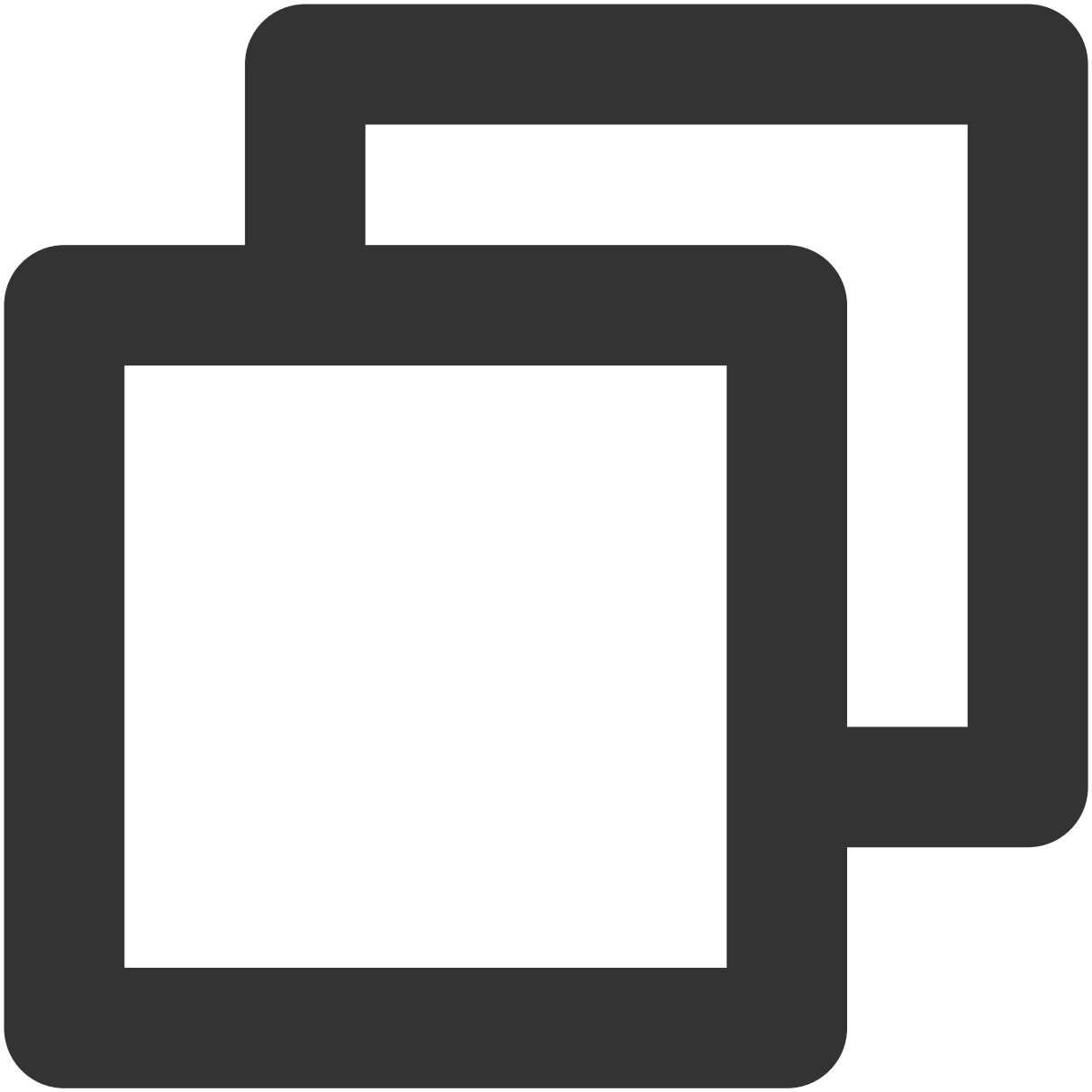
```
testdb1=> create user mapping for user1 server srv_test1 options (user 'user2',pass
CREATE USER MAPPING
```

步骤5：创建外部表



```
testdb1=> create foreign table foreign_table1(id integer) server srv_test1 options(
CREATE FOREIGN TABLE
```

步骤6：访问外部数据



```
testdb1=> select * from foreign_table1;
  id
----
   1
(1 row)
```

参考链接

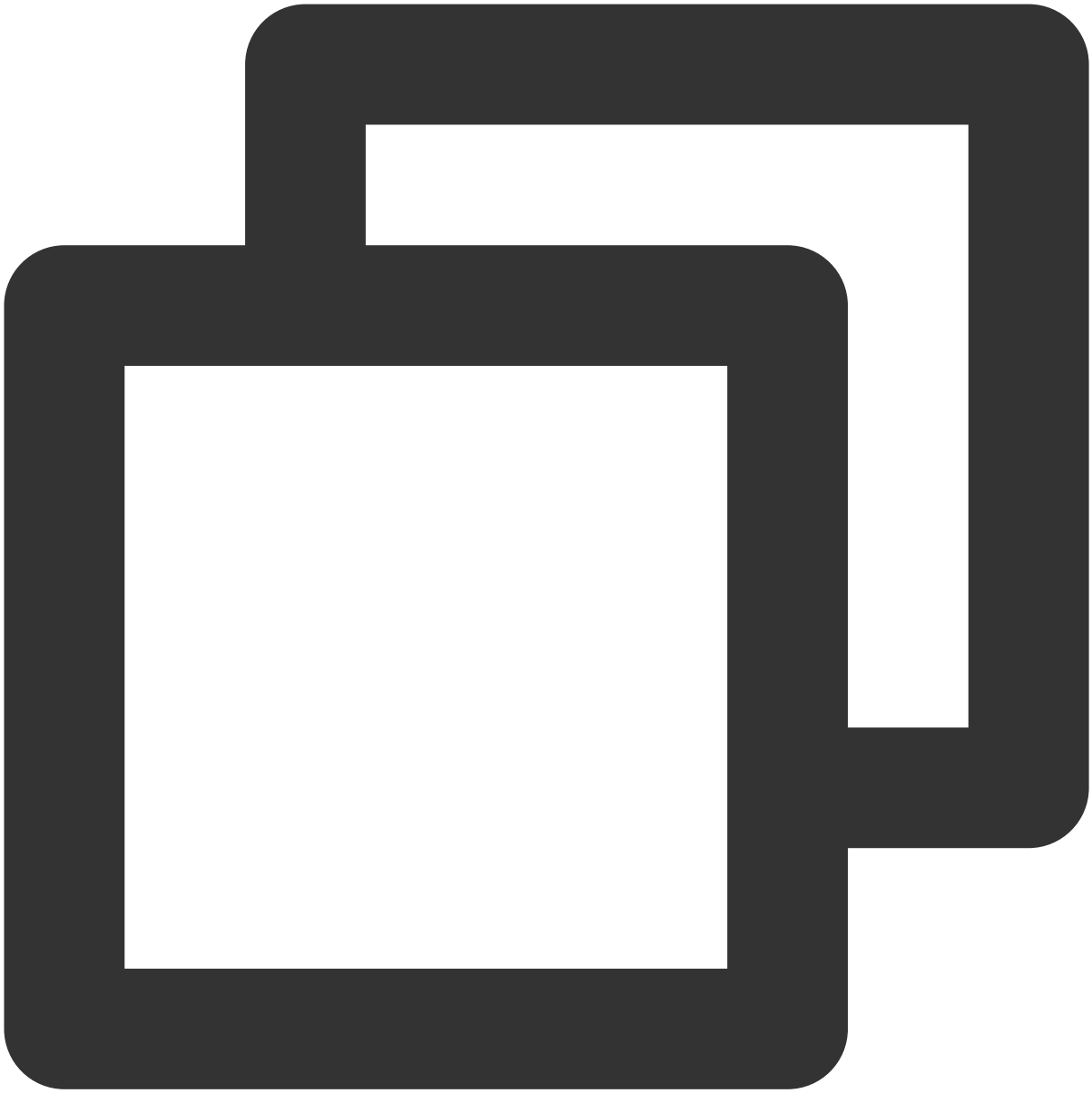
postgres_fdw 官方介绍

9.3 版本 SERVER 创建

9.5 版本 SERVER 创建
10 版本 SERVER 创建
11 版本 SERVER 创建
12 版本 SERVER 创建
13 版本 SERVER 创建
14 版本 SERVER 创建

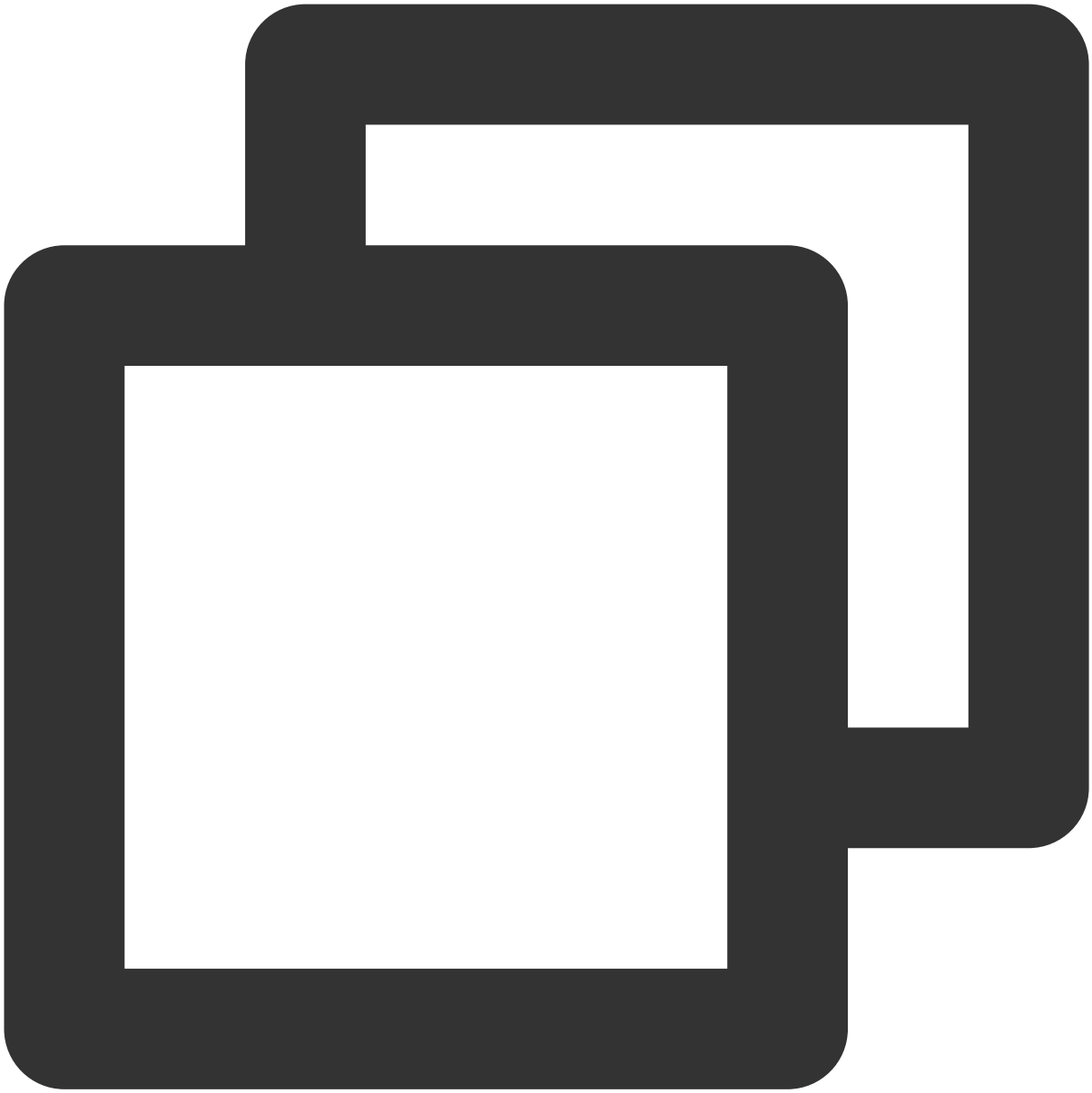
使用 dblink 示例

步骤一：创建 dblink 插件



```
postgres=> create extension dblink;
postgres=> \\dx
                                        List of installed extensions
        Name         | Version |   Schema   |                              Description
---------------------+---------+------------+-------------------------------------
 dblink              | 1.2     | public     | connect to other PostgreSQL databases
 pg_stat_log         | 1.0     | public     | track runtime execution statistics of
 pg_stat_statements  | 1.6     | public     | track execution statistics of all SQL
 plpgsql             | 1.0     | pg_catalog | PL/pgSQL procedural language
(4 rows)
```

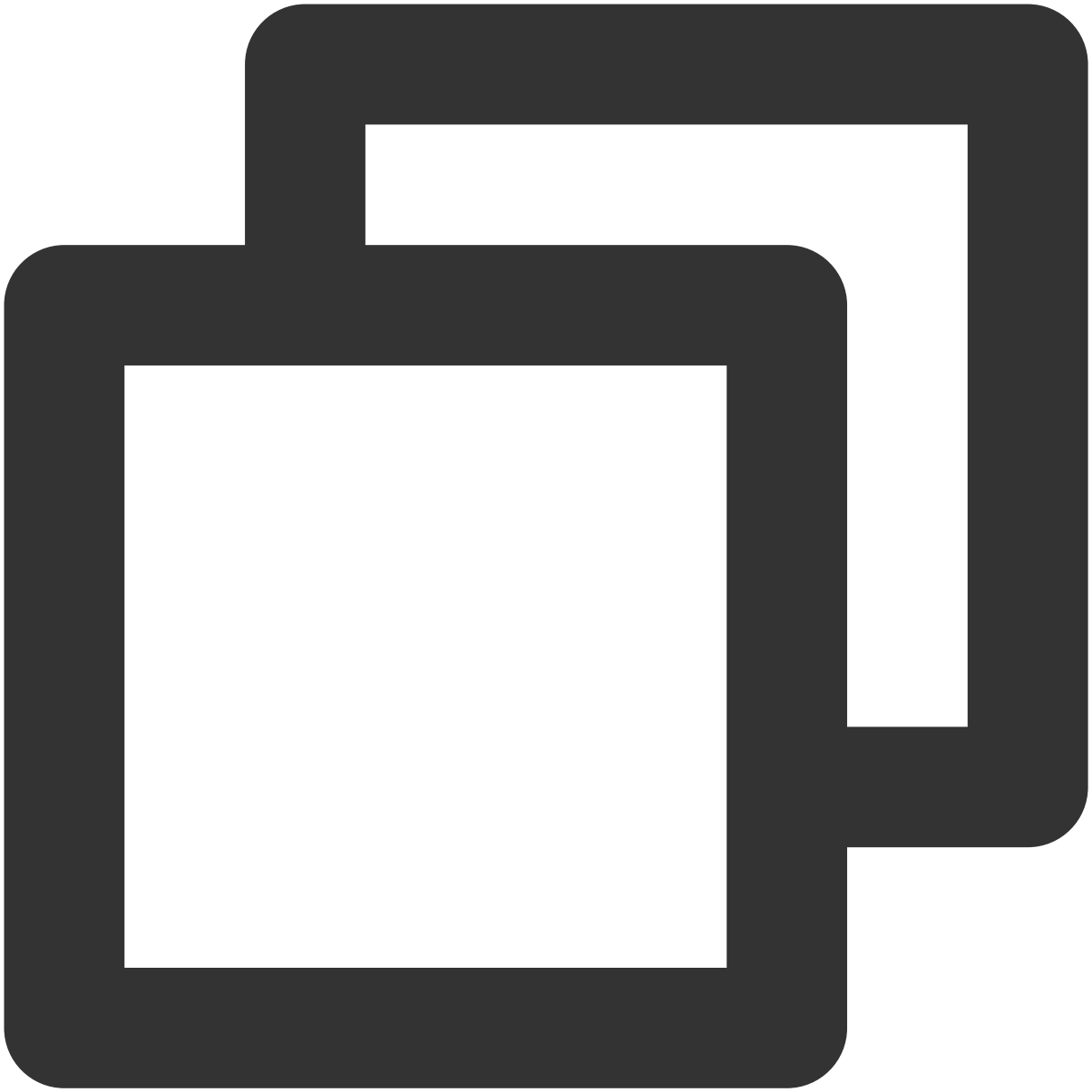
步骤二：建立 dblink 链接



```
select dblink_connect('yunpg1','host=10.10.10.11 port=5432 instanceid=postgres-2123
 dblink_connect
----------------
 OK
(1 row)
```

步骤三：访问外部数据

版权所有：腾讯云计算（北京）有限责任公司



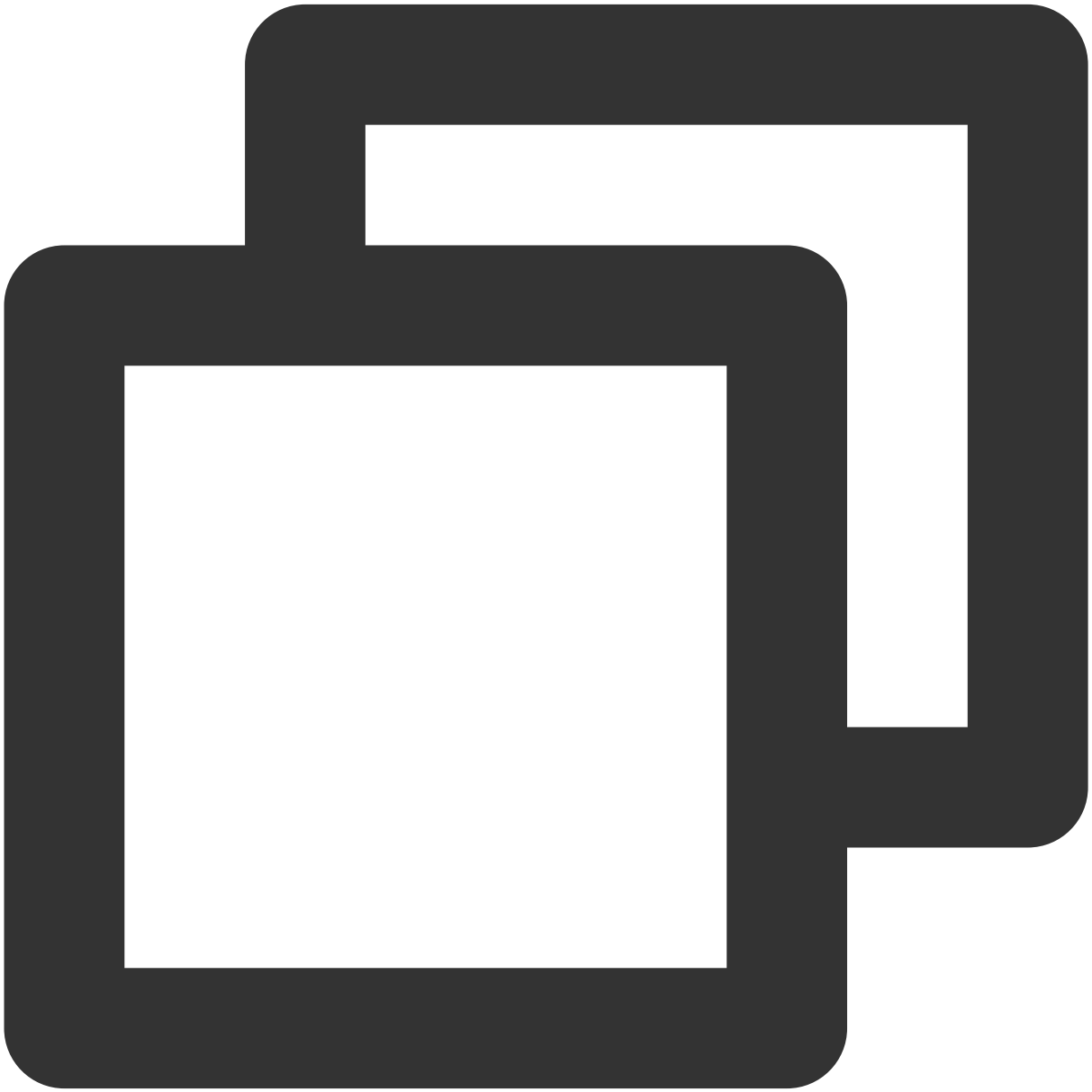
```
postgres=> select * from dblink('yunpg1','select catalog_name,schema_name,schema_ow
    a      |          b           |     c
----------+---------------------+---------
 postgres | pg_toast            | user_00
 postgres | pg_temp_1           | user_00
 postgres | pg_toast_temp_1     | user_00
 postgres | pg_catalog          | user_00
 postgres | public              | user_00
 postgres | information_schema | user_00
(6 rows)
```

参考链接

dblink 官方介绍

使用 mysql_fdw 示例

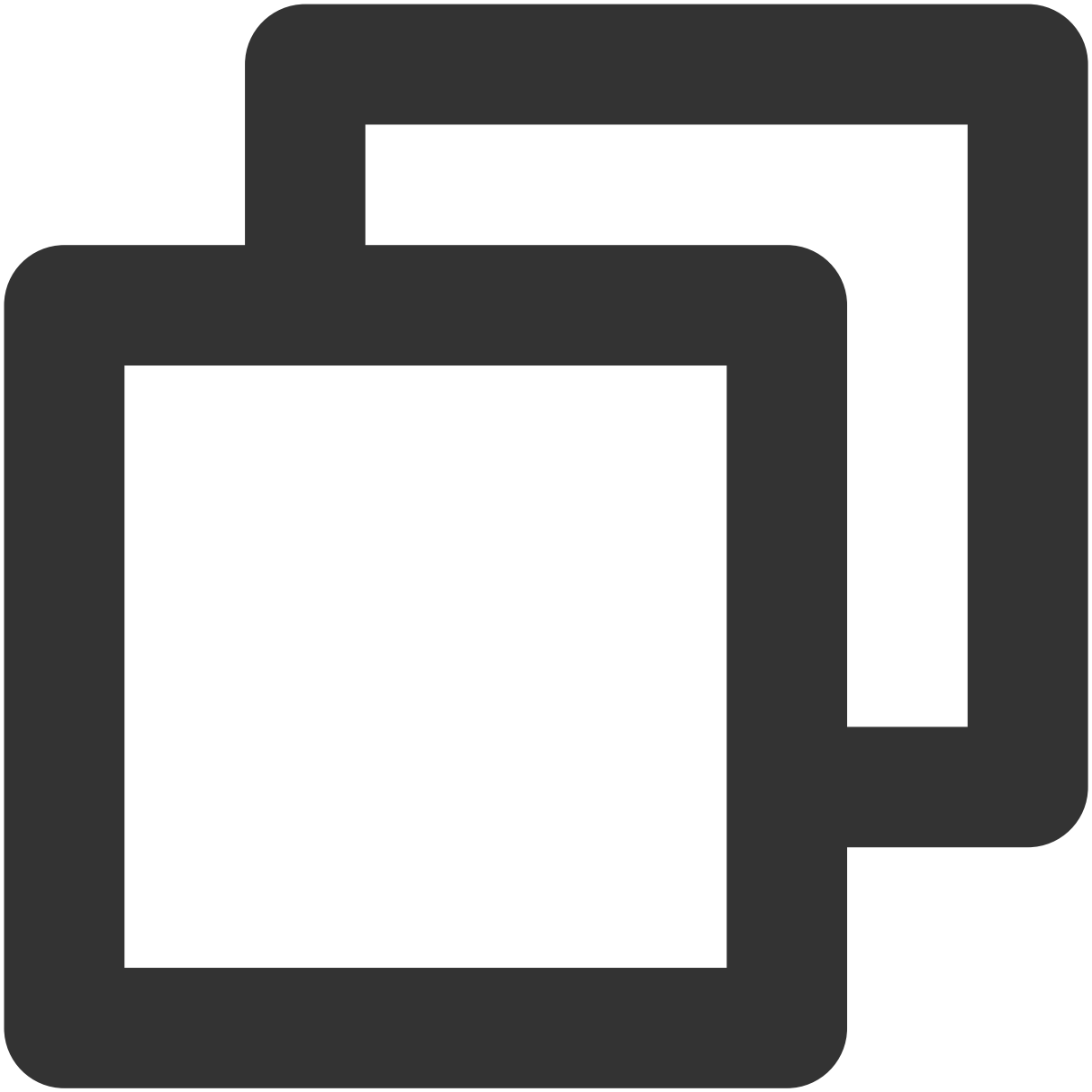
步骤一：创建 mysql_fdw 插件



```
postgres=> create extension mysql_fdw;
CREATE EXTENSION
```

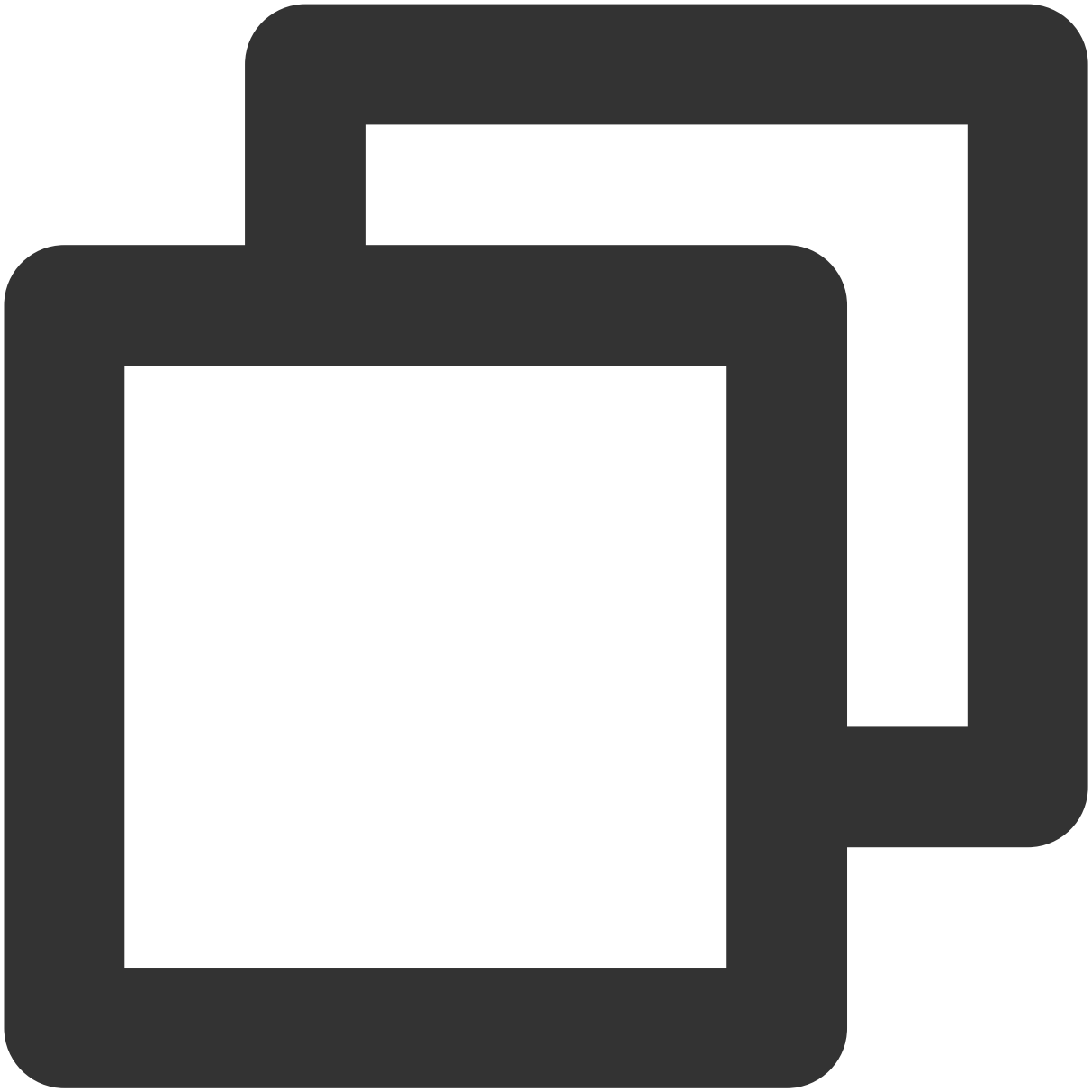
```
postgres=> \\dx;
                                            List of installed extensions
        Name         | Version |   Schema   |                              Descripti
---------------------+---------+------------+-------------------------------------
 dblink              | 1.2     | public     | connect to other PostgreSQL databases
 mysql_fdw           | 1.1     | public     | Foreign data wrapper for querying a My
 pg_stat_log         | 1.0     | public     | track runtime execution statistics of
 pg_stat_statements  | 1.9     | public     | track planning and execution statistic
 plpgsql             | 1.0     | pg_catalog | PL/pgSQL procedural language
(5 rows)
```

步骤二：创建 SERVER



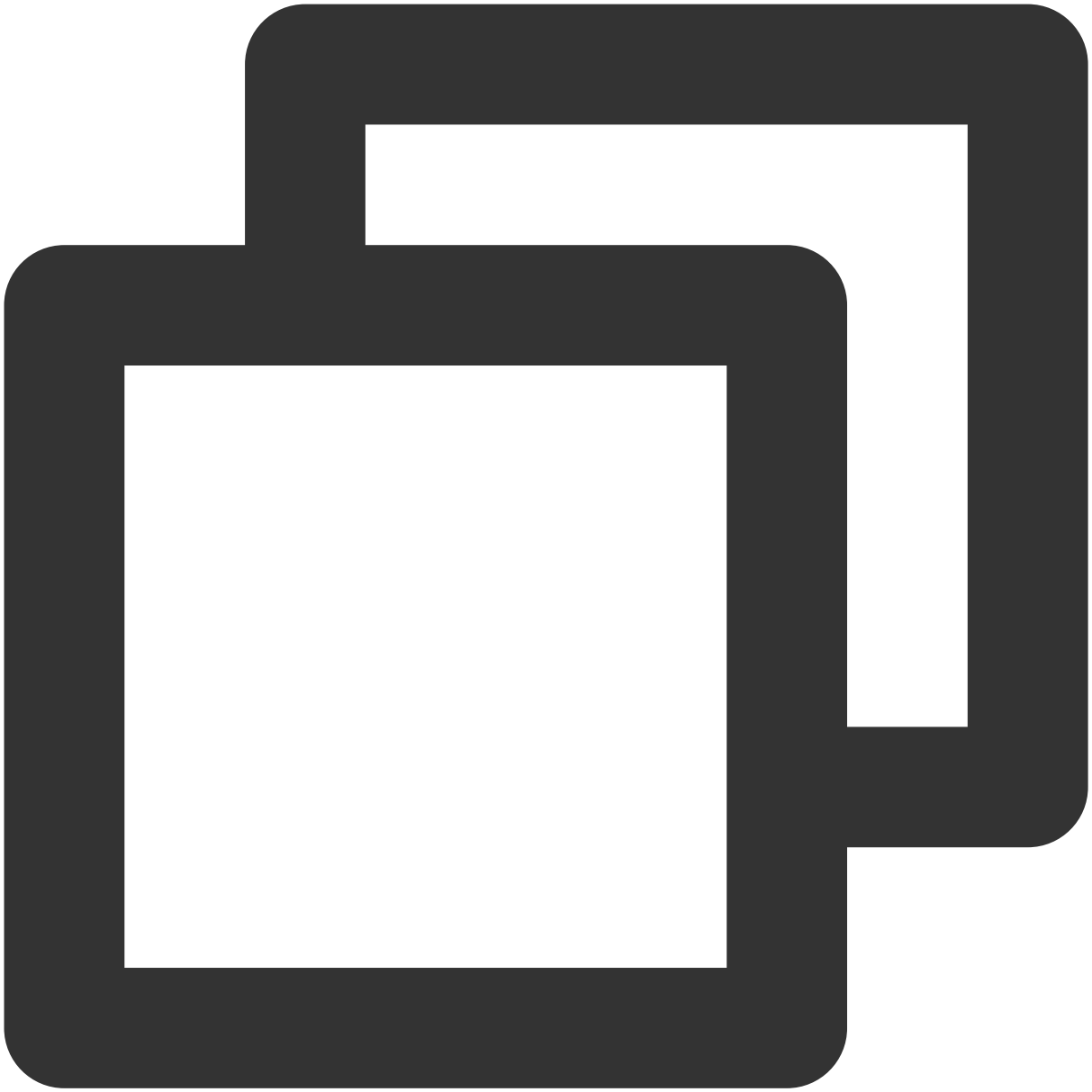
```
postgres=> CREATE SERVER mysql_svr  FOREIGN DATA WRAPPER mysql_fdw OPTIONS (host '1
CREATE SERVER
```

步骤三：创建外部用户映射



```
postgres=> CREATE USER MAPPING FOR PUBLIC SERVER mysql_svr OPTIONS (username 'fdw_u
CREATE USER MAPPING
```

步骤四：访问外部数据



```
postgres=> IMPORT FOREIGN SCHEMA hrdb FROM SERVER mysql_svr INTO public;
```

参考链接

mysql_fdw 官方介绍

使用 cos_fdw 示例

cos_fdw 使用示例请参考文档 通过 cos_fdw 插件支持分级存储能力。

使用注意

目标实例，需要注意以下几点：

1. 需要放开 PostgreSQL 的 hba 限制，允许创建的映射用户（如：user2）以 MD5 方式访问。hba 的修改可参考 PostgreSQL 官方说明。
2. 如果目标实例非 TencentDB 实例，且搭建有热备模式，当主备切换后，需要自行更新 server 连接地址或者重新创建 server。

位图计算 pg_roaringbitmap 插件

最近更新时间：2024-01-22 16:22:57

云数据库 PostgreSQL 提供 pg_roaringbitmap 插件，可以使用位图计算功能，提高查询性能。

前提条件

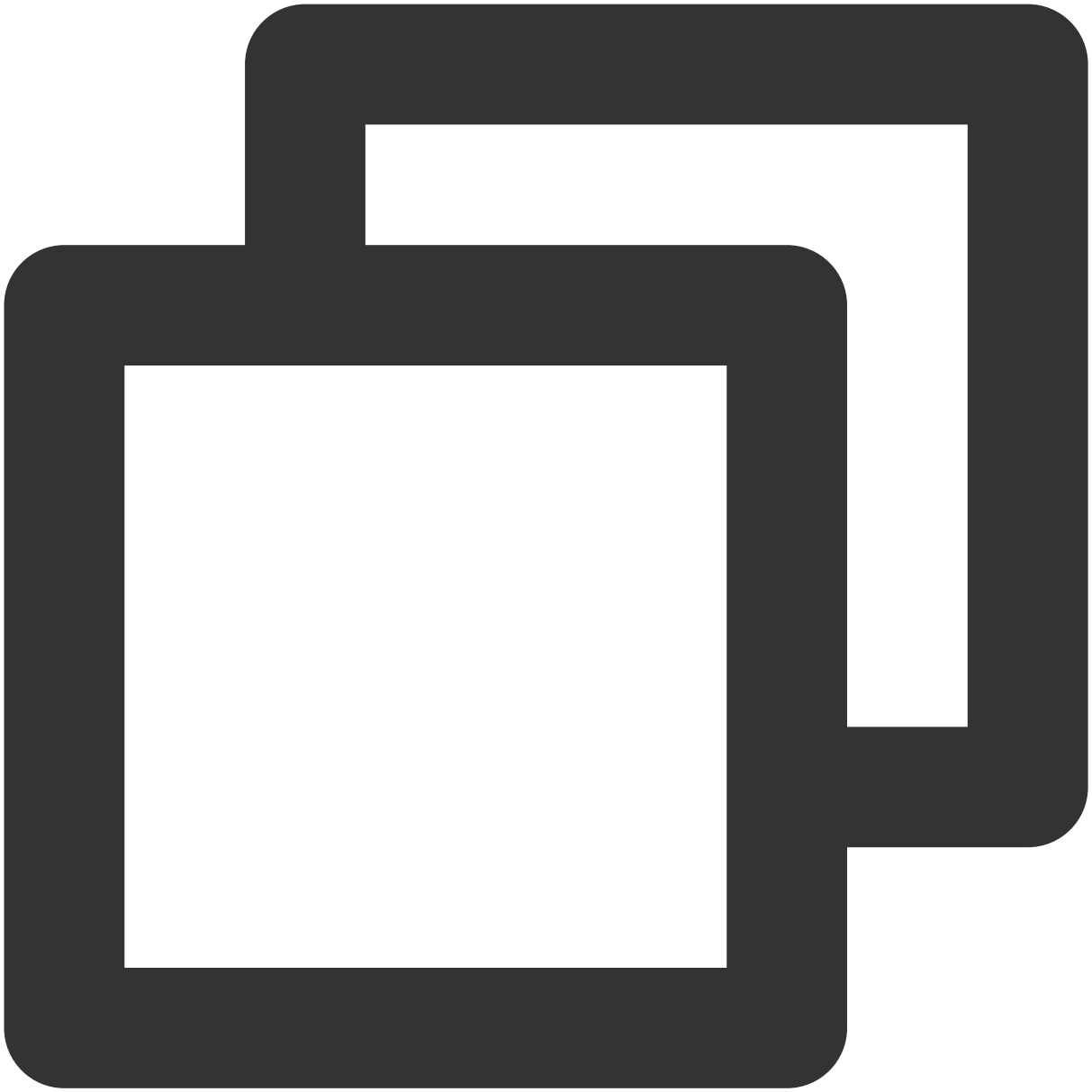
实例为云数据库 PostgreSQL 10、11、12、13、14、15 全新版本。

