

PostgreSQL Replication setup with pgBackRest:

OS: ubuntu

PostgreSQL version:15.7

Primary server IP: 172.31.80.61

Standby server IP: 172.31.87.47

On primary server:

Step1) Create Replication user:

CREATE USER rep_user REPLICATION PASSWORD 'rep_user@123';

```
postgres=# CREATE USER rep_user REPLICATION PASSWORD 'rep_user@123';
CREATE ROLE
postgres=# \q
postgres@ip-172-31-80-61:~$
```

Step2) Allow the standby server IP to connect the primary server:

#Edit \$PGDATA/pg_hba.conf file

vim /db_data/data/pg_hba.conf

host all all 172.31.87.47/32 scram-sha-256

host replication rep_user 172.31.87.47/32 scram-sha-256

save&exit

```
root@ip-172-31-80-61:/home/ubuntu# vim /db_data/data/pg_hba.conf
root@ip-172-31-80-61:/home/ubuntu# cat /db_data/data/pg_hba.conf |grep "172.31.87.47"
host all all 172.31.87.47/32 scram-sha-256
host replication rep_user 172.31.87.47/32 scram-sha-256
root@ip-172-31-80-61:/home/ubuntu#
```

#Restart the postgresQL services

systemctl restart postgresql-15.service

systemctl status postgresql-15.service

Step3) Perform the full base back with pgbackrest:

#Pefrom the backup

sudo -u postgres pgbackrest --stanza=master --log-level-console=info --type=full backup

```
root@ip-172-31-80-61:/home/ubuntu# sudo -u postgres pgbackrest --stanza=master --log-level-console=info --type=full backup
2024-08-12 10:31:32.793 P00 INFO: backup command begin 2.50: --compress-level=6 --exec-id=23702-a8ec4fbb --log-level-console=info --pgl-path=/db_data/data
--pgl-socket-path=/tmp --repol-block --repol-bundle --repol-path=/db_backup/base_backup --repol-retention-diff=1 --repol-retention-full=2 --stanza=master --s
tart-fast --type=full
2024-08-12 10:31:33.499 P00 INFO: execute non-exclusive backup start: backup begins after the requested immediate checkpoint completes
2024-08-12 10:31:34.001 P00 INFO: backup start archive = 00000001000000000000000014, lsn = 0/14000028
2024-08-12 10:31:34.001 P00 INFO: check archive for prior segment 00000001000000000000000013
2024-08-12 10:31:36.587 P00 INFO: execute non-exclusive backup stop and wait for all WAL segments to archive
2024-08-12 10:31:36.787 P00 INFO: backup stop archive = 00000001000000000000000014, lsn = 0/14000100
2024-08-12 10:31:36.798 P00 INFO: check archive for segment(s) 00000001000000000000000014:000000010000000000000000
2024-08-12 10:31:36.815 P00 INFO: new backup label = 20240812-103133F
2024-08-12 10:31:36.866 P00 INFO: full backup size = 21.8MB, file total = 993
2024-08-12 10:31:36.867 P00 INFO: backup command end: completed successfully (4076ms)
2024-08-12 10:31:36.867 P00 INFO: expire command begin 2.50: --exec-id=23702-a8ec4fbb --log-level-console=info --repol-path=/db_backup/base_backup --repol-
retention-diff=1 --repol-retention-full=2 --stanza=master
2024-08-12 10:31:36.868 P00 INFO: repol: expire full backup 20240810-051717F
2024-08-12 10:31:36.869 P00 INFO: repol: expire diff backup set 20240810-051827F_20240810-052906D, 20240810-051827F_20240810-053231I
2024-08-12 10:31:36.878 P00 INFO: repol: remove expired backup 20240810-051827F_20240810-053231I
2024-08-12 10:31:36.878 P00 INFO: repol: remove expired backup 20240810-051827F_20240810-052906D
2024-08-12 10:31:36.879 P00 INFO: repol: remove expired backup 20240810-051717F
2024-08-12 10:31:36.882 P00 INFO: repol: 15-1 remove archive, start = 00000001000000000000000003, stop = 000000010000000000000004
2024-08-12 10:31:36.882 P00 INFO: expire command end: completed successfully (15ms)
```

