

1.2 PostgreSQL

En un cliente de Fedora 36, instalamos PGAdmin:

```
victor@fedora:~  
[victor@fedora ~]$ sudo rpm -i https://ftp.postgresql.org/pub/pgadmin/pgadmin4/yum/pgadmin4-fedora-repo-2-1.noarch.rpm  
[sudo] password for victor:  
warning: /var/tmp/rpm-tmp.oBgVfK: Header V3 RSA/SHA256 Signature, key ID 210976f2: NOKEY  
[victor@fedora ~]$ sudo yum install pgadmin4-desktop  
pgadmin4                                720 B/s | 833 B    00:01  
pgadmin4                                3.8 MB/s | 3.8 kB  00:00  
Importing GPG key 0x210976F2:  
  Userid      : "Package Manager (Package Signing Key) <packages@pgadmin.org>"  
  Fingerprint: E869 7E2E EF76 C02D 3A63 3277 8881 B2A8 2109 76F2  
  From        : /etc/pki/rpm-gpg/PGADMIN_PKG_KEY  
Is this ok [y/N]: y  
pgadmin4                                176 kB/s | 397 kB  00:02  
Last metadata expiration check: 0:00:01 ago on Thu 13 Oct 2022 08:40:40 AM CEST.  
Dependencies resolved.  
=====
```

Package	Architecture	Version	Repository	Size
---------	--------------	---------	------------	------

```
=====
```

Installing:

pgadmin4-desktop	x86_64	6.14-1.fc36	pgAdmin4	82 M
------------------	--------	-------------	----------	------

Installing dependencies:

libatomic	x86_64	12.2.1-2.fc36	updates	40 k
libpq	x86_64	14.1-2.fc36	fedora	198 k
pgadmin4-server	x86_64	6.14-1.fc36	pgAdmin4	96 M

Transaction Summary

```
=====
```

Install 4 Packages

Total download size: 178 M
Installed size: 611 M

En un servidor de Ubuntu Server 22.04, instalamos el servidor de PostgreSQL:

```
victor@fedora:~  
victor@server: ~  
victor@server:~$ sudo apt install postgresql  
[sudo] password for victor:  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following additional packages will be installed:  
  libcommon-sense-perl libjson-perl libjson-xs-perl libllvm14 libpq5 libsensors-config libsensors5  
  libtypes-serialiser-perl postgresql-14 postgresql-client-14 postgresql-client-common postgresql-common ssl-cert  
  sysstat  
Suggested packages:  
  lm-sensors postgresql-doc postgresql-doc-14 isag  
The following NEW packages will be installed:  
  libcommon-sense-perl libjson-perl libjson-xs-perl libllvm14 libpq5 libsensors-config libsensors5  
  libtypes-serialiser-perl postgresql postgresql-14 postgresql-client-14 postgresql-client-common postgresql-common  
  ssl-cert sysstat  
0 upgraded, 15 newly installed, 0 to remove and 0 not upgraded.  
Need to get 42.4 MB of archives.  
After this operation, 161 MB of additional disk space will be used.  
Do you want to continue? [Y/n] _
```

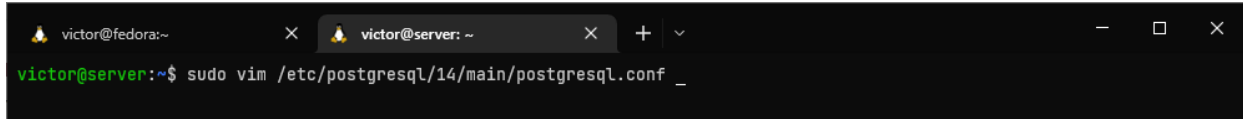
```
victor@fedora:~  
victor@server: ~  
victor@server:~$ systemctl status postgresql  
● postgresql.service - PostgreSQL RDBMS  
   Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; vendor preset: enabled)  
   Active: active (exited) since Thu 2022-10-13 06:52:45 UTC; 19s ago  
     Process: 4410 ExecStart=/bin/true (code=exited, status=0/SUCCESS)  
    Main PID: 4410 (code=exited, status=0/SUCCESS)  
       CPU: 2ms  
  
Oct 13 06:52:45 server systemd[1]: Starting PostgreSQL RDBMS...  
Oct 13 06:52:45 server systemd[1]: Finished PostgreSQL RDBMS.  
victor@server:~$ _
```

