

Streaming Replication di PostgreSQL 18

Dr. Bambang Purnomosidi D. P. (<https://linktre.ee/bpdp>)

Fakultas Teknologi Informasi

Universitas Teknologi Digital Indonesia

Materi ini membahas tentang cara mengkonfigurasi *streaming replication* di PostgreSQL 18. Secara prinsip, sebenarnya sama dengan versi-versi PostgreSQL sebelumnya, hanya saja letak dari direktori data bukan di `/var/lib/postgresql/data` lagi tapi berubah di `/var/lib/postgresql/18/docker`. Materi ini ditulis berdasarkan materi pada <https://medium.com/@eremeykin/how-to-setup-single-primary-postgresql-replication-with-docker-compose-98c48f233bbf> dengan perubahan signifikan.

0. Prasyarat

1. Instalasi Docker telah dilaksanakan. Lihat ke <https://docs.docker.com/engine/install/> jika belum.
2. Instalasi Docker Compose telah dilaksanakan. Lihat ke <https://docs.docker.com/compose/install/> jika belum.
3. *Daemon dockerd* telah diaktifkan.
4. Petunjuk di materi ini menggunakan Linux, jika menggunakan sistem operasi lainnya, sila membuat penyesuaian seperlunya.

1. Struktur Direktori dan File

File yang diperlukan adalah sebagai berikut:

```
streaming-replication/  
├── 00_init.sql  
├── docker-compose.yaml  
└── env.sh  
  
1 directory, 3 files
```

Berikut adalah penjelasan dari masing-masing file tersebut:

1. `00_init.sql`: file yang berisi perintah-perintah SQL yang akan dijalankan saat primary server diinisialisasi dan dijalankan. Fungsi utamanya adalah untuk membuat user yang akan melakukan replikasi ke primary server serta slot replikasinya.
2. `docker-compose.yaml`: file konfigurasi yang akan dijalankan oleh **docker-compose**.
3. `env.sh`: file yang berisi berbagai definisi alias di shell (Bash), hanya untuk memudahkan shortcut penulisan perintah, opsional.

`00_init.sql`

```
CREATE USER replicator WITH REPLICATION ENCRYPTED PASSWORD 'replicator_password';  
SELECT pg_create_physical_replication_slot('replication_slot');
```

`docker-compose.yaml`

```

x-postgres-common:
  &postgres-common
  image: postgres:18-alpine
  user: postgres
  restart: always
  healthcheck:
    test: 'pg_isready -U zuser --dbname=zdb'
    interval: 10s
    timeout: 5s
    retries: 5

services:
  postgres_primary:
    <<: *postgres-common
    ports:
      - 5432:5432
    environment:
      POSTGRES_USER: zuser
      POSTGRES_DB: zdb
      POSTGRES_PASSWORD: zpass
      POSTGRES_HOST_AUTH_METHOD: "scram-sha-256\nhost replication replicator
0.0.0.0/0 md5"
      POSTGRES_INITDB_ARGS: "--auth-host=scram-sha-256"
    command: |
      postgres
      -c wal_level=replica
      -c hot_standby=on
      -c max_wal_senders=10
      -c max_replication_slots=10
      -c hot_standby_feedback=on
    volumes:
      - ./00_init.sql:/docker-entrypoint-initdb.d/00_init.sql

  postgres_replica:
    <<: *postgres-common
    ports:
      - 5433:5432
    environment:
      PGUSER: replicator
      PGPASSWORD: replicator_password
      PGDATA: /var/lib/postgresql/18/docker
    command: |
      bash -c "
          until pg_basebackup --pgdata=/var/lib/postgresql/18/docker -R
--slot=replication_slot --host=postgres_primary --port=5432 -X stream; do
            echo 'pg_basebackup failed. Retrying in 5 seconds ...'
            sleep 5
          done

          echo 'Backup done, starting replica...'
          chmod 0700 /var/lib/postgresql/18/docker

          postgres
          "
    depends_on:
      - postgres_primary

