PostgreSQL version:13.0

PGPool version:pgpool-II version 4.4.0

OS :Redhat Linux9

Primary_Server IP:172.31.34.171

Standby_Server IP:172.31.37.126

STEP1)POSTGRESQL INSTALLATION FROM SOURCE CODE ON EACH SERVER

i)On Primary_Server

wget https://ftp.postgresql.org/pub/source/v13.0/postgresql-13.0.tar.gz

tar -xvzf postgresql-13.0.tar.gz

cd /home/ec2-user/postgresql-13.0

yum install cc* gcc* readline-devel zlib-devel

./configure

make

su

make install

adduser postgres

mkdir /usr/local/pgsql/data

chown postgres:postgres /usr/local/pgsql/data

su - postgres

/usr/local/pgsql/bin/initdb -D /usr/local/pgsql/data

/usr/local/pgsql/bin/pg_ctl -D /usr/local/pgsql/data -l logfile start

Environment Variables Setup

vim ~./bash_profile

User specific environment and startup programs

export PATH=/usr/local/pgsql/bin:$PATH

export PGDATA=/usr/local/pgsql/data

export PGUSER=postgres

export PGPORT=5432

export PGDATABASE=postgres

save and exit

Run . ~./bash_profile or source ~./bash_profile

ii)On Standby_Server

wget https://ftp.postgresql.org/pub/source/v13.0/postgresql-13.0.tar.gz

tar -xvzf postgresql-13.0.tar.gz

cd /home/ec2-user/postgresql-13.0

yum install cc* gcc* readline-devel zlib-devel

./configure

make

su

make install

adduser postgres

mkdir /usr/local/pgsql/data

chown postgres:postgres /usr/local/pgsql/data

su - postgres

/usr/local/pgsql/bin/initdb -D /usr/local/pgsql/data

/usr/local/pgsql/bin/pg_ctl -D /usr/local/pgsql/data -l logfile start

Environment Variables Setup

vim ~./bash_profile

User specific environment and startup programs

export PATH=/usr/local/pgsql/bin:$PATH

export PGDATA=/usr/local/pgsql/data

export PGUSER=postgres

export PGPORT=5432

export PGDATABASE=postgres

save and exit

Run . ~./bash_profile or source ~./bash_profile

STEP2)STREAMING REPLICATION SETUP

i)Edit the configuration file $PGDATA/postgresql.conf on Primary_Server

listen_addresses = '*'

port=5432

archive_mode = on

archive_command = 'rsync -a %p /mnt/server/archivedir/%f'

max_wal_senders = 10

max_replication_slots = 10

wal_level = replica

hot_standby = on

save&exit

ii)Create Replication user on the Primary_Server

CREATE ROLE replica_user WITH REPLICATION LOGIN PASSWORD 'password';

iii)Edit $PGDATA/pg_hba.conf on Primary_Server

host replication all 172.31.37.126/32 md5

host all all 172.31.37.126/32 md5

host replication all 172.31.34.171/32 md5

host all all 172.31.34.171/32 md5

save&exit

iv)Create archive log directory and assign ownership

mkdir -p /mnt/server/archivedir/

chown postgres:postgres /mnt/server/archivedir/

v)Restart PostgreSQL services on Primary_Server

su - postgres

/usr/local/pgsql/bin/pg_ctl -D /usr/local/pgsql/data -l logfile restart

vi)Setup SSH Passwordless Login on both server

su - postgres

ssh-keygen -t rsa

ssh-copy-id postgres@IP

vii)Stop Postgresql services on Standby_Server

su - postgres

/usr/local/pgsql/bin/pg_ctl -D /usr/local/pgsql/data -l logfile stop

viii)Remove existing files in a Standby_Server data directory

su - postgres

rm -rf /usr/local/pgsql/data/*

ix)Run pg_basebackup on Standby_Server to copy data from Primary_Server to Standby_Server

su - postgres

pg_basebackup -h 172.31.34.171 -U replica_user -X stream -C -S replica_1 -v -R -W -D /usr/local/pgsql/data/

-h: This option specifies the host, in this case, the IP address of the Primary_Server.

-U: The option specifies the replication user. This is the user that was configured on the Primary_Server and which will be used by the Standby_Server to connect to it. In our case, the replication user is called replica_user.

-X: The option along with the stream value instructs the pg_basebackup utility to stream and include the WAL files in the backup.