En el apartado de redes, ambas máquinas se van a conectar a través de una red interna de VirtualBox, las IP són la siguientes:

10.10.10.1/24 → Servidor

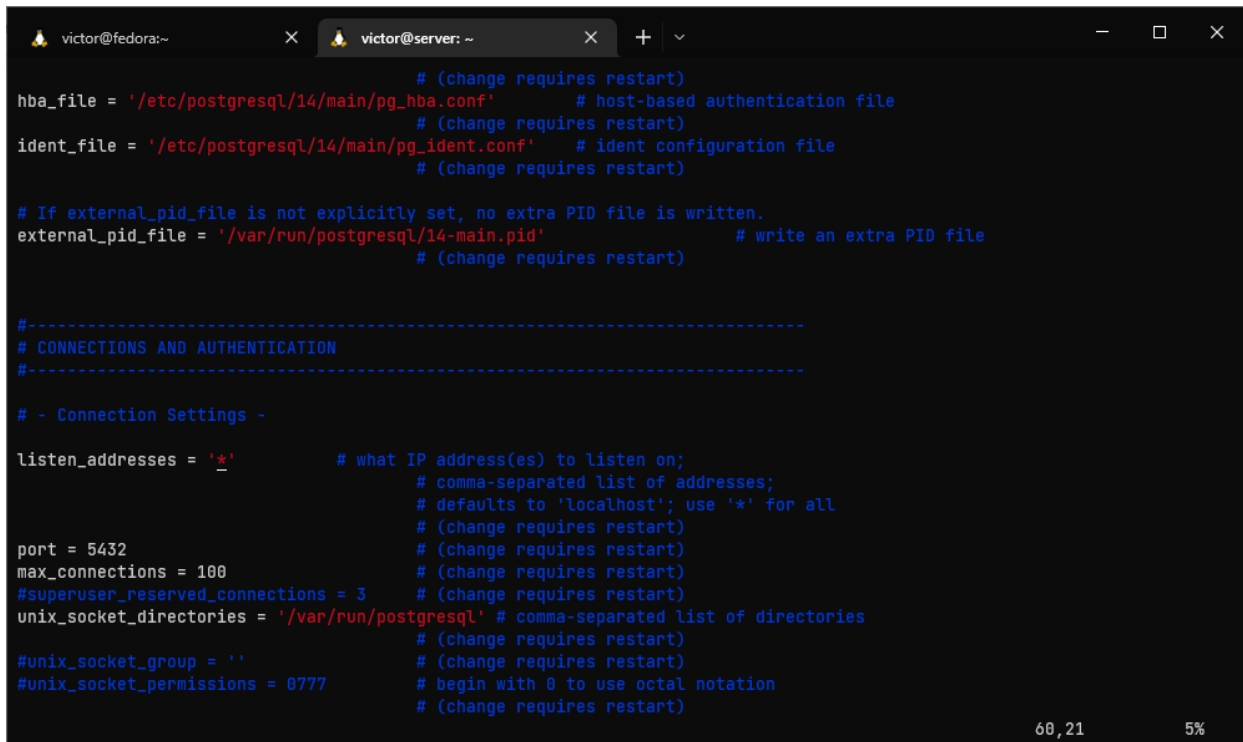
10.10.10.2/24 → Cliente

Por defecto, postgres solo acepta conexiones locales, para que acepte conexiones remotas, configuramos el servidor:



```
victor@fedora:~  
victor@server: ~  
victor@server:~$ sudo vim /etc/postgresql/14/main/postgresql.conf _
```

