

Linux环境PostgreSQL源码编译安装

Linux环境PostgreSQL源码编译安装

Linux版本: Red Hat 6.5

PostgreSQL版本: postgresql-9.6.2.tar.gz

数据存放目录: /var/postgresql/data

软件安装位置: /usr/local/postgresql

依赖gcc环境,

yum install gcc (需要配置yum源, iso源即可)

安装步骤及脚本:

#创建用户和组

```
groupadd postgres
```

```
useradd -g postgres postgres
```

#进入安装目录

```
cd /usr/local/
```

#删除原有安装

```
rm -rf postgresql
```

#解压

路径 /usr/local/postgresql

```
tar zxvf postgresql-9.6.2.tar.gz
```

```
cd postgresql
```

#配置

```
./configure --prefix=/usr/local/postgresql --without-readline --without-zlib
```

#编译安装

```
make
```

```
make install
```

#安装contrib目录下的一些工具, 是第三方组织的一些工具代码, 建议安装

```
cd /usr/local/postgresql/contrib
```

```
make
```

```
make install
```

#添加环境变量, 并使之生效 (初级用户请自行学习设置)

```
export PATH=/usr/local/postgresql/bin:$PATH
```

#建postgres数据库的目录

```
mkdir -p /var/postgresql/data
```

```
chown postgres:postgres /var/postgresql/*
```

```
chown postgres:postgres /usr/local/postgresql/*
```

```
chmod -R 775 /var/postgresql/*
```

#切换用户

```
su postgres
```

#初始化数据库

```
initdb -D /var/postgresql/data
```

#启动服务

```
pg_ctl -D /var/postgresql/data -l /var/postgresql/log/pg_server.log start
```

```
ps -A | grep postgres
```

```
[root@localhost ~]#  
[root@localhost ~]# tail -100f /var/postgresql/log/pg_server.log  
FATAL: data directory "/var/postgresql/data" has group or world access  
DETAIL: Permissions should be u=rwx (0700).  
LOG: database system was shut down at 2017-09-05 23:35:09 EDT  
LOG: MultiXact member wraparound protections are now enabled  
LOG: database system is ready to accept connections  
LOG: autovacuum launcher started  
LOG: received fast shutdown request  
LOG: aborting any active transactions  
LOG: autovacuum launcher shutting down  
LOG: shutting down  
LOG: database system is shut down  
LOG: database system was shut down at 2017-09-06 00:37:14 EDT  
LOG: MultiXact member wraparound protections are now enabled  
LOG: autovacuum launcher started  
LOG: database system is ready to accept connections  
ERROR: column "rolcatupdate" does not exist at character 67  
STATEMENT: SELECT rolname, rolsuper, rolinherit, rolcreatorole, rolcreatedb, rolcatupdate, rolcanlogin, rolconnlimit, rolvaliduntil, rolconfig, oid FROM pg_roles  
STATEMENT: SELECT rolname, rolsuper, rolinherit, rolcreatorole, rolcreatedb, rolcatupdate, rolcanlogin, rolconnlimit, rolvaliduntil, rolconfig, oid FROM pg_roles  
ERROR: column "rolcatupdate" does not exist at character 67  
STATEMENT: SELECT rolname, rolsuper, rolinherit, rolcreatorole, rolcreatedb, rolcatupdate, rolcanlogin, rolconnlimit, rolvaliduntil, rolconfig, oid FROM pg_roles  
ERROR: column "rolcatupdate" does not exist at character 67  
STATEMENT: SELECT rolname, rolsuper, rolinherit, rolcreatorole, rolcreatedb, rolcatupdate, rolcanlogin, rolconnlimit, rolvaliduntil, rolconfig, oid FROM pg_roles  
ERROR: column "rolcatupdate" does not exist at character 67  
STATEMENT: SELECT rolname, rolsuper, rolinherit, rolcreatorole, rolcreatedb, rolcatupdate, rolcanlogin, rolconnlimit, rolvaliduntil, rolconfig, oid FROM pg_roles  
ERROR: column "rolcatupdate" does not exist at character 67  
STATEMENT: SELECT rolname, rolsuper, rolinherit, rolcreatorole, rolcreatedb, rolcatupdate, rolcanlogin, rolconnlimit, rolvaliduntil, rolconfig, oid FROM pg_roles  
LOG: received SIGHUP, reloading configuration files
```

