

Montado de un servidor CentOS

con PostGreSQL 8.4, PostGis 1.5, IP estática, Inicio automático de servicios, etc.

Creado por Maximiliano Castañón
bajo licencia Creative Commons
<http://www.creativecommons.com>

Esta obra está bajo una [Licencia Creative Commons Atribución-NoComercial-CompartirIgual 3.0 Unported](#).

Índice de contenido

Tiempo estimado de instalación y configuración del sistema operativo	3
Configurando repositorios de CentOS con EPEL de Fedora.....	3
Montado de PostGreSQL 8.4 y de PostGIS 1.5.....	3
¿Cómo puedo ubicar los archivos SQL necesarios en mi equipo?.....	4
Inicializando PostGreSQL: configuración, creación de usuarios y otros pasos.....	4
Creando una B.D. con lenguaje PLPGSQL y un usuario con PostGreSQL.....	4
Añadiendo usuarios de PostGreSQL.....	5
Valores para perfiles de usuarios, otorgar o quitar funciones a un ROL.....	5
Eliminando un usuario de PostGreSQL.....	6
Configurando PG_HBA.CONF para permitir conexiones TCP al servidor PGSQL.....	7
Inicio automatizado de PostGreSQL.....	8
Cortafuegos.....	8
IP estática.....	8
Asignar ruta por defecto al iniciar:.....	8
Asignar IP estática para la tarjeta de red eth0:.....	8
Transferencia de archivos con SCP.....	9
Descargando datos con SCP.....	9
Subiendo archivos con SCP.....	9
Transformar archivos SHP y DBF de ArcGis de la compañía ESRI a SQL:.....	10
Opciones para el comando SHP2PGSQL.....	10
Volcado de la BD (Respaldos).....	10
Volcado de la Base de Datos PostGreSQL con PG_DUMP.....	10
Volcando todas las Bases de Datos con PG_DUMPALL.....	11
Volcando usuarios del sistema de Base de Datos PostGreSQL.....	12
Opciones del comando PG_DUMPALL para volcados.....	13
Realizando respaldo con PG_BASEBACKUP.....	14
¿Cómo subir un respaldo de la Base de Datos?.....	14
Subiendo respaldos con PSQL.....	14
MERGE.....	14
Restaurando la B.D. con PG_RESTORE.....	14
Usuarios con PG_IDENT.CONF.....	15

Tiempo estimado de instalación y configuración del sistema operativo

Entre 2 y 4 horas, considerando que todo depende de la velocidad de descarga y el tipo de máquina, cabe recordar que toda la instalación es vía consola porque es un servidor y no requiere entorno gráfico.

Configurando repositorios de CentOS con EPEL de Fedora

Se puede acceder a FedoraProject.org para realizar la descarga del paquete de Fedora para CentOS.

El enlace hasta el momento de creación de esta guía era:

<http://fedoraproject.org/wiki/EPEL/es>

<http://download.fedoraproject.org/pub/epel/6/i386/epel-release-6-8.noarch.rpm>

Para instalar el paquete se hace un

```
rpm -i http://download.fedoraproject.org/pub/epel/6/i386/epel-release-6-8.noarch.rpm
```

O sin firma de paqueterías:

```
rpm --nosignature -i http://download.fedoraproject.org/pub/epel/6/i386/epel-release-6-8.noarch.rpm
```

Montado de PostgreSQL 8.4 y de PostGIS 1.5

Para montar un servidor CentOS se requiere descargar la imagen de la arquitectura del hardware, 64bits o 32bits, etc.

