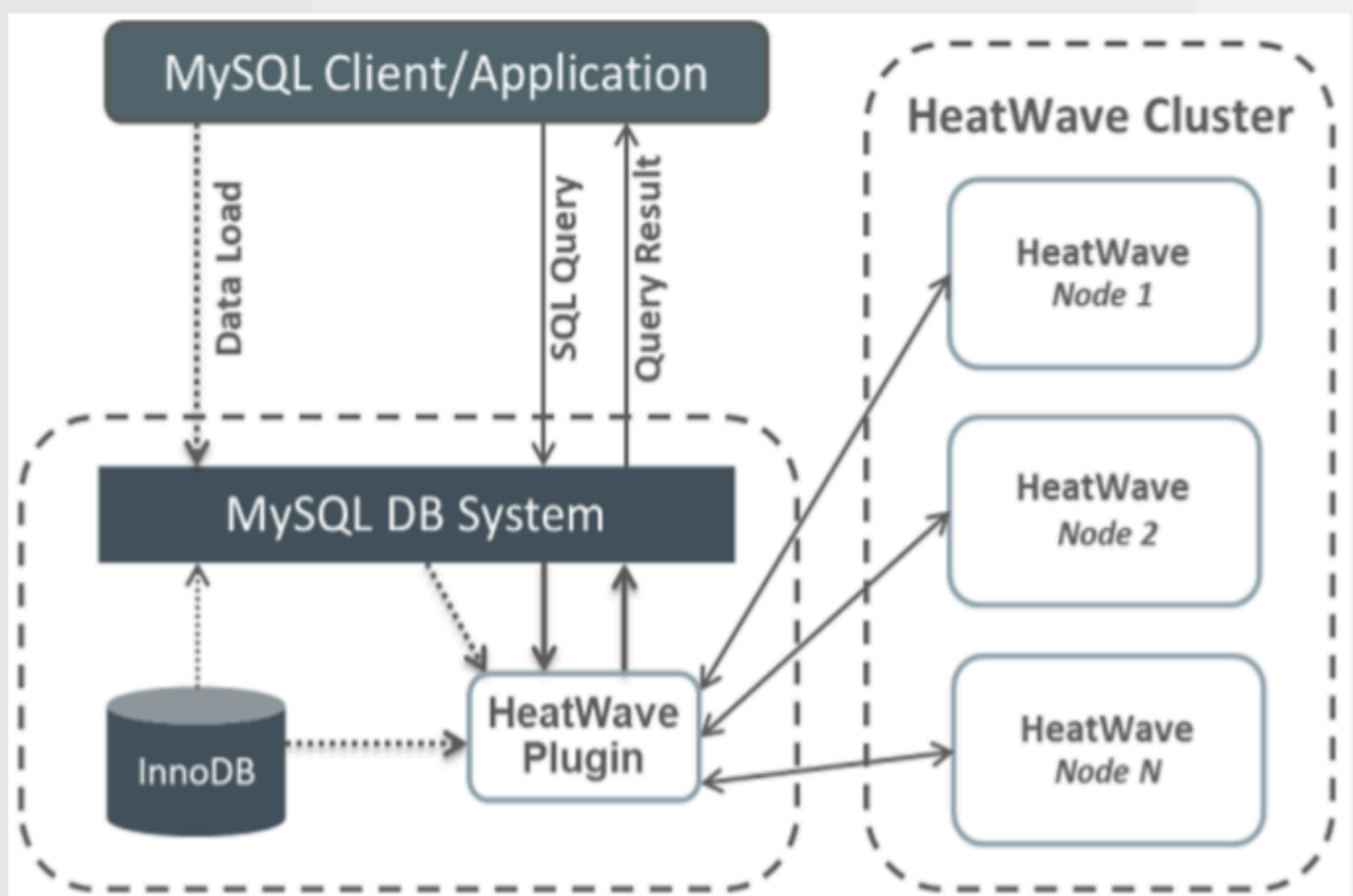


BASE DE DADES

ALTA DISPONIBILITAT



NOM: *Bixiang Zhu*

CURS: *1r ASIX*

ESCOLA: *InstitutMVM*



ÍNDEX

| | |
|------------------------------|-----------|
| ÍNDEX | 2 |
| MYSQL InnoDB Cluster | 3 |
| Oracle MySQL HeatWave | 6 |
| POSTGRESQL | 10 |
| INSTAL·LACIÓ | 10 |
| CONFIGURACIÓ | 11 |
| INTRODUCCIÓ .SQL | 13 |
| ERROR INTRODUCCIÓ .SQL | 13 |
| SOLUCIÓ | 14 |