背景信息

Roaring Bitmap 算法是将32位的 INT 类型数据划分为216个数据块（Chunk），每一个数据块对应整数的高16位，并使用一个容器（Container）来存放一个数值的低16位。
Roaring Bitmap 将这些容器保存在一个动态数组中，作为一级索引。容器使用两种不同的结构：数组容器（Array Container）和位图容器（Bitmap Container）。数组容器存放稀疏的数据，位图容器存放稠密的数据。如果一个容器里面的整数数量小于4096，就用数组容器来存储值。若大于4096，就用位图容器来存储值。
采用这种存储结构，Roaring Bitmap 可以快速检索一个特定的值。在做位图计算（AND、OR、XOR）时，Roaring Bitmap 提供了相应的算法来高效地实现在两种容器之间的运算。使得 Roaring Bitmap 无论在存储和计算性能上都表现优秀。

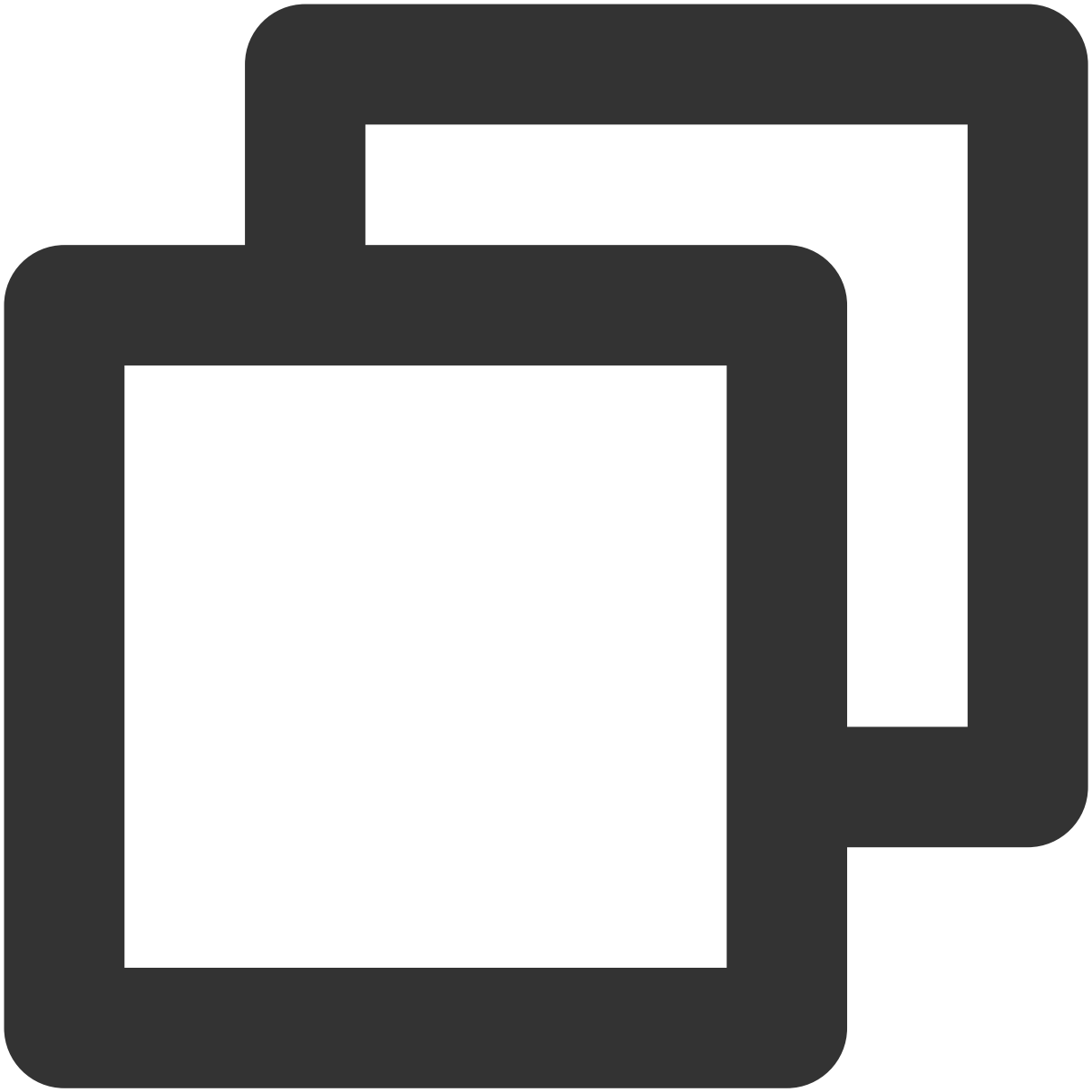
操作步骤

1. 创建插件。示例如下：



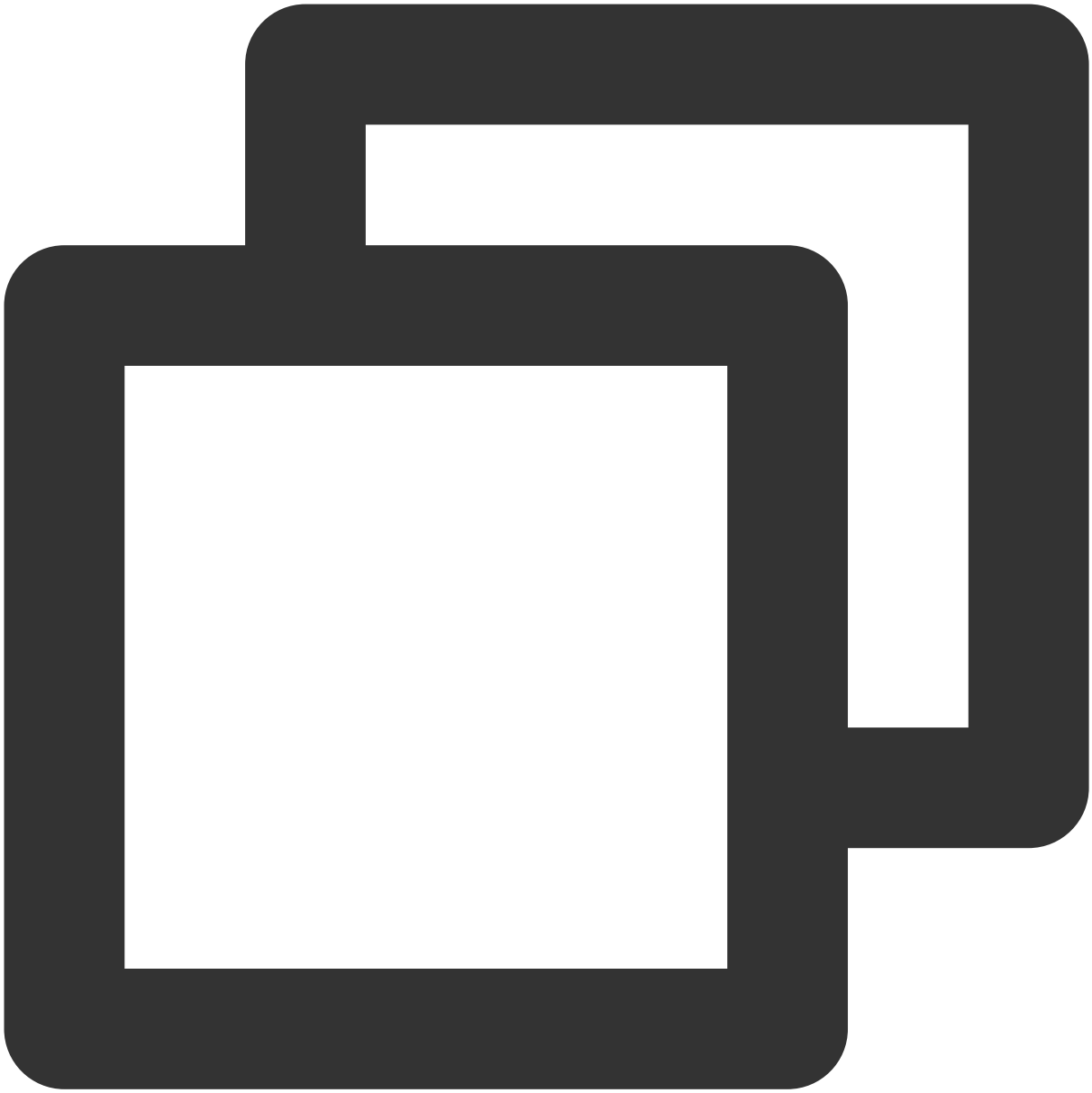
```
CREATE EXTENSION roaringbitmap;
```

2. 创建带有 RoaringBitmap 数据类型的表。示例如下：



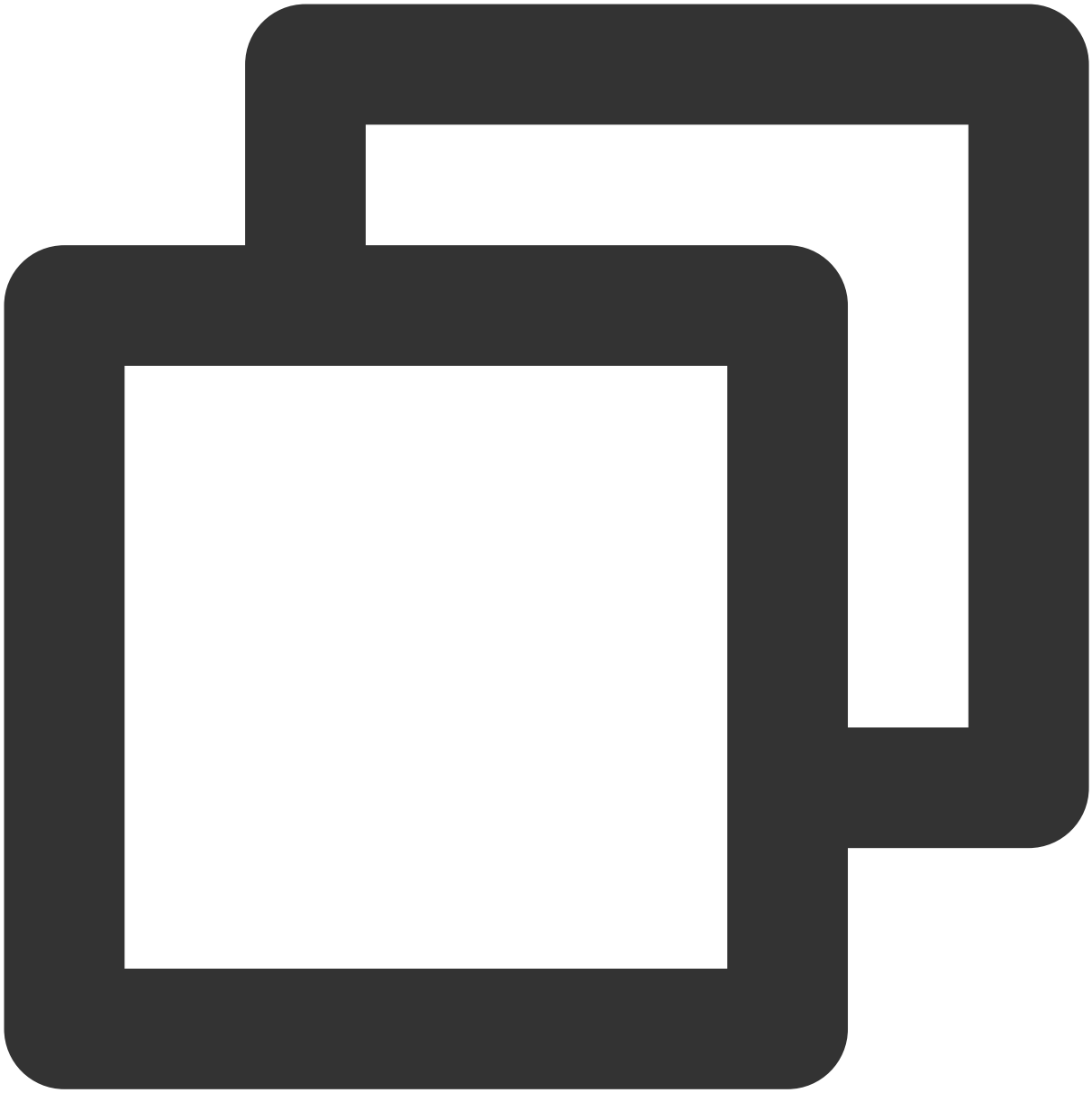
```
CREATE TABLE t1 (id integer, bitmap roaringbitmap);
```

3. 使用 rb_build 函数插入 roaringbitmap 的数据。示例如下：



```
--数组位置对应的 BIT 值为1
INSERT INTO t1 SELECT 1,RB_BUILD(ARRAY[1,2,3,4,5,6,7,8,9,200]);
--将输入的多条记录的值对应位置的 BIT 值设置为1，最后聚合为一个 roaringbitmap
INSERT INTO t1 SELECT 2,RB_BUILD_AGG(e) FROM GENERATE_SERIES(1,100) e;
```

4. 进行 Bitmap 计算（OR、AND、XOR、ANDNOT）。示例如下：

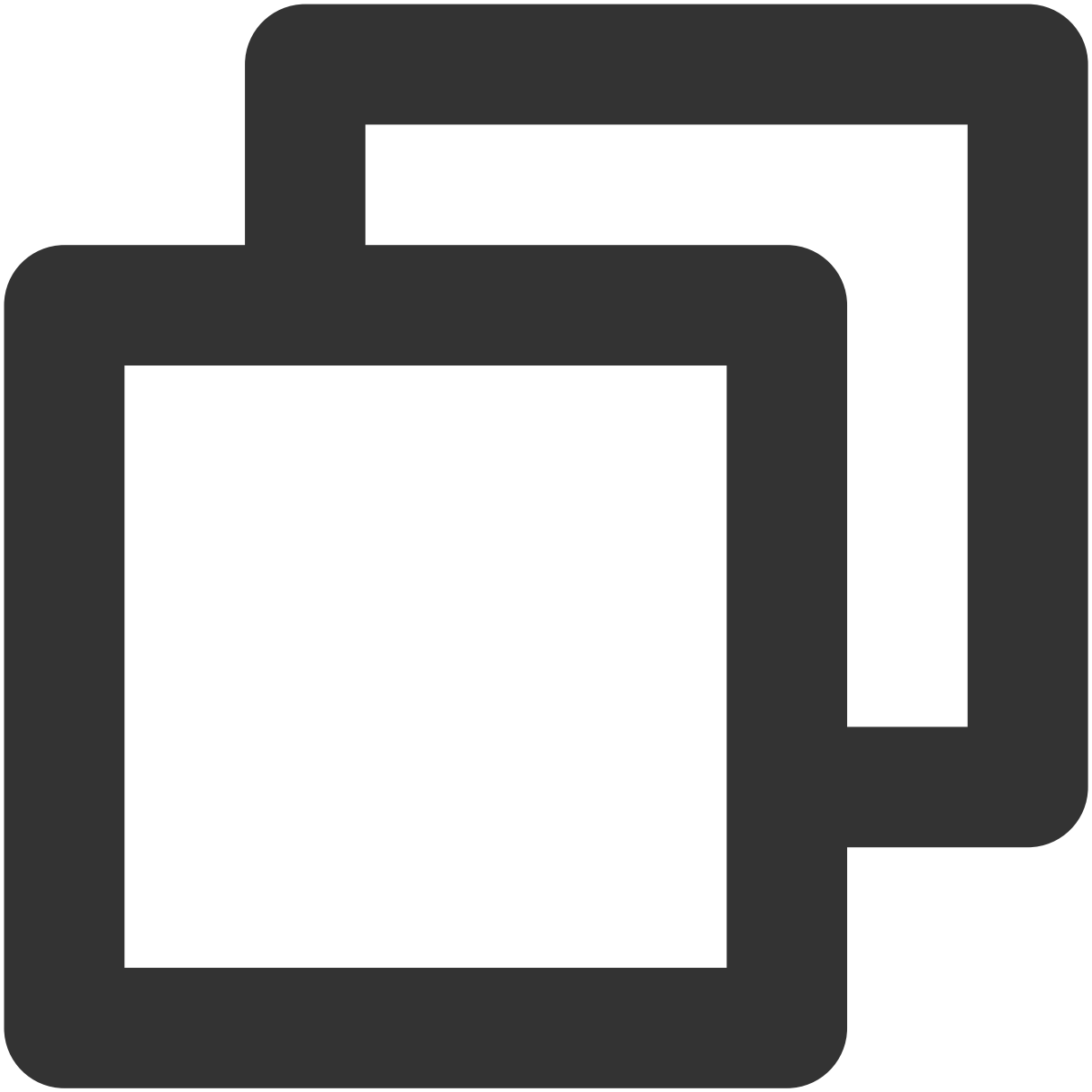


```
--数组位置对应的 BIT 值为1
SELECT RB_OR(a.bitmap,b.bitmap) FROM (SELECT bitmap FROM t1 WHERE id = 1) AS a,(SEL
```

5. 进行 Bitmap 聚合计算（OR、AND、XOR、BUILD），并生成新的 roaringbitmap 类型。示例如下：

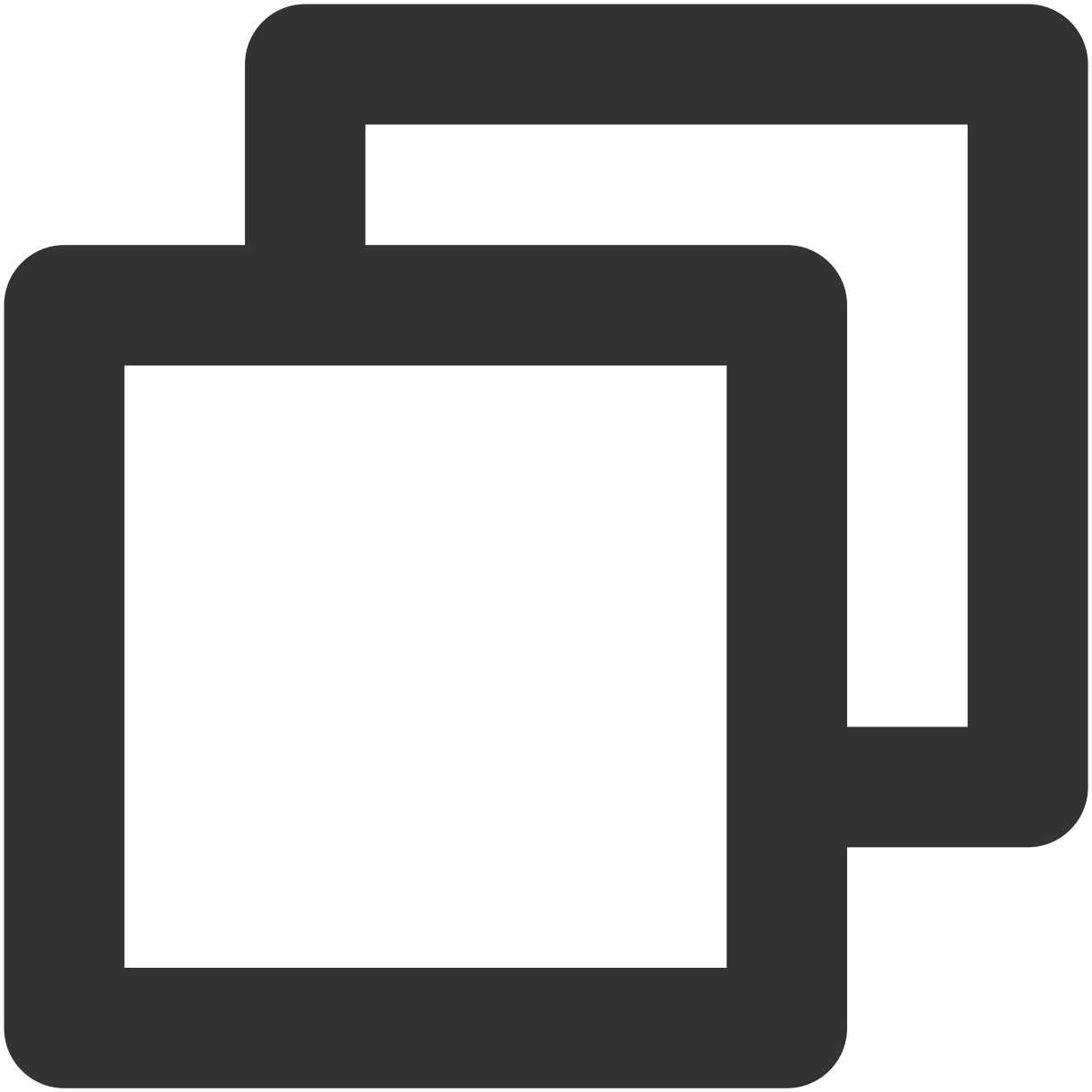
```
SELECT RB_OR_AGG(bitmap) FROM t1;
SELECT RB_AND_AGG(bitmap) FROM t1;
SELECT RB_XOR_AGG(bitmap) FROM t1;
SELECT RB_BUILD_AGG(e) FROM GENERATE_SERIES(1,100) e;
```

6. 统计基数（Cardinality），即统计 roaringbitmap 中包含多少个位置为1的 BIT 位。示例如下：



```
SELECT RB_CARDINALITY(bitmap) FROM t1;
```

7. 从 roaringbitmap 中返回位置为1的 BIT 下标（即位置值）。示例如下：



```
SELECT RB_ITERATE(bitmap) FROM t1 WHERE id = 1;
```

功能函数列表

函数名	输入	输出	描述
rb_build	integer[]	roaringbitmap	Create roaringbitmap from integer array

rb_index	roaringbitmap,integer	bigint	Return the 0-based index of element i this roaringbitmap, or -1 if do not exsits
rb_cardinality	roaringbitmap	bigint	Return cardinality of the roaringbitmap
rb_and_cardinality	roaringbitmap,roaringbitmap	bigint	Return cardinality of the AND of two roaringbitmaps
rb_xor_cardinality	roaringbitmap,roaringbitmap	bigint	Return cardinality of the XOR of two roaringbitmaps
rb_andnot_cardinality	roaringbitmap,roaringbitmap	bigint	Return cardinality of the ANDNOT of two roaringbitmaps
rb_is_empty	roaringbitmap	boolean	Check if roaringbitmap is empty.
rb_fill	roaringbitmap,range_start bigint,range_end bigint	roaringbitmap	Fill the specified range (not include the range_end)
rb_clear	roaringbitmap,range_start bigint,range_end bigint	roaringbitmap	Clear the specified range (not include the range_end)
rb_flip	roaringbitmap,range_start bigint,range_end bigint	roaringbitmap	Negative the specified range (not include the range_end)
rb_range	roaringbitmap,range_start bigint,range_end bigint	roaringbitmap	Return new set with specified range (not include the range_end)
rb_range_cardinality	roaringbitmap,range_start bigint,range_end bigint	bigint	Return the cardinality of specified range (not include the range_end)
rb_min	roaringbitmap	integer	Return the smallest offset in roaringbitmap. Return NULL if the bitmap is empty
rb_max	roaringbitmap	integer	Return the greatest offset in roaringbitmap. Return NULL if the bitmap is empty
rb_rank	roaringbitmap,integer	bigint	Return the number of elements that are smaller or equal to the specified offset
rb_jaccard_dist	roaringbitmap,roaringbitmap	double precision	Return the jaccard distance(or the Jaccard similarity coefficient) of two bitmaps

rb_select	roaringbitmap,bitset_limit bigint,bitset_offset bigint=0,reverse boolean=false,range_start bigint=0,range_end bigint=4294967296	roaringbitmap	Return subset [bitset_offset,bitset_offset+bitset_limit of bitmap between range [range_start,range_end)
rb_to_array	roaringbitmap	integer[]	Convert roaringbitmap to integer array
rb_iterate	roaringbitmap	SET of integer	Return set of integer from a roaringbitmap data.

聚合函数列表

聚合函数名	输入	输出	描述	示例
rb_build_agg	integer	roaringbitmap	Build a roaringbitmap from a integer set	`select rb_build_agg(id (values (1),(2),(3)) t(`
rb_or_agg	roaringbitmap	roaringbitmap	AND Aggregate calculations from a roaringbitmap set	`select rb_or_agg(bitma from (values (roaringbitmap('{1,2,3} (roaringbitmap('{2,3,4} t(bitmap)`
rb_and_agg	roaringbitmap	roaringbitmap	AND Aggregate calculations from a roaringbitmap set	`select rb_and_agg(bitm from (values (roaringbitmap('{1,2,3} (roaringbitmap('{2,3,4} t(bitmap)`
rb_xor_agg	roaringbitmap	roaringbitmap	XOR Aggregate calculations from a roaringbitmap set	`select rb_xor_agg(bitm from (values (roaringbitmap('{1,2,3} (roaringbitmap('{2,3,4} t(bitmap)`
rb_or_cardinality_agg	roaringbitmap	bigint	OR	`select`

			Aggregate calculations from a roaringbitmap set, return cardinality.	`rb_or_cardinality_agg(b` `from (values` `(roaringbitmap('{1,2,3}` `(roaringbitmap('{2,3,4}` `t(bitmap)`
rb_and_cardinality_agg	roaringbitmap	bigint	AND Aggregate calculations from a roaringbitmap set, return cardinality	`select` `rb_and_cardinality_agg(` `from (values` `(roaringbitmap('{1,2,3}` `(roaringbitmap('{2,3,4}` `t(bitmap)`
rb_xor_cardinality_agg	roaringbitmap	bigint	XOR Aggregate calculations from a roaringbitmap set, return cardinality	`select` `rb_xor_cardinality_agg(` `from (values` `(roaringbitmap('{1,2,3}` `(roaringbitmap('{2,3,4}` `t(bitmap)`

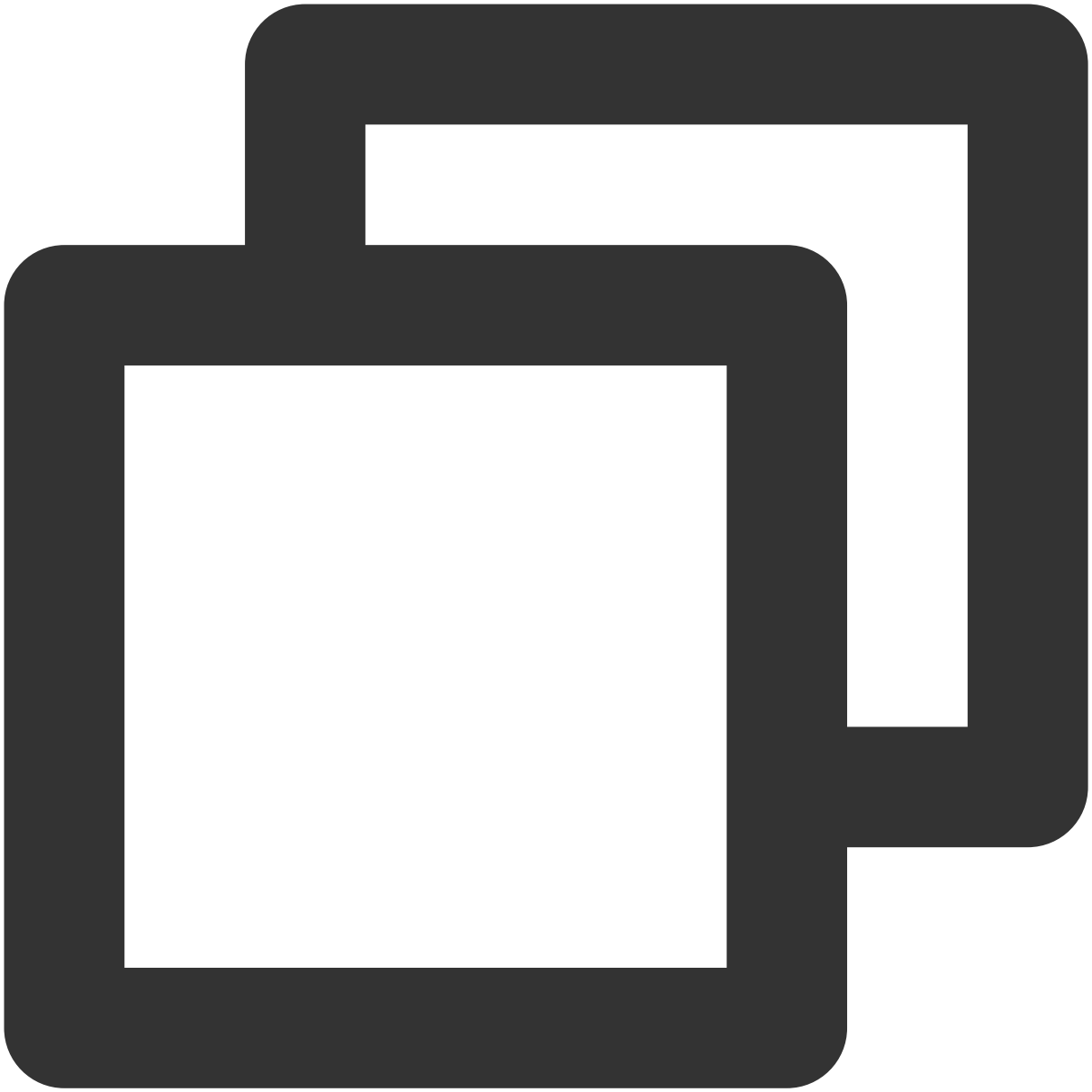
定时任务 pg_cron 插件

最近更新时间：2024-01-22 16:22:57

pg_cron 是一个简单的基于 cron 的 PostgreSQL（10或更高版本）任务调度器，作为扩展在数据库中运行。它使用与常规 cron 相同的语法，允许您直接从数据库定时调度并执行数据库命令。
本文为您介绍 PostgreSQL pg_cron 插件的使用方法。

启用 pg_cron 扩展

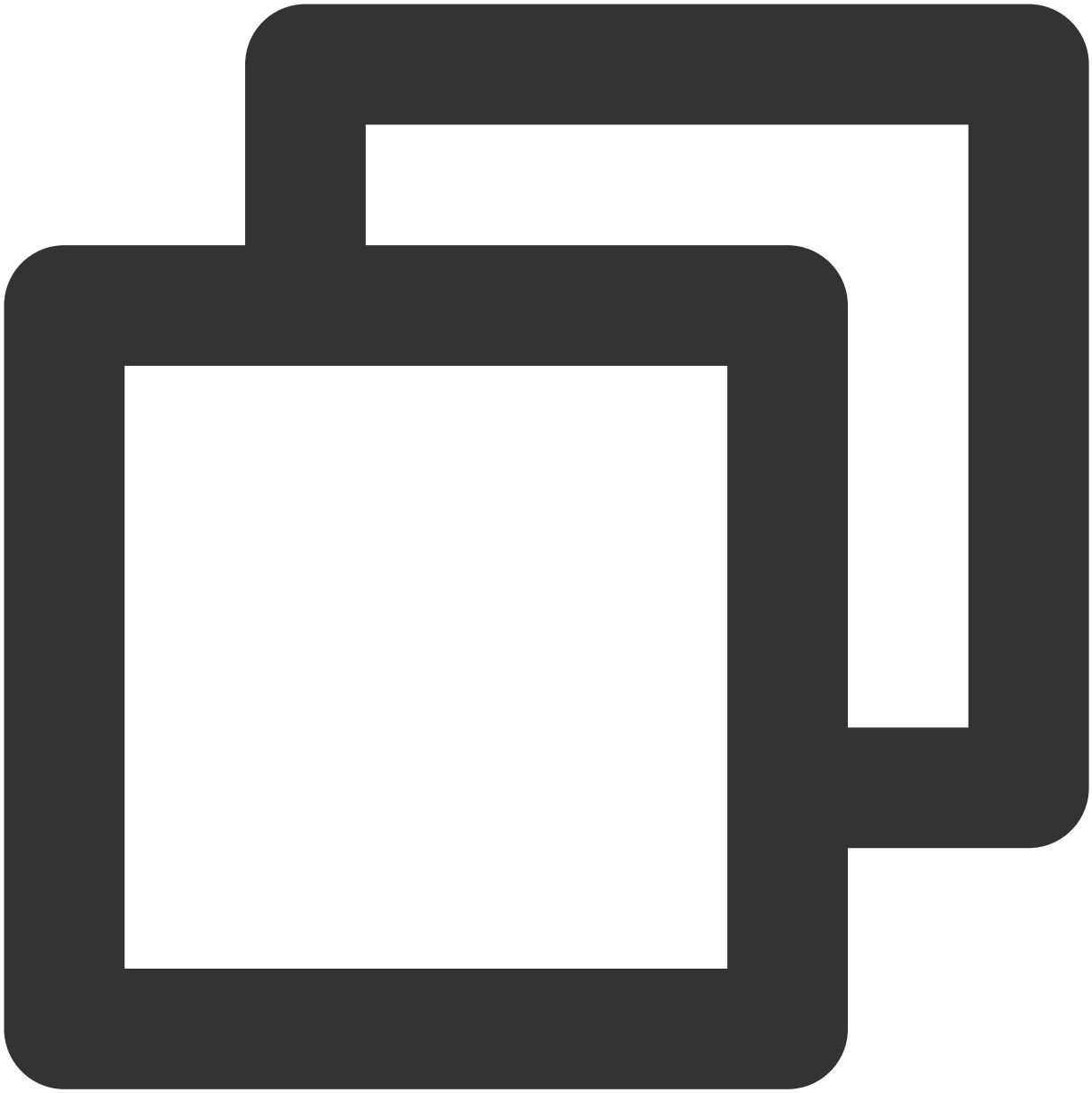
1. 若需使用 pg_cron，请先 提交工单，将 pg_cron 添加至数据库 shared_preload_libraries 参数中。修改此参数，需要重启实例，请选择一个业务空闲期去进行操作。
2. 完成参数修改后，请进入 postgres 库中执行，使用管理员账户运行以下命令：



```
CREATE EXTENSION pg_cron;
```

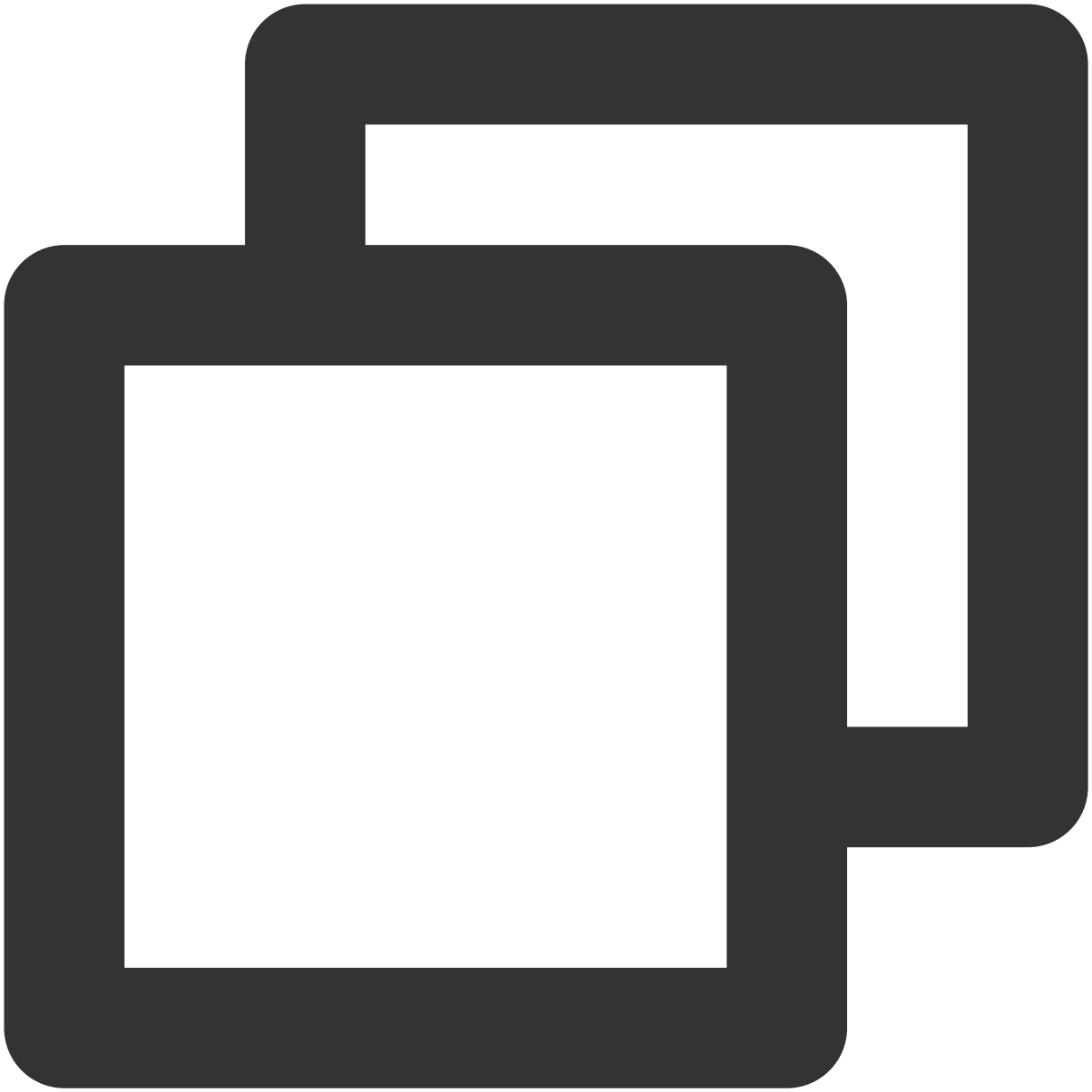
3. 当前 pg_cron 配置仅能在 postgres 库中执行计划任务，若需要在其他 database 中运行计划任务，请参见 对 postgres 以外的数据库设置定时任务。