Una vez instalado el sistema operativo deben instalarse los repositorios EPEL que otorgan más paquetes en el repositorio, finalizadas las actualizaciones se debe instalar, PostgreSQL (8.4) y PostGIS (1.5) que deberían estar en el repositorio, cabe recordar que estamos probando las versiones 8.4 y 1.5 que son para CentOS 6.X. Para instalar PostgreSQL y PostGIS tendremos que usar el siguiente comando:

```
Yum update  
Yum install postgresql postgresql-server postgis
```

Aún así, faltan por configurar varios parámetros más en el sistema antes de tener funcionando todo, por ejemplo PostGIS requiere pasos adicionales para configurar sus funciones dentro de la BD de PostgreSQL... ¿Cómo lo hago? Es muy simple, debes buscar y ubicar la ruta a tu archivo **postgis-64.sql** o **postgis.sql** dentro del sistema, a mí personalmente me dió la ruta:

```
/usr/share/postgresql/9.1/contrib/postgis-1.5/postgis.sql  
/usr/share/postgresql/contrib/postgis-64.sql
```

Luego ubicar el archivo spatial_ref_sys.sql, a mí me dió:

```
/usr/share/postgresql/9.1/contrib/postgis-1.5/spatial_ref_sys.sql
```

¿Cómo puedo ubicar los archivos SQL necesarios en mi equipo?

Primero existe un programa llamado “updatedb”, debemos actualizar la Base de Datos del programa, ejecutaremos como administrador “updatedb”, pero debemos tener instalado “mlocate”:

```
yum install mlocate
updatedb
```

y luego:

```
locate postgis.sql
locate spatial_ref_sys.sql
```

Inicializando PostGreSQL: configuración, creación de usuarios y otros pasos

Entonces, antes que todo inicializaremos PostGreSQL con:

```
service postgresql initdb
service postgresql start
```

Creando una B.D. con lenguaje PLPGSQL y un usuario con PostGreSQL

Como usuario administrador con los poderes de super vaca (**root**), haremos:

```
su postgres
createdb -E utf8 postgis
psql -d postgis
    create language plpgsql
```

Este paso depende de la ruta del archivo donde se encuentre:

```
psql -d postgis -f /usr/share/postgresql/9.1/contrib/postgis-1.5/postgis.sql
psql -d postgis -f /usr/share/postgis/contrib/postgis-64.sql
```

```
psql -d postgis -f /usr/share/postgresql/9.1/contrib/postgis-1.5/spatial_ref_sys.sql
createuser -s administrador
createdb mibd -T postgis -O administrador
```

<http://www.postgresql.org/docs/current/static/user-manag.html>

El último comando ingresado, hereda los datos de la BD, entonces permite usar PostGis y además se le asigna como dueño el usuario administrador que hemos creado nosotros, ejemplo:

```
bash-4.1$ createdb mibd -T postgis -O administrador
could not change directory to "/root"
bash-4.1$ psql mibd
```

```
could not change directory to "/root"
psql (8.4.13)
Type "help" for help.
mibd=# select postgis_full_version();
               postgis_full_version
-----
POSTGIS="1.5.3" GEOS="3.3.2-CAPI-1.7.2" PROJ="Rel. 4.7.1, 23 September 2009"
LIBXML="2.7.6" USE_STATS
(1 row)
```

Añadiendo usuarios de PostGreSQL

Para añadir los primeros usuarios de sistema para PostGreSQL necesitamos acceder con el usuario administrador de la B.D., este usuario es “postgres”, para acceder a el deberemos hacer en la consola:

```
maximi89@SistemasBDmaximi89 ~ $ su root
Contraseña:
SistemasBDmaximi89 maximi89 # su postgres
postgres@SistemasBDmaximi89 /home/maximi89 $
```

Como se darán cuenta, accedí desde mi usuario que tiene privilegios para acceder a “root”, usando “su root”, para luego acceder al usuario Unix “postgres”, de donde podremos realizar todas las operaciones a la B.D. PGSQL. En caso que uses otro sistema *nix es probable que sólo puedas acceder usando “sudo su”:

```
maximi89@SistemasBDmaximi89 ~ $ sudo su
[sudo] password for maximi89:
SistemasBDmaximi89 maximi89 #
```

