



HEROKU POSTGRES

DWS
CFGs DAW

Francisco Aldarias Raya
paco.aldarias@ceedcv.es

2019/2020

Fecha 17/12/19

Versión:191217.1905

Licencia



Reconocimiento - NoComercial - CompartirIgual (by-nc-sa): No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

Nomenclatura

A lo largo de este tema se utilizarán distintos símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:



Importante



Atención



Interesante

Revisiones

18/11/2019. Conexión remota.

26/11/2019. Configurar pgadmin3 con servidor remoto.

17/12/2019. Apartado 5.6 Página 16. Dataclip. Botón share.

ÍNDICE DE CONTENIDO

1.Introducción.....	4
1.1 Heroku postgres.....	4
2.Provisionando heroku postgres.....	5
3.Instalación de postgres en local.....	6
3.1 Instalación de postgres en local con linux.....	7
4.Usando el cli.....	7
4.1 pg:info.....	7
4.2 Pg:sql.....	8
4.3 pg:pull.....	9
4.4 pg:push.....	9
5.Instalando postgres y pgadmin3.....	10
5.1 Instalar postgresql en local.....	10
5.2 Instalar pgadmin en local.....	10
5.3 Configurar pgadmin3 con servidor remoto.....	10
5.4 Creación de tablas.....	14
5.5 Ventana SQL.....	15
5.6 Dataclips.....	16
6.Exportando desde heroku postgres.....	17
6.1 Exportando.....	17
6.2 Restaurando.....	17
7.Importando desde heroku postgres.....	17
7.1 Crear el dump file.....	17
7.2 Importar a Heroku Postgres.....	17
8.Conectar pgadmin3 con el servidor de postgresql local.....	18
9.Bibliografía.....	20

HEROKU POSTGRES

1. INTRODUCCIÓN

Este documento pretende explicar como trabajar con el servidor web heroku para que se pueda configurar el servidor de base de datos postgres.

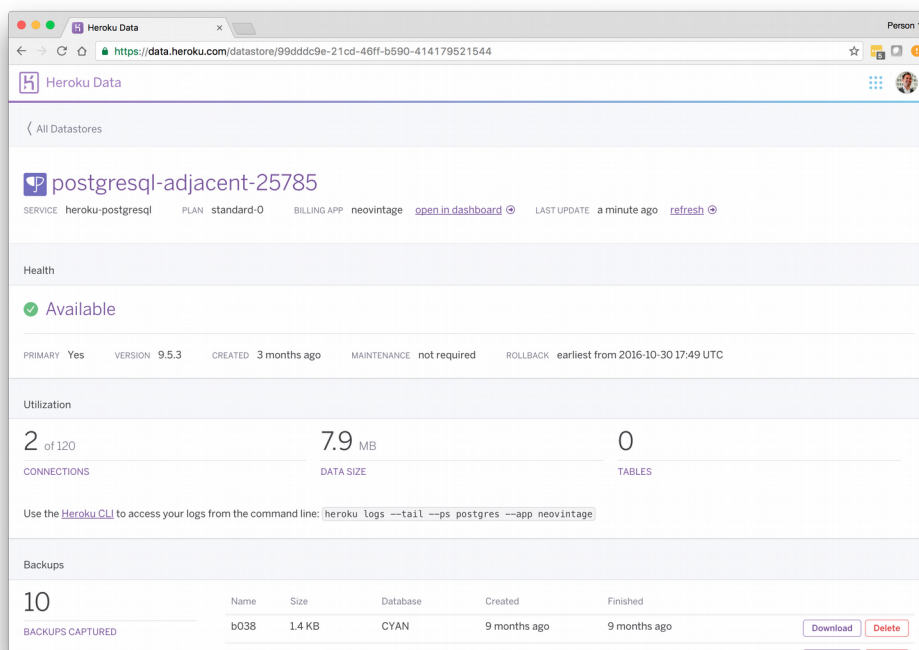
Se parte de que ya se tiene cuenta en el servidor web heroku. Su web es:

<https://dashboard.heroku.com/apps>



1.1 Heroku postgres

Heroku postgres es un Sistema de Gestión de Base de datos SQL proporcionado directamente por Heroku. Se puede acceder a las bases de datos de Postgres desde cualquier lenguaje con drivers de PostgreSQL, incluyendo todos los lenguajes soportados por Heroku tales como Node, Java, Php, Python, etc.



Además soporta una variedad de comandos accesibles desde Heroku CLI (aplicación cliente). Heroku Postgres proporciona un tablero web (dashboard web), para compartir consultas con dataclips y otras útiles características.

Crearemos un proyecto en heroku para poder asociar una base de datos. El contenido mínimo se puede clonar de:

```
git clone https://bitbucket.org/pacoaldarias/pacoaldariasheroku.git
```

2. PROVISIONANDO HEROKU POSTGRES

Al proceso de activar postgres en heroku se llama provisionar. Heroku no deja provisionar sino tenemos una app creada a la cual asociarle la base de datos.

Antes de provisionar debemos comprobar si ya lo hicimos con el comando `heroku addons`.

```
paco@pacocasa:~/www/pacoaldariasheroku$ heroku addons
> Warning: heroku update available from 7.21.0 to 7.35.0
No add-ons for app pacoaldariasheroku.
```

Si `heroku-postgresql` no aparece podemos añadirlo con el comando CLI:

```
heroku addons:create heroku-postgresql:<PLAN_NAME>
```

Por ejemplo para provisionar el plan base de datos gratuito debemos usar `hobby-dev`:

```
heroku addons:create heroku-postgresql:hobby_dev
```

Podemos especificar la versión de Postgres incluyendo `--version` en el comando:

```
heroku addons:create heroku-postgresql:hobby-dev --version=9.5
```

```
paco@pacocasa:~/www/pacoaldariasheroku$ heroku addons:create heroku-postgresql:hobby-dev --version=9.5
> Warning: heroku update available from 7.21.0 to 7.35.0
Creating heroku-postgresql:hobby-dev on pacoaldariasheroku... free
Database has been created and is available
! This database is empty. If upgrading, you can transfer
! data from another database with pg:copy
Created postgresql-spherical-83146 as DATABASE_URL
Use heroku addons:docs heroku-postgresql to view documentation
```