4. 默认 pg_cron 创建完成后，其配置数据以及任务执行只能由管理员用户进行设置。若需要其他用户进行 pg_cron 设置或者执行，则需要向其他用户授予 cron 元数据库的权限，请运行以下命令。



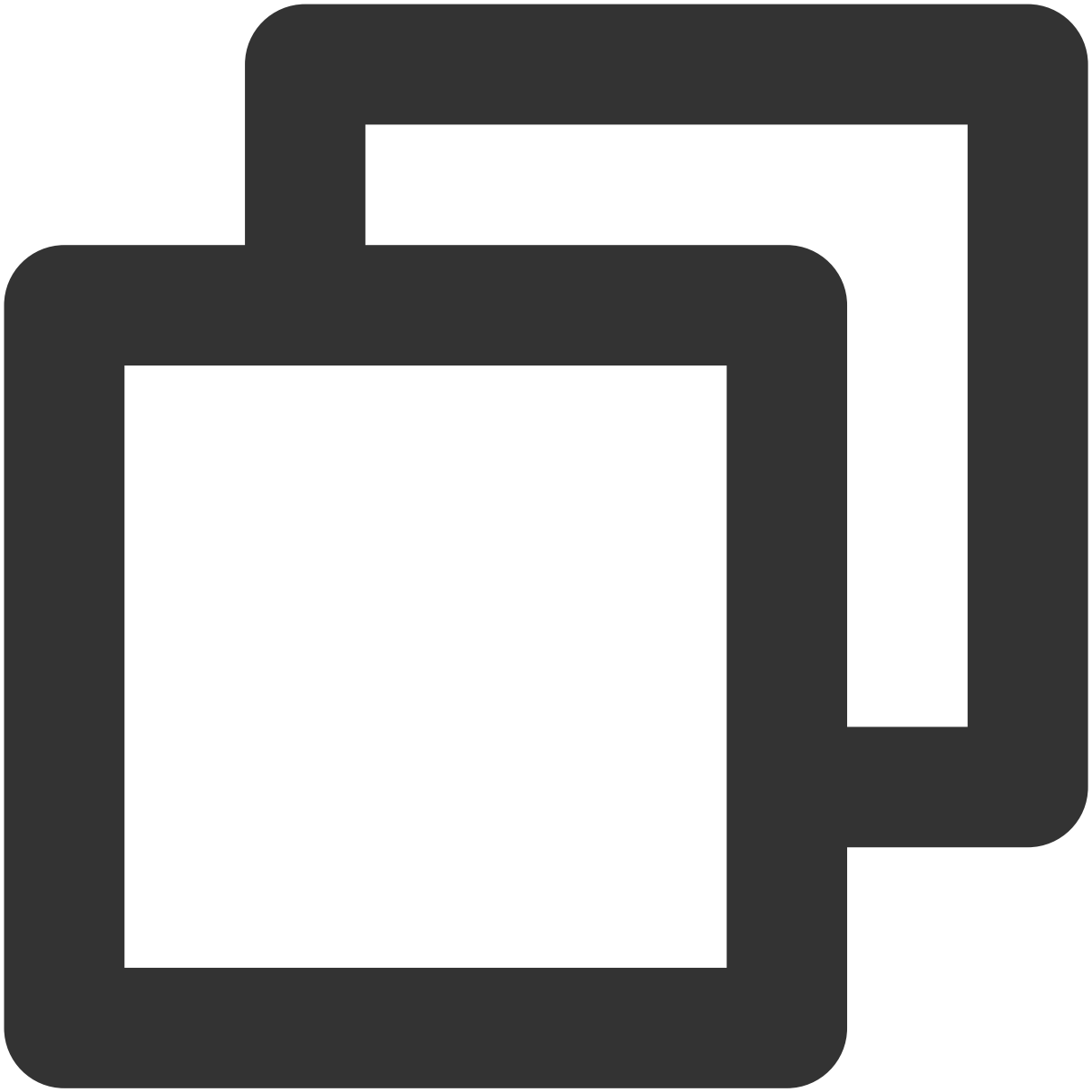
```
postgres=> GRANT USAGE ON SCHEMA cron TO other-user;
```

此权限为 other-user 提供了对 cron 元数据的访问权限以计划和取消计划 cron 任务。但是，为了成功运行 cron 任务，还需要具有对 cron 任务中对象的访问权限。如果用户没有权限，则任务将失败，postgresql.log 中将显示错误。如下实例，用户没有访问 pgbench_accounts 表的权限。



```
2020-12-08 16:41:00 UTC::@:[30647]:ERROR: permission denied for table pgbench_accou
2020-12-08 16:41:00 UTC::@:[30647]:STATEMENT: update pgbench_accounts set abalance
2020-12-08 16:41:00 UTC::@:[27071]:LOG: background worker "pg_cron" (PID 30647) exi
```

cron.job_run_details 表中的其他消息如下所示：



```
postgres=> select jobid, username, status, return_message, start_time from cron.job
jobid |  username   | status |                    return_message                    |
-------+------------+--------+-------------------------------------------------------
143 | unprivuser | failed | ERROR: permission denied for table pgbench_accounts | 2
143 | unprivuser | failed | ERROR: permission denied for table pgbench_accounts | 2
143 | unprivuser | failed | ERROR: permission denied for table pgbench_accounts | 2
143 | unprivuser | failed | ERROR: permission denied for table pgbench_accounts | 2
143 | unprivuser | failed | ERROR: permission denied for table pgbench_accounts | 2
(5 rows)
```

pg_cron 计划任务配置描述

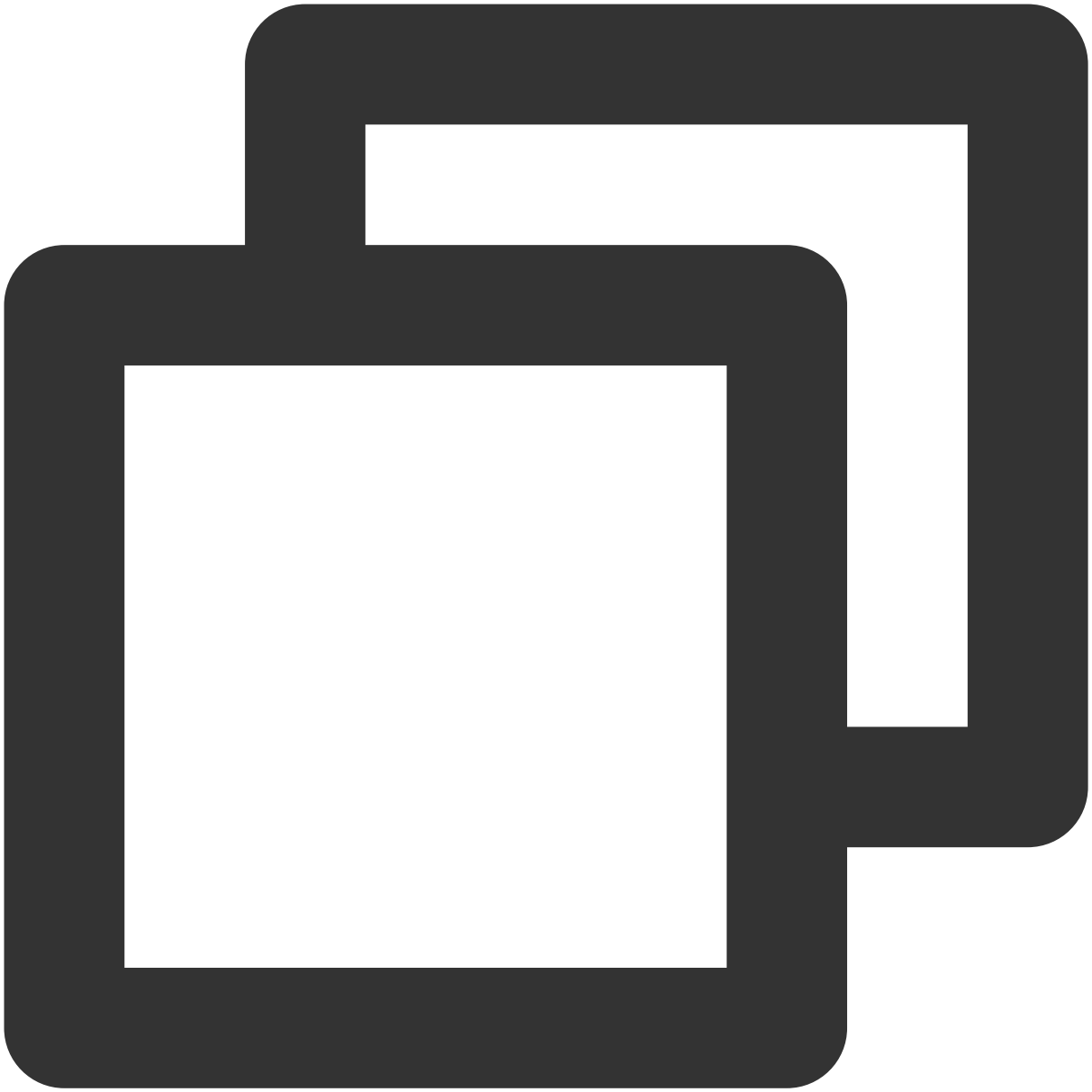
pg_cron 提供三个主要操作：增加任务项、删除任务项、查看任务信息。

cron.schedule() 函数

此函数计划 cron 任务。任务最初是在默认 postgres 数据库中计划的。该函数返回一个表示任务标识符的 bigint 值。要计划任务在 PostgreSQL 数据库实例的其他数据库中运行，请参阅 对 postgres 以外的数据库设置定时任务 中的示例。

该函数有两种语法格式。

语法



```
cron.schedule (job_name,
    schedule,
    command
);

cron.schedule (schedule,
    command
);
```

参数

参数	描述
job_name	cron 任务的名字，可为空不设置。
schedule	表示 cron 任务时间表的文本。格式是标准 cron 格式。
command	要运行的命令的文本。

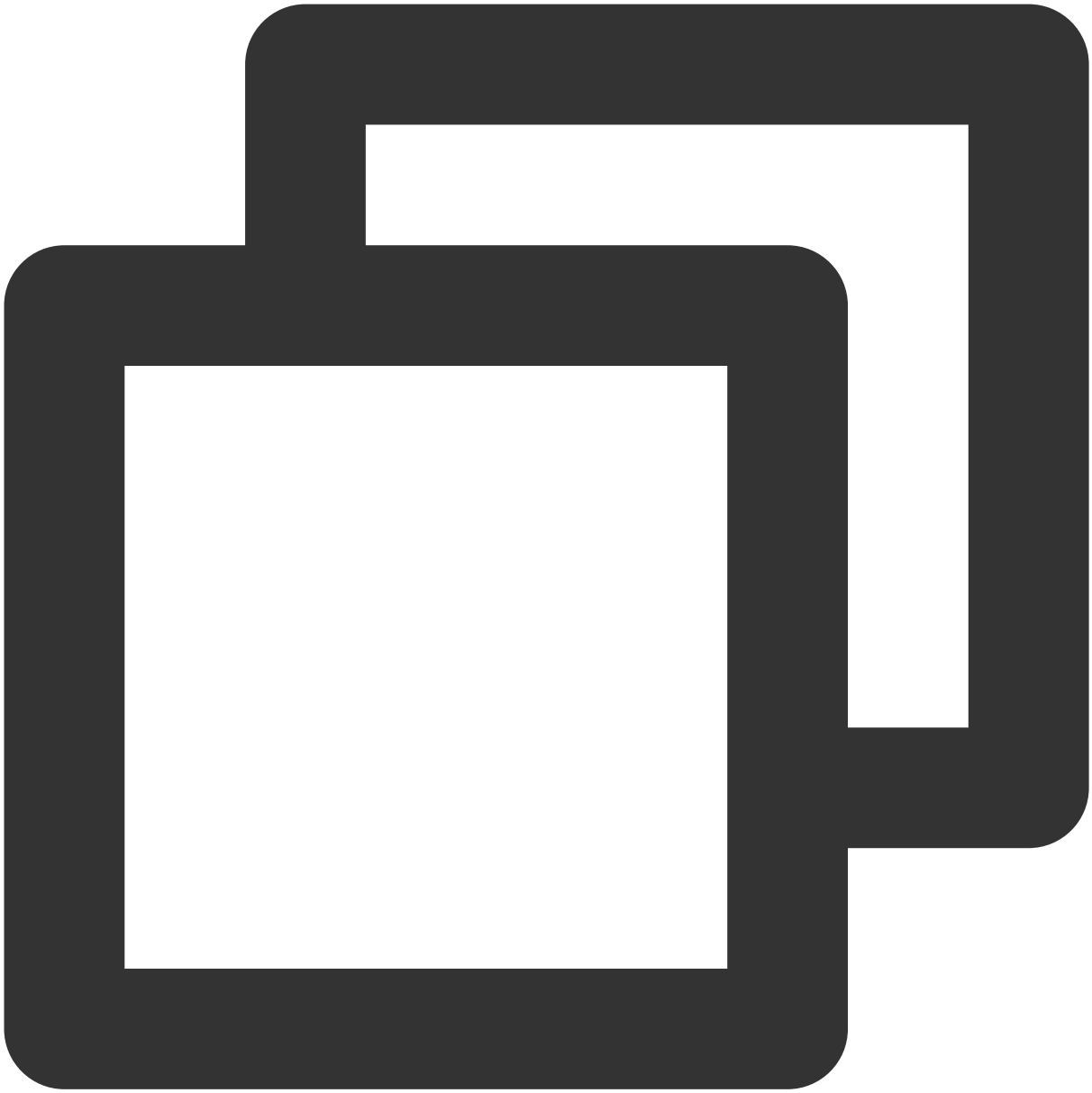
示例

```
postgres=> SELECT cron.schedule ('test','0 10 * * *', 'VACUUM pgbench_history');
```

```
 schedule
----------
      145
(1 row)

postgres=> SELECT cron.schedule ('0 15 * * *', 'VACUUM pgbench_accounts');
 schedule
----------
      146
(1 row)
```

schedule 使用标准的 cron 语法，其中 * 表示“每个该时间运行”，特定数字表示“仅在这个数字时运行”。



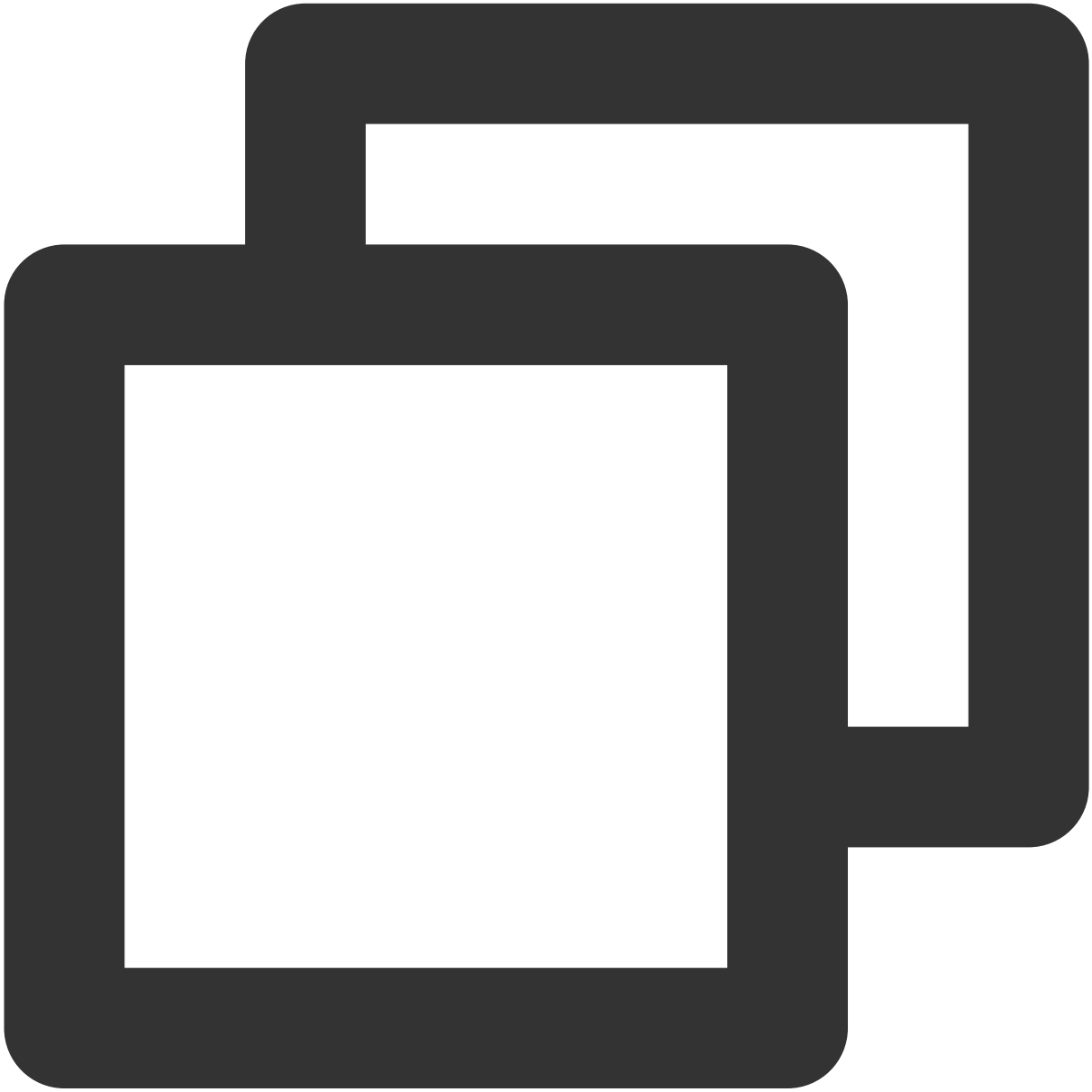
```
# 格式是：分 时 日 月 星期
# week (0 - 6) = sun,mon,tue,wed,thu,fri,sat
# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,...,sat
# |  |  |  |  |
# *  *  *  *  *
```

cron.unschedule() 函数

此函数删除 cron 任务。您可以传入 job_name 或 job_id。请确保您是当前 job_id 所对应的策略的拥有者。该函数返回一个布尔值，指示成功或失败。

该函数使用以下语法格式。

语法



```
cron.unschedule (job_id);

cron.unschedule (job_name);
```

参数

参数	描述
job_id	计划 cron 任务时从 cron.schedule 函数返回的任务标识符。
job_name	使用该 cron.schedule 函数计划的 cron 任务的名称。

示例

```
postgres=> select cron.unschedule(108);
```

```
 unschedule
------------
 t
(1 row)

postgres=> select cron.unschedule('test');
 unschedule
------------
 t
(1 row)
```

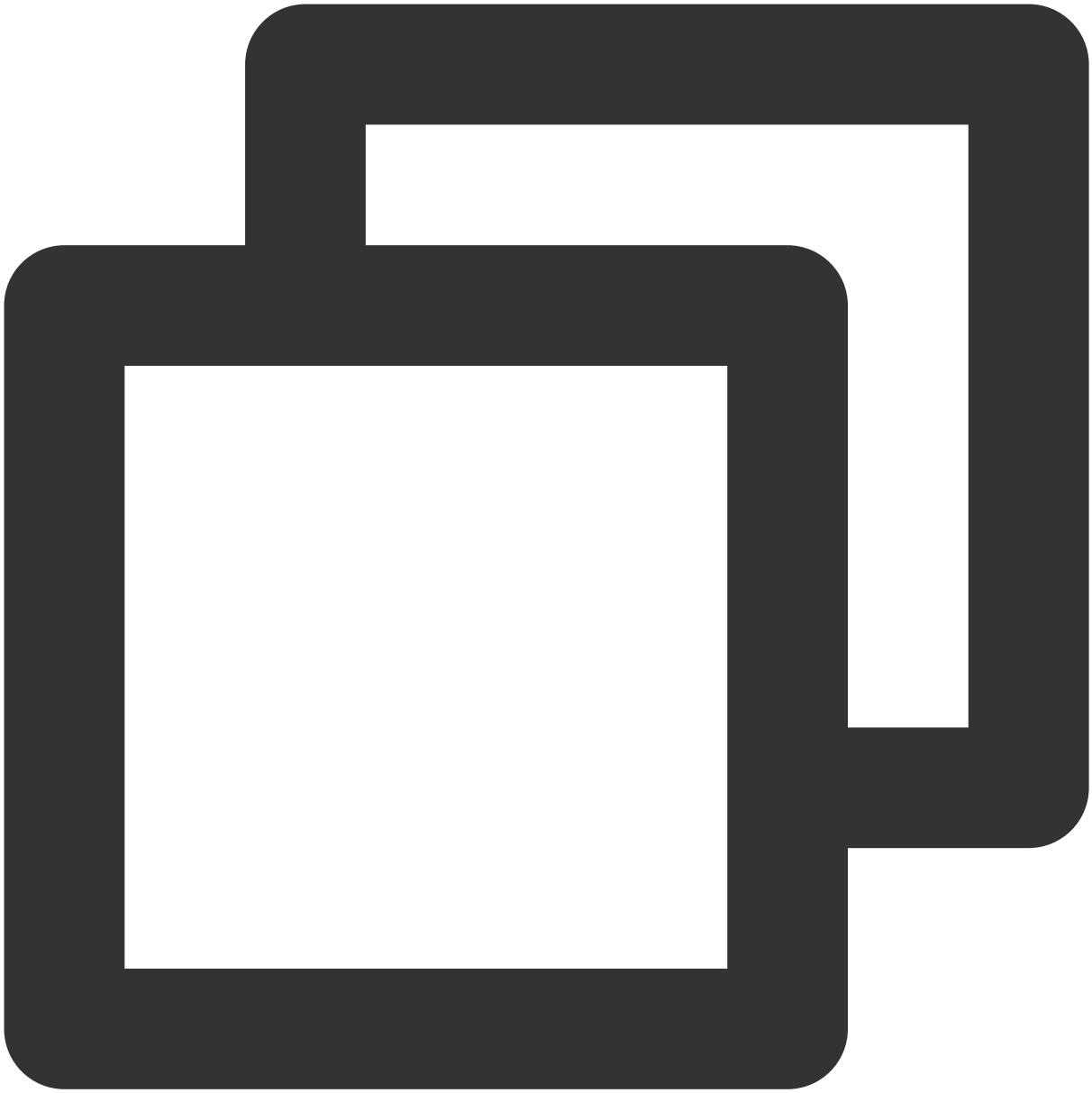
pg_cron 表

将以下各表用于计划任务和记录任务完成的方式。

表	描述
cron.job	包含有关每个计划任务的元数据。与此表的大多数交互应使用 cron.schedule 和 cron.unschedule 函数完成。 注意：不建议直接向此表授予更新或插入权限。
cron.job_run_details	包含过去运行的计划任务的历史信息。这对于调查运行的任务的状态、返回消息以及开始和结束时间非常有用。 为了防止此表无限增长，请定期清除此表。

设置 pg_cron 定时任务

1. 如果您可能希望在选择的时间计划对特定表执行 vacuum 操作，则可以使用 cron.schedule 函数设置定时任务，如每天22:00 (GMT) 在特定表上使用 VACUUM FREEZE。执行设置语句所返回的数字是指当前任务的任务编号。

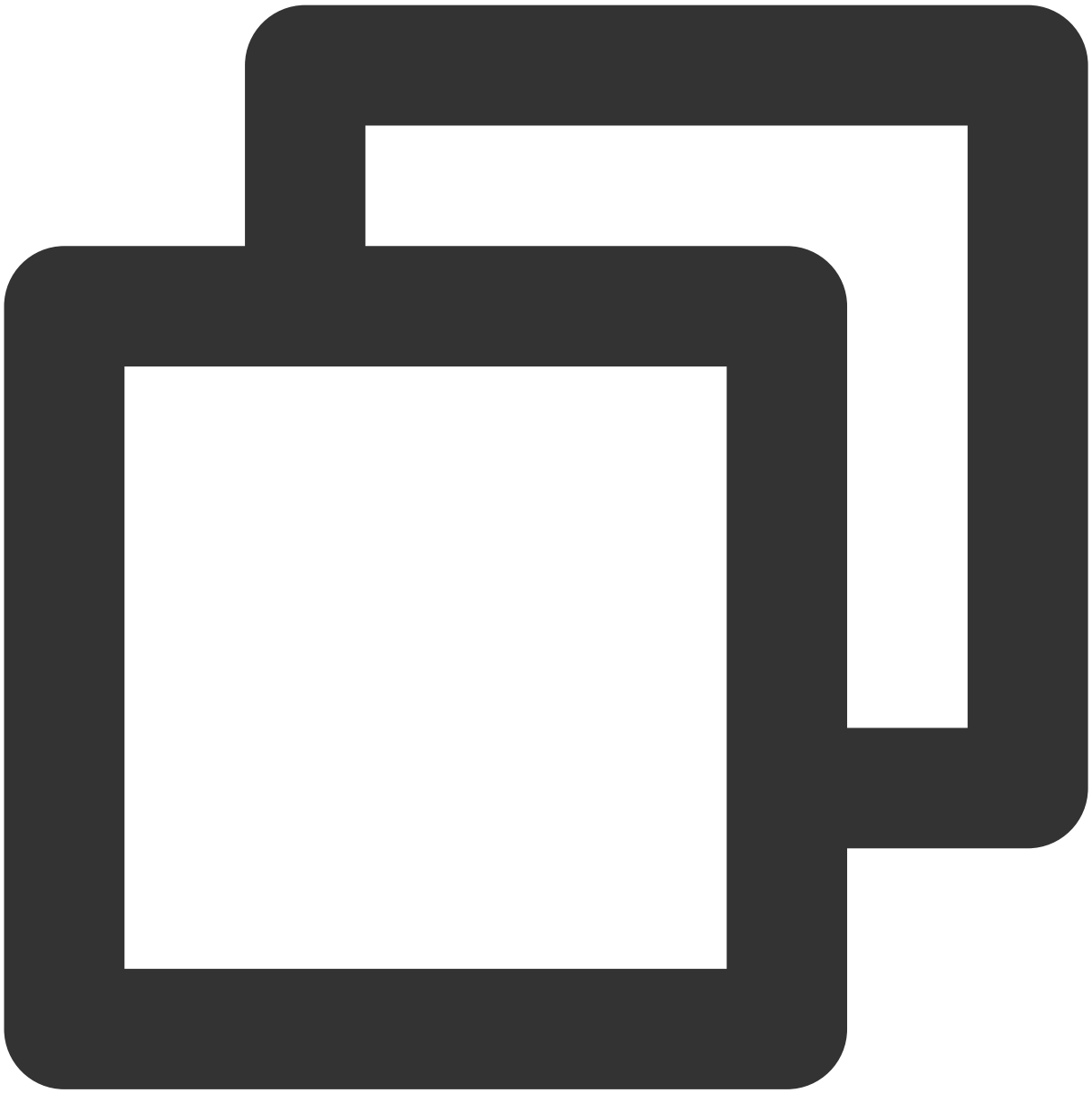


```
SELECT cron.schedule('manual vacuum', '0 22 * * *', 'VACUUM FREEZE pgbench_accounts
 schedule
----------
1
(1 row)
```

2. 其中函数存在3个入参，第一个入参为任务名（字符型），第二个为 cron 定时语法，第三个为具体的执行 SQL。

查看 pg_cron 定时任务

在设置了定时任务后，可通过 cron.job 表查看已经配置了的定时任务，如执行以下语句。

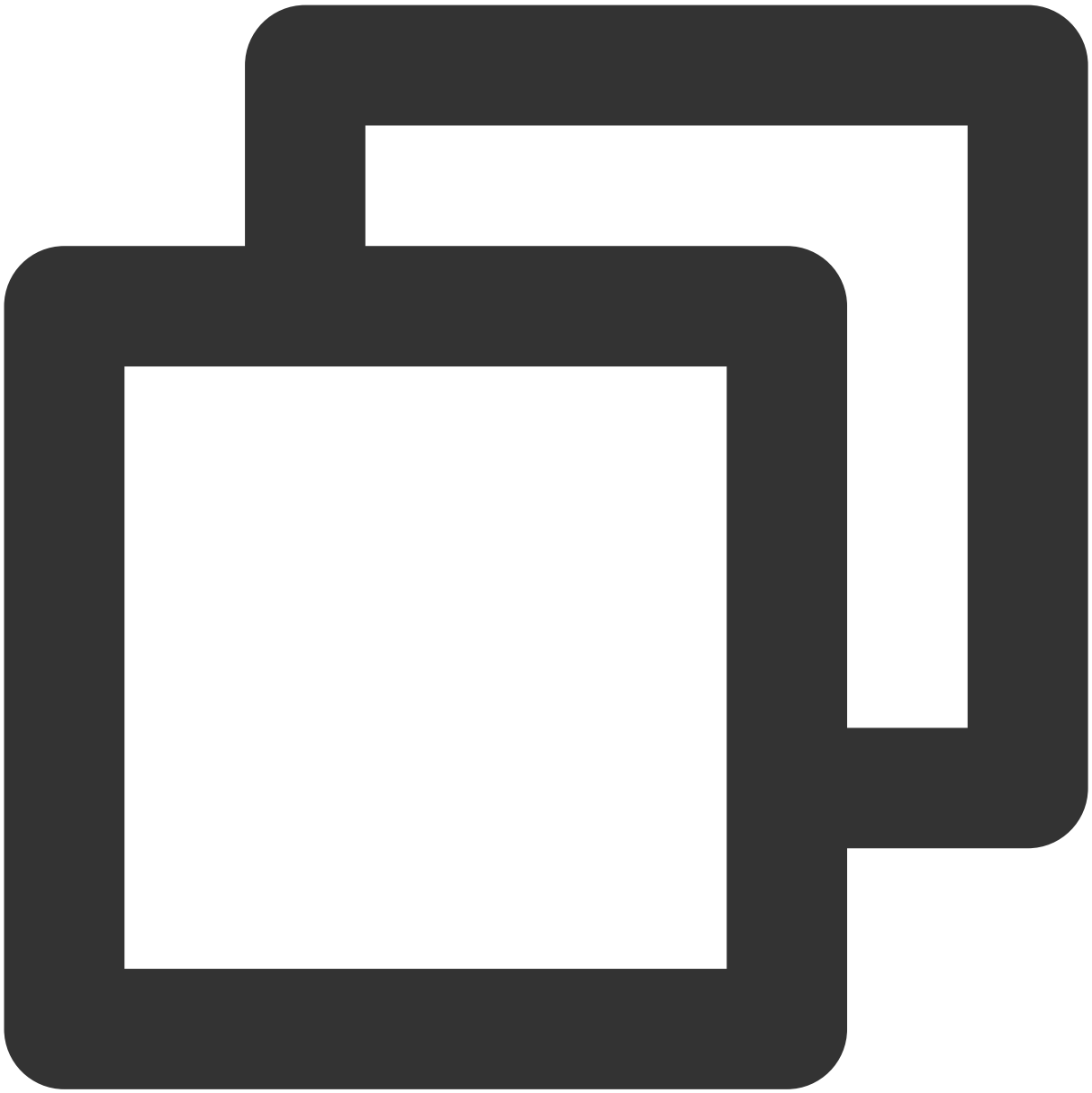


```
SELECT * FROM cron.job;

jobid | schedule    |  command   | nodename   | nodeport | database | username | activ
-------+------------+-----------+-----------+----------+----------+----------+-----
    1 | 0 22 * * * |   VACUUM ... | localhost |     5432 | postgres | test     | t
```

删除 pg_cron 定时任务

如果某一条定时任务无需使用，则需要将此定时任务删除。执行下列语句可将定时任务删除。

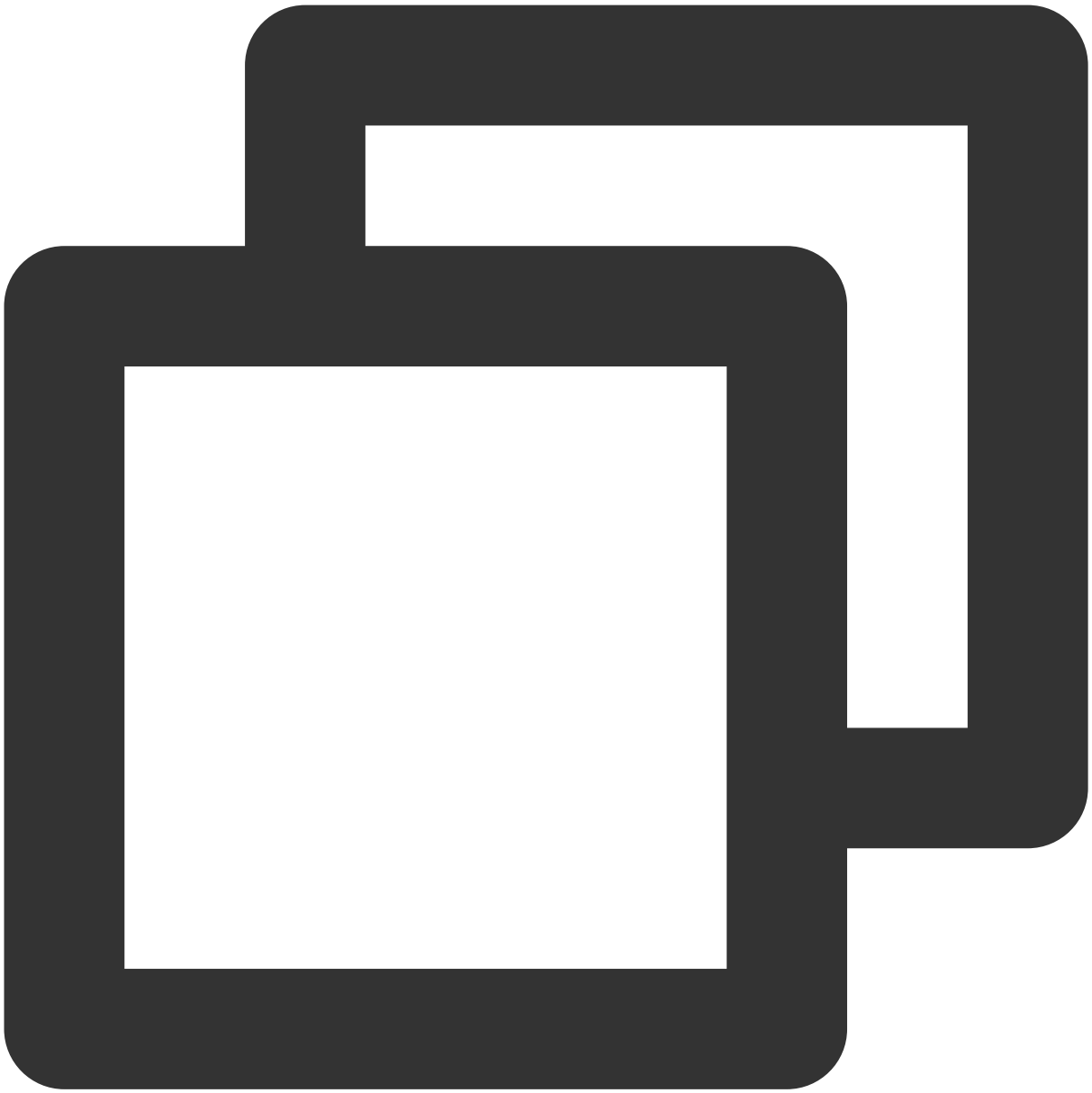


```
SELECT cron.unschedule(1);

 unschedule
------------
          t
```