Valores para perfiles de usuarios, otorgar o quitar funciones a un ROL

-d, --createdb	el rol podrá crear bases de datos
-D, --no-createdb	el rol no podrá crear bases de datos (predeterm.)
-e, --echo	mostrar las órdenes a medida que se ejecutan
-E, --encrypted	almacenar la contraseña cifrada
-i, --inherit	el rol heredará los privilegios de los roles de los cuales es miembro (predeterminado)
-I, --no-inherit	rol no heredará privilegios
-l, --login	el rol podrá conectarse (predeterminado)
-L, --no-login	el rol no podrá conectarse
-N, --unencrypted	almacenar la contraseña sin cifrar
-P, --pwprompt	asignar una contraseña al nuevo rol
-r, --createrole	el rol podrá crear otros roles

-R, --no-createrole	el rol no podrá crear otros roles (predeterminado)
-s, --superuser	el rol será un superusuario
-S, --no-superuser	el rol no será un superusuario (predeterminado)
--interactive	preguntar los nombres y atributos de rol faltantes en lugar de asumir los valores por omisión
--replication	el rol podrá iniciar replicación
--no-replication	el rol no podrá iniciar replicación

Ejemplo de creación de un perfil administrador con clave, sería:

```
createuser -s administradorsinclave
createuser -s -P administradorconclave
createuser -P usuario
createuser -P -d usuario
createuser -L grupoALectura
```

Eliminando un usuario de PostGreSQL

Desde la **shell**:

Si queremos borrar un usuario, haremos:

```
dropuser administrador
```

Si queremos borrar una BD, haremos:

```
dropdb mibd
```

Para verificar que funciona y se ha cargado correctamente postgis haremos:

```
select version();
```

```
PostgreSQL 8.4.13 on x86_64-redhat-linux-gnu, compiled by GCC gcc (GCC) 4.4.6 20120305 (Red
Hat 4.4.6-4), 64-bit
(1 row)
```

```
select postgis_full_version();
```

```
POSTGIS="1.5.3" GEOS="3.3.2-CAPI-1.7.2" PROJ="Rel. 4.7.1, 23 September 2009"
LIBXML="2.7.6" USE_STATS
(1 row)
```

```
select ST_Point(1, 2) AS MyFirstPoint;
```

```
select ST_SetSRID(ST_Point(-77.036548, 38.895108),4326); -- using WGS 84
```

Actualmente lo he probado con Quantum 1.7.5 que ha funcionado sin problemas, sin embargo GvSIG no ha funcionado con la configuración que hemos usado, ya que la aplicación me tira un error al cargar los datos.

[cant write gvgis postgis](#)

O algo por el estilo...

Configurando PG_HBA.CONF para permitir conexiones TCP al servidor PGSQL

Para permitir conexiones a los servidores donde se aloja PostgreSQL debemos modificar ciertos parámetros en el archivo pg_hba.conf que está ubicado en:

LinuxMint (Probablemente también en Ubuntu)

/etc/postgresql/9.2/main/pg_hba.conf

CentOS

/var/lib/pgsql/data/pg_hba.conf

Explicación sobre algunos métodos:

```
# METHOD can be "trust", "reject", "md5", "password", "gss", "sspi",  
# "krb5", "ident", "peer", "pam", "ldap", "radius" or "cert". Note that  
# "password" sends passwords in clear text; "md5" is preferred since  
# it sends encrypted passwords.
```

Tal como dice el ejemplo:

```
# local   DATABASE USER METHOD [OPTIONS]  
# host    DATABASE USER ADDRESS METHOD [OPTIONS]  
# hostssl DATABASE USER ADDRESS METHOD [OPTIONS]  
# hostnossl DATABASE USER ADDRESS METHOD [OPTIONS]
```

Ejemplo práctico:

```
host all all 192.10.10.0/0 password
```

NOTA: Tal como se explica en “Explicación sobre algunos métodos”, el modo PASSWORD envía la clave en texto plano, en cambio MD5 envía la clave en modo encriptado, aún así MD5 se comprobó hace algún tiempo que es vulnerable por lo que se recomienda seguir buscando otros métodos más seguros...

