

Postgresql高可用方案

Architecture

1. Operating System

```
[root@pgpool01 ~]# cat /etc/os-release
NAME="CentOS Linux"
VERSION="7 (Core)"
ID="centos"
ID_LIKE="rhel fedora"
VERSION_ID="7"
PRETTY_NAME="CentOS Linux 7 (Core)"
ANSI_COLOR="0;31"
CPE_NAME="cpe:/o:centos:centos:7"
HOME_URL="https://www.centos.org/"
BUG_REPORT_URL="https://bugs.centos.org/"

CENTOS_MANTISBT_PROJECT="CentOS-7"
CENTOS_MANTISBT_PROJECT_VERSION="7"
REDHAT_SUPPORT_PRODUCT="centos"
REDHAT_SUPPORT_PRODUCT_VERSION="7"
```

2. Hostname and IP address

Hostname	IP Address	Virtual IP
pgpool01	192.168.0.2	192.168.0.6
pgpool02	192.168.0.3	
postgresql01	192.168.0.4	
postgresql02	192.168.0.5	

3. Configure Hosts

Edit hosts file

```
cat >> /etc/hosts << EOT
192.168.0.2    pgpool01
192.168.0.3    pgpool02
192.168.0.4    postgresql01
192.168.0.5    postgresql02
EOT
```

4. Configure ssh passwordless on all servers

```
[root@pgpool01 ~]# cd ~/.ssh
[root@pgpool01 ~]# ssh-keygen -t rsa -f id_rsa_pgpool
[root@pgpool01 ~]# ssh-copy-id -i id_rsa_pgpool.pub postgres@pgpool01
[root@pgpool01 ~]# ssh-copy-id -i id_rsa_pgpool.pub postgres@pgpool02
[root@pgpool01 ~]# ssh-copy-id -i id_rsa_pgpool.pub postgres@postgresql01
[root@pgpool01 ~]# ssh-copy-id -i id_rsa_pgpool.pub postgres@postgresql02

[root@pgpool01 ~]# su - postgres
[root@pgpool01 ~]# cd ~/.ssh
[root@pgpool01 ~]# ssh-keygen -t rsa -f id_rsa_pgpool
[root@pgpool01 ~]# ssh-copy-id -i id_rsa_pgpool.pub postgres@pgpool01
[root@pgpool01 ~]# ssh-copy-id -i id_rsa_pgpool.pub postgres@pgpool02
[root@pgpool01 ~]# ssh-copy-id -i id_rsa_pgpool.pub postgres@postgresql01
[root@pgpool01 ~]# ssh-copy-id -i id_rsa_pgpool.pub postgres@postgresql02
```

Install

1. Install PostgreSQL 12.5:

```
sudo yum install -y https://download.postgresql.org/pub/repos/yum/reporpms/EL-7-
x86_64/pgdg-redhat-repo-latest.noarch.rpm
sudo yum install -y postgresql12-server
sudo /usr/pgsql-12/bin/postgresql-12-setup initdb
sudo systemctl enable postgresql-12
sudo systemctl start postgresql-12
psql --version    # display PostgreSQL version
psql (PostgreSQL) 12.5

# In order to use the online recovery functionality, we need install follow:
yum install -y https://www.pgpool.net/yum/rpms/4.1/redhat/rhel-7-x86_64/pgpool-II-
release-4.1-2.noarch.rpm
yum install -y pgpool-II-pg12-extensions
```

2. Install pgpool-II-release-4.1.5:

<https://www.pgpool.net/docs/latest/en/html/install-rpm.html>

```
yum install -y https://www.pgpool.net/yum/rpms/4.1/redhat/rhel-7-x86_64/pgpool-II-
release-4.1-2.noarch.rpm
yum install -y pgpool-II-pg12
yum install -y pgpool-II-pg12-debuginfo
yum install -y pgpool-II-pg12-devel
# On all the PostgreSQL servers you need to install:
yum install -y pgpool-II-pg12-extensions      # not install, don't need
pgpool --version      # display pgpool version
# pgpool-II version 4.1.5 (karasukiboshi)

# Starting/stopping Pgpool-II
mkdir -p /var/log/pgpool      # create log directory
chown -R postgres:postgres /var/log/pgpool      # 赋予相应读写权限
systemctl enable pgpool.service
systemctl start pgpool.service

# install postgresql client for getting the test command
sudo yum install -y https://download.postgresql.org/pub/repos/yum/repos/EL-7-
x86_64/pgdg-redhat-repo-latest.noarch.rpm
sudo yum install -y postgresql10
```

