

Aaron Andal : Acceso

El fichero de configuración está en `/etc/postgresql/13/main/pg_hba.conf`. locate `pg_hba.conf` - updatedb

TYPE (local o remoto) DATABASE (En que BD o ALL) USER (Que usuario o ALL) ADDRESS (Que IP o limitado) METHOD (Tipo de auth)

Ejemplo:

```
# Permet connexió local a qualsevol base de dades als usuaris del sistema
local all all peer

# Permet connexió local a qualsevol base de dades a qualsevol usuari sense posar contrasenya
local all all trust

# Accés local (però amb connexió TCP/IP) per a usuaris del sistema
host all all 127.0.0.1/32 ident

# El mateix que l'anterior però usant màscara de xarxa
host all all 127.0.0.1 255.255.255.255 ident

# Permet connexió de qualsevol usuari de Postgres amb contrasenya des de
# la màquina amb IP 192.168.12.10 connectar-se a la base de dades training
host training all 192.168.12.10/32 md5

# Rebutja tota connexió des de la màquina amb IP 192.168.54.1
host all all 192.168.54.1/32 reject

# Orden matters

# Reject a todos los usuarios y bases de datos desde un PC pero permites al resto de su red

host all all 192.168.1.224/32 reject

host all all 192.168.1.0/24 md5

# Si lo pones al principio acepta

host all all 192.168.1.0/24 md5

host all all 192.168.1.224/32 reject

# TRUST SIN PASSWORD

1. 1ª línea: TYPE: host DB: con_rem USER: con_rem ADD: ip_con_rem METHOD: trust

* Trust: **SIN PASSWORD**

## Permite a con_rem acceder a su BD desde ese ordenador solo y sin password

2. 2ª línea: TYPE: host DB: con_rem USER: con_rem ADD: 0.0.0.0/0 METHOD: reject

* ALL HOST: 0.0.0.0/0
```

* Same que el de antes pero 0.0.0.0/0 reject --> No podrá acceder desde otro PC

No permite acceder a con_rem a su BD con_rem desde otro ordenador

3. 3ª línea: TYPE: host DB: con_rem USER: con_rem ADD: 0.0.0.0/0 METHOD: reject

* ALL HOST: 0.0.0.0/0

4. Si los usuarios no existen en la BD, hay que crearlos con un ``CREATE USER`` --> ``TIENE LOGIN`` y luego un ``CREATE DB`` de la misma.

Permite a con_rem acceder a su BD desde ese ordenador solo y sin password

```
host con_rem      con_rem      172.18.0.2/32  trust
```

No permite acceder a con_rem a su BD con_rem desde otro ordenador

```
host con_rem      con_rem      0.0.0.0/0      reject
```

Cualquier otro usuario puede acceder a la BD con_rem

```
host  con_rem      all          0.0.0.0/0      trust
```

CREATE USER con_rem;

CREATE USER otheruser;

CREATE DATABASE con_rem;

+ Costat client.

Des de `172.18.0.2`:

```
psql -h ip_servidor -U con_rem con_rem
```

Des de `192.168.0.23`:

```
psql -h ip_servidor -U con_rem con_rem
```

Ens deixa entrar

...

```
psql -h ip_servidor -U otheruser con_rem
```

...

Ens deixa entrar

Des d'una màquina diferent a `172.18.0.2`:

...

```
psql -h ip_servidor -U con_rem con_rem
```

...

Error

...

```
psql -h ip_servidor -U otheruser con_rem
```

...

Ens deixa entrar

rem_admin puede entrar a todas las BD desde su PC con password Md5 TENDRÁ SUPERUSER

```
host all          rem_admin  192.168.0.23/32 md5
```

rem_admin no puede entrar desde otro ORDENADOR.

```
host all          rem_admin  0.0.0.0/0          reject
```

Otros usuarios si pueden entrar

```
host all all          0.0.0.0/0          trust
```

```
CREATE USER rem_admin SUPERUSER PASSWORD 'ra';
```

```
CREATE USER other_user;
```

Explica qué configuración debe tener el postgresql y qué sentencias se deben

ejecutado para que un __usuario__ pueda __accedir__ a __todas__ las bases de datos desde __cualquier ordenador__ del aula.

El usuario debe llamarse "semi_admin" y la

contraseña, almacenada en el postgresql, debe ser "sa". Este usuario debe

tener permisos para poder crear bases de datos y nuevos usuarios. Explica cómo comprobar el correcto funcionamiento.

-

Keshi:

* Server:

pg_hba.conf:

...

host	all	semi_admin	172.18.0.2/16	md5
------	-----	------------	---------------	-----

...

postgresql.conf

...

listen_addresses='*'

...

...

su -l postgres ; psql template1

...

```sql

CREATE USER semi\_admin WITH PASSWORD 'sa' CREATEDB CREATEROLE;

...

\du --> Revisar

\* Cliente:

...

psql -h 172.18.0.2 -U semi\_admin template1;

...

#### + Costat servidor

Editem el fitxer `/etc/postgresql/13/main/pg_hba.conf` afegint les línies:

...

```
host all semi_admin 192.168.0.0/24 md5
```

...

Editem el fitxer `/etc/postgresql/13/main/postgresql.conf` i modifiquem la línia:

...

```
listen_addresses='*'
```

...

Reiniciem el servei postgresql perquè torni a llegir els fitxers de configuració:

...

```
systemctl restart postgresql
```

...

Continuem al costat servidor, canviem a usuari administrador de la base de dades, potsgres.

...

