

云数据库 PostgreSQL

操作指南

产品文档



腾讯云

【版权声明】

©2013-2018 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模式的承诺或保证。

文档目录

操作指南

- 使用限制

 - 实例类型

 - 操作约束

- 监控功能

- 告警功能

- 慢查询分析

- 错误日志

- 在云服务器上恢复PostgreSQL的数据

- 迁移数据

- 重启实例

- 支持插件

- fdw插件

操作指南

使用限制

实例类型

最近更新时间：2017-07-17 11:57:23

目前云数据库存在以下实例类型：

- **主实例**：直接申请购买的实例，拥有云数据库的全部功能。
- **主实例（独享）[暂未支持]**：独享集群数据库模式下，在独享集群内申请的数据库实例，拥有云数据库的全部功能。
- **灾备实例[暂未支持]**：部署在异地（不同可用区）与主实例实时（异步）同步数据的实例，通常用于数据库容灾，仅拥有部分功能。
- **临时实例[暂未支持]**：由回档功能生成的临时存在的实例，用于回档后数据校验，可通过切换为主实例功能，将校验后的临时实例恢复为主实例。
- **只读实例[暂未支持]**仅提供读功能的实例被称为只读实例。只读实例无法单独存在，必须隶属于某个主实例，唯一数据来源是从主实例同步数据，在实例列表中不可见。

操作约束

最近更新时间：2018-08-29 17:52:23

为了提供安全和稳定的实例服务，腾讯云 TencentDB for PostgreSQL 约定了如下使用约束，请事先了解。

操作	权限说明
数据库超级管理员权限	不提供超级用户权限
重启实例	暂未提供
修改数据库参数设置	暂未提供
数据恢复	支持使用psql还原由pg_dump创建的转储数据
搭建数据库复制	系统自动搭建了基于 PostgreSQL 流复制的 HA 模式，无需手动搭建 PostgreSQL Standby 节点对用户不可见，不能直接用于访问

监控功能

最近更新时间：2018-09-14 12:13:46

性能监控

为方便用户查看和掌握实例的运行信息，PostgreSQL提供了丰富的性能监控项，用户可进入PostgreSQL管理控制台对应【实例管理】-【系统监控】查看。

pg_stat_statements

您还可以通过 [pg_stat_statements](#) 视图查询pg详细性能指标。

已提供监控指标

指标中文名	指标英文名	指标含义	指标单位
CPU利用率	cpu	实例CPU使用率，由于在闲时采用灵活的CPU限制策略，CPU利用率可能大于100%	%
已用存储空间	storage	实例占用磁盘的可用空间	GB
磁盘IOPS	iops	实例的IOPS（每秒的请求次数）	次/秒
输入流量	in_flow	实例读写输入的流量	KB/秒
输出流量	out_flow	实例读写输出的流量	KB/秒
连接数	connections	实例的活跃连接历史变化趋势	个
请求数	read_write_calls	读写（增删改查）请求每分钟总数	次/分钟
读请求数	read_calls	读请求每分钟总数	次/分钟
写请求数	write_calls	写请求每分钟总数	次/分钟
其他请求数	other_calls	除了读和写以外的请求总数（如Drop），按分钟累加	次/分钟
缓冲区缓存命中率	hit_percent	数据缓存命中率	%

指标中文名	指标英文名	指标含义	指标单位
平均执行时延	sql_runtime_avg	所有SQL请求的平均执行时间，不包含事务里面的SQL。	ms
最长TOP10执行时延	sql_runtime_avg	执行时间最长的top10的SQL的平均值	ms
最短TOP10执行时延	sql_runtime_min	执行时间最短的top10的SQL的平均值	ms
剩余XID数量	remain_xid	剩余的Transaction Id数量，Transaction Id最大有 2^{32} 个，小于1000000建议手工执行vacuum full。	个
主备XLOG同步差异	xlog_diff	每分钟采样，主备XLOG的同步的大小差异，代表着同步的延迟，越小越好。	byte

告警功能

最近更新时间：2018-09-14 12:10:49

云数据库已支持关键性能指标的短信、邮件、站内信等告警，欢迎登录腾讯云控制台 > [云监控](#) > [告警策略](#) 进行设置。

慢查询分析

最近更新时间：2018-09-14 12:12:32

1. 功能说明

将超过指定时间的SQL语句查询称为“慢查询”，对应语句称为“慢查询语句（简称慢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。

腾讯云云数据库（TencentDB for PostgreSQL）在【实例管理】-【性能优化】模块下，提供慢查询分析能力。如下图：

The screenshot displays the '慢查询分析' (Slow Query Analysis) interface. On the left, there's a table listing slow queries with columns for execution time, SQL statement, database, account, first execution time, total count, total time, and total time percentage. One query is highlighted: 'select pg_sleep(?);' with a total count of 3 and a total time of 30.03 seconds.

