

LABORATORIO 8

Enunciado del problema

Las pruebas de sharding de la base de datos fueron un éxito, pero el equipo DBA no está del todo satisfecho por algunas de las limitaciones que encontraron utilizando los data wrappers. En tal sentido han considerado que sería una mejor solución montar réplicas de la BD central, lo cual además agregaría la posibilidad de tener una estructura robusta con la que se pudiera tanto balancear carga como proporcionar alta disponibilidad.

Para ello han decidido comenzar probando una solución de replicación lógica, con la que esperan que cualquier sentencia DML que se ejecute en el servidor central se reproduzca en las diferentes réplicas y que, incluso, se puedan igualmente replicar sentencias DDL sin mucho esfuerzo.

Por ahora no se van a complicar con varios servidores físicos, por lo que utilizarán la posibilidad de mantener varios clusters de BD, con su respectiva configuración y ubicación en disco, en el mismo servidor y así simular como se realizará la replicación en producción más adelante.

Ejercicios

1. Comience la instalación de PostgreSQL desde cero (por si acaso), ya que hemos hecho muchos cambios en los archivos de configuración con laboratorios pasados.

a) Elimine la versión de postgresql con el comando:

```
apt purge postgresql postgresql-11 postgresql-common
```

Si pregunta "Remove PostgreSQL directories when package is purged?" responda Sí.

b) Elimine paquetes innecesarios y directorios de configuración

```
apt autoremove
```

```
rm -r /var/lib/postgresql/
```

```
rm -r /etc/postgresql-common/
```

c) Instale PostgreSQL

```
apt install postgresql
```

d) Corrobore la instalación del cluster por defecto y el estado del mismo con los siguientes comandos:

```
pg_lsclusters
```

```
pg_ctlcluster 11 main status
```

2. Para la simulación de otro servidor cree otro cluster con el siguiente comando:

```
pg_createcluster 11 standby
```

Compruebe que se tienen dos directorios de configuración en /etc/postgresql (en postgresql.conf al nuevo cluster se le ha asignado otro puerto, 5433).

3. Inicie el nuevo cluster manualmente con el siguiente comando:

```
pg_ctlcluster 11 standby start
```

También puede utilizar el script de gestión general del servicio de PostgreSQL que ejecuta el comando indicado sobre todos los clusters

4. Instale el paquete de la extensión para replicación lógica denominada pglogical (<https://www.2ndquadrant.com/es/resources/pglogical/>)

```
apt-get install postgresql-11-pglogical
```

5. Configure los siguientes parámetros en el servidor de BD. Recuerde que habrá dos subdirectorios, main (que corresponde al server primario) y standby (que corresponde a la réplica).

a) En los dos postgresql.conf,

```
wal_level = 'logical'
```

```
max_worker_processes = 10
max_replication_slots = 10
max_wal_senders = 10
shared_preload_libraries = 'pglogical'
track_commit_timestamp = on
```

- b) En los dos `pg_hba.conf`, cambie de `peer` por `md5` en las líneas de acceso local y de replicación para usuarios no postgres.

```
local all          all          md5
local replication  all          md5
```

- c) Reinicie todo el SGBD (ambos clusters):

```
service postgresql restart
(o /etc/init.d/postgresql restart)
```

6. Con el usuario postgres conéctese al nodo primario y ejecute los siguientes comandos:

```
psql -p 5432
```

- a) crear el usuario dueño de la BD, mismo que ejecutará la replicación

```
create user admin password 'lerolero' createdb superuser
replication;
```

- b) conectarse con dicho usuario, crear la BD y su estructura

```
\c - admin
\i estructura-ucasoft.sql
```

- c) crear la extensión `pglogical` en la BD `ucasoft`

```
create extension pglogical;
```

- d) crear el nodo primario (proveedor) en la BD como usuario `admin`

```
SELECT pglogical.create_node(
    node_name := 'proveedor',
    dsn := 'host=localhost port=5432 dbname=ucasoft
           user=admin password=lerolero');
```

- e) Agregamos (en este caso todas) las tablas del esquema public que serán replicadas a un set que denominaremos default.

```
SELECT pglogical.replication_set_add_all_tables  
('default', ARRAY['public']);
```

7. Abra otra terminal y con el usuario postgres conéctese al nodo réplica. Para ello ejecute los siguientes comandos:

```
psql -p 5433
```

- a) crear el usuario dueño de la BD, mismo que ejecutará la replicación

```
create user admin password 'lerolero' createdb superuser  
replication;
```

- b) conectarse con dicho usuario, crear la BD y su estructura

```
\c - admin  
\i estructura-ucasoft.sql
```

- c) crear la extensión pglogical en la BD ucasoft

```
create extension pglogical;
```

- d) crear el nodo réplica (suscriptor) en la BD como usuario admin

```
SELECT pglogical.create_node(  
    node_name := 'suscriptor',  
    dsn := 'host=localhost port=5433 dbname=ucasoft  
          user=admin password=lerolero');
```

- e) Active la conexión con el proveedor para iniciar la replicación:

```
SELECT pglogical.create_subscription(  
    subscription_name := 'ucasoft_suscriptor',  
    provider_dsn := 'host=localhost port=5432 dbname=ucasoft  
                   user=admin password=lerolero');
```

8. Vuelva a la terminal del nodo primario e inserte los datos de prueba

```
\i datos-ucasoft.sql
```

9. Compruebe en la terminal del nodo réplica que fueron replicados los datos

```
select * from contrata;
```

10. El equipo DBA está emocionado por la facilidad con la que configuró este método de replicación, así que ha decidido probar la posibilidad anunciada en la documentación de pglogical de réplica de comandos DDL como la creación de tablas, lo cual es imprescindible para el reproductor de música que quieren utilizar en la próxima superpachanga.

Para ello es necesario utilizar un tipo de trigger especial (trigger de eventos) ya que será necesario incluir las tablas nuevas en el set de tablas replicado en el nodo primario, tal como sigue:

```
CREATE OR REPLACE FUNCTION pglogical_assign_repset()
RETURNS event_trigger AS $$
DECLARE obj record;
BEGIN
    FOR obj IN SELECT * FROM pg_event_trigger_ddl_commands()
    LOOP
        IF obj.object_type = 'table' THEN
            PERFORM pglogical.replication_set_add_table('default',
                obj.objid);
        END IF;
    END LOOP;
END;
$$ LANGUAGE plpgsql;

CREATE EVENT TRIGGER pglogical_assign_repset_trg
ON ddl_command_end
WHEN TAG IN ('CREATE TABLE', 'CREATE TABLE AS')
EXECUTE PROCEDURE pglogical_assign_repset();
```

11. Ejecute en el nodo primario las sentencias contenidas en el archivo reproductor-ucasoft.sql

```
\i reproductor-ucasoft.sql
```

12. Corrobore que efectivamente se transmitieron las sentencias de creación de tablas, foreign keys y datos ejecutando en la réplica la siguiente consulta:

```
SELECT orden,nombre_cancion,artista_cancion,url
FROM playlist,cancion,superpachanga
WHERE superpachanga.anyo=2020
      AND nombre_cancion=cancion.nombre
      AND artista_cancion=cancion.artista
ORDER BY orden ASC;
```

Instrucciones

1. Utilice los archivos con sentencias SQL que se comparten adjuntos a este enunciado.
2. Registre todas las sentencias y comandos requeridos para realizar lo solicitado en un archivo de texto plano.
3. Suba al moodle el archivo comprimido antes generado.