查看定时任务执行历史

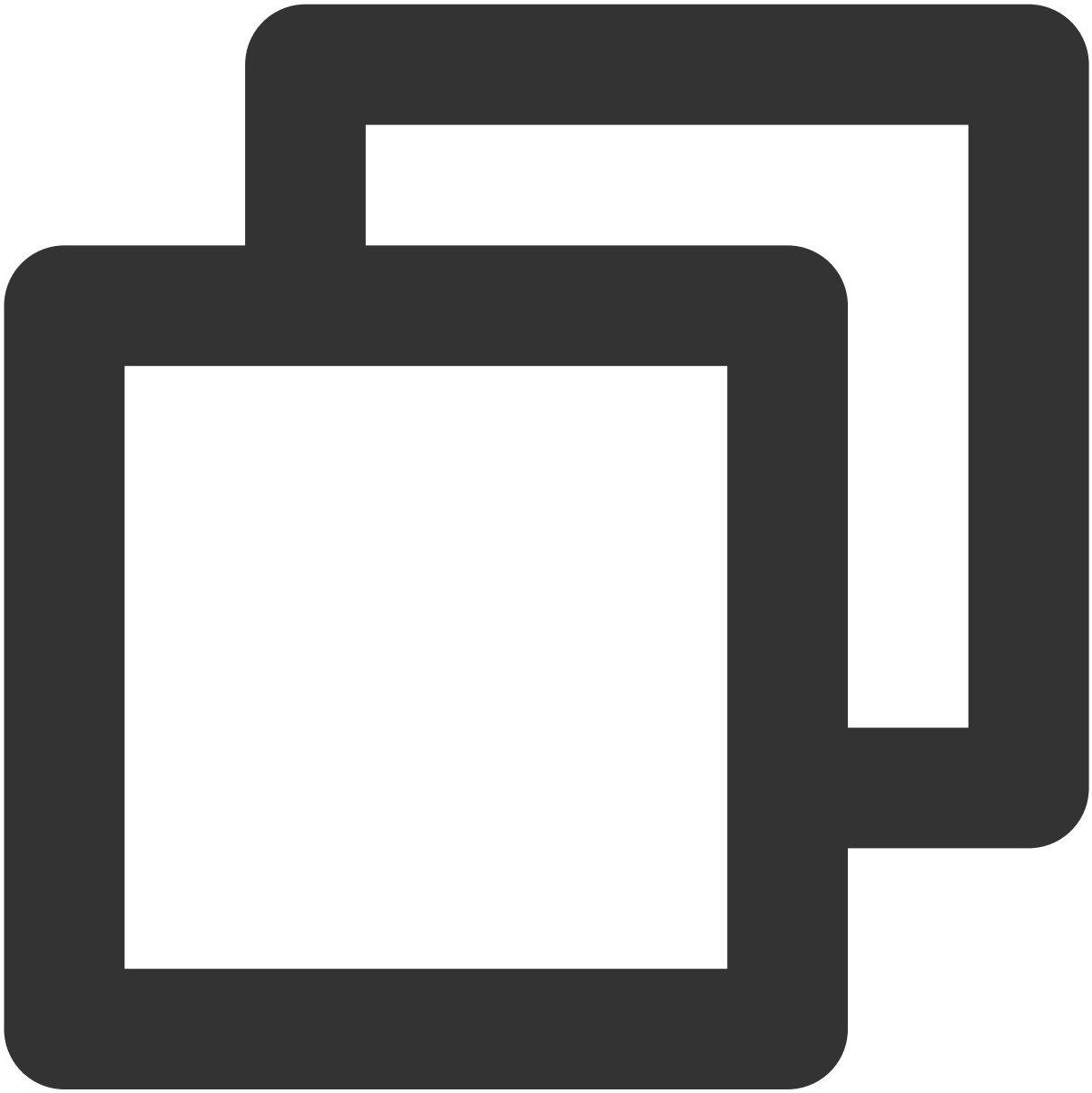
运行上述示例之后，您可以按如下方式检查 cron.job_run_details 表中的任务状态以及执行情况。



```
postgres=> select * from cron.job_run_details;
 jobid | runid | job_pid | database | username | command | status | return_message
-------+-------+---------+----------+----------+-----------------------------------
 1 | 1 | 3395 | postgres | adminuser| vacuum freeze pgbench_accounts | succeeded |
(1 row)
```

清除 pg_cron 历史记录表

1. cron.job_run_details 表包含 cron 任务的历史记录，随着时间的推移，这些历史记录可能会变得非常大。我们建议您计划清除此表的记录。例如，保留一周的条目可能足以进行故障排除。
2. 以下示例使用 cron.schedule 函数计划每天午夜运行以清除 cron.job_run_details 表的记录。如只保留了过去七天的历史记录，如下所示。



```
SELECT cron.schedule('0 0 * * *', $$DELETE
 FROM cron.job_run_details
 WHERE end_time < now() - interval '7 days'$$);
```

禁用 pg_cron 历史记录

如果需要完全禁用向 cron.job_run_details 表中写入任何内容，请在控制台修改将 cron.log_run 参数设置为 off。
如果您执行此操作，pg_cron 扩展不再写入表，只会在 postgresql.log 文件中生成错误。所有的错误信息将都可以在控制台中的错误日志中查看到。
使用以下命令检查 cron.log_run 参数的值。

```
postgres=> SHOW cron.log_run;
```

对 postgres 以外的数据库设置定时任务

默认场景下 pg_cron 的元数据全部保存在 postgres 数据库中。若需要对其他 database 中的对象执行定时任务，则需要执行以下操作。

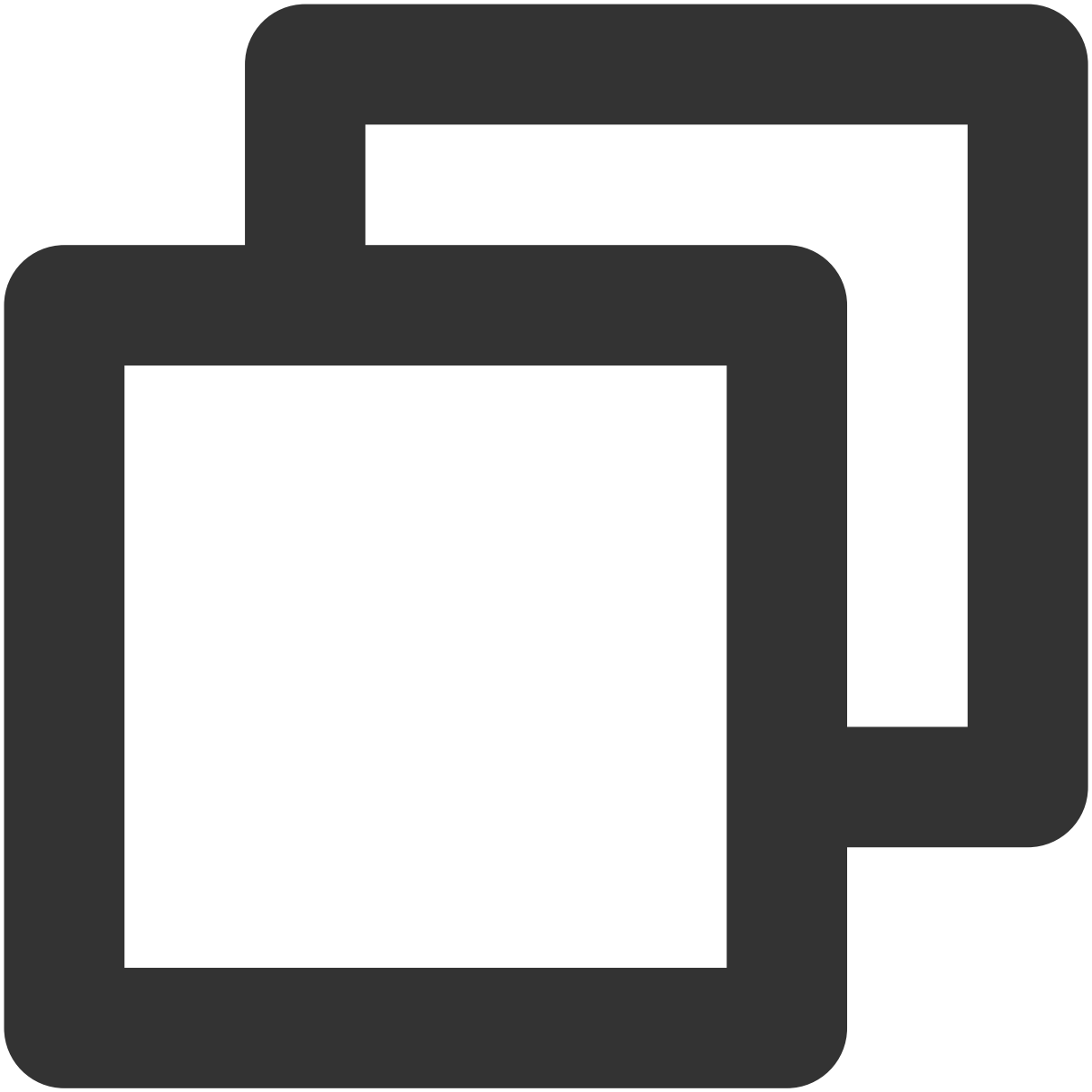
1. 如需要对 test 数据库执行某一个表的 vacuum 操作，则首先需要使用 pg_cron 的管理员用户在 postgres 库中执行 cron.schedule 函数，设置定时任务。

```
postgres=> SELECT cron.schedule('test manual vacuum', '29 03 * * *', 'vacuum freeze
```

2. 使用管理员用户，执行数据库更改，使定时任务执行 database 设置为想要设置的库。请注意 jobid 一定要为第一步中执行返回的 jobid。如下命令

```
postgres=> UPDATE cron.job SET database = 'test' WHERE jobid = 106;
```

3. 通过查询 cron.job 表进行验证。



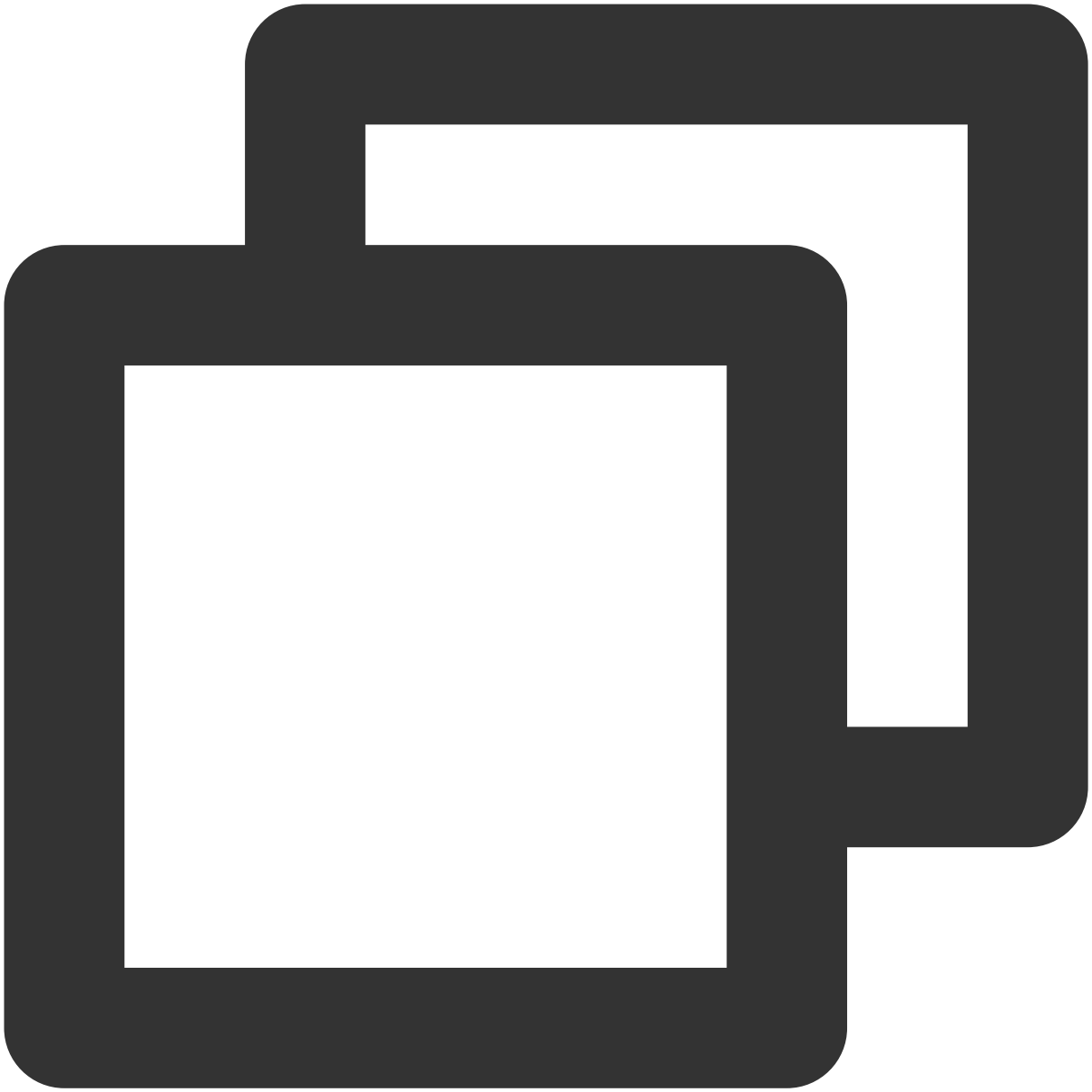
```
postgres=> select * from cron.job;
 jobid | schedule | command | nodename | nodeport | database | username | active |
-------+-------------+----------------------------------------+-----------+--------
 2 | 29 03 * * * | vacuum freeze test_table | localhost | 8192 | test | adminuser |
 1 | 59 23 * * * | vacuum freeze pgbench_accounts | localhost | 8192 | postgres | a
(2 rows)
```

pg_cron 参数

以下是用于控制 pg_cron 扩展行为的参数列表。

参数	描述
cron.database_name	保存 pg_cron 元数据的数据库。
cron.host	要连接到 PostgreSQL 的主机名。您无法修改此值。
cron.log_run	将运行的所有任务记录到 job_run_details 表中。值为 on 或 off。
cron.log_statement	在运行所有 cron 语句之前将其记入日志。值为 on 或 off。
cron.max_running_jobs	可以同时运行的最大任务数，若需要支持更多的任务，请 提交工单 支持。
cron.use_background_workers	使用后台工作程序而不是客户端会话。您无法修改此值。

使用以下 SQL 命令来显示这些参数及其值。



```
postgres=> SELECT name, setting, short_desc FROM pg_settings WHERE name LIKE 'cron.
```

网络管理
网络管理概述

最近更新时间：2024-01-22 16:22:57

云数据库 PostgreSQL 提供网络管理，保障您的实例安全、高效、自由地对外对内提供服务。本文为您介绍 PostgreSQL 网络相关信息。

网络类型

云数据库网络环境可以分为私有网络（Virtual Private Cloud，VPC）和基础网络两种。
私有网络：是一块您在腾讯云上自定义的逻辑隔离网络空间。即使在相同地域下，不同的私有网络之间默认无法互相通信。与您在数据中心运行的传统网络相似，托管在腾讯云私有网络内的是您在腾讯云上的服务资源。
基础网络：是腾讯云上所有用户的公共网络资源池。用户所有云上的资源都由腾讯云统一管理，管理简单、快捷。
说明：
目前已不支持基础网络创建。

功能区别

功能	基础网络	私有网络
网络自定义	不支持	支持
路由自定义	不支持	支持
自定义 IP	不支持	支持
互通规则	同地域互通	同地域同 VPC 内各子网互通
安全控制	安全组	安全组

网络访问

同处于腾讯云上的云产品可以经由外网访问，也可经由内网访问。
外网访问：是腾讯云提供给实例进行公开数据传输的服务。实例开启外网 IP 地址以实现与网络上其他计算机进行通信或通过外网访问实例。
内网访问：内网访问即局域网（LAN）服务，是腾讯云通过提供给实例内网 IP 地址，以实现同地域下完全免费的内网通信服务或通过内网访问实例。
注意：

当前仅北京、上海、广州、成都四地域支持外网安全组功能，其他地域外网开启后有受到攻击的风险。请谨慎开启，不建议业务系统直接使用外网地址。开启外网地址后，必须配置安全组规则才能访问。

网络配置

云数据库 PostgreSQL 可提供实例最多2套、最少1套的网络配置。
实例支持多套网络配置的场景：
同一个实例可通过不同的 VIP（虚拟 IP 地址）访问，其 VIP 可以属于不同的 VPC 和子网。
可通过此功能对实例进行网络修改，如基础网络转为私有网络，或私有网络 A 转为私有网络 B。
可支持此功能进行网络多平面的功能，如有两套不同 VPC 内的业务需要访问同一个数据库实例的场景。

管理实例所属网络

云数据库 PostgreSQL 控制台提供对实例所属网络的管理操作，包括新增网络、删除网络，变更网络。详细您可参见修改网络。

修改网络

最近更新时间：2024-01-22 16:22:57

您可以根据业务需求对实例的所属网络配置进行新增、删除、修改网络。本文为您介绍通过云数据库 PostgreSQL 控制台实现对实例所属网络进行配置和管理。

操作场景

腾讯云网络分为**基础网络**和**私有网络 VPC**，为用户带来不同的优质服务。在此基础之上，我们提供如下更灵活的服务，方便您配置和管理网络。

网络变更

基础网络切换为私有网络：支持单台云数据库主实例的基础网络切换至私有网络。

私有网络 A 切换为私有网络 B：支持单台云数据库主实例、只读实例的私有网络 A 切换至私有网络 B。

自定义访问 IP 地址

自定义主实例 IP：支持在主实例详情页新增网络时指定 IP 地址。

自定义只读实例 IP：支持在只读实例详情页新增网络时指定 IP 地址。

注意事项

基础网络变更为私有网络后不可逆，云数据库切换至私有网络后与其他私有网络及基础网络的云服务不互通。

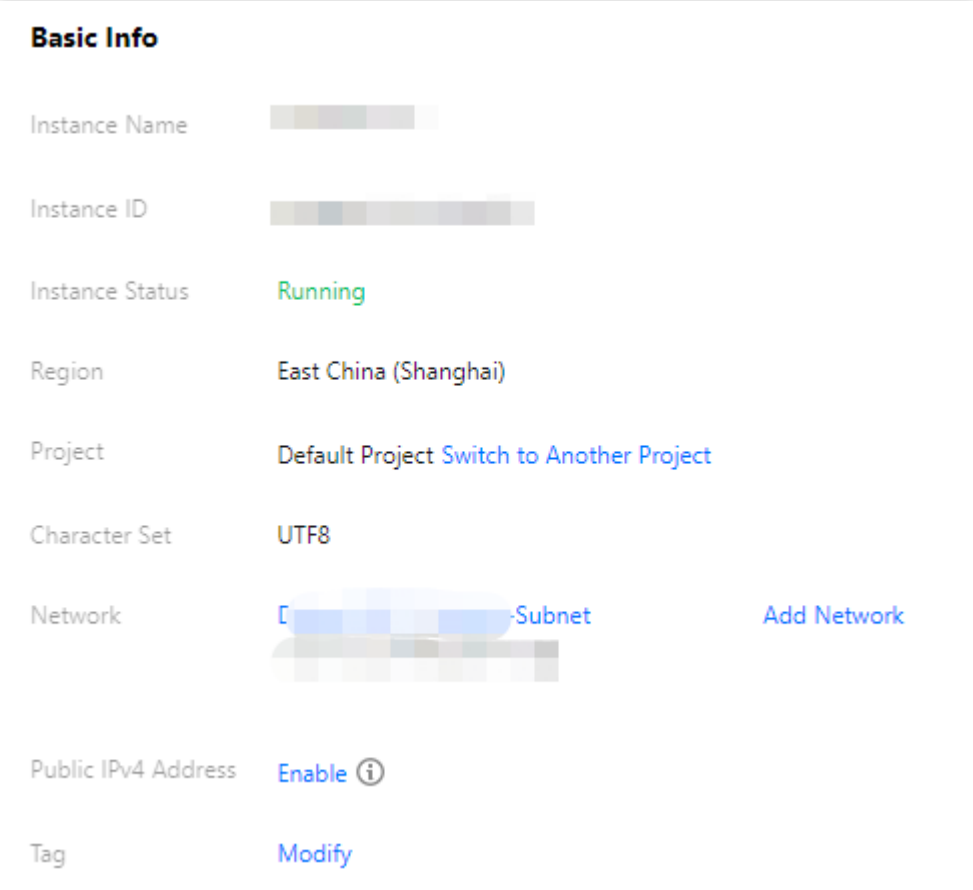
若变更的云数据库为主实例，有挂载只读实例，当主实例变更网络后，挂载的只读实例不会随主实例自动切换网络，需单独进行网络切换。

实例所属网络新增1套网络配置，不会对原网络配置的 IP 地址产生影响。

操作步骤

新增网络

1. 登录 PostgreSQL 控制台，在实例列表，单击实例 ID 或**操作**列的**管理**，进入实例管理页面。
2. 在实例详情页的**基本信息** > **所属网络**后，单击**新增网络**。



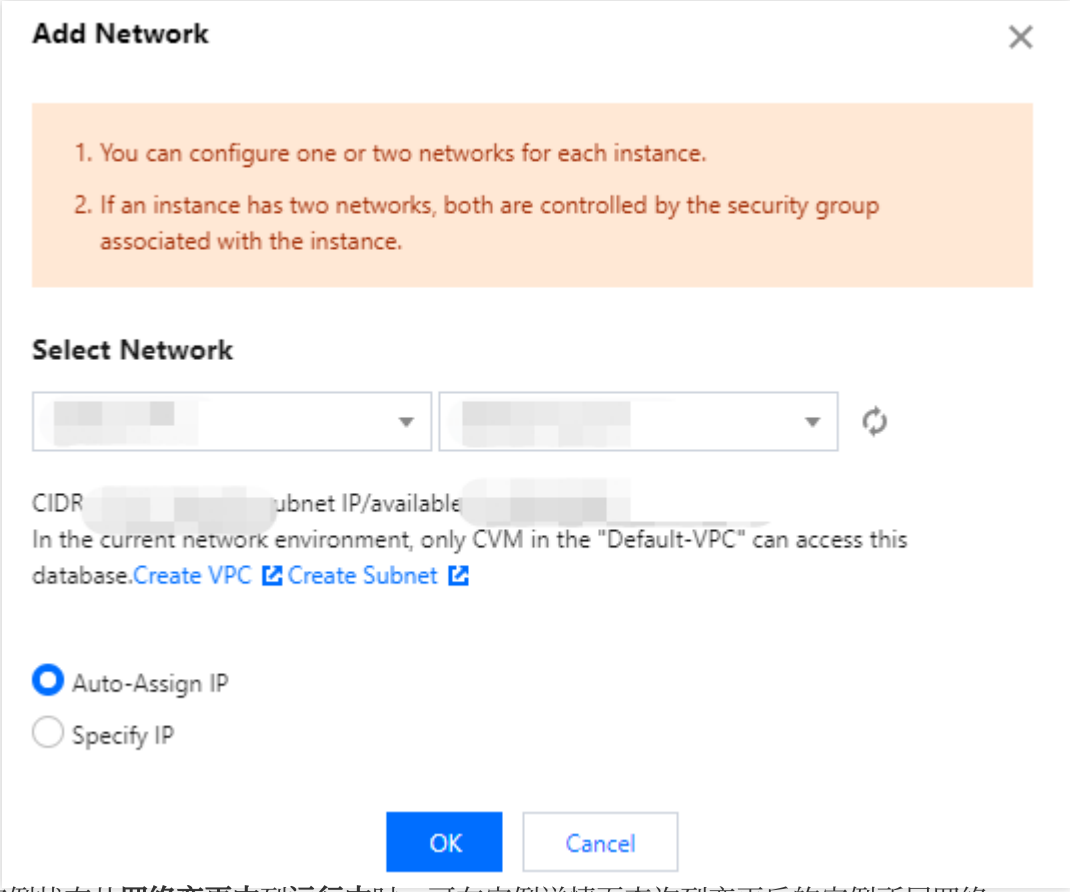
3. 在弹出的对话框，选择网络，可自动设置 IP 或指定 IP，确认后单击**确定**。

说明：

实例最多同时存在两套网络配置，最少必须拥有一套网络配置。

若同实例存在两套网络时，安全组会对两套网络同时生效，请注意配置安全组策略。

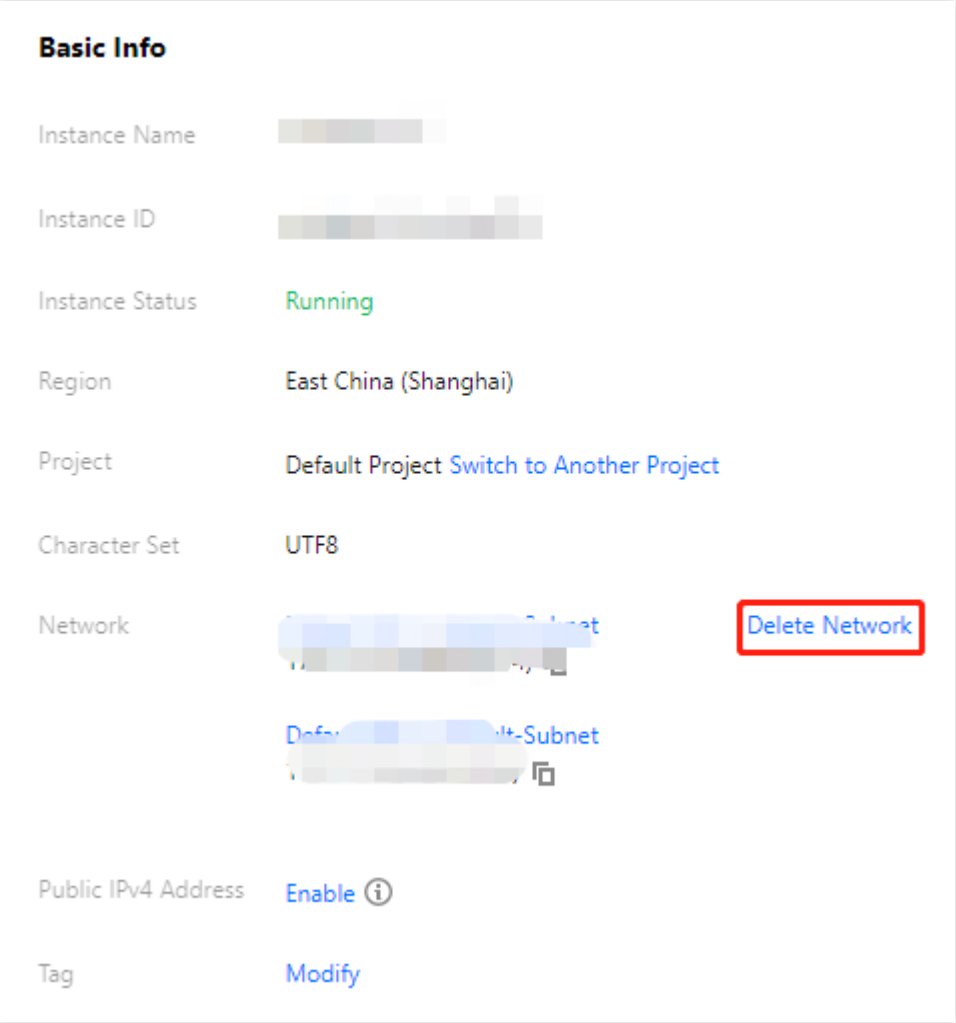
只能选择实例所在地域内的 VPC 网络与子网。



4. 待实例状态从**网络变更中**到**运行中**时，可在实例详情页查询到变更后的实例所属网络。

删除网络

1. 登录 PostgreSQL 控制台，在实例列表，单击实例 ID 或**操作**列的**管理**，进入实例管理页面。
2. 在实例详情页的**基本信息** > **所属网络**后，单击**删除网络**。

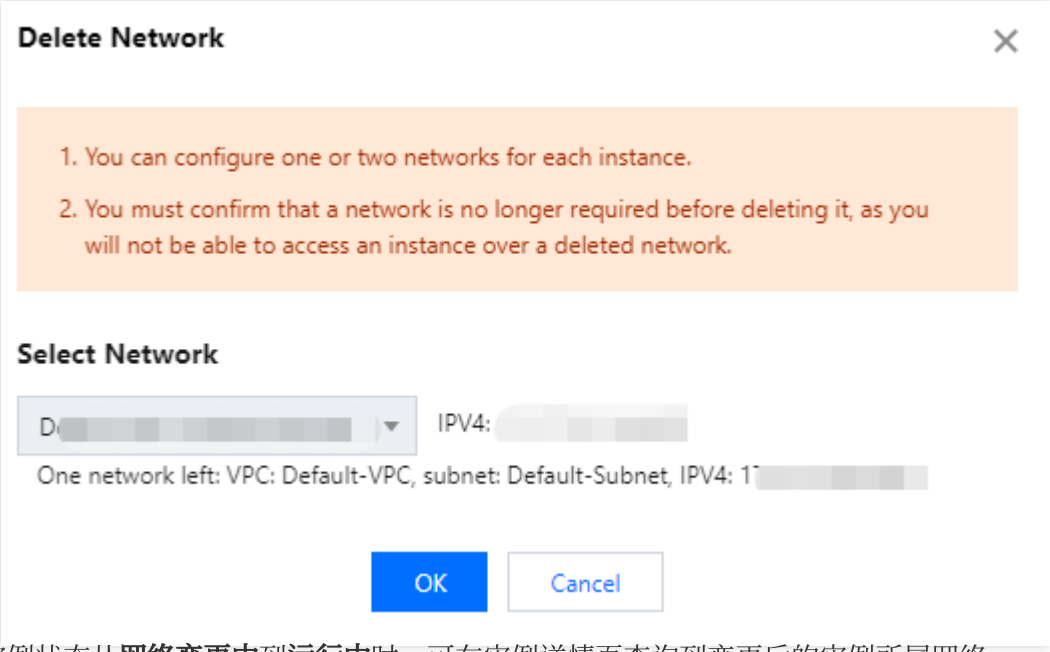


3. 在弹出的对话框，选择需要删除的网络，单击**确定**。

说明：

实例最多同时存在两套网络配置，最少必须拥有一套网络配置。

请确保网络的确不再使用后即可删除，删除后无法通过此网络访问到实例。



4. 待实例状态从**网络变更中**到**运行中**时，可在实例详情页查询到变更后的实例所属网络。

修改网络

如您需要对实例当前所属网络进行变更，如基础网络修改为私有网络，或私有网络 A 修改为私有网络 B，可通过以上新增和删除网络的操作实现。

示例1：基础网络修改为私有网络

1. 登录 PostgreSQL 控制台，在实例列表，单击实例 ID 或**操作**列的**管理**，进入实例管理页面。
2. 在实例详情页的**基本信息** > **所属网络**后，单击**新增网络**，选择需要的私有网络，并单击**确定**。
3. 待实例状态为**运行中**时，在所属网络后单击**删除网络**，选择基础网络，单击**删除**，该实例所属网络即可由原来的基础网络变更为新的私有网络。

示例2：私有网络 A 修改为私有网络 B

1. 登录 PostgreSQL 控制台，在实例列表，单击实例 ID 或**操作**列的**管理**，进入实例管理页面。
2. 在实例详情页的**基本信息** > **所属网络**后，单击**新增网络**，选择需要的私有网络 B，并单击**确定**。
3. 待实例状态为**运行中**时，在所属网络后单击**删除网络**，选择私有网络 A，单击**删除**，该实例所属网络即可由私有网络 A 变更为私有网络 B。

注意：

删除网络后，无法通过此网络访问到实例，请确保该网络的确不再使用。

开启外网地址

最近更新时间：2024-01-22 16:22:57

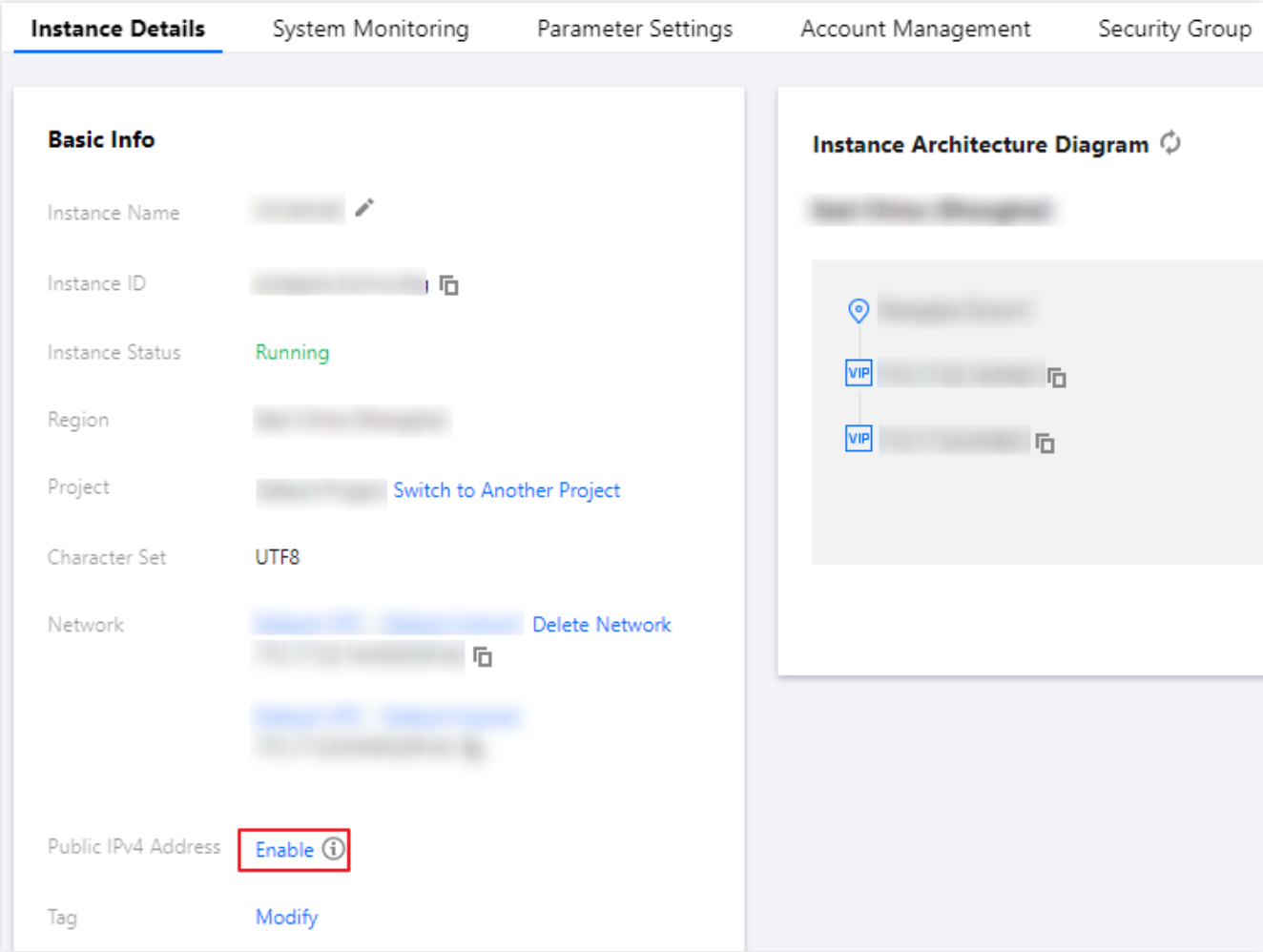
云数据库 PostgreSQL 支持内网地址和外网地址两种地址类型，默认提供内网地址供您内部访问实例，如果需要外网访问，您可通过控制台开启外网地址。

