

Installation & Configuration PGPool 2 and PostgreSQL-16 with Streaming Replication in Ubuntu 24.04 LTS

1.3k views

asked in PostgreSQL by superadmin

Installation & Configuration PGPool 2 and PostgreSQL-16 with Streaming Replication in Ubuntu 24.04 LTS

postgresql

streaming-replication

replication

ha postgresql

1 Answer

answered by superadmin

inchirags@gmail.com Chirag's PostgreSQL DBA Tutorial <https://www.chirags.in>

Installation & Configuration PGPool 2 and PostgreSQL-16 with Streaming Replication in Ubuntu 24.04 LTS

Please, Subscribe and like for more videos

To set up PostgreSQL-16 with replication between two servers and PGPool 2 on Ubuntu 24.04 LTS, follow these steps:

Pre-requisites:

Two PostgreSQL servers:

```
Master Server1: 192.168.224.134
Replica Server2: 192.168.224.135
PGPool server: 192.168.224.147
```

ALL PostgreSQL servers should have Ubuntu 24.04 LTS installed.

Ensure both PostgreSQL servers can communicate with each other and also communicate with PGPool server.

On Master and Slave servers, PostgreSQL 16 must have installed.

Or you can install with below command:

```
sudo apt update
sudo apt upgrade -y

sudo apt install postgresql postgresql-contrib
```

Step1: Configurations on master server (192.168.224.134)

1. On master server, configure the IP address(es) listen to for connections from clients in postgresql.conf by removing # in front of listen_address and give *. Which means listen connections from all.

```

sudo nano /etc/postgresql/16/main/postgresql.conf
listen_addresses = '*'
wal_level = replica
max_wal_senders = 10
wal_keep_size = 64MB
hot_standby = on
password_encryption = 'md5'

```

2. Now, connect to PostgreSQL on master server and create replica login.

```

sudo -u postgres psql
SHOW password_encryption;
ALTER SYSTEM SET password_encryption = 'md5';
\q

```

```
sudo systemctl restart postgresql
```

```
CREATE USER replicator WITH REPLICATION ENCRYPTED PASSWORD 'admin@123';
```

```
ALTER USER postgres WITH PASSWORD 'admin@123';
```

3. Enter the following entry pg_hba.conf file which is located in /etc/postgresql/16/main on Ubuntu(debian systems).

```
sudo nano /etc/postgresql/16/main/pg_hba.conf
```

```
host replication replicator 192.168.224.135/24 md5
```

How to setup streaming replication in PostgreSQL step by step on Ubuntu

4. Now, restart the PostgreSQL on Master server by using below command.

```
sudo systemctl restart postgresql
```

Step3: Configurations on slave(standby) server (192.168.224.135)

1. We have to stop PostgreSQL on Slave server by using following command.

Install PostgreSQL using below command:

```
sudo apt update
```

```
sudo apt upgrade -y
```

```
sudo apt install postgresql postgresql-contrib
```

```
sudo systemctl stop postgresql
```

2. Now, switch to postgres user and take backup of main(data) directory.

```
su - postgres
```

```
cp -R /var/lib/postgresql/16/main/ /var/lib/postgresql/16/main_old/
```

3. Now, remove the contents of main(data) directory on slave server.

```
rm -rf /var/lib/postgresql/16/main/
```

4. Now, use basebackup to take the base backup with the right ownership with postgres(or any user with right permissions).

```
pg_basebackup -h 192.168.224.134 -D /var/lib/postgresql/16/main/ -U replicator -P -v -R -X stream -C -S slaveslot1
```

Then provide the password for user replicator created in master server.

```
pg_basebackup: initiating base backup, waiting for checkpoint to complete
```

```
pg_basebackup: checkpoint completed
```

```
pg_basebackup: write-ahead log start point: 0/11000028 on timeline 1
```

```
pg_basebackup: starting background WAL receiver
```

```
pg_basebackup: created replication slot "slaveslot1"
```

```
0/74620 kB (0%), 0/1 tablespace (.../postgresql/16/main/backup_13369/74620 kB (17%), 0/1 tablespace (...stgresql/16/main/base/491532739/74620 kB (43%), 0/1 tablespace (...stgresql/16/main/base/409651750/74620 kB (69%), 0/1 tablespace (...stgresql/16/main/base/655469912/74620 kB (93%), 0/1 tablespace (...stgresql/16/main/base/134674630/74630 kB (100%), 0/1 tablespace (...gresql/16/main/global/pg_74630/74630 kB (100%), 1/1 tablespace
```

```
pg_basebackup: write-ahead log end point: 0/11000100
```

```
pg_basebackup: waiting for background process to finish streaming ...
```

```
pg_basebackup: syncing data to disk ...
```

```
pg_basebackup: base backup completed
```

5. Notice that standby.signal is created and the connection settings are appended to postgresql.auto.conf.

```
ls -ltrh /var/lib/postgresql/16/main/
```

6. A replication slave will run in “Hot Standby” mode if the hot_standby parameter is set to on (the default value) in postgresql.conf and there is a standby.signal file present in the data directory.