#Check the backup info

sudo -u postgres pgbackrest --stanza=master --log-level-console=info

```

root@ip-172-31-80-61:/home/ubuntu# sudo -u postgres pgbackrest --stanza=master --log-level-console=info info
stanza: master
status: ok
cipher: none

db (current)
wal archive min/max (15): 000000010000000000000005/000000010000000000000014

full backup: 20240810-051827F
timestamp start/stop: 2024-08-10 05:18:27+00 / 2024-08-10 05:18:30+00
wal start/stop: 000000010000000000000005 / 000000010000000000000005
database size: 21.4MB, database backup size: 21.4MB
repol: backup size: 2.8MB

full backup: 20240812-103133F
timestamp start/stop: 2024-08-12 10:31:33+00 / 2024-08-12 10:31:36+00
wal start/stop: 000000010000000000000014 / 000000010000000000000014
database size: 21.8MB, database backup size: 21.8MB
repol: backup size: 2.8MB

```

Step4) Check the data and tablespace directories:

psql

show data_directory;

\db+

```

postgres@ip-172-31-80-61:~$ psql
psql (15.7)
Type "help" for help.

postgres=# show data_directory;
 data_directory
-----
 /db_data/data
(1 row)

postgres=# \db+
               List of tablespaces
  Name   | Owner  | Location          | Access privileges | Options | Size  | Description
-----+-----+-----+-----+-----+-----+-----
 pg_default | postgres |                  |                   |         | 21 MB |
 pg_global | postgres |                  |                   |         | 531 kB |
 tbl_space1 | postgres | /db_data1/tbl_space1 |                   |         | 116 kB |
 tbl_space2 | postgres | /db_data2/tbl_space2 |                   |         | 60 kB |
(4 rows)

postgres=#

```

On standby server:

Step1) Check and install PostgreSQL and pgbackrest:

NOTE: The data_directory must be created and initialized (even though it will be overwritten on restore) in order to create the PostgreSQL configuration files.

```

root@ip-172-31-87-47:/home/ubuntu# su - postgres
postgres@ip-172-31-87-47:~$ initdb -D /db_data/data/
The files belonging to this database system will be owned by user "postgres".
This user must also own the server process.

The database cluster will be initialized with locale "C.UTF-8".
The default database encoding has accordingly been set to "UTF8".
The default text search configuration will be set to "english".

Data page checksums are disabled.

fixing permissions on existing directory /db_data/data ... ok
creating subdirectories ... ok
selecting dynamic shared memory implementation ... posix
selecting default max connections ... 100
selecting default shared buffers ... 128MB
selecting default time zone ... Etc/UTC
creating configuration files ... ok
running bootstrap script ... ok
performing post-bootstrap initialization ... ok
syncing data to disk ... ok

initdb: warning: enabling "trust" authentication for local connections
initdb: hint: You can change this by editing pg_hba.conf or using the option -A, or --auth-local and --auth-host, the next time you run initdb.

Success. You can now start the database server using:

    pg_ctl -D /db_data/data/ -l logfile start

postgres@ip-172-31-87-47:~$

```

#Install and configure pgbackrest

https://www.linkedin.com/posts/naveen-gomangi-aa85bb184_pgbackrest-activity-7222671024805363713-YJw0?utm_source=share&utm_medium=member_desktop

Step2) Check the tablespace directories, create if not exist:

mkdir -p /db_data1/tbl_space1 /db_data2/tbl_space2

chown -R postgres:postgres /db_data1/ /db_data2/

Step3) Setup Password less SSH between primary and standby server

Step4) Configure pgbackrest:

#Edit pgbackrest.conf file

vim /etc/pgbackrest.conf

[global]

repo1-block=y

repo1-bundle=y

repo1-host=172.31.80.61

repo1-host-user=postgres

repo1-path=/db_backup/base_backup

repo1-retention-diff=1

repo1-retention-full=2

start-fast=y

compress-level=6

[global:archive-push]

compress-level=6

[master]

pg1-path=/db_data/data

pg1-socket-path=/tmp

recovery-option=primary_conninfo=host=172.31.80.61 port=5432 user=rep_user

password=rep_user@123

save&exit

```
root@ip-172-31-87-47:/home/ubuntu# vim /etc/pgbackrest.conf
root@ip-172-31-87-47:/home/ubuntu# cat /etc/pgbackrest.conf
[global]
repo1-block=y
repo1-bundle=y
repo1-host=172.31.80.61
repo1-host-user=postgres
repo1-path=/db_backup/base_backup
repo1-retention-diff=1
repo1-retention-full=2
start-fast=y
compress-level=6

[global:archive-push]
compress-level=6

[master]
pg1-path=/db_data/data
pg1-socket-path=/tmp
recovery-option=primary_conninfo=host=172.31.80.61 port=5432 user=rep_user password=rep_user@123
root@ip-172-31-87-47:/home/ubuntu#
```

Step5) Check the available backups on primary server:

sudo -u postgres pgbackrest --stanza=master info

```

root@ip-172-31-87-47:/home/ubuntu# sudo -u postgres pgbackrest --stanza=master info
stanza: master
  status: ok
  cipher: none

  db (current)
    wal archive min/max (15): 000000010000000000000005/000000010000000000000014

    full backup: 20240810-051827F
      timestamp start/stop: 2024-08-10 05:18:27+00 / 2024-08-10 05:18:30+00
      wal start/stop: 000000010000000000000005 / 000000010000000000000005
      database size: 21.4MB, database backup size: 21.4MB
      repl: backup size: 2.8MB

    full backup: 20240812-103133F
      timestamp start/stop: 2024-08-12 10:31:33+00 / 2024-08-12 10:31:36+00
      wal start/stop: 000000010000000000000014 / 000000010000000000000014
      database size: 21.8MB, database backup size: 21.8MB
      repl: backup size: 2.8MB
root@ip-172-31-87-47:/home/ubuntu#

```

Step6) Restore the backup:

sudo -u postgres pgbackrest --stanza=master --delta --type=standby --log-level-console=info restore

```

root@ip-172-31-87-47:/home/ubuntu# sudo -u postgres pgbackrest --stanza=master --delta --type=standby --log-level-console=info restore
2024-08-12 10:41:36.497 P00 INFO: restore command begin 2.50: --delta --exec-id=3432-8a9b1a2b --log-level-console=info --pgl-path=/db_data/data --recovery-option="primary_conninfo=host=172.31.80.61 port=5432 user=rep_user password=rep_user@123" --repl-host=172.31.80.61 --repl-host-user=postgres --repl-path=/db_backup/base backup --stanza=master --type=standby
2024-08-12 10:41:37.103 P00 INFO: repl: restore backup set 20240812-103133F, recovery will start at 2024-08-12 10:31:33
2024-08-12 10:41:37.105 P00 INFO: remove invalid files/links/paths from '/db_data/data'
2024-08-12 10:41:38.305 P00 INFO: write updated /db_data/data/postgresql.auto.conf
2024-08-12 10:41:38.311 P00 INFO: restore global/pg_control (performed last to ensure aborted restores cannot be started)
2024-08-12 10:41:38.313 P00 INFO: restore size = 21.8MB, file total = 992
2024-08-12 10:41:38.314 P00 INFO: restore command end: completed successfully (1820ms)

```

Step7) Enable hot_standby parameter in a \$PGDATA/postgresql.conf file:

NOTE: The hot_standby setting must be enabled before starting PostgreSQL to allow read-only connections on pg-standby. Otherwise, connection attempts will be refused.

vim /db_data/data/postgresql.conf

hot_standby = on

save&exit

```

root@ip-172-31-87-47:/home/ubuntu# vim /db_data/data/postgresql.conf
root@ip-172-31-87-47:/home/ubuntu# cat /db_data/data/postgresql.conf |grep "hot_standby"
hot_standby = on                # "off" disallows queries during recovery
#hot_standby_feedback = off     # send info from standby to prevent
root@ip-172-31-87-47:/home/ubuntu#