Inicio automatizado de PostgreSQL

Para iniciar automáticamente el servicio de PostgreSQL se debe ingresar como administrador con privilegios de super vaca (root) o sudo (si está configurado).

```
chkconfig postgresql --level 35 onCortafuegos e IP estática
```

Cortafuegos

Editamos /etc/rc.local agregando nuestra regla al pie del archivo:

```
[root@postgresql ~]# cat /etc/rc.local
#!/bin/sh
#
# This script will be executed *after* all the other init scripts.
# You can put your own initialization stuff in here if you don't
# want to do the full Sys V style init stuff.

touch /var/lock/subsys/local

iptables -I INPUT -p tcp --dport 5432 -j ACCEPT
```

IP estática

Asignar ruta por defecto al iniciar:

```
[root@postgresql ~]# cat /etc/sysconfig/network
NETWORKING=yes
HOSTNAME=postgresql.localdomain
GATEWAY=192.10.10.150
```

Asignar IP estática para la tarjeta de red eth0:

```
[root@postgresql ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=eth0
HWADDR=00:19:D1:82:EC:05
```



```
TYPE=Ethernet
UUID=fd0e1482-2939-42a2-aeef-b63dfb229532
ONBOOT=yes
NM_CONTROLLED=yes
BOOTPROTO=static
IPADDR=192.10.10.222
NETMASK=255.255.255.0
```

Transferencia de archivos con SCP

Descargando datos con SCP

SCP usa túneles encriptados para transferencia de datos, aplica la misma seguridad que SSH.

NOTA: scp copies files between hosts on a network. It uses ssh(1) for data transfer, and uses the same authentication and provides the same security as ssh(1). Unlike rcp(1), scp will ask for passwords or passphrases if they are needed for authentication.

```
[root@postgresl ~]# scp maximi89@192.168.1.1:/home/maximi89/Descargas/propiedades* .
maximi89@192.168.1.1's password:
```

propiedades_04_2012.dbf	100%	10MB	9.8MB/s	00:01
propiedades_04_2012.dbf.tar.gz	100%	14MB	14.0MB/s	00:01
propiedades_04_2012.prj	100%	409	0.4KB/s	00:00
propiedades_04_2012.rar	100%	13MB	13.0MB/s	00:01
propiedades_04_2012.shp	100%	22MB	7.4MB/s	00:03
propiedades_04_2012.shx	100%	238KB	238.1KB/s	00:00

Subiendo archivos con SCP

```
scp /var/www/taekwondo_canchas/* taekwondo@192.168.1.5:/home/taekwondo/html/canchastkd
```

*: significa que todo el contenido que pueda extraer que no sean carpetas pasarán inmediatamente, caso contrario usaremos la opción “-r” para hacerlo recursivo, osea que incluya carpetas.

```
scp /root/propiedades_04_2012-prueba_esquema_tabla.sql root@192.168.1.1:/home/maximi89/
```

```
scp root@192.168.1.1:/root/propiedades_04_2012-prueba_esquema_tabla.sql /home/maximi89/
```

Ambas formas sirven, sólo debemos cambiar de posición el “[root@192.168.1.1:](#)” ya que esto controla la función de origen o destino en donde debemos autenticar nuestros usuarios.

Transformar archivos SHP y DBF de ArcGis de la compañía ESRI a SQL:

Para transformar los archivos de capas “shape” haremos un:

```
[root@postgresql ~]# shp2pgsql propiedades_04_2012.shp barbara_casas.propiedades_04_2012 > propiedades_04_2012-esquema.sql
```

Subir archivo SQL al servidor:

```
[root@postgresql ~]# psql -d bd -f ./propiedades_04_2012.sql -U administra -W
```

Opciones para el comando SHP2PGSQL

-W, te permite cambiar la codificación (cabe recordar que no pudimos cambiar la codificación)

-I, inserta índice geoespacial.