说明：

当前北京、上海、广州、成都地域支持外网安全组功能，支持通过控制台开启外网地址。

通过控制台开启外网地址

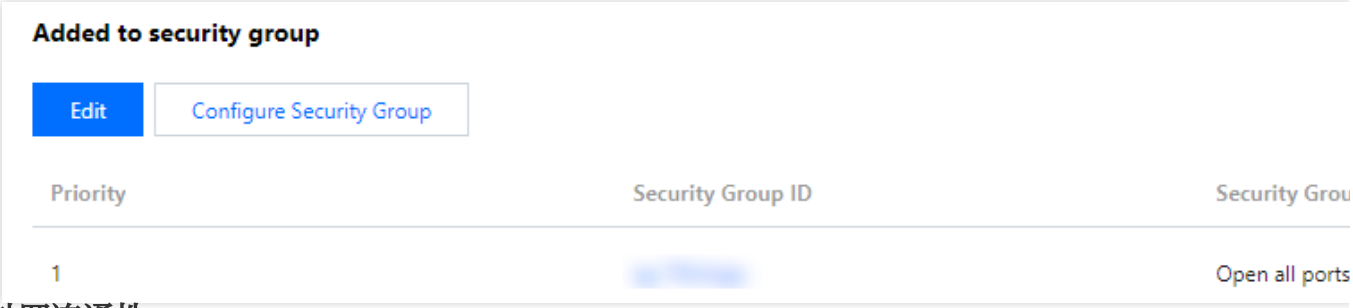
1. 登录 PostgreSQL 控制台，选择地域，在实例列表，单击实例 ID 或**操作**列的**管理**，进入实例管理页面。
2. 在**实例详情**页的**基本信息** > **外网IPv4地址**后，单击**开启**。



3. 在弹出的对话框，阅读注意事项后，单击**确定**。
4. 待实例状态更新为**运行中**，即可在实例详情页查看外网地址。

配置 PodtgreSQL 安全组

1. 登录 PodtgreSQL 控制台，选择地域，在实例列表，单击实例 ID 或**操作**列的**管理**，进入实例管理页面。
2. 在实例管理页面，选择**安全组**页，单击**配置安全组**，配置安全组规则为放通全部端口，确认安全组允许外部 IP 访问，详细配置方法请参见 管理安全组。



验证外网连通性

通过任意外网客户端工具访问 PodtgreSQL。具体操作方法可参考 连接 PostgreSQL 实例。

访问管理
访问管理概述

最近更新时间：2024-01-22 16:22:57

存在问题

如果您在腾讯云中使用到了云数据库 PostgreSQL、私有网络、云服务器、其他数据库等多项服务，这些服务由不同的人管理，但都共享您的云账号密钥，将存在如下问题：
您的密钥由多人共享，泄密风险高。
您无法限制其它人的访问权限，易产生误操作造成安全风险。

解决方案

您可以通过子帐号实现不同的人管理不同的服务来规避以上的问题。默认情况下，子帐号没有使用 PostgreSQL 的权利或者 PostgreSQL 相关资源的权限。因此，我们就需要创建策略来允许子帐号使用他们所需要的资源或权限。
访问管理（Cloud Access Management，CAM）是腾讯云提供的一套访问控制服务，主要用于帮助用户安全管理腾讯云账户下资源的访问权限。通过 CAM，您可以创建、管理和销毁用户（组），并通过身份管理和策略管理控制指定用户可以使用的腾讯云资源。
当您使用 CAM 的时候，可以将策略与一个用户或一组用户关联起来，策略能够授权或者拒绝用户使用指定资源完成指定任务。有关 CAM 策略的更多基本信息，请参见 策略语法。
若您不需要对子账户进行 PostgreSQL 相关资源的访问管理，您可以跳过此章节。跳过这些部分不会影响您对文档中其余部分的理解和使用。

快速入门

CAM 策略必须授权使用一个或多个 PostgreSQL 实例操作或者必须拒绝使用一个或多个 PostgreSQL 实例操作。同时还必须指定可以用于操作的资源（可以是全部资源，某些操作也可以是部分资源），策略还可以包含操作资源所设置的条件。
PostgreSQL 部分云 API 操作不支持资源级权限，意味着，对于该类 API 操作，您不能在使用该类操作的时候指定某个具体的资源来使用，而必须要指定全部资源来使用。

任务	链接
了解策略基本结构	策略语法
在策略中定义操作	PostgreSQL 的操作
在策略中定义资源	PostgreSQL 的资源路径

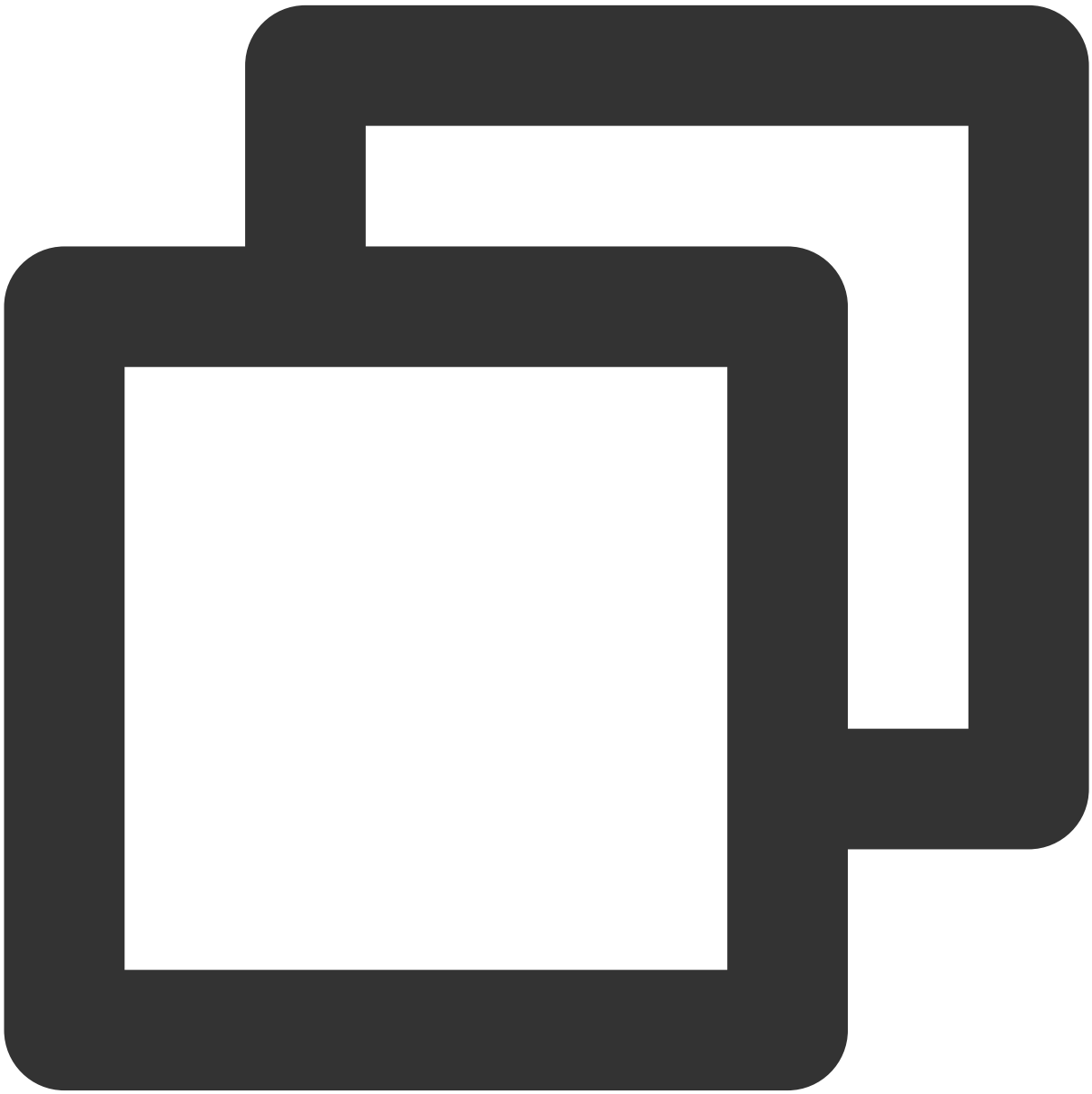
PostgreSQL 支持的资源级权限	PostgreSQL 支持的资源级权限
控制台示例	控制台示例

授权策略语法

最近更新时间：2024-01-22 16:22:57

策略语法

CAM 策略：



```
{
        "version":"2.0",
        "statement":
```

```
        [
            {
                "effect":"effect",
                "action":["action"],
                "resource":["resource"],
                 "condition": {"key":{"value"}}
            }
        ]
}
```

版本 version：必填项，目前仅允许值为"2.0"。

语句 statement：用来描述一条或多条权限的详细信息。该元素包括 effect、action、resource、condition 等多个其他元素的权限或权限集合。一条策略有且仅有一个 statement 元素。

影响 effect：必填项，描述声明产生的结果是“允许”还是“显式拒绝”。包括 allow（允许）和 deny（显式拒绝）两种情况。

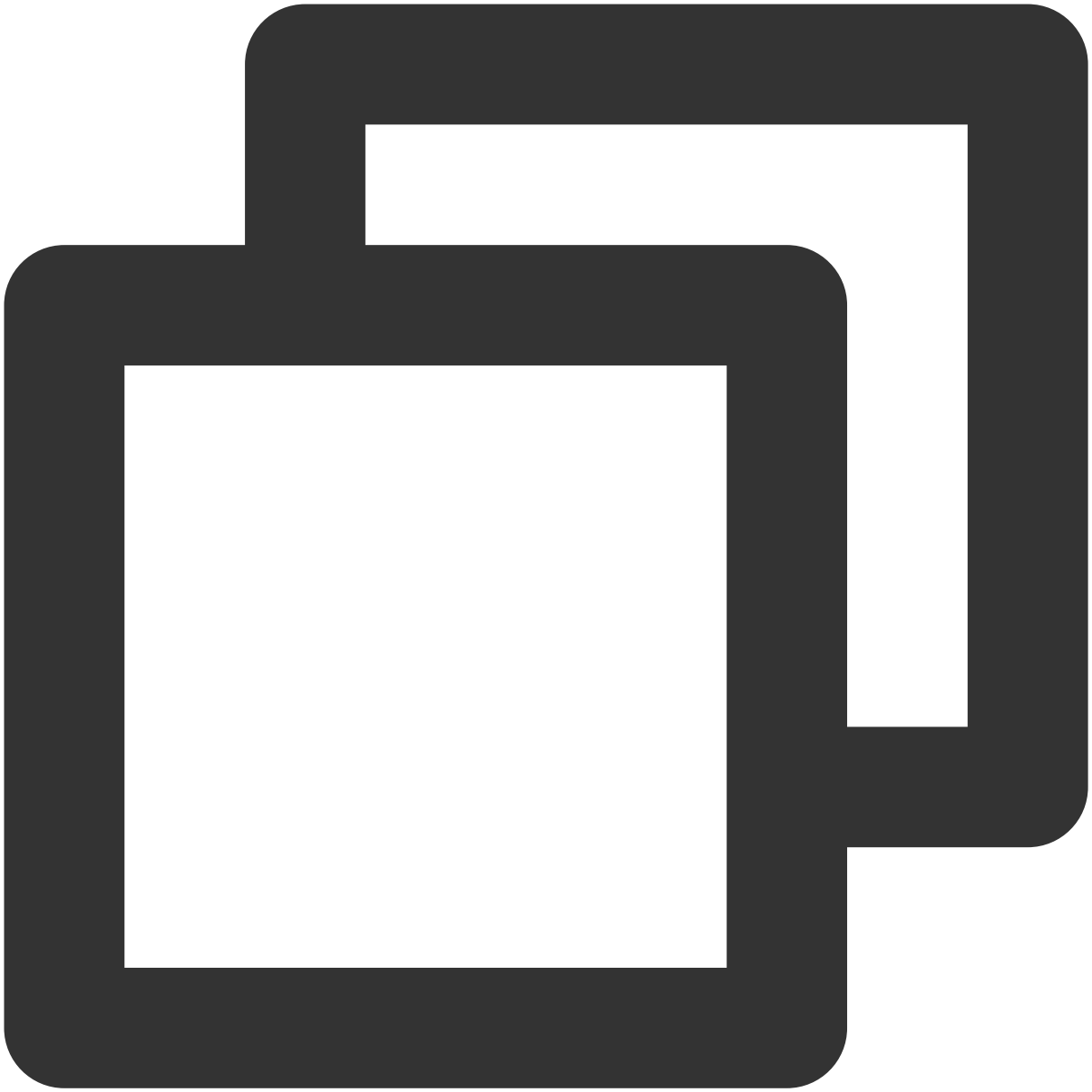
操作 action：必填项，用来描述允许或拒绝的操作。操作可以是 API 或者功能集（一组特定的 API ，以 permid 前缀描述）。

资源 resource：必填项，描述授权的具体数据。资源是用六段式描述，每款产品的资源定义详情会有所区别。

生效条件 condition：描述策略生效的约束条件。条件包括操作符、操作键和操作值组成。PostgreSQL 目前并不支持特殊的生效条件，所以此项可不进行配置。

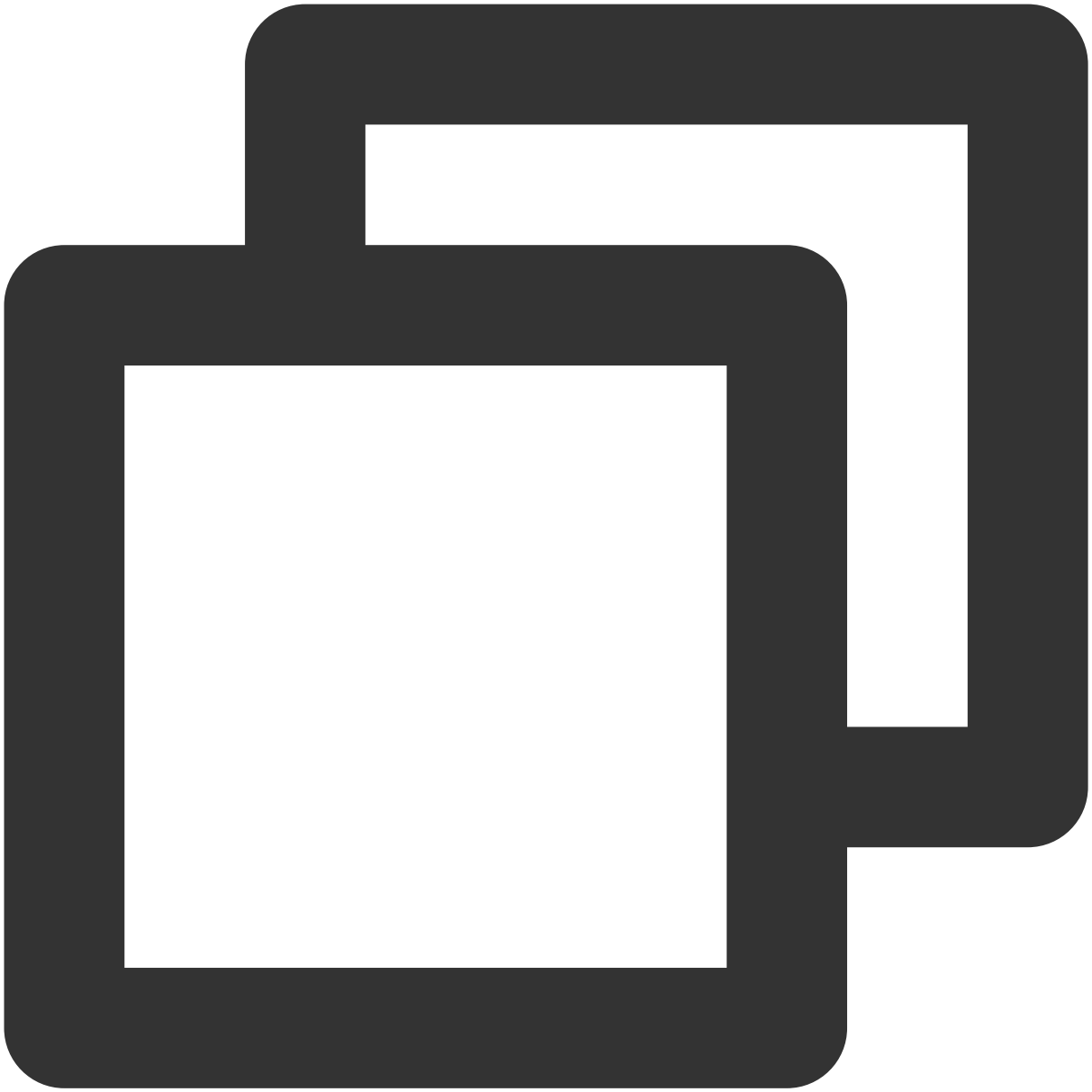
PostgreSQL 的操作

在 CAM 策略语句中，您可以从支持 CAM 的任何服务中指定任意的 API 操作。对于 PostgreSQL，请使用以 postgres: 为前缀的 API。例如 postgres:DescribeDBInstances 或者 postgres:DescribeDBInstanceAttribute。
如果您要在单个语句中指定多个操作的时候，请使用逗号将它们隔开，如下所示：



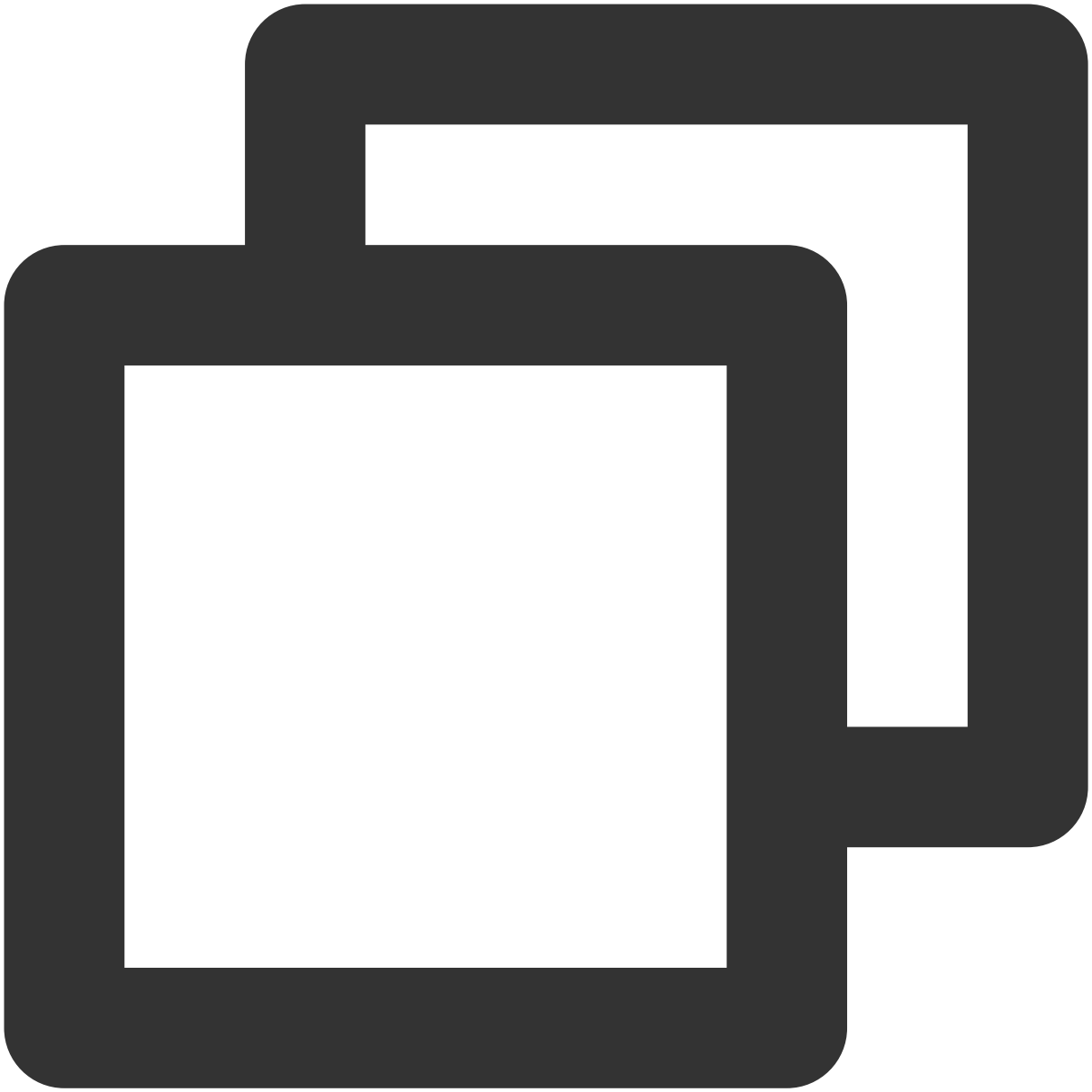
```
"action":["postgres:action1","postgres:action2"]
```

您也可以使用通配符指定多项操作。例如，您可以指定名字以单词 "Describe" 开头的所有操作，如下所示：



```
"action":["postgres:Describe*"]
```

如果您要指定 PostgreSQL 中所有操作，请使用 * 通配符，如下所示：

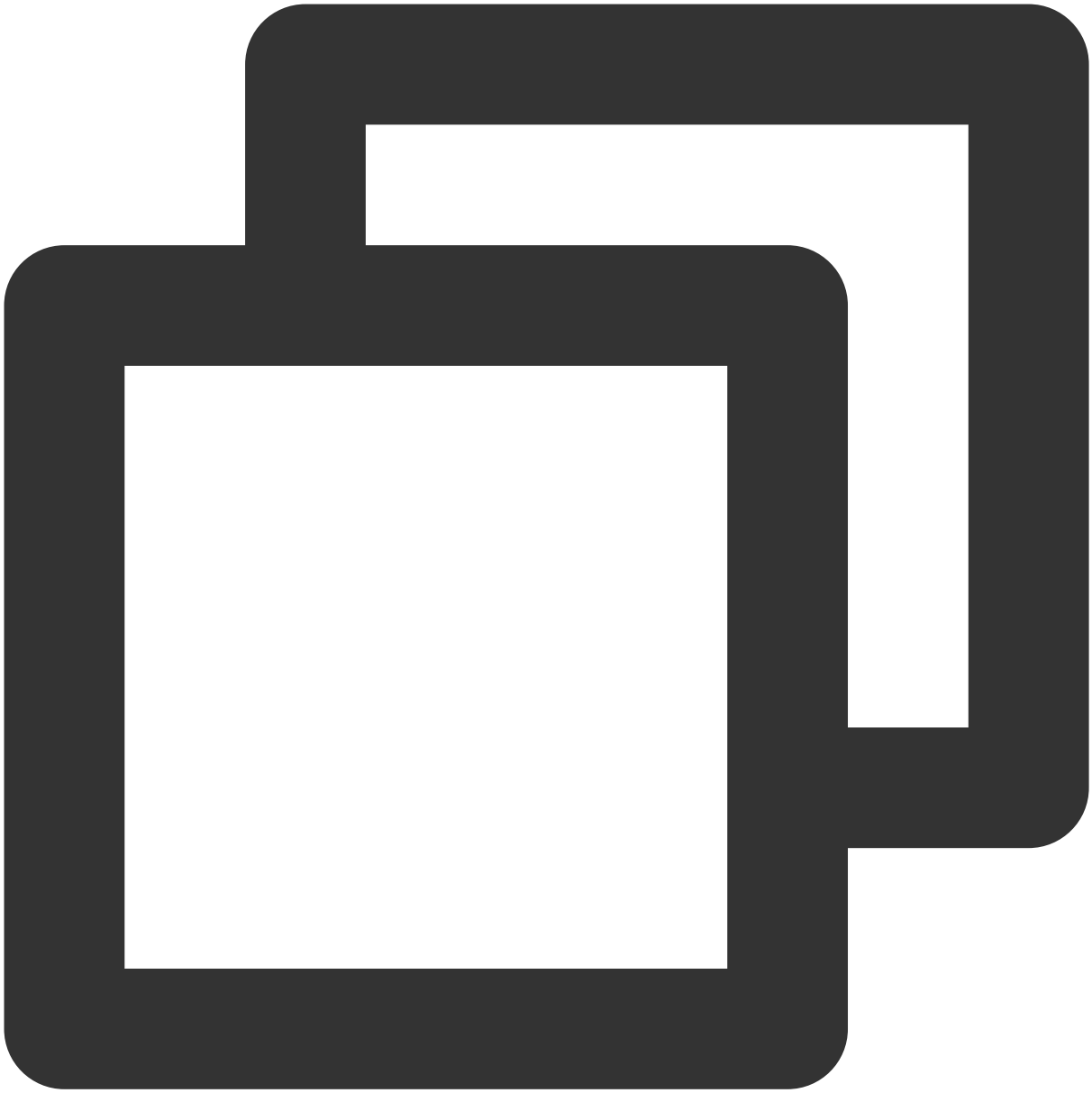


```
"action"：["postgres:*"]
```

PostgreSQL 的资源路径

每个 PostgreSQL 策略语句都有适用于自己的资源。

资源路径的一般形式如下：



```
qcs:project_id:service_type:region:account:resource
```

project_id：描述项目信息，仅为了兼容 CAM 早期逻辑，无需填写。

service_type：产品简称 postgres。

region：地域信息，如 ap-shanghai。

account：资源拥有者的主帐号信息（即 账号信息 页的“账号 ID”），如164xxx472。

resource：各产品的具体资源详情，如实例为 DBInstanceId/postgres-0xssvm8e 或者 DBInstanceId/*，下表描述了 PostgreSQL 能够使用的资源和对应的资源描述方法。

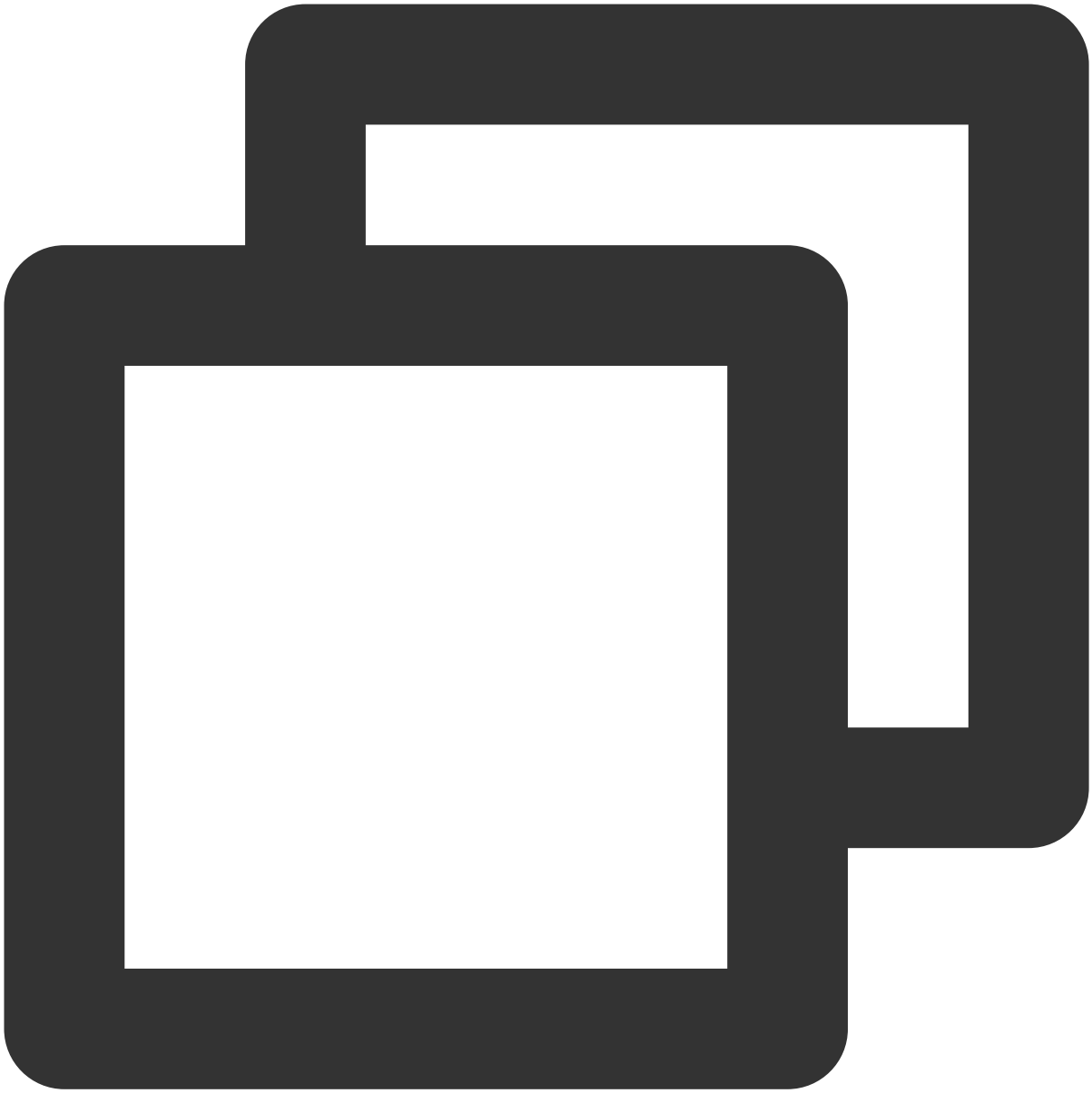
资源	授权策略中的资源描述方法

实例	qcs::postgres:$region:$account:DBInstanceId/$DBInstanceId

例如，您可以使用特定实例（实例 ID 为 postgres-0xssvm8e）语句中指定它，如下所示：

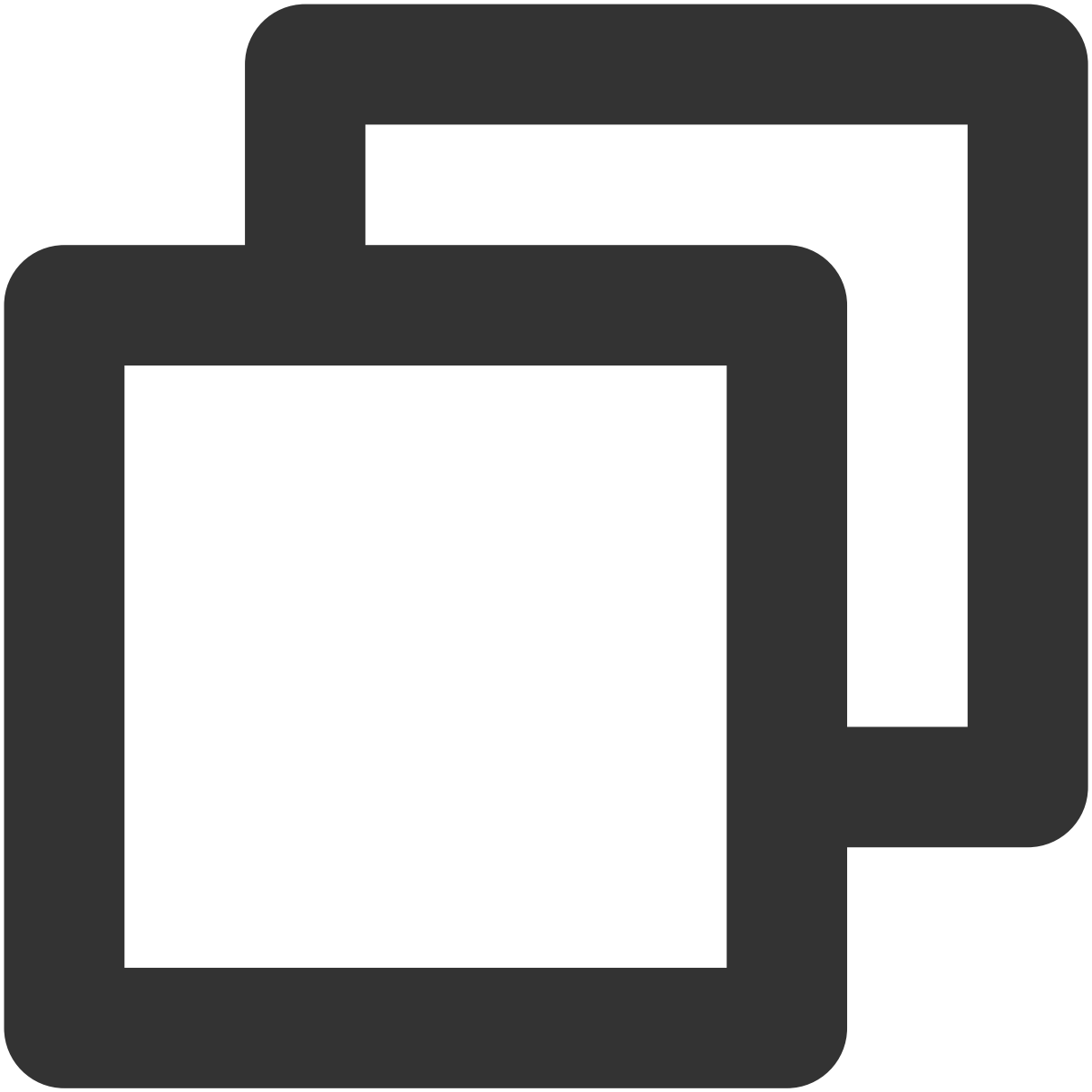
```
"resource":[ "qcs::postgres:ap-shanghai:164xxx472:DBInstanceId/postgres-0xssvm8e"]
```

您还可以使用 * 通配符指定属于特定账户上海地域的所有实例，如下所示：



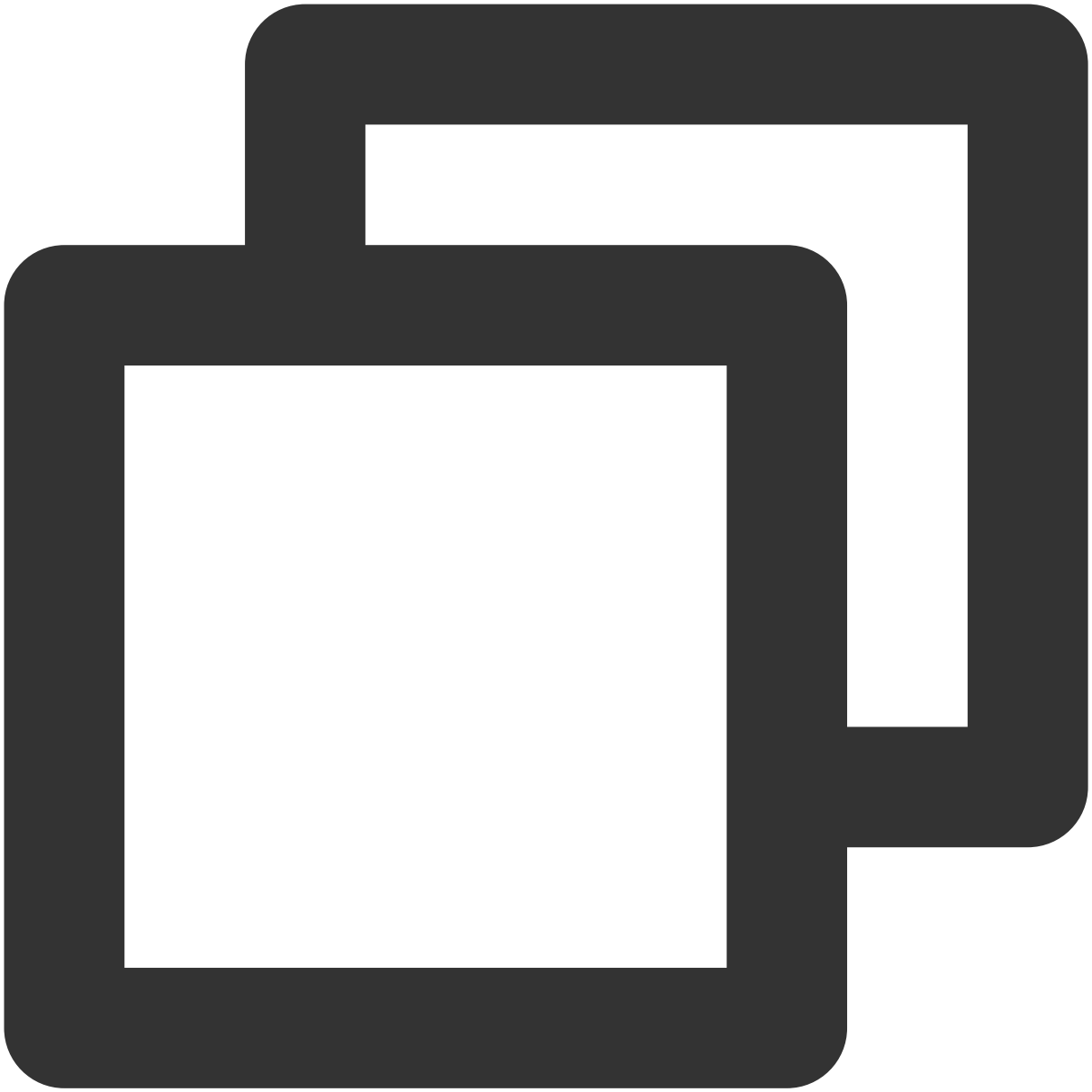
```
"resource":[ "qcs::postgres:ap-shanghai:164xxx472:DBInstanceId/*"]
```

