

Este trabajo tiene licencia CC BY-NC-SA 4.0. Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by-nc-sa/4.0/>

U8P01-Amazon EC2. Despliegue de un SGBD no administrado-MariaDB

1. Descripción

En un servicio de bases de datos **no administrado**, el administrador es responsable de **instalar, configurar y mantener todo lo relacionado con el sistema gestor de bases de datos base de datos**. Esto significa gestionar el servidor, el software y cualquier tarea operativa relacionada.

En esta práctica se instalará **MariaDB Server** en una instancia de **Amazon EC2** y se configurará para poder establecer conexiones desde un cliente externo.

Nota: no es una buena práctica de seguridad exponer una base de datos a internet. Lo adecuado es crearla en una subred privada y que sea accesible solo desde las instancias/servicios que necesiten usarla. En esta práctica se realiza con fines didácticos.

2. Formato de entrega

- Realiza la opción que indique el profesor
 - MOSTRAR: muestra en la clase tu configuración al profesor.
 - ENTREGAR: Entrega un documento que incluya capturas de pantalla de los diferentes pasos realizados a lo largo de la práctica.

3. Trabajo a realizar

3.1. Desplegar una instancia

1. Inicia su entorno AWS *Learner Lab* y accede a la consola de administración de AWS.
2. Despliega una instancia en **AWS EC2** con las siguientes propiedades.
 - Nombre: **zeldaXX-A** (XX es tú numero en el aula).
 - Imágenes de aplicaciones y sistemas operativos (Imagen de máquina de Amazon): **Debian 12 para arquitectura x86 (64 bits)**. Observa que es **Apta para la capa gratuita**.
 - Tipo de instancia: **tipo t2.micro (1 CPU virtual y 1 GiB de memoria)** , que también es **apta para la capa gratuita**.
 - Par de claves: **vockey**.
 - Configuraciones de red:
 - VPC: **VPC Predeterminada**
 - Subred : la **subred** asignada a la **zona de disponibilidad us-east-1a**
 - Asignar automáticamente la IP pública: **Habilitar**.

- **Firewall (grupos de seguridad):**
 - Nombre del grupo de seguridad: **sg_BD-XX**
 - Descripción: **Acceso a las instancias de BD**
 - **Regla para permitir el acceso desde cualquier lugar al puerto 22/TCP.**
- **Configurar almacenamiento:** volumen de 8 GB.

3. Consulta la **subred en la que se ha creado la instancia y por lo tanto en que zona de disponibilidad** en la que se ha creado la instancia.

3.2. Instalación de MariaDB

1. Accede por SSH a la instancia.
2. Actualiza la lista de paquetes disponibles en los repositorios.

```
sudo apt update
```

3. Instala el SGBD

```
sudo apt install mariadb-server
```

4. Habita el servicio para que se inicia cada vez que inicie el sistema.

```
sudo systemctl enable mariadb.service
```

5. Asegurate de que mariadb se ha iniciado y escucha en el puerto **3306/TCP en local (127.0.0.1)**.

```
sudo systemctl status mariadb.service  
sudo ss -ltn
```

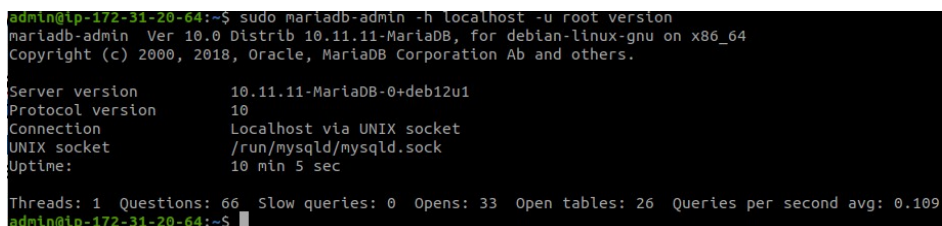
3.3. Conexión a la al servidor de bases de datos

Al instalar el servidor también se han instalado un conjunto de aplicaciones cliente.

Por defecto mariadb esta configurado para que te puedas conectar como usuario **root** (el usuario administrador por defecto de mariadb) si se cumple la siguiente condición. El cliente se tiene que ejecutare en la misma maquina donde está instalado el servidor y se tiene que ejecutar con el usuario administrador de la máquina (para ello en nuestro caso hay que ejecutar el cliente con sudo para hacerlo como usuario root).

1. Ejecuta el siguiente comando para verificar que mysql se está ejecutando y su versión.

```
sudo mariadb-admin -h localhost -u root version
```



```
admin@ip-172-31-20-64:~$ sudo mariadb-admin -h localhost -u root version  
mariadb-admin Ver 10.0 Distrib 10.11.11-MariaDB, for debian-linux-gnu on x86_64  
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.  
  
Server version          10.11.11-MariaDB-0+deb12u1  
Protocol version        10  
Connection              Localhost via UNIX socket  
UNIX socket              /run/mysqld/mysqld.sock  
Uptime:                  10 min 5 sec  
  
Threads: 1 Questions: 66 Slow queries: 0 Opens: 33 Open tables: 26 Queries per second avg: 0.109  
admin@ip-172-31-20-64:~$
```

2. Establece una conexión como usuario root

```
sudo mysql -h localhost -u root
```

-h → servidor

-u → usuario del servidor

Observa que ejecutas el cliente con sudo para que se haga con permisos de administrador y que no se pide contraseña al iniciar sesión con el usuario de **root** de mariadb.

```
admin@ip-172-31-20-64:~$ sudo mysql -h localhost -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 41
Server version: 10.11.11-MariaDB-0+deb12u1 Debian 12

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> 
```

3. Ejecuta `show databases;` para mostrar los esquemas de bases de datos disponibles

```
MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.000 sec)

MariaDB [(none)]> 
```

4. Cierra la conexión con el comando `exit`.

3.4. Conexiones desde clientes remotos

Por defecto el servidor solo atiende conexiones desde clientes locales. Vamos a configurarlo para que atienda conexiones de clientes remotos.

1. Verifica que el servidor mysql solo escucha peticiones el puerto **3306/TCP** del interfaz **localhost (127.0.0.1)**.

```
ss -ltn
```

```
admin@ip-172-31-20-64:~$ ss -ltn
State      Recv-Q    Send-Q    Local Address:Port      Peer Address:Port      Process
LISTEN     0          4096      127.0.0.54:53           0.0.0.0:*               nssd
LISTEN     0          4096      127.0.0.53%lo:53        0.0.0.0:*               nssd
LISTEN     0          80       127.0.0.1:3306          0.0.0.0:*               mysqld
LISTEN     0          128      0.0.0.0:22              0.0.0.0:*               sshd
LISTEN     0          4096      0.0.0.0:5355            0.0.0.0:*               nssd
LISTEN     0          128      [::]:22                 [::]:*                  nssd
LISTEN     0          4096      [::]:5355               [::]:*                  nssd
```

2. Edita el archivo de configuración de MariaDB para permitir conexiones desde direcciones IP específicas, busca la línea que comienza con `bind-address` y cámbiala para que acepte conexiones en todos los interfaces de red del equipo (`0.0.0.0`).

```
sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf
```

```
#user                = mysql
pid-file             = /run/mysqld/mysqld.pid
basedir              = /usr
#datadir              = /var/lib/mysql
#tmpdir               = /tmp

# Broken reverse DNS slows down connections considerably and name resolve is
# safe to skip if there are no "host by domain name" access grants
#skip-name-resolve

# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address          = 0.0.0.0

#
# * Fine Tuning
#
```

3. Reinicia el servidor para que se aplique la nueva configuración.

```
sudo systemctl restart mariadb
```

4. Asegurate de que mysql se ha iniciado y escucha en el puerto **3306/TCP** en **todo los interfaces de red (0.0.0.0)**

```
sudo systemctl status mariadb.service
sudo ss -ltn
```

```
admin@ip-172-31-20-64:~$ sudo ss -ltn
```

State	Recv-Q	Send-Q	Local Address:Port	Peer Address:Port	Process
LISTEN	0	4096	127.0.0.54:53	0.0.0.0:*	
LISTEN	0	4096	127.0.0.53%lo:53	0.0.0.0:*	
LISTEN	0	128	0.0.0.0:22	0.0.0.0:*	
LISTEN	0	80	0.0.0.0:3306	0.0.0.0:*	
LISTEN	0	4096	0.0.0.0:5355	0.0.0.0:*	
LISTEN	0	128	:::22	:::*	
LISTEN	0	4096	:::5355	:::*	

```
admin@ip-172-31-20-64:~$
```

3.5. Crear una base de datos y un usuario

Ahora vamos a crear una base de datos y un usuario que tenga todo los permisos sobre la base de datos creada.

1. Establece una conexión como usuario root con el servidor de bases de datos.

```
sudo mariadb -h localhost -u root
```

2. Ejecuta la siguiente sentencia SQL para crear la base de datos (esquema) juegos.

```
MariaDB [(none)]>create database juegos;
```

3. Ejecuta las siguientes sentencias SQL para crear el usuario **juegos** con password **juegos** tanto para conexiones locales como conexiones desde cualquier origen.

```
MariaDB [(none)]>create user juegos@'localhost' identified by 'juegos';  
MariaDB [(none)]>create user juegos@'%' identified by 'juegos';
```

4. Ejecuta las siguientes sentencias SQL para asignarle todos los privilegios al usuario juegos sobre la base de datos juegos.

```
MariaDB [(none)]>grant all privileges on juegos.* to juegos@'localhost';  
MariaDB [(none)]>grant all privileges on juegos.* to juegos@'%';
```

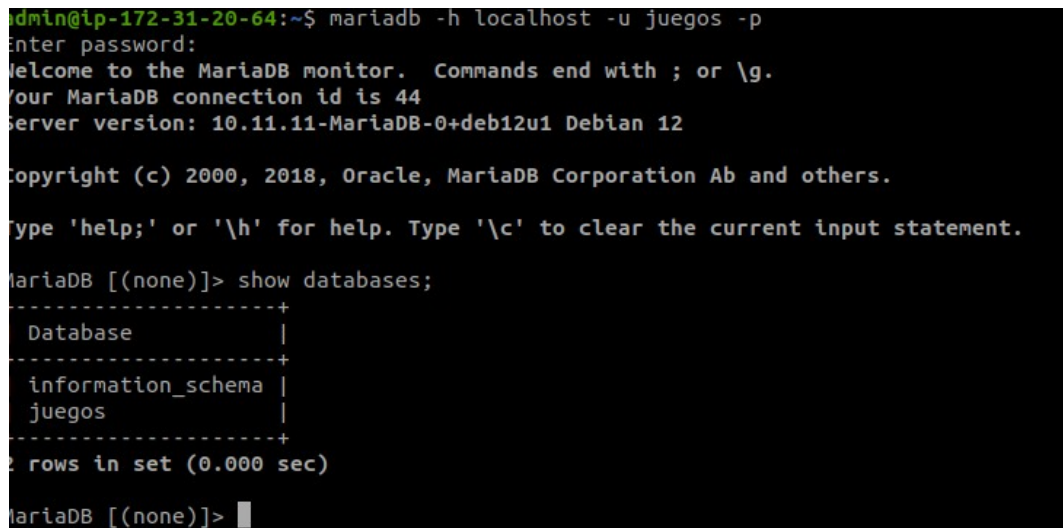
5. Cierra la conexión con `exit`.

6. Establece una conexión con el usuario **juegos**.

```
mariadb -h localhost -u juegos -p
```

7. Consulta las bases de datos a las que tiene acceso.

```
MariaDB [(none)]>show databases;
```



```
admin@ip-172-31-20-64:~$ mariadb -h localhost -u juegos -p  
Enter password:  
Welcome to the MariaDB monitor.  Commands end with ; or \g.  
Your MariaDB connection id is 44  
Server version: 10.11.11-MariaDB-0+deb12u1 Debian 12  
  
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
MariaDB [(none)]> show databases;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| juegos |  
+-----+  
2 rows in set (0.000 sec)  
  
MariaDB [(none)]> █
```

8. Cierra la conexión con `exit`.

3.6. Establecer conexiones desde clientes

3.6.1. Habilitar el tráfico en el grupo de seguridad

Para que clientes externos se puedan conectar al servidor MariaDB es necesario habilitar en el grupo de seguridad al que pertenece la instancia el tráfico de entrada hacia el puerto 3306/TCP.

1. Accede a la consola de EC2.
2. Crear una regla en el grupo de seguridad **sg_BD-XX** para permitir el tráfico de entrada hacia el puerto 3306/TCP desde cualquier origen (0.0.0.0/0). Observa que ya hay una regla predefinida con el nombre Aurora/Mysql que podemos usar.

Editar reglas de entrada [Información](#)

Las reglas de entrada controlan el tráfico entrante que puede llegar a la instancia.

ID de la regla del grupo de seguridad	Tipo	Protocolo	Intervalo de puertos	Origen	Descripción: opcional	
sgr-0afe634fd6fab7e9	MYSQL/Aurora	TCP	3306	Person...	0.0.0.0/0	Eliminar
sgr-0d705efa0b3519695	SSH	TCP	22	Person...	0.0.0.0/0	Eliminar

[Agregar regla](#)

3.6.2. Conexión desde otra instancia

Vamos a establecer una conexión desde otra instancia de EC2.

1. Despliega una instancia en **AWS EC2** con las siguientes propiedades.
 - Nombre: **linkXX-A** (XX es tú número en el aula).
 - Imágenes de aplicaciones y sistemas operativos (Imagen de máquina de Amazon): **Ubuntu Server 24.04 LTS para arquitectura x86 (64 bits)**. Observa que es **Apta para la capa gratuita**.
 - Tipo de instancia: **tipo t2.micro (1 CPU virtual y 1 GiB de memoria)**, que también es **apta para la capa gratuita**.
 - Par de claves: **vockey**.
 - Configuraciones de red:
 - VPC: **VPC Predeterminada**
 - Subred : la **subred** asignada a la **zona de disponibilidad us-east-1a**
 - Asignar automáticamente la IP pública: **Habilitar**.
 - **Firewall (grupos de seguridad)**:
 - Nombre del grupo de seguridad: **sg_SSH-XX**
 - Descripción: **Acceso SSH**
 - **Regla para permitir el acceso desde cualquier lugar al puerto 22/TCP.**
 - **Configurar almacenamiento**: volumen de 8 GB.

2. Accede por SSH a la instancia.

3. Instala un cliente de mariadb.

```
sudo apt update
sudo apt install mariadb-client
```

4. Establece una conexión contra el servidor de bases de datos con el usuario **juegos**.

```
mariadb -h <ip_privada_zeldaXX> -u juegos -p
```

```
ubuntu@ip-172-31-31-99:~$ mariadb -h 172.31.20.64 -u juegos -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 31
Server version: 10.11.11-MariaDB-0+deb12u1 Debian 12

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| juegos |
+-----+
2 rows in set (0.003 sec)

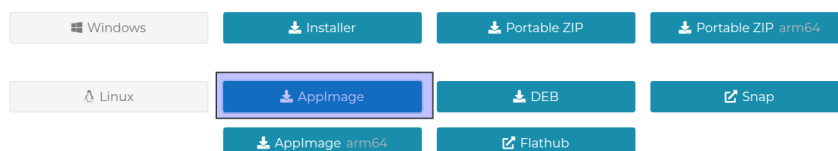
MariaDB [(none)]> █
```

3.6.3. Conexión desde tu máquina real

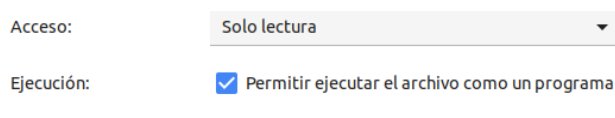
Vamos a establecer una conexión desde tu máquina real. Par ello puedes usar el cliente que consideres, existen muchas opciones que puedes instalar en Windows, Linux, y Mac, así como extensiones para IDEs como Visual Studio Code.

En esta práctica te propongo usar **Dbgate** (<https://dbgate.org/>) un cliente sencillo. Dispone de versiones para instalar y de versiones portables tanto para Windows como para Linux.

1. En tu máquina real accede a <https://dbgate.org/download/> y descarga la versión portable (en Linux, es la versión Appliance).

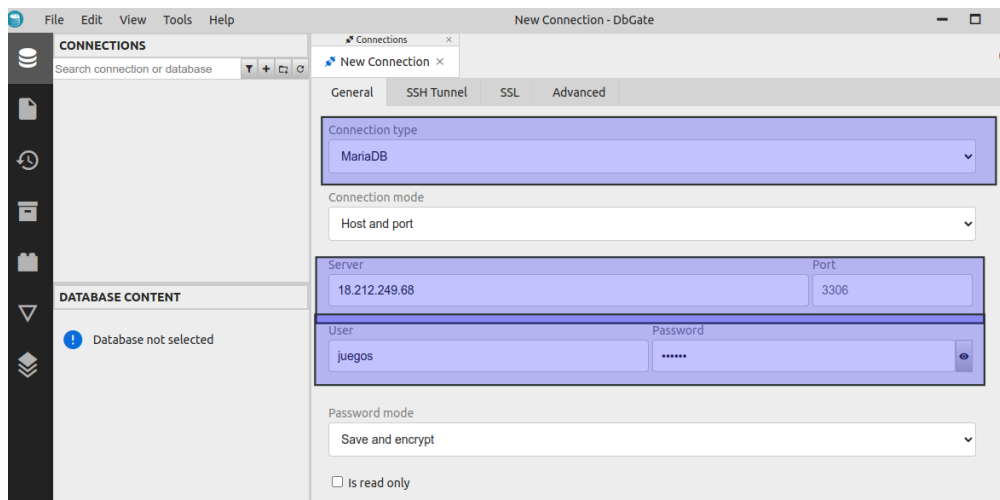


2. Ejecuta el cliente. En los equipos de clase previamente tienes que darle permisos de ejecución, botón derecho del ratón, Permisos y marcar la opción **“Permitir ejecutar el archivo como un programa”**.



3. Selecciona un tipo de conexión para **MariaDB** y configura los datos del servidor al que te vas a conectar.

- Server: <IP pública de zeldaXX-A>
- Usuario: **juegos**
- Password: **juegos**

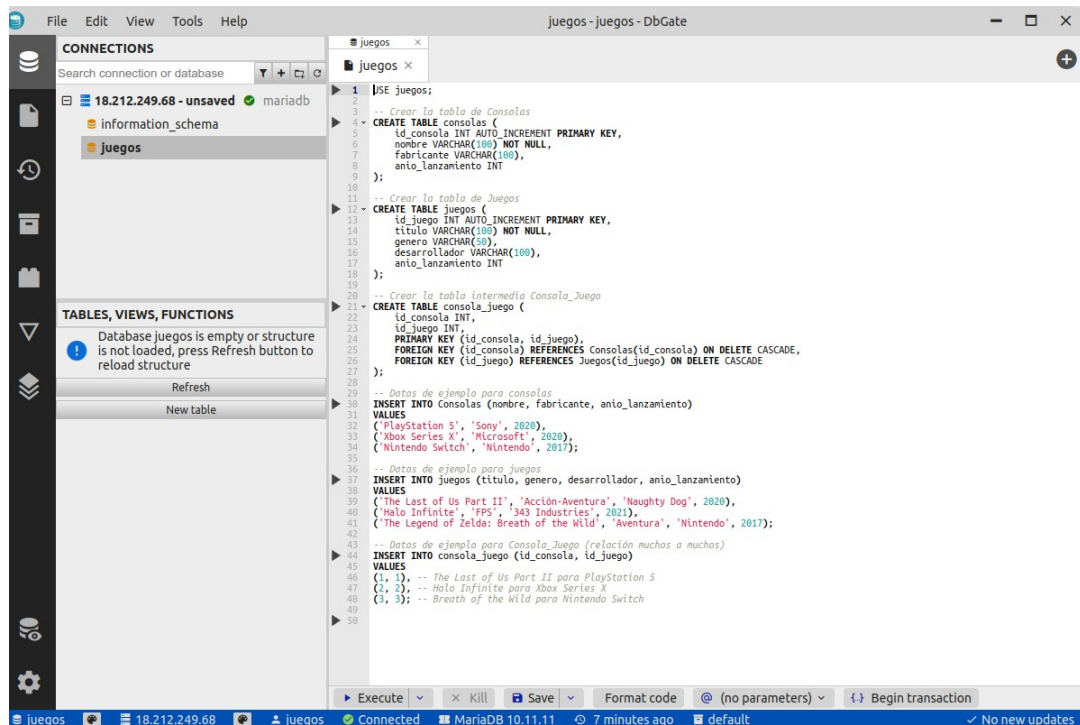


Establece la conexión pinchando en **Connect**.

4. Selecciona la base de datos **juegos**.

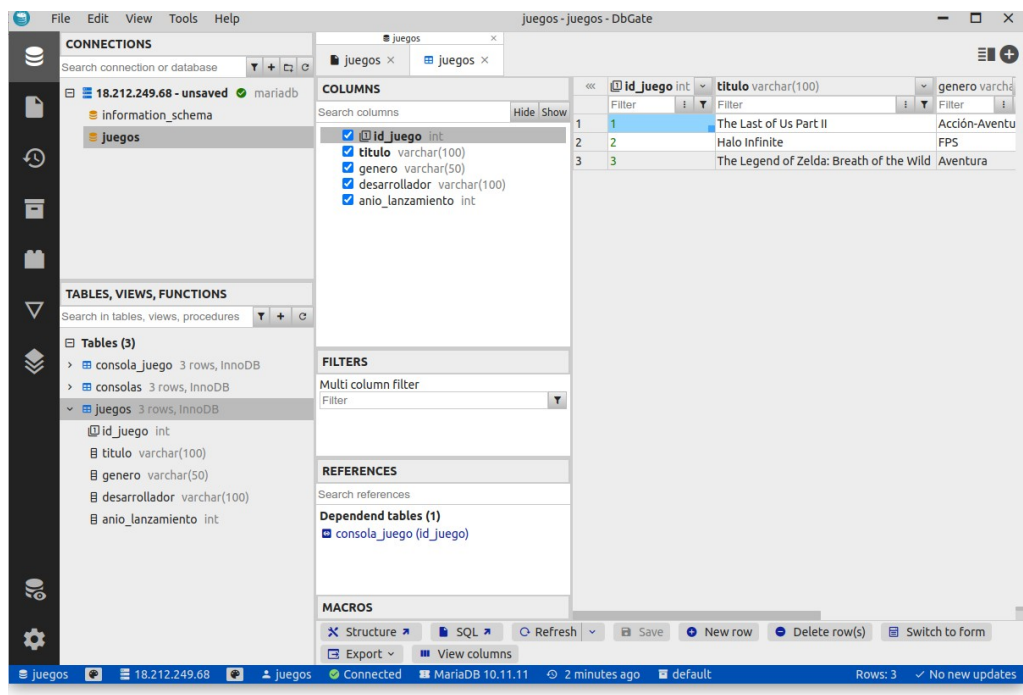


5. Abre el archivo **juegos.sql** que te proporcione el profesor con sentencias sql para crear tablas y datos de prueba.

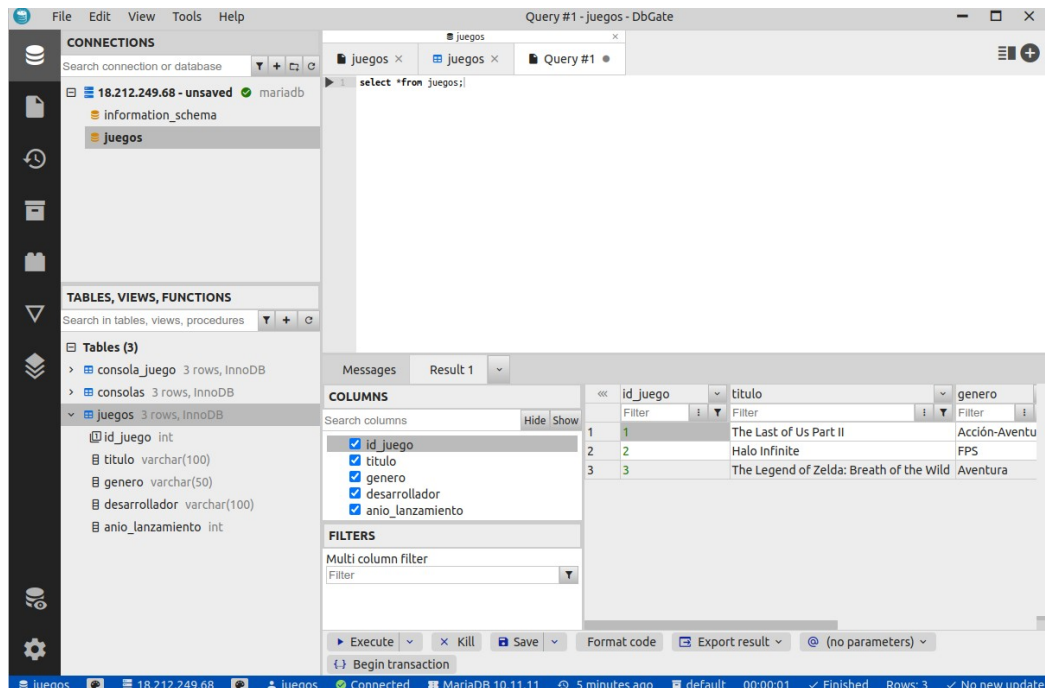


6. Ejecuta las sentencias con el botón **Execute**.

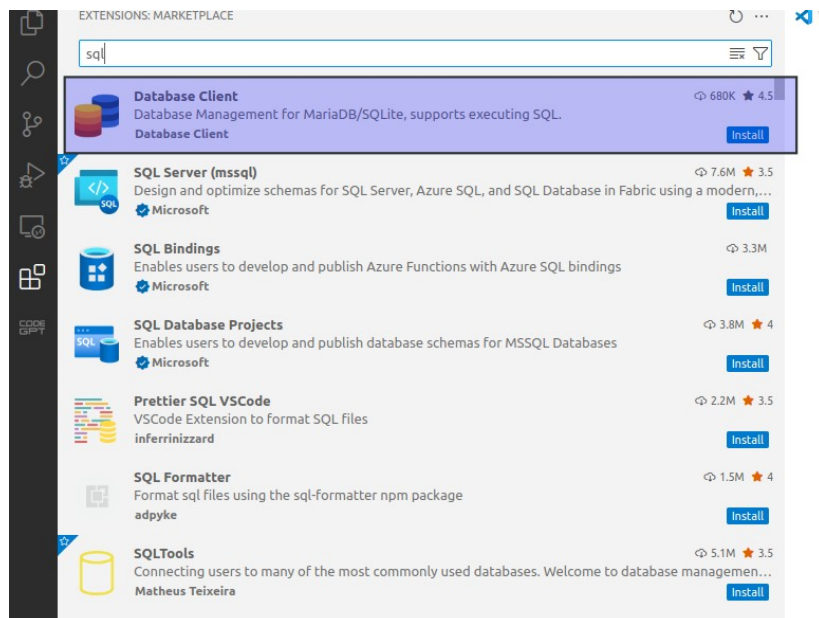
7. Refresca la vista para ver como se han creado las tablas y sus datos.

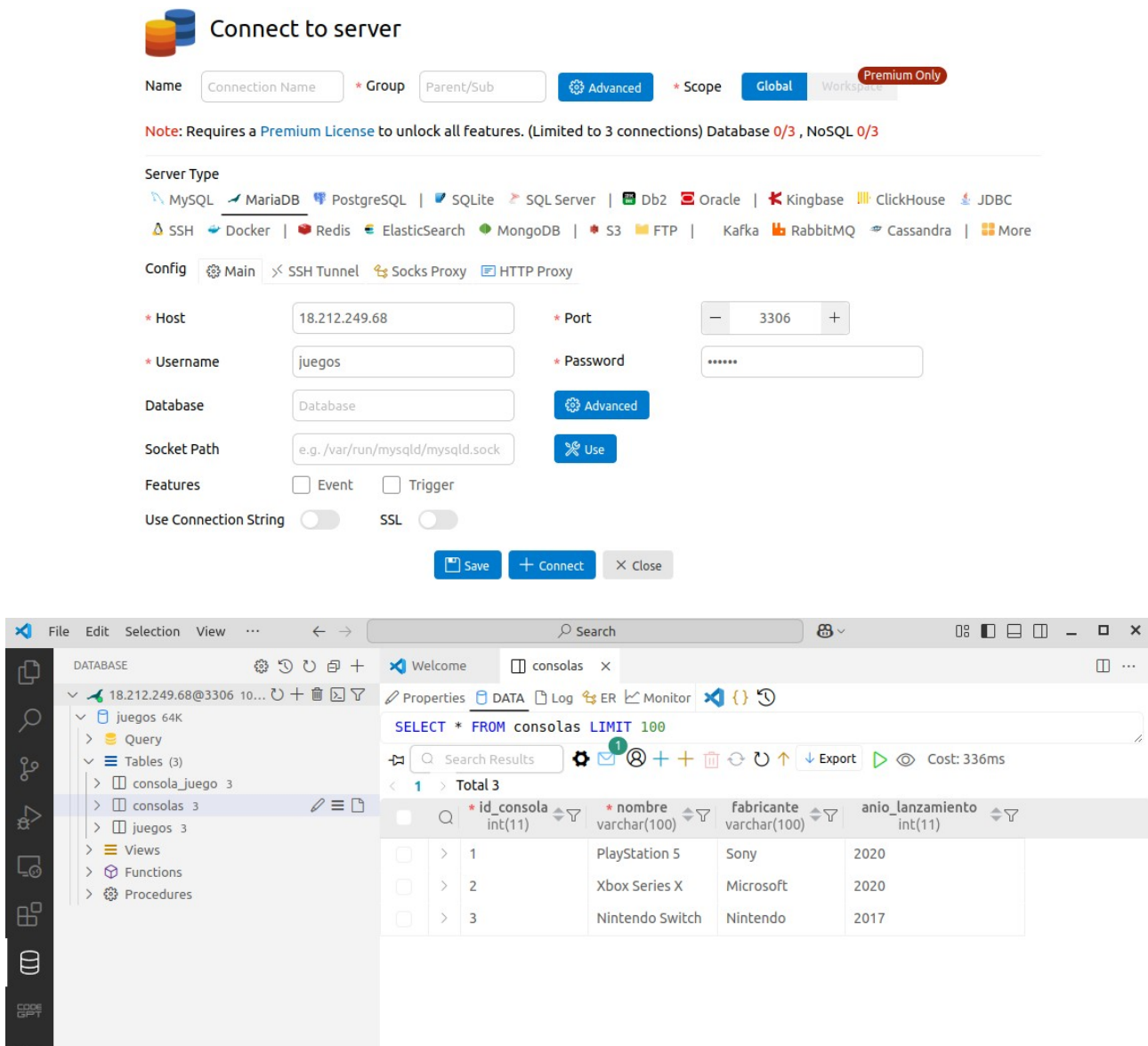


8. Prueba a hacer una consulta.



Puedes probar a usar alguna de las extensiones de **Visual Studio Code** que permiten actuar como cliente SQL, por ejemplo, **Database Client**.





Connect to server

Name: * Group: * Scope: Premium Only

Note: Requires a [Premium License](#) to unlock all features. (Limited to 3 connections) Database 0/3, NoSQL 0/3

Server Type

MySQL | MariaDB | PostgreSQL | SQLite | SQL Server | Db2 | Oracle | Kingbase | ClickHouse | JDBC

SSH | Docker | Redis | Elasticsearch | MongoDB | S3 | FTP | Kafka | RabbitMQ | Cassandra | More

Config ☒ Main ☐ SSH Tunnel ☐ Socks Proxy ☐ HTTP Proxy

* Host: * Port:

* Username: * Password:

Database:

Socket Path:

Features: ☐ Event ☐ Trigger

Use Connection String: ☐ SSL: ☐

Database Client Screenshot:

File Edit Selection View ... Search

Database: 18.212.249.68@3306 10...

Tree View:

- juegos 64K
 - Query
 - Tables (3)
 - consola_juego 3
 - consolas 3
 - juegos 3
 - Views
 - Functions
 - Procedures

Properties | DATA | Log | ER | Monitor

SELECT * FROM consolas LIMIT 100

Search Results

Cost: 336ms

	* id_consola int(11)	* nombre varchar(100)	fabricante varchar(100)	anio_lanzamiento int(11)
1	1	PlayStation 5	Sony	2020
2	2	Xbox Series X	Microsoft	2020
3	3	Nintendo Switch	Nintendo	2017

3.7. Eliminar (terminar) las instancias y grupos de seguridad

Para eliminar la instancia y que no consuma recursos de AWS vamos a destruir (terminar) la instancia.

1. Elimina las instancias creadas.
3. Elimina los grupos de seguridad creados.