The right panel, titled '慢查询监控' (Slow Query Monitoring), provides a detailed view of the selected query. It includes a table with the following data:

数据库	vitostest	帐号	root
总次数	3	总次数占比	100.00%
首次执行时间	2017-08-09 16:26:21	最后执行时间	2017-08-09 16:26:49
查询信息统计		平均时间 (秒)	10.01
		最小时间 (秒)	10.01
		最大时间 (秒)	10.01
读共享内存块数	0	写共享内存块数	0
读IO总耗时 (秒)	10.01	写IO总耗时 (秒)	10.01

Below the table, the abstract SQL statement is shown as 'select pg_sleep(?);'. A line graph plots the query frequency (调用次数) over time, showing a single data point at 16:26:21.000 with a value of 3.5.

2. 主要参数说明

2.1. 慢查询分析默认设置

慢查询功能：默认开启

慢SQL记录时间（log_min_duration_statement）：默认配置为1s，即慢查询语句查询时间超过1s的才被记录。如有需要调整，请提交工单，说明实例ID和需要配置该参数的时间可修改。

分析数据输出延迟：1~5分钟。

日志记录时长：7天（最大最近10000条）。

2.2. 分析列表字段说明

目前慢查询分析功能对慢SQL有如下分析：

基本信息

- a) **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。
- b) **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。
- c) **数据库**：该语句调用了哪个数据库
- d) **账号**：该语句是使用哪个帐号来运行的。
- e) **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）。
- f) **总次数**：在统计范围内，该慢SQL总共出现的次数。
- g) **总次数占比**：在统计范围内，该慢SQL总共出现的次数，占有所有慢SQL出现的次数的比值。
- d) **发送行数**：在统计范围内，数据库服务器发送（返回）给客户端多少行结果。

查询信息统计

- a) **总时间**：在统计范围内，慢查询语句查询总耗时。
- b) **总时间占比**：在统计范围内，慢查询语句占有所有慢查询语句时间的占比。
- c) **平均时间**：慢查询语句总时间除以总次数的平均时间；
- d) **最小时间**：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。
- e) **** 最大时间**：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。

读写数据统计

- a) **读共享内存块数**：在统计范围内，该慢SQL总共从共享内存中读取多数据。
- b) **写共享内存块数**：在统计范围内，该慢SQL总共写入共享内存中多数据。

读写IO耗时统计

- a) **读IO总耗时**：在统计范围内，该慢SQL总共消耗多少读IO时间；用于帮助判断该语句是否在做全盘扫描等耗时操作数据。
- b) **写IO总耗时**：在统计范围内，该慢SQL总共消耗多少写IO时间；用于帮助判断该语句是否一次性写入大量（临时）数据。