Cambiamos la línea "listen_addresses" de "localhost" a "*".



```
victor@fedora:~  
victor@server: ~  
# (change requires restart)  
hba_file = '/etc/postgresql/14/main/pg_hba.conf' # host-based authentication file  
# (change requires restart)  
ident_file = '/etc/postgresql/14/main/pg_ident.conf' # ident configuration file  
# (change requires restart)  
  
# If external_pid_file is not explicitly set, no extra PID file is written.  
external_pid_file = '/var/run/postgresql/14-main.pid' # write an extra PID file  
# (change requires restart)  
  
#-----  
# CONNECTIONS AND AUTHENTICATION  
#-----  
  
# - Connection Settings -  
  
listen_addresses = '*' # what IP address(es) to listen on;  
# comma-separated list of addresses;  
# defaults to 'localhost'; use '*' for all  
# (change requires restart)  
port = 5432 # (change requires restart)  
max_connections = 100 # (change requires restart)  
#superuser_reserved_connections = 3 # (change requires restart)  
unix_socket_directories = '/var/run/postgresql' # comma-separated list of directories  
# (change requires restart)  
#unix_socket_group = '' # (change requires restart)  
#unix_socket_permissions = 0777 # begin with 0 to use octal notation  
# (change requires restart)  
  
60,21 5%
```

Finalmente reiniciamos el servicio:

```
victor@fedora:~  
victor@server: ~  
victor@server:~$ sudo vim /etc/postgresql/14/main/postgresql.conf  
victor@server:~$ systemctl restart postgresql  
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ===  
Authentication is required to restart 'postgresql.service'.  
Authenticating as: victor  
Password:  
==== AUTHENTICATION COMPLETE ===  
victor@server:~$ _
```

Si aún así postgres rechaza nuestra conexión (como en mi caso), tenemos que configurar el siguiente fichero:

```
victor@fedora:~  
victor@server: ~  
victor@server:~$ sudo vim /etc/postgresql/14/main/pg_hba.conf _
```

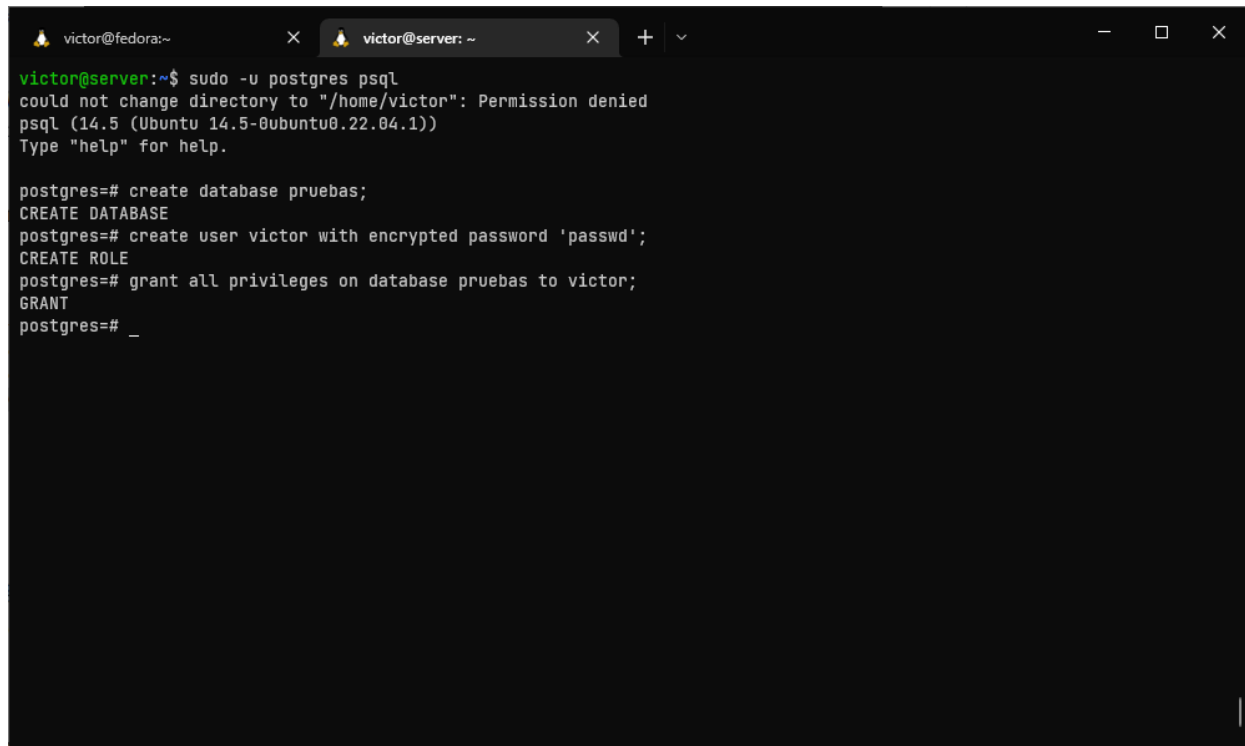
Añadimos las líneas del final:

```
victor@fedora:~  
victor@server: ~  
  
# DO NOT DISABLE!  
# If you change this first entry you will need to make sure that the  
# database superuser can access the database using some other method.  
# Noninteractive access to all databases is required during automatic  
# maintenance (custom daily cronjobs, replication, and similar tasks).  
#  
# Database administrative login by Unix domain socket  
local all postgres peer  
  
# TYPE DATABASE USER ADDRESS METHOD  
  
# "local" is for Unix domain socket connections only  
local all all peer  
# IPv4 local connections:  
host all all 127.0.0.1/32 scram-sha-256  
# IPv6 local connections:  
host all all ::1/128 scram-sha-256  
# Allow replication connections from localhost, by a user with the  
# replication privilege.  
local replication all peer  
host replication all 127.0.0.1/32 scram-sha-256  
host replication all ::1/128 scram-sha-256  
  
# Aceptar las conexiones externas  
host all all 0.0.0.0/0 md5  
host all all ::/0 md5  
"/etc/postgresql/14/main/pg_hba.conf" 108L, 51918 108,24 Bot
```