-C: The option allows you to create a replication slot before the backup gets underway. The option is used along with the -S option to specify the slot name. In this case, our replication slot is called replica_1.

-v: This prints out verbose output that indicates the progress of the backup process from the Primary_Server to the replica.

-R: The option creates two files; an empty recovery configuration file called standby.signal and a Primary_Server connection settings file called postgresql.auto.conf. The standby.signal file contains connection information about the Primary_Server and the postgresql.auto.conf file informs your replica cluster that it should operate as a standby server.

-W: This prompts you to provide a password for the replica_user replication user.

-D: Lastly, the -D option allows you to include the directory that you want to export the backup files.

xi)Edit $PGDATA/postgresql.conf on Standby_Server

restore_command = 'rsync -a postgres@172.31.34.171:/mnt/server/archivedir/%f %p'

recovery_target_timeline = 'latest'

save&exit

xii)Start the PostgreSQL services on Standby_Server

su - postgres

/usr/local/pgsql/bin/pg_ctl -D /usr/local/pgsql/data -l logfile start

xiii)Verify the status on Primary_Server

su - postgres

psql -p 5432

select * from pg_stat_replication;

##

postgres=# create database test;

CREATE DATABASE

postgres=# \c test

You are now connected to database "test" as user "postgres".

test=# create table test_table(id int,name varchar(20));

CREATE TABLE

test=# insert into test_table values(1,'Naveen'),(2,'Ramu');

INSERT 0 2

test=# select * from test_table;

 id | name

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

  1 | Naveen

  2 | Ramu

(2 rows)

test=#

##

xiv)Verify the status on Standby_Server

su - postgres

psql -p 5432

select * from pg_stat_wal_receiver;

##

```
[postgres@ip-172-31-32-192 ~]$ psql
psql (13.0)
Type "help" for help.
postgres=# \c test
You are now connected to database "test" as user "postgres".
test=# select * from test_table;
 id |  name
----+--------
  1 | Naveen
  2 | Ramu
(2 rows)


test=#
```

##

STEP3)PGPOOL-II INSTALLATION FROM SOURCE CODE ON PRIMARY_SERVER

i)pgpool-II Installation

```
wget https://www.pgpool.net/mediawiki/download.php?f=pgpool-II-4.4.0.tar.gz
tar -xvzf download.php\?f\=pgpool-II-4.4.0.tar.gz
cd pgpool-II-4.4.0/
./configure
make
make install
mkdir /var/run/pgpool/
```

ii) pgpool-II Configuration

```
bin_path :/usr/local/bin/
Config_files_path:/usr/local/etc/
```

Sample Configuration files:

cd /usr/local/etc/

-rw-r--r--. 1 root root 52K Feb 10 00:28 pgpool.conf.sample

-rw-r--r--. 1 root root 858 Feb 10 00:28 pcp.conf.sample

-rw-r--r--. 1 root root 3.4K Feb 10 00:28 pool_hba.conf.sample

-rw-r--r--. 1 root root 2.8K Feb 10 00:28 failover.sh.sample

-rw-r--r--. 1 root root 6.9K Feb 10 00:28 follow_primary.sh.sample

-rw-r--r--. 1 root root 3.4K Feb 10 00:28 recovery_1st_stage.sample

-rw-r--r--. 1 root root 1.2K Feb 10 00:28 pgpool_remote_start.sample

-rw-r--r--. 1 root root 1.2K Feb 10 00:28 replication_mode_recovery_2nd_stage.sample

-rw-r--r--. 1 root root 3.3K Feb 10 00:28 replication_mode_recovery_1st_stage.sample

-rw-r--r--. 1 root root 643 Feb 10 00:28 escalation.sh.sample

-rw-r--r--. 1 root root 2.5K Feb 10 00:28 aws_rtb_if_cmd.sh.sample

-rw-r--r--. 1 root root 1.6K Feb 10 00:28 aws_eip_if_cmd.sh.sample

a)Copy pgpool.conf.sample to pgpool.conf

cp pgpool.conf.sample pgpool.conf

b)Edit /usr/local/etc/pgpool.conf

#--

CONNECTIONS

#--

- pgpool Connection Settings -

listen_addresses = '*'