MYSQL InnoDB Cluster

Llegiu l'article sobre MySQL InnoDB Cluster i respongueu el següent:

1.- Quantes màquines necessites almenys per crear un clúster de servidors amb alta disponibilitat?

3 màquines com a mínim per crear un cluster de servidors amb alta disponibilitat (grup de replicació).

2.- Quina creus que és la configuració mínima de cada màquina?

Això depèn de les necessitats específiques de l'aplicació. La pàgina posa que utilitza 1 GB de ram pero crec que cada màquina hauria de tenir:

CPU: 2-4 nuclis

Memoria RAM: 4-8 GB

Disc: 100GB

Connexió rapida

És més que suficient però clarament sent un servidor és millor tenir una mica més perquè no passi ningun error de inestabilitat.

En cas de configuració del sistema hi haurà 1 que farà de router, 1 de client i 3 de servidor en el clúster. Totes les màquines s'han de instal·lar i configurar el mysql-shell, mysql-community-server, la interfície de les xarxes i que el S.O (sistema operatiu) sigui debian o GNU/Linux.

3.- Algun dels servidor necessita més recursos que altres?

No ja que en un cluster InnoDB, tots els nodes han de ser capaços de fer les accions del node primari en cas de fallada, la qual cosa totes les màquines haurien de tenir una configuració i components similars per assegurar la disponibilitat i el rendiment.

4.- Com podem monitoritzar el clúster?

Podem monitoritzar el clúster utilitzant la eina MySQL Enterprise Monitor, Prometheus amb Grafana o altres eines de monitoratge que soportin MySQL. Aquests eines mencionats proporcionen detalls sobre l'estat de clúster, rendiment i problemes potencials. En el propi InnoDB Cluster es pot monitoritzar utilitzant la funció `cluster.describe()` i veure el seu estat amb `cluster.status()`.

5.- Es pot controlar a la vegada totes les màquines del clúster? Cóm?

Si, es pot controlar totes les màquines del clúster a la vegada amb la eina MySQL Shell amb comandes de gestió del cluster. Altres eines com MySQL Router ajuda a gestionar la distribució de les connexions a les diferents màquines.

6.- Per a què serveix a efectes pràctics tenir un clúster de servidors?

Tenir un cluster de servidors serveix per assegurar l'alta disponibilitat, l'escalabilitat i la redundància dels serveix de base de dades. Per exemple si en cas de fallada d'un servidor, els altres poden seguir funcionant sense interrupcions, minimitzant el temps d'inactivitat.

7.- Quines diferències fonamentals veus entre un clúster d'alta disponibilitat i una replicació de servidors màster-slave?

Replicació Master-Slave: El master processa l'escriptura i modificacions en la base de dades mentre que el slave només és per la lectura. Si hagués alguna fallada hauriem de solucionar manualment i hauria causes de interrupcions.

Cluster Alta Disponibilitat: Es una replicacio entre els maquinari o nodes en que cada node poden ser el primari i els canvis realitzats són aplicats per totes. Si hagués alguna falla en el node primari no passaria res perquè el node secondary faria el treball del primari.

8.- Que passa si cau una màquina del clúster? i si cauen dues?

Si cau una maquina del cluster els altres nodes poden seguir fent el treball i operant normalment assegurant la consistència i l'alta disponibilitat. Si per algun cas cauen 2 màquines o més, depenent de la configuració específica del cluster podria entrar en mode de només lectura o deixar de funcionar.

9.- Es pot fer el mateix amb altres tipus de servidors? Amb quins? Posa exemples

Si, es pot fer el mateix amb altres tipus de servidors.

Servidor de fitxers: Utilitzar cluster de GlusterFS o Ceph per alta disponibilitat de fitxers.

