PostgreSQL

De Guía Ubuntu

PostgreSQL es un potente motor de bases de datos, que tiene prestaciones y funcionalidades equivalentes a muchos gestores de bases de datos comerciales. Es más completo que MySQL ya que permite métodos almacenados, restricciones de integridad, vistas, etc. aunque en las últimas versiones de MySQL se han hecho grandes avances en ese sentido.

Tabla de contenidos

- 1 Instalación
- 2 Configuración
  - 2.1 Permitir conexiones remotas
  - 2.2 Configurar la lista de acceso
- 3 Gestión de usuarios
- 4 Copia de seguridad
- 5 Ver también
- 6 Enlaces externos

Instalación

En Ubuntu disponemos paquetes para varias versiones de PostgreSQL: 7.4, 8.0, 8.1 y 8.2 de modo que instalaremos la última versión a no ser que necesitemos alguna anterior. Los paquetes necesarios para una instalación completa son los de la parte cliente (**postgresql-client-8.2**) y los de la parte servidora (**postgresql-8.2**). En un ordenador que usemos como cliente sólo necesitamos los paquetes del cliente. También es recomentable instalar un cliente gráfico que nos facilitará la interacción con el servidor. Un cliente gráfico recomendable es pgAdmin III, que funciona muy bien.

Si tienes dudas sobre que necesitas, puedes instalar los tres paquetes:

```
$ sudo aptitude install postgresql-8.2 postgresql-client-8.2 pgadmin3
```

Recuerda desinstalar cualquier versión de postgresql que no necesites para no afectar el rendimiento de tu sistema.

Por razones de seguridad estableceremos la nueva contraseña al usuario del sistema creado por PostgreSQL:

```
$ sudo passwd postgres
```

Cambiar los privilegios de acceso al shell del usuario postgres con el siguiente comando:

```
$ sudo vipw
```

Y cambiamos el shell del usuario postgres de "/bin/false" a "/bin/bash". Luego salimos grabando con escape ":wq". Para verificar si la instalación fue satisfactoria accedemos a la shell del servidor de bases de datos:

```
$ sudo su postgres -c "psql template1"
```

Si el acceso ha sido satisfactorio cambiamos la contraseña al usuario predeterminado del servidor de bases de datos:

```
template1=# ALTER USER postgres WITH PASSWORD 'nueva_contraseña';
```

Te saldrá el siguiente mensaje confirmando la operación:

```
ALTER ROLE
```

Sal de la shell del servidor de bases de datos con el comando \q :

```
template1=# \q
```

Configuración

Permitir conexiones remotas

Por motivos de seguridad, la configuración por defecto no admite conexiones externas. Para habilitarlas tenemos que editar el fichero */etc/postgresql/8.2/main/postgresql.conf*.

```
$ sudo gedit /etc/postgresql/8.2/main/postgresql.conf
```

Ahora buscamos las siguientes líneas que se encuentran comentadas:

```
#listen_addresses = 'localhost'
```

Y la substituimos por la siguiente línea:

```
listen_addresses = '*'
```

Posteriormente buscamos la siguiente línea y le quitamos la marca de comentario:

```
#password_encryption = on
```

Y nos debe quedar lo siguiente:

```
password_encryption = on
```

Guardamos los cambios y reiniciamos el dominio para que los cambios surjan efecto:

```
$ sudo /etc/init.d/postgresql-8.2 restart
```

Configurar la lista de acceso

La configuración de la lista de acceso permite decirle a PostgreSQL qué método de autentificación usar y establecer relaciones de confianza para ciertas máquinas y redes. Hay que editar el fichero */etc/postgresql/8.2/main/pg_hba.conf*:

```
$ sudo vi /etc/postgresql/8.2/main/pg_hba.conf
```

Al final del archivo se encuentra una lista de acceso predeterminada, ahora, dependiendo de su necesidad puedes hacer lo siguiente:

- Si necesita que cualquier usuario se conecte por medio de una dirección IP en especifico, agregue al final la siguiente línea:

```
host all all 192.168.1.4 255.255.255.0 md5
```

La dirección IP y la SubMascara de Red son ejemplos, cámbielas por datos del usuario que requiera realizar la conexión.

