

---

# Práctica 1. Instalación y configuración de MySQL Server en Ubuntu Server

Apuntes de BD para DAW, DAM y ASIR

José Juan Sánchez Hernández

Curso 2024/2025

# Índice

<b>1</b>	<b>SGBDRs más usados actualmente</b>	<b>1</b>
<b>2</b>	<b>MySQL Server</b>	<b>2</b>
2.1	Instalación y configuración de MySQL Server en Ubuntu Server . . . . .	2
2.2	Opción 1. Instancia en OpenStack . . . . .	2
2.3	Opción 2. Máquina virtual en VirtualBox . . . . .	3
<b>3</b>	<b>MySQL</b>	<b>5</b>
3.1	Instalación de MySQL server . . . . .	5
3.2	Cómo acceder a MySQL Server desde consola con el usuario root . . . . .	5
3.3	Cómo cambiar la contraseña del usuario root (Método 1) . . . . .	5
3.4	Cómo cambiar la contraseña del usuario root (Método 2 --skip-grant-tables) . . . . .	7
3.5	Configuración de MySQL para permitir conexiones remotas . . . . .	8
3.6	Consultar los usuarios creados en MySQL Server . . . . .	9
3.7	Creamos un usuario para conectarnos desde una máquina remota . . . . .	10
3.8	Comprobamos que podemos conectarnos a MySQL . . . . .	11
3.9	Configuración de MySQL Workbench . . . . .	12
3.10	Cómo iniciar, parar y consultar el estado de MySQL . . . . .	16
3.10.1	Método 1. systemctl . . . . .	16
3.10.2	Método 2. /etc/init.d/mysql . . . . .	16
3.11	Archivos de configuración de MySQL . . . . .	16
3.12	Archivos de log de MySQL . . . . .	16
<b>4</b>	<b>Intérprete de comandos de MySQL</b>	<b>17</b>
4.1	Conexión con un MySQL local . . . . .	17
4.2	Conexión con un MySQL remoto . . . . .	17
4.3	Algunos comandos útiles para MySQL desde consola . . . . .	17
4.3.1	Listar todas las bases de datos disponibles . . . . .	17
4.3.2	Crear una nueva base de datos . . . . .	18
4.3.3	Borrar una base de datos . . . . .	18
4.3.4	Seleccionar una base de datos . . . . .	18
4.3.5	Listar las tablas que contiene una base de datos . . . . .	18
4.3.6	Mostrar la estructura de una tabla . . . . .	18
4.3.7	Cerrar la sesión y salir . . . . .	18
4.4	Ejecutar un script .sql desde la consola . . . . .	18
4.5	Usuarios y permisos en MySQL desde consola . . . . .	19
4.5.1	Tipos de permisos que podemos aplicar . . . . .	19

---

4.5.2	Crear un nuevo usuario . . . . .	19
4.5.3	Eliminar un usuario . . . . .	19
4.5.4	Asignar permisos a un usuario . . . . .	19
4.5.5	Eliminar permisos a un usuario . . . . .	20
4.5.6	Consultar los usuarios creados en MySQL . . . . .	20
<b>5</b>	<b>Licencia</b>	<b>21</b>

# Índice de figuras

# Índice de cuadros

# 1 SGBDRs más usados actualmente

Un **Sistema de Gestión de Bases de Datos Relacionales (SGBDRs)** es un programa que nos permite crear, actualizar y administrar una base de datos relacional. La mayoría de los sistemas de gestión de bases de datos relacionales utilizan [SQL](#) para acceder a la base de datos.

Hay que tener en cuenta que **pueden existir diferencias en la sintaxis SQL utilizada por los distintos SGBDRs.**

Los SGBDRs más usados actualmente son:

- [SQLite](#)
- [MySQL](#)
- [MariaDB](#)
- [PostgreSQL](#)
- [OracleDB](#)
- [SQL Server](#)

## 2 MySQL Server

### 2.1 Instalación y configuración de MySQL Server en Ubuntu Server

En esta práctica vamos a crear (1) una instancia en [OpenStack][17\_p1 o (2) una máquina virtual en [VirtualBox](#) con la última versión de [Ubuntu Server](#) donde vamos a instalar todo el software necesario para realizar nuestras primeras prácticas con el sistema gestor de bases de datos [MySQL](#).