Servidors web: Utilitzar cluster de servidors web amb HAProxy o Nginx per distribuir la carrega i assegurar la disponibilitat.

Servidors d'aplicacions: Cluster de Kubernetes per desplegar i gestionar aplicacions en contenidors amb alta disponibilitat.

10.- Explica de forma resumida:

a) Per a què serveix aquest clúster.

El clúster MySQL InnoDB serveix per oferir una base de dades que es pot repartir entre diferents servidors i que sempre estigui disponible. Està fet per assegurar que la base de dades sigui segura i funcioni bé si algun dels servidors falla ja que sincronitza totes les dades.

b) Quins son els 5 punts més importants a l'hora de crear-lo

Configuració dels servidors: S'ha de tenir la mateixa versió de MySQL, la mateixa configuració perquè no produeixin errors i que estiguin configurats per treballar juntament.

Replica de dades: Totes les nodes han de tenir la mateixa informació, configurar el sistema perquè faci còpies de dades entre els servidors.

Creació del Clúster: Configurar els fitxers i utilitzar les eines de MySQL per crear i connectar els servidors tot en un sol cluster.

Monitorització: Utilitzar eines per veure l'estat del clúster o servidor i assegurar que tot funciona bé.

Proves de Fallades: Realitzar proves de falles de node i a veure si segueix funcionant el sistema.

c) Que és l'alta disponibilitat segons les teves paraules i que has llegit a l'article

Que el sistema sempre estigui actiu i disponible de forma que si surt algún error del servidor segueix funcionant i que no tingui tampoc aturades ni pèrdues de informació, en cas greu pugui retornar en el estat funcional.

Oracle MySQL HeatWave

Explica que significa aquesta tecnologia

MySQL Heatwave és un servei integrat dins de la plataforma Oracle que combina MySQL. Aquest servei executa consultes analítiques grans amb alta velocitat combinat amb avantatges d'un sistema anomenat Online Transaction Processing(OLTP) amb funcions Online Analytical Processing(OLAP).

Quina relació existeix entre el llenguatge SQL i el llenguatge de comunicació natural per a la resolució de consultes.

El llenguatge SQL és un llenguatge per gestionar i manipular la base de dades relacionals. SQL permet als usuaris mostrar, definir i manipular dades de manera estructurada. En canvi el llenguatge de comunicació natural és més flexible i menys estructurat però pot ser interpretat per sistemes avançats per traduir-lo a consultes SQL que permet als usuaris fer consultes complexes utilitzant termes més accessibles.

De quina forma permet Oracle als usuaris treballar amb MySQL HeatWave

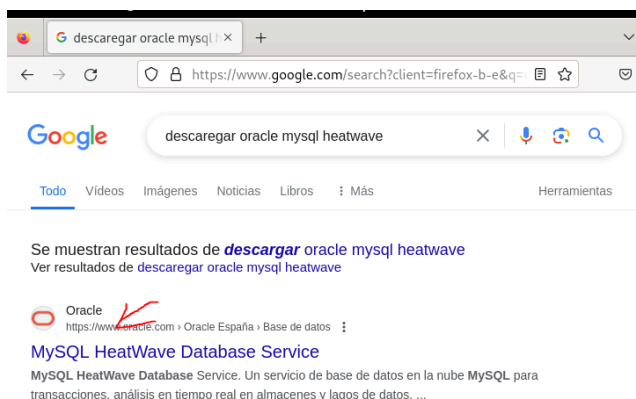
Oracle permet als usuaris treballar amb MySQL HeatWave a través de la seva plataforma de núvol Oracle Cloud Infrastructure (OCI). Els usuaris poden crear instàncies de MySQL HeatWave des del panell d'OCI on permet configurar i gestionar la base de dades amb facilitat. Proporciona eines per migrar dades, monitoritzar el rendiment i optimitzar les consultes.

Es pot descarregar el servidor en local per a fer-lo servir?

No, MySQL HeatWave està dissenyat per ser utilitzat en el núvol a través de Oracle Cloud Infrastructure (OCI).

- **Descarrega una instància i fes la prova en una MV**

Anar a la pàgina oficial per fer la descàrrega de MySQL HeatWave, he buscat en altres llocs però són de paga.



