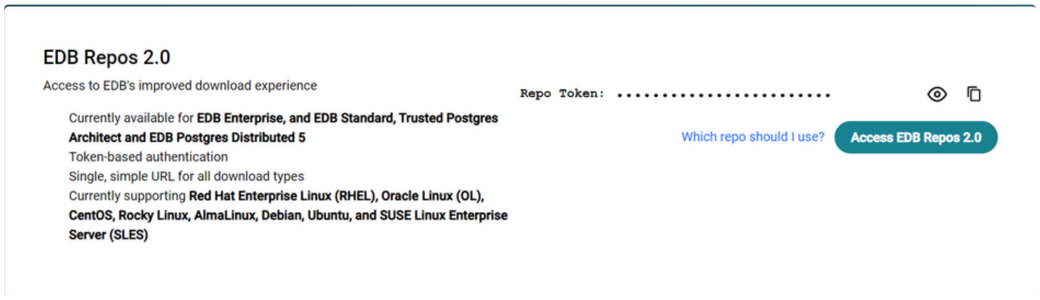
Setting Up Bi-Directional Replication in PostgreSQL with BDR Clustering

Description:

BDR (Bi-Directional Replication) is a tool for PostgreSQL that lets you set up a group of servers where you can make changes on any server, and those changes will be shared with all the other servers in the group. This allows multiple servers to work together, keeping the data in sync across them.

Installation : For installation follow these Steps

1. Create a user account on the EnterpriseDB website to obtain a token.



EDB Repos 2.0

Access to EDB's improved download experience

- Currently available for **EDB Enterprise, and EDB Standard, Trusted Postgres Architect and EDB Postgres Distributed 5**
- Token-based authentication
- Single, simple URL for all download types
- Currently supporting **Red Hat Enterprise Linux (RHEL), Oracle Linux (OL), CentOS, Rocky Linux, AlmaLinux, Debian, Ubuntu, and SUSE Linux Enterprise Server (SLES)**

Repo Token: ••••••••••••••••••••••••

2. Set up the following repositories after completing this step.

```
curl -1sLf 'https://downloads.enterprisedb.com/<Token>/enterprise/setup.rpm.sh' | sudo -E bash

```

```
dnf install yum-utils
rpm --import 'https://downloads.enterprisedb.com/<Token>/enterprise/gpg.E71EB0829F1EF813.key'
curl -1sLf 'https://downloads.enterprisedb.com/<Token>/enterprise/config.rpm.txt?distro=el&codename=8' > /tmp/enterprise.repo
dnf config-manager --add-repo '/tmp/enterprise.repo'
dnf -q makecache -y --disablerepo='*' --enablerepo='enterprisedb-enterprise'
```

3. Install the following BDR packages next:

```
sudo dnf -y install edb-bdr5-pg16 edb-pgd5-proxy edb-pgd5-cli
```

Upon executing this command, the following packages will be installed:

```
                                                                                           324  B/s | 659  B     00:02
Dependencies resolved.
==============================================================================================================================================================
 Package                                      Architecture            Version                                    Repository                                    Size
==============================================================================================================================================================
Installing:
 edb-bdr5-pg16                                x86_64                  4:5.5.1-3.el8                              enterprisedb-postgres_distributed                  715 k
 edb-pgd5-cli                                 x86_64                  5.5.0-1                                    enterprisedb-postgres_distributed                  4.4 M
 edb-pgd5-proxy                               x86_64                  5.5.0-1                                    enterprisedb-postgres_distributed                  3.7 M
```

```
Installing dependencies:
 postgresql16-contrib                  x86_64                  16.4-1PGDG.rhel8                    pgdg16
760 k

Transaction Summary
==============================================================================================================
==============================================
```

4. Set the password for the postgres OS user.

```
passwd postgres
Changing password for user postgres.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
```

5. Add the postgres user to the sudoer list.

   Execute this command.

```
sudo visudo
```

   Add the following line

```
postgres ALL=(ALL) ALL
```

6. Performing two initial database setups.

```
./initdb -D /var/lib/pgsql/16/node_data -A trust
```

```
 ./initdb -D /var/lib/pgsql/16/node2_data -A trust
```

7. Configure the following settings in both instances of the database:

   DB1 Instance

```
echo -e "shared_preload_libraries = '\$libdir/bdr'" | sudo -u postgres tee -a
/var/lib/pgsql/16/node_data/postgresql.conf >/dev/null
echo -e "wal_level = 'logical'" | sudo -u postgres tee -a /var/lib/pgsql/16/node_data/postgresql.conf
>/dev/null
echo -e "track_commit_timestamp = 'on'" | sudo -u postgres tee -a /var/lib/pgsql/16/node_data/postgresql.conf
>/dev/null
echo -e "listen_addresses = '*'" | sudo -u postgres tee -a /var/lib/pgsql/16/node_data/postgresql.conf
>/dev/null
echo -e "max_worker_processes = '16'" | sudo -u postgres tee -a /var/lib/pgsql/16/node_data/postgresql.conf
>/dev/null
echo -e "host all all all md5\nhost replication all all md5" | sudo tee -a
/var/lib/pgsql/16/node_data/pg_hba.conf >/dev/null

 ./psql -U postgres
psql (16.4)
Type "help" for help.

postgres=# ALTER USER postgres WITH PASSWORD 'test'
```

