Manual Técnico
Proyecto Fase 2
Sistemas de bases de datos 2

1. Descripción general

En este manual veremos el funcionamiento del sistema implementado en Docker para la replicación maestro - esclavo en PostgreSQL, junto con los mecanismos de failover, failback y respaldos automáticos utilizando pgBackRest. Esto tambien utilizando Redis para tener una bitacora de los movimientos.

2. Requisitos previos para utilizar el programa

- Docker y Docker Compose instalados.
- docker-compose.yml
- pg_hba.conf
- initdb/01_init_replication.sql
- Imagen de PostgreSQL configurada (versión 16 o superior).
- Carpeta de trabajo con permisos adecuados.

3. Arquitectura del proyecto

El sistema está compuesto por dos contenedores:

- pg-bases2 → Nodo maestro (Primario)

- pg-replica → Nodo esclavo (Secundario)

El maestro realiza todas las operaciones de escritura, mientras que el esclavo se mantiene sincronizado en modo lectura mediante replicación física streaming.
Redis se utiliza para registrar los backups automáticos y pgBackRest se encarga de las copias completas, diferenciales e incrementales.

4. **Como se utiliza**
Iniciar los contenedores
Crear el rol de replicación
docker exec -it pg-bases2 psql -U postgres -f /initdb/01_init_replication.sql
Crear el slot de replicación
docker exec -it pg-bases2 psql -U postgres -d bases2_proyectos -c "SELECT pg_create_physical_replication_slot('replica1');"
Clonar la base y levantar la réplica
docker compose run --rm -u postgres --entrypoint bash pg-replica -lc "
export PGPASSWORD='replica_pass'
pg_basebackup -h pg-bases2 -U replicator -D /var/lib/postgresql/data -X stream -R -S replica1 -v
"
docker compose up -d pg-replica

5. Verificación del estado del backup

- Confirmar que la réplica está en modo recuperación:

docker exec -it pg-replica psql -U postgres -c "SELECT pg_is_in_recovery();"
Debe devolver: t

- Verificar conexión entre maestro y esclavo:

docker exec -it pg-bases2 psql -U postgres -c "SELECT application_name, state, sync_state FROM pg_stat_replication;"

- Probar replicación de datos:

docker exec -it pg-bases2 psql -U postgres -d bases2_proyectos -c "INSERT INTO public.replica_test(val) VALUES ('replica-ok');"
docker exec -it pg-replica psql -U postgres -d bases2_proyectos -c "TABLE public.replica_test;"

6. Procedimiento de Failover (convertir el esclavo a maestro)

En caso de falla del maestro:

Detener el contenedor primario:
docker stop pg-bases2

Promover la réplica a primaria:
docker exec -it pg-replica psql -U postgres -c "SELECT pg_promote(wait => true);"

Verificar:
docker exec -it pg-replica psql -U postgres -c "SELECT pg_is_in_recovery();" # Debe devolver 'f'

La réplica ahora deberia funcionar como nuevo maestro.

7. Procedimiento de Failback (restaurar el maestro anterior)

Detener el contenedor antiguo del maestro:
docker stop pg-bases2

Limpiar sus datos:
docker compose run --rm -u postgres --entrypoint bash pg-bases2 -lc "rm -rf /var/lib/postgresql/data/*"

Crear un nuevo slot desde el nodo activo:
docker exec -it pg-replica psql -U postgres -c "SELECT pg_create_physical_replication_slot('primary1');"

Re-clonar el antiguo maestro:
docker compose run --rm -u postgres --entrypoint bash pg-bases2 -lc "
export PGPASSWORD='replica_pass'
pg_basebackup -h pg-replica -U replicator -D /var/lib/postgresql/data -X stream -R -S primary1 -v
"

Iniciar el contenedor nuevamente:
docker compose up -d pg-bases2

Con esto, el antiguo maestro deberia quedar reintegrado como esclavo y la replica se restablece.

8. Respaldos automáticos con pgBackRest
Instalación
docker exec -it pg-bases2 bash -lc "apt-get update && apt-get install -y pgbackrest"

Crear configuración
docker exec -it pg-bases2 bash -lc "cat > /etc/pgbackrest/pgbackrest.conf <<'EOF'
[global]
repo1-path=/var/lib/pgbackrest
log-level-console=info

[bases2-db]
pg1-path=/var/lib/postgresql/data
pg1-port=5432
pg1-user=postgres
EOF"

Crear y verificar la “stanza”
docker exec -u postgres pg-bases2 pgbackrest --stanza=bases2-db stanza-create
docker exec -u postgres pg-bases2 pgbackrest --stanza=bases2-db check

Tipos de backup

Completo:
docker exec -u postgres pg-bases2 pgbackrest --stanza=bases2-db --type=full backup

Diferencial:
docker exec -u postgres pg-bases2 pgbackrest --stanza=bases2-db --type=diff backup

Incremental:
docker exec -u postgres pg-bases2 pgbackrest --stanza=bases2-db --type=incr backup

9. Monitoreo y Redis

Cada vez que se ejecuta un backup, se guarda un registro en Redis con la fecha, tipo y ubicación del archivo.
Esto permite consultar rápidamente los últimos respaldos sin acceder directamente al contenedor.