port = 9999

- pgpool Communication Manager Connection Settings -

```
pcp_listen_addresses = '*'
pcp_port = 9898

# - Backend Connection Settings -

#Primary_Server
backend_hostname0 = '172.31.34.171'
backend_port0 = 5432
backend_weight0 = 1

#Standby_Server
backend_hostname1 = '172.31.37.126'
backend_port1 = 5432
backend_weight1 = 1

#------------------------------------------------------------------------------
# LOGS
#------------------------------------------------------------------------------
# - What to log -
log_statement = on
log_per_node_statement = on

#------------------------------------------------------------------------------
# REPLICATION MODE
#------------------------------------------------------------------------------

replicate_select = on

#------------------------------------------------------------------------------
# LOAD BALANCING MODE
```

```
#------------------------------------------------------------------------------

load_balance_mode = on

#------------------------------------------------------------------------------
# STREAMING REPLICATION MODE
#------------------------------------------------------------------------------

sr_check_user = 'postgres'
sr_check_password = '' #Leave it empty

#------------------------------------------------------------------------------
# HEALTH CHECK GLOBAL PARAMETERS
#------------------------------------------------------------------------------

health_check_timeout = 10
health_check_user = 'postgres'
health_check_password = '' #Leave it empty
save&exit
```

b)add postgresql user with md5 password in pool_password file

```
postgres=# alter user postgres password 'password';
ALTER ROLE

postgres=# select passwd from pg_shadow where usename ='postgres';
            passwd
-------------------------------------
 md56e59194ac6c1b45bc912c48358c38a41
```

vim /usr/local/etc/pool_passwd

postgres:md56e59194ac6c1b45bc912c48358c38a41

save&exit

iii)Start pgpool

##
##

[root@ip-172-31-45-179 etc]# /usr/local/bin/pgpool -n &

[1] 32105

[root@ip-172-31-45-179 etc]# 2024-02-12 09:41:11.019: main pid 32105: LOG: health_check_stats_shared_memory_size: requested size: 12288

2024-02-12 09:41:11.019: main pid 32105: LOG: memory cache initialized

2024-02-12 09:41:11.019: main pid 32105: DETAIL: memcache blocks :64

2024-02-12 09:41:11.019: main pid 32105: LOG: allocating (136981824) bytes of shared memory segment

2024-02-12 09:41:11.019: main pid 32105: LOG: allocating shared memory segment of size: 136981824

2024-02-12 09:41:11.110: main pid 32105: LOG: health_check_stats_shared_memory_size: requested size: 12288

2024-02-12 09:41:11.110: main pid 32105: LOG: health_check_stats_shared_memory_size: requested size: 12288

2024-02-12 09:41:11.110: main pid 32105: LOG: memory cache initialized

2024-02-12 09:41:11.110: main pid 32105: DETAIL: memcache blocks :64

2024-02-12 09:41:11.112: main pid 32105: LOG: pool_discard_oid_maps: discarded memqcache oid maps

2024-02-12 09:41:11.117: main pid 32105: LOG: unix_socket_directories[0]: /tmp/.s.PGSQL.9999

2024-02-12 09:41:11.117: main pid 32105: LOG: listen address[0]: *

2024-02-12 09:41:11.117: main pid 32105: LOG: Setting up socket for 0.0.0.0:9999

2024-02-12 09:41:11.117: main pid 32105: LOG: Setting up socket for :::9999

2024-02-12 09:41:11.121: main pid 32105: LOG: find_primary_node_repeatedly: waiting for finding a primary node

2024-02-12 09:41:11.130: main pid 32105: LOG: find_primary_node: primary node is 0

2024-02-12 09:41:11.130: main pid 32105: LOG: find_primary_node: standby node is 1

2024-02-12 09:41:11.130: main pid 32105: LOG: listen address[0]: localhost

2024-02-12 09:41:11.130: main pid 32105: LOG: Setting up socket for ::1:9898

2024-02-12 09:41:11.130: main pid 32105: LOG: Setting up socket for 127.0.0.1:9898

2024-02-12 09:41:11.131: health_check pid 32142: LOG: process started

2024-02-12 09:41:11.131: sr_check_worker pid 32141: LOG: process started

2024-02-12 09:41:11.131: pcp_main pid 32140: LOG: PCP process: 32140 started

2024-02-12 09:41:11.131: health_check pid 32143: LOG: process started

2024-02-12 09:41:11.133: main pid 32105: LOG: pgpool-II successfully started. version 4.4.0 (nurikoboshi)