Reiniciamos el servicio una vez más:

```
victor@fedora:~  
victor@server: ~  
  
victor@server:~$ systemctl restart postgresql  
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ===  
Authentication is required to restart 'postgresql.service'.  
Authenticating as: victor  
Password:  
==== AUTHENTICATION COMPLETE ===  
victor@server:~$ _
```

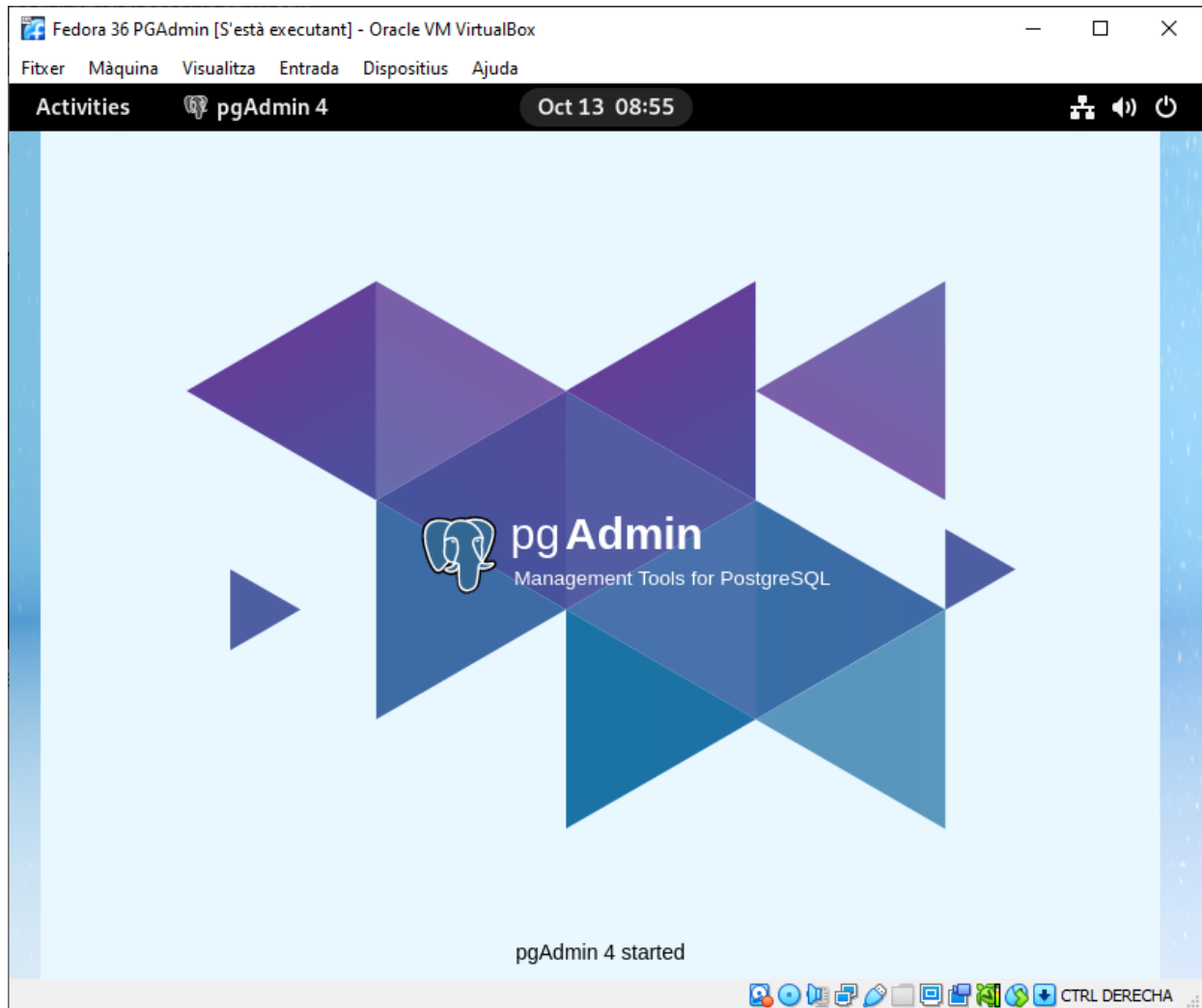
Creamos una base de datos nueva, y un usuario para acceder a esta:

A terminal window with two tabs. The first tab is titled 'victor@fedora:~' and the second, active tab is 'victor@server: ~'. The terminal output shows a user running 'sudo -u postgres psql' on the server. This opens a PostgreSQL prompt where several SQL commands are entered: 'create database pruebas;', 'CREATE DATABASE', 'create user victor with encrypted password 'passwd';', 'CREATE ROLE', and 'grant all privileges on database pruebas to victor;'. The prompt ends with 'postgres=# _'.