您要指定所有资源，或者如果特定 API 操作不支持资源级权限，请在 Resource 元素中使用 * 通配符，如下所示：



```
"resource": ["*"]
```

如果您想要在一条指令中同时指定多个资源，请使用逗号将它们隔开，如下所示为指定两个实例的例子：



"resource":["qcs::postgres::164xxx472:DBInstanceId/postgres-0xf1f41e","qcs::postgre

可授权的资源类型

最近更新时间：2024-01-22 16:22:57

资源级别权限介绍

资源级权限指的是能够指定用户对哪些资源具有执行操作的能力。PostgreSQL 部分支持资源级权限，即允许用户执行操作或是允许用户使用的特定资源。
访问管理 CAM 中可对 PostgreSQL 授权的资源类型如下：

资源类型	授权策略中的资源描述方法
实例	`qcs::postgres:$region:$account:DBInstanceId/$DBInstanceId` `qcs::postgres:$region:$account:DBInstanceId/*`

PostgreSQL 实例相关分别介绍了当前支持资源级权限的 PostgreSQL API 操作，以及每个操作支持的资源和条件密钥。设置资源路径时，您需要将 `$region` 、 `$account` 等变量参数修改为您实际的参数信息，同时您也可以在路径中使用 * 通配符。相关操作示例可参见 控制台示例。

注意：

不支持资源级权限的 PostgreSQL API 操作，您仍可以向用户授予使用该操作的权限，但是策略语句的资源元素必须指定为 *。

不支持资源级权限的 API 列表

API 操作	API 介绍
CreateDBInstances	创建实例
CreateServerlessDBInstance	创建 ServerlessDB 实例
DescribeOrders	获取订单信息
DescribeRegions	查询售卖地域
DescribeZones	查询售卖可用区
DescribeProductConfig	查询售卖规格配置
InquiryPriceCreateDBInstances	查询售卖价格

DescribeServerlessDBInstances	查询 ServerlessDB 实例列表

支持资源级权限的 API 列表

[PostgreSQL 实例相关]

ServerlessDB 相关接口

接口名称	接口功能
CloseServerlessDBExtranetAccess	关闭 serverlessDB 实例外网
DeleteServerlessDBInstance	删除 ServerlessDB 实例
OpenServerlessDBExtranetAccess	开通 ServerlessDB 实例外网

备份与恢复相关接口

接口名称	接口功能
DescribeDBBackups	查询实例备份列表
DescribeDBErrlogs	获取错误日志
DescribeDBSlowlogs	获取慢查询日志
DescribeDBXlogs	获取实例 Xlog 列表

实例相关接口

接口名称	接口功能
CloseDBExtranetAccess	关闭实例外网链接
DescribeDBInstanceAttribute	查询实例详情
DescribeDatabases	拉取数据库列表
DestroyDBInstance	销毁实例
InitDBInstances	初始化实例
InquiryPriceRenewDBInstance	查询实例续费价格
InquiryPriceUpgradeDBInstance	查询升级实例价格
ModifyDBInstanceName	修改实例名字

版权所有：腾讯云计算（北京）有限责任公司

ModifyDBInstancesProject	将实例转至其他项目
OpenDBExtranetAccess	开通外网
RenewInstance	续费实例
RestartDBInstance	重启实例
SetAutoRenewFlag	设置自动续费
UpgradeDBInstance	升级实例
DescribeDBInstances	查询实例列表

账号相关接口

接口名称	接口功能
DescribeAccounts	获取实例用户列表
ModifyAccountRemark	修改账号备注
ResetAccountPassword	重置账号密码

版权所有：腾讯云计算（北京）有限责任公司

控制台示例

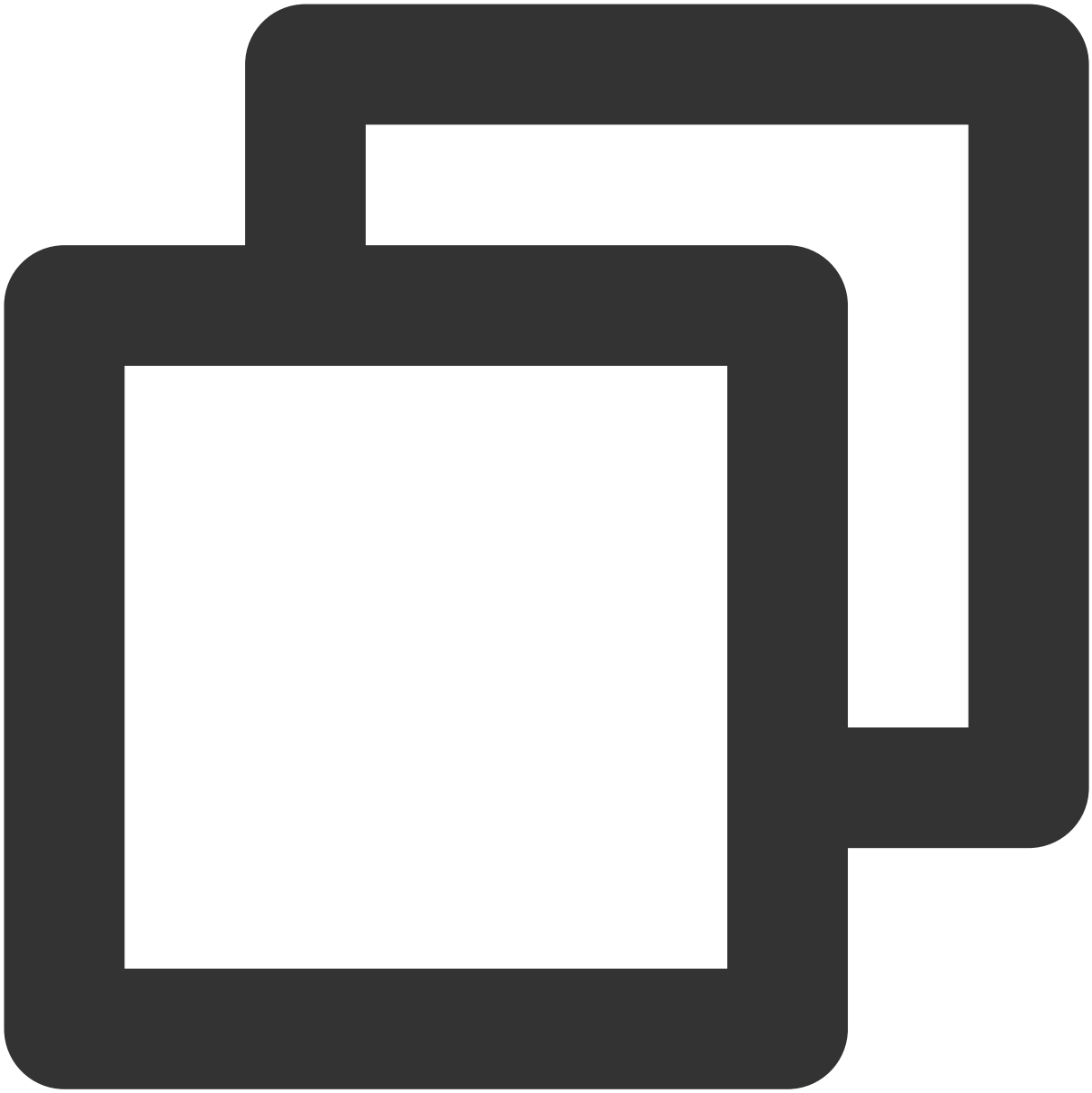
最近更新时间：2024-01-22 16:22:57

操作场景

您可以通过使用访问管理（Cloud Access Management，CAM）策略让用户拥有在 PostgreSQL 控制台 中查看和使用特定资源的权限。本文档提供了查看和使用特定资源的权限示例，指导用户如何使用控制台的特定部分的策略。

操作步骤

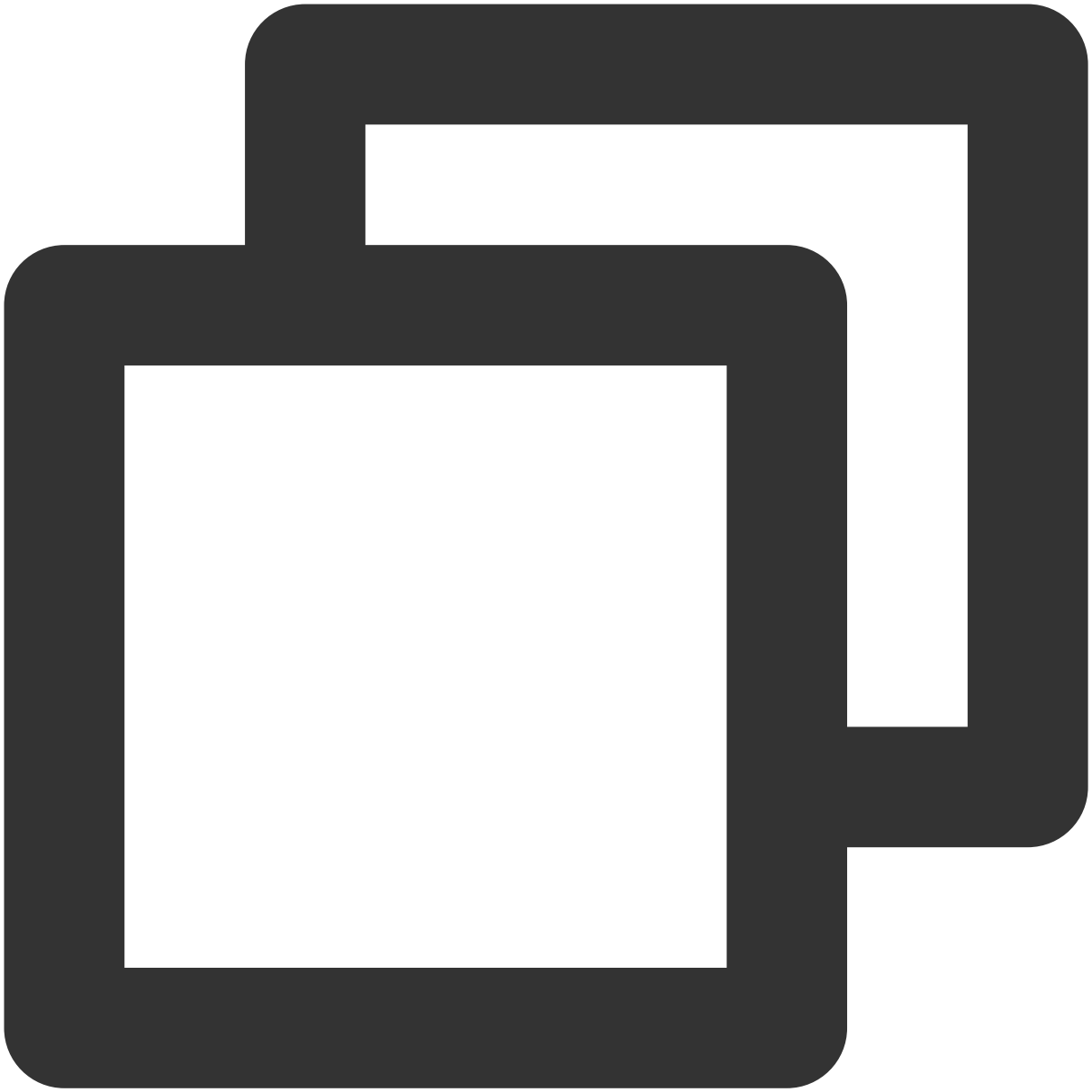
注意：
如果您想要控制某一个用户只拥有某些特定的接口权限，那么需要至少包含以下接口的权限，否则控制台将无法正常显示。
其中 action 示例为：



```
"action": [
"postgres:DescribeProductConfig",
"postgres:InquiryPriceCreateDBInstances",
"postgres:DescribeRegions",
"postgres:DescribeZones"
]
```

注意：

如果您想要用户具有实例的监控查看权限，则需要添加 monitor 相关的接口权限，action 示例如下：



```
{"effect": "allow",
"action": [
"monitor:Get*",
"monitor:Describe*"
],
"resource": "*"
}
```

PostgreSQL 的全读写策略

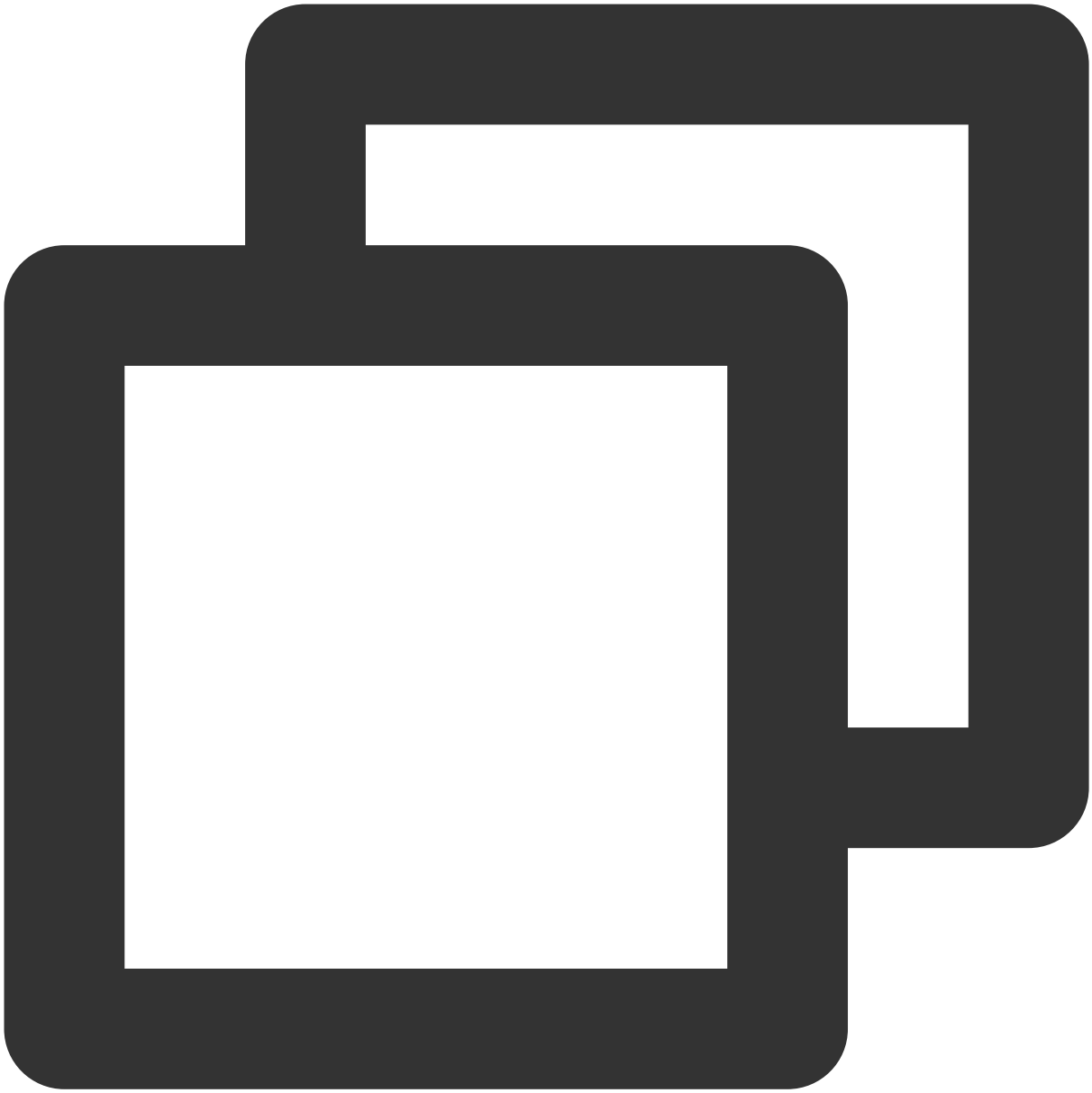
如果您希望用户拥有创建和管理 PostgreSQL 实例的权限，您可以对该用户使用名称为：QcloudPostgreSQLFullAccess 的策略。
该策略可让用户拥有 PostgreSQL 中所有资源的操作权限。具体操作步骤如下：
参考 授权管理，将预设策略 QcloudPostgreSQLFullAccess 授权给用户。

PostgreSQL 的只读策略

如果您希望用户拥有查询 PostgreSQL 实例的权限，但是不具有创建、删除、修改的权限，您可以对该用户使用名称为：QcloudPostgreSQLReadOnlyAccess 的策略。
该策略可让用户拥有 PostgreSQL 中所有以单词 “Describe” 和 “Inquiry” 开头的操作的权限。具体操作步骤如下：
参考 授权管理，将预设策略 PostgreSQL 授权给用户。

授权用户拥有特定实例的操作权限策略

如果您希望授权用户拥有特定 PostgreSQL 操作权限，可将以下策略关联到该用户。具体操作步骤如下：
1. 根据 策略，创建一个自定义策略。
该示例策略允许用户拥有实例 ID 为 postgres-0xxxx8e 的 PostgreSQL 实例的所有操作权限，策略内容可参考以下策略语法进行设置：



```
{
 "version": "2.0",
 "statement": [
     {
         "action": "postgres:*",
         "resource": "qcs::postgres:ap-shanghai:103xxx1481:DBInstanceId/postgres-0x
         "effect": "allow"
     }
 ]
}
```

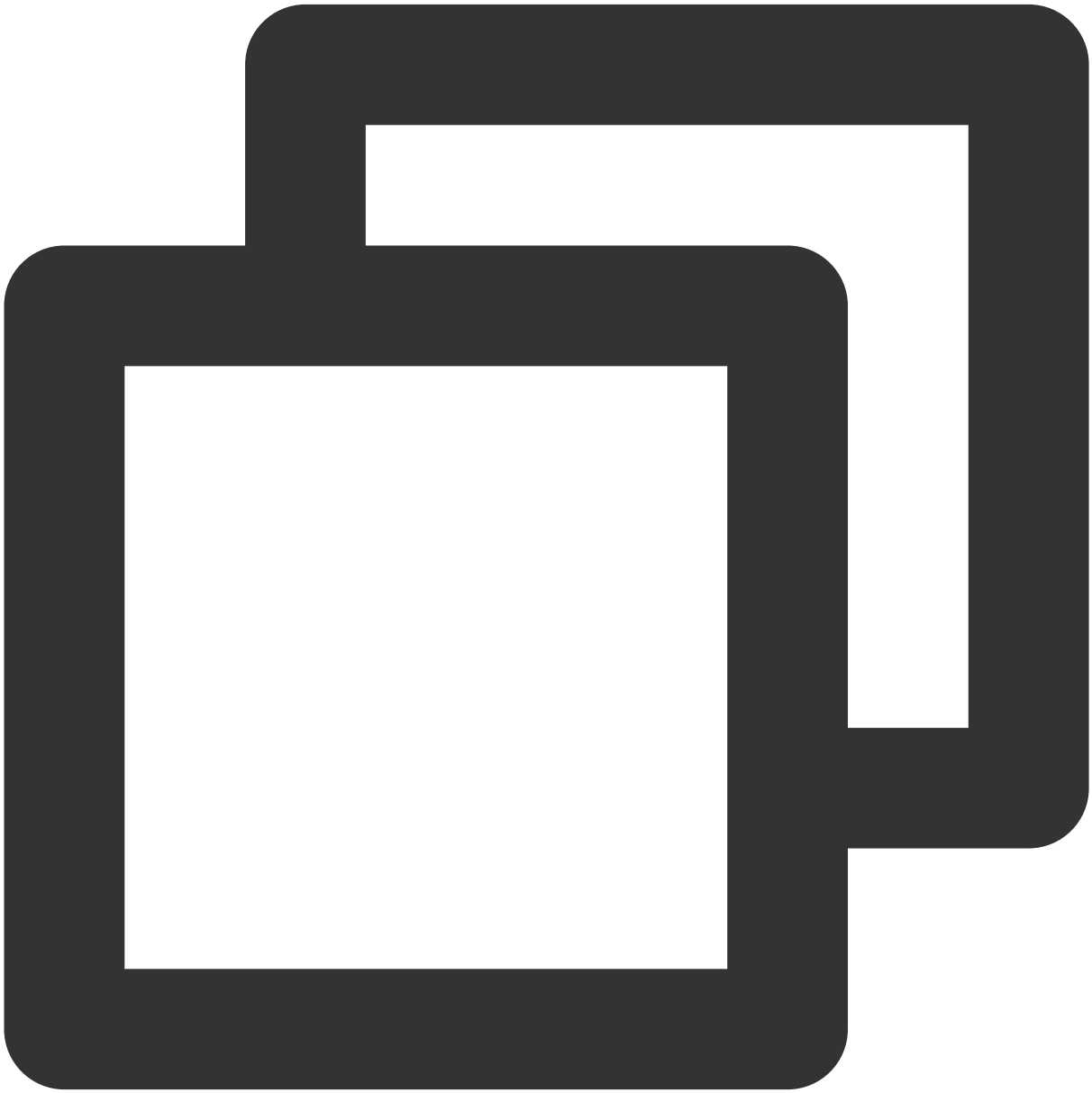
2. 找到创建的策略，在该策略行的**操作**列中，单击**关联用户/组**。
3. 在弹出的关联用户/用户组窗口中，选择您需要授权的用户/组，单击**确定**。

授权用户拥有 PostgreSQL 所有资源的操作权限策略

如果您希望授权用户拥有 PostgreSQL 所有资源的操作权限，可将以下策略关联到该用户。具体操作步骤如下：

1. 根据 策略，创建一个自定义策略。

该示例策略允许用户拥有对 PostgreSQL 所有资源的操作权限，策略内容可参考以下策略语法进行设置：



```
{
 "version": "2.0",
```

```
  "statement": [
      {
          "action": "postgres:*",
          "resource": "qcs::postgres:::*",
          "effect": "allow"
      }
  ]
}
```

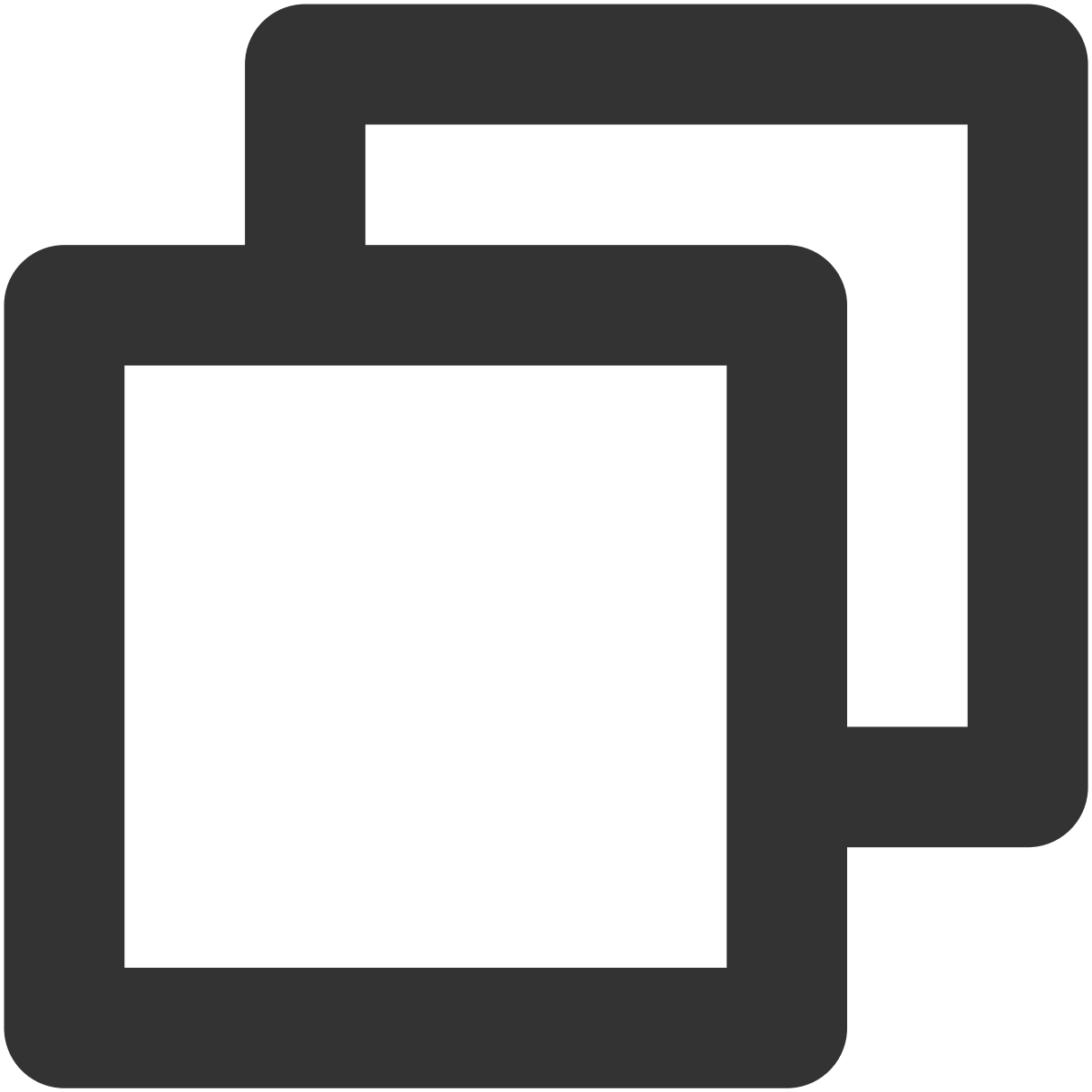
2. 找到创建的策略，在该策略行的**操作**列中，单击**关联用户/组**。
3. 在弹出的关联用户/用户组窗口中，选择您需要授权的用户/组，单击**确定**。

禁止用户拥有特定 PostgreSQL 部分实例的所有权限策略

如果您希望禁止用户拥有特定 PostgreSQL 部分实例的操作权限，可将以下策略关联到该用户。具体操作步骤如下：

1. 根据 策略，创建一个自定义策略。

该示例策略禁止用户拥有对实例（ID 为 postgres-c8xxxa4 和 postgres-d8xxxb4）的操作权限，策略内容可参考以下策略语法进行设置：



```
{
 "version": "2.0",
 "statement": [
     {
         "action": "postgres:*",
         "resource": [
                     "qcs::postgres::16xxx472:DBInstanceId/postgres-c8xxxa4",
                     "qcs::postgres::16xxx472:DBInstanceId/postgres-d8xxxb4",
                     ],
         "effect": "deny"
     }
```

```
  ]
}
```

2. 找到创建的策略，在该策略行的**操作**列中，单击**关联用户/组**。
3. 在弹出的关联用户/用户组窗口中，选择您需要授权的用户/组，单击**确定**。

自定义策略

如果您觉得预设策略不能满足您的要求，您可以通过创建自定义策略达到目的。

具体操作步骤请参考 策略。

更多 PostgreSQL 相关的策略语法请参考 授权策略语法。

数据加密
透明数据加密概述

最近更新时间：2024-01-22 16:22:57

概述

透明数据加密（Transparent Data Encryption，以下简称 TDE），提供文件级别的加密，可以做到对数据库上层的应用透明，用户不需要修改业务代码，存储在硬盘上的数据是加密的，对磁盘数据读取和写入时透明地进行加解密。
透明数据加密常用于解决一些安全合规问题，如 PCI DSS、等级安全保护等要求静态数据需要被保护的场景。

加密是什么

在密码学中，加密（Encryption）是将明文信息改变为难以读取的密文内容，使之不可读的过程。
现代密码学是基于数论和概率论的一门学科，加密的终极目标是香农定义的 Perfect security（也称 **Information-theoretic security**）：
Let E = (E, D) be a Shannon cipher defined over (K,M, C). Consider a probabilistic experiment in which the random variable k is uniformly distributed over K. If for all m0, m1 ∈ M, and all c ∈ C, we have
Pr[E(k, m0) = c] = Pr[E(k, m1) = c],
then we say that E is a perfectly secure Shannon cipher.
简单来说，对于密文 c，是可由任何一个明文 m 加密而来的，从密文中找不到它和明文之间的关联性。

加密算法的分类

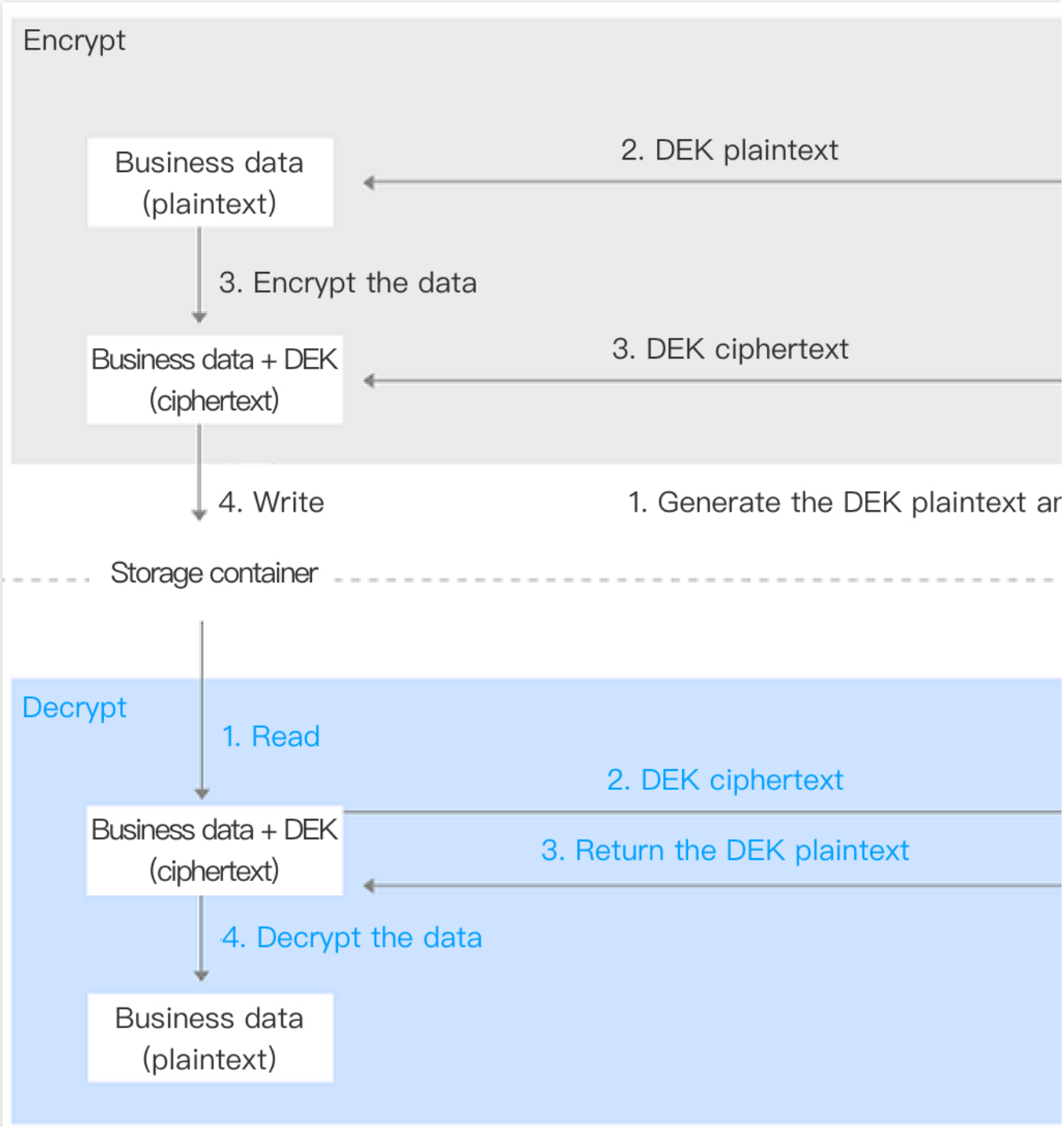
加密算法可以分为两类：对称加密和非对称加密。
对称加密：加密和解密使用同样的密钥，对称加密的速度比非对称加密快很多，在很多场合都需要对称加密。
非对称加密：又称公开密钥加密，是加密和解密使用不同密钥的算法，主要用户信息传输。
对称加密算法常见的有 AES，3DES 等。非对称加密算法最广为使用的是 RSA 算法，其可靠性源自于对极大整数做因数分解的难度。

TDE 威胁模型（Threat model）

TDE 主要用于保护静态数据 data at rest，防止磁盘被盗窃导致数据泄漏。

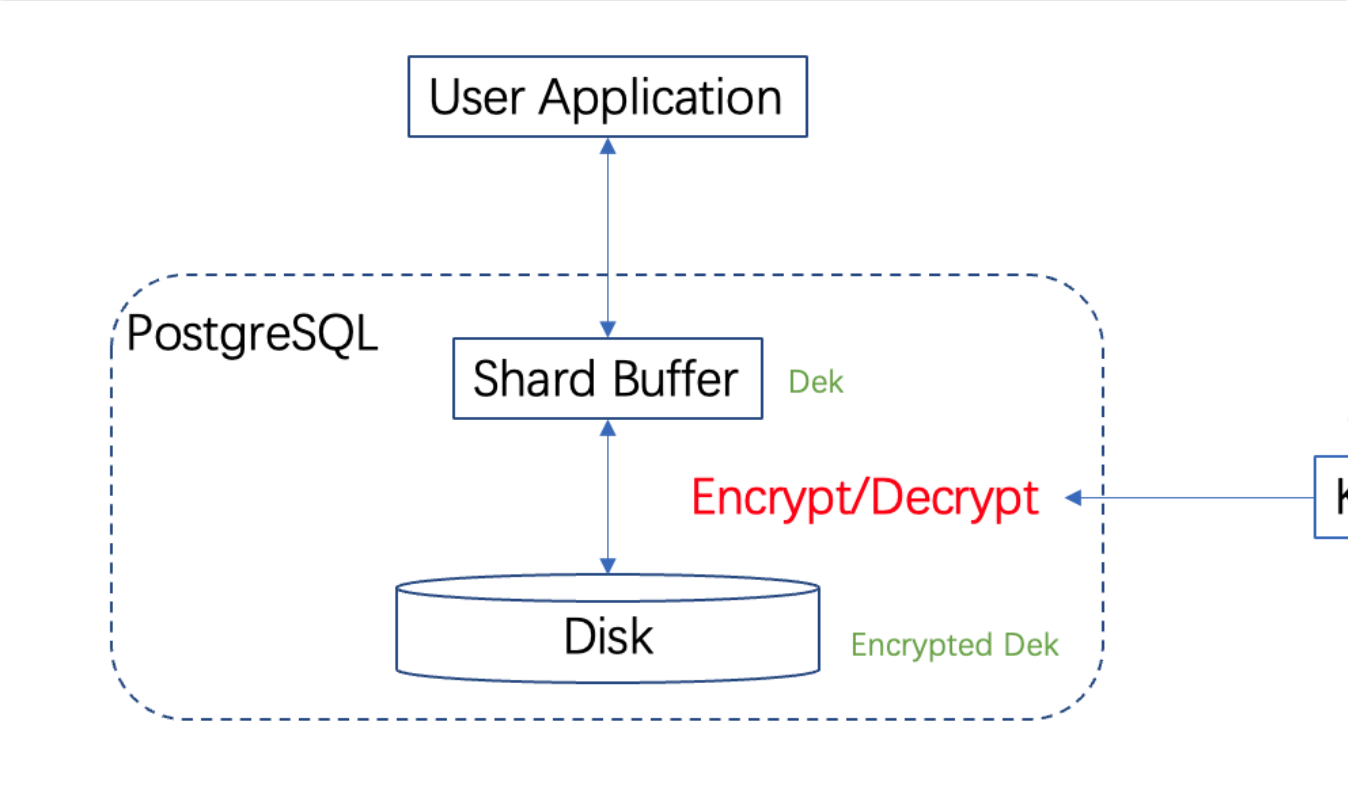
云数据库 PostgreSQL 加密实现方案

腾讯云数据库 PostgreSQL 通过向用户申请使用 KMS（Key Management Service）服务中保存的主密钥，生成 DEK（Data Encryption Key）密文与 DEK 明文对云产品加密所使用的密钥进行数据加密和解密。