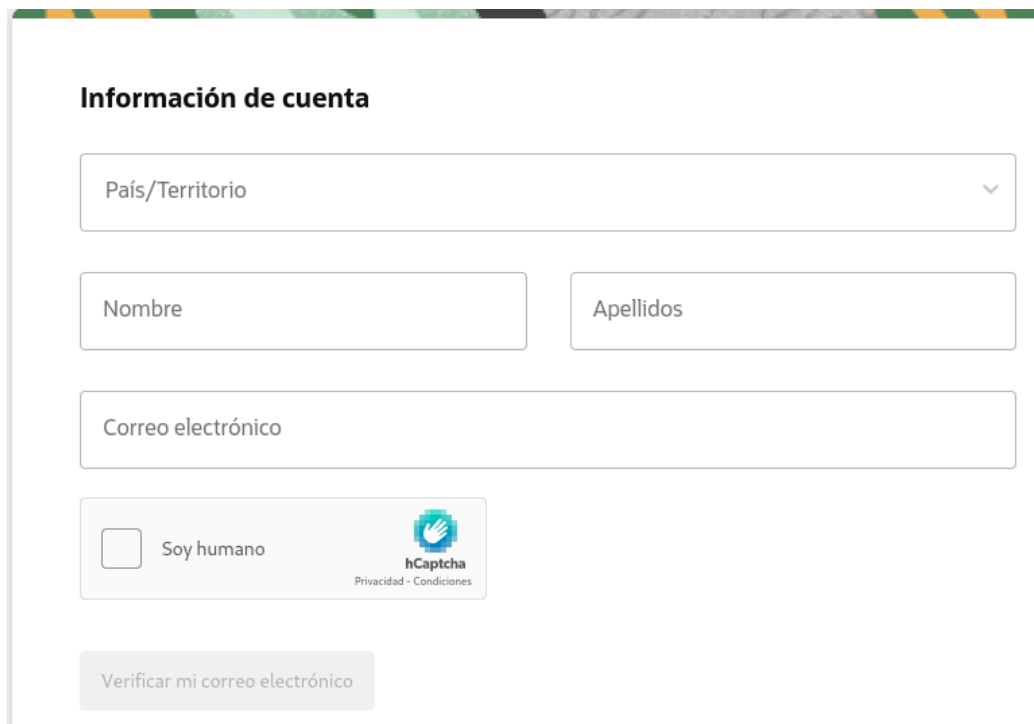
Li donem a començar gratis



Els 2 opcions son gratuïts temporalment pero utilitzarem el OCI (oracle Cloud Infrastructure)



En entrar ens pedirà dades que hem de completar i verificar.



The screenshot shows a registration form titled "Información de cuenta". It includes a dropdown menu for "País/Territorio", input fields for "Nombre" and "Apellidos", and a "Correo electrónico" field. Below these is a checkbox for "Soy humano" next to an hCaptcha logo and the text "Privacidad - Condiciones". At the bottom is a button labeled "Verificar mi correo electrónico".

Al final tampoc ha pogut descarregar ja que tenim que posar un mètode de pagament amb informació personal.

Address Information ^

Payment/Identity Verification - You will not be charged v

You won't be charged for using OCI Free Tier. If you choose to upgrade your account to Paid tier, you will only be charged for services used over your Always Free limits. If you don't upgrade your account to Paid tier, you will continue to have access to Always Free services.

You may see a small, temporary charge on your payment method. This is a verification hold that will be removed automatically. See the [FAQ](#) for more information.

Add payment verification method

Oracle uses third-party payment processor CyberSource for Oracle Store payment processing. CyberSource will request and collect certain information as part of the payment processing. Please refer to CyberSource's privacy statement at: <http://www.cybersource.com/privacy> for the terms applicable to the data collected..

De quines formes podem fer servir MySQL HeatWave?

MySQL HEatWave es pot utilitzar de diverses maneres:

1. **Anàlisi de dades en temps real:** Permet executar consultes analítiques en temps real sense necessitat de moure dades a un sistema OLAP separat.
2. **Aplicacions híbrides:** Combina càrregues de treball OLTP i OLAP en una sola base de dades, simplificant l'arquitectura de l'aplicació.
3. **Escalabilitat:** Escala automàticament segons la càrrega de treball que permet gestionar un gran volum de dades amb eficiència.