```

env.sh

```
alias dcu="sudo docker-compose up -d"
alias dcd="sudo docker-compose down"
alias dps="sudo docker ps"
alias der="sudo docker exec -it streaming-replication-postgres_replica-1 bash"
alias dep="sudo docker exec -it streaming-replication-postgres_primary-1 bash"
alias dlr="sudo docker logs streaming-replication-postgres_replica-1"
alias dlp="sudo docker logs streaming-replication-postgres_primary-1"
```

2. Menjalankan docker-compose

Catatan: *command* yang dituliskan di sini memanfaatkan alias yang telah didefinisikan di file `env.sh`.

```
[bpdp@ZimeraArtix ~/kerjaan/utdi/kuliah/ganjil-2025-2026/sistem-terdist-terdes/04/src/bdp/streaming-replication$ dcu
[sudo] password for bpdp:
[+] Running 3/3
  ✓ Network streaming-replication_default          Created
  ✓ Container streaming-replication-postgres_primary-1 Started
  ✓ Container streaming-replication-postgres_replica-1 Started
[bpdp@ZimeraArtix ~/kerjaan/utdi/kuliah/ganjil-2025-2026/sistem-terdist-terdes/04/src/bdp/streaming-replication$ ]
```

Jika *image* PostgreSQL 18 pada OS Alpine Linux (**postgres:18-alpine** - https://hub.docker.com/_/postgres) belum di-pull (diambil) dari DockerHub, maka docker-compose akan mengambil terlebih dahulu, tampilannya akan berbeda dari tampilan di atas.

Periksa apakah kedua *image* untuk PostgreSQL 18 di Alpine Linux tersebut telah berhasil diaktifkan:

```
[bpdp@ZimeraArtix ~/kerjaan/utdi/kuliah/ganjil-2025-2026/sistem-terdist-terdes/04/src/bdp/streaming-replication$ dps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
ed5973af9efb   postgres:18-alpine "docker-entrypoint.s..." 2 minutes ago Up 2 minutes (healthy) 0.0.0.0:5433->5432/tcp, [::]:5433->5432/tcp streaming-replication-po
stgres_replica-1
f13c77a3454f   postgres:18-alpine "docker-entrypoint.s..." 2 minutes ago Up 2 minutes (healthy) 0.0.0.0:5432->5432/tcp, [::]:5432->5432/tcp streaming-replication-po
stgres_primary-1
[bpdp@ZimeraArtix ~/kerjaan/utdi/kuliah/ganjil-2025-2026/sistem-terdist-terdes/04/src/bdp/streaming-replication$ ]
```

