Cómo instalar PostgreSQL en Ubuntu 10.10

Acabo de incorporarme a un proyecto en **Rails** en el que usan **PostgreSQL**, y como recién formateé y le puse **Ubuntu** **9.04** a la laptop, necesito ahora instalar este manejador de base de datos.

Y es un buen pretexto para por fin escribir la guía de cómo instalarlo en **Ubuntu**.

Como siempre, vamos a la terminal (**Aplicaciones → Accesorios → Terminal**) y tecleamos:

```
1 sudo apt-get install postgresql postgresql-client postgresql-contrib libpq-dev pgadmin3
```

Eso instala el cliente y servidor de la base de datos, algunos scripts de utilería y la aplicación **pgAdmin** para administrar la base de datos. El paquete de **libpq-dev** nos servirá para poder compilar la gema **Ruby** de **PostgreSQL** más adelante.

Confirmemos que la instalación terminó adecuadamente teclando:

```
1 psql --version
```

En mi caso la respuesta fue:
psql (PostgreSQL) 8.4.7

Cambiar la contraseña del usuario administrador

Ahora necesitamos establecer la contraseña del usuario administrador **postgres**. Teclea la siguiente línea en la terminal (cambia la palabra *password* por la contraseña que desees usar):

```
1 sudo su postgres -c psql
2 ALTER USER postgres WITH PASSWORD 'password';
3 \q
```

Eso altera la contraseña dentro de la base de datos, ahora necesitamos hacer lo mismo para el usuario Linux **postgres**:

```
1 sudo passwd -d postgres
2 sudo su postgres -c passwd
```

Te aparecerá un *prompt*, introduce la misma contraseña que pusiste antes.

Poner a punto pgAdmin

Listo, de ahora en adelante podemos usar **pgAdmin** o la terminal para administrar nuestra base de datos como el usuario **postgres**. Pero antes de que te metas a **pgAdmin** deberías configurar el **PostgreSQL Admin Pack**, que te permite llevar un mejor registro y monitoreo de tu base de datos.

Ejecuta lo siguiente desde la línea de comandos en tu terminal:

```
1 sudo su postgres -c psql < /usr/share/postgresql/8.4/contrib/adminpack.sql
```

Para ejecutar **pgAdmin** ve a tu menú de aplicaciones:
Aplicaciones → Programación → pgAdmin III

Cambiar el esquema de autentificación de PostgreSQL

Al ejecutar algunos comandos de base de datos, es posible que te encuentres con un error que dice algo como:

FATAL: la autentificación Ident falló para el usuario «x»

Para evitarlo necesitas editar el archivo **/etc/postgresql/8.4/main/pg_hba.conf** y cambiar el esquema de autentificación. Abre el archivo con privilegios de *root*:

```
1 sudo gedit /etc/postgresql/8.4/main/pg_hba.conf
```

Y cambia esto:

```
1 # "local" is for Unix domain socket connections only
2 local all all ident
```

Por:

```
1 # "local" is for Unix domain socket connections only
2 local all all md5
```

Reinicia el servidor de **PostgreSQL** tecleando en tu terminal:

```
1 sudo /etc/init.d/postgresql restart
```

PostgreSQL Ruby gem

Si planeas usar **PostgreSQL** dentro de **Ruby**, necesitarás esto:

```
1 gem install pg
```

Si deseas instalar la gema desde código fuente, sigue estas instrucciones.

En **Rails 3**, puedes crear una aplicación configurada para usar **PostgreSQL** con este comando:

```
1 rails new mi-nueva-aplicacion -d postgresql
```

Referencia: http://lobotuerto.com/blog/2009/07/20/como-instalar-postgresql-en-ubuntu-9-04-jaunty-jackalope/

Instalar PostgreSQL y pgAdmin en Ubuntu

Vamos a aprender como instalar el manejador de base de datos postgres, asi como tambien la herramienta grafica pgAdmin, que nos permite gestionar de manera sencilla nuestras bases de datos.

Primero necesitamo instalar los siguientes paquetes, ya sea desde consola o el gestor de paquetes Synaptic:

- **postgresql.**
- **pgadmin3.**

Luego que tenemos los paquetes instalador lo primero que necesitamos hacer es crear nuestro usuario en postgres, para hacerlo, en un terminal ejecutamos los comandos:

sudo bash

su postgres

createuser oftc007

El nombre de usuario “oftc007″ es para efectos de ejemplo, pero ustedes colocan el nombre de usuario que quieran. Luego que ejecutemos el comando nos aparece el siguiente mensaje en el terminal:

¿Será el nuevo rol un superusuario? (s/n)

Escribimos una S y damos en enter. El terminal debe devolver “CREATE ROLE” si el usuario fue creado correctamente.

Luego que tenemos un usuario creado, vamos a crear una base de datos llamada “prueba”, para hacerlo en un terminal ejecutamos:

createdb prueba

La consola devuelve “CREATE DATABASE” si la base de datos fue creada correctamente.

Para entrar a nuestra base escribimos:

psql prueba

En nuestro caso “prueba” es el nombre de la base de datos. La salida de este comando es algo como lo siguiente:

Bienvenido a psql 8.2.6, la terminal interactiva de PostgreSQL.