-s, permite insertar el SRID en la tabla AddGeometryColumn()

¿Cómo borrar una tabla de la B.D. evitando problemas al restaurar valores a las tablas?

Volcado de la BD (Respaldos)

PG_DUMP puede generar volcados binarios, en texto plano y comprimidos, pero dependiendo del tipo de respaldo deberá ser usada o con uno u otra herramienta:

PG_DUMP con -F c, deberá ser usada con PG_RESTORE.

PG_DUMP con -F t, deberá ser usada con PG_RESTORE.

PG_DUMP con -F p, es el valor por defecto, es texto plano, por lo tanto se usa con PSQL.

Volcado de la Base de Datos PostGreSQL con PG_DUMP

```
maximi89@SistemasBDmaximi89 ~ $ pg_dump -F c -b -c -C -f volcado7 postgres
```

- -a, --data-only extrae sólo los datos, no el esquema
- -F, --format=c|d|t|p Formato del archivo de salida (c=personalizado, d=directorio, t=tar,

p=texto (por omisión))

- -b, --blobs incluye objetos grandes en la extracción
- -c, --clean tira (drop) la base de datos antes de crearla (incompatible con “-C”)
- -C, --create incluye órdenes para crear la base de datos en la extracción

```
[root@postgresql ~]# pg_dump -f volcado_bd -U administra -W -x -O bd
```

Existen varias opciones de pg_dump, sin embargo, por defecto se extraen los permisos, propietarios y datos de la B.D., los valores más importantes para nosotros son las instrucciones SQL:

- Retornar sin privilegios para X usuario, para esto se usa la opción “-O”.
 - -O, --no-owner en formato de sólo texto, no reestablece los dueños de los objetos
 - -x, --no-privileges no extrae los privilegios (grant/revoke)
- Para retirar sólo el esquema, se usa: “-s” (no es un valor requerido).
- Para retirar sólo los datos, se usa: “-a” (no es un valor requerido).
- -U, usuario.
- -W, con clave.

El último valor de la línea indica cual Base de Datos se revisará.

Extraer esquema, datos y propietarios de tablas:

```
pg_dump -f volcado_bd -U administra -W -s bd
```

Extraer sólo datos:

```
pg_dump -f volcado_bd -U administra -W -a bd
```

Extraer sólo el esquema:

```
pg_dump -f volcado_bd -U administra -W -s bd
```

Volcando todas las Bases de Datos con PG_DUMPALL

Habiendo varias opciones más, que pueden revisar más adelante, si queremos extraer todas las B.D. con todos los datos, entonces haremos un:

```
postgres@SistemasBDmaximi89 ~ $ pg_dumpall -f salida.sql
```

Esto producirá una salida de la B.D. completamente en texto plano en el archivo llamado “salida.sql”.

```
postgres@SistemasBDmaximi89 ~ $ pg_dumpall -f salida.sql -l postgres -c
```

Eso producirá un volcado en el archivo “salida.sql” (-f) pero conteniendo sólo los datos de la B.D. “postgres” (-l) y ejecutando un “drop” (-c) antes de volver a crear la B.D. respaldada.

Volcando usuarios del sistema de Base de Datos PostgreSQL

SistemasBDmaximi89 maximi89 # su postgres

postgres@SistemasBDmaximi89 /home/maximi89 \$ pg_dumpall -r

```
--
-- PostgreSQL database cluster dump
--

SET client_encoding = 'UTF8';
SET standard_conforming_strings = on;

--
-- Roles
--

CREATE ROLE maximi89;
ALTER ROLE maximi89 WITH NOSUPERUSER INHERIT NOCREATEROLE NOCREATEDB
LOGIN NOREPLICATION;
CREATE ROLE maximi891;
ALTER ROLE maximi891 WITH NOSUPERUSER INHERIT NOCREATEROLE NOCREATEDB
LOGIN NOREPLICATION;
CREATE ROLE maximi892;
ALTER ROLE maximi892 WITH NOSUPERUSER INHERIT NOCREATEROLE NOCREATEDB
LOGIN NOREPLICATION PASSWORD 'md597b11f5aeafbbecd8fa9291053490f75';
CREATE ROLE postgres;
ALTER ROLE postgres WITH SUPERUSER INHERIT CREATEROLE CREATEDB LOGIN
REPLICATION;

--
-- PostgreSQL database cluster dump complete
--
```