以上通过日志观察第一次启动存在权限问题，修正后：chmod 700 /var/postgresql/data

重新启动正常

#创建测试数据库

```
createdb test;
```

#连接数据库

```
psql -U postgres
```

```
[root@localhost ~]# psql -U postgres  
psql (9.6.2)  
type "help" for help.  
  
postgres=# \c
```

#切换到david 数据库

```
\c test
```

```
postgres=# \c test;  
You are now connected to database "test" as user "postgres".  
test=#
```

#创建测试表

```
create table test (id integer, name text);
```

#插入测试数据

```
insert into test values (1,'david');
```

#选择数据

```
select * from test ;
```

#修改linux 系统用户postgres 的密码

```
passwd postgres
```

```
#postgres
```

#修改PostgreSQL 数据库配置实现远程访问

#修改postgresql.conf 文件

```
vi /var/postgresql/data/postgresql.conf
```

```

#external_pid_file = # write an extra PID file
# (change requires restart)

#-----
# CONNECTIONS AND AUTHENTICATION
#-----

# - Connection Settings -
listen_addresses = '*' # what IP address(es) to listen on;
# comma-separated list of addresses;
# defaults to 'localhost'; use '*' for all
# (change requires restart)
port = 5432 # (change requires restart)
max_connections = 100 # (change requires restart)
#superuser_reserved_connections = 3 # (change requires restart)
#unix_socket_directories = '/tmp' # comma-separated list of directories
# (change requires restart)
#unix_socket_group = '' # (change requires restart)
#unix_socket_permissions = 0777 # begin with 0 to use octal notation
# (change requires restart)
#bonjour = off # advertise server via Bonjour
# (change requires restart)
#bonjour_name = '' # defaults to the computer name
# (change requires restart)

# - Security and Authentication -

```

#-----允许远程连接-----

#修改客户端认证配置文件pg_hba.conf，将需要远程访问数据库的IP地址或地址段加入该文件

vi /var/postgresql/data/pg_hba.conf

#在文件的86行修改下面的这句话

```

67 # Put your actual configuration here
68 # -----
69 #
70 # If you want to allow non-local connections, you need to add more
71 # "host" records. In that case you will also need to make PostgreSQL
72 # listen on a non-local interface via the listen_addresses
73 # configuration parameter, or via the -i or -h command line switches.
74 #
75 # CAUTION: Configuring the system for local "trust" authentication
76 # allows any local user to connect as any PostgreSQL user, including
77 # the database superuser. If you do not trust all your local users,
78 # use another authentication method.
79 #
80 #
81 # TYPE DATABASE USER ADDRESS METHOD
82 #
83 # "local" is for Unix domain socket connections only
84 local all all trust
85 # IPv4 local connections:
86 host all all 0.0.0.0/0 trust
87 # IPv6 local connections:
88 host all all ::1/128 trust
89 # Allow replication connections from localhost, by a user with the
90 # replication privilege.
91 #local replication postgres trust
92 #host replication postgres 127.0.0.1/32 trust
93 #host replication postgres ::1/128 trust

```

```
host all all 0.0.0.0/0 trust
```

备注：设置 **trust**，本地可以使用psql -U postgres直接登录服务器；设置 **peer**，本地可以使用psql -h 127.0.0.1 -d postgres -U postgres直接登录服务器；

#重启服务

```
pg_ctl -D /var/postgresql/data -l /var/postgresql/log/pg_server.log restart
tail -100f /var/postgresql/log/pg_server.log
```