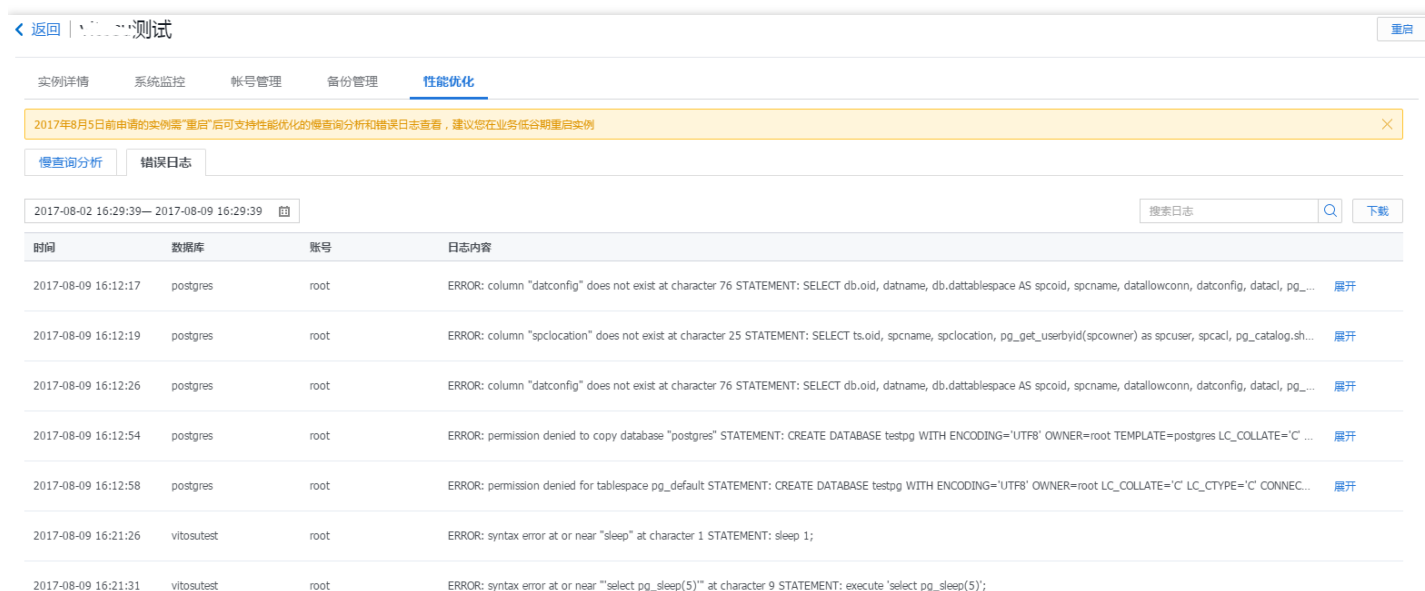
错误日志

最近更新时间：2018-09-14 12:15:17

1. 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。

腾讯云数据库（TencentDB for PostgreSQL）在【实例管理】-【性能优化】模块下，提供错误日志查看能力。如下图：



The screenshot shows the 'Performance Optimization' (性能优化) tab in the Tencent Cloud console. It displays a table of error logs for a PostgreSQL instance. The table has columns for 'Time' (时间), 'Database' (数据库), 'Account' (账号), and 'Log Content' (日志内容). The logs show various errors such as 'column does not exist' and 'permission denied'.

时间	数据库	账号	日志内容
2017-08-09 16:12:17	postgres	root	ERROR: column "datconfig" does not exist at character 76 STATEMENT: SELECT db.oid, datname, db.dattablespace AS spcoid, spcname, dataallowconn, datconfig, datacl, pg_...
2017-08-09 16:12:19	postgres	root	ERROR: column "spclocation" does not exist at character 25 STATEMENT: SELECT ts.oid, spcname, spclocation, pg_get_userbyid(spcowner) as spcuser, spcacl, pg_catalog.sh...
2017-08-09 16:12:26	postgres	root	ERROR: column "datconfig" does not exist at character 76 STATEMENT: SELECT db.oid, datname, db.dattablespace AS spcoid, spcname, dataallowconn, datconfig, datacl, pg_...
2017-08-09 16:12:54	postgres	root	ERROR: permission denied to copy database "postgres" STATEMENT: CREATE DATABASE testpg WITH ENCODING='UTF8' OWNER=root TEMPLATE=postgres LC_COLLATE='C' ...
2017-08-09 16:12:58	postgres	root	ERROR: permission denied for tablespace pg_default STATEMENT: CREATE DATABASE testpg WITH ENCODING='UTF8' OWNER=root LC_COLLATE='C' LC_CTYPE='C' CONNEC...
2017-08-09 16:21:26	vitostest	root	ERROR: syntax error at or near "sleep" at character 1 STATEMENT: sleep 1;
2017-08-09 16:21:31	vitostest	root	ERROR: syntax error at or near "select pg_sleep(5)" at character 9 STATEMENT: execute 'select pg_sleep(5);

2. 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：1~5分钟。

日志记录时长：7天（最大最近10000条）。

在云服务器上恢复PostgreSQL的数据

最近更新时间：2018-09-14 12:11:58

1. 创建恢复目录

```
mkdir recovery
```

注: recovery只是示例目录，用户可以自行修改为自己的目录。

2. 下载全量备份文件

进入PostgreSQL的管理页面——“腾讯云控制台>云产品>关系型数据库>PostgreSQL实例列表>管理>备份管理”单击下载，获取您所需的备份文件下周地址。（服务器需与数据库处于同一网络中）

```
[tommy@100 ~/recovery]$ ls -lh
total 4.5M
-rw-r--r-- 1 tommy users 4.5M Sep  5 11:22 20170905010143.tar.gz
```