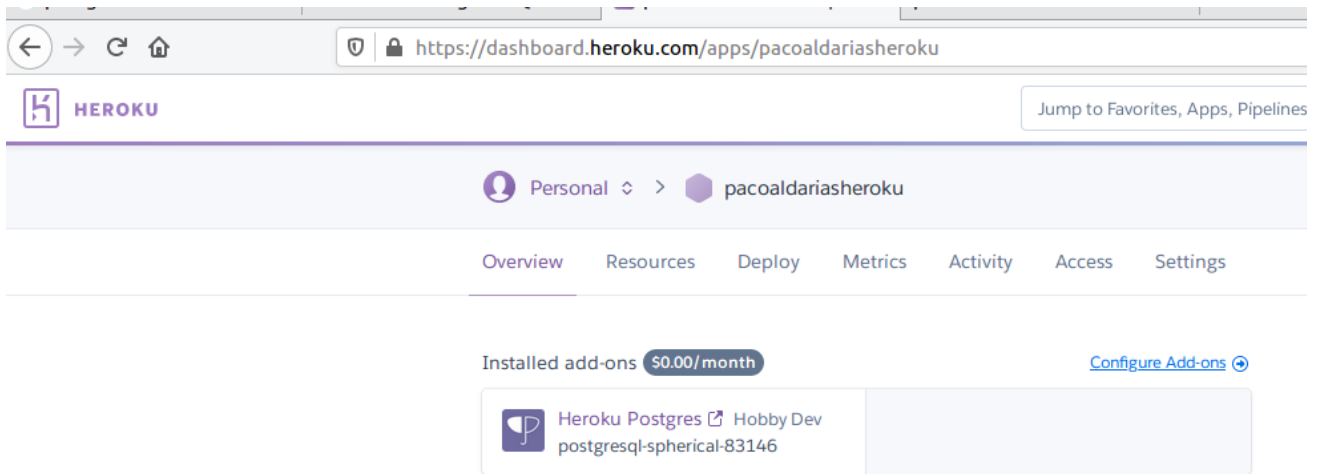
Instalar herramientas adicionales:

```
heroku plugins:install heroku-pg-extras
```

```
warning heroku-pg-extras > execa > cross-spawn-async@2.2.5: cross-spawn no longer requires a
build toolchain, use it instead
```

```
Installing plugin heroku-pg-extras... installed v1.2.1
```

Si entramos a la web de heroku y pulsamos sobre la aplicación nos saldrá que tenemos instalado postgres:



Dependiendo del plan elegido tardara más o menos en estar accesible.

Como parte del proceso de provisión la variable DATABASE_URL es añadida el cual te permite al acceder a la base de datos. Podemos verla con el comando:

```
heroku config
```

```
paco@pacocasa:~/www/pacoaldariasheroku$ heroku config
> Warning: heroku update available from 7.21.0 to 7.35.0
=== pacoaldariasheroku Config Vars
DATABASE_URL: postgres://faujpuhfwfapcf:4f7ef1fd226ea24fde0c
```

En este punto una base de datos PostgreSQL esta provisionada.

3. INSTALACIÓN DE POSTGRES EN LOCAL

La instalación de postgres depende en local del S.O se puede hacer siguiente los pasos:

- Windows:
<https://devcenter.heroku.com/articles/heroku-postgresql#set-up-postgres-on-windows>
- Linux:
<https://devcenter.heroku.com/articles/heroku-postgresql#set-up-postgres-on-linux>
- Mac
<https://devcenter.heroku.com/articles/heroku-postgresql#set-up-postgres-on-mac>

3.1 Instalación de postgres en local con linux

Instalaremos postgres con:

```
sudo apt-get install postgresql
```

Cambiamos al usuario postgres

```
sudo su - postgres
```

Entramos en postgres local:

```
psql
psql (9.5.19)
Type "help" for help.
```

Podemos ver la información de la conexión con el comando: \conninfo

```
postgres=# \conninfo
You are connected to database "postgres" as user "postgres" via socket in "/var/run/postgresql" at port "5432".
```

Crear la base de datos mydb

```
postgres=# CREATE DATABASE mydb;
```

4. USANDO EL CLI

Comando de heroku sobre postgres

4.1 pg:info

Podemos ver información general con el comando:

heroku pg:info

```
paco@pacocasa:~/www/pacoaldariasheroku$ heroku pg:info
> Warning: heroku update available from 7.21.0 to 7.35.0
=== DATABASE_URL
Plan:          Hobby-dev
Status:        Available
Connections:   0/20
PG Version:    9.5.19
Created:       2019-11-14 08:04 UTC
Data Size:     7.6 MB
Tables:        0
Rows:          0/10000 (In compliance) - refreshing
Fork/Follow:   Unsupported
```

```
Rollback:      Unsupported
Continuous Protection: Off
Region:        Europe
Add-on:        postgresql-spherical-83146
```

4.2 Pg:sql

El comando pg:sql es el comando de terminal para conectar y ejecutar consultas a la base de datos.

```
heroku pg:sql
--> Connecting to postgresql-spherical-83146
psql (9.5.19)
SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256,
compression: off)
Type "help" for help.

pacoaldariasheroku::DATABASE=>
```

Ver las tablas de nuestra base de datos con: \d

```
pacoaldariasheroku::DATABASE=> \d
               List of relations
Schema |      Name      |  Type   | Owner
-----+-----+-----+-----
public | departamentos | table   | faujpuhfwfapcf
public | users          | table   | faujpuhfwfapcf
public | users_id_seq    | sequence | faujpuhfwfapcf
(3 rows)

pacoaldariasheroku::DATABASE=> █
```

Para mostrar las columnas de una tabla sería para la tabla departamentos:

```
select column_name, data_type, character_maximum_length, column_default from
INFORMATION_SCHEMA.COLUMNS where table_name = 'departamentos';
```

Saldremos de psql con \q

4.3 pg:pull

Permite traer a postgres local de heroku la base de datos completa. El problema es que hay que borrar la base de datos si existe.

Ejemplo: Se trae la bd postgresql-spherical-83146 mydb de heroku y la coloca en nuestro postgresql local.:

```
dropdb mydb
```

```
heroku pg:pull postgresql-spherical-83146 mydb --app pacoaldariasheroku
```

Requiere que tenga permisos en postgres el usuario que lanza desde linux esta orden.

Daremos permisos al usuario paco asi:

```
paco@pacocasa:~$ sudo passwd postgres
[sudo] password for paco:
Introduzca la nueva contraseña de UNIX:
Vuelva a escribir la nueva contraseña de UNIX:
passwd: contraseña actualizada correctamente
paco@pacocasa:~$ su - postgres
Contraseña:
postgres@pacocasa:~$ createuser --interactive --pwprompt
Enter name of role to add: paco
Enter password for new role:
Enter it again:
Shall the new role be a superuser? (y/n) y
```

Desde el ceed no deja hacer pull por tener el puerto cerrado:

```
paco@pacoceed:~/www/pacoaldariasheroku$ heroku pg:pull postgresql-spherical-83146 mydb --app pacoaldariasheroku
heroku-cli: Pulling postgresql-spherical-83146 --> mydb
pg_dump: [archiver (db)] connection to database "d78jr0cbftec0f" failed: could not connect to server: Expiró el tiempo de conexión
Is the server running on host "ec2-54-228-219-2.eu-west-1.compute.amazonaws.com" (54.228.219.2) and accepting
TCP/IP connections on port 5432?
pg_restore: [custom archiver] could not read from input file: end of file
pg_dump errored with 1
```

La solución sería crear un fichero .sql con las instrucciones a realizar.