#停止服务

```
pg_ctl -D /var/postgresql/data -l /var/postgresql/log/pg_server.log stop
```

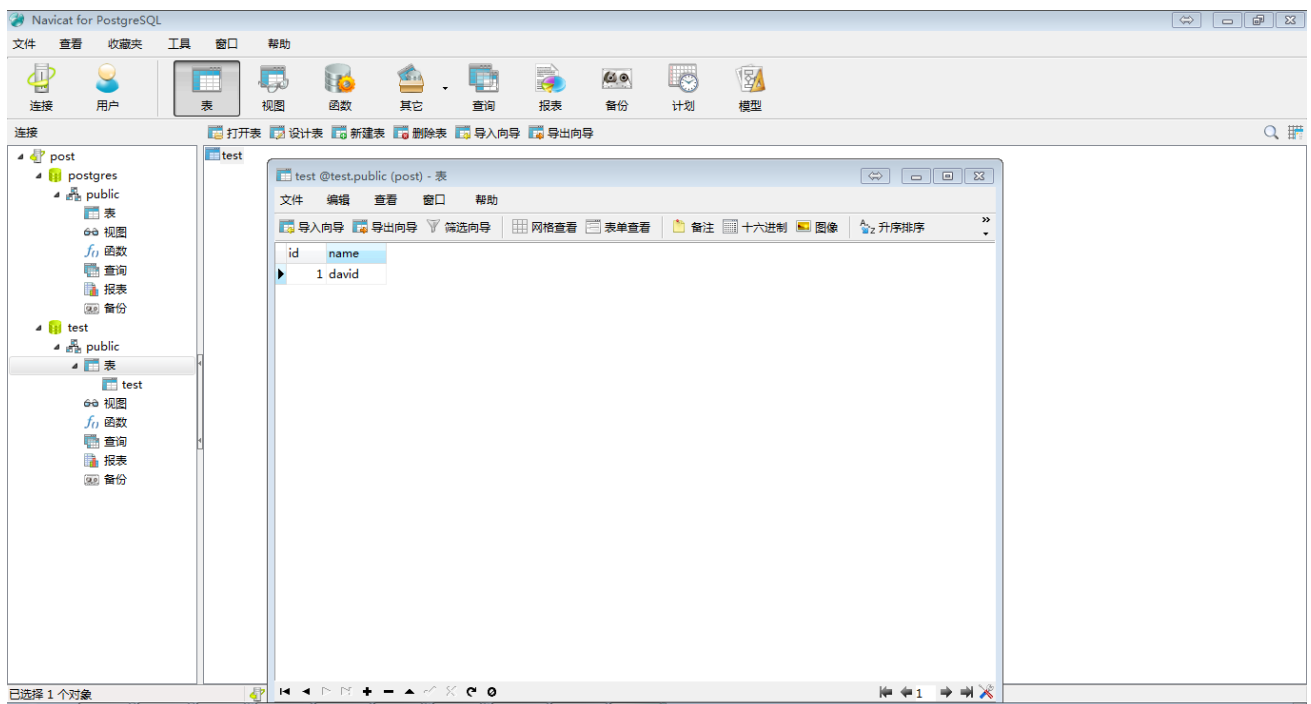
#端口是否启用

```
netstat -anp | grep 5432
```

ALTER USER postgres WITH PASSWORD 'postgres'; //添加密码

select * from pg_shadow;

\q回车 退出登录



常用操作:

有三种方式可以查看postgresql运行是否运行

`ps -ef | grep postgres` 查postgres的进程信息

`netstat -tlnl | grep 5432` 查postgres的端口号5432是否已经打开

`service postgresql-9.4 status` 查看服务状态

`psql -U postgres //连接pgsql server`

`psql -h 127.0.0.1 -d postgres -U postgres //连接pgsql server`

`select pg_reload_conf();`

`show data_directory;`