### 2.2 Opción 1. Instancia en OpenStack

Para crear una instancia en la infraestructura de [OpenStack del centro](#), se recomienda seguir los pasos de la guía de uso:

- [Cómo crear instancias en OpenStack Celia](#).

Al crear la instancia podemos hacer uso de [cloud-init](#) para realizar la configuración inicial.

La configuración se define en unos archivos de texto plano con formato YAML que contienen las instrucciones de configuración inicial de la instancia. Por ejemplo, el siguiente archivo de configuración va a realizar las siguientes acciones:

- Configurar la contraseña del usuario del sistema que vamos a utilizar para conectarnos a la instancia por SSH.
- Instalar el servidor de base de datos MySQL.
- Configurar el parámetro bind-address para permitir conexiones remotas.
- Reiniciar el servicio de MySQL para que se apliquen los cambios de la configuración.
- Crear un usuario para conectarnos a MySQL desde cualquier host.

```
1 #cloud-config
2 password: nueva_password
3 chpasswd: { expire: False }
4 ssh_pwauth: True
5 package_update: true
6 package_upgrade: true
7 packages:
8   - mysql-server
9
10 runcmd:
11   # Configuramos el parámetro bind-address para permitir conexiones remotas
12   - sed -i 's/^bind-address.*/bind-address = 0.0.0.0/' /etc/mysql/mysql.conf.d/
    mysql.cnf
```

```
13
14 # Reiniciamos el servicio de MySQL
15 - systemctl restart mysql
16
17 # Creamos un usuario para conectarnos a MySQL desde cualquier host
18 - |
19 mysql -e "CREATE USER 'root'@'%' IDENTIFIED WITH 'mysql_native_password' BY
        'root';"
20 mysql -e "GRANT ALL PRIVILEGES ON *.* TO 'root'@'%';"
21 mysql -e "FLUSH PRIVILEGES;"
```

**Nota:** Recuerde que tendrá que sustituir los valores de las contraseñas del ejemplo por los valores que desee.

## 2.3 Opción 2. Máquina virtual en VirtualBox

Si decide crear una máquina virtual en [VirtualBox](#), tenga en cuenta que la tarjeta de red de la máquina virtual tiene que estar configurada en modo **adaptador puente**, y que debemos asignarle una **dirección IP estática manualmente**.

### Configuración de la interfaz de red en Ubuntu

La configuración de las interfaces de red se encuentra en el directorio `/etc/netplan`. En este directorio encontraremos un archivo de configuración con extensión `.yaml`.

Los archivos de configuración de las interfaces de red utilizados en las versiones de Ubuntu 18.04 y 20.04, tienen los siguientes nombres:

- Ubuntu 18.04: `50-cloud-init.yaml`
- Ubuntu 20.04: `00-installer-config.yaml`

Si nuestro archivo de configuración se llama `00-installer-config.yaml` deberíamos editarlo para añadir nuestra configuración de red.

```
1 sudo nano /etc/netplan/00-installer-config.yaml
```

El archivo deberá tener un contenido similar al este:

```
1 # This file is generated from information provided by
2 # the datasource. Changes to it will not persist across an instance.
3 # To disable cloud-init's network configuration capabilities, write a file
4 # /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
5 # network: {config: disabled}
6 network:
7   ethernets:
8     enp0s3:
9       addresses: []
10      dhcp4: true
11   version: 2
```

Ahora deberemos añadir nuestra configuración de red. Tenga en cuenta que en los archivos **YAML** no está permitido el uso de caracteres de tabulación para indentar, para indentar sólo podemos usar espacios en blanco.