Quines avantatges ens facilita l'ús d'aquesta tecnologia.

- **Explica-ho amb exemples.**

1. **Rendiment alt:** La seva arquitectura en memòria i paral·lelització permet processar grans volums de dades ràpidament.

Exemple: Suposem que una empresa de comerç electrònic necessita generar informes diaris sobre les vendes de milions de productes. Sense HeatWave, aquest procés podria trigar hores. Amb MySQL HeatWave amb la seva capacitat de paral·lelització i processament en memòria, aquests informes es poden generar en minuts.

2. **Simplicitat:** Permet utilitzar la mateixa base de dades per a transaccions en línia (OLTP) i anàlisi (OLAP), eliminant la necessitat de mantenir dos sistemes separats.

Exemple: Una empresa de logística utilitza MySQL HeatWave per gestionar les operacions diàries de seguiment d'enviaments i per analitzar les rutes de lliurament. Amb HeatWave, no necessiten sistemes separats per gestionar operacions i analitzar dades, facilitant prendre decisions en temps real.

3. **Cost-efectivitat:** Redueix els costos d'operacions al consolidar les funcions de base de dades i anàlisi de la plataforma. Permet estalviar costos associats amb la gestió i manteniment de múltiples sistemes incluint costos d'infraestructura, llicències de programari i la gestió de dades duplicades.

Exemple: Una startup tecnològica utilitza MySQL HeatWave per a la seva aplicació de xarxes socials, la qual necessita publicacions, likes i comentaris com anàlisi de dades. Amb HeatWave, estalvien costos en infraestructura i llicències, ja que no necessiten sistemes separats.

POSTGRESQL

INSTAL·LACIÓ

Per instal·lar el PostgreSQL en Debian 12 hem de posar la comanda 'sudo apt install postgresql' en la terminal

```
usuari@debian12:~$ sudo apt install postgresql
[sudo] contrasenya para usuari:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  libcommon-sense-perl libjson-perl libjson-xs-perl
```

Després de finalitzar la instal·lació activarem el seu servei 'systemctl start postgresql.service' veurem el seu estat amb la comanda 'systemctl status postgresql.service'

```
usuari@debian12:~$ systemctl start postgresql.service
usuari@debian12:~$ systemctl status postgresql.service
• postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; preset: e
   Active: active (exited) since Mon 2024-05-20 12:34:34 CEST; 10min ago
   Main PID: 7464 (code=exited, status=0/SUCCESS)
   CPU: 447us
lines 1-5/5 (END)
```

Per veure la versió de postgresql utilitzarem la comanda 'psql --version'.

```
usuari@debian12:~$ psql --version
psql (PostgreSQL) 15.6 (Debian 15.6-0+deb12u1)
usuari@debian12:~$
```

CONFIGURACIÓ

Entrarem dins de la base de dades amb la comanda 'sudo -u postgres psql', postgres ens dona un usuari de base de dades predeterminat anomenat postgres

```
usuari@debian12:~$ sudo -u postgres psql
could not change directory to "/home/usuari": Permiso denegado
psql (15.6 (Debian 15.6-0+deb12u1))
Type "help" for help.

postgres=#
```

Per veure les databases s'utilitza la comanda '\l'

```
postgres=# \l
          List of databases
  Name      | Owner   | Encoding | Collate | Ctype   | ICU Locale | Locale Provider | Access privileges
-----+-----+-----+-----+-----+-----+-----+-----
 postgres   | postgres | UTF8     | es_ES.UTF-8 | es_ES.UTF-8 |              | libc              |
 template0  | postgres | UTF8     | es_ES.UTF-8 | es_ES.UTF-8 |              | libc              | =c/postgres      +
 template1  | postgres | UTF8     | es_ES.UTF-8 | es_ES.UTF-8 |              | libc              | =c/postgres      +
              |          |          |          |          |          |          | postgres=CTc/postgres
(3 rows)

postgres=#
```