3. 解压全量备份文件

输入命令

```
tar xzf 20170905010143.tar.gz -C recovery
```

解压后得到

```
[tommy@100 ~/recovery]$ ls -lh
total 4.7M
-rw-r--r-- 1 tommy users 4.5M Sep  5 11:22 20170905010143.tar.gz
-rw----- 1 tommy users  4 Jul  4 15:25 PG_VERSION
-rw----- 1 tommy users 215 Sep  5 01:01 backup_label
drwx----- 7 tommy users 4.0K Aug 11 11:42 base
drwx----- 2 tommy users 4.0K Aug 18 00:21 global
drwx----- 2 tommy users 4.0K Jul  4 15:25 pg_clog
drwx----- 2 tommy users 4.0K Jul  4 15:25 pg_commit_ts
drwx----- 2 tommy users 4.0K Jul  4 15:25 pg_dynshmem
-rw----- 1 tommy users 4.7K Jul  4 15:25 pg_hba.conf
-rw----- 1 tommy users 1.6K Jul  4 15:25 pg_ident.conf
drwx----- 4 tommy users 4.0K Jul  4 15:25 pg_logical
drwx----- 4 tommy users 4.0K Jul  4 15:25 pg_multixact
drwx----- 2 tommy users 4.0K Aug 18 00:21 pg_notify
drwx----- 2 tommy users 4.0K Jul  4 15:25 pg_replslot
drwx----- 2 tommy users 4.0K Jul  4 15:25 pg_serial
drwx----- 2 tommy users 4.0K Jul  4 15:25 pg_snapshots
drwx----- 2 tommy users 4.0K Aug 18 00:21 pg_stat
drwx----- 2 tommy users 4.0K Sep  5 01:00 pg_stat_tmp
drwx----- 2 tommy users 4.0K Jul  4 15:25 pg_subtrans
drwx----- 2 tommy users 4.0K Jul  4 15:25 pg_tblspc
drwx----- 2 tommy users 4.0K Jul  4 15:25 pg_twophase
drwxr-xr-x 2 tommy users 4.0K Sep  5 01:01 pg_xlog
```

4. 安装相同版本的postgresql (如已安装可跳过此步骤)

例如采用yum源的安装方式，yum源可到 <https://yum.postgresql.org/> 查找您所需要的版本：

```
yum install http://yum.postgresql.org/9.3/RedHat/rhel-6-x86_64/pgdg-redhat93-9.3-1.noarch.rpm
yum install postgresql93-server postgresql93-contrib
```

//初始化

```
service postgresql-9.3 initdb
```

开启

```
service postgresql-9.3 start或者
```

```
/etc/init.d/postgresql-9.3 start
```

查看安装结果

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

可以看到

```
[root@i-87-575-VM vmuser]# rpm -aq| grep postgres
postgresql93-libs-9.3.4-1PGDG.rhel6.x86_64
postgresql93-contrib-9.3.4-1PGDG.rhel6.x86_64
postgresql93-9.3.4-1PGDG.rhel6.x86_64
postgresql93-server-9.3.4-1PGDG.rhel6.x86_64
```

5. 配置文件修改

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
rds_extension_list.names
pg_stat_statements.max
pg_stat_statements.track
archive_mode
archive_command
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 = "
```

6. 更改文件夹权限

```
chmod 0700 recovery
```

修改后如下图

```
[tommy@100 ~/recovery]$ ls -al .
total 4744
drwx----- 19 tommy users 4096 Sep 5 11:23 .
drwx----- 55 tommy users 4096 Sep 5 11:22 ..
-rw-r--r-- 1 tommy users 4682340 Sep 5 11:22 20170905010143.tar.gz
-rw----- 1 tommy users 4 Jul 4 15:25 PG_VERSION
-rw----- 1 tommy users 215 Sep 5 01:01 backup_label
drwx----- 7 tommy users 4096 Aug 11 11:42 base
drwx----- 2 tommy users 4096 Aug 18 00:21 global
```

7. 应用增量备份文件[可选]

该步骤为可选步骤，如果跳过该步骤，则数据库的内容为开始做全量备份时数据库的内容。

说明：将xlog文件放入pg_xlog文件夹下，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_xlog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。

下载增量备份文件（xlog）

```
[tommy@100 ~/recovery/pg_xlog]$ ls -lh
total 33M
-rw----- 1 tommy users 16M Sep 5 13:14 0000000100000003000000F0
-rw----- 1 tommy users 16M Sep 5 13:21 0000000100000003000000F1
-rw-r--r-- 1 tommy users 783K Sep 5 13:23 20170904010214_20170905010205.tar.gz
drwx----- 2 tommy users 4.0K Sep 5 13:14 archive_status
[tommy@100 ~/recovery/pg_xlog]$
```

解压在 pg_xlog文件夹下

