

MTP02. AUTENTICACIÓN EXTERNA DE SISTEMAS DE BASES DE DATOS USANDO OPENLDAP

A la hora de autenticarnos en un sistema de bases de datos podemos hacerlo con los usuarios que hemos creado en el mismo, pero también podemos hacerlo mediante servicios externos a él, como es el caso de OpenLDAP. Para este trabajo voy a explicar que hay que hacer para poder autenticarnos en PostgreSQL con un usuario que hayamos creado en OpenLDAP.

1. INSTALACIÓN DE OPENLDAP

OpenLDAP está en los repositorios de Ubuntu, así que con hacer un update e instalar los paquetes que correspondan es más que suficiente. Primero empezamos con slapd y ldap-utils:

```
miguelgulu@asgbd:~$ sudo apt install slapd ldap-utils
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  libldap-2.4-2 libodbc1
Paquetes sugeridos:
  libsasl2-modules-gssapi-mit | libsasl2-modules-gssapi-heimdal libmyodbc
  odbc-postgresql tdsodbc unixodbc-bin
Se instalarán los siguientes paquetes NUEVOS:
  ldap-utils libodbc1 slapd
Se actualizarán los siguientes paquetes:
  libldap-2.4-2
1 actualizados, 3 nuevos se instalarán, 0 para eliminar y 191 no actualizados.
Se necesita descargar 1.708 kB/1.863 kB de archivos.
Se utilizarán 17,7 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]
```

A continuación nos pedirá añadir la contraseña del usuario administrador de OpenLDAP. Le pondremos la que nosotros queramos.

Seguiremos configurando el archivo de hosts, donde añadiremos la dirección y dominio del servidor de ldap que vayamos a crear:

```
miguelgulu@asgbd: ~
GNU nano 4.8 /etc/hosts
127.0.0.1    localhost
127.0.1.1    asgbd
192.168.122.201 postgresql.local ldapserver
# The following lines are desirable for IPv6 capable hosts
::1         ip6-localhost ip6-loopback
```

Una vez hecho esto, instalamos más paquetes de ldap que contendrán la configuración del servidor:

```
miguelgulu@asgbd:~$ sudo apt install libnss-ldap -y
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  ldap-auth-client ldap-auth-config libpam-ldap
Paquetes sugeridos:
  nscd
Se instalarán los siguientes paquetes NUEVOS:
  ldap-auth-client ldap-auth-config libnss-ldap libpam-ldap
0 actualizados, 4 nuevos se instalarán, 0 para eliminar y 191 no actualizados.
Se necesita descargar 109 kB de archivos.
Se utilizarán 353 kB de espacio de disco adicional después de esta operación.
0% [Trabajando]
```

- Se nos pedirá la dirección IP de nuestro servidor, el cual será el de la máquina donde vamos a trabajar. Tiene el formato **ldapi://[dirección_IP]** .
- Luego escribimos nuestro dominio separado por dc, el cual en nuestro caso será **dc=postgresql, dc=local** .
- Elegimos la **versión 3** de LDAP.
- Haremos el root local como administrador de la base de datos, así que le daremos a que **sí**.
- Nos preguntará si conectarse a la base de datos ldap requiere inicio de sesión. Le daremos a que **no**.
- Escribimos nuestra cuenta root para ldap el cual será **cn=admin, dc=postgresql, dc=local**.
- Y escribimos la contraseña...

Una vez configurado el servidor vamos a reconfigurar los paquetes de slapd respecto a los datos que vienen de fábrica:

```
miguelgulu@asgbd:~$ sudo dpkg-reconfigure slapd
```

- Nos pregunta si queremos omitir la configuración del servidor: **No**.
- Introducimos el dominio DNS que será **postgresql.local**.
- El nombre de la organización: **postgresql**.

- Y la **contraseña**.
- ¿Queremos que se borre la base de datos cuando purguemos el slapd? **No**.
- ¿Mover la base de datos antigua? **Sí**.