4.4 pg:push

Podemos llevar nuestra base de datos local de postgres a heroku postgres con pg:push

Usaremos el comando: pg:push

```
heroku pg:push mydb postgresql-spherical-83146 --app pacoaldariasheroku
```

```
paco@pacocasa:~/www/pacoaldariasheroku$ heroku pg:push mydb postgresql-spherical-83146 --app pacoaldariasheroku
heroku-cli: Pushing mydb --> postgresql-spherical-83146
Remote database is not empty. Please create a new database or use heroku pg:reset
paco@pacocasa:~/www/pacoaldariasheroku$
```

Requiere borrar todo de la base de datos.

```
paco@pacocasa:~/www/pacoaldariasheroku$ heroku pg:reset
> WARNING: Destructive action
> postgresql-spherical-83146 will lose all of its data
> To proceed, type pacoaldariasheroku or re-run this command with --confirm pacoaldariasheroku
>
> pacoaldariasheroku
Resetting postgresql-spherical-83146... done
```

Ahora ya podemos importar a heroku postgres con:

```
heroku pg:push mydb postgresql-spherical-83146 --app pacoaldariasheroku
```

5. INSTALANDO POSTGRES Y PGADMIN3

La instalación de la base de datos postgresql es separada del entorno de trabajo pgadmin3.

5.1 Instalar postgresql en local

Podemos instalarlo descargando por la web:

Web: <https://www.postgresql.org/download/>

Probamos que tenemos instalado postgres :

```
createdb --version
createdb (PostgreSQL) 9.5.19
```

5.2 Instalar pgadmin en local

Podemos instalarlo descargando de la web:

Web: <https://www.pgadmin.org/download/>

Con linux desde la terminal instalaremos pgadmin con:

```
sudo apt install pgadmin3
```

5.3 Configurar pgadmin3 con servidor remoto

Tenemos que ir a web de heroku con nuestro usuario: <https://dashboard.heroku.com/> y consultar nuestros datos de conexión con la base de datos.

Heroku App – Datastores – Setting – Data Base Credential para coger los datos. En mi caso sería:

DATA

Databases > postgresql-spherical-83146

SERVICE heroku-postgresql PLAN hobby-dev BILLING APP pacoaldariasheroku

Overview Durability **Settings** Dataclips

ADMINISTRATION

Database Credentials

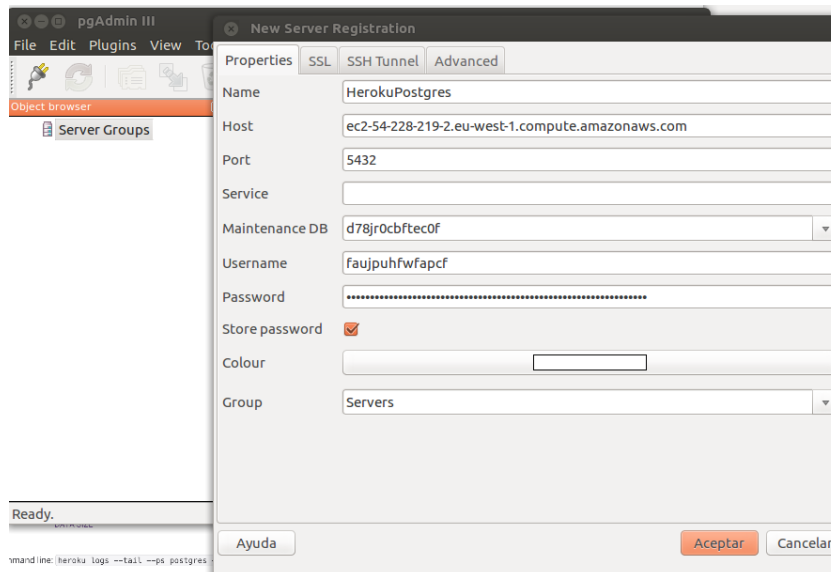
Get credentials for manual connections to this database.

Please note that **these credentials are not permanent**.
Heroku rotates credentials periodically and updates applications where this database is attached.

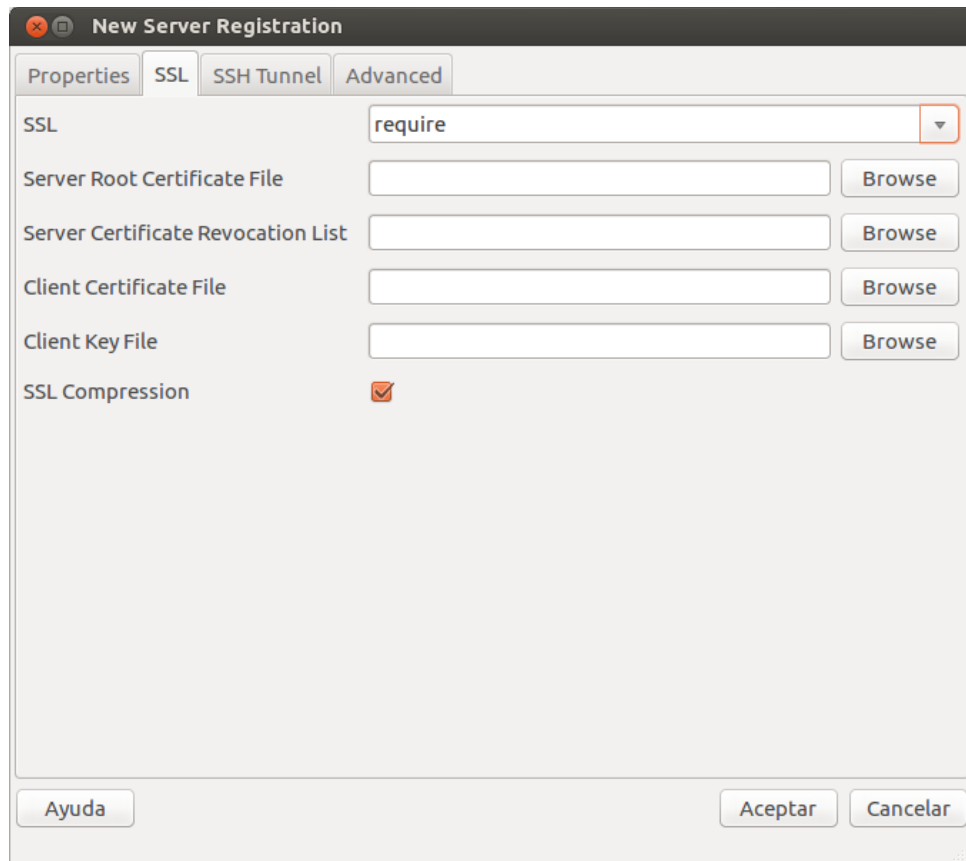
Host	ec2-54-228-219-2.eu-west-1.compute.amazonaws.com
Database	d78jr0cbftec0f
User	faujpuhfwfapcf
Port	5432
Password	4f7ef1fd226ea24fde0df26a5c0425e5c0154571c64b3115920f1e80048
URI	postgres://faujpuhfwfapcf:4f7ef1fd226ea24fde0df26a5c0425e5c0154571c64b3115920f1e80048@ec2-54-228-219-2.eu-west-1.compute.amazonaws.com:5432/postgresql-spherical-83146
Heroku CLI	heroku pg:psql postgresql-spherical-83146 --app pacoaldariasheroku

