
PostgreSQL

Instalación, Configuración y Trucos

Manuel Angel Rubio Jiménez

Copyright © 2009 Manuel Angel Rubio Jiménez

Resumen

Sobre PostgreSQL se ha escrito mucho, realmente, los manuales que hay disponibles en el sitio oficial para las versiones actuales, cubren casi todos los aspectos, con lo que es complicado aportar algo nuevo. Lo que intento con este escrito es dar información concisa y escueta en lo que respecta a la instalación del servidor, la configuración y un inicio rápido con él. Si necesitas más información, te sugiero el documento oficial, que de seguro es más completo, tanto como guía de referencia, como de aprendizaje.

Instalación, Configuración y Trucos

Esto es algo que siempre me toca buscar en Internet, puesto que es algo que hago una vez cada tantos meses, y siempre se me olvida de cómo empezar, así que, para tener la *chuleta* a mano, he decidido escribir esta entrada que, además de servirme ahora, seguro que me servirá en el futuro para cuando configure más servidores de este tipo.

La ventaja de instalar desde paquetería, es que no hay que generar el directorio de *data* donde almacenar las bases de datos y demás información del sistema que se ejecutará pero, en la mayoría de los casos, la versión de paquetería suele ser una versión algo antigua de PostgreSQL. El sistema de PostgreSQL mantiene un sistema de mantenimiento impresionante, ya que no solo mantienen la versión estable y la anterior, sino también unas 4 ó 5 versiones anteriores. A día de hoy, se mantienen las ramas 7.4, 8.0, 8.1, 8.2 y 8.3. Las diferencias entre cada versión se pueden ver aquí [<http://www.postgresql.org/about/featurematrix>].

Yo me voy a basar en la descarga, compilación e instalación de una versión de código fuente, más en concreto, la versión 8.3.6 [<http://www.postgresql.org/ftp/source/v8.3.6/>].

El método a seguir es:

```
# wget http://wwwmaster.postgresql.org/redis/149/h/source/v8.3.6/postgresql-8.3.6.tar.bz2
# ./configure
# make
# make install
```

Con esto, tendremos instalado el sistema de postgresql en `/usr/local/pgsql`. Si leemos el documento de `INSTALL` que vienen con el código fuente, veremos que se completa la instalación ejecutando:

```
adduser postgres
mkdir /usr/local/pgsql/data
chown postgres /usr/local/pgsql/data
su - postgres
/usr/local/pgsql/bin/initdb -D /usr/local/pgsql/data
/usr/local/pgsql/bin/postgres -D /usr/local/pgsql/data >logfile 2>&1 &
/usr/local/pgsql/bin/createdb test
/usr/local/pgsql/bin/psql test
```

Si estamos en la consola de PostgreSQL tras la ejecución del último comando, es que todo ha salido bien, y no necesitaremos más. Otra cosa, es el script de inicio para que postgresql se inicie cada vez. Yo he realizado un script muy simple, para lanzarlo y pararlo cada vez, sé que hay soluciones mejores, pero bueno, algo es algo.

```
#!/bin/sh

IP=192.168.1.1
PGSQL=/usr/local/pgsql/bin/postgres
DATA=/usr/local/pgsql/data

case $1 in
  start)
    su -c "$PGSQL -D $DATA -i -h $IP >logfile 2>&1" - postgres &
  *)
    exit 1
  esac
```

```
        echo "Iniciado postgres"
        ;;
stop)
    killall postgres
    echo "Parado postgres"
    ;;
*)
    echo "Sintaxis: $(basename $0) {start|stop}"
    ;;
esac
```

Esto lo guardamos como `/usr/local/bin/postgresql` y le damos permisos de ejecución, con lo que, al ejecutarlo con el parámetro `start` o `stop`, nos hará las veces de script de inicio y fin. Se puede colocar o enlazar dentro de `/etc/init.d` para que lo haga automáticamente al arrancar y apagar, respectivamente, el ordenador.

Para la configuración, habrá que entrar en el directorio `/usr/local/pgsql/data`, ahí encontraremos el fichero `pg_hba.conf`. En este fichero se pueden establecer permisos para acceso basándonos en el tipo (vía socket ip versión 4 o versión 6, o socket local de unix), la base de datos, el usuario, la dirección IP de origen o el método: de confianza (trust), rechazo (reject) o autenticación (md5, crypt, password, gss, sspi, krb5, ident, pam o ldap).

Por ejemplo, si tenemos una base de datos a la que queremos acceder desde otro equipo mediante el usuario existente en el sistema `admin` y queremos darle permiso a acceder a una aplicación web, también de otro equipo, para acceso mediante un usuario creado solo a nivel de postgres, pongamos por ejemplo `wordpress`, podemos agregar las siguientes líneas:

```
# TYPE DATABASE USER CIDR-ADDRESS METHOD
host all admin 0.0.0.0/0 pam
host wordpress wordpress 0.0.0.0/0 md5
```

Con esto, ya tenemos el sistema preparado para el acceso, pero no tenemos aún ni el usuario creado a nivel de postgresql, ni los permisos otorgados. Para eso, crearemos la base de datos a la que queramos darle acceso, por ejemplo `wordpress`. Entramos a la consola y asignamos los permisos. Sería algo como esto:

```
# su - postgres
$ /usr/local/pgsql/bin/createdb wordpress
$ /usr/local/pgsql/bin/psql wordpress
Welcome to psql 8.3.6, the PostgreSQL interactive terminal.

Type: \copyright for distribution terms
      \h for help with SQL commands
      \? for help with psql commands
      \g or terminate with semicolon to execute query
      \q to quit

wordpress=# CREATE ROLE wordpress
wordpress=# NOSUPERUSER
wordpress=# NOCREATEDB
wordpress=# CREATEROLE
wordpress=# INHERIT
wordpress=# LOGIN PASSWORD 'wordpress';
wordpress=# GRANT ALL ON DATABASE wordpress TO wordpress;
wordpress=# \q
```

Con esto ya hemos creado el usuario (o rol) `wordpress` dentro de la base de datos del mismo nombre, asignándole la clave y dándole permisos para acceder a esa base de datos.

Para comprobar esto, podemos usar una herramienta como pgAdmin3 [<http://www.pgadmin.org/>] y conectarnos desde otro equipo (o desde el mismo especificando la IP en lugar de `localhost`). Con lo que veremos que se accede correctamente y sin problemas... o eso espero :-P.