```

Step8) Start and check the status of PostgreSQL:

sudo systemctl start postgresql-15.service

sudo systemctl status postgresql-15.service

```

root@ip-172-31-87-47:/db_data/data# sudo systemctl start postgresql-15.service
root@ip-172-31-87-47:/db_data/data# sudo systemctl status postgresql-15.service
● postgresql-15.service - PostgreSQL database server
   Loaded: loaded (/etc/systemd/system/postgresql-15.service; enabled; preset: enabled)
   Active: active (running) since Mon 2024-08-12 11:01:10 UTC; 6s ago
     Docs: https://www.postgresql.org/docs/
   Process: 3600 ExecStart=/usr/local/pgsql/bin/pg_ctl start -D /db_data/data (code=exited, status=0/SUCCESS)
    Main PID: 3603 (postgres)
      Tasks: 6 (limit: 1130)
     Memory: 62.8M (peak: 62.8M)
        CPU: 958ms
   CGroup: /system.slice/postgresql-15.service
           └─3603 /usr/local/pgsql/bin/postgres -D /db_data/data
             └─3604 "postgres: logger "
               └─3605 "postgres: checkpointer "
                 └─3606 "postgres: background writer "
                   └─3607 "postgres: startup recovering 000000010000000000000016"
                     └─3623 "postgres: walreceiver streaming 0/160000D8"

Aug 12 11:01:08 ip-172-31-87-47 systemd[1]: Starting postgresql-15.service - PostgreSQL database server...
Aug 12 11:01:08 ip-172-31-87-47 pg_ctl[3600]: waiting for server to start....
Aug 12 11:01:08 ip-172-31-87-47 pg_ctl[3603]: 2024-08-12 11:01:08.372 UTC [3603] LOG: redirecting log output to logging collector process
Aug 12 11:01:08 ip-172-31-87-47 pg_ctl[3603]: 2024-08-12 11:01:08.372 UTC [3603] HINT: Future log output will appear in directory "log".
Aug 12 11:01:10 ip-172-31-87-47 pg_ctl[3600]: .. done
Aug 12 11:01:10 ip-172-31-87-47 pg_ctl[3600]: server started
Aug 12 11:01:10 ip-172-31-87-47 systemd[1]: Started postgresql-15.service - PostgreSQL database server.

```

#Check the logs

tail -f \$PGDATA/log/postgresql-Mon.log

```

root@ip-172-31-87-47:/db_data/data# su - postgres
postgres@ip-172-31-87-47:~$ tail -f $PGDATA/log/postgresql-Mon.log
2024-08-12 11:01:08.385 UTC [3607] LOG: database system was interrupted; last known up at 2024-08-12 10:31:33 UTC
2024-08-12 11:01:09.362 UTC [3607] LOG: entering standby mode
2024-08-12 11:01:09.362 UTC [3607] LOG: starting backup recovery with redo LSN 0/14000028, checkpoint LSN 0/14000060, on timeline ID 1
2024-08-12 11:01:09.852 UTC [3607] LOG: restored log file "000000010000000000000000000014" from archive
2024-08-12 11:01:09.888 UTC [3607] LOG: redo starts at 0/14000028
2024-08-12 11:01:10.383 UTC [3607] LOG: restored log file "000000010000000000000000000015" from archive
2024-08-12 11:01:10.858 UTC [3607] LOG: completed backup recovery with redo LSN 0/14000028 and end LSN 0/14000100
2024-08-12 11:01:10.858 UTC [3607] LOG: consistent recovery state reached at 0/14000100
2024-08-12 11:01:10.858 UTC [3603] LOG: database system is ready to accept read-only connections
2024-08-12 11:01:11.319 UTC [3623] LOG: started streaming WAL from primary at 0/16000000 on timeline 1

```

Check the replication status:

On primary server:

#Check the replication status

SELECT * FROM pg_stat_replication;

```

postgres@ip-172-31-80-61:~$ psql
psql (15.7)
Type "help" for help.

postgres=# \x
Expanded display is on.
postgres=# SELECT * FROM pg_stat_replication;
-[ RECORD 1 ]-----
 pid                | 24545
 usesysid           | 16442
 username           | rep_user
 application_name    | walreceiver
 client_addr         | 172.31.87.47
 client_hostname     |
 client_port        | 40886
 backend_start      | 2024-08-12 11:01:11.307973+00
 backend_xmin        |
 state              | streaming
 sent_lsn            | 0/160001C0
 write_lsn           | 0/160001C0
 flush_lsn           | 0/160001C0
 replay_lsn          | 0/160001C0
 write_lag           |
 flush_lag           |
 replay_lag          |
 sync_priority       | 0
 sync_state          | async
 reply_time          | 2024-08-12 11:04:44.286563+00

```

#Create test table and insert data

```

CREATE TABLE testproducts (
    testproduct_id SERIAL NOT NULL PRIMARY KEY,
    product_name VARCHAR(255),
    category_id INT
);

```

INSERT INTO testproducts (product_name, category_id)

VALUES

```

('Johns Fruit Cake', 3),
('Marys Healthy Mix', 9),
('Peters Scary Stuff', 10),
('Jims Secret Recipe', 11);

```

```

root@ip-172-31-80-61:~# su - postgres
postgres@ip-172-31-80-61:~$ psql
psql (15.7)
Type "help" for help.

postgres=# CREATE TABLE testproducts (
    testproduct_id SERIAL NOT NULL PRIMARY KEY,
    product_name VARCHAR(255),
    category_id INT
);
CREATE TABLE
postgres=# INSERT INTO testproducts (product_name, category_id)
VALUES
    ('Johns Fruit Cake', 3),
    ('Marys Healthy Mix', 9),
    ('Peters Scary Stuff', 10),
    ('Jims Secret Recipe', 11);
INSERT 0 4
postgres=# SELECT * FROM testproducts;
 testproduct_id | product_name | category_id
-----+-----+-----
          1 | Johns Fruit Cake |          3
          2 | Marys Healthy Mix |          9
          3 | Peters Scary Stuff |         10
          4 | Jims Secret Recipe |         11
(4 rows)

postgres=#

```

SELECT * FROM testproducts;

On standby server:

#Check the wal receiver status

```

postgres@ip-172-31-87-47:~$ psql
psql (15.7)
Type "help" for help.

postgres=# \x
Expanded display is on.
postgres=# SELECT * FROM pg_stat_wal_receiver;
-[ RECORD 1 ]-----+-----
 pid                | 3623
 status              | streaming
 receive_start_lsn   | 0/16000000
 receive_start_tli   | 1
 written_lsn         | 0/160001C0
 flushed_lsn         | 0/160001C0
 received_tli        | 1
 last_msg_send_time  | 2024-08-12 11:05:44.4257+00
 last_msg_receipt_time | 2024-08-12 11:05:44.42587+00
 latest_end_lsn      | 0/160001C0
 latest_end_time     | 2024-08-12 11:01:43.872157+00
 slot_name           |
 sender_host         | 172.31.80.61
 sender_port         | 5432
 conninfo             | user=rep_user password=***** channel_binding=disable dbname=replication host=172.31.80.61 port=5432 fallback_application_name=walr
 receiver            | sslmode=disable sslcompression=0 sslsnl=1 ssl_min_protocol_version=TLSv1.2 gsencmode=disable krbsrvname=postgres target_session_attrs=any
postgres=#

```

#Check the table data

SELECT * FROM pg_stat_wal_receiver;

```

root@ip-172-31-87-47:~# su - postgres
postgres@ip-172-31-87-47:~$ psql
psql (15.7)
Type "help" for help.

postgres=# SELECT * FROM testproducts;
 testproduct_id | product_name | category_id
-----+-----+-----
          1 | Johns Fruit Cake |          3
          2 | Marys Healthy Mix |          9
          3 | Peters Scary Stuff |         10
          4 | Jims Secret Recipe |         11
(4 rows)

postgres=#

```