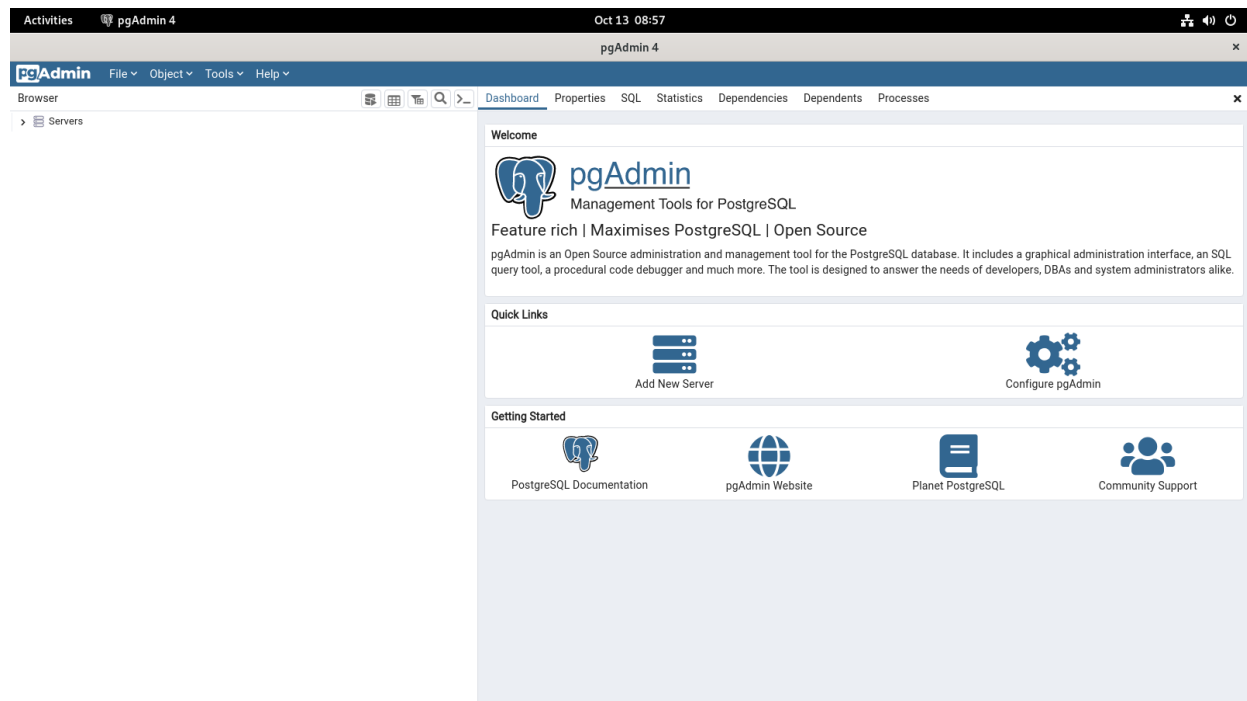
```
victor@server:~$ sudo -u postgres psql
could not change directory to "/home/victor": Permission denied
psql (14.5 (Ubuntu 14.5-0ubuntu0.22.04.1))
Type "help" for help.

postgres=# create database pruebas;