3. Resolve the problems

Encounter an error startup pgpool "FATAL: failed to bind a socket:
"/var/run/postgresql/.s.PGSQL.9999"",

You can do some modify "pgpool.conf" file to fix it.

Change default setting

```
socket_dir = '/var/run/postgresql' pcp_socket_dir = '/var/run/postgresql' wd_ipc_socket_dir =
'/var/run/postgresql'
```

to

```
socket_dir = '/tmp' pcp_socket_dir = '/tmp' wd_ipc_socket_dir = '/tmp'
```

Save then restart pgpool.

Configure

<https://www.pgpool.net/docs/pgpool-II-4.1.5/en/html/example-cluster.html>

1. Set up PostgreSQL streaming replication on the primary server

In this project, we use WAL archiving.

```
# Edit the config file /var/lib/pgsql/12/data/postgresql.conf with follow content
listen_addresses = '*'
archive_mode = on
archive_command = 'cp "%p" "/var/lib/pgsql/archivedir/%f"'
max_wal_senders = 10
max_replication_slots = 10
wal_level = replica
hot_standby = on
wal_log_hints = on
```

Because of the security reasons, we create a user `repl` solely used for replication purpose, and a user `pgpool` for streaming replication delay check and health check of Pgpool-II.

User Name	Password	Detail
repl	repl	PostgreSQL replication user
pgpool	pgpool	Pgpool-II health check (health_check_user) and replication delay check (sr_check_user) user
postgres	postgres	User running online recovery

```
[server1]# sudo -u postgres psql postgres
postgres=# SET password_encryption = 'scram-sha-256';
postgres=# CREATE ROLE pgpool WITH LOGIN;
postgres=# CREATE ROLE repl WITH REPLICATION LOGIN;
postgres=# \password pgpool
postgres=# \password repl
postgres=# \password postgres
```

If you want to show "replication_state" and "replication_sync_state" column in [SHOW POOL NODES](#) command result, role `pgpool` needs to be PostgreSQL super user or or in `pg_monitor` group (Pgpool-II 4.1 or later). Grant `pg_monitor` to `pgpool`:

```
GRANT pg_monitor TO pgpool;
```

2. Postgresql and Pgpool manage command:

```
systemctl status postgresql-12
systemctl start postgresql-12
systemctl stop postgresql-12
systemctl restart postgresql-12
journalctl -f -u postgresql-12.service
```

```
systemctl status pgpool.service
systemctl start pgpool.service
systemctl restart pgpool.service
systemctl stop pgpool.service
journalctl -f -u pgpool.service
```

The way to judge if a postgresql server is master or slave

1. Display the database cluster state

```
[root@postgresql01 ~]# /usr/pgsql-12/bin/pg_ctlcluster -D /var/lib/pgsql/12/data/ |
grep 'Database cluster state'
Database cluster state:           in archive recovery      # postgresql server is
slave

[root@postgresql02 data]# /usr/pgsql-12/bin/pg_ctlcluster -D /var/lib/pgsql/12/data/
| grep 'Database cluster state'
Database cluster state:           in production          # postgresql server is
master
```

2. Display the walsender or walreceiver status

```
[root@postgresql02 data]# ps -ef | grep postgres | grep 'walsender\|walreceiver' |
grep -v grep
postgres 25856 21767 0 11:14 ?          00:00:00 postgres: walsender repl
192.168.0.4(48546) streaming 0/C01F358
# display "walsender", the server is master

[root@postgresql01 ~]# ps -ef | grep postgres | grep 'walsender\|walreceiver' | grep -
v grep
postgres 7822 7646 0 11:14 ?          00:00:00 postgres: walreceiver streaming
0/C01F358
# display "walreceiver", the server is slave
```