Rellenar los datos solicitados en pgadmin. Ir al menu: File – New Server

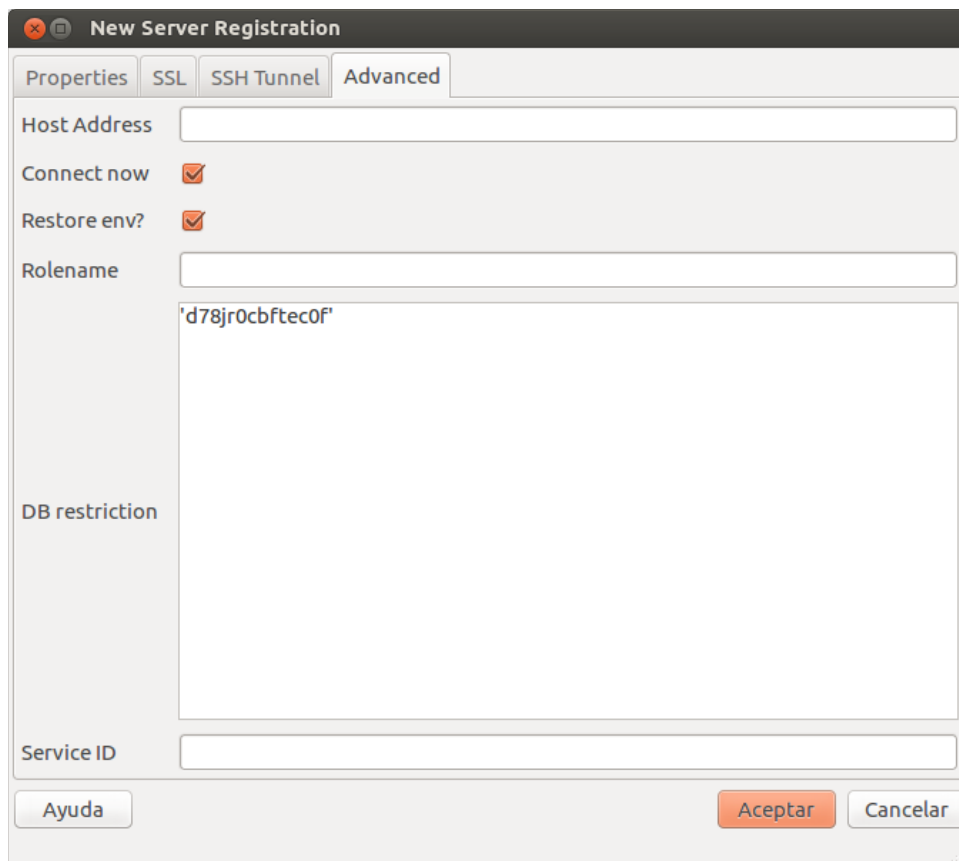
Tab Propiedades:



Tab SSL: require



Tab Advanced – DB description: El mismo valor que Propiedades Maintenance Required con comillas simples, para filtrar nuestra base de datos del resto.

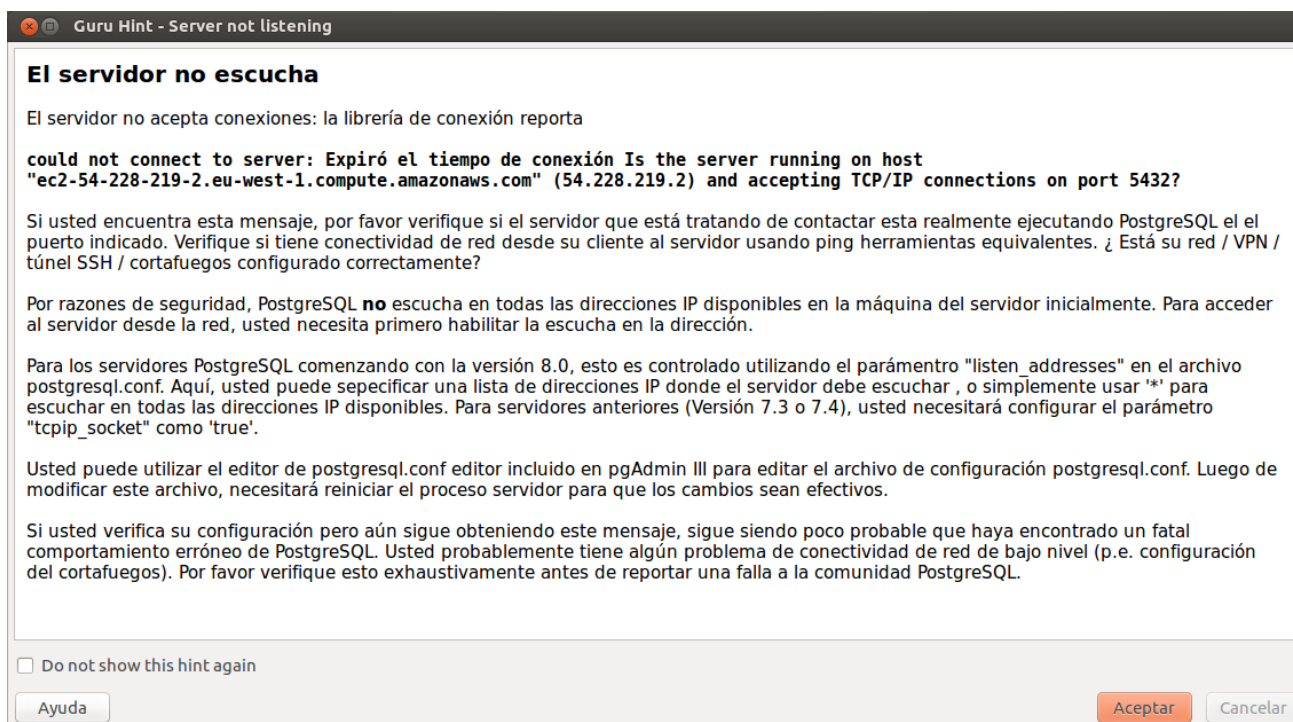


The image shows a 'New Server Registration' dialog box with four tabs: Properties, SSL, SSH Tunnel, and Advanced. The 'Advanced' tab is selected. It contains the following fields and options:

- Host Address:** An empty text input field.
- Connect now:** A checked checkbox.
- Restore env?:** A checked checkbox.
- Rolename:** An empty text input field.
- DB restriction:** A large text area containing the text `'d78jr0cbftec0f'`.
- Service ID:** An empty text input field.