Opciones del comando PG_DUMPALL para volcados

Existen más comandos, pero estos son algunos de los más importantes a mí parecer:

Opciones generales:

-f, --file=ARCHIVO nombre del archivo de salida

Opciones que controlan el contenido de la salida:

-a, --data-only extrae sólo los datos, no el esquema
-c, --clean tira (drop) la base de datos antes de crearla
-g, --globals-only extrae sólo los objetos globales, no bases de datos
-o, --oids incluye OIDs en la extracción
-O, --no-owner no reestablece los dueños de los objetos
-r, --roles-only extrae sólo los roles, no bases de datos
 ni tablespaces
-s, --schema-only extrae sólo el esquema, no los datos

Opciones de la conexión:

-h, --host=ANFITRIÓN anfitrión de la base de datos o
 directorio del enchufe (socket)
-l, --database=NOMBRE especifica la base de datos a la cual conectarse
-p, --port=PUERTO número del puerto de la base de datos
-U, --username=USUARIO nombre de usuario con el cual conectarse
-w, --no-password nunca pedir una contraseña
-W, --password fuerza un prompt para la contraseña
 (debería ser automático)

Realizando respaldo con PG_BASEBACKUP (granja de servidores)

Este al parecer está disponible desde PostgreSQL 9.1 en adelante, ya que se usa con las funciones de respaldo base en granjas de servidores o clúster.

<http://www.postgresql.org/docs/current/static/app-pgbasebackup.html>

¿Cómo subir un respaldo de la Base de Datos?

Subiendo respaldos con PSQL

```
maximi89@SistemasBDmaximi89 ~ $ psql -d bd -U administra -W -f /home/maximi89/volcado3_bd -h 192.168.1.2
```

- -d, indica cual Base de Datos a usar.
- -U, usuario.
- -W, con clave.
- -f, archivo con el respaldo de la B.D., osea tablas y datos.

NOTA: la BD debe existir y no deben repetirse los datos.

MERGE

Existen otras opciones para subir una BD habiendo contenido en las tablas.

MERGE que viene de SQL:2003 y SQL:2008.

https://wiki.postgresql.org/wiki/SQL_MERGE

Restaurando la B.D. con PG_RESTORE

Otra manera es con PG_RESTORE, el cual permite subir o el esquema, o los datos o ambas cosas.

Habiendo una condición para su uso, esta herramienta sólo sirve si se ha aplicado PG_DUMP para generar el volcado, ya que no encontré manera de hacerlo con PG_DUMPALL, porque este último genera un archivo que debe ser usado con PSQL y PG_DUMP dependiendo del formato de salida si es binario PG_DUMP -F c o PG_DUMP -F t, deben usar PG_RESTORE, finalmente PG_DUMP -F p debe usar PSQL (texto plano).

PG_CATALOG.PG_AUTHID: contiene los datos de las cuentas y sus respectivas claves.

SistemasBDmaximi89 maximi89 # su postgres

postgres@SistemasBDmaximi89 /home/maximi89 \$ pg_restore -U administrador -W -h 192.10.10.222
volcadobinario_tar

postgres@SistemasBDmaximi89 /home/maximi89 \$ pg_restore -d maxi -v -F c prueba4

Revisar si sucede con diferentes permisos de usuario en las diferentes tablas.

RELACIONES DE LAS TABLAS

CONTENIDOS y ESQUEMAS

Usuarios con PG_IDENT.CONF

POR COMPLETAR