2024-02-12 09:41:11.133: main pid 32105: LOG: node status[0]: 1

2024-02-12 09:41:11.133: main pid 32105: LOG: node status[1]: 2

##
##

iv)Connect database with pgpool default port

##
##

[root@ip-172-31-45-179 etc]# /usr/local/pgsql/bin/psql -p 9999 -d postgres -U postgres

psql (13.0)

Type "help" for help.

postgres=# create table pool_test(id int,name varchar(30));

CREATE TABLE

postgres=# insert into pool_test values (1,'Naveen'),(2,'Raju'),(3,'Ramu');

INSERT 0 3

postgres=# select * from pool_test;

```
 id |  name
----+--------
  1 | Naveen
  2 | Raju
  3 | Ramu
(3 rows)
```

##
##

v)check the pgpool log

##
##

```
[root@ip-172-31-45-179 etc]# /usr/local/bin/pgpool -n &
[1] 36708
[root@ip-172-31-45-179 etc]# 2024-02-12 12:07:37.375: main pid 36708: LOG:
health_check_stats_shared_memory_size: requested size: 12288
2024-02-12 12:07:37.375: main pid 36708: LOG:  memory cache initialized
2024-02-12 12:07:37.375: main pid 36708: DETAIL:  memcache blocks :64
2024-02-12 12:07:37.375: main pid 36708: LOG:  allocating (136981824) bytes of shared
memory segment
2024-02-12 12:07:37.375: main pid 36708: LOG:  allocating shared memory segment of size:
136981824
2024-02-12 12:07:37.464: main pid 36708: LOG:  health_check_stats_shared_memory_size:
requested size: 12288
2024-02-12 12:07:37.464: main pid 36708: LOG:  health_check_stats_shared_memory_size:
requested size: 12288
2024-02-12 12:07:37.464: main pid 36708: LOG:  memory cache initialized
2024-02-12 12:07:37.464: main pid 36708: DETAIL:  memcache blocks :64
2024-02-12 12:07:37.467: main pid 36708: LOG:  pool_discard_oid_maps: discarded
memqcache oid maps
```

2024-02-12 12:07:37.471: main pid 36708: LOG: unix_socket_directories[0]: /tmp/.s.PGSQL.9999

2024-02-12 12:07:37.471: main pid 36708: LOG: listen address[0]: *

2024-02-12 12:07:37.472: main pid 36708: LOG: Setting up socket for 0.0.0.0:9999

2024-02-12 12:07:37.472: main pid 36708: LOG: Setting up socket for :::9999

2024-02-12 12:07:37.476: main pid 36708: LOG: find_primary_node_repeatedly: waiting for finding a primary node

2024-02-12 12:07:37.483: main pid 36708: LOG: find_primary_node: primary node is 0

2024-02-12 12:07:37.483: main pid 36708: LOG: find_primary_node: standby node is 1

2024-02-12 12:07:37.483: main pid 36708: LOG: listen address[0]: *

2024-02-12 12:07:37.483: main pid 36708: LOG: Setting up socket for 0.0.0.0:9898

2024-02-12 12:07:37.483: main pid 36708: LOG: Setting up socket for :::9898

2024-02-12 12:07:37.484: sr_check_worker pid 36744: LOG: process started

2024-02-12 12:07:37.484: health_check pid 36746: LOG: process started

2024-02-12 12:07:37.485: health_check pid 36745: LOG: process started

2024-02-12 12:07:37.485: pcp_main pid 36743: LOG: PCP process: 36743 started

2024-02-12 12:07:37.486: main pid 36708: LOG: pgpool-II successfully started. version 4.4.0 (nurikoboshi)

2024-02-12 12:07:37.486: main pid 36708: LOG: node status[0]: 1

2024-02-12 12:07:37.486: main pid 36708: LOG: node status[1]: 2

2024-02-12 12:07:56.392: psql pid 36738: LOG: statement: create table pool_test(id int,name varchar(30));

2024-02-12 12:07:56.392: psql pid 36738: LOG: DB node id: 0 backend pid: 36749 statement: SELECT version()

2024-02-12 12:07:56.393: psql pid 36738: LOG: pool_reuse_block: blockid: 0

2024-02-12 12:07:56.393: psql pid 36738: CONTEXT: while searching system catalog, When relcache is missed

2024-02-12 12:07:56.394: psql pid 36738: LOG: DB node id: 0 backend pid: 36749 statement: create table pool_test(id int,name varchar(30));