At the bottom, there are three buttons: 'Ayuda' (Help), 'Aceptar' (Accept), and 'Cancelar' (Cancel).

Desde el ceedcv no podemos conectarnos por tener el puerto cerrado:



The image shows a 'Guru Hint - Server not listening' dialog box. It contains the following text:

El servidor no escucha

El servidor no acepta conexiones: la librería de conexión reporta

could not connect to server: Expiró el tiempo de conexión Is the server running on host "ec2-54-228-219-2.eu-west-1.compute.amazonaws.com" (54.228.219.2) and accepting TCP/IP connections on port 5432?

Si usted encuentra esta mensaje, por favor verifique si el servidor que está tratando de contactar esta realmente ejecutando PostgreSQL en el puerto indicado. Verifique si tiene conectividad de red desde su cliente al servidor usando ping herramientas equivalentes. ¿ Está su red / VPN / túnel SSH / cortafuegos configurado correctamente?

Por razones de seguridad, PostgreSQL **no** escucha en todas las direcciones IP disponibles en la máquina del servidor inicialmente. Para acceder al servidor desde la red, usted necesita primero habilitar la escucha en la dirección.

Para los servidores PostgreSQL comenzando con la versión 8.0, esto es controlado utilizando el parámetro "listen_addresses" en el archivo postgresql.conf. Aquí, usted puede sepecificar una lista de direcciones IP donde el servidor debe escuchar , o simplemente usar '*' para escuchar en todas las direcciones IP disponibles. Para servidores anteriores (Versión 7.3 o 7.4), usted necesitará configurar el parámetro "tcpip_socket" como 'true'.

Usted puede utilizar el editor de postgresql.conf editor incluido en pgAdmin III para editar el archivo de configuración postgresql.conf. Luego de modificar este archivo, necesitará reiniciar el proceso servidor para que los cambios sean efectivos.

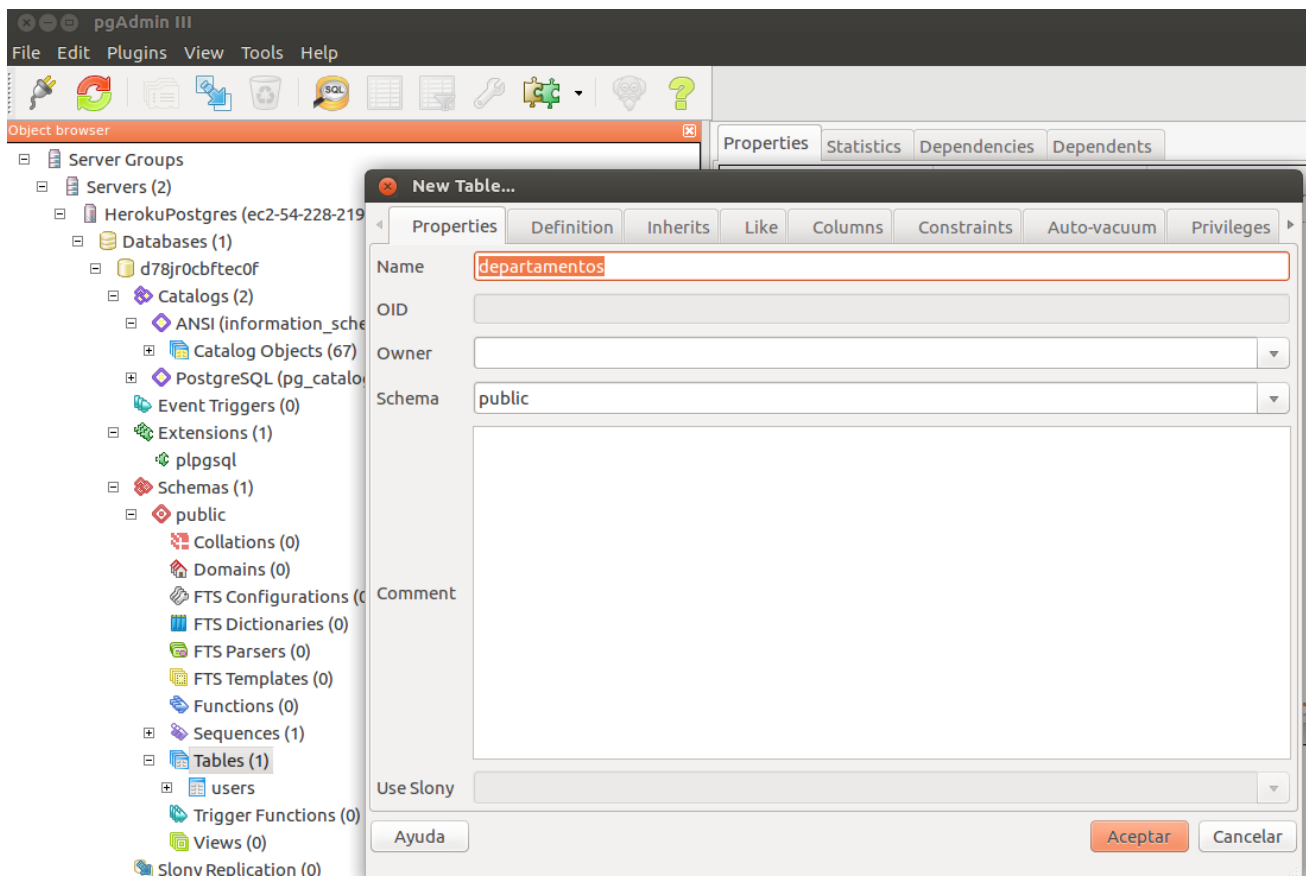
Si usted verifica su configuración pero aún sigue obteniendo este mensaje, sigue siendo poco probable que haya encontrado un fatal comportamiento erróneo de PostgreSQL. Usted probablemente tiene algún problema de conectividad de red de bajo nivel (p.e. configuración del cortafuegos). Por favor verifique esto exhaustivamente antes de reportar una falla a la comunidad PostgreSQL.

☐ Do not show this hint again

At the bottom, there are three buttons: 'Ayuda' (Help), 'Aceptar' (Accept), and 'Cancelar' (Cancel).

