### postgresql操作手册

**前言：**

1.蓝色字体表示命令行命令，正式执行时不要复制前面的#号，#号只是提示应该使用root权限操作

2.绿色字体表示注释，有时注释太多就不用绿色表示了

3.注意：本文档的所有操作请先在测试机里进行实践，请不要直接在真实的服务器中操作！

**版权声明**：

本文档以开源的形式发布，所有条款如下：

 （1）无担保：作者不保证文档内容的准确无误，亦不承担由于使用此文档所导致的任何后果

 （2）自由使用：任何人可以出于任何目的而自由地 阅读/链接/打印/转载/引用/分发/再创作 此文档，无需任何附加条件

若您 阅读/链接/打印/转载/引用/分发/再创作 本文档，则说明接受以上2个条款。

作者：李茂福

更新日期：2023-12-27

**★第1章、安装postgresql**

# yum install postgresql-server -y # EL8安装光盘AppSteam里自带有10.x版本的软件包

# postgresql-setup --initdb --unit postgresql

# systemctl enable postgresql

# systemctl start postgresql

如果想安装其他版本，可到官网查询安装方法

https://www.postgresql.org/download/linux/redhat/

 

本文档选择PG-13版本作为讲解，使用el8系列的系统，x86\_64平台

# yum install -y https://download.postgresql.org/pub/repos/yum/reporpms/EL-8-x86\_64/pgdg-redhat-repo-latest.noarch.rpm

#或者直接创建repo文件，内容如下

[pgdg13]

name=PostgreSQL 13 for RHEL / Rocky $releasever - $basearch

baseurl=https://download.postgresql.org/pub/repos/yum/13/redhat/rhel-$releasever-$basearch

enabled=1

gpgcheck=0

# yum module disable postgresql -qy

# yum install postgresql13-server -y

# /usr/pgsql-13/bin/postgresql-13-setup initdb

# systemctl enable postgresql-13

# systemctl start postgresql-13

**★第2章、设置数据存储路径**

PostgreSQL 初始化数据库之后，

10版本的默认的数据目录是/var/lib/pgsql/

13版本的默认的数据目录是/var/lib/pgsql/13/

现在想修改为/data\_pg/ 目录

★PG-13版本

# systemctl stop postgresql-13 #先停止服务

# mkdir /data\_pg

# chown postgres:postgres /data\_pg

# vi /var/lib/pgsql/13/data/postgresql.conf #pg-13版本，添加或修改以下配置

data\_directory = '/data\_pg/data/'

# mv /var/lib/pgsql/13/\* /data\_pg/ #将原数据目录下所有内容移动到新数据目录下

# vi /usr/lib/systemd/system/postgresql-13.service #pg-13版本，修改systemd服务文件，指定数据目录

#修改这行：Environment=PGDATA=/var/lib/pgsql/13/data/ 改为新的路径，如下：

Environment=PGDATA=/data\_pg/data/

# systemctl daemon-reload

# systemctl restart postgresql-13

★PG-10版本

# systemctl stop postgresql #先停止服务

# mkdir /data\_pg

# chown postgres:postgres /data\_pg

# vi /var/lib/pgsql/data/postgresql.conf #pg-10版本，添加或修改以下配置

data\_directory = '/data\_pg/data/'

# mv /var/lib/pgsql/\* /data\_pg/ #将原数据目录下所有内容移动到新数据目录下

# vi /usr/lib/systemd/system/postgresql.service #pg-10版本，修改systemd服务文件，指定数据目录

#修改这行：Environment=PGDATA=/var/lib/pgsql/data/ 改为新的路径，如下：

Environment=PGDATA=/data\_pg/data/

# systemctl daemon-reload

# systemctl restart postgresql

**★第3章、设置监听ip及端口号**

pg默认监听127.0.0.1的ip及5432/tcp端口

★PG-13版本

# vi /var/lib/pgsql/13/data/postgresql.conf #pg-13版本

#若修改了存储路径，则在新的路径下**/data\_pg/data/postgresql.conf**，默认监听127.0.0.1:5432

listen\_addresses = '\*'

port = 5432

# vi /var/lib/pgsql/13/data/pg\_hba.conf #pg-13版本，添加以下一行表示允许所有客户端ip远程连接

#若修改了存储路径，则在新的路径下**/data\_pg/data/pg\_hba.conf**

host all all 0.0.0.0/0 password

# systemctl restart postgresql-13 #pg-13版本

★PG-10版本

# vi /var/lib/pgsql/data/postgresql.conf #pg-10版本

#若修改了存储路径，则在新的路径下**/data\_pg/data/postgresql.conf**

listen\_addresses = '\*'