Una vez instalados todos los paquetes tendremos que añadir los datos del servidor al archivo ldap.conf:

```
GNU nano 4.8 /etc/ldap/ldap.conf Modificado
#
# LDAP Defaults
#
# See ldap.conf(5) for details
# This file should be world readable but not world writable.

BASE    dc=postgresql,dc=local
URI      ldap://192.168.122.201:389
```

Si no tenemos instalado apache2 lo instalamos y si lo tenemos reiniciamos el servicio con 'systemctl restart apache2'.

Para terminar abrimos los puertos 389 y 80 con 'ufw allow [puerto]'.

Y si ejecutamos el comando 'ldapsearch -x' debe salir como 'result: 0 success':

```
#
# postgresql.local
dn: dc=postgresql,dc=local
objectClass: top
objectClass: dcObject
objectClass: organization
o: postgresql
dc: postgresql

# admin, postgresql.local
dn: cn=admin,dc=postgresql,dc=local
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator

# search result
search: 2
result: 0 Success

# numResponses: 3
# numEntries: 2
```

2. INSTALACIÓN DE PHPLDAPADMIN

Si queremos instalar un software para administrar las cuentas de LDAP podemos usar varios como por ejemplo Apache Directory Studio y PHPLDAPAdmin. Yo usaré este último.

Para instalarlo, ya que está en los repositorios de Ubuntu, basta hacerlo con apt:

```
miguelgulu@asgbd:~$ sudo apt install phpldapadmin -y
[sudo] contraseña para miguelgulu:
```

Una vez instalado, añadimos el enlace simbólico:

```
miguelgulu@asgbd:~$ sudo ln -s /usr/share/phpldapadmin/ /var/www/html/phpldapadm
ln
```

Y para terminar basta con cambiar los siguientes parámetros en el archivo de configuración de phpldapadmin:

- La zona horaria:

```
GNU nano 4.8 /etc/phpldapadmin/config.php
30 seconds or the setting of max_execution_time if this is null. */
// $config->custom->session['timelimit'] = 30;

/* Our local timezone
This is to make sure that when we ask the system for the current time, we
get the right local time. If this is not set, all time() calculations will
assume UTC if you have not set PHP date.timezone. */
// $config->custom->appearance['timezone'] = null;
# $config->custom->appearance['timezone'] = 'Australia/Melbourne';
```

- La dirección del servidor y el dominio:

```
GNU nano 4.8 /etc/phpldapadmin/config.php Modificado
(Unix socket at /usr/local/var/run/ldap) */
$servers->setValue('server','host','192.168.122.201');

/* The port your LDAP server listens on (no quotes). 389 is standard. */
// $servers->setValue('server','port',389);

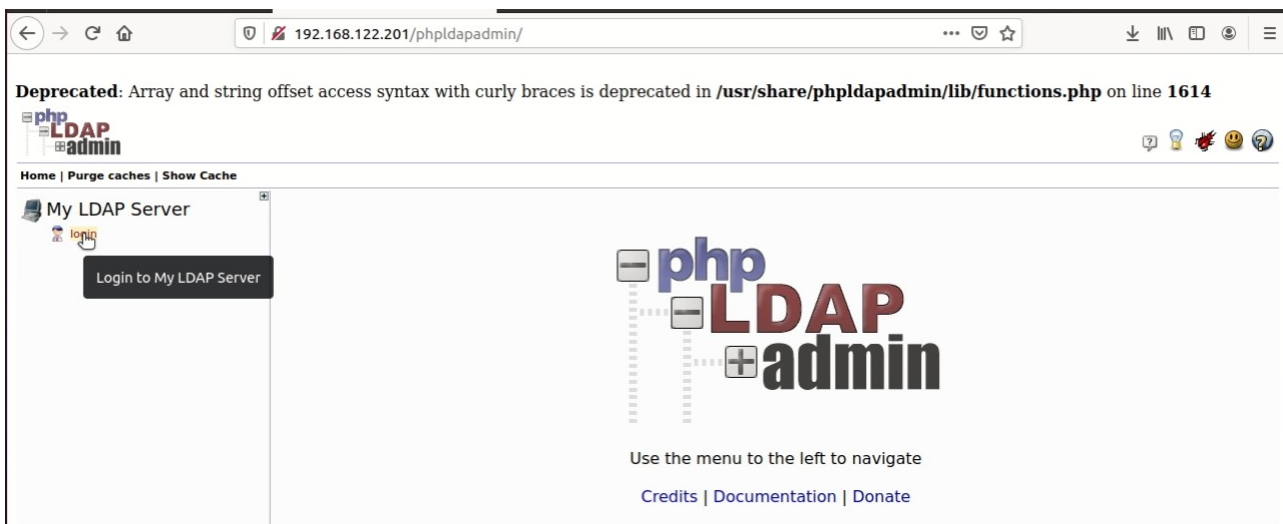
/* Array of base DNS of your LDAP server. Leave this blank to have phpldapadmin
auto-detect it for you. */
$servers->setValue('server','base',array('dc=postgresql,dc=local'));
```


