BDR Replication PostgreSQL 9.4

***Lakukan di kedua server/node**

A. Install beberapa dependency
   1. Untuk ubuntu 20.04

```
sudo apt install libreadline-dev
sudo apt install bison
sudo apt install openjade
sudo apt install flex
sudo apt install build-essential
sudo apt install zlib1g-dev
sudo apt install libssl-dev
sudo apt install libxml2-dev
sudo apt install libxslt1-dev
sudo apt install docbook
sudo apt install docbook-dsssl
sudo apt install docbook-xsl
sudo apt install docbook-xml
sudo apt install docbook-utils
sudo apt install xmlto
sudo apt install xsltproc

sudo apt-get install libxml2=2.9.10+dfsg-5ubuntu0.20.04.7
sudo apt-get install libssl1.1=1.1.1f-1ubuntu2.23
sudo apt-get install postgresql-common
```

   2. Verifikasi (opsional)

```
ls -l /usr/lib/x86_64-linux-gnu/libcrypto.so
ldconfig -p | grep libssl
dpkg -l | grep libssl-dev
find /usr -name collateindex.pl
```

B. Install dan konfigurasi di Node1
   1. Tambah repository PostgreSQL 9.4

```
sudo touch /etc/apt/sources.list.d/pgdg.list
sudo nano /etc/apt/sources.list.d/pgdg.list
```

      lalu tambahkan

```
deb [arch=amd64] http://apt.postgresql.org/pub/repos/apt/ bionic-pgdg main
deb-src http://apt.postgresql.org/pub/repos/apt/ bionic-pgdg main
```

   2. Buat directory /opt/build lalu Download Postgres 9.4 yang telah di modifikasi dan plugin bdr:

```
sudo mkdir /opt/build
cd /opt/build
wget https://github.com/2ndQuadrant/bdr/archive/bdr-plugin/1.0.2.tar.gz
wget https://github.com/2ndQuadrant/bdr/archive/bdr-pg/REL9_4_12-1.tar.gz
```

   3. Extract postgres 9.4 dan plugin bdr lalu hapus file tar.gz:

```
tar -xzvf REL9_4_12-1.tar.gz
tar -xzvf 1.0.2.tar.gz
sudo rm -f REL9_4_21-1.tar.gz
```

```
sudo rm -f 1.0.7.tar.gz
```

4. Konfigurasi dan Instalasi PostgreSQL:

```
sudo apt-get -y install apt-transport-https
sudo apt-get -y install curl ca-certificates
curl https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add -
sudo apt-get update && sudo apt-get -y build-dep postgresql-9.4
```

Masuk ke directory postgre 9.4

```
cd bdr-postgres-bdr-pg-REL9_4_21-1
sudo ./configure --prefix=/usr/lib/postgresql/9.4 --sysconfdir=/etc/postgresql/9.4 --libdir=/usr/lib/postgresql/9.4 --datadir=/var/lib/postgresql/9.4 --bindir=/usr/lib/postgresql/9.4/bin --enable-debug
sudo make -j4 && sudo make install
```

5. Install plugin bdr

```
cd bdr-bdr-plugin-1.0.7
sudo PATH=/usr/lib/postgresql/9.4/bin:"$PATH" ./configure
sudo make -j4 -s all
sudo make -s install
```

6. Membuat dan memulai cluster PostgreSQL versi 9.4

```
sudo pg_createcluster 9.4 nama_cluster --start
```

Lalu ubah di postgresql.conf dan pg_hba.conf

```
cd /etc/postgresql/9.4/nama_cluster/postgresql.conf
listen_addresses = '*'
max_connections = 100
shared_pre'load_libraries = 'bdr'
wal_level = 'logical'
max_wal_senders = 10
max_replication_slots = 10
max_worker_processes = 10
track_commit_timestamp = on

tambahkan dipaling bawah
bdr.default_apply_delay=2000   # milliseconds
bdr.log_conflicts_to_table=on

cd /etc/postgresql/9.4/nama_cluster/pg_hba.conf
# Database administrative login by Unix domain socket
local   all             postgres                                peer

# TYPE  DATABASE        USER            ADDRESS                 METHOD

# "local" is for Unix domain socket connections only
local   all             all                                     peer
# IPv4 local connections:
host    all             all             127.0.0.1/32            md5
# IPv6 local connections:
host    all             all             ::1/128                 md5
# Allow replication connections from localhost, by a user with the
```

```
local   replication     postgres                                trust
host    replication     postgres        127.0.0.1/32            trust
host    replication     postgres        ::1/128                 trust

# Allow replication connections between nodes
host    replication     all             192.168.18.81/32        trust
host    replication     all             192.168.18.82/32        trust

# Allow access to all db for all user
host    all             all             192.168.18.81/32        password
host    all             all             192.168.18.82/32        password
```

restart postgresql:

```
sudo systemctl restart postgresql
```

cek cluster

```
pg_lsclusters
```

jika cluster tidak berjalan dikarenakan btree_gist, ikuti langkah berikut:
cari btree_gist

```
sudo find / -name "btree_gist*"
```

lalu arahkan ke btree_gist dan install lagi

```
cd ke path/btree_gist
sudo make
sudo make install

restart postgresql
sudo systemctl restart postgresql

cek cluster
pg_lsclusters
```

```
me@node-1:~$ pg_lsclusters
Ver Cluster Port Status Owner    Data directory              Log file
9.4 rep     5432 online postgres /var/lib/postgresql/9.4/rep /var/log/postgresql/postgresql-9.4-rep.log
me@node-1:~$
```

jika tidak ada masalah lanjut ke langkah berikutnya

7. Menambahkan auto start postgresql
   Agar postgresql auto start Ketika os di reboot, karena menggunakan pg_createcluster, cukup ketikkan perintah berikut:

```
sudo systemctl enable postgresql@9.4-nama_cluster
```

Jika ingin menambahkan di crontab:

```
sudo nano /etc/crontab
@reboot /usr/lib/postgresql/9.4/bin/pg_ctl -D /var/lib/postgresql/9.4-nama_cluster start
```

8. Masuk ke database (lakukan di kedua node)

```
su -l postgres
psql
ALTER USER postgres PASSWORD 'postgres';
create database rep;
\c rep
create extension btree_gist;
create extension bdr;
```

```
rep=# create extension btree_gist;
CREATE EXTENSION
rep=# create extension bdr;
CREATE EXTENSION
```

lalu cek apakah sudah terbuat

```
\dx
```

```
rep=# \dx
                              List of installed extensions
    Name    | Version |   Schema   |                     Description
------------+---------+------------+-----------------------------------------------
 bdr        | 1.0.7.0 | pg_catalog | Bi-directional replication for PostgreSQL
 btree_gist | 1.0     | public     | support for indexing common datatypes in GiST
 plpgsql    | 1.0     | pg_catalog | PL/pgSQL procedural language
(3 rows)
```

Jika sudah terbuat maka akan tampil seperti diatas

Buat bdr group di node1:

```
SELECT bdr.bdr_group_create( local_node_name := 'node1', node_external_dsn := 'host=192.168.18.81 user=postgres dbname=rep password=postgres' );
```

```
rep=# SELECT bdr.bdr_group_create( local_node_name := 'node1', node_external_dsn := 'host=192.168.18.81 user=postgres dbname=rep password=postgres' );
 bdr_group_create
------------------

(1 row)
```

Buat bdr group join di node2:

```
SELECT bdr.bdr_group_join( local_node_name := 'node2', node_external_dsn := 'host=192.168.18.82 user=postgres dbname=rep password=postgres', join_using_dsn := 'host=192.168.18.81 user=postgres dbname=rep password=postgres' );
```

```
rep=# SELECT bdr.bdr_group_join( local_node_name := 'node2', node_external_dsn := 'host=192.168.18.82 user=postgres dbname=rep password=postgres', join_using_dsn := 'h
ost=192.168.18.81 user=postgres dbname=rep password=postgres' );
 bdr_group_join
----------------

(1 row)
```

9. Buat table di node1:

```
rep=# create table users ( name VARCHAR(255) );
CREATE TABLE
```

10. Cek table di node2:

```
rep=# \dt
          List of relations
 Schema | Name  | Type  |  Owner
--------+-------+-------+----------
 public | users | table | postgres
(1 row)
```

11. Tambahkan data dari node1:

```
rep=# insert into users (name) values ('test@881 node 1 data 1');
INSERT 0 1
```

12. Tampilkan data di node2, dan tambahkan data dari node2:

```
rep=# select * from users;
           name
-------------------------
 test@881 node 1 data 1
(1 row)

rep=# insert into users (name) values ('test@882 node 2 data 2');
INSERT 0 1
```

13. Tampilkan data di node1:

```
rep=# select * from users;
          name
-----------------------
 test@881 node 1 data 1
 test@882 node 2 data 2
(2 rows)
```