5.4 Creación de tablas

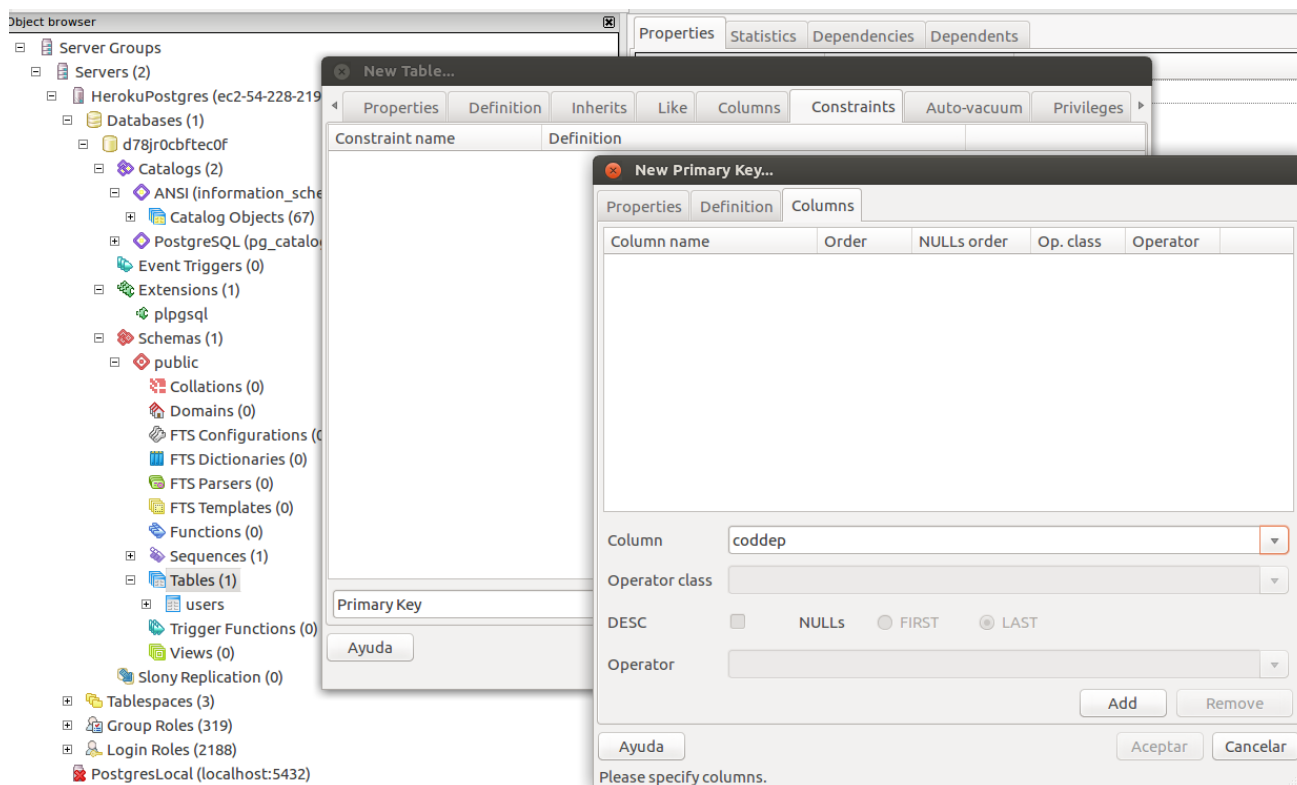
Nos vamos a Databases – Nombre Base Datos – Schemas – Public – Table. Botón derecho New Table.



En Columns añadimos los atributos.

En Definition le asociacion el Table Space pg_default

En constraint añadimos las restricciones como por ejemplo la clave primaria.

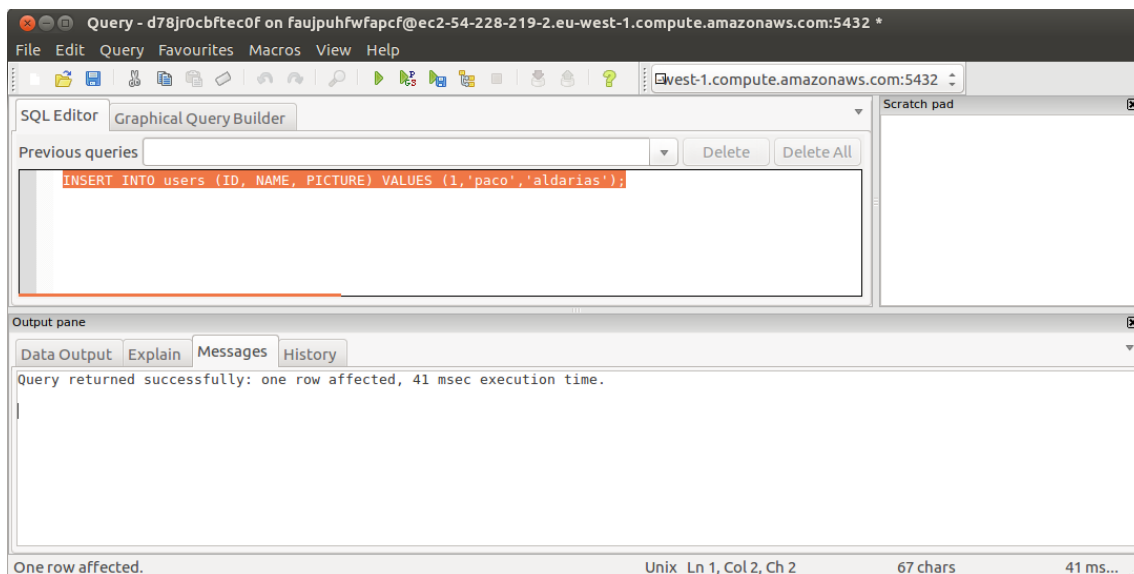


Una vez creada la tabla si pulsamos sobre el nombre nos aparece la sentencia sql.

```
CREATE TABLE public.departamentos
(
  coddep character varying(10) NOT NULL,
  nombre character varying(30) NOT NULL,
  ubicacion character varying(30),
  CONSTRAINT dep_pk PRIMARY KEY (coddep)
);
```