Periksa terutama pada **STATUS**. Jika berjalan dengan baik, maka akan muncul **healthy**. Jika tidak, maka bisa di-*trace* menggunakan log dari Docker:

```
[bpdp@ZimeraArtix ~/kerjaan/utdi/kuliah/ganjil-2025-2026/sistem-terdist-terdes/04/src/bdp/streaming-replication$ dlr
pg_basebackup: error: connection to server at "postgres_primary" (172.19.0.2), port 5432 failed: Connection refused
Is the server running on that host and accepting TCP/IP connections?
pg_basebackup failed. Retrying in 5 seconds ...
Backup done, starting replica...
2025-10-26 04:56:22.436 UTC [1] LOG: starting PostgreSQL 18.0 on x86_64-pc-linux-musl, compiled by gcc (Alpine 14.2.0)
2025-10-26 04:56:22.436 UTC [1] LOG: listening on IPv4 address "0.0.0.0", port 5432
2025-10-26 04:56:22.436 UTC [1] LOG: listening on IPv6 address ":::", port 5432
2025-10-26 04:56:22.441 UTC [1] LOG: listening on Unix socket "/var/run/postgresql/.s.PGSQL.5432"
2025-10-26 04:56:22.448 UTC [17] LOG: database system was interrupted; last known up at 2025-10-26 04:56:21 UTC
2025-10-26 04:56:22.965 UTC [17] LOG: starting backup recovery with redo LSN 0/2000028, checkpoint LSN 0/2000080, on
entering standby mode
2025-10-26 04:56:22.975 UTC [17] LOG: redo starts at 0/2000028
2025-10-26 04:56:22.975 UTC [17] LOG: completed backup recovery with redo LSN 0/2000028 and end LSN 0/2000120
2025-10-26 04:56:22.975 UTC [17] LOG: consistent recovery state reached at 0/2000120
2025-10-26 04:56:22.975 UTC [1] LOG: database system is ready to accept read-only connections
2025-10-26 04:56:23.009 UTC [18] LOG: started streaming WAL from primary at 0/3000000 on timeline 1
2025-10-26 05:01:37.481 UTC [15] LOG: restartpoint starting: time
2025-10-26 05:01:41.740 UTC [15] LOG: restartpoint complete: wrote 42 buffers (0.3%), wrote 2 SLRU buffers; 0 WAL fil
=0.013 s, total=4.260 s; sync files=10, longest=0.008 s, average=0.002 s; distance=16701 kB, estimate=16701 kB; lsn=0/
2025-10-26 05:01:41.740 UTC [15] LOG: recovery restart point at 0/304F530
2025-10-26 05:01:41.740 UTC [15] DETAIL: Last completed transaction was at log time 2025-10-26 04:57:17.96269+00.
[bpdp@ZimeraArtix ~/kerjaan/utdi/kuliah/ganjil-2025-2026/sistem-terdist-terdes/04/src/bdp/streaming-replication$ ]
```

Jika berada pada status **healthy**, maka pada titik ini kita telah mempunyai 2 *instances* dari PostgreSQL 18:

1. streaming-replication-postgres_primary-1
2. streaming-replication-postgres_replica-1

Server **streaming-replication-postgres_primary-1** berfungsi sebagai server utama dan akan menerima operasi *read-write* data. Sementara itu, server **streaming-replication-postgres_replica-1** berfungsi sebagai *standby* server yang akan melakukan sinkronisasi dengan server utama sehingga data di server utama maupun di *standby* server merupakan data yang sama. Semua operasi penulisan dan manipulasi data di server utama akan direplikasi ke server replica.

3. Pengujian

Untuk melihat apakah replikasi ini telah berfungsi, kita bisa melihat dari berbagai informasi di tabel diagnostik PostgreSQL 18 maupun dengan melakukan percobaan manipulasi data tabel.

```
[bpdp@ZimeraArtix ~/kerjaan/utdi/kuliah/ganjil-2025-2026/sistem-terdist-terdes/04/src/bpdp/streaming-replication]$ dep
bad30a7284fa:/$ psql -U zuser -d zdb
psql (18.0)
Type "help" for help.

zdb=# \x
Expanded display is on.
zdb=# SELECT * FROM pg_stat_replication;
-[ RECORD 1 ]-----
pid          | 59
usesysid     | 16385
username     | replicator
application_name | walreceiver
client_addr  | 172.19.0.3
client_hostname |
client_port  | 43154
backend_start | 2025-10-26 09:20:05.401535+00
backend_xmin  |
state        | streaming
sent_lsn     | 0/304F638
write_lsn    | 0/304F638
flush_lsn    | 0/304F638
replay_lsn   | 0/304F638
write_lag    |
flush_lag    |
replay_lag   |
sync_priority | 0
sync_state   | async
reply_time   | 2025-10-26 09:25:56.405494+00
```

```

zdb=# SELECT * FROM pg_replication_slots;
-[ RECORD 1 ]-----+-----
slot_name          | replication_slot
plugin              |
slot_type           | physical
datoid              |
database            |
temporary           | f
active              | t
active_pid          | 59
xmin                |
catalog_xmin         |
restart_lsn          | 0/304F638
confirmed_flush_lsn |
wal_status           | reserved
safe_wal_size        |
two_phase            | f
two_phase_at         |
inactive_since       |
conflicting           |
invalidation_reason  |
failover             | f
syncd                | f

zdb=#

```

Pada server *replica* kita bisa melihat status:

```

zdb=# SELECT pg_is_in_recovery();
 pg_is_in_recovery
-----
t
(1 row)

```

Hasil *true* (t) menunjukkan bahwa server tersebut merupakan *replica* / *standby server* yang hanya berfungsi mereplikasi data dan kemudian menyediakan akses *read* bagi data.

Untuk melihat efek dari replikasi ini pada kedua server, kita akan membuat tabel dan kemudian mengisi data tabel tersebut di server utama (*primary*). Setelah itu kita akan melihat kondisi di server *replica*.

Sebelum pembuatan tabel

Primary

```
bad30a7284fa:/$ psql -U zuser -d zdb
psql (18.0)
Type "help" for help.

zdb=# \dt
Did not find any tables.
zdb=#
```

Replica

```
38f81150f98c:/$ psql -U zuser -d zdb
psql (18.0)
Type "help" for help.

zdb=# \dt
Did not find any tables.
zdb=#
```

Manipulasi data pada *Primary*

```

bad30a7284fa:/$ psql -U zuser -d zdb
psql (18.0)
Type "help" for help.

zdb=# \dt
Did not find any tables.
zdb=# CREATE TABLE testrep (first INT, second VARCHAR(15));
CREATE TABLE
zdb=# SELECT * FROM testrep;
 first | second
-----+-----
(0 rows)

zdb=# INSERT INTO testrep (first, second) VALUES (21, 'Test pertama');
INSERT 0 1
zdb=# INSERT INTO testrep (first, second) VALUES (25, 'Test ke dua');
INSERT 0 1
zdb=# SELECT * FROM testrep;
 first | second
-----+-----
    21 | Test pertama
    25 | Test ke dua
(2 rows)

zdb=# 

```

Secara otomatis, manipulasi data tersebut akan direplikasi ke server *replica*:

```

38f81150f98c:/$ psql -U zuser -d zdb
psql (18.0)
Type "help" for help.

zdb=# \dt
Did not find any tables.
zdb=# \dt
          List of tables
 Schema |   Name   | Type  | Owner
-----+-----+-----+-----
 public | testrep  | table | zuser
(1 row)

zdb=# SELECT * from testrep;
 first |      second
-----+-----
    21 | Test pertama
    25 | Test ke dua
(2 rows)

zdb=# █

```

4. **High-Availability**

Model replikasi ini bisa digunakan untuk keperluan HA (*High Availability*), jika salah satu atau *primary server* dalam kondisi *down*, maka salah satu *replica* bisa menggantikan. Berikut adalah cara yang bisa digunakan.

Primary server mati

```

[bpdp@ZimeraArtix ~/kerjaan/utdi/kuliah/ganjil-2025-2026/sistem-terdist-terdes/04/src/bpdp/streaming-replication$ sudo docker-compose stop postgres_primary
[sudo] password for bpdp:
[*] Stopping 1/1
✓ Container streaming-replication-postgres_primary-1 Stopped
[bpdp@ZimeraArtix ~/kerjaan/utdi/kuliah/ganjil-2025-2026/sistem-terdist-terdes/04/src/bpdp/streaming-replication$ █

```

Setelah kita mengetahui bahwa *primary server* dalam kondisi *down*, maka kita bisa mempromosikan salah satu *replica server* untuk menjadi *primary* - dari kondisi hanya bisa menyediakan akses *read* menjadi *primary server* yang bisa menyediakan akses *read-write*. Berikut adalah perintah *pg_promote()* untuk mempromosikan *replica* menjadi *primary*:


```
zdb=# SELECT pg_is_in_recovery();
pg_is_in_recovery
```

```
-----
```

```
 t
(1 row)
```

```
zdb=# SELECT pg_promote();
pg_promote
```

```
-----
```

```
 t
(1 row)
```

```
zdb=# SELECT pg_is_in_recovery();
pg_is_in_recovery
```

```
-----
```

```
 f
(1 row)
```

```
zdb=# █
```