Crearem una base de dades anomenada starwars

```
postgres=# CREATE DATABASE starwars;
CREATE DATABASE
postgres=# \l
          List of databases
  Name      | Owner   | Encoding | Collate | Ctype   | ICU Locale | Locale Provider | Access privileges
-----+-----+-----+-----+-----+-----+-----+-----
 postgres   | postgres | UTF8     | es_ES.UTF-8 | es_ES.UTF-8 |              | libc              |
 starwars   | postgres | UTF8     | es_ES.UTF-8 | es_ES.UTF-8 |              | libc              |
 template0  | postgres | UTF8     | es_ES.UTF-8 | es_ES.UTF-8 |              | libc              | =c/postgres      +
 template1  | postgres | UTF8     | es_ES.UTF-8 | es_ES.UTF-8 |              | libc              | =c/postgres      +
              |          |          |          |          |          |          | postgres=CTc/postgres
(4 rows)
```

Crearem un nou usuari anomenat bixi amb tots els privilegis de la base de dades creada.

```
usuari@debian12:~$ sudo -u postgres psql
could not change directory to "/home/usuari": Permiso denegado
psql (15.6 (Debian 15.6-0+deb12u1))
Type "help" for help.

postgres=# CREATE ROLE bixi WITH LOGIN PASSWORD '@MVM2016';
CREATE ROLE
```

També li donarem tots els privilegis de la base de dades creada

```
postgres=# GRANT ALL PRIVILEGES ON DATABASE starwars TO bixi;
GRANT
postgres=# exit
usuari@debian12:~$
```

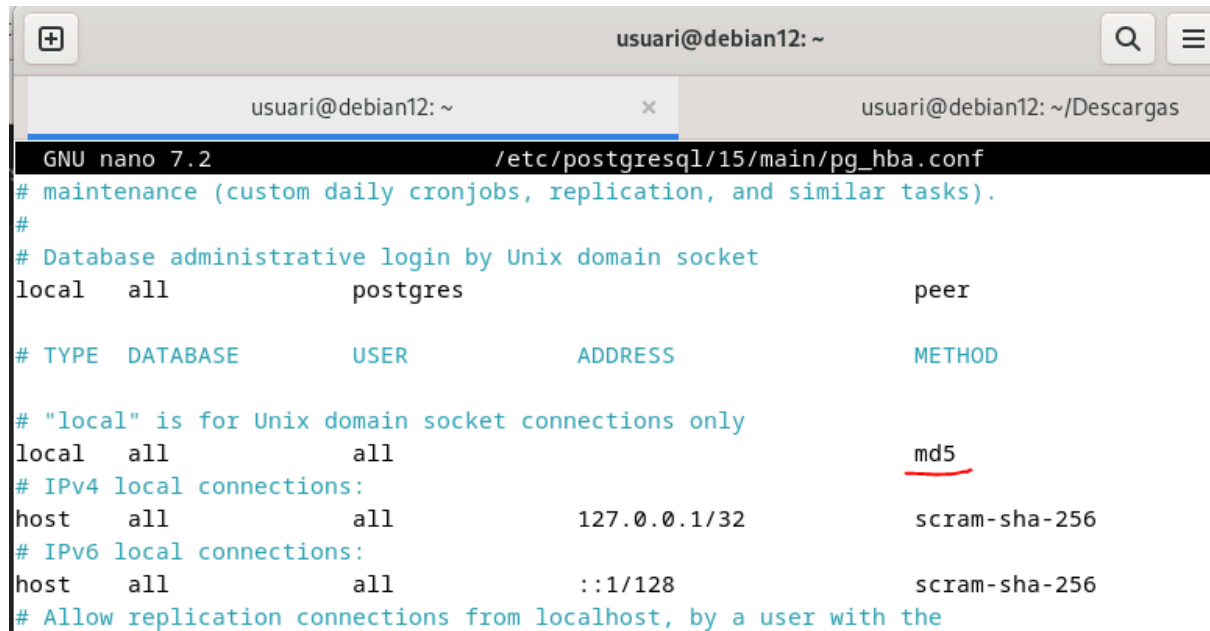
Ens surt error perquè intenta verificar que el nom d'usuari creat sigui el mateix que l'usuari de la màquina.

```
usuari@debian12:~$ psql -U bixi -d starwars
psql: error: falló la conexión al servidor en el socket «/var/run/postgresql/.s.PGSQL.5432»: FATAL:
la autenticación Peer falló para el usuario «bixi»
```