Digite: \copyright para ver los términos de distribución
\h para ayuda de comandos SQL
\? para ayuda de comandos psql
\g o or termine con punto y coma para ejecutar una consulta
\q para salir

prueba=#

Esto nos indica que estamos dentro de nuestra base de datos, por lo cual ya podemos ejecutar cualquier comando SQL en nuestra BD. Por razones de seguridad vamos a cambiar el password de nuestro usuario, para eso ejecutamos el siguiente comando dentro de nuestra BD:

alter user oftc007 with password 'password';

Donde deben reemplazar "oftc007″ por su nombre de usuario y las palabra "password" que se encuentra dentro de las comillas simples por la clave que desean para su cuenta. Es importante saber que la clave siempre debe ir dentro de comillas simple, y se debe colocar un punto y coma al final del comando para que pueda ser ejecutado una vez que damos en enter. Si el comando se ejecuto correctamente la consola nos devuelve "ALTER ROLE".

El comando para salir de nuestra base de datos es:

\q

Ahora vamos a ver nuestra base de datos desde pgAdmin y aprender cosas basicas de su uso y algunas de sus configuraciones.

Para ejecutar pgAdmin en la barra del menu del panel vamos a:

Aplicaciones -> Herramientas del sistema -> pgAdmin III

Lo primero es revisar si tenemos el pgAdmin en español, si lo tenemos en ingles, vamos a:

File -> Options..

En "User language" seleccionamos "(es_ES) Spanish" y hacemos click en "Ok". Nos aparece un mensaje, le damos en aceptar y reiniciamos el pgAdmin.

Ahora para conectarnos a nuestra base de datos vamos a:

Archivo -> Añadir Servidor…

Nos aparece una ventana en la cual necesitamos rellenar unos campos, vamos a explicar lo que debemos colocar en cada uno:

- **Dirección**: Aqui colocamos el servidor donde se encuentra la Base de Datos, en nuestro caso como estamos trabajando de manera local colocamos "localhos" (sin comillas). En caso de querer conectarnos a una Base de Datos en otro equipo, en este campo debemos colocar la direccion IP del computador donde esta la base de datos.
- **Descripción**: Aqui colocaremos el nombre de la conexion, aqui pondremos el nombre de nuestro gusto. Por ejemplo: Mis Bases de Datos.
- **Servicio/Puerto/SSL/BD de Mantenimiento**: Estos campos los dejamos como estan.
- **Nombre de usuario**: Colocamos el nombre de usuario que acabos de crear.
- **Contraseña**: Colocamos la contraseña que le asignamos a nuestra cuenta que acabamos de crear.

Hacemos click en aceptar.

Luego de esto veremos que aparece un servidor en esta lista, que es el que acabamos de agregar, y en el cual esta nuestra base de datos de prueba. Para ver las tablas hacemos doble click sobre nuestro servidor, luego sobre base de datos y asi veremos todas nuestras bases de datos creadas, en nuestro caso solo aparecera la BD prueba. Hacemos doble click sobre ella, luego sobres esquemas y por ultimo en public.

En este punto ya podemos ver todos los elementos que conforman nuestra BD como tablas, vistas,

store procedures, etc.

Habilitar conexiones remotas en PostgreSQL

Una conexion remota no es mas que podernos conectar a nuestra base de datos desde otro equipo. Esta es una funcion que por lo general necesitaremos, ya que si estamo desarrollando una aplicacion que se conecta a una base de datos, habilitar esta funcion y colocar nuestra base de datos en un solo equipo, nos permitira probar la concurrencia de nuestra base de datos.

Por motivos de seguridad postgres no tiene activo el aceptar conexiones remotas, en esta parte del tutoria aprenderemos como habilitar esta funcion.

Primero necesitamos editar el archivo postgresql.conf, para abrir el archivo en un terminal ejecutamos el comando:

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

Una vez que abrimos el archivo ubicamos la siguiente linea:

#listen_addresses = 'localhost'

Subtituimos esa linea por la siguiente:

listen_addresses = ''*

Luego buscamos la siguiente linea y la descomentamos:

#password_encryption = on

Para descomentarla, le quitamos el # al principio de la linea, quedandonos:

password_encryption = on

Guardamos y cerramos el archivo, para posteriormente reinciar el postgres, ejecutando el siguien comando en un terminal:

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

Luego de esto necesitamos configurar la lista de acceso, la cual permite establecer relaciones de confianza para ciertos equipos y redes. Para hacer esto tenemos que editar el fichero pg_hba.conf. Para abrir el archivo, en un terminal ejecutamos el comando:

sudo gedit /etc/postgresql/8.2/main/pg_hba.conf

Al final del archivo debemos agregar lo siguiente:

host all all 0.0.0.0 0.0.0.0 md5

De este modo cualquier con previa autentificacion puede conectarse al postgres de nuestra equipo, o nosotros conectarnos a dicho equipo.

Guardamos y cerramos el archivo, para por ultimo reiniciar el postgresql y que los cambios hagan efecto.

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

Eso es todo para este tutorial, en unos dias estare publicando un tutorial de como hacer la conexion entre Java y PostgreSQL.

Si quieren mas informacion o conocer mas a fondo PostgreSQL visiten el siguiente link.

Referencia: http://linpox.wordpress.com/2008/04/10/instalar-postgresql-y-pgadmin-en-ubuntu/