2024-02-12 12:08:09.332: psql pid 36738: LOG: statement: SELECT pg_catalog.quote_ident(c.relname) FROM pg_catalog.pg_class c WHERE c.relkind IN ('r', 'f', 'v', 'p') AND substring(pg_catalog.quote_ident(c.relname),1,3)='poo' AND pg_catalog.pg_table_is_visible(c.oid) AND c.relnamespace <> (SELECT oid FROM pg_catalog.pg_namespace WHERE nspname = 'pg_catalog')

    UNION

SELECT pg_catalog.quote_ident(n.nspname) || '.' FROM pg_catalog.pg_namespace n WHERE substring(pg_catalog.quote_ident(n.nspname) || '.',1,3)='poo' AND (SELECT pg_catalog.count(*) FROM pg_catalog.pg_namespace WHERE substring(pg_catalog.quote_ident(nspname) || '.',1,3) = substring('poo',1,pg_catalog.length(pg_catalog.quote_ident(nspname))+1)) > 1

UNION

SELECT pg_catalog.quote_ident(n.nspname) || '.' || pg_catalog.quote_ident(c.relname) FROM pg_catalog.pg_class c, pg_catalog.pg_namespace n WHERE c.relnamespace = n.oid AND c.relkind IN ('r', 'f', 'v', 'p') AND substring(pg_catalog.quote_ident(n.nspname) || '.' || pg_catalog.quote_ident(c.relname),1,3)='poo' AND substring(pg_catalog.quote_ident(n.nspname) || '.',1,3) = substring('poo',1,pg_catalog.length(pg_catalog.quote_ident(n.nspname))+1) AND (SELECT pg_catalog.count(*) FROM pg_catalog.pg_namespace WHERE substring(pg_catalog.quote_ident(nspname) || '.',1,3) = substring('poo',1,pg_catalog.length(pg_catalog.quote_ident(nspname))+1)) = 1

LIMIT 1000

2024-02-12 12:08:09.332: psql pid 36738: LOG: DB node id: 0 backend pid: 36749 statement: SELECT count(*) FROM pg_catalog.pg_proc AS p, pg_catalog.pg_namespace AS n WHERE p.proname = 'quote_ident' AND n.oid = p.pronamespace AND n.nspname = 'pg_catalog' AND p.provolatile = 'v'

2024-02-12 12:08:09.333: psql pid 36738: LOG: DB node id: 0 backend pid: 36749 statement: SELECT count(*) FROM pg_catalog.pg_proc AS p, pg_catalog.pg_namespace AS n WHERE p.proname = 'substring' AND n.oid = p.pronamespace AND n.nspname ~ '.*' AND p.provolatile = 'v'

2024-02-12 12:08:09.334: psql pid 36738: LOG: DB node id: 0 backend pid: 36749 statement: SELECT count(*) FROM pg_catalog.pg_proc AS p, pg_catalog.pg_namespace AS n WHERE p.proname = 'pg_table_is_visible' AND n.oid = p.pronamespace AND n.nspname = 'pg_catalog' AND p.provolatile = 'v'

2024-02-12 12:08:09.334: psql pid 36738: LOG: DB node id: 0 backend pid: 36749 statement: SELECT count(*) FROM pg_catalog.pg_proc AS p, pg_catalog.pg_namespace AS n WHERE p.proname = 'count' AND n.oid = p.pronamespace AND n.nspname = 'pg_catalog' AND p.provolatile = 'v'

2024-02-12 12:08:09.334: psql pid 36738: LOG: DB node id: 0 backend pid: 36749 statement: SELECT count(*) FROM pg_catalog.pg_proc AS p, pg_catalog.pg_namespace AS n WHERE p.proname = 'length' AND n.oid = p.pronamespace AND n.nspname = 'pg_catalog' AND p.provolatile = 'v'

2024-02-12 12:08:09.335: psql pid 36738: LOG: DB node id: 0 backend pid: 36749 statement: SELECT count(*) FROM pg_class AS c, pg_namespace AS n WHERE c.oid = pg_catalog.to_regclass('"pg_class"') AND c.relnamespace = n.oid AND n.nspname = 'pg_catalog'

2024-02-12 12:08:09.336: psql pid 36738: LOG: DB node id: 0 backend pid: 36749 statement: SELECT count(*) FROM pg_class AS c, pg_namespace AS n WHERE c.oid =

pg_catalog.to_regclass('"pg_namespace"') AND c.relnamespace = n.oid AND n.nspname = 'pg_catalog'

2024-02-12 12:08:09.336: psql pid 36738: LOG: DB node id: 0 backend pid: 36749 statement: SELECT pg_catalog.quote_ident(c.relname) FROM pg_catalog.pg_class c WHERE c.relkind IN ('r', 'f', 'v', 'p') AND substring(pg_catalog.quote_ident(c.relname),1,3)='poo' AND pg_catalog.pg_table_is_visible(c.oid) AND c.relnamespace <> (SELECT oid FROM pg_catalog.pg_namespace WHERE nspname = 'pg_catalog')

UNION

SELECT pg_catalog.quote_ident(n.nspname) || '.' FROM pg_catalog.pg_namespace n WHERE substring(pg_catalog.quote_ident(n.nspname) || '.',1,3)='poo' AND (SELECT pg_catalog.count(*) FROM pg_catalog.pg_namespace WHERE substring(pg_catalog.quote_ident(nspname) || '.',1,3) = substring('poo',1,pg_catalog.length(pg_catalog.quote_ident(nspname))+1)) > 1

UNION

SELECT pg_catalog.quote_ident(n.nspname) || '.' || pg_catalog.quote_ident(c.relname) FROM pg_catalog.pg_class c, pg_catalog.pg_namespace n WHERE c.relnamespace = n.oid AND c.relkind IN ('r', 'f', 'v', 'p') AND substring(pg_catalog.quote_ident(n.nspname) || '.' || pg_catalog.quote_ident(c.relname),1,3)='poo' AND substring(pg_catalog.quote_ident(n.nspname) || '.',1,3) = substring('poo',1,pg_catalog.length(pg_catalog.quote_ident(n.nspname))+1) AND (SELECT pg_catalog.count(*) FROM pg_catalog.pg_namespace WHERE substring(pg_catalog.quote_ident(nspname) || '.',1,3) = substring('poo',1,pg_catalog.length(pg_catalog.quote_ident(nspname))+1)) = 1

LIMIT 1000

2024-02-12 12:08:47.921: psql pid 36738: LOG: statement: insert into pool_test values (1,'Naveen'),(2,'Raju'),(3,'Ramu');

2024-02-12 12:08:47.921: psql pid 36738: LOG: DB node id: 0 backend pid: 36749 statement: insert into pool_test values (1,'Naveen'),(2,'Raju'),(3,'Ramu');

2024-02-12 12:08:58.221: psql pid 36738: LOG: statement: SELECT pg_catalog.quote_ident(c.relname) FROM pg_catalog.pg_class c WHERE c.relkind IN ('r', 'S', 'v', 'm', 'f', 'p') AND substring(pg_catalog.quote_ident(c.relname),1,3)='poo' AND pg_catalog.pg_table_is_visible(c.oid) AND c.relnamespace <> (SELECT oid FROM pg_catalog.pg_namespace WHERE nspname = 'pg_catalog')

UNION

SELECT pg_catalog.quote_ident(n.nspname) || '.' FROM pg_catalog.pg_namespace n WHERE substring(pg_catalog.quote_ident(n.nspname) || '.',1,3)='poo' AND (SELECT pg_catalog.count(*) FROM pg_catalog.pg_namespace WHERE substring(pg_catalog.quote_ident(nspname) || '.',1,3) = substring('poo',1,pg_catalog.length(pg_catalog.quote_ident(nspname))+1)) > 1

UNION

SELECT pg_catalog.quote_ident(n.nspname) || '.' || pg_catalog.quote_ident(c.relname) FROM pg_catalog.pg_class c, pg_catalog.pg_namespace n WHERE c.relnamespace = n.oid AND c.relkind IN ('r', 'S', 'v', 'm', 'f', 'p') AND substring(pg_catalog.quote_ident(n.nspname) || '.' || pg_catalog.quote_ident(c.relname),1,3)='poo' AND substring(pg_catalog.quote_ident(n.nspname) || '.',1,3) = substring('poo',1,pg_catalog.length(pg_catalog.quote_ident(n.nspname))+1) AND (SELECT pg_catalog.count(*) FROM pg_catalog.pg_namespace WHERE substring(pg_catalog.quote_ident(nspname) || '.',1,3) = substring('poo',1,pg_catalog.length(pg_catalog.quote_ident(nspname))+1)) = 1

LIMIT 1000

2024-02-12 12:08:58.222: psql pid 36738: LOG: DB node id: 0 backend pid: 36749 statement: SELECT pg_catalog.quote_ident(c.relname) FROM pg_catalog.pg_class c WHERE c.relkind IN ('r', 'S', 'v', 'm', 'f', 'p') AND substring(pg_catalog.quote_ident(c.relname),1,3)='poo' AND pg_catalog.pg_table_is_visible(c.oid) AND c.relnamespace <> (SELECT oid FROM pg_catalog.pg_namespace WHERE nspname = 'pg_catalog')

UNION

SELECT pg_catalog.quote_ident(n.nspname) || '.' FROM pg_catalog.pg_namespace n WHERE substring(pg_catalog.quote_ident(n.nspname) || '.',1,3)='poo' AND (SELECT pg_catalog.count(*) FROM pg_catalog.pg_namespace WHERE substring(pg_catalog.quote_ident(nspname) || '.',1,3) = substring('poo',1,pg_catalog.length(pg_catalog.quote_ident(nspname))+1)) > 1

UNION

SELECT pg_catalog.quote_ident(n.nspname) || '.' || pg_catalog.quote_ident(c.relname) FROM pg_catalog.pg_class c, pg_catalog.pg_namespace n WHERE c.relnamespace = n.oid AND c.relkind IN ('r', 'S', 'v', 'm', 'f', 'p') AND substring(pg_catalog.quote_ident(n.nspname) || '.' || pg_catalog.quote_ident(c.relname),1,3)='poo' AND substring(pg_catalog.quote_ident(n.nspname) || '.',1,3) = substring('poo',1,pg_catalog.length(pg_catalog.quote_ident(n.nspname))+1) AND (SELECT pg_catalog.count(*) FROM pg_catalog.pg_namespace WHERE substring(pg_catalog.quote_ident(nspname) || '.',1,3) = substring('poo',1,pg_catalog.length(pg_catalog.quote_ident(nspname))+1)) = 1

LIMIT 1000

2024-02-12 12:09:00.401: psql pid 36738: LOG: statement: select * from pool_test;

2024-02-12 12:09:00.401: psql pid 36738: LOG: DB node id: 0 backend pid: 36749 statement: SELECT count(*) FROM pg_class AS c, pg_namespace AS n WHERE c.oid = pg_catalog.to_regclass('"pool_test"') AND c.relnamespace = n.oid AND n.nspname = 'pg_catalog'

2024-02-12 12:09:00.402: psql pid 36738: LOG: DB node id: 0 backend pid: 36749 statement: SELECT count(*) FROM pg_catalog.pg_class AS c, pg_namespace AS n WHERE c.relname = 'pool_test' AND c.relnamespace = n.oid AND n.nspname ~ '^pg_temp_'

2024-02-12 12:09:00.403: psql pid 36738: LOG: DB node id: 0 backend pid: 36749 statement: SELECT count(*) FROM pg_catalog.pg_class AS c WHERE c.oid = pg_catalog.to_regclass('"pool_test"') AND c.relpersistence = 'u'

2024-02-12 12:09:00.403: psql pid 36738: LOG: DB node id: 1 backend pid: 28727 statement: select * from pool_test;

2024-02-12 12:14:25.931: psql pid 36714: LOG: statement: select * from test_table;

2024-02-12 12:14:25.931: psql pid 36714: LOG: DB node id: 0 backend pid: 36813 statement: SELECT count(*) FROM pg_class AS c, pg_namespace AS n WHERE c.oid = pg_catalog.to_regclass('"test_table"') AND c.relnamespace = n.oid AND n.nspname = 'pg_catalog'

2024-02-12 12:14:25.932: psql pid 36714: LOG: DB node id: 0 backend pid: 36813 statement: SELECT count(*) FROM pg_catalog.pg_class AS c, pg_namespace AS n WHERE c.relname = 'test_table' AND c.relnamespace = n.oid AND n.nspname ~ '^pg_temp_'

2024-02-12 12:14:25.933: psql pid 36714: LOG: DB node id: 0 backend pid: 36813 statement: SELECT count(*) FROM pg_catalog.pg_class AS c WHERE c.oid = pg_catalog.to_regclass('"test_table"') AND c.relpersistence = 'u'

2024-02-12 12:14:25.933: psql pid 36714: LOG: DB node id: 1 backend pid: 28774 statement: select * from test_table;

##

vi)To Stop pgpool

[root@ip-172-31-45-179 etc]# /usr/local/bin/pgpool -m fast stop