CREATE DATABASE
postgres=# create user victor with encrypted password 'passwd';
CREATE ROLE
postgres=# grant all privileges on database pruebas to victor;
GRANT
postgres=# _
```

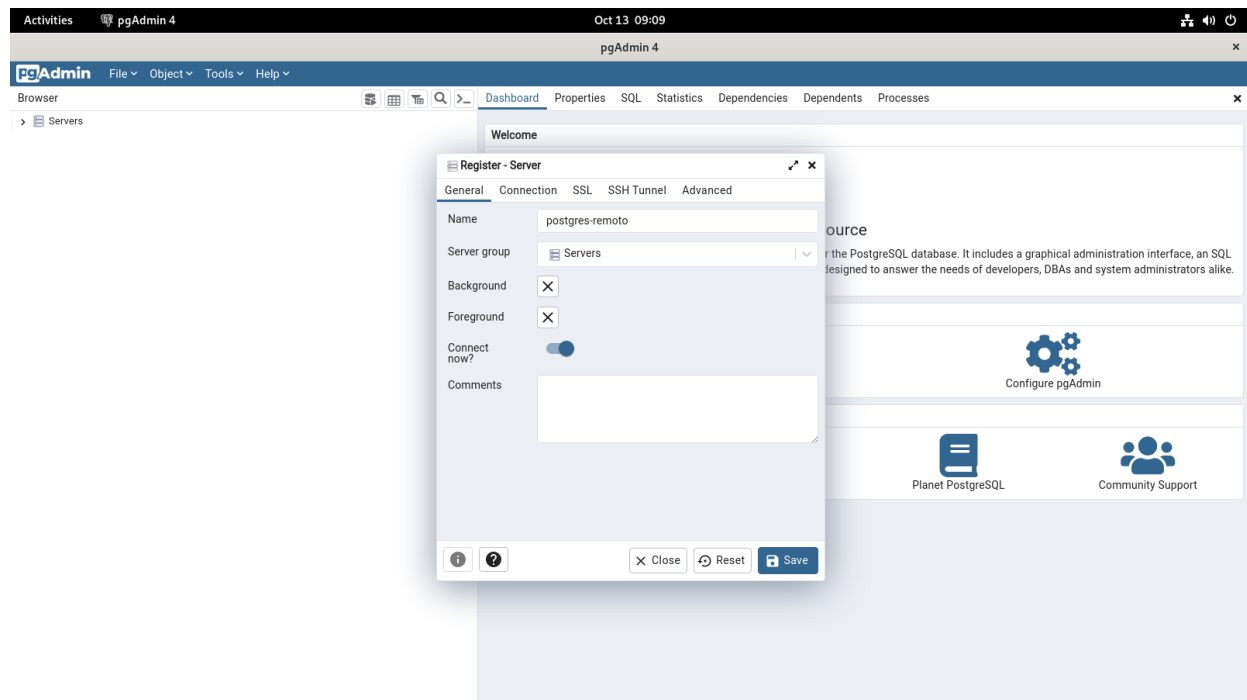
En el cliente, iniciamos PGAdmin4:



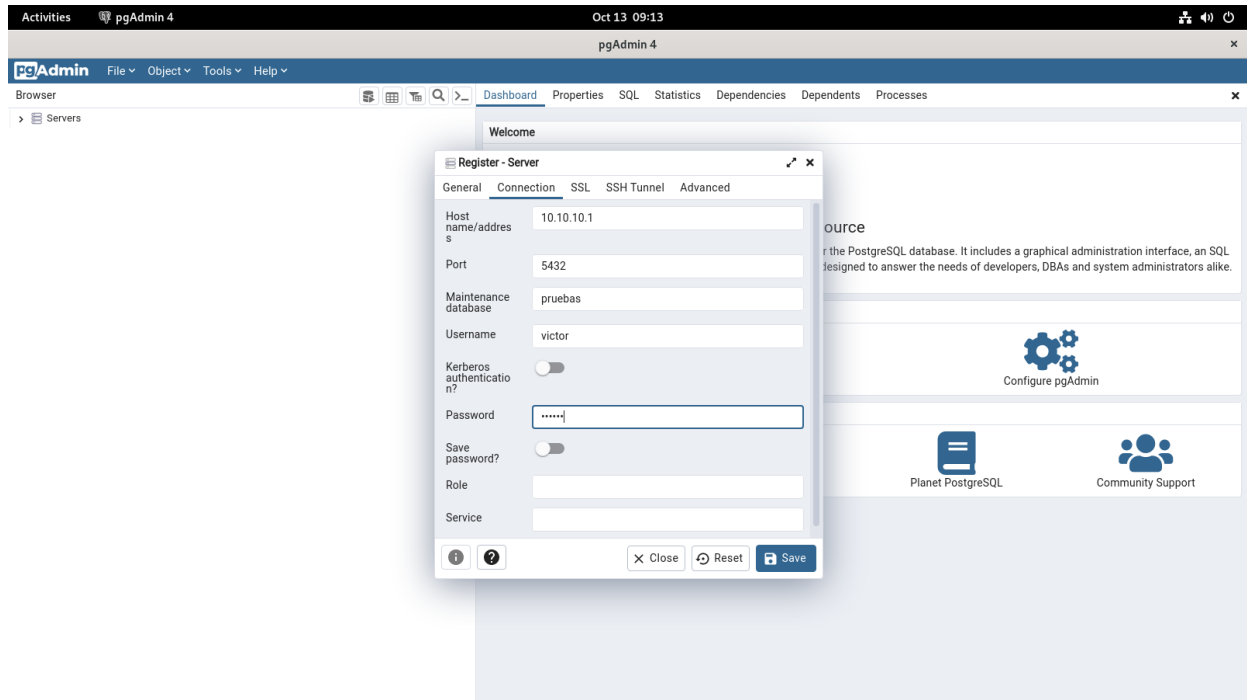
Nos pide introducir una contraseña maestra, y nos muestra el panel. Hacemos click en "Add New Server":



Añadimos un nombre al servidor:



Configuramos el acceso a la base de datos remota:



Hacemos click en save y ya podremos acceder a la base de datos remota:

