

Proceso de replicación

A continuación se mostrará la lista de comandos a ejecutar para realizar streaming replication del proyecto 02 del Curso de Base de Datos (exclusivo para Ubuntu con PostgreSQL 14):

- Primero nos movemos al usuario postgres con el siguiente comando

```
$ sudo -i -u postgres
```

- De no tener el comando `initdb` en el path, estos se encuentran en el directorio `/usr/lib/postgresql/14/bin`. Para agregarlos como instrucción ejecutable se puede usar el siguiente comando:

```
$ sudo ln -s /usr/lib/postgresql/9.1/bin/initdb /usr/local/bin/
```

Nota: la versión varía dependiendo de la que se tenga instalada, además, este es el directorio para el sistema operativo Ubuntu, este puede variar dependiendo del OS.

- Creamos el cluster que va actuar como `instancia master` o `primary database`

```
$ cd 14/  
$ initdb -D rep_primary_db/data
```

En este caso especificamos el directorio donde el cluster va a ser almacenado. Como yo tengo instalado PostgreSQL 14, el directorio al que quiero ingresar con `cd 14/` es la versión del mismo.

- Editamos el archivo `postgresql.conf` del cluster creado (este archivo debería de estar dentro del directorio especificado en el paso anterior)

```
$ nano rep_primary_db/data/postgresql.conf
```

Dentro del editor con `Ctrl + W` ingresamos `listen_address` para encontrar la línea. Le quitamos `#` para descomentarlo y le colocamos la dirección IP a la cual deseamos que escuche, en este caso usamos `*` para indicar que escuche para cualquier cliente

```
listen_address = '*'      # what IP address(es) to listen on;
```

Luego pasamos a la línea `port` y asignamos algún puerto libre, en mi caso sería `5434`

```
port = 5433
```

Nota: con `netstat` se puede verificar si el puerto está siendo usado. En Ubuntu se puede instalar con el comando `sudo apt install net-tools`. Y con el comando `sudo netstat -lntp | grep -w '5434'` se realiza la verificación.

- Iniciamos la instancia `master/primary`

```
$ pg_ctl -D rep_primary_db/data start
```

- Ahora nos conectamos a `psql` usando el puerto especificado y el nombre de la base de datos

```
$ psql postgres --port=5434
```

Creamos la base de datos `dvdrental` la cual va a ser generada por medio del comando `pg_restore`

```
postgres=# CREATE DATABASE dvdrental;  
CREATE DATABASE
```

Salimos de `psql`

```
postgres=# \q
```

Ahora estando en el directorio donde tenemos guardado el archivo `dvdrental.tar`, ejecutamos

```
$ pg_restore -U postgres -p 5434 -d dvdrental dvdrental.tar
```

Con `-U` especificamos el usuario (el cual es `postgres`), con `-p` especificamos el puerto y con `-d` el nombre de la base de datos.

- Volvemos a ingresar dentro de `psql` en el puerto de la base principal pero en este caso en la base de datos `dvdrental`

```
$ psql dvdrental --port=5434
```

Y creamos un usuario dueño de la replicación con privilegios de replicación

```
dvdrental=# CREATE USER repuser REPLICATION;
```

Salimos de `psql` con

```
dvdrental=# \q
```

- Ahora editamos el archivo `pg_hba.conf`

```
$ nano rep_primary_db/data/pg_hba.conf
```

Y agregamos

```
host      all             repuser      127.0.0.1/32      trust
```

El archivo debería de quedar algo parecido a lo siguiente

```
# TYPE      DATABASE      USER      ADDRESS      METHOD

# "local" is for Unix domain socket connections only
local      all             all              trust
# IPv4 local connections:
host      all             all          127.0.0.1/32      trust
host      all             repuser      127.0.0.1/32      trust
```

Guardamos con `Ctrl + O` y cerramos el archivo con `Ctrl + X`.

- Reiniciamos la instancia

```
$ pg_ctl -D rep_primary_db/data restart
```

- Ahora pasamos a crear la base de datos réplica de la primary database con el comando `pg_basebackup`, el cual simplemente se va a conectar a la base de datos principal y va a copiar todos los archivos de datos a la réplica

```
$ pg_basebackup -h localhost -U repuser --checkpoint=fast -D
rep_replica_db/data -R --slot=replica_dvdrental -C --port=5434
```

Se especifica la dirección, en este caso `localhost`, el usuario `repuser` dueño de la replicación, con `checkpoint=fast` se asegura que el proceso de copia se inicie instantáneamente, el directorio donde se va a guardar la base de datos réplica `rep_replica_db/data`, `-R` indica replicación, se le

da un nombre significativo con `slot`, con `-C` la base principal pueda reciclar el archivo `wal` solo después que la réplica lo haya consumido por completo, y por último el puerto `5434`.

- Ahora, abrimos una nueva terminal e iniciamos sesión como el usuario `postgres`

```
$ sudo -i -u postgres
```

- Abrimos el archivo `postgresql.conf` y cambiamos el puerto de la réplica para que no haya conflictos

```
nano 14/rep_replica_db/data/postgresql.conf
```

```
port = 5435
```

Guardamos y salimos del archivo con `Ctrl + O` y luego `Ctrl + X`.

- Iniciamos la base de datos réplica con el comando `pg_ctl`

```
$ pg_ctl -D 14/rep_replica_db/data start
```

- En la terminal original entramos en `psql` en el puerto que indicamos para la base de datos principal

```
$ psql dvdrental --port=5434
```

Realizamos un `SELECT` de los registros de la tabla `staff`, por ejemplo,

```
dvrental=# SELECT * FROM staff;
```

Y en la otra terminal ingresamos también a `psql` pero en esta ocasión en el puerto de la base de datos réplica

```
$ psql dvdrental --port=5435
```

Y realizamos el mismo `SELECT` en tabla `staff`

```
dvdrental=# SELECT * FROM staff;
```

Si ambas transacciones devuelven los mismos datos esto quiere decir que la réplica está funcionando. Cualquier dato que agregemos a la base principal será reflejada en la base réplica.

Verificar estado de las base de datos

Con

```
dvdrental=# \x
Extended display is on
dvdrental=# SELECT * FROM PG_STAT_REPLICATION;
```

Podemos ver el estado de la base de datos principal con respecto a su réplica (este comando se ejecuta dentro de la base de datos principal).

Y con

```
dvdrental=# \x
Extended display is on
dvdrental=# SELECT * FROM PG_STAT_WAL_RECEIVER;
```

Podemos ver el estado de la base de datos réplica (este comando se ejecuta dentro de la base de datos réplica).

Referencias:

- <https://www.postgresql.org/docs/current/app-pgrestore.html>
- <https://www.postgresql.org/docs/current/creating-cluster.html>
- <https://www.postgresqltutorial.com/postgresql-getting-started/load-postgresql-sample-database/>
- <https://www.youtube.com/watch?v=Yy0GJJRQcRQ>
- <https://www.postgresql.org/docs/current/app-initdb.html>
- <https://www.postgresql.org/docs/current/app-pgbasebackup.html>