port = 5432

# vi /var/lib/pgsql/data/pg\_hba.conf #pg-10版本，添加以下一行表示允许所有客户端ip远程连接

#若修改了存储路径，则在新的路径下**/data\_pg/data/pg\_hba.conf**

host all all 0.0.0.0/0 password

# systemctl restart postgresql #pg-10版本

**★防火墙放通5432/tcp**

# firewall-cmd --add-port=5432/tcp --permanent

# firewall-cmd --reload

**pg\_hba.conf配置**保存在postgres库下面的名为pg\_catalg的schema下的pg\_hba\_file\_rules表里：



**INSERT** **INTO** "pg\_hba\_file\_rules" ("line\_number", "type", "database", "user\_name", "address", "netmask", "auth\_method", "options", "error") **VALUES** (94, 'host', '{all}', '{all}', '0.0.0.0', '0.0.0.0', 'password', **NULL**, **NULL**);

第一个94为自动生成的规则编号，前面的0.0.0.0为网段，后面的0.0.0.0为掩码

**★第4章、创建用户并授权**

# su postgres

# psql #初次登录需要切换为postgres用户，直接登录，无需密码

postgres=# alter user postgres with password 'passwdxxyy'; #修改管理员账号postgres的密码

postgres=#

postgres=# create user cof with password 'passwdxxyy2'; #创建一个名为cof的用户，并设置密码

postgres=# grant all on database *db\_test* to cof; #给cof授权，有操作db\_test库的所有权限

postgres=#

postgres=# revoke all on database db\_test from cof; #撤销用户权限

postgres=# drop user xxx; #删除用户

postgres=# exit; #退出PG交互界面

# psql -h localhost -p 5432 -U postgres #使用postgres用户登录，需要密码，默认登录postgres库

postgres=# #提示符 => 表示普通账号，=#表示管理员账号

postgres=# #提示符=>前的postgres表示当前数据库名

# psql -h x.x.x.x -p 5432 -U cof -d db\_test #使用cof用户登录pg，连接db\_test这个库

db\_test=> #提示符 => 表示普通账号，=#表示管理员账号

db\_test=> #提示符=>前的db\_test表示当前数据库名

用户信息保存在postgres库下面的名为pg\_catalg的schema下的pg\_user表里：



**★第5章、使用客户端连接**

**①命令行客户端连接pg数据库**

# psql -h 10.99.1.122 -p 5432 -U postgres #连接pg数据库，

# -h指定地址，-p指定端口号，-U指定用户（默认为当前shell用户名），-W指定用户密码，

# -d指定数据库，默认连接的数据库名同用户名

Password for user postgres: #输入密码

psql (13.13)

Type "help" for help.

postgres=# #提示符#前的postgres表示当前数据库名

postgres=# select datname from pg\_database; #列出所有数据库（标准的sql语句）

 datname

-----------

 postgres

 template1

 template0

 db\_test

postgres=# \l #列出所有数据库（pg内置查询命令）

 List of databases

 Name | Owner | Encoding | Collate | Ctype | Access privileges

-----------+----------+----------+-------------+-------------+-----------------------

 db\_test | postgres | UTF8 | en\_US.UTF-8 | en\_US.UTF-8 |

 postgres | postgres | UTF8 | en\_US.UTF-8 | en\_US.UTF-8 |

 template0 | postgres | UTF8 | en\_US.UTF-8 | en\_US.UTF-8 | =c/postgres +

 | | | | | postgres=CTc/postgres

postgres=# select version(); #查看版本号

 version

----------------------------------------------------------------------------------------------------

 PostgreSQL 13.13 on x86\_64-pc-linux-gnu, compiled by gcc (GCC) 8.5.0 20210514 (Red Hat 8.5.0-18)...

postgres=# select usename from pg\_user; #查看系统用户

 usename

----------

 postgres

postgres=# exit; #退出连接，用quit;也行（标准的sql语句）

postgres=# \q #退出连接（pg内置命令）★不建议使用带\反斜杠的内置命令

**②可视化客户端连接pg数据库**

开源免费的客户端有HeidiSQL及pgAdmin

HeidiSQL官网： https://www.heidisql.com/

pgAdmin官网下载： https://www.pgadmin.org/download/

下载并安装最新版本的HeidiSQL，新建会话，网络类型为PostgreSQL(TCP/IP)，依赖库要>=目标pg数据库版本





下载最新版本的pgAdmin并安装运行





**★第6章、数据库及表基础操作**

pg数据库名称及表名称不区分大小写

**★库操作**

postgres=# create database db\_test; #新建1个名为db\_test的数据库

postgres=# select datname from pg\_database; #列出所有数据库（标准的sql语句）

postgres=# \l #列出所有数据库（pg内置命令）

 List of databases

 Name | Owner | Encoding | Collate | Ctype | Access privileges

-----------+----------+----------+-------------+-------------+-----------------------