Per solucionar modificarem l'arxiu '/etc/postgresql/15/main/pg_hba.conf' amb un `sudo nano`

```
usuari@debian12:~$ sudo nano /etc/postgresql/15/main/pg_hba.conf
```

Cambiarem en el paràmetre local de peer a md5 com mostra la imatge:



```
GNU nano 7.2 /etc/postgresql/15/main/pg_hba.conf
# 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                  md5
# 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
```

Farem un reinici perquè actualitza els canvis realitzats en el servidor amb la comanda '`sudo systemctl restart postgresql`'

```
usuari@debian12:~$ sudo systemctl restart postgresql
usuari@debian12:~$
```

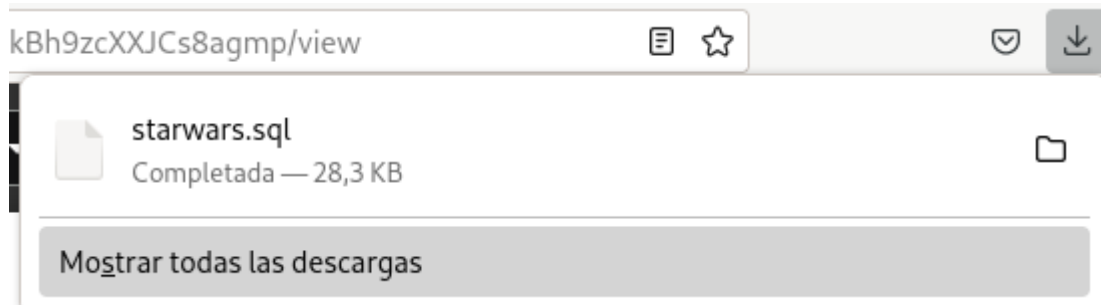
Tornarem a intentar entrar i ja ens hauria de deixar

```
usuari@debian12:~$ psql -U bixi -d starwars
Contraseña para usuario bixi:
psql (15.6 (Debian 15.6-0+deb12u1))
Digite «help» para obtener ayuda.

starwars=>
```

INTRODUCCIÓ .SQL

Descarregarem el fitxer .sql



Introduïrem el .sql dins de la base de dades starwars que hem assignat a l'usuari bixi.

ERROR INTRODUCCIÓ .SQL

Hem surten errors de sintaxis, he canviat totes les comilles per res, doble comilles i hem surt altres errors.

```
usuari@debian12:~$ psql -U bixi -d starwars < ~/Descargas/starwars.sql
Contraseña para usuario bixi:
ERROR:  error de sintaxis en o cerca de «`»
LÍNEA 1: DROP TABLE IF EXISTS `affiliations`;
                        ^

ERROR:  error de sintaxis en o cerca de «`»
LÍNEA 1: CREATE TABLE `affiliations` (
                        ^

ERROR:  error de sintaxis en o cerca de «`»
LÍNEA 1: LOCK TABLES `affiliations` WRITE;
                        ^

ERROR:  error de sintaxis en o cerca de «`»
LÍNEA 1: INSERT INTO `affiliations` VALUES (0,'Without Affiliation'),...
                        ^

usuari@debian12:~$ psql -U bixi -d starwars < ~/Descargas/starwars.sql
Contraseña para usuario bixi:
NOTICE:  la tabla «affiliations» no existe, omitiendo
DROP TABLE
ERROR:  error de sintaxis en o cerca de «ENGINE»
LÍNEA 5: ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;
        ^

ERROR:  error de sintaxis en o cerca de «"affiliations"»
LÍNEA 1: LOCK TABLES "affiliations" WRITE;
        ^

ERROR:  no existe la relación «affiliations»
LÍNEA 1: INSERT INTO "affiliations" VALUES (0,'Without Affiliation'),...
        ^

ERROR:  error de sintaxis en o cerca de «UNLOCK»
LÍNEA 1: UNLOCK TABLES;
        ^

NOTICE:  la tabla «deaths» no existe, omitiendo
DROP TABLE
ERROR:  error de sintaxis en o cerca de «AUTO_INCREMENT»
LÍNEA 2:  "id" int NOT NULL AUTO_INCREMENT,
                        ^
```

He creat un arxiu per crear les taules i m'ha sortit errors.

```
usuari@debian12:~$ psql -U bixi -d starwars < ~/Descargas/starwars/create.sql
Contraseña para usuario bixi:
ERROR:  permiso denegado al esquema public
LÍNEA 2: CREATE TABLE IF NOT EXISTS personas(
      ^
ERROR:  permiso denegado al esquema public
LÍNEA 1: CREATE TABLE IF NOT EXISTS prova(
      ^
ERROR:  permiso denegado al esquema public
LÍNEA 1: CREATE TABLE IF NOT EXISTS prova2(
      ^
ERROR:  permiso denegado al esquema public
LÍNEA 1: CREATE TABLE IF NOT EXISTS prova3(
```

SOLUCIÓ

Com que fallaba en inserir les taules li he donat totes les permisos tant schema, seqüències, públic com el superusuari. També he intentat tornar a inserir .sql de star wars però tampoc funciona.

```
usuari@debian12:~$ sudo -u postgres psql
[sudo] contraseña para usuari:
could not change directory to "/home/usuari": Permiso denegado
psql (15.6 (Debian 15.6-0+deb12u1))
Type "help" for help.

postgres=# GRANT ALL PRIVILEGES ON DATABASE starwars TO bixi;
GRANT
postgres=# GRANT USAGE, CREATE ON SCHEMA public TO bixi;
GRANT
postgres=# GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO bixi;
GRANT
postgres=# GRANT ALL PRIVILEGES ON ALL SEQUENCES IN SCHEMA public TO bixi;
GRANT
postgres=# ALTER DEFAULT PRIVILEGES IN SCHEMA public GRANT ALL PRIVILEGES ON TABLES TO bixi;
ALTER DEFAULT PRIVILEGES
postgres=# ALTER DEFAULT PRIVILEGES IN SCHEMA public GRANT ALL PRIVILEGES ON SEQUENCES TO bixi;
ALTER DEFAULT PRIVILEGES
postgres=#

postgres=# ALTER ROLE bixi WITH SUPERUSER;
ALTER ROLE
postgres=#
```

Després de donar els permisos s'ha pogut inserir perfectament les taules creades i els inserts

```
usuari@debian12:~$ psql -U bixi -d starwars < ~/Descargas/starwars/create.sql
Contraseña para usuario bixi:
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
```

```
usuari@debian12:~$ psql -U bixi -d starwars < ~/Descargas/starwars/insert.sql
Contraseña para usuario bixi:
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
usuari@debian12:~$
```

Com que no existeix SHOW TABLES hem de utilitzar la següent comanda per poder veure les taules: `SELECT table_name FROM information_schema.tables WHERE table_schema = 'public' AND table_type = 'BASE TABLE';`

```
usuari@debian12:~$ psql -U bixi -d starwars
Contraseña para usuario bixi:
psql (15.6 (Debian 15.6-0+deb12u1))
Digite «help» para obtener ayuda.

starwars=# SELECT table_name FROM information_schema.tables WHERE table_schema
= 'public' AND table_type = 'BASE TABLE';
 table_name
-----
 personas
  prova
  prova2
  prova3
(4 filas)

starwars=#
```

I ja després podem fer els selects normals

```
starwars=# SELECT * FROM personas;
 id_persones |      nom
-----+-----
          1 | Juan Pérez
          2 | María López
          3 | Carlos García
(3 filas)

starwars=#
```

```
starwars=# SELECT * FROM prova;
id_prova | nom
-----+-----
          1 | Prueba 1
          2 | Prueba 2
          3 | Prueba 3
(3 filas)

starwars=# █

starwars=# SELECT * FROM prova2;
id_prova2 | nom
-----+-----
          1 | Prueba2 1
          2 | Prueba2 2
          3 | Prueba2 3
(3 filas)

starwars=# SELECT * FROM prova3;
id_prova3 | nom
-----+-----
          1 | Prueba3 1
          2 | Prueba3 2
          3 | Prueba3 3
(3 filas)

starwars=#
```