此类加密方案被称为信封加密，即用另一个密钥对密钥进行加密的过程。信封加密是一种应对海量数据的高性能加解密方案。其可通过生成 DEK 来对本地数据进行加解密，保证了业务加密性能的要求，同时也由 KMS 确保了数据密钥的随机性和安全性。

所有的加密解密操作均由数据库在内存中进行，每一次数据库重启以及存在关闭内存的操作时，均会重新从 KMS 获取密钥材料。本地存储中不保存任何可用于解密的密钥材料。



开启透明数据加密

最近更新时间：2024-01-22 16:22:57

操作场景

云数据库 PostgreSQL 提供透明数据加密（Transparent Data Encryption，TDE）功能，透明加密指数据的加解密操作对用户透明，支持对数据文件进行实时 I/O 加密和解密，在数据写入磁盘前进行加密，从磁盘读入内存时进行解密，可满足静态数据加密的合规性要求。

限制条件

TDE 加密功能仅可在创建实例时开通，且开通后无法关闭。
内核版本为 PostgreSQL v10.17_r1.2、v11.12_r1.2、v12.7_r1.2、v13.3_r1.2、v14.2_r1.0 才支持加密功能。
须先开通密钥管理服务 KMS。如未开通，可通过 开通 KMS 提前购买 KMS 密钥管理服务。
若使用子账号进行操作，必须要创建用于授权云数据库 PostgreSQL 操作 KMS 的服务角色，可使用主账号访问 此链接 进行角色创建。
子账号必须具有“cam:PassRole”，“kms:GetServiceStatus”，“kms:GetRegions”的权限，如无权限，可使用主账号为操作账号进行赋予。
说明：
加密使用的密钥由 密钥管理服务 KMS 产生和管理，云数据库 PostgreSQL 不提供加密所需的密钥和证书。
透明数据加密 TDE 功能不会额外收费，但密钥管理服务 KMS 有可能产生额外费用，请参见 计费概述。
当账号处于欠费状态时，无法从 KMS 获取密钥，可能导致迁移、升级等任务无法正常进行，请参见 欠费说明。

注意事项

透明数据加密功能开启后不可关闭，如果解除了密钥授权，重启数据库会导致不可用。
开启 TDE 加密功能后，数据备份也会被同时加密，当备份文件被泄露时，无需担心数据泄露。若需要通过备份恢复数据，请使用云数据库 PostgreSQL 克隆实例 功能。
开启 TDE 加密功能后，可提高静态数据的安全性，但同时会影响访问加密数据库的读写性能，请结合实际情况选择开启 TDE 加密功能。根据实际测试平均损耗在2% - 3%。
如果主实例关联只读实例，只读实例会自动打开加密功能，且无法被管理。
开启 TDE 加密功能后，账户余额需大于等于0，否则会因为无法访问密钥管理系统而导致实例迁移失败。
为了避免实例误删除等意外场景，腾讯云特设置了密钥删除保护措施，若实例配置了 数据加密，则在实例隔离与实例下线后不会立即解绑密钥，直到实例在回收站中下线后的第三天后才支持从 KMS 中 删除密钥。

操作步骤

1. 登录 PostgreSQL 购买页，在**是否开启加密**选项中，开启数据库加密功能。
2. 在弹出的对话框，选择密钥，单击**确认加密**。

注意：

启用数据加密功能的实例，不支持使用物理备份恢复至其他主机上的自建数据库。

数据加密开通后不可关闭。

KMS 服务：若未开启 KMS 密钥管理服务，则需要 购买 KMS 密钥管理服务。

KMS 密钥授权：若提示未授权，可通过单击**授权链接**，进入角色授权页面，使得云数据库 PostgreSQL 可以使用服务角色操作 KMS 密钥管理服务。

选择密钥：

根据实例地域选择 KMS 密钥管理服务的地域，若提示“可选的 KMS 地域为空”，则代表当前地域不支持 KMS 服务，无法开启加密。

选择**使用腾讯云自动生成的密钥**时，由腾讯云自动生成密钥。

选择**使用已有自定义密钥（BYOK）**时，可选择自己创建的密钥。

说明：

如无自定义密钥，可单击**前往创建**，在密钥管理系统控制台创建密钥，详情请参见 创建密钥。

租户及资源隔离
DataBase 资源隔离

最近更新时间：2024-08-09 15:51:43

使用场景

在一些业务场景下，数据库对象 database 对应相关的业务逻辑，典型的如 SaaS 场景下database 对应租户，因此需要解决 database 资源隔离的问题。本文主要讲述如何在云数据库 PostgreSQL 实例中实现 database 的 CPU 隔离。

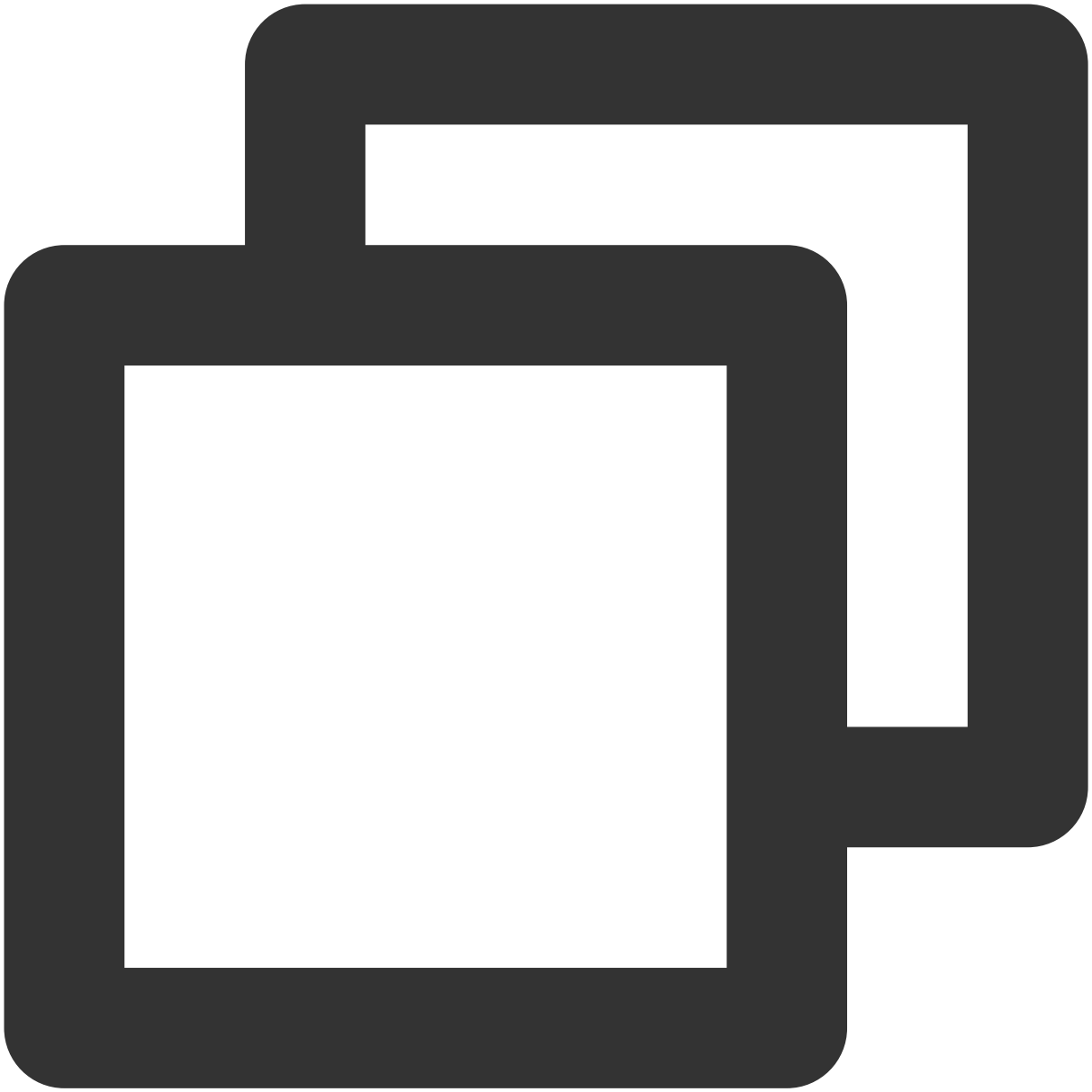
设置 database 资源隔离

说明：

当前只有 v14.11_r1.21及以上的 PostgreSQL 14的内核版本支持资源隔离能力。如需开启 database 资源隔离模式，请 提交工单 联系我们开启插件 tencentdb_serverless 并设置相关参数。

其中 tencentdb_serverless.min_cpu_cores 为实例可设置的最小 CPU 核数，tencentdb_serverless.max_cpu_cores 为实例可设置的最大 CPU 核数。这两个参数主要用于后台管理系统控制实例中 database 的资源隔离，用户不需要修改。

当您已经开启 tencentdb_serverless 插件并设置了相关插件参数 tencentdb_serverless.min_cpu_cores 和 tencentdb_serverless.max_cpu_cores 后，可以正式开始配置。您可以通过如下命令检查插件已经安装成功：



```
postgres=> \\dx;
                                                  List of installed extensions
         Name           | Version |   Schema   |                                Descri
------------------------+---------+------------+-----------------------------------
 pg_stat_log            | 1.0     | public     | track runtime execution statistics
 pg_stat_statements     | 1.9     | public     | track planning and execution statis
 plpgsql                | 1.0     | pg_catalog | PL/pgSQL procedural language
 tencentdb_serverless   | 1.0     | public     | extension for serverless mode
 tencentdb_system_stat  | 1.0     | public     | track execution statistics of quers
(5 rows)
```

tencentdb_serverless.min_cpu_cores 和 tencentdb_serverless.max_cpu_cores 参数的初始值为实例当前核心数，这两个参数主要用于后台管理系统控制实例中 database 的资源隔离，用户不需要修改。后续如果实例发生变配，则此二参数会随之改变。您可以基于如下命令来检查参数的当前值。

```
postgres=> show tencentdb_serverless.min_cpu_cores;
 tencentdb_serverless.min_cpu_cores
------------------------------------
 8
(1 row)

postgres=> show tencentdb_serverless.max_cpu_cores;
```

```
 tencentdb_serverless.max_cpu_cores
------------------------------------
 8
(1 row)
```

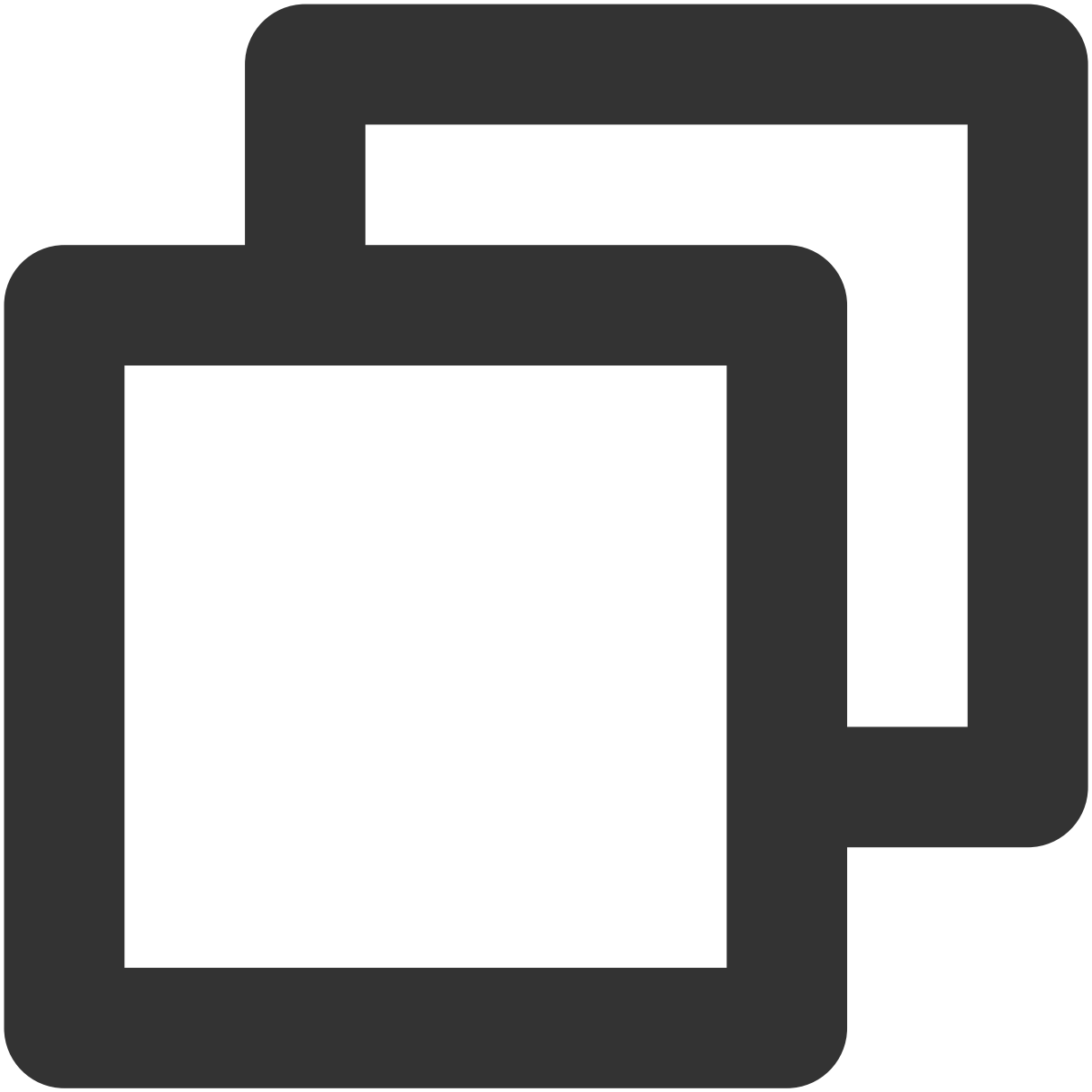
说明：

当实例中有多个 database 对象时，需要每个 database 都设置 CPU 资源限制才能使配置生效。

插件和参数检查完毕之后，您可以开始设置 database 的 CPU 资源上限和下限。我们提供了相应的函数或视图供您调用。分别说明如下：

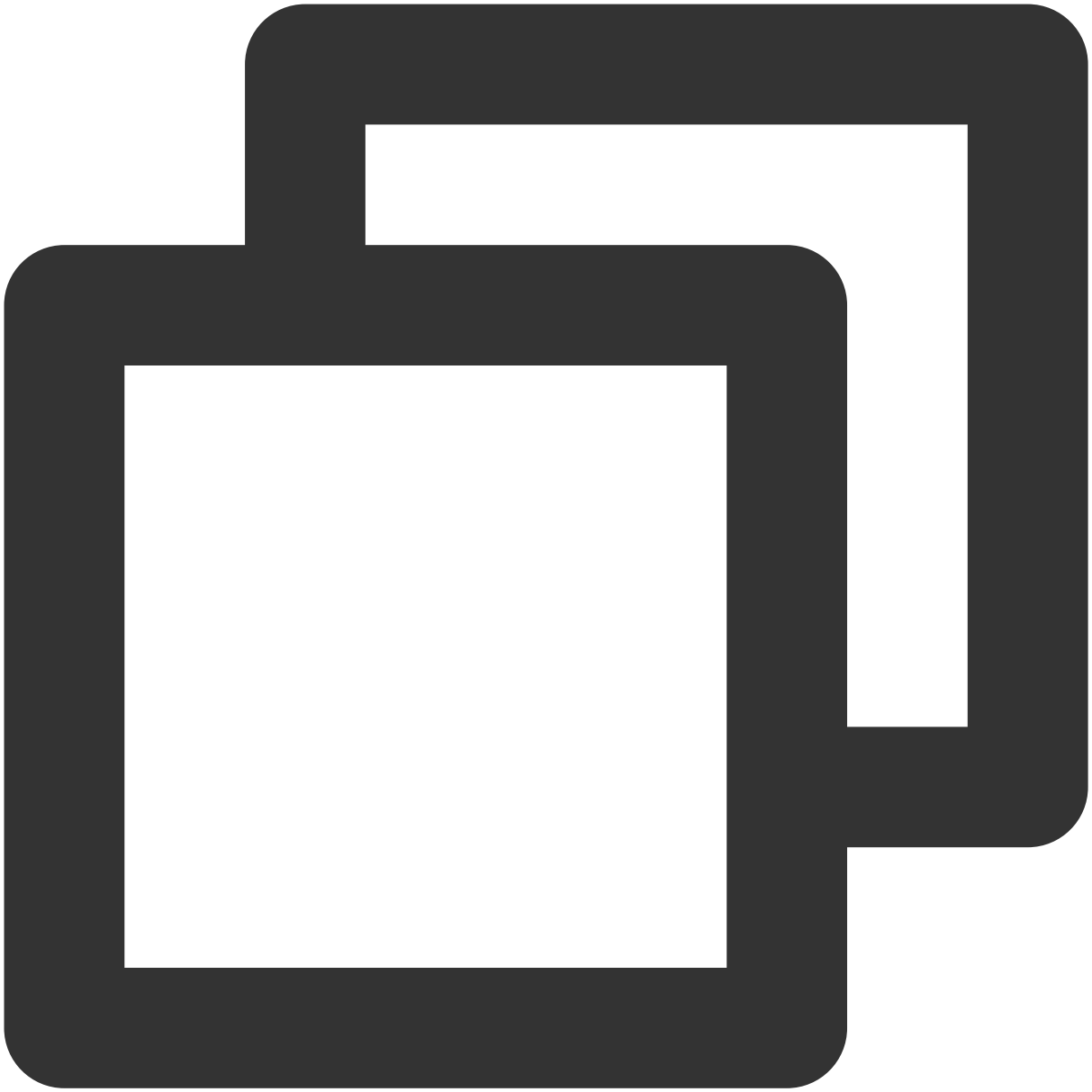
设置指定 database 的 CPU 资源限制

函数定义如下：



```
tencentdb_serverless.set_database_cpu_limit(database_name text [, min_cpu_cores num
```

调用示例：

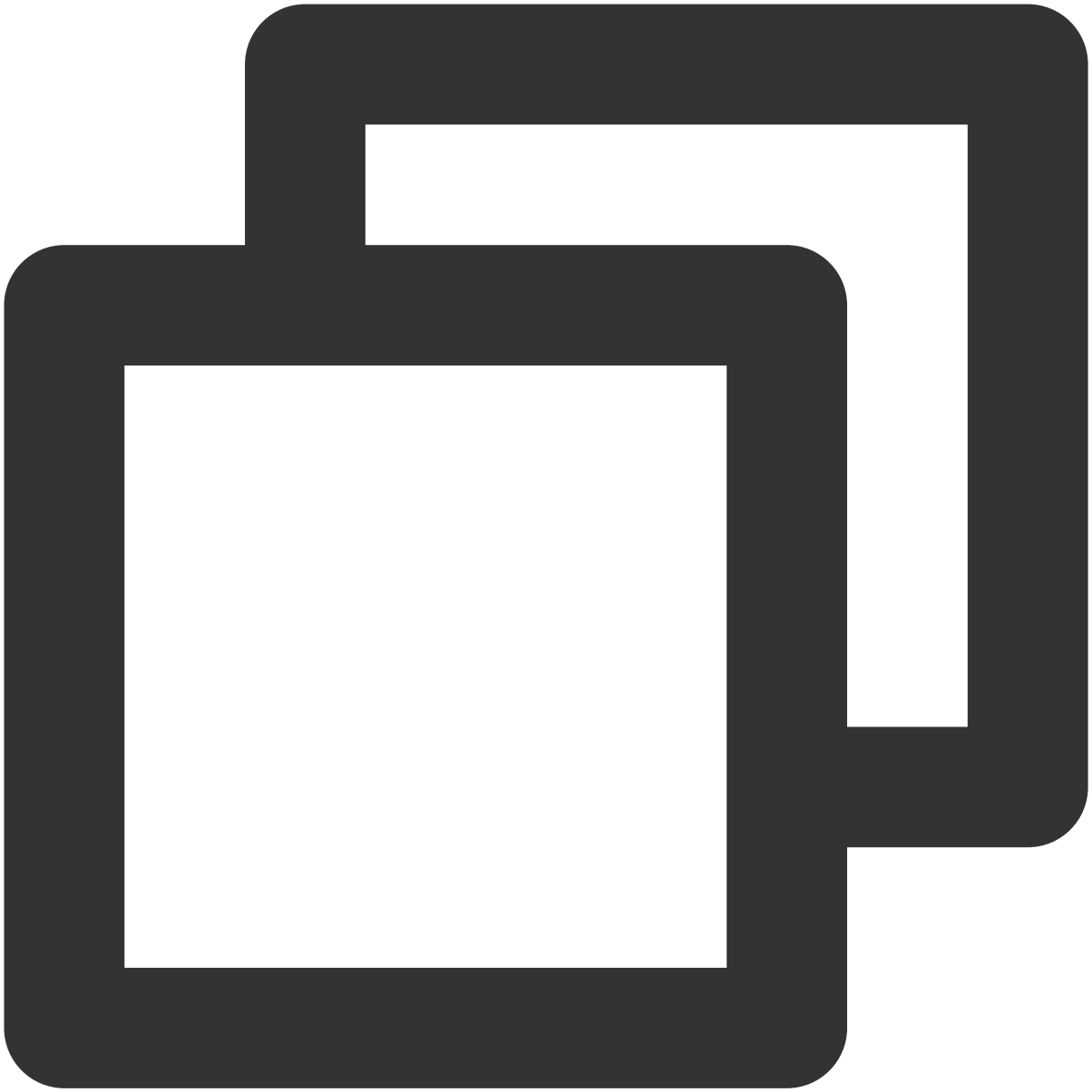


```
postgres=> select tencentdb_serverless.set_database_cpu_limit('tenant_001',2,2.5);
 set_database_cpu_limit
------------------------

(1 row)
```

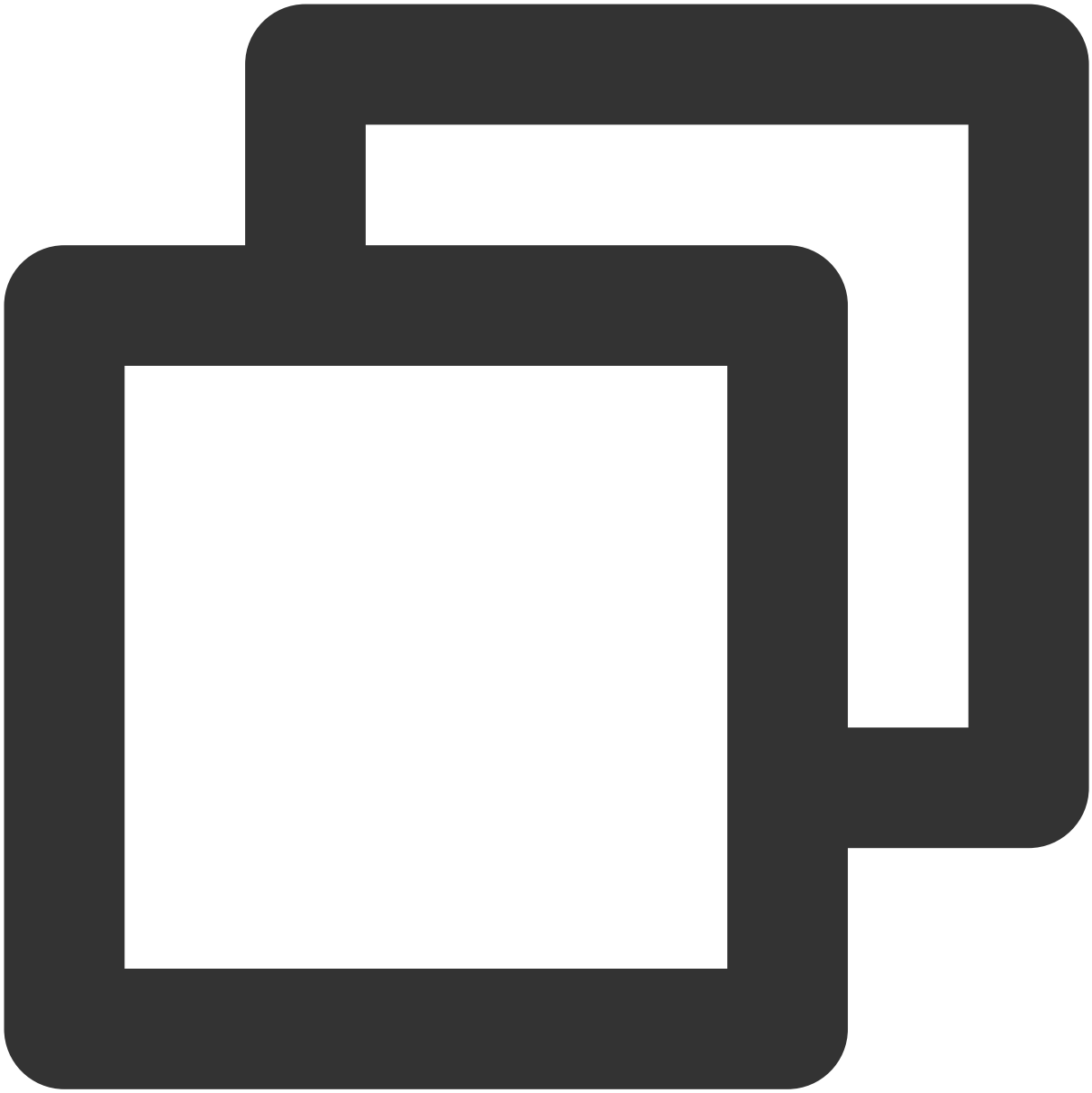
清除指定 database 的 CPU 资源限制

函数定义如下：



```
tencentdb_serverless.reset_database_limit(database_name text)
```

调用示例：

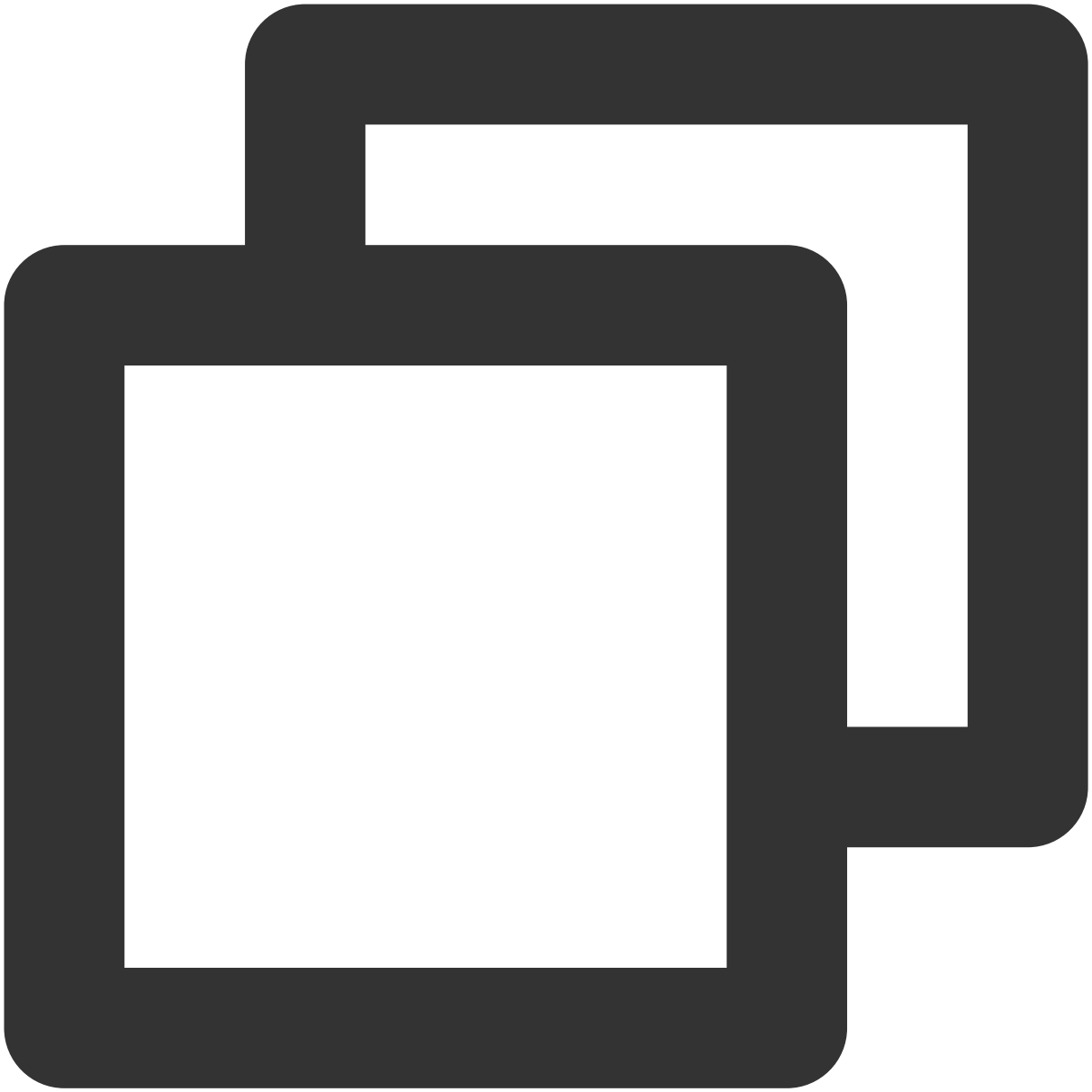


```
postgres=> select tencentdb_serverless.reset_database_limit('tenant_001');
 reset_database_limit
----------------------

(1 row)
```

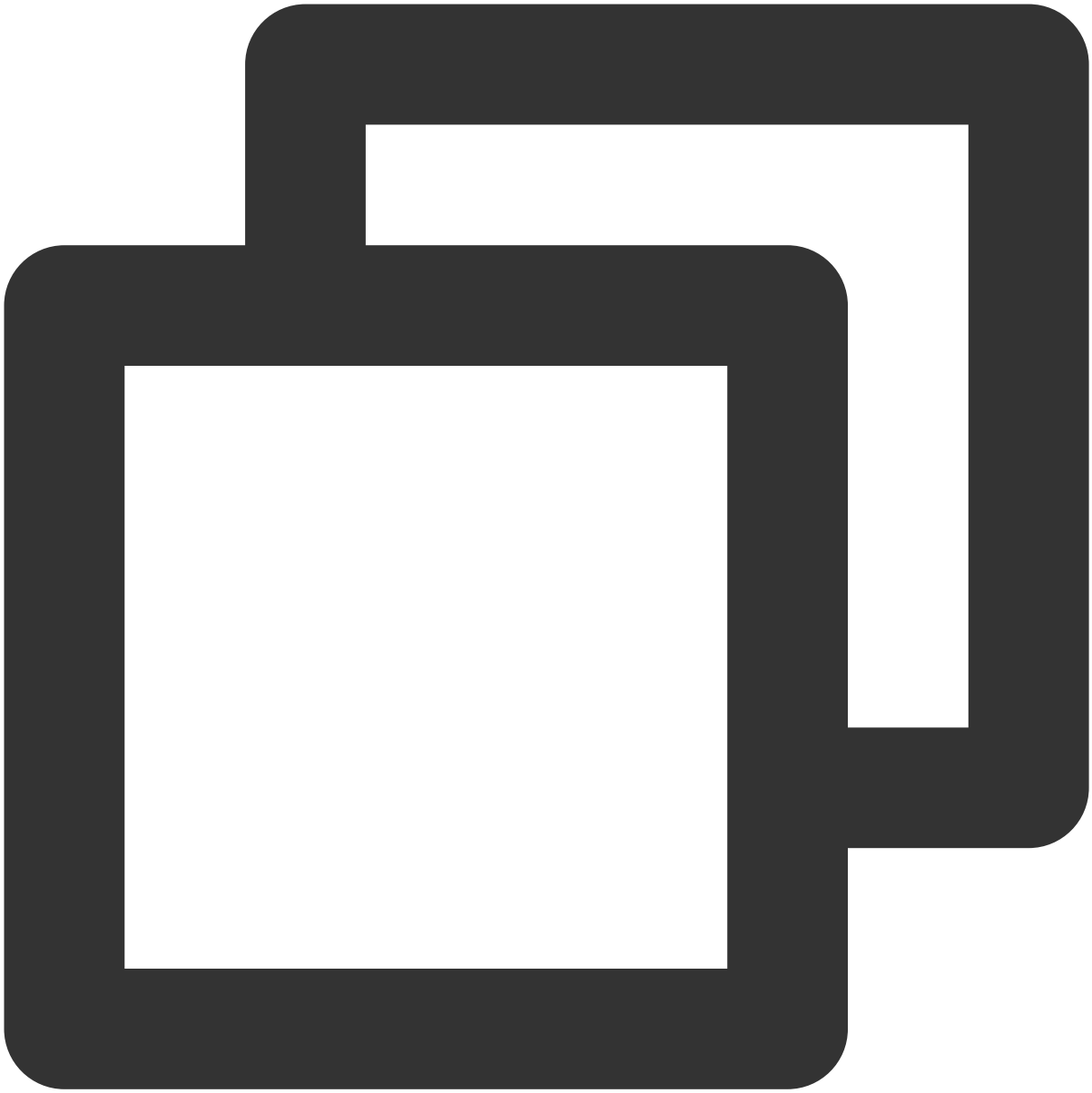
清空实例中所有 database 的 CPU 资源限制

函数定义如下：



```
tencentdb_serverless.reset_all_database_limit()
```

调用示例：



```
postgres=> select tencentdb_serverless.reset_all_database_limit();
 reset_all_database_limit
--------------------------

(1 row)
```

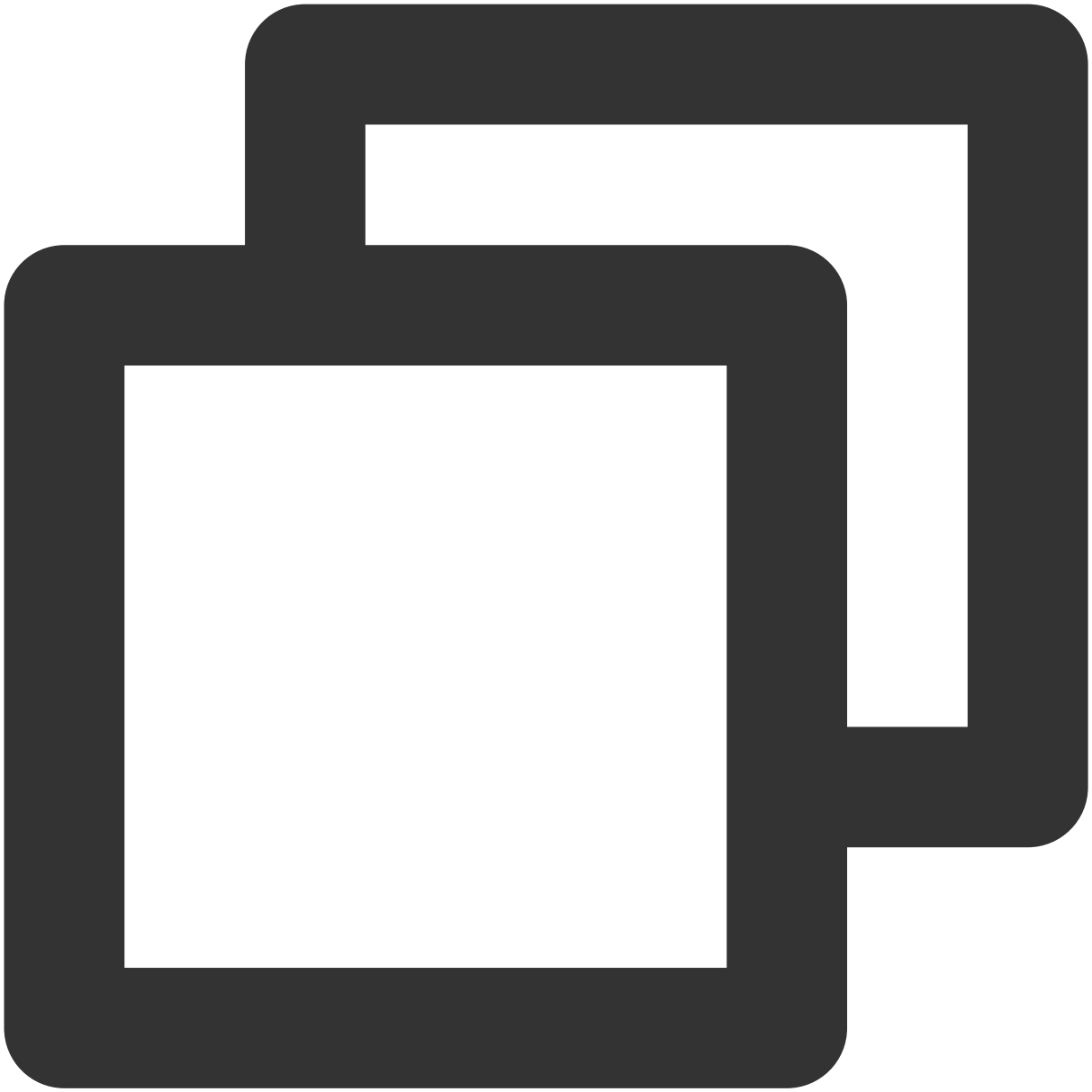
查看当前实例中所有已配置的 CPU 资源限制详情

我们提供视图 tencentdb_serverless.resource_limit_view，用于您查看当前实例中所有的已配置的 CPU 资源限制详情，字段定义如下：