```
tar xzf 20170904010214_20170905010205.tar.gz
```

```
[tommy@100 ~/recovery/pg_xlog]$ pwd
/data2/tommy/recovery/pg_xlog
[tommy@100 ~/recovery/pg_xlog]$ ls -lh
total 465M
-rw----- 1 tommy users 16M Sep 4 01:02 0000000100000003000000E3
-rw----- 1 tommy users 16M Sep 4 01:02 0000000100000003000000E4
-rw----- 1 tommy users 16M Sep 4 01:02 0000000100000003000000E5
-rw----- 1 tommy users 16M Sep 4 01:02 0000000100000003000000E6
-rw----- 1 tommy users 16M Sep 4 01:02 0000000100000003000000E7
-rw----- 1 tommy users 16M Sep 4 01:02 0000000100000003000000E8
-rw----- 1 tommy users 16M Sep 4 01:02 0000000100000003000000E9
-rw----- 1 tommy users 16M Sep 4 01:02 0000000100000003000000EA
-rw----- 1 tommy users 16M Sep 4 01:02 0000000100000003000000EB
-rw----- 1 tommy users 16M Sep 4 01:02 0000000100000003000000EC
-rw----- 1 tommy users 16M Sep 4 01:02 0000000100000003000000ED
-rw----- 1 tommy users 16M Sep 4 01:02 0000000100000003000000EE
-rw----- 1 tommy users 16M Sep 5 01:01 0000000100000003000000EF
-rw----- 1 tommy users 16M Sep 5 01:01 0000000100000003000000F0
-rw----- 1 tommy users 312 Sep 5 01:01 0000000100000003000000F0.00000028.backup
-rw----- 1 tommy users 16M Sep 5 01:01 0000000100000003000000F1
-rw----- 1 tommy users 16M Sep 5 01:01 0000000100000003000000F2
-rw----- 1 tommy users 16M Sep 5 01:01 0000000100000003000000F3
-rw----- 1 tommy users 16M Sep 5 01:01 0000000100000003000000F4
-rw----- 1 tommy users 16M Sep 5 01:01 0000000100000003000000F5
-rw----- 1 tommy users 16M Sep 5 01:01 0000000100000003000000F6
-rw----- 1 tommy users 16M Sep 5 01:01 0000000100000003000000F7
-rw----- 1 tommy users 16M Sep 5 01:01 0000000100000003000000F8
-rw----- 1 tommy users 16M Sep 5 01:01 0000000100000003000000F9
-rw----- 1 tommy users 16M Sep 5 01:02 0000000100000003000000FA
-rw----- 1 tommy users 16M Sep 5 01:02 0000000100000003000000FB
-rw----- 1 tommy users 16M Sep 5 01:02 0000000100000003000000FC
-rw----- 1 tommy users 16M Sep 5 01:02 0000000100000003000000FD
-rw----- 1 tommy users 16M Sep 5 01:02 0000000100000003000000FE
-rw----- 1 tommy users 16M Sep 5 01:02 0000000100000003000000FF
```

8. 启动数据库

```
pg_ctl start -D ~/recovery
```

```
[tommy@100 ~/recovery]$ pg_ctl start -D ~/recovery
server starting
[tommy@100 ~/recovery]$ LOG:  redirecting log output to logging collector process
HINT:  Future log output will appear in directory "pg_log".
```

9. 登录验证

登录

```
[tommy@100 ~/recovery]$ psql -d postgres -U postgres -p 50034
psql (9.5.4)
Type "help" for help.

postgres=#
```

数据库是否运行

```
[tommy@100 ~/recovery]$ pg_ctl status -D ~/recovery
pg_ctl: server is running (PID: 26564)
/data2/tommy/pgbin-9.5.4/bin/postgres "-D" "/data2/tommy/recovery"
```


迁移数据

最近更新时间：2018-09-14 12:14:12

使用腾讯云DTS服务

PostgreSQL 数据库可使用腾讯云【数据传输服务DTS】的【数据迁移】功能做数据导入

详细操作参考 [数据迁移服务](#)

使用psql命令

如果你的数据已经在PostgreSQL上，你可以psql命令将数据轻松的迁移到CDB for PostgreSQL中。

迁移数据的工作主要分为两大部分：

- (1) 使用pg_dump命令进行逻辑备份，创建转储数据；
- (2) 将上一步的备份数据恢复到CDB for PostgreSQL上。

准备操作：你应该已经完成PostgreSQL实例的准备，如未完成，请参见[申请实例和连接实例的操作指引](#)。

第一步

使用PostgreSQL客户端连接本地数据库，执行如下命令，备份数据。

```
pg_dump -U username -h hostname -p port -d databasename -f filename
```

参数说明如下

参数	意义
username	本地数据库用户名
hostname	本地数据库主机名，如果是在本地数据库主机登录，可以使用 localhost
port	本地数据库端口号
databasename	要备份的本地数据库名

参数	意义
filename	要生成的备份文件名称，如mydump.sql

第二步

准备操作：请先将备份好的数据上传到同一内网的云服务器主机上，然后通过内网进行数据恢复，以保证网络的稳定和数据的安全。

在您登录云服务器后，在PostgreSQL客户端中，执行如下命令，恢复数据。