7. Now connect the master server, you should be able to see the replication slot called slotslave1 when you open the pg_replication_slots view as follows.

```
sudo -u postgres psql
```

```
SELECT * FROM pg_replication_slots;
```

Step 3: Test replication setup

1. Now start PostgreSQL on slave(standby) server.

```
systemctl start postgresql
```

2. Now, try to create object or database in slave(standby) server. It throws error, because slave(standby) is read-only server.

```
create database slave1;
```

3. WE can check the status on standby using below command.

```
SELECT * FROM pg_stat_wal_receiver;
```

4. Now, verify the replication type synchronous or asynchronous using below command on master database server.

```
SELECT * FROM pg_stat_replication;
```

5. Lets create a database in master server and verify its going to replicate to slave or not.

```
create database stream;
```

6. Now, connect to slave and verify the database copied or not.

```
select datname from pg_database;
```

7. If you want to enable synchronous, the run the below command on master database server and reload postgresql service.

```
sudo -u postgres psql
```

```
ALTER SYSTEM SET synchronous_standby_names TO '*';
```

```
\q
```

```
systemctl reload postgresql
```

Thats all. We have successfully setup streaming replication in PostgreSQL step by step on Ubuntu.

Test Replication.

On master:

```
sudo -u postgres psql -c "select * from pg_stat_replication;"
```

On replica (streaming replication in my case):

```
sudo -u postgres psql -c "select * from pg_stat_wal_receiver;"
```

Step 4: Install and Configure PGPool 2 on Server3 - 192.168.224.147

Pre-requisites- in Master and Slave Server:

```
sudo nano /etc/postgresql/16/main/pg_hba.conf
```

```
host all all 192.168.224.147/24 md5
```

in Slave:

```
sudo nano /etc/postgresql/16/main/postgresql.conf
```

```
listen_addresses = '*'
```

Save above file then restart PostgreSQL:

```
systemctl restart postgresql
```

Install Dependencies:

```
sudo apt install pgpool2 -y
```

```
apt-get -y install postgresql-16-pgpool2
```

Configure PGPool Edit the configuration file for PGPool:

```
sudo nano /etc/pgpool2/pgpool.conf
```

Set the following parameters:

```
listen_addresses = '*'
```

```
port = 9999
```

Backend connections:

backend_hostname0 = '192.168.224.134'

backend_port0 = 5432

backend_weight0 = 1

backend_data_directory0 = '/var/lib/postgresql/16/main'

backend_flag0 = 'ALLOW_TO_FAILOVER'

backend_application_name0 = 'server0'

backend_hostname1 = '192.168.224.135'

backend_port1 = 5432

backend_weight1 = 1

backend_data_directory1 = '/var/lib/postgresql/16/main'

backend_flag1 = 'ALLOW_TO_FAILOVER'

backend_application_name1 = 'server1'

enable_pool_hba = on

auth_type = md5

pool_passwd = '/etc/pgpool2/pool_passwd'

log_destination = 'stderr'

logging_collector = on

log_directory = '/var/log/pgpool'

log_filename = 'pgpool.log'

load_balance_mode = on

sr_check_period = 10

sr_check_user = 'postgres'

sr_check_password = 'admin@123'

sr_check_database = 'postgres'

health_check_period = 10

health_check_timeout = 20

health_check_user = 'postgres'

health_check_password = 'admin@123'

health_check_database = 'postgres'

health_check_max_retries = 3

```

health_check_retry_delay = 5

failover_command = '/etc/pgpool2/failover.sh %d %P %H %M %R'

failback_command = '/etc/pgpool2/failback.sh %d %P %H %M %R'

failover_on_backend_error = on

#recovery_1st_stage_command = 'recovery_1st_stage.sh'

hostname0 = "

Pool settings:

num_init_children = 32

max_pool = 4

Replication configuration:

load_balance_mode = on

//replication_mode = on

//Configure pg_hba.conf

sudo nano /etc/pgpool2/pcp.conf

Add the authentication line:

replicator:admin@123

sudo vi /etc/pgpool2/pool_hba.conf

host all postgres 192.168.224.134/32 md5

host all postgres 192.168.224.135/32 md5

Create failover.sh and failback.sh file:

sudo vi /etc/pgpool2/failover.sh

#!/bin/bash

FAILED_NODE_ID=$1

NEW_PRIMARY_HOST=$3

# If the master fails (node 0), promote the standby to primary

if [ $FAILED_NODE_ID -eq 0 ]; then

    echo "Failover detected. Promoting standby (node 1) to primary..."

    psql -h $NEW_PRIMARY_HOST -U postgres -c "SELECT pg_promote();"

fi

exit 0