· Y el usuario administrador:

```
the directory for users (ie, if your LDAP server does not allow anonymous binds. */
$servers->setValue('login','bind_id','cn=admin,dc=postgresql,dc=local');
# $servers->setValue('login','bind_id','cn=Manager,dc=example,dc=com');

/* Your LDAP password. If you specified an empty bind_id above, this MUST also
```

Una vez terminado, reiniciamos el servicio de apache y entramos en nuestro navegador en la dirección IP del servidor seguido de `/phpldapadmin`:



Hacemos **login** con nuestro usuario administrador **cn=admin, dc=postgresql, dc=local** . Cuando hayamos iniciado sesión, crearemos una unidad organizativa, un grupo posix y un usuario respectivamente, de forma que quede tal que así:

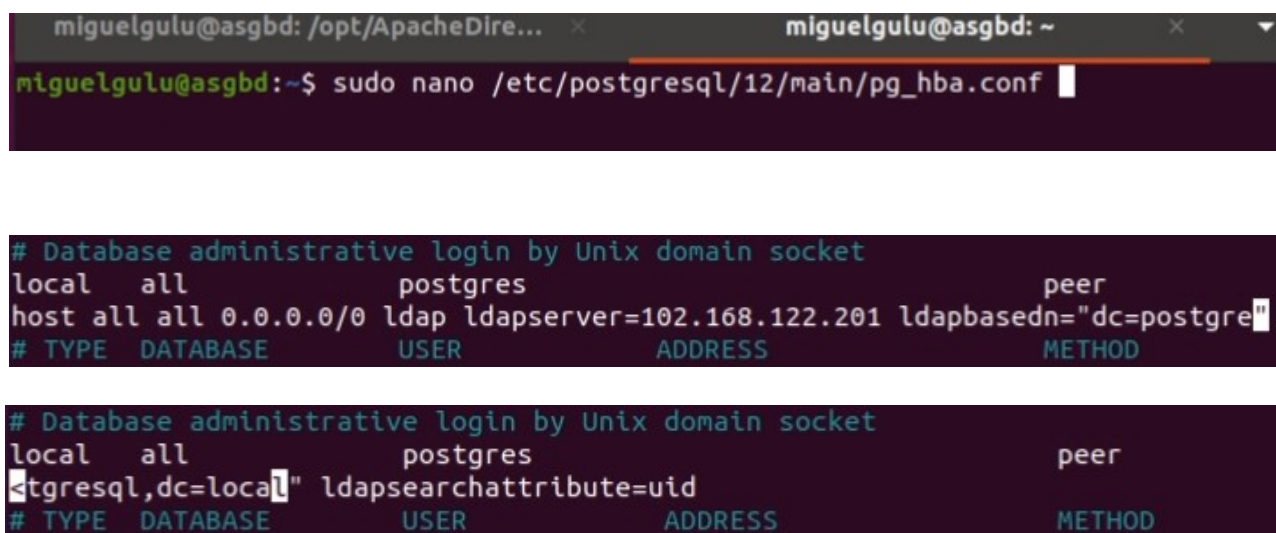


3. INSTALACIÓN DE POSTGRESQL Y CONFIGURACIÓN DE ENLACE

Cuando hayamos seguido todos los pasos anteriores nos faltará instalar **postgresql** y configurar los archivos necesarios para autenticarnos con el usuario que hemos creado. Primero instalaremos postgres con el siguiente comando:

```
$ sudo apt install postgresql-12
```

Y configuramos el archivo `pg_hba.conf` con los parámetros que indicaran donde se aloja nuestro servidor y que criterio se usa para identificar a los usuarios:



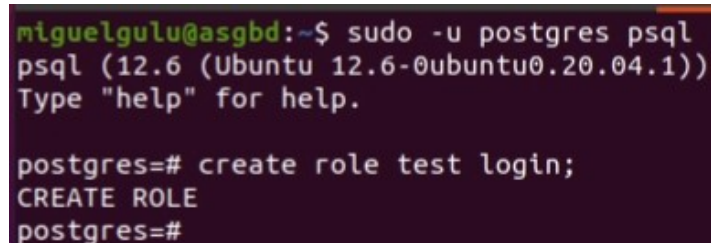
```
miguelgulu@asgbd: /opt/ApacheDire... x miguelgulu@asgbd: ~ x
miguelgulu@asgbd:~$ sudo nano /etc/postgresql/12/main/pg_hba.conf

# Database administrative login by Unix domain socket
local all postgres peer
host all all 0.0.0.0/0 ldap ldapserver=102.168.122.201 ldapbasedn="dc=postgres"
# TYPE DATABASE USER ADDRESS METHOD

# Database administrative login by Unix domain socket
local all postgres peer
"postgresql,dc=local" ldapsearchattribute=uid
# TYPE DATABASE USER ADDRESS METHOD
```

Esta línea indica es que vamos a usar un método de conexión basado en LDAP apuntando a la dirección del servidor LDAP, añadiendo el Base DN del dominio y añadiéndole que busque en el árbol el atributo uid , si el usuario coincide con el nombre y la contraseña que tiene el objeto uid, se completa el inicio de sesión.

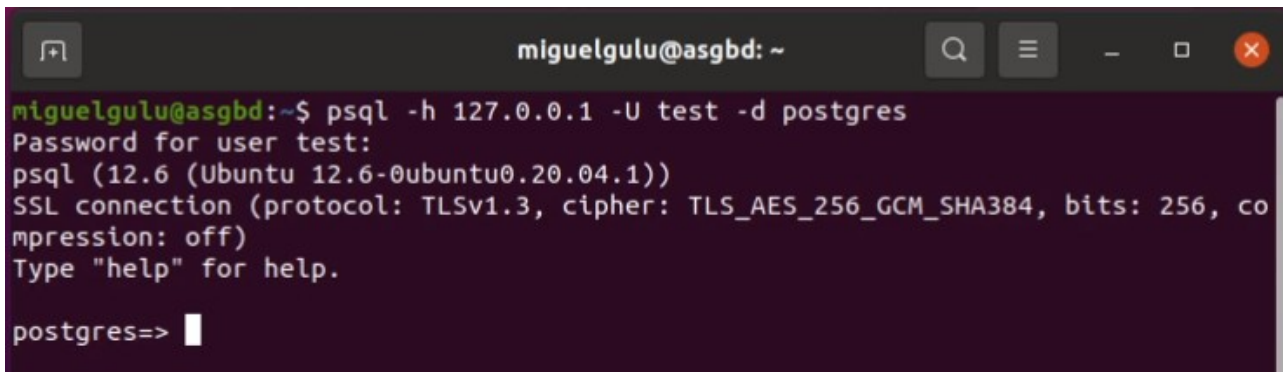
Con todo esto, iniciamos el cliente de postgres con el usuario administrador y creamos el rol de login para el usuario:



```
miguelgulu@asgbd:~$ sudo -u postgres psql
psql (12.6 (Ubuntu 12.6-0ubuntu0.20.04.1))
Type "help" for help.

postgres=# create role test login;
CREATE ROLE
postgres=#
```

Finalmente, iniciamos sesión con nuestro usuario de LDAP de la siguiente manera:

A terminal window with a dark background. The title bar shows 'miguelgulu@asgbd: ~'. The terminal text is as follows:

```
miguelgulu@asgbd:~$ psql -h 127.0.0.1 -U test -d postgres
Password for user test:
psql (12.6 (Ubuntu 12.6-0ubuntu0.20.04.1))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, co
mpression: off)
Type "help" for help.

postgres=> |
```

Y listo! Así podremos crear cualquier usuario con LDAP e iniciar sesión con él.