```
psql -U username -h hostname -d desintationdb -p port -f dumpfilename
```

参数说明如下

参数	意义
username	CDB for PostgreSQL 的数据库用户名
hostname	CDB for PostgreSQL 的数据库地址
desintationdb	CDB for PostgreSQL 的数据库名
port	CDB for PostgreSQL 的数据库端口号
dumpfilename	备份数据文件名，如mydump.sql

重启实例

最近更新时间：2018-09-14 12:14:42

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。

使用控制台重启实例

- 1) 打开[云数据库PostgreSQL控制台](#)。
- 2) 对于单个运行中的 PostgreSQL 实例，在实例列表操作栏中，点击【重启】，即可对 PostgreSQL 实例进行重启。

数据库重启注意事项

- 1) 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。
- 2) 一般说运行重启操作后需要十几秒~几分钟时间，重启期间实例将无法提供正常服务。
- 3) **重启数据库有几率失败**，这是**正常**现象，如果失败可再次（手工）点击重启功能继续重启。
- 4) 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。
- 5) 重启后，业务需要重连数据库，请确保您的业务有重连机制。

支持插件

最近更新时间：2018-09-30 15:13:19

目前，云数据库PostgreSQL已经支持自主创建插件插件（`create extension pluginName`）；支持的插件列表如下（PostgreSQL 10 版本由于功能更强大，部分插件不再支持）：

插件名	9.3.5	9.5.4	10.4
btree_gin	√	√	√
btree_gist	√	√	√
chkpass	√	√	√
citext	√	√	√
cube	√	√	√
dict_int	√	√	√
earthdistance	√	√	√
postgres_fdw	√	√	暂不支持
mysql_fdw	√	√	暂不支持
cos_fdw	√	√	暂不支持
fuzzystrmatch	√	√	√
hstore	√	√	√
intagg	√	√	√
intarray	√	√	√
isn	√	√	√
jsonbx	x	√	x
ltree	√	√	√
pg_hint_plan	√	√	x
pg_prewarm	x	√	√
pg_stat_statements	√	√	√

插件名	9.3.5	9.5.4	10.4
pg_trgm	√	√	√
pgcrypto	√	√	√
pgrouting	√	√	√
pgrowlocks	√	√	√
pl/perl	√	√	√
pl/pgsql	√	√	√
pl/tcl	√	√	√
plcoffee	√	√	√
plls	√	√	√
plv8	√	√	√
postgis	√	√	√
postgis_tiger_geocoder	√	√	√
postgres_fdw	√	√	√
sslinfo	√	√	√
tablefunc	√	√	√
tsearch2	√	√	x
unaccent	√	√	√
uuid-oss	√	√	√
zhparser	√	√	x

fdw插件

最近更新时间：2018-09-30 15:13:44

FDW(FOREIGN DATA WRAPPER, 外部数据包装器)是 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。FDW插件有很多分类，根据不同目标数据库实例类型可以定义不同的FDW插件，例如：postgres_fdw、mysql_fdw、mongo_fdw等。使用过程包含以下步骤：

1. 使用“CREATE EXTENSION”语句安装 FDW 插件。
2. 使用“CREATE SERVER”语句,为每个需要连接的远程数据库创建一个外部服务器对象。指定除了 user 和 password 以外的连接信息作为服务器对象的选项。
3. 使用“CREATE USER MAPPING”语句，为每个需要通过外部服务器访问的数据库创建用户映射。指定远程的账号和密码作为映射用户的 user 和 password。
4. 使用“CREATE FOREIGN TABLE”语句，为每个需要访问的远程表创建外部表。创建的外部表的对应列必须与远程表匹配。也可以在外表中与远程表不同的表名和列名，但前提是你必须将正确的远程对象名作为创建外部表对象的选项。

由于 FDW 插件可以直接跨实例访问，基于安全性考虑，TencentDB For 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。

CREATE SERVER 辅助参数

postgres_fdw、mysql_fdw 等插件辅助参数

- host
必须项。目标实例的内网 ip 地址，postgres_fdw 使用。
- address
必须项。目标实例的内网 ip 地址，mysql_fdw 使用。
- port
必须项。目标实例的内网 port。
- instanceid
必须项。目标实例的资源 ID。
 - i. 如果目标实例类型为 CDB 类型，则为实例 ID，例如格式类似 postgres-xxxxx、mysql-xxxxx 等，可在实例控制台查看，如 PostgreSQL 为：



ii. 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。



- access_type

非必须项。目标实例所属类型：

- i. 目标实例为 CDB 实例，包括 TencentDB For PostgreSQL、TencentDB For MySQL 等，如果不显示指定，则默认该项；
- ii. 目标实例在腾讯云 CVM 机器上；
- iii. 目标实例为腾讯云外网自建；
- iv. 目标实例为云 vpn 接入的实例；
- v. 目标实例为自建 vpn 接入的实例；
- vi. 目标实例为专线接入的实例；
- vii. 目标实例为腾讯云 COS 数据；

- uin
非必须项。实例所属的账号 ID，通过该信息鉴定用户权限，可在这里查询：[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](#) - 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 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。

COS_FDW 插件的辅助参数

- host
必须项。腾讯云对象存储 COS 的访问域名，例如：`https://xxxx-xxxxxx.cos.ap-beijing.myqcloud.com`。
- bucket
必须项。腾讯云对象存储 COS 的存储桶名。
- id
必须项。腾讯云 API 密钥的 SecretId 值，可从 [访问管理控制台](#) - 云 API 密钥中查询。
- key
必须项。腾讯云 API 密钥的 SecretKey 值，可从 [访问管理控制台](#) - 云 API 密钥中查询。

FDW 支持

名称	是否可直接使用	跨地域使用
postgres_fdw	2018 年 4 月 26 日之前创建的实例重启后可使用，新实例可以直接使用	默认不支持，可 提交工单 申请
mysql_fdw	2018 年 4 月 26 日之前创建的实例重启后可使用，新实例可以直接使用	默认不支持，可 提交工单 申请
cos_fdw	灰度中，可 提交工单 申请试用	默认不支持，可 提交工单 申请

参考链接

[9.3 版本 SERVER 创建](#)

[9.5 版本 SERVER 创建](#)

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。

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
```

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
```

创建 postgres_fdw 插件

```
#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 servers

(2 rows)

创建 SERVER

1. 目标实例为 TencentDB 实例类型。

#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 'xxx.xxx.xxx.xxx',d
bname 'testdb2', port '5432', instanceid 'postgres-xxxxx');
CREATE SERVER
```

2. 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 'xxx.xxx.xxx.xxx', d
bname 'testdb2', port '5432', instanceid 'ins-xxxxx', access_type '2', region 'ap-guangzhou' , uin 'xx
xxxx' , own_uin 'xxxxxx');
CREATE SERVER
```

3. 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 'xxx.xxx.xxx.xxx',d
bname 'testdb2', port '5432', instanceid 'ins-xxxxx', access_type '2', region 'ap-guangzhou', uin 'xxx
xxx', own_uin 'xxxxxx', vpcid 'vpc-xxxxxx', subnetid 'subnet-xxxxx');
CREATE SERVER
```

4. 目标实例在腾讯云外网自建。

```
testdb1=>create server srv_test1 foreign data wrapper postgres_fdw options (host 'xxx.xxx.xxx.xxx',d
bname 'testdb2', port '5432', access_type '3', region 'ap-guangzhou', uin 'xxxxxx', own_uin 'xxxxxx'
);
CREATE SERVER
```

5. 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 'xxx.xxx.xxx.xxx',d
bname 'testdb2', port '5432', access_type '4', region 'ap-guangzhou', uin 'xxxxxx', own_uin 'xxxxxx',
vpngwid 'xxxxxx');
```

6. 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 'xxx.xxx.xxx.xxx',d
bname 'testdb2', port '5432', access_type '5', region 'ap-guangzhou', uin 'xxxxxx', own_uin 'xxxxxx',
vpngwid 'xxxxxx');
```

7. 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 'xxx.xxx.xxx.xxx',d
bname 'testdb2', port '5432', access_type '6', region 'ap-guangzhou', uin 'xxxxxx', own_uin 'xxxxxx',
dcgid 'xxxxxx');
CREATE SERVER
```

创建用户映射

```
testdb1=> create user mapping for user1 server srv_test1 options (user 'user2',password 'password2');
CREATE USER MAPPING
```

创建外部表

```
testdb1=> create foreign table foreign_table1(id integer) server srv_test1 options(table_name 'test_ta
ble2');
CREATE FOREIGN TABLE
```

访问外部数据

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

postgres_fdw 使用注意

如果目标实例在 CVM 上，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 CDB 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。

参考文档

[postgres_fdw介绍](#)

[9.3 版本 SERVER 创建](#)

[9.5 版本 SERVER 创建](#)

[9.3 版本 pg_hba 介绍](#)

[9.5 版本 pg_hba 介绍](#)

使用 mysql_fdw 示例

使用 mysql_fdw 插件可以访问其他 mysql 实例的数据。

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
```

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
```

创建 mysql_fdw 插件

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

List of installed extensions

Name | Version | Schema | Description

```
-----+-----+-----+-----
plpgsql | 1.0 | pg_catalog | PL/pgSQL procedural language
mysql_fdw | 1.0 | public | Foreign data wrapper for querying a MySQL server
(2 rows)
```

创建 SERVER

1. 目标实例为 TencentDB 实例类型。

#从本实例的testdb1访问目标实例testdb2的数据

```
testdb1=>create server srv_test1 foreign data wrapper mysql_fdw options (host 'xxx.xxx.xxx.xxx', port '3306', instanceid 'cdb-xxxxx', uin 'xxxxxx', region 'ap-guangzhou');
CREATE SERVER
```

2. 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 mysql_fdw options (host 'xxx.xxx.xxx.xxx', port '3306', instanceid 'ins-xxxxx', access_type '2', region 'ap-guangzhou', uin 'xxxxxx', own_uin 'xxxxxx');
CREATE SERVER
```

3. 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 mysql_fdw options (host 'xxx.xxx.xxx.xxx', port '3306', instanceid 'ins-xxxxx', access_type '2', region 'ap-guangzhou', uin 'xxxxxx', own_uin 'xxxxxx', vpcid 'vpc-xxxxxx', subnetid 'subnet-xxxxx');
CREATE SERVER
```

4. 目标实例在腾讯云外网自建

```
testdb1=>create server srv_test1 foreign data wrapper mysql_fdw options (host 'xxx.xxx.xxx.xxx', port '3306', access_type '3', region 'ap-guangzhou', uin 'xxxxxx', own_uin 'xxxxxx');
CREATE SERVER
```

5. 目标实例在腾讯云VPN接入的实例

```
testdb1=>create server srv_test1 foreign data wrapper mysql_fdw options (host 'xxx.xxx.xxx.xxx', port '3306', access_type '4', region 'ap-guangzhou', uin 'xxxxxx', own_uin 'xxxxxx', vpngwid 'xxxxxx');
```

6. 目标实例在自建VPN接入的实例

```
testdb1=>create server srv_test1 foreign data wrapper mysql_fdw options (host 'xxx.xxx.xxx.xxx', port '3306', access_type '5', region 'ap-guangzhou', uin 'xxxxxx', own_uin 'xxxxxx', vpngwid 'xxxxxx');
```

7. 目标实例在腾讯云专线接入的实例

```
testdb1=>create server srv_test1 foreign data wrapper mysql_fdw options (host 'xxx.xxx.xxx.xxx', port '3306', access_type '6', region 'ap-guangzhou', uin 'xxxxxx', own_uin 'xxxxxx', dcgid 'xxxxxx');  
CREATE SERVER
```

创建用户映射

```
testdb1=> create user mapping for user1 server srv_test1 options (user 'user2',password 'password2');  
CREATE USER MAPPING
```

创建外部表

```
testdb1=> create foreign table foreign_table1(id integer) server srv_test1 options(dbname 'testdb2', table_name 'test_table2');  
CREATE FOREIGN TABLE
```

访问外部数据

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

参考链接

[9.3 版本 SERVER 创建](#)

[9.5 版本 SERVER 创建](#)

使用cos_fdw示例

使用 cos_fdw 插件可以在 TencentDB For PostgreSQL 实例中获取腾讯云对象存储COS中的文本数据。

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
```

2. 在 [对象存储控制台](#) 创建存储桶“test1”，在该存储桶中上传文本文件至“/testdir/test.txt”。

创建 cos_fdw 插件

```
#创建
postgres=> \c testdb1
You are now connected to database "testdb1" as user "user1".
testdb1=> create extension cos_fdw;
CREATE EXTENSION
#查看
testdb1=> \dx
List of installed extensions
Name | Version | Schema | Description
-----+-----+-----+-----
plpgsql | 1.0 | pg_catalog | PL/pgSQL procedural language
cos_fdw | 1.0 | public | foreign-data wrapper for flat qcloud cos access
(2 rows)
```

创建 SERVER

```
#从本实例的testdb1访问对象存储test1上的数据
testdb1=>create server srv_cos foreign data wrapper cos_fdw options(host 'test11-xxxxxx.cos.ap-chengdu.myqcloud.com', bucket 'test1', id 'xxxxxx', key 'xxxxxx');
CREATE SERVER
```

创建外部表

参数：filepath，本文文件在存储桶中的相对路径

```
testdb1=> create foreign table test_cos(id integer) server srv_cos options(filepath '/testdir/test.txt');
CREATE FOREIGN TABLE
```

访问外部数据

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

参考链接

[对象存储文档。](#)