3. Execute pg_is_in_recovery function

```
postgres=# select pg_is_in_recovery();
 pg_is_in_recovery 
-----
 f
(1 row)
# display f is master

postgres=# select pg_is_in_recovery();
 pg_is_in_recovery 
-----
 t
(1 row)
# display t is slave
```

4. Select from master server table pg_stat_replication

```
postgres=# select pid,application_name,client_addr,client_port,state,sync_state from
pg_stat_replication;

 pid | application_name | client_addr | client_port | state | sync_state 
-----+-----+-----+-----+-----+-----
 1401 | walreceiver      | 10.1.83.135 | 48860       | streaming | async
```

在主机字典表中是能查到记录，备机中是查询不到的。

Verify that system are working normally

Run the follow command at any server (pgpool01, pgpool02, postgresql01, postgresql02):

```
#
[root@pgpool-01 pgpool-II]# psql -h 192.168.0.6 -p 9999 -U pgpool postgres -c "show
pool_nodes"
Password for user pgpool:
 node_id | hostname | port | status | lb_weight | role | select_cnt |
load_balance_node | replication_delay | replication_state | re
plication_sync_state | last_status_change
-----+-----+-----+-----+-----+-----+-----+-----
 0       | postgresql01 | 5432 | up     | 0.500000 | standby | 0           |
| 0           |           |           |           |           |           | false
```

```

| 2020-12-22 11:54:14
1 | postgresql02 | 5432 | up | 0.500000 | primary | 0 | true
| 0 | | |
| 2020-12-22 11:54:14
(2 rows)
[root@pgpool-01 pgpool-II]# pcp_watchdog_info -h 192.168.0.6 -p 9898 -U pgpool
Password:
2 YES 192.168.0.2:9999 Linux pgpool-01 192.168.0.2

192.168.0.2:9999 Linux pgpool-01 192.168.0.2 9999 9000 4 MASTER
192.168.0.3:9999 Linux pgpool-02 192.168.0.3 9999 9000 7 STANDBY

```

Stop the pgpool-01's pgpool master

```

[root@pgpool-01 pgpool-II]# systemctl stop pgpool.service
[root@pgpool-01 pgpool-II]# pcp_watchdog_info -h 192.168.0.6 -p 9898 -U pgpool
Password:
2 YES 192.168.0.3:9999 Linux pgpool-02 192.168.0.3

192.168.0.3:9999 Linux pgpool-02 192.168.0.3 9999 9000 4 MASTER
192.168.0.2:9999 Linux pgpool-01 192.168.0.2 9999 9000 10 SHUTDOWN
[root@pgpool-01 pgpool-II]#

```

Start the pgpool-01's pgpool service

```

[root@pgpool-01 pgpool-II]# systemctl start pgpool.service
[root@pgpool-01 pgpool-II]# pcp_watchdog_info -h 192.168.0.6 -p 9898 -U pgpool
Password:
2 YES 192.168.0.3:9999 Linux pgpool-02 192.168.0.3

192.168.0.3:9999 Linux pgpool-02 192.168.0.3 9999 9000 4 MASTER
192.168.0.2:9999 Linux pgpool-01 192.168.0.2 9999 9000 7 STANDBY
[root@pgpool-01 pgpool-II]# psql -h 192.168.0.6 -p 9999 -U pgpool postgres -c "show
pool_nodes"
Password for user pgpool:
 node_id | hostname | port | status | lb_weight | role | select_cnt |
load_balance_node | replication_delay | replication_state | re
plication_sync_state | last_status_change
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
0 | postgresql01 | 5432 | up | 0.500000 | standby | 0 | false
| 0 | | |
| 2020-12-22 12:03:54
1 | postgresql02 | 5432 | up | 0.500000 | primary | 0 | true
| 0 | | |
| 2020-12-22 12:03:54
(2 rows)