列名	含义

database_name	数据库的名字
min_cpu_cores	当前 database 可以使用的最小 CPU 核数
max_cpu_cores	当前 database 可以使用的最大 CPU 核数
min_mem_kilobytes	当前 database 可以使用的最大内存大小，单位为 kB。**预留字段，暂时未使用**。
max_mem_kilobytes	当前 database 可以使用的最大内存大小，单位为 kB。**预留字段，暂时未使用**。

调用示例：



```
postgres=> select * from tencentdb_serverless.resource_limit_view;
 database_name | min_cpu_cores | max_cpu_cores | min_mem_kilobytes | max_mem_kiloby
---------------+---------------+---------------+-------------------+---------------
 tenant_001    |           2.0 |           2.5 |                   |
 tenant_002    |           2.0 |           2.5 |                   |
(2 rows)
```

进程监控

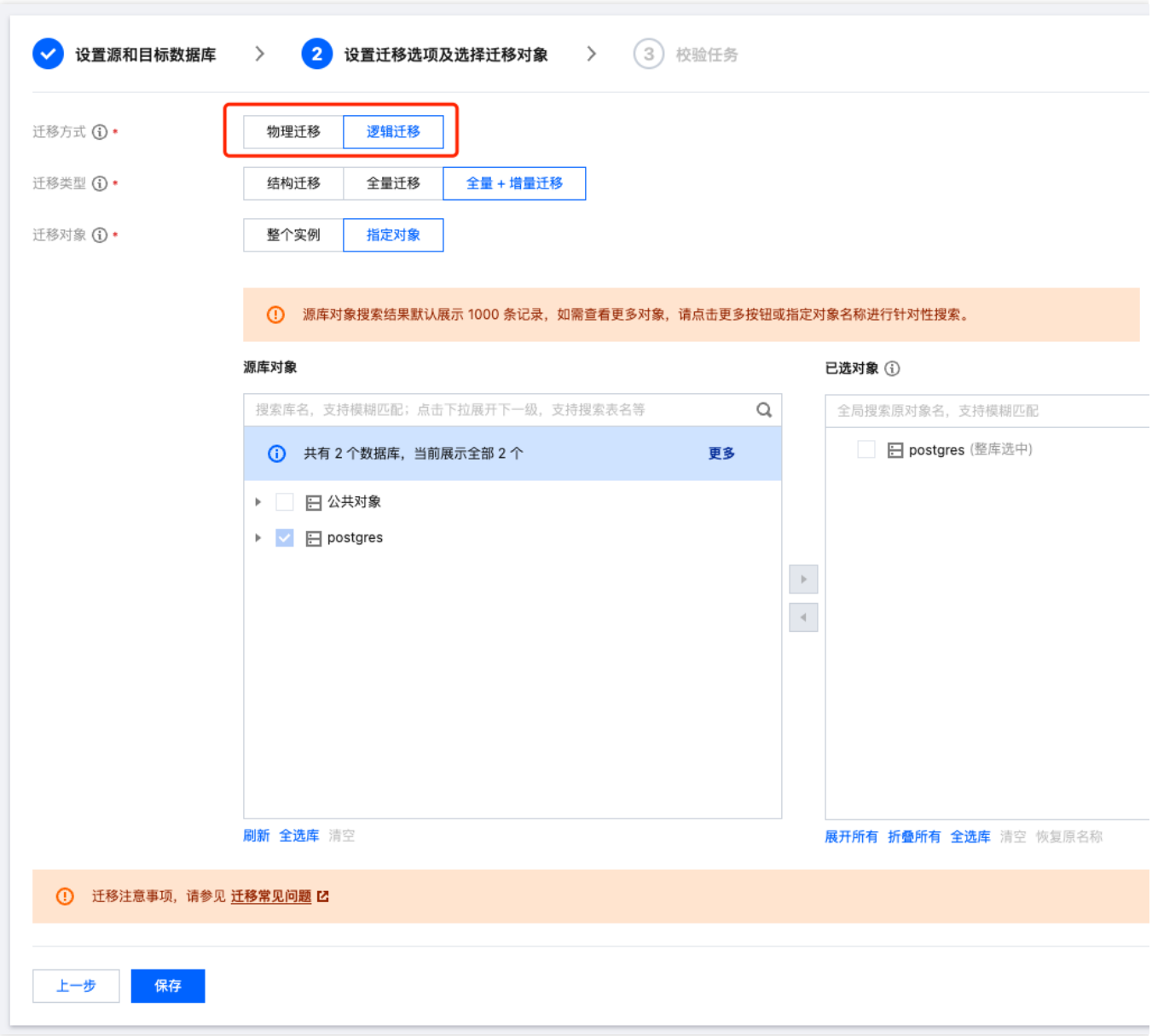
当实例中所有 database 配置了 CPU 资源隔离后，如果实例整体资源利用率比较高，则每个 database 能保证最低配置的 CPU 核数的使用。同时，如您需要查看当前系统中哪些 database 使用的资源较多，则可以使用进程监控 能力。查看当前所有 database 的 CPU 资源使用详情可以使用如下语句：

```
postgres=> select datname,sum(cpu_usage) as cpu_usage from tencentdb_process_system
  datname    | cpu_usage
------------+-----------
 postgres    |         3
 tenant_001 |      1.99
 tenant_002 |         1
(3 rows)
```

当进程监控发现某些 database 资源使用率比较高时，您可以实时调整该 database 的 CPU 的配置，该配置可以**实时生效**。

资源迁移

当 database 对应租户，且我们通过进程监控发现该 database 的资源使用一直很高需要进行资源再整合时，云数据库 PostgreSQL 提供了数据迁移能力，您可以配置云数据库 PostgreSQL 的 逻辑迁移。配置迁移任务如下图所示：



关闭 database 资源隔离

如您需要关闭 database 的 CPU 资源隔离模式，请 提交工单 联系我们进行清理数据。后端工程师收到工单后，会重置所有的 CPU 资源参数tencentdb_serverless.min_cpu_cores 和 tencentdb_serverless.max_cpu_cores ，取消所有的资源配置，最后会删除插件 tencentdb_serverless。

安全组
管理安全组

最近更新时间：2024-01-22 16:22:57

操作场景

安全组 是一种有状态的包含过滤功能的虚拟防火墙，用于设置单台或多台云数据库的网络访问控制，是腾讯云提供的重要的网络安全隔离手段。安全组是一个逻辑上的分组，您可以将同一地域内具有相同网络安全隔离需求的云数据库实例加到同一个安全组内。云数据库与云服务器等共享安全组列表，安全组内基于规则匹配，具体规则与限制请参见 安全组详细说明。

说明：

云数据库 PostgreSQL 安全组目前仅支持私有网络 VPC 内网访问和外网访问的网络控制，暂不支持对基础网络的网络控制。

数据库外网访问安全组功能当前仅 北京、上海、广州、成都地域支持。

由于云数据库没有主动出站流量，因此出站规则对云数据库不生效。

云数据库 PostgreSQL 安全组支持主实例、只读实例与只读实例组。

配置安全组

步骤一：创建安全组

1. 登录 云服务器控制台。
2. 在左侧导航选择**安全组**页，选择地域，单击**新建**。
3. 在弹出来的对话框中，完成如下配置，确认后单击**确定**即可。

模板：根据安全组中的数据库实例需要部署的服务，选择合适的模板，简化安全组规则配置。如下表所示：

模板	说明	场景
放通全部端口	默认放通全部端口到公网和内网，具有一定安全风险。	-
放通22，80，443，3389端口和 ICMP 协议	默认放通22，80，443，3389端口和 ICMP 协议，内网全放通。	此模板对云数据库不生效。
自定义	安全组创建成功后，按需自行添加安全组规则。具体操作请参见下文“添加安全组规则”。	建议选择此模板，自定义一个关于访问指定实例的模板。

名称：自定义设置安全组名称。

所属项目：默认选择“默认项目”，可指定为其他项目，便于后期管理。

备注：自定义，简短地描述安全组，便于后期管理。

步骤二：添加安全组规则

1. 在 安全组页，在需要设置规则的安全组行中，单击**操作**列的**修改规则**。
2. 在安全组规则页面，选择**入站规则>添加规则**。
3. 在弹出的对话框中，设置规则。

类型：默认选择“自定义”。

来源：流量的源（入站规则） 或目标（出站规则），请指定以下选项之一：

指定的源/目标	说明
单个 IPv4 地址或 IPv4 地址范围	用 CIDR 表示法（如203.0.113.0、203.0.113.0/24或者0.0.0.0/0，其中0.0.0.0/0代表匹配所有 IPv4 地址）。
单个 IPv6 地址或 IPv6 地址范围	用 CIDR 表示法（如FF05::B5、FF05:B5::/60、::/0或者0::0/0，其中::/0或者0::0/0代表匹配所有 IPv6 地址）。
引用安全组 ID，您可以引用以下安全组的 ID： 安全组 ID 其他安全组	当前安全组表示与安全组关联的云服务器。 其他安全组表示同一区域中同一项目下的另一个安全组 ID。
引用 参数模板 中的 IP 地址对象或 IP 地址组对象	-

协议端口：填写协议类型和端口范围，您也可以引用 参数模板 中的协议端口或协议端口组。

说明：

连接云数据库 PostgreSQL 须开通5432协议端口。

策略：默认选择“允许”。

允许：放行该端口相应的访问请求。

拒绝：直接丢弃数据包，不返回任何回应信息。

备注：自定义，简短地描述规则，便于后期管理。

4. 单击**完成**，完成安全组入站规则的添加。

案例

场景：您创建了一台云数据库 PostgreSQL，并希望通过云服务器 CVM 访问云数据库 PostgreSQL。

解决方法：添加安全组规则时，在入站规则中配置 TCP:5432协议端口的访问权限。

您还可以根据实际需求，放通全部 IP 或指定 IP（IP 段），配置可通过 CVM 访问云数据库 PostgreSQL 的 IP 来源。

方向	类型	来源	协议端口	策略
入方向	自定义	全部 IP：0.0.0.0/0 指定 IP：输入您指定的 IP 或 IP 段	TCP:5432	允许

导入安全组规则

1. 在 安全组页，选择需要的安全组，单击安全组 ID/名称。
2. 在入/出站规则页签，单击**导入规则**。
3. 在弹出的对话框，选择已编辑好的入站/出站规则模板文件，单击**开始导入**。

说明：

如果需要导入规则的安全组下已存在安全组规则，建议您先导出现有规则，否则导入新规则时，将覆盖原有规则。

克隆安全组

1. 在 安全组页，在列表的**操作**列选择**更多>克隆**。
2. 在弹出的对话框，选定目标地域、目标项目后，单击**确定**。若新安全组需关联 CVM，请重新进行管理安全组内云服务器。

删除安全组

1. 在 安全组页，选择需要删除的安全组，在**操作**列选择**更多>删除**。
2. 在弹出的对话框，单击**确定**。若当前安全组有关联的 CVM 则需要先解除安全组才能进行删除。

关联实例至安全组

最近更新时间：2024-01-22 16:22:57

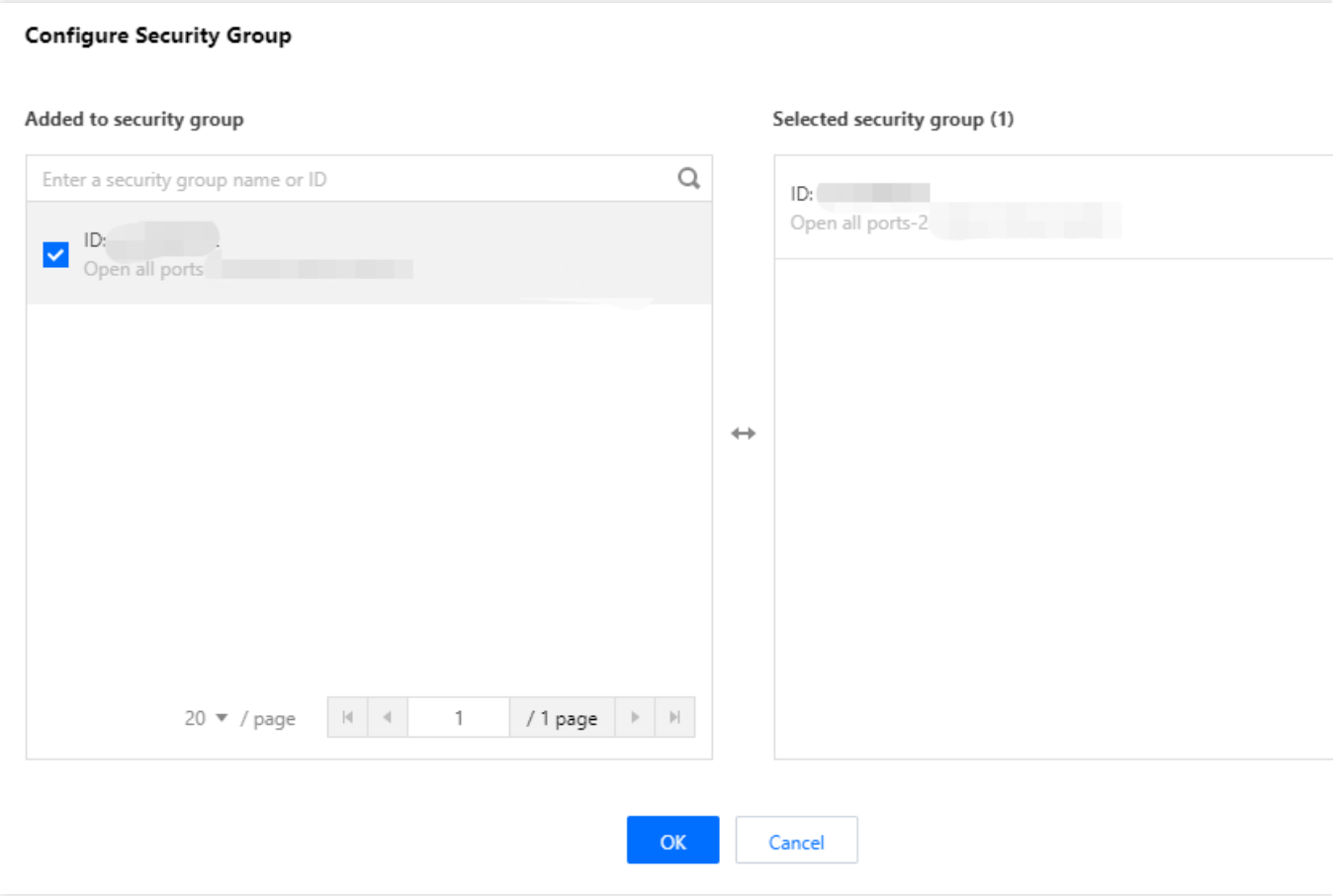
安全组是腾讯云提供的实例级别防火墙，可以对云数据库进行入流量控制。您可以在购买实例时绑定安全组，也可以购买实例后在控制台绑定安全组。
云数据库 PostgreSQL 支持为主实例、只读实例、只读实例组绑定安全组，安全组互相之间不会造成影响。
说明：
云数据库 PostgreSQL 一个实例对象仅支持绑定最多5个安全组。
对只读实例组绑定安全组，不会对只读实例自身的访问地址的连接产生影响，仅影响通过只读实例组的访问地址来对实例进行访问的连接。如设置只读实例的安全组为拒绝某一 IP 访问，但是在只读实例组的安全组设置为允许此 IP 访问；那么用户通过只读实例组的访问地址进行访问时，是可以正常访问到只读实例的；而通过只读实例自身的访问地址将会被拒绝。

前提条件

参考 管理安全组，在安全组控制台中为实例创建安全组。

为主实例/只读实例绑定安全组

1. 登录 PostgreSQL 控制台，在实例列表，单击需要绑定安全组的实例 ID，进入实例管理页面。
2. 在实例管理页面，选择**安全组**页，单击**配置安全组**。
3. 在弹出的对话框，选择需要绑定的安全组，单击**确定**，即可完成安全组绑定云数据库的操作。



为只读实例组绑定安全组

1. 登录 PostgreSQL 控制台，在实例列表，找到只读实例组管理的只读实例，单击实例 ID，进入只读实例管理页面。
2. 在只读实例管理页面，选择**安全组**页，在“安全组生效对象”中选择对象类型为“只读实例组”，然后再单击**配置安全组**。
3. 在弹出的对话框，选择需要绑定的安全组，单击**确定**，即可完成安全组绑定只读实例组的操作。

调整安全组优先级

一个云数据库实例可以绑定最多5个安全组，当实例绑定多个安全组时，多个安全组将按照优先级顺序（如1、2）依次匹配执行，您可以根据以下操作调整安全组的优先级。

1. 登录 PostgreSQL 控制台，在实例列表，单击实例 ID，进入实例管理页面。
2. 在实例管理页面，选择**安全组**页。
3. 在“已加入安全组”模块中，单击**编辑**，选中安全组右侧的上移和下移按钮，调整安全组的优先级，位置越靠上，安全组的优先级越高，所有安全组配置取并集。如设置两个互相冲突的安全组配置，则以优先级高的为准。
4. 完成调整后，单击**保存**即可。

监控与告警
监控功能

最近更新时间：2024-01-22 16:24:46

控制台监控

为方便用户查看和掌握实例的运行信息，云数据库 PostgreSQL 提供了丰富的性能监控项，用户可登录 PostgreSQL 控制台，在对应实例管理页的**系统监控**页查看。

监控指标

指标名	监控指标名	单位	指标描述
CPU 利用率	cpu	%	CPU 实际利用率
已用存储空间	storage	GB	整实例空间占用大小
数据文件大小	data_file_size	GB	数据文件占用空间大小
日志文件大小	log_file_size	GB	wal 日志文件占用空间大小
临时文件大小	temp_file_size	MB	临时文件的大小
存储空间使用率	storage_rate	%	总的存储空间使用率，包括临时文件、数据文件、日志文件以及其他类型的数据库文件
每秒查询数	qps	次/秒	平均每秒执行的 SQL 语句数量
连接数	connections	个	对数据库发起采集时，数据库当前总连接数
5秒内新建连接数	new_conn_in5s	次	对数据库发起采集时，查询出关于最近5秒内建立的所有连接数
活跃连接数	active_conns	个	对数据库发起采集时，数据库瞬时活跃连接（非 idle 连接）
空闲连接数	idle_conns	个	对数据库发起采集时，查询出来的数据库瞬时空闲连接（idle 连接）

等待会话数	waiting	个	对数据库发起采集时，数据库正在等待的会话数量（状态为 waiting）
等待超过5秒的会话数	long_waiting	个	一个采集周期内，数据库等待超过5秒的会话数量（状态为 waiting，且等待状态维持了5秒）
空闲事务数	idle_in_xact	个	对数据库发起采集时，数据库正在处于 idle 状态的事务数量
执行时长超过1秒的事务数目	long_xact	个	一个采集周期内，执行时间超过1秒的事务数量
超过5秒的空闲事务数	long_idle_in_xact	个	一个采集周期内，空闲时间超过5秒的事务数量
每秒事务数	tps	次/秒	平均每秒执行成功的事务数（包括回滚和提交）
事务提交数	xact_commit	次/秒	平均每秒提交的事务数
事务回滚数	xact_rollback	次/秒	平均每秒回滚的事务数
请求数	read_write_calls	次	一个统计周期内的总请求数
读请求数	read_calls	次	一个统计周期内的读请求数
写请求数	write_calls	次	一个统计周期内的写请求数
其他请求数	other_calls	次	一个统计周期内的其他请求数（begin、create、非 DML、DDL、DQL 操作）
缓冲区缓存命中率	hit_percent	%	一个请求周期内的所有 SQL 语句执行的命中率
平均执行时延	sql_runtime_avg	ms	一次统计周期内所有 SQL 语句的平均执行时延
最长 TOP10 执行时延	sql_runtime_max	ms	一次统计周期内最长 TOP10 的 SQL 平均执行时延
最短 TOP10 执行时延	sql_runtime_min	ms	一次统计周期内最短 TOP10 的 SQL 平均执行时延
剩余 XID 数	remain_xid	个	对数据库发起采集时，显示当前剩余 xid 最少的库的剩余 xid 数

版权所有：腾讯云计算（北京）有限责任公司

量			量。只读实例无此指标
备库日志发送与回放位置差异	xlog_diff	byte	日志从主库发送至备库与备库回放完成之间的大小差异，主要反映了备库日志应用的速度，主要能够通过此指标查看出备库的性能、网络传输的速度。只读实例无此指标
备库日志落盘时间延迟	xlog_diff_time	byte	日志从主库发送至备库与备库接收到日志并落盘之间的时间差异。只读实例无此指标，且实例版本在10.x以上版本才有此指标
主备数据同步延迟	slave_apply_delay	s	主备数据同步延迟,对于主实例而言，此指标可以体现出故障切换的 RTO。反映到只读实例在多少时间后能够查询到在主库进行写入的数据. 只读实例指标名为：与主实例数据同步延迟
慢查询数量	slow_query_cnt	个	一个采集周期内，出现的慢查询个数
执行时长超过1秒的 SQL 数	long_query	个	对数据库发起采集时，查询出来 执行时间超过1s的 SQL 数量
2pc 事务数	2pc	个	对数据库发起采集时，当前的2pc事务数量
超过5s未提交的2PC 事务数	long_2pc	个	对数据库发起采集时，当前执行时间超过5s的 2PC 事务数量
每秒删除记录数	tup_deleted	个	平均每秒删除的 tupe 数量。只读实例无此指标
每秒插入记录数	tup_inserted	个	平均每秒插入的 tupe 数量。只读实例无此指标
每秒更新记录数	tup_updated	个	平均每秒更新的 tupe 数量。只读实例无此指标
每秒索引扫描回表记录数	tup_fetched	个	平均每秒通过索引扫描的 tupe 数量
每秒全表扫描记录数	tup_returned	个	平均每秒全表扫描的 tupe 数量
死锁数	deadlocks	个	在一个采集周期内的所有死锁数

版权所有：腾讯云计算（北京）有限责任公司

告警功能

最近更新时间：2024-01-22 16:24:46

操作场景

告警功能用于在云数据库状态改变时触发警报并发送相关消息。创建的告警会根据每隔一段时间监控的指标相对于给定阈值的情况判断是否需要触发相关通知。
状态改变触发告警后，您可以及时进行相应的预防或补救措施。因此，合理地创建告警能帮助您提高应用程序的健壮性和可靠性。有关告警的更多信息，请参见云监控的 告警配置。
当用户需要针对某个产品的某个状态发送告警时，需要先创建告警策略。告警策略包括名称、类型和告警触发条件三个必要组成部分。每个告警策略是一系列告警触发条件的集合，告警触发条件是“或”关系，即一个条件满足，就会发送告警。告警将发送至告警策略关联的所有人，用户接收到告警后可以及时查看并采取相应措施。
注意：
请确认您已经设置默认告警接收人，否则腾讯云云数据库的默认告警策略将无法通知到您。

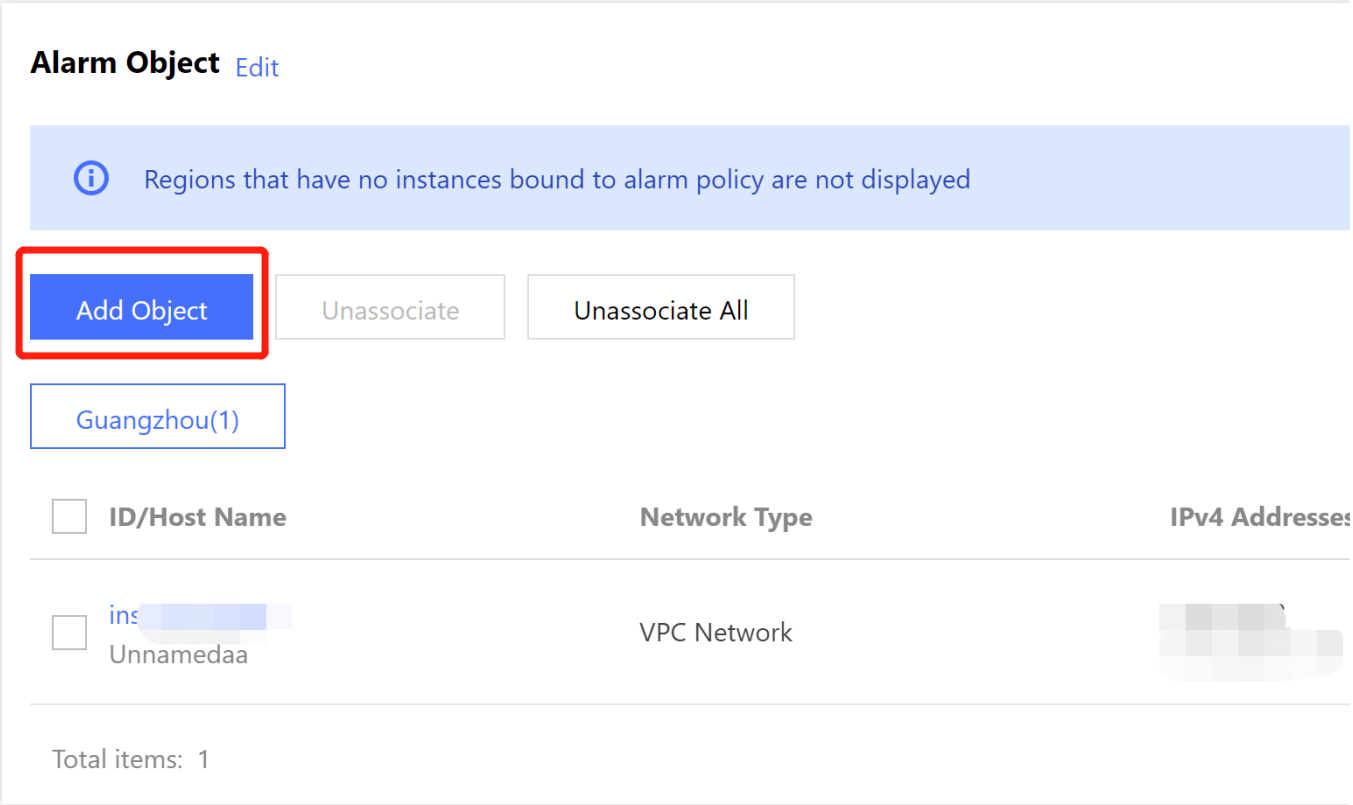
操作步骤

创建告警策略

1. 登录 云监控控制台，在左侧导航选择**告警配置>告警策略**页。
2. 在告警策略列表页中，单击**新建**。
3. 在新建策略页中，设置策略名称、策略类型、告警对象、触发条件等内容。
告警触发条件是指标、比较关系、阈值、统计周期和持续周期组成的一个有语义的条件。例如配置触发条件：指标为磁盘使用率、比较关系为>、阈值为80%、统计周期为5分钟、持续周期为2个周期。则表示：云监控每5分钟收集一次磁盘使用率数据，若某台云数据库的磁盘使用率连续两次大于80%则触发告警。
可通过选择对象所在的地域或搜索对象的实例 ID 找到需要关联的对象实例。
4. 确认无误后，单击**完成**即可。

关联告警对象

创建完告警策略后，您可以为其关联一些告警对象，对象达到告警触发条件时会发送告警。
1. 在 告警策略列表页，单击告警策略名称，进入管理告警策略页。
2. 在管理告警策略页的“告警对象”栏，单击**新增对象**。



3. 在弹出的对话框，选择您需要关联的告警对象，单击**确定**，即可关联告警对象。

标签

标签概述

最近更新时间：2024-01-22 16:24:46

简介

标签是腾讯云提供的用于标识云上资源的标记，是一个键-值对（Key-Value），具体介绍请参见 标签概述。
您可以根据各种维度（例如，业务、用途、负责人等）使用标签对云数据库 PostgreSQL 资源进行分类管理。通过标签可以非常方便地筛选过滤出对应的资源。标签键值对对腾讯云没有任何语义意义，会严格按字符串进行解析匹配，在使用过程中只需要注意 使用限制 即可。
下面通过一个具体的案例来介绍标签的使用。

案例背景

某公司在腾讯云上拥有3个云数据库 PostgreSQL 实例，分属3个不同的业务，三个不同游戏业务对应的运维负责人为张三、文四、王五。

设置标签

为了方便管理，该公司使用标签来分类管理对应的云数据库 PostgreSQL 实例，定义了下述标签键/值。

标签键	标签值
业务	游戏1、游戏2、游戏3
运维负责人	张三、文四、王五

将这些标签键/值绑定到云数据库 PostgreSQL 上，资源与标签键值的关系如下表所示：

instance-id	业务	运维负责人
postgres-abcdef1	游戏1	王五
postgres-abcdef2	游戏2	文四
postgres-abcdef3	游戏3	张三

使用标签

创建与删除标签方法请参见 标签快速入门。
编辑云数据库 PostgreSQL 标签方法请参见 编辑标签。

版权所有：腾讯云计算（北京）有限责任公司

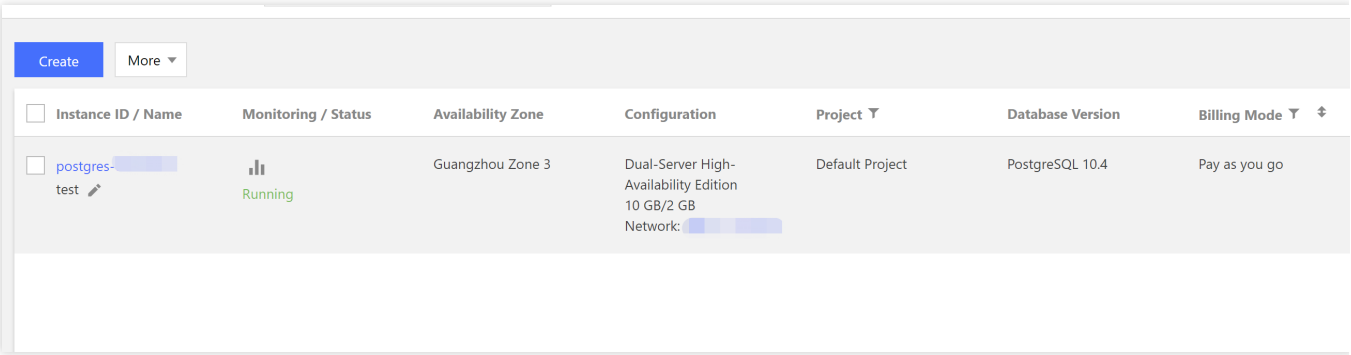
编辑标签

最近更新时间：2024-01-22 16:24:46

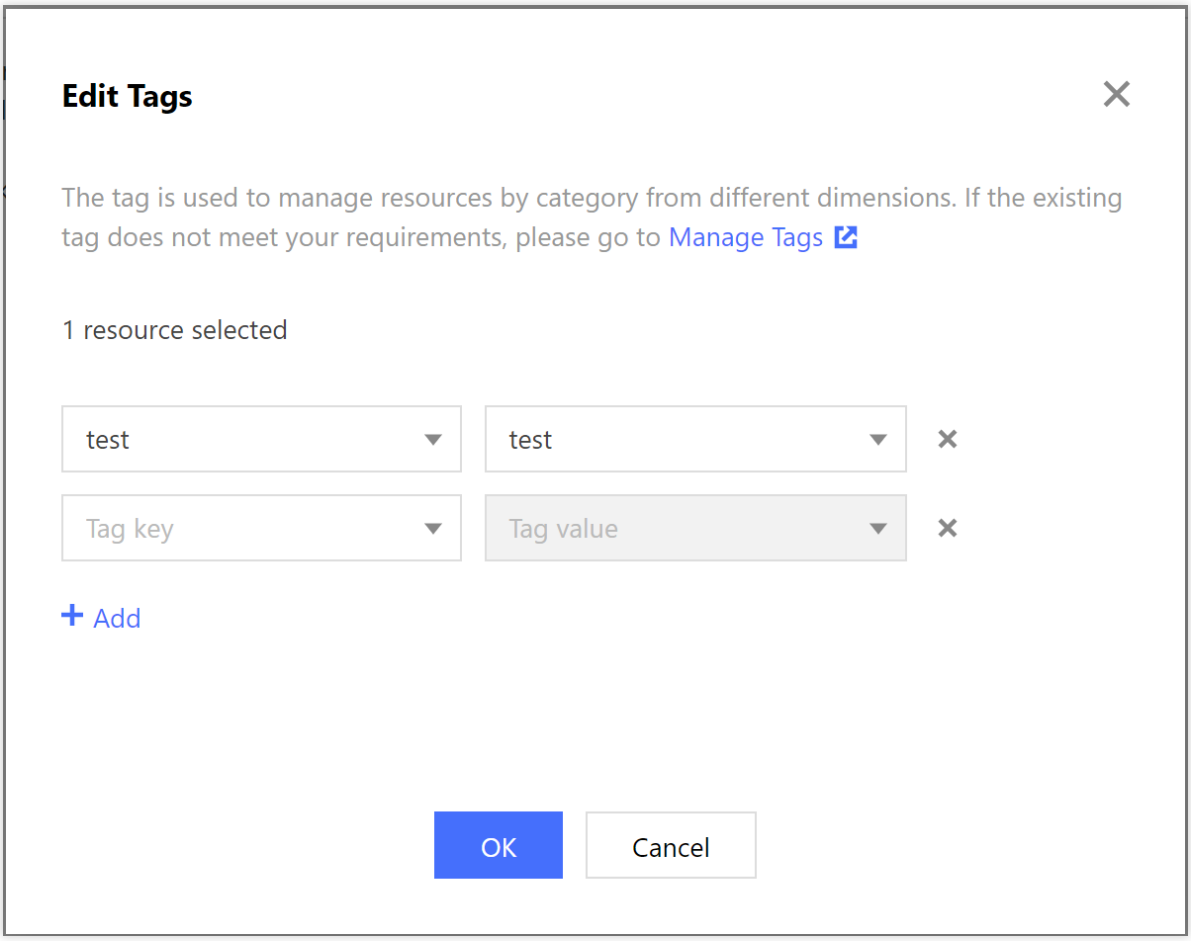
您可以按照下列操作对资源进行标签的编辑操作。

对单个实例编辑标签

1. 登录 PostgreSQL 控制台，在实例列表选择所需实例，在“操作”列选择**更多>编辑标签**。

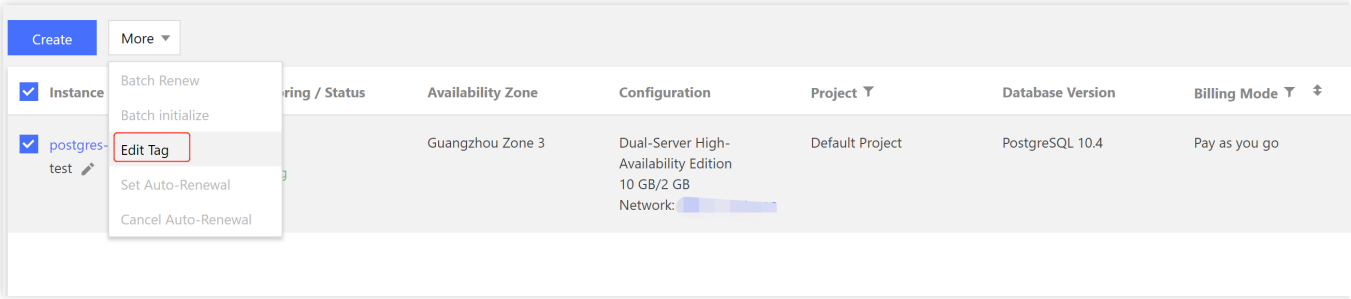


2. 在弹出的对话框，可添加、修改、删除标签，确认无误后，单击**确定**即可。



对多个实例编辑标签

1. 登录 PostgreSQL 控制台，在实例列表选择所需实例，在列表上方选择**更多操作>编辑标签**。



2. 在弹出的对话框，可添加、修改、删除标签，确认无误后，单击**确定**即可。