5.5 Ventana SQL

Con la ventana SQL de pgadmin podemos realizar insert en heroku postgres

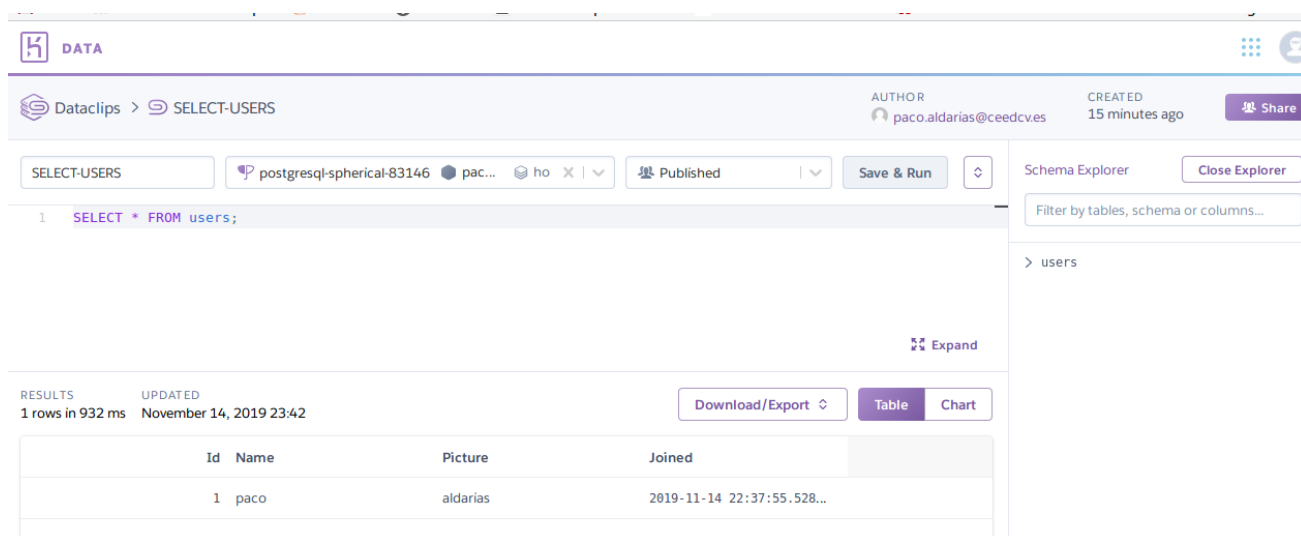


5.6 Dataclips

Heroku postgres dispone de dataclips que permite:

- Tener un asistente para consultas SQL
- Permite guardar las consultas.
- Permite compartirlas via web los data clips guardados. Pulsando el botón **share** genera un enlace que accede a los datos del data clip..

Se pueden ver los resultados en Heroku con un dataclip



Más info en:

<https://devcenter.heroku.com/articles/dataclips>

6. EXPORTANDO DESDE HEROKU POSTGRES

Para traernos datos desde heroku postgres a postgres local, debemos primero exportar de heroku para luego importar del postgres local.

6.1 Exportando

Para exportar los datos de desde Heroku Postgres, crearemos un backup y lo descargaremos.

```
heroku pg:backups:capture  
heroku pg:backups:download
```

Esto descargará el fichero **latest.dump**

6.2 Restaurando

Podemos restaurar en la bd local de postgres.

```
pg_restore --verbose --clean --no-acl --no-owner -h localhost -U myuser -d mydb latest.dump
```

7. IMPORTANDO DESDE HEROKU POSTGRES

7.1 Crear el dump file

Usamos `pg_dump` para comprimir nuestra base de datos local

```
PGPASSWORD=mypassword pg_dump -Fc --no-acl --no-owner -h localhost -U myuser mydb >  
mydb.dump
```

7.2 Importar a Heroku Postgres

Para que PG Backup pueda acceder e importar al fichero dump necesitaremos subirlo a un lugar con url http-accesible. [Recomiendan usar Amazon S3](#) con [signed url](#).

Generate a signed URL using the aws console:

```
aws s3 presign s3://your-bucket-address/your-object
```

Use the raw file URL in the `pg:backups restore` command:

```
heroku pg:backups:restore '<SIGNED URL>' DATABASE_URL
```

Mas información en Heroku <https://devcenter.heroku.com/articles/heroku-postgres-import-export#import>.

8. CONECTAR PGADMIN3 CON EL SERVIDOR DE POSTGRESQL LOCAL

Después de instalar postgresQL lo que necesitamos es conectarnos con un administrador de postgresQL para trabajar con las bases de datos porque desde la línea de comandos es bastante incomodo.

Tenemos que hacer cambios en los ficheros de configuración de postgresQL que están en /etc/postgresql/9.5/main, primero en el fichero postgresql.conf modificando la línea

```
listen_addresses = '*'
```

La descomentamos y ponemos el * si queremos que se conecten al servidor postgresQL desde cualquier sitio o podemos poner la IP desde donde se puede conectar a postgresQL.

```
listen_addresses = '192.168.1.112'
```

Ahora modificaremos el fichero pg_hba.conf y cambiaremos la línea

```
local all postgres md5
```

```
por
```

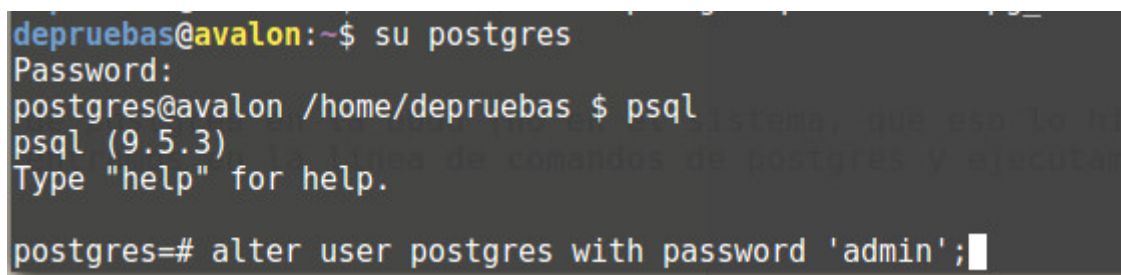
```
local all postgres trust
```

Y por ultimo ponemos al final de todo la línea

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

Ahora, entramos en la línea de comandos de postgres y ejecutamos la siguiente instrucción:

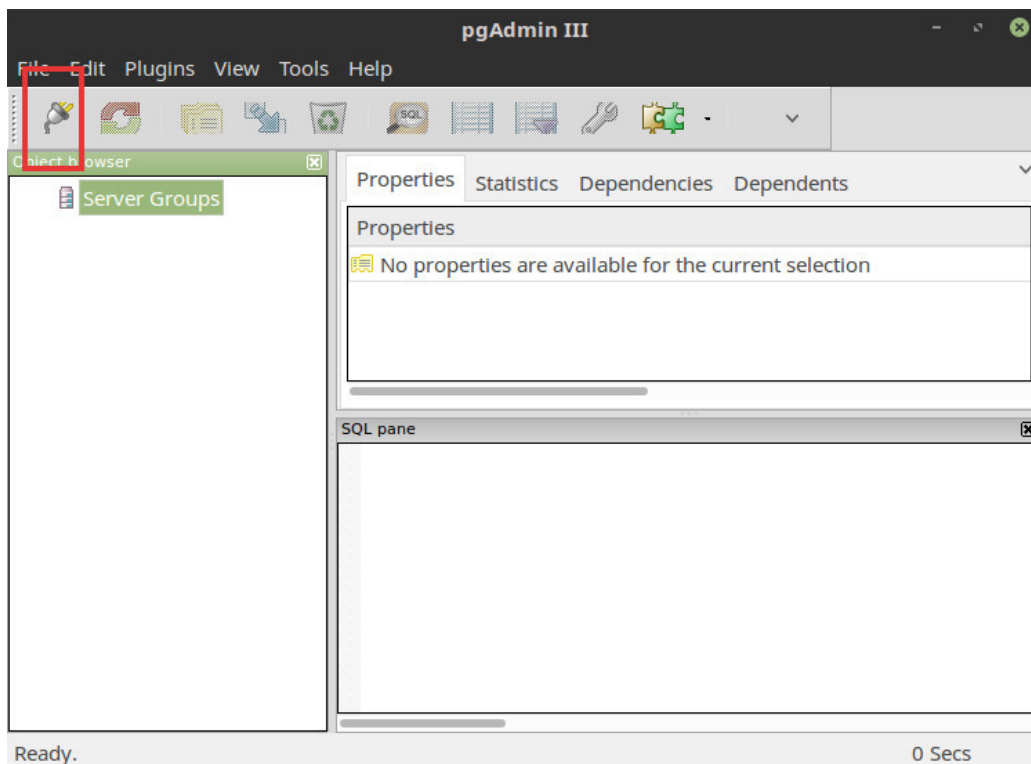
```
alter user postgres with password 'admin';
```



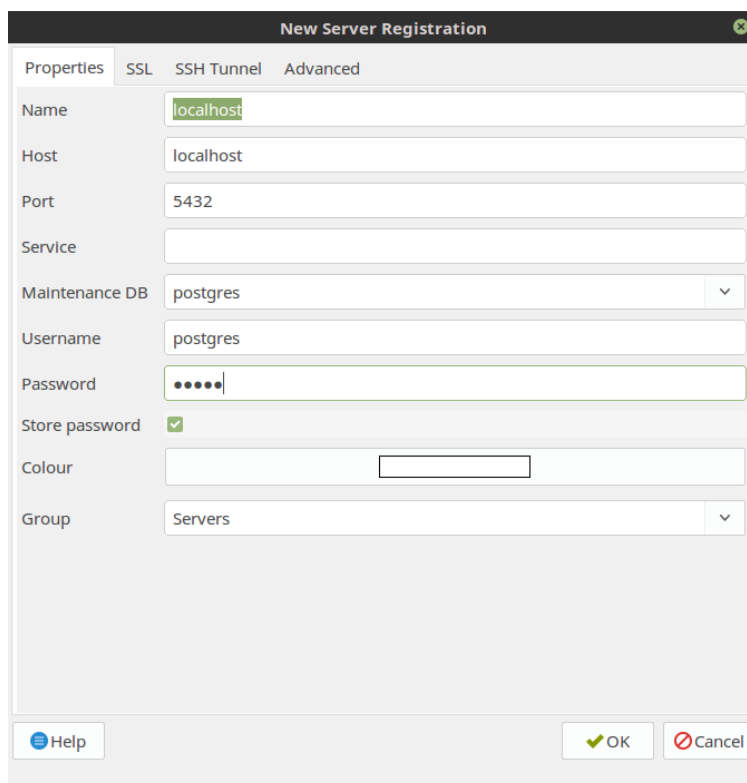
```
depruebas@avalon:~$ su postgres
Password:
postgres@avalon /home/depruebas $ psql
psql (9.5.3)
Type "help" for help.

postgres=# alter user postgres with password 'admin';
```

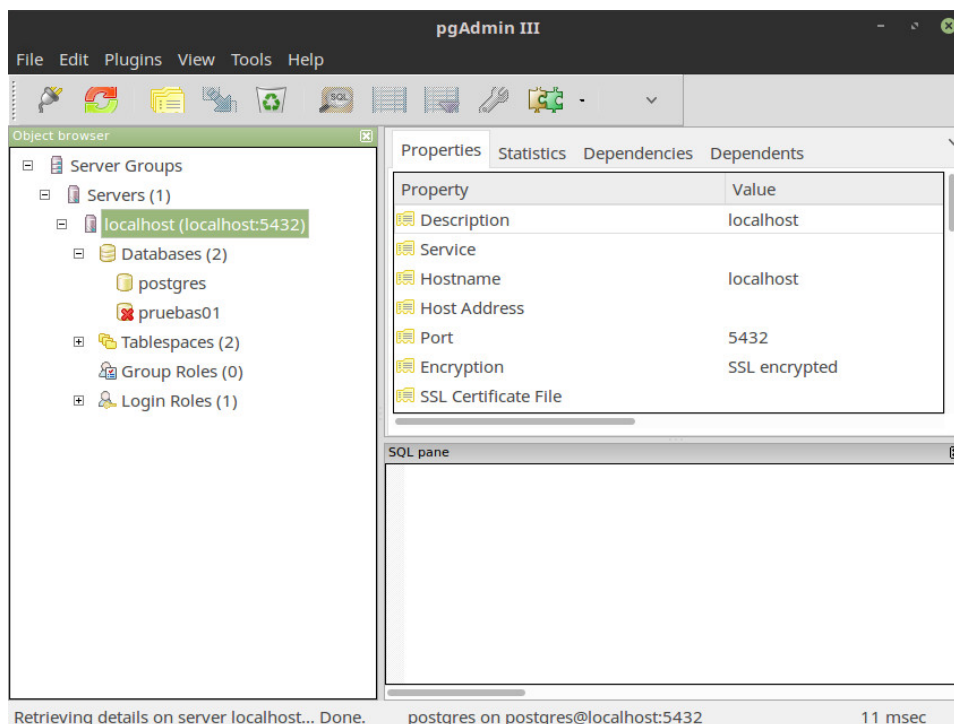
Y ahora y podemos conectar con el pgadmin3



Seleccionamos el enchufe como se ve en la imagen y creamos una nueva conexión con un servidor postgres



Y ya esta



9. BIBLIOGRAFÍA

- Video Heroku Postgress: <https://www.youtube.com/watch?v=KE7swNWvlok>
- Heroku Postgres: <https://devcenter.heroku.com/articles/heroku-postgresql#local-setup>
- Postgress: Instalación y primeros pasos: <https://evilnapsis.com/2016/02/05/guia-postgresql-instalacion-y-primeros-pasos-en-linux/>
- Conectar pgadmin3 con el servidor de postgresQL: <https://www.netveloper.com/conectar-pgadmin3-con-el-servidor-de-postgreSQL>
- How to manage PostgreSQL databases and users from the command line. <https://www.a2hosting.es/kb/developer-corner/postgresql/managing-postgresql-databases-and-users-from-the-command-line>
- Documentación oficial postgresql 9.5. <https://www.postgresql.org/docs/9.5/index.html>