 db\_test | postgres | UTF8 | en\_US.UTF-8 | en\_US.UTF-8 |

 postgres | postgres | UTF8 | en\_US.UTF-8 | en\_US.UTF-8 |

 template0 | postgres | UTF8 | en\_US.UTF-8 | en\_US.UTF-8 | =c/postgres +

 | | | | | postgres=CTc/postgres

 template1 | postgres | UTF8 | en\_US.UTF-8 | en\_US.UTF-8 | =c/postgres +

 | | | | | postgres=CTc/postgres

postgres=# drop database db\_namexx; #删除数据库，谨慎操作

**创建用户并授权：**

postgres=# create user user1 with password 'passwdxxyy3'; #创建一个名为user1的用户，并设置密码

postgres=# grant all on database *db\_test* to user1; #给user1授权，有操作db\_test库的所有权限

**★表操作**

postgres=# \c db\_test; #切换到目标数据库

db\_test=# create table tablexx (name VARCHAR(32), age INT, class INT); #创建名为tablexx的表，3个字段

db\_test=# \d tablexx; #查看表结构

 Table "public.tablexx"

 Column | Type | Collation | Nullable | Default

--------+-----------------------+-----------+----------+---------

 name | character varying(32) | | |

 age | integer | | |

 class | integer | | |

db\_test=# insert into tablexx values ('cof-lee', 28, 1511); #插入一条数据

db\_test=# insert into tablexx (name, age, class) values ('Tom', 23, 1512); #插入一条数据

db\_test=# **SELECT** \* **FROM** "public"."tablexx" **LIMIT** 1000; #列出表的所有数据

 name | age | class

---------+-----+-------

 cof-lee | 28 | 1511

 Tom | 23 | 1512

(2 rows)

db\_test=# drop table table\_namexx; #删除数据表（谨慎操作）

★说明：

"public"."tablexx"中的public不是数据库名称，而是schemaname，后面的tablexx为表名称

pg数据库，使用HeidiSQL连接到某个库后，一个数据库的结构为：

 

默认有4个schema，所有的表默认在名为public的schema下

**★存储空间使用量查询**

postgres=# select pg\_size\_pretty(pg\_tablespace\_size('pg\_default')); #查看PG使用的磁盘空间大小

 pg\_size\_pretty

----------------

 46 MB

postgres=# select pg\_database\_size('postgres'); #查看指定数据库大小，单位Byte

 pg\_database\_size

------------------

 8090159

postgres=# select pg\_size\_pretty (pg\_database\_size('postgres')); #查看指定数据库大小，单位经过优化显示

 pg\_size\_pretty

----------------

 7901 kB

#列出所有database的大小，单位经过优化显示

postgres=# select pg\_database.datname, pg\_size\_pretty (pg\_database\_size(pg\_database.datname)) AS size from pg\_database;

 datname | size

-----------+---------

 postgres | 7901 kB

 testdb | 7753 kB

 template1 | 7753 kB

 template0 | 7753 kB

 db\_test | 7901 kB

# select pg\_relation\_size('table\_name'); #查看表的大小，单位字节

# select pg\_size\_pretty(pg\_relation\_size('table\_name')); #以方便阅读的单位列出表的大小

★不建议使用带\反斜杠的内置命令，有些sql终端工具不支持发送带有\反斜杠的命令

**★第7章、表 字段操作**

**★第8章、表 数据操作**

db\_test=# \d tablexx; #查看表结构

 Table "public.tablexx"

 Column | Type | Collation | Nullable | Default

--------+-----------------------+-----------+----------+---------

 name | character varying(32) | | |

 age | integer | | |

 class | integer | | |

db\_test=# insert into tablexx values ('Neo', 22, 1511); #插入一条数据

db\_test=# delete from tablexx where name='Tom'; #删除一条数据

db\_test=# update tablexx set age=30 where name='Neo'; #修改一条数据

db\_test=# select \* from tablexx; #查询表的数据

 name | age | class

---------+-----+-------

 cof-lee | 28 | 1511

 Neo | 22 | 1511

(2 rows)

**★第9章、数据导入导出**

**★第10章、主从同步**