A continuación se muestra un ejemplo de cómo sería la configuración de red para una máquina con la dirección IP **192.168.21.101**.

```
1 # This file is generated from information provided by
2 # the datasource. Changes to it will not persist across an instance.
3 # To disable cloud-init's network configuration capabilities, write a file
4 # /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
5 # network: {config: disabled}
6 network:
7     ethernets:
8         enp0s3:
9             addresses: [192.168.21.101/24]
10            gateway4: 192.168.21.99
11            nameservers:
12                addresses: [8.8.8.8,8.8.4.4]
13            dhcp4: no
14    version: 2
```

Tenga en cuenta que en su caso tendrá que sustituir la dirección IP **192.168.21.101** por la dirección IP que le haya asignado el profesor.

Una vez hecho esto aplicamos la nueva configuración de red.

```
1 sudo netplan apply
```

## 3 MySQL

### 3.1 Instalación de MySQL server

En primer lugar actualizamos la lista de paquetes del repositorio:

```
1 sudo apt update
```

Realizamos la instalación de `mysql-server`:

```
1 sudo apt install mysql-server -y
```

### 3.2 Cómo acceder a MySQL Server desde consola con el usuario root

Una vez que hemos instalado MySQL Server en nuestro sistema vamos a acceder a la consola de MySQL. Esto podemos hacerlo de dos formas.

#### Opción 1

Ejecutamos el siguiente comando:

```
1 sudo mysql
```

Una vez hecho esto ya tendríamos acceso a la consola de MySQL como `root`.

#### Opción 2

Iniciamos una sesión como `root`:

```
1 sudo su
```

Una vez que hemos iniciado una sesión como `root` vamos a iniciar la consola de MySQL también como `root` sin necesidad de indicarle ningún password (no es necesario usar el modificador `-p`).

```
1 mysql -u root
```

Una vez hecho esto ya tendríamos acceso a la consola de MySQL como `root`.

### 3.3 Cómo cambiar la contraseña del usuario root (Método 1)

En primer lugar accedemos a la consola de MySQL como `root` y seleccionamos la base de datos `mysql`.

```
1 USE mysql;
```

Vamos a revisar los usuarios que existen en MySQL y qué método tienen establecido para autenticar.

```
1 SELECT User, Host, plugin FROM user;
2
3 +-----+-----+-----+
4 | User          | Host       | plugin          |
5 +-----+-----+-----+
6 | root          | localhost  | auth_socket     |
7 | mysql.session | localhost  | caching_sha2_password |
8 | mysql.sys     | localhost  | caching_sha2_password |
9 | debian-sys-maint | localhost  | caching_sha2_password |
10 +-----+-----+-----+
11 4 rows in set (0.00 sec)
```

Podemos ver que para el usuario `root@localhost` el método de autenticación es `auth_socket`. Esto quiere decir que el usuario `root` usará las mismas credenciales que tiene en el sistema operativo.

En la columna `plugin` también podemos encontrar estas dos opciones:

- `mysql_native_password`: Este plugin cifra la contraseña utilizando una función hash basada en SHA-1. Actualmente no se considera segura pero se puede seguir utilizando para mantener la compatibilidad con clientes antiguos que siguen utilizando este método de cifrado.
- `caching_sha2_password`: Este plugin cifra la contraseña utilizando una función hash basada en SHA-256. Es más seguro que el anterior y es la opción que se utiliza por defecto a partir de MySQL 8.0

Por lo tanto, si queremos cambiar la contraseña de `root` tendremos que cambiar el método de autenticación a `mysql_native_password` o `caching_sha2_password`. Podemos hacerlo ejecutando alguna de las siguientes sentencias SQL, teniendo en cuenta que tendrá que sustituir `nueva_contraseña` por la contraseña que desee.

#### Opción `mysql_native_password` (Antigua)

```
1 ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY '
   nueva_contraseña';
```

Con esta sentencia SQL estamos indicando que la nueva contraseña del usuario `root` estará cifrada con el plugin `mysql_native_password`.

#### Opción `caching_sha2_password` (Nueva)

```
1 ALTER USER 'root'@'localhost' IDENTIFIED WITH caching_sha2_password BY '
   nueva_contraseña';
```

Con esta sentencia SQL estamos indicando que la nueva contraseña del usuario `root` estará cifrada con el plugin `caching_sha2_password`.

La [documentación oficial de MySQL](#) indica que la nueva opción `caching_sha2_password` es más segura que la antigua opción `mysql_native_password` y que por lo tanto es la opción recomendada.

Después habrá que ejecutar el siguiente comando para que las actualizaciones realizadas tengan efecto.

```
1 FLUSH PRIVILEGES;
```

### 3.4 Cómo cambiar la contraseña del usuario root (Método 2 --skip-grant-tables)

Otra opción para modificar la contraseña del usuario `root` en MySQL es iniciar el servicio con la opción `--skip-grant-tables`, que permite que cualquier usuario se pueda conectar sin necesidad de realizar el proceso de autenticación.

En primer lugar detenemos el servicio de MySQL.

```
1 sudo systemctl stop mysql
```

Como vamos a iniciar el proceso de MySQL de forma manual vamos a necesitar crear de forma manual el directorio `/var/run/mysql` y asignarle que el propietario de este directorio es `mysql` del grupo `mysql`.

```
1 sudo mkdir -p /var/run/mysql
2 sudo chown mysql:mysql /var/run/mysql
```

Una vez que hemos detenido el servicio de MySQL, lo volvemos a iniciar pero haciendo uso del comando que está en la ruta `/usr/sbin/mysqld` y pasándole como parámetro la opción `--skip-grant-tables` que permite conectarnos sin contraseña y con todos los privilegios. Esta opción deshabilita la gestión de usuarios y por lo tanto, no es posible utilizar sentencias como `ALTER USER` y `SET PASSWORD`. Al utilizar la opción `--skip-grant-tables` se habilita automáticamente la opción `--skip-networking` para desactivar las conexiones remotas como mecanismo de seguridad. Observe que hemos añadido `&` al final del comando para que el proceso se ejecute en segundo plano.

```
1 sudo /usr/sbin/mysqld --skip-grant-tables &
```

Podemos comprobar que el proceso se está ejecutando de forma correcta haciendo un listado de todos los procesos que están en ejecución con `ps aux` y buscando en la lista el nombre del proceso con `grep mysqld`.

```
1 ps aux | grep mysqld
```

Una vez que hemos comprobado que el proceso está en ejecución, podemos conectarnos al servidor de MySQL. No es necesario indicar ningún usuario ni contraseña, nos conectaremos automáticamente con todos los privilegios como `root`.

```
1 mysql
```

Ahora tendremos que indicarle al servidor que tiene que recargar las tablas encargadas de la autenticación de los usuarios para poder activar la gestión de usuarios. Recuerda que la opción `--skip-grant-tables` deshabilita esta funcionalidad.

```
1 FLUSH PRIVILEGES;
```

Modificamos la contraseña del usuario 'root'@'localhost', reemplazando `nueva_contraseña` por la contraseña que queramos asignar.

```
1 ALTER USER 'root'@'localhost' IDENTIFIED BY 'nueva_contraseña';
```

Salimos del cliente de MySQL.

```
1 exit;
```

Detenemos el proceso `mysqld`.

```
1 sudo pkill mysqld
```

Reiniciamos el servicio de MySQL.

```
1 sudo systemctl start mysql
```

### 3.5 Configuración de MySQL para permitir conexiones remotas

En la instalación por defecto, MySQL Server sólo acepta conexiones que vengan de `localhost`. Por lo tanto, vamos a ver cómo tenemos que configurar MySQL Server para que acepte conexiones remotas.

En primer lugar, edita el siguiente archivo de configuración:

```
1 sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf
```

Busca la directiva de configuración `bind-address` dentro del bloque de `[mysqld]`:

```
1 [mysqld]
2 bind-address = 127.0.0.1
```

En la configuración por defecto, MySQL sólo permite conexiones desde `localhost` (127.0.0.1). Habrá que modificar este valor por la dirección IP de la máquina donde se está ejecutando el servicio de MySQL.

```
1 [mysqld]
2 bind-address = IP_SERVIDOR_MYSQL
```

Si nuestra máquina dispone más de una interfaz de red podemos poner la dirección IP `0.0.0.0` para permitir que se puedan conectar a MySQL desde cualquiera de las interfaces de red disponibles de la máquina.

```
1 [mysqld]
2 bind-address = 0.0.0.0
```

Una vez hecho esto tenemos que reiniciar el servicio de MySQL:

```
1 sudo systemctl restart mysql
```

### 3.6 Consultar los usuarios creados en MySQL Server

Los usuarios de MySQL se almacenan en la tabla `mysql.user`. La clave primaria de esta tabla está formada por los valores `user` y `host`, de modo que cada fila vendrá identificada por un nombre de usuario y el host desde el que puede conectarse.

La siguiente consulta nos devuelve el listado de usuarios que tenemos en MySQL y desde qué interfaz de red del servidor MySQL pueden conectarse:

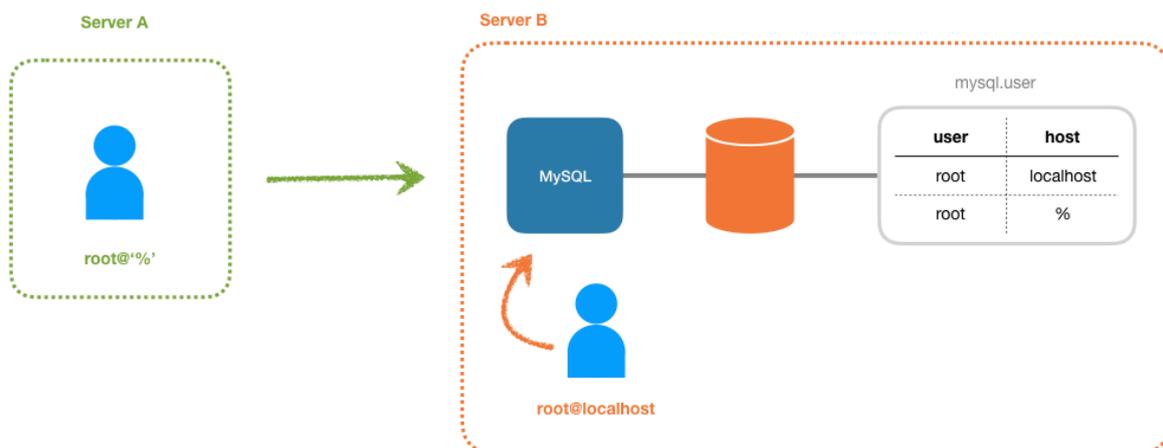
```
1 SELECT user,host FROM mysql.user;
```

En nuestra caso la consulta anterior devuelve el siguiente resultado:

```
1 +-----+-----+
2 | user          | host          |
3 +-----+-----+
4 | root          | localhost     |
5 | debian-sys-maint | localhost     |
6 | mysql.session | localhost     |
7 | mysql.sys     | localhost     |
8 +-----+-----+
```

Todos los usuarios que existen actualmente en la base de datos sólo pueden conectarse desde la propia máquina donde está instalado MySQL Server (`localhost`).

El siguiente diagrama muestra un ejemplo de dos usuarios que se están conectando a una máquina con MySQL Server. El usuario `root@localhost` es un usuario que sólo puede conectarse desde la máquina local y el usuario `root@'%'` es un usuario que se puede conectar desde una máquina remota.



<https://josejuansanchez.org>

Veamos cómo podemos crear un nuevo usuario que pueda conectarse desde una máquina remota.

### 3.7 Creamos un usuario para conectarnos desde una máquina remota

Ahora vamos a crear un usuario `root@%` para conectarnos a MySQL Server desde una máquina remota.

En primer lugar, nos conectamos a la consola de MySQL Server con el usuario `root@localhost` desde la misma máquina.

```
1 sudo mysql
```

A partir de la versión 8.0 de MySQL no es posible crear un usuario con la sentencia `GRANT` de SQL. Por lo tanto, sólo usaremos `GRANT` para aquellos usuarios que ya han sido creados previamente. Puede encontrar [más información en la documentación oficial](#).

En este caso tendremos que crear un usuario con la sentencia `CREATE USER` y luego asignarle los privilegios que queramos con la sentencia `GRANT`.

#### Paso 1. Creación de un usuario

```
1 CREATE USER 'nombre_usuario'@'IP_MÁQUINA_CLIENTE' IDENTIFIED BY 'contraseña';
```

Donde el valor de `IP_MÁQUINA_CLIENTE` indica que el usuario `nombre_usuario` sólo podrá conectarse al servidor de MySQL desde esa dirección IP. Los valores que podemos utilizar para `IP_MÁQUINA_CLIENTE` son los siguientes:

- `localhost`: No permitimos conexiones remotas, sólo se permiten conexiones dentro de la misma máquina.
- `%`: Este comodín indica que permitimos conexiones desde cualquier máquina.
- Una dirección IP desde la máquina que ese usuario puede conectarse al servidor de MySQL.

Tenga en cuenta que tendrá que reemplazar los valores `nombre_usuario`, `contraseña` y `IP_MÁQUINA_CLIENTE` por los valores que necesite.

Por lo tanto, para crear el usuario `root@%` con la contraseña `contraseña` ejecutaremos la sentencia SQL:

```
1 CREATE USER 'root'@'%' IDENTIFIED BY 'contraseña';
```

#### Paso 2. Asignación de privilegios

Una vez que hemos creado el usuario le asignamos los privilegios que sean necesarios sobre la base de datos de la aplicación. En este caso le vamos a asignar todos los privilegios con `ALL PRIVILEGES`.

```
1 GRANT ALL PRIVILEGES ON 'base_de_datos'.* TO 'nombre_usuario'@'IP_MÁQUINA_CLIENTE';
```

Tenga en cuenta que tendrá que reemplazar los valores `base_de_datos`, `nombre_usuario` y `IP_MÁQUINA_CLIENTE` por los valores que necesite.

Los diferentes tipos de permisos que podemos aplicar son los siguientes:

- `ALL PRIVILEGES`: permite a un usuario de MySQL acceder a todas las bases de datos asignadas en el sistema.
- `CREATE`: permite crear nuevas tablas o bases de datos.

- **DROP**: permite eliminar tablas o bases de datos
- **DELETE**: permite eliminar registros de tablas.
- **INSERT**: permite insertar registros en tablas.
- **SELECT**: permite leer registros en las tablas.
- **UPDATE**: permite actualizar registros seleccionados en tablas.
- **GRANT OPTION**: permite asignar privilegios a otros usuarios.

Por lo tanto, para asignarle todos los privilegios al usuario `root@%` sobre todas las bases de datos ejecutaremos la siguiente sentencia SQL:

```
1 GRANT ALL PRIVILEGES ON *.* TO 'root'@'%';
```

### Paso 3. Actualizamos las tablas de privilegios

Para que los cambios que hemos realizado sobre los privilegios de los usuarios se apliquen de forma inmediata, tendremos que ejecutar el siguiente comando.

```
1 FLUSH PRIVILEGES;
```

### Ejemplo 1

Por ejemplo, si queremos crear un usuario `root` con contraseña `root`, que se puede conectar desde cualquier máquina, que tiene privilegios sobre todas las bases de datos que existen y cuya contraseña es `root` tendríamos que ejecutar la siguiente sentencia SQL:

```
1 CREATE USER 'root'@ '%' IDENTIFIED BY 'root';
2 GRANT ALL PRIVILEGES ON *.* TO 'root'@ '%';
3 FLUSH PRIVILEGES;
```

### Ejemplo 2

En este ejemplo vamos a indicar el plugin que se utilizará para cifrar la contraseña del usuario. En este caso se ha seleccionado el plugin `mysql_native_password`.

```
1 CREATE USER 'root'@ '%' IDENTIFIED WITH 'mysql_native_password' BY 'root';
2 GRANT ALL PRIVILEGES ON *.* TO 'root'@ '%';
3 FLUSH PRIVILEGES;
```

## 3.8 Comprobamos que podemos conectarnos a MySQL

Ahora vamos a comprobar que podemos conectarnos con MySQL desde la máquina donde está corriendo el servicio de Apache HTTP. Podemos comprobarlo conectando con el shell de `mysql`:

```
1 mysql -u USERNAME -p -h IP-SERVIDOR-MYSQL
```

O haciendo un telnet al puerto donde está corriendo el servicio de MySQL:

```
1 telnet IP-SERVIDOR-MYSQL 3306
```

Si no podemos conectarnos a MySQL revisaremos que el servicio está activo y que no tenemos ningún firewall que nos esté filtrando el puerto del servicio donde se ejecuta MySQL.

### 3.9 Configuración de MySQL Workbench

En nuestro caso, el **cliente** que vamos a utilizar para conectarnos a MySQL Server será [MySQL Workbench](#).

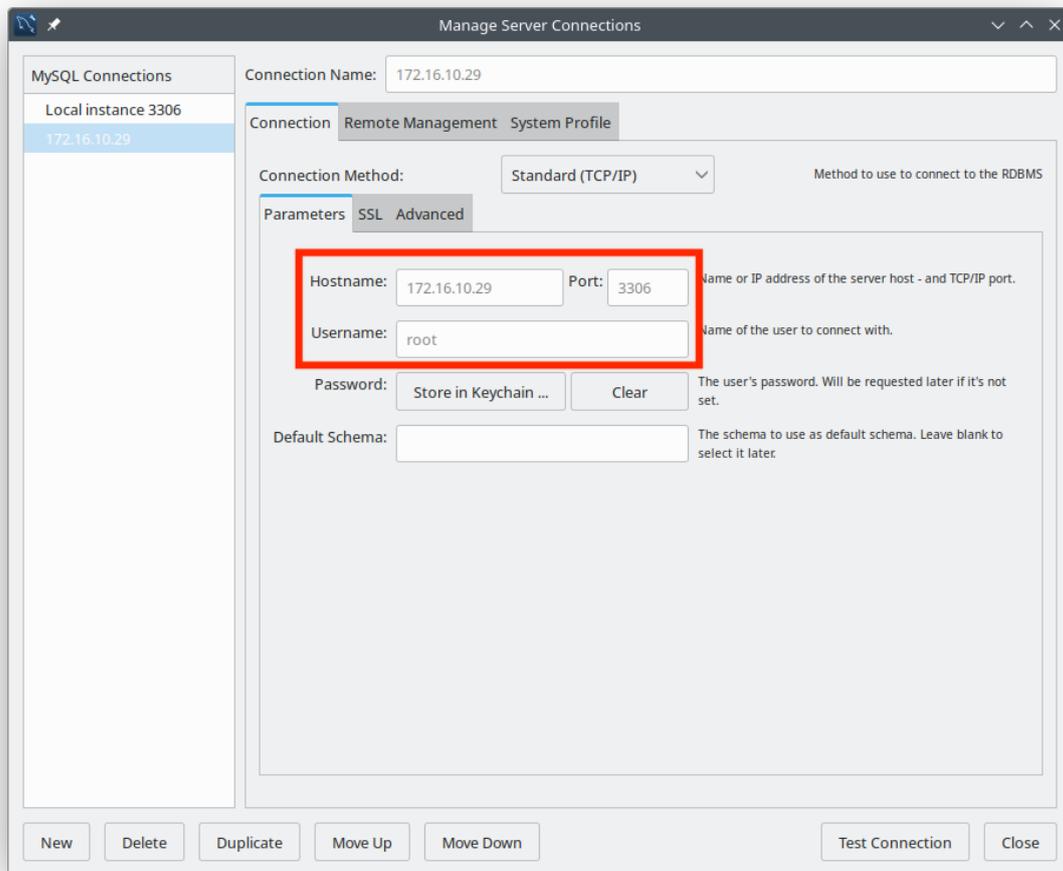
[MySQL Workbench](#) es una aplicación con interfaz gráfica que nos permite administrar y gestionar bases de datos MySQL de una forma sencilla.

Una vez que tengamos instalado [MySQL Workbench](#) en nuestra máquina tendremos que configurar una nueva conexión.