```

```
[root@pgpool-01 pgpool-II]#
```

Failover

First, use `psql` to connect to PostgreSQL via virtual IP, and verify the backend information.

```
[root@pgpool-01 pgpool-II]# psql -h 192.168.0.6 -p 9999 -U pgpool postgres -c "show pool_nodes"
```

Password for user pgpool:

node_id	hostname	port	status	lb_weight	role	select_cnt	load_balance_node	replication_delay	replication_state	replication_sync_state	last_status_change
0	postgresql01	5432	up	0.500000	standby	0	0				2020-12-22 12:03:54
1	postgresql02	5432	up	0.500000	primary	1	0				2020-12-22 12:03:54

(2 rows)

```
[root@pgpool-01 pgpool-II]#
```

Next, stop primary PostgreSQL server `postgresql02`, and verify automatic failover.

```
[root@postgresql02 data]# systemctl stop postgresql-12
```

After stopping PostgreSQL on `postgresql02`, failover occurs and PostgreSQL on `postgresql01` becomes new primary DB.


```
[root@postgresql02 data]# psql -h 192.168.0.6 -p 9999 -U pgpool postgres -c "show pool_nodes"
```

Password for user pgpool:

node_id	hostname	port	status	lb_weight	role	select_cnt	load_balance_node	replication_delay	replication_state	replication_sync_state	last_status_change
0	postgresql01	5432	up	0.500000	primary	0	0				2020-12-22 12:19:17
1	postgresql02	5432	down	0.500000	standby	1	1				2020-12-22 12:19:17

(2 rows)

```
[root@postgresql02 data]#
```

postgresql01 is running as new primary.

```
[root@postgresql02 data]# psql -h postgresql01 -p 5432 -U pgpool postgres -c "select pg_is_in_recovery()"
```

Password for user pgpool:

pg_is_in_recovery
f

(1 row)

```
[root@postgresql02 data]#
```

Online Recovery

```
[root@pgpool-01 pgpool-II]# psql -h 192.168.0.6 -p 9999 -U pgpool postgres -c "show pool_nodes"
Password for user pgpool:
 node_id | hostname | port | status | lb_weight | role | select_cnt |
load_balance_node | replication_delay | replication_s
tate | replication_sync_state | last_status_change
-----+-----+-----+-----+-----+-----+-----+-----
-----+-----+-----+-----+-----+-----+-----+-----
0        | postgresql01 | 5432 | up      | 0.500000 | primary | 0          | true
        | 0            |      |         |          |         |            |
        |              |      | 2020-12-22 12:19:17
1        | postgresql02 | 5432 | down    | 0.500000 | standby | 1          | false
        | 0            |      |         |          |         |            |
        |              |      | 2020-12-22 12:19:17
(2 rows)

[root@pgpool-01 pgpool-II]#
```

Next, run the command to execute script `recovery_1st_stage` and then `pgpool_remote_start`.

```
pcp_recovery_node -h 192.168.0.6 -p 9898 -U pgpool -n 1
```

We can use command `tail -f /var/log/postgres_shell.log` on `postgresql01` to watch the process.

数据操作命令

1. 创建数据库和表并插入数据

```
postgres=# create database test;
postgres=# \c test
test=# create table tt(id serial not null,name text);
test=# insert into tt(name) values ('china');
test=# \q
```

2. postgresql用户创建相关

```
sudo -u postgres psql postgres
psql -U username -d database_name -h host -p port -W # 指定参数登录
# 参数含义: -U指定用户 -d要连接的数据库 -h要连接的主机 -w提示输入密码。

create user dbuser # 创建用户
create database exampledb owner dbuser # 创建数据库
```

参考

[Watchdog01](#) [Watchdog02](#) [RECOVERY.CONF IS GONE IN POSTGRESQL V12](#) [PostgreSQL master-slave database replication](#) [pg_rewind](#) [PostgreSQL流复制热备](#) [PGPool-II+PG流复制实现HA](#) [linux日志logger命令详解](#) [postgresql replication slots 的一些个人理解](#) [PostgreSQL常用命令01](#) [PostgreSQL常用命令02](#)