```
su -l postgres
```

...

Ens connectem a la base de dades `template1` i afegim l'usuari `semi_admin` amb els permisos corresponents:

...

```
CREATE USER semi_admin PASSWORD 'sa';
```

...

#### + Costat client.

...

```
psql -h ip_servidor -U semi_admin training
```

...

Ens deixa entrar

Per veure que té els permisos demanats:

...

CREATE DATABASE test\_db;

CREATE USER test\_user;

DROP DATABASE test\_db;

DROP USER test\_user;

...

Explica qué configuración debe tener el postgresql y qué sentencias se deben ejecutado para que un usuario sólo pueda acceder a la base de datos "db123" desde cualquier ordenador. El usuario debe llamarse "us123" y la contraseña, almacenada en el postgresql, debe ser "123".

El usuario sólo puede tener 2 conexiones simultáneas y sólo tiene que poder conectarse hasta el 31-12-2022, incluido. Explica cómo comprobar el correcto funcionamiento.

pg\_hba.conf

...

|      |       |       |               |     |
|------|-------|-------|---------------|-----|
| host | db123 | us123 | 172.16.0.0/16 | md5 |
|------|-------|-------|---------------|-----|

...

editar el fichero postgresql.conf

reiniciar

iniciar como postgres

crear el usuario --> CREATE USER db123 WITH PASSWORD '123' CONNECTION LIMIT 2 VALID UNTIL '2022-12-31'

—

+ Lado servidor

Editamos el archivo `/etc/postgresql/13/main/pg\_hba.conf` añadiendo las líneas:

...

host db123 us123 0.0.0.0/0 md5

...

Editamos el archivo `/etc/postgresql/13/main/postgresql.conf` y modificamos la línea:

...

`listen_addresses='*'`

...

Reiniciamos el servicio postgresql para que vuelva a leer los archivos de configuración:

...

`systemctl restart postgresql`

...

Continuamos al lado servidor, cambiamos a usuario administrador de la base de datos, postgres.

...

`su -l postgres`

...

Nos conectamos a la base de datos `template1` y añadimos al usuario us123 con los permisos correspondientes:

...

`CREATE USER us123 PASSWORD '123' CONNECTION LIMIT 2 VALID UNTIL '2022-12-31';`

...

+ Lado cliente.

...

`psql -h ip_servidor -U us123 training`

...

ERROR

...

`psql -h ip_servidor -U us123 db123`

...

Nos deja entrar

Repetimos la instrucción anterior desde otra terminal:

...

```
psql -h ip_servidor -U us123 db123
```

...

Nos deja entrar

Ejecutamos por 3a vez la misma instrucción:

...

```
psql -h ip_servidor -U us123 db123
```

...

**ERROR**

Ahora jugamos con la fecha. Cambiamos la fecha límite por una fecha del pasado y comprobamos que al intentar conectarse da un error.

Vigilamos de no estar conectados a la base de datos antes de realizar el cambio

Cualquiera es 0.0.0.0/0

Per fer accessos remots, hem de dir a Postgresql que escolti a les adreces de xarxa que volem en el fitxer `/etc/postgresql/13/main/postgresql.conf`:

```
port=5432

listen_addresses='localhost' # per defecte
listen_addresses='*' # per escoltar tot
listen_addresses='192.168.0.24, 192.168.0.26' # Per escoltar IPs determinades
```

**Siempre que se toca los ficheros de configuración hay que reiniciar SYSTEMCTL RESTART POSTGRESQL.**

Si tuviéramos error en el FIREWALL: `systemctl disable|stop iptables // systemctl disable|stop ip6tables // systemctl disable|stop firewalld`. **CUIDADO CON GUEST o cualquier USUARIO porque no está registrado en GANDHI del INSTI.**

#### TIPO DE CONEXIÓN

1a columna de `pg_hba.conf` → SOCKET UNIX (local) / SOCKET TCP/IP (host). → **PSQL TRAINING (local) - PSQL -H localhost/ip TRAINING;**

**LAST COLUMNA: peer** → Solo para LOCALES. **ident:** Conex remoto. **md5:** Comprueba y pide password. **reject:** Deniega acceso en concreto.

Ejemplo: **Conexión de nuestra RED, sí puede entrar con CONTRASEÑA.**



```
psql -h 192.168.1.23 training
```

```
IPv4 local connections:
```

```
host all all 192.168.1.0/24 md5
```

Ejemplo 1: Acceso por TCP/IP (RED) a la BD test001 como usuario test desde el ordenador IP 10.0.0.100 y método md5:

```
host test001 test 10.0.0.100/32 md5
```

Ejemplo 2: Acceso por TCP/IP (RED) a todas las BD en los PCs 10.0.0.100 y 10.1.1.100 desde todos los ordenadores de la red 10.0.0.0 /24 y método md5:

```
host all test 10.0.0.0/24 md5
```

Ejemplo 3: Acceso por TCP/IP (RED) a todas las BD en los PCs 10.0.0.100 y 10.1.1.100 método md5:

```
host all test 10.0.0.100/32 md5
```

```
host all test 10.0.1.100/32 md5
```

Ejemplo 4: Denegar acceso a todas las BD y a todos los ordenadores de la red 10.0.0.0/24 y dar acceso al resto del mundo a ALL con MD5.

```
host all test 10.0.0.0/24 reject
```

```
host all test 0.0.0.0/0 md5
```