```

```
sudo vi /etc/pgpool2/failback.sh
```

```
#!/bin/bash
```

```
FAILED_NODE_ID=$1
```

```
echo "Reattaching node $FAILED_NODE_ID to pgpool-II after recovery."
```

```
exit 0
```

```
#/etc/pgpool2/pool_passwd
```

```
sudo pg_md5 -m -u postgres 'admin@123'
```

```
sudo chown postgres:postgres /etc/pgpool2/failover.sh
```

```
sudo chown postgres:postgres /etc/pgpool2/failback.sh
```

```
sudo chmod +x /etc/pgpool2/failover.sh
```

```
sudo chmod +x /etc/pgpool2/failback.sh
```

```
sudo mkdir -p /var/log/pgpool
```

```
sudo chown postgres:postgres /var/log/pgpool
```

Start PGPool:

```
sudo systemctl start pgpool2
```

```
sudo systemctl enable pgpool2
```

Open PostgreSQL in Master Server:

```
sudo tail -f /var/log/postgresql/postgresql-16-main.log
```

Restart PGPool 2 Server:

```
sudo systemctl restart pgpool2
```

Open PGPool in PGPool Server:

```
sudo tail -f /var/log/pgpool/pgpool.log
```

Connect from any client:

```
psql -h 192.168.224.147 -p 9999 -U postgres
```

```
SHOW pool_nodes;
```

For any doubts and query, please write on YouTube video comments section.

Note : Flow the Process shown in video.

Please, Subscribe and like for more videos:

<https://youtube.com/@chiragstutorial>

Don't forget to, Follow, Like, Share &, Comment

Tutorial Link :

Thanks & Regards,

Chitt Ranjan Mahto "Chirag"

Note: All scripts used in this demo will be available in our website.

Link will be available in description.

Categories

All categories

Technology (10)

Asp.net C# (1)

Asp.Net MVC (1)

Python (3)

Laravel (28)

MongoDB (10)

Oracle (1)

MySQL (6)

MS SQL Server (2)

PostgreSQL (20)

Asp.Net Core MVC (2)

Windows (1)

Windows Server 2025 (6)

MariaDB (0)

Most popular tags

laravel postgresql laravel-10 replication ha postgresql mongodb laravel-11 mongodb database mongodb tutorial ubuntu 24.04 lts streaming-replication

mysql database laravel postgresql backup laravel login register logout database mysql php laravel 11 - login with otp valid for 10 minutes. user and admin registration user and admin login multiauth technology asp.net asp.net c# mysql master slave replication centos linux laravel sql server schedule backup autobackup postgresql django python haproxy load balancer install self sign ssl laravel 11 guards zabbix 7 how to install graylog on ubuntu 24.04 lts [step-by-step asp.net core mvc .net mvc network upload c# ssl integration sql server on ubuntu 22.04 lts mssql server ms sql server sql server user access in postgres mysql password change cent os linux configure replica laravel 11 socialite login with google account google login kubernetes (k8s) install nginx load balancer install install and configure .net 8.0 in ubuntu 24.04 lts php in iis php with iis php tutorial chirags php tutorial chirags php tutorial chirags tutorial laravel 11 guards mongodb sharding metabase business analytics metabase postgresql 16 to postgresql 17 postgresql migration letsencrypt mongodb crud rocky linux laravel custom captcha laravel 11 captcha laravel captcha mongo dll php.ini debian 12 nginx apache nextcloud gitea in ubuntu git gitea npm error node js mysql ndb cluster mysql cluster ssl oracle login register logout in python debian windows shell batch file bat file time stamp date time shopping cart in laravel centos rhel swap memory rhel 5.5

Related questions

Installation, Configuration & Tuning of PostgreSQL 17 and pgAdmin4 in Ubuntu 24.04 LTS

Streaming Replication of PostgreSQL 17 in Ubuntu 24.04 LTS

PostgreSQL 16 - Promote Slave to Master and Master to Slave in Ubuntu 24.04 LTS step by step process

How to setup streaming replication in PostgreSQL 14 step by step on Ubuntu 22.04 LTS

PostgreSQL Tutorial - How to setup streaming replication in PostgreSQL step by step on Ubuntu 24.04 LTS