Synchronizing PostgreSQL databases

Prerequisites

Two servers, db-master and db-replica the PostgreSQL installed on both servers.

Configuring PostgreSQL for Logical Replication

- Configure Postgres to listen on the private network interface .
- On **db-master**, open /etc/postgresql/10/main/postgresql.conf, the main server configuration file:
- $*sudo nano /etc/postgresql/10/main/postgresql.conf*

Now add dm-master private ip to enable connections on the private network, **find and uncomment this lines in the file**:

```
listen_addresses = 'localhost,192.168.8.104' # what IP address(es) to listen on;
wal_level = logical                # minimal, replica, or logical
```

Save file and close.

Edit /etc/postgresql/10/main/pg_hba.conf, the file that controls allowed hosts, authentication,and access to databases use replica's ip:

```
...
# TYPE  DATABASE  USER   ADDRESS         METHOD

…

host    all       all    192.168.8.1.03/24   md5
```

Save file and close

Set our firewall rules to allow traffic from **db-replica** to port 5432 on **db-master**:

-
  ```
  $ sudo ufw allow from 192.168.8.103 to any port 5432
  ```

Restart the PostgreSQL server:

-
  ```
  $ sudo systemctl restart postgresql
  ```

Setting Up a Database, User Role, and Table

Using the \connect meta-command on postgreSQL shell, connect to the databases created on both server use similar names:

```
postgres= # \c example_db
```

Create tables with arbitrary fields on both hosts column names.

NB: The table on **db-replica** does not need to be identical to its **db-master** counterpart. However, it must contain every single column present on the table at **db-master**. Additional columns must not have NOT NULL or other constraints. If they do, replication will fail.

- On **db-master**, create a new user role with the REPLICATION option and a login password. The REPLICATION attribute must be assigned to any role used for replication:

  ```
  example_db=# CREATE ROLE replica_user WITH REPLICATION LOGIN
  PASSWORD 'P@ssword123';
  ```

- Grant full privileges on the database to the user role you just created:

```
example_db=# GRANT ALL PRIVILEGES ON DATABASE example_db TO
replica_user;
```

Setting Up a Publication

1. *Publications* are the mechanism that PostgreSQL uses to make tables available for replication. The database server will keep track internally of the connection and replication status of any replica servers associated with a given publication. On **db-master**, you will create a publication:

   ```
   example_db=# CREATE PUBLICATION my_publication;
   ```

2. Add the tables created to *my_publication:*

   ```
   ALTER PUBLICATION  my_publication ADD TABLE table_1, table_2,...,
   table_n;
   ```

   NB: With your publication in place, now add a subscriber that will pull data from it.

Creating a Subscription

1. *Subscriptions* are used by PostgreSQL to connect to existing publications. A publication can have many subscriptions across different replica servers, and replica servers can also have their own publications with subscribers. To access the data from the table you created on **db-master**, create a subscription to the publication created in the previous**:**

   ```
   example_db=# CREATE  SUBSCRIPTION my_subscription CONNECTION
   'host=192.168.8.104        port=5432 password=P@ssword123
   user=replica_user dbname=example_db'  PUBLICATION my_publication;
   ```

DONE.