- Si necesita que cualquier usuario se conecte por medio de una IP determinada sin importar el password (confiamos en dicha IP), la línea es:

```
host all all 192.168.1.4 255.255.255.255 trust
```

- Si necesita que cualquier usuario *(usuario de base de datos autentificándose, claro)* se conecte por medio de cualquier dirección IP, agregue al final la siguiente línea:

```
host all all 0.0.0.0 0.0.0.0 md5
```

- Si necesita que un usuario determinado se conecte a una base de datos determinada

por medio de una dirección IP en especifico, agregue al final la siguiente línea:

```
host MyDataBase MyUser 192.168.1.4 255.255.255.0 md5
```

- Guarda los cambios realizados en el archivo y reinicia el demonio para que los cambios surjan efecto:

```
$ sudo /etc/init.d/postgresql-8.2 restart
```

Gestión de usuarios

Los usuarios de PostgreSQL tienen un par de capacidades que definimos en su creación. Me explico: un usuario puede o no crear más usuarios y un usuario puede o no crear bases de datos. En el ejemplo que puedes ver a continuación creamos un usuario que no puede crear más usuarios (no es un administrador) pero puede crear más bases de datos. El modificador -P hace que nos pregunte por el password que le asignaremos al usuario. De otra manera el usuario se creará sin password.

```
$ createuser -A -d -P -h host -U usuario nuevo_usuario
Enter password for user "nuevo_usuario":
Enter it again:
```

Como acabo de comentar, existen usuario administradores (pueden crear otros usuarios). Como es lógico este comando debe ser ejecutado por un usuario con esa característica.

Análogamente podemos eliminar un usuario de esta forma:

```
$ dropuser -h host -U usuario usuario_borrar
```

Copia de seguridad

Para hacer una copia de seguridad de una base de datos tiene el siguiente comando:

```
$ pg_dump -h host -U usuario nombre_bd > nombre_bd.sql
```

Para hacer una copia de seguridad de todas las bases de datos PostgreSQL de un servidor, usa este escript:

```
#!/bin/bash

## BEGIN CONFIG ##
HOST=localhost
BACKUP_DIR=tmp
## END CONFIG ##

if [ ! -d $BACKUP_DIR ]; then
  mkdir -p $BACKUP_DIR
fi
```

```
POSTGRE_DBS=$(psql -h $HOST -U postgres -l | awk ' (NR > 2) && (/[a-zA-Z0-9]+[ ]+[|]/) && ( $0 !

for DB in $POSTGRE_DBS ; do
  echo "* Backuping PostgreSQL data from $DB@$HOST..."
  pg_dump -h $HOST -U postgres $DB > $BACKUP_DIR/pg_$DB.sql
done
```

Para restaurar una copia de seguridad:

```
psql -d nombre_base_datos -f archivo.pgdump
```

Ver también

- pgAdmin III
- MySQL

Enlaces externos

- Cómo instalar PostgreSQL en Ubuntu 11.04 (http://lobotuerto.com/blog/2009/07/20/como-instalar-postgresql-en-ubuntu-9-04-jaunty-jackalope/) (*en español*)
- PostgreSQL Práctico (http://www.sobl.org/traducciones/practical-postgres/node1.html) (*en inglés*)
- Sitio web oficial de PostgreSQL (http://www.postgresql.org/) (*en inglés*)
- Comunidad de usuarios de PostgreSQL de Chile (http://www.postgresql.cl/) (*en español*)
- Permitir conexiones entrantes a un servidor PostgreSQL (http://www.linuxhispano.net/2011/02/15/permitir-conexiones-entrantes-a-un-servidor-postgresql/) (*en español*)

Obtenido de "http://www.guia-ubuntu.org/index.php?title=PostgreSQL"

Categoría: Administración del sistema

- Esta página fue modificada por última vez el 17:56 13 jul 2011.
- Esta página ha sido visitada 210.466 veces.
- El contenido está disponible bajo los términos de la *Attribution-NonCommercial-ShareAlike 2.5*
- Política de protección de datos
- Acerca de Guía Ubuntu
- Aviso legal