   DB2 Instance

```
echo -e "shared_preload_libraries = '\$libdir/bdr'" | sudo -u postgres tee -a
/var/lib/pgsql/16/node2_data/postgresql.conf >/dev/null
echo -e "wal_level = 'logical'" | sudo -u postgres tee -a /var/lib/pgsql/16/node2_data/postgresql.conf
>/dev/null
```

```
echo -e "track_commit_timestamp = 'on'" | sudo -u postgres tee -a /var/lib/pgsql/16/node2_data/postgresql.conf
>/dev/null
echo -e "listen_addresses = '*'" | sudo -u postgres tee -a /var/lib/pgsql/16/node2_data/postgresql.conf
>/dev/null
echo -e "port = 6000" | sudo -u postgres tee -a /var/lib/pgsql/16/node2_data/postgresql.conf >/dev/null
echo -e "max_worker_processes = '16'" | sudo -u postgres tee -a /var/lib/pgsql/16/node2_data/postgresql.conf
>/dev/null
echo -e "host all all all md5\nhost replication all all md5" | sudo tee -a
/var/lib/pgsql/16/node2_data/pg_hba.conf >/dev/null

./psql -U postgres -p 5433
psql (16.4)
Type "help" for help.

postgres=# ALTER USER postgres WITH PASSWORD 'test'
```

8. Start both database instances.

```
./pg_ctl -D /var/lib/pgsql/16/node_data/ start
```

```
./pg_ctl -D /var/lib/pgsql/16/node2_data/ start
```

Setting Up the Bi-Directional Replication Cluster

1. Setup first master node

   Create database

```
postgres=# CREATE DATABASE node1;
CREATE DATABASE
```

   Create BDR extension

```
CREATE EXTENSION bdr CASCADE;
CREATE EXTENSION
```

   Execute the creation of a node to register it as a PGD node.

```
 SELECT bdr.create_node(node_name := 'master', local_dsn := 'port=5432 dbname=node1 host=localhost
user=postgres');
 create_node
-------------
  3582890138
(1 row)
```

   Create the PGD node group

```
node1=# SELECT bdr.create_node_group(node_group_name := 'bdrtest-group');
 create_node_group
-------------------
         117160161
(1 row)
```

2. Set up the second master node.

   Create database

```
postgres=# CREATE DATABASE node2;
CREATE DATABASE
```

   Create BDR extension

```
CREATE EXTENSION bdr CASCADE;
CREATE EXTENSION
```

Create the second node

```
pgdtest=# SELECT bdr.create_node(node_name := 'secondmasternode', local_dsn := 'port=5433 dbname=node2 host=localhost user=postgres');
 create_node
-------------
  3139556418
(1 row)
```

Execute the command to join the existing node group by using the "join node group" .

```
node2=# SELECT bdr.join_node_group(join_target_dsn := 'port=5432 dbname=node1 host=localhost user=postgres');
 join_node_group
-----------------

(1 row)
```

Sample data and replication

1. Create the following table on the first node and insert some data into the table.

```
CREATE TABLE cities (
    id SERIAL PRIMARY KEY,
    name VARCHAR(50),
    population INTEGER
);
INSERT INTO cities (name, population)
VALUES
('New York', 8419000),
('Los Angeles', 3980000),
('Chicago', 2716000);

```

2. Validate the data on the 2nd master node.

```
[postgres@f8949e21db0c bin]$ ./psql -U postgres -p 5433
psql (16.4)
Type "help" for help.

postgres=# \c node2
You are now connected to database "pgdtest" as user "postgres".
pgdtest=# select * from cities;
 id |    name     | population
----+-------------+------------
  2 | New York    |    8419000
  3 | Los Angeles |    3980000
  4 | Chicago     |    2716000
(3 rows)

pgdtest=#
```

3. Now, input data on the 2nd master node and proceed to verify it on the 1st master.

```
INSERT INTO cities (name, population)
VALUES ('Houston', 2328000),
('Phoenix', 1690000);

```

Check the availability of data on the first master node.

```
./psql -U postgres
psql (16.4)
Type "help" for help.

postgres=# \c node1
You are now connected to database "pgdtest" as user "postgres".
pgdtest=# select * from cities;
   id    |    name     | population
---------+-------------+------------
       2 | New York    |    8419000
       3 | Los Angeles |    3980000
       4 | Chicago     |    2716000
 2000002 | Houston     |    2328000
 2000003 | Phoenix     |    1690000
(5 rows)

```

4. Get BDR nodes detail

```
pgdtest=# select node_id,node_name,node_state,target_state,node_dsn from bdr.node;
  node_id   |    node_name     | node_state | target_state |                       node_dsn
------------+------------------+------------+--------------+-------------------------------------------------------
 3582890138 | master           |         80 |           80 | port=5432 dbname=node1 host=localhost user=postgres
 3139556418 | secondmasternode |         80 |           80 | port=5433 dbname=node2 host=localhost user=postgres
(2 rows)

```

5. Print the node group details

```
pgdtest=# select node_group_id,node_group_name from bdr.node_group;
 node_group_id | node_group_name
---------------+-----------------
     117160161 | pgdtest-group
(1 row)
```