

## Laboratorio

En este laboratorio comenzarás a practicar instalar pgPool y conectarte a un cluster de Postgres a través de PgPool.

Nota:

Las tareas se realizará sobre una máquina virtual VMWare y un sistema operativo Ubuntu.

### ***Práctica 1: Instalación y conexión de PgPool***

#### ***Objetivos***

El objetivo de la práctica es saber como se instala y se configura pgPool y hacer la conexión utilizando este software.

#### ***Notas***

Para facilitar las prácticas, se ofrece una máquina virtual ya creada ([Ubuntu 16 - pgPool](#)). La máquina virtual a utilizar tiene la siguiente información:

- Usuario: postgresql96
- Contraseña: postgresql
- Cluster: /postgresql/data → port: 5432
- Binarios: /postgresql/bin

Para utilizar la máquina virtual puede utilizar el software gratuito VMWare Player.

Todas las tareas deberán ser ejecutadas en la consola de Linux (Aplicación Terminal)

Para realizar las siguientes tareas, debéis de seguir la [documentación oficial de PgPool](#).



## **Tareas**

- 1.- Descargar (mediante código fuente) y descomprimir pgPool.
- 2.- Compilar e instalar PgPool incluyendo PgPool-recovery. El directorio padre de PgPool debe ser: /postgresql/pgpool. Nota: instalar librería (libpq-dev) si falla el configure. Editar la ruta de "pg\_config" en el archivo Makefile.
- 3.- Añadir el archivo pgpool-recovery.sql a template 1 y editar la configuración de postgresql. [Ver la documentación.](#)
- 4.- Configurar PgPool en el modo RAW. Y la conexión con postgresql (Backend Connection Settings)
- 5.- Crear los directorios necesarios para el PID y para el log
- 6.- Realizar la conexión a postgres mediante pgpool. Nota: arranca pgpool con la opción -n.
- 7.- Desde la conexión a pgpool creamos una base de datos nueva y validamos que la misma se haya creado desde tanto desde la conexión de pgpool como desde la conexión directa al servidor.
- 8.- Conectarse a la nueva base de datos, crear una tabla con los campos id, nombre y apellido e insertar varios valores en ella.  
Revisamos el contenido de la tabla.
- 9.- Realizar el ejercicio anterior, pero con una tabla con fechas.



## ***Soluciones***

1- Descargar (mediante código fuente) y descomprimir pgPool.

<http://www.pgpool.net/mediawiki/index.php/Downloads>

<http://www.pgpool.net/download.php?f=pgpool-ll-3.6.5.tar.gz>

```
gunzip pgpool-ll-3.6.5.tar.gz
```

```
tar xf pgpool-ll-3.6.5.tar
```

2.- Compilar e instalar PgPool incluyendo PgPool-recovery. El directorio padre de PgPool debe ser: /postgresql/pgpool. Nota: instalar librería (libpq-dev) si falla el configure. Editar la ruta de "pg\_config" en el archivo Makefile.

```
Sudo apt-get install libpq-dev
```

```
./configure --prefix=/postgresql/pgpool
```

```
sudo make
```

```
sudo make install
```

```
cd /src/sql/pgpool-recovery
```

```
sudo nano Makefile → PG_CONFIG? = /postgresql/bin/pg_config
```

```
sudo make
```

```
sudo make install
```

3.- Añadir el archivo pgpool-recovery.sql a template 1 y editar la configuración de postgresql.

[Ver la documentación.](#)

```
psql -f ~/Downloads/pgpool-ll-3.6.6/src/sql/pgpool-recovery/pgpool-recovery.sql template1
```

```
nano postgresql.conf → pgpool.pg_ctl = '/postgresql/bin/pg_ctl'
```

```
pg_ctl reload
```



4.- Configurar PgPool en el modo RAW. Y la conexión con postgresql (Backend Connection Settings)

```
cd /postgresql/pgpool/etc
```

```
Sudo cp pgpool.con.sample pgpool.conf
```

```
sudo nano pgpool.conf
```

```
→ backend_hostname0 = 'localhost'
```

```
→ backend_data_directory0 = 'postgresql/data'
```

5.- Crear los directorios necesarios para el PID (/var/run/pgpool) y para el log (/var/log/pgpool)

```
sudo mkdir /var/run/pgpool
```

```
sudo /mkdir /var/log/pgpool
```

6.- Realizar la conexión a postgres mediante pgpool. Nota: arranca pgpool con la opción -n.

```
sudo ./pgpool -n -f ../etc/pgpool.conf
```

```
postgresql96@ubuntu:~$ sudo pgpool -n -f /usr/local/etc/pgpool.conf
2017-08-31 08:00:04: pid 16488: LOG: Backend status file /var/log/pgpool/pgpool_status does not exist
2017-08-31 08:00:04: pid 16488: LOG: Setting up socket for 127.0.0.1:9999
2017-08-31 08:00:04: pid 16488: LOG: pgpool-II successfully started. version 3.6.5 (subaruboshi)
```

```
postgresql96@ubuntu:~$ psql -p 9999 postgres
psql (9.6.4)
Type "help" for help.
Files
postgres=# \conninfo
You are connected to database "postgres" as user "postgresql96" via socket in "/tmp" at port "9999".
postgres=#
```



7.- Desde la conexión a pgpool creamos una base de datos nueva y validamos que la misma se haya creado desde tanto desde la conexión de pgpool como desde la conexión directa al servidor.

```
CREATE DATABASE bd1;
```

```
\l
```

8.- Conectarse a la nueva base de datos, crear una tabla con los campos id, nombre y apellido e insertar varios valores en ella.

Revisamos el contenido de la tabla.

```
\c bd1
```

```
CREATE TABLE tabla1 (id serial, nombre varchar(70), apellido varchar(50));
```

```
INSERT INTO tabla1 (nombre,apellido) VALUES ('Jose','Segovia'),('otroNombre','otroApellido');
```

```
SELECT * FROM tabla1;
```

9.- Realizar el ejercicio anterior, pero con una tabla con fechas.

```
CREATE TABLE prueba_tiempo (id int, tiempo timestamp without time zone);
```

```
INSERT INTO prueba_tiempo VALUES (1,now());
```

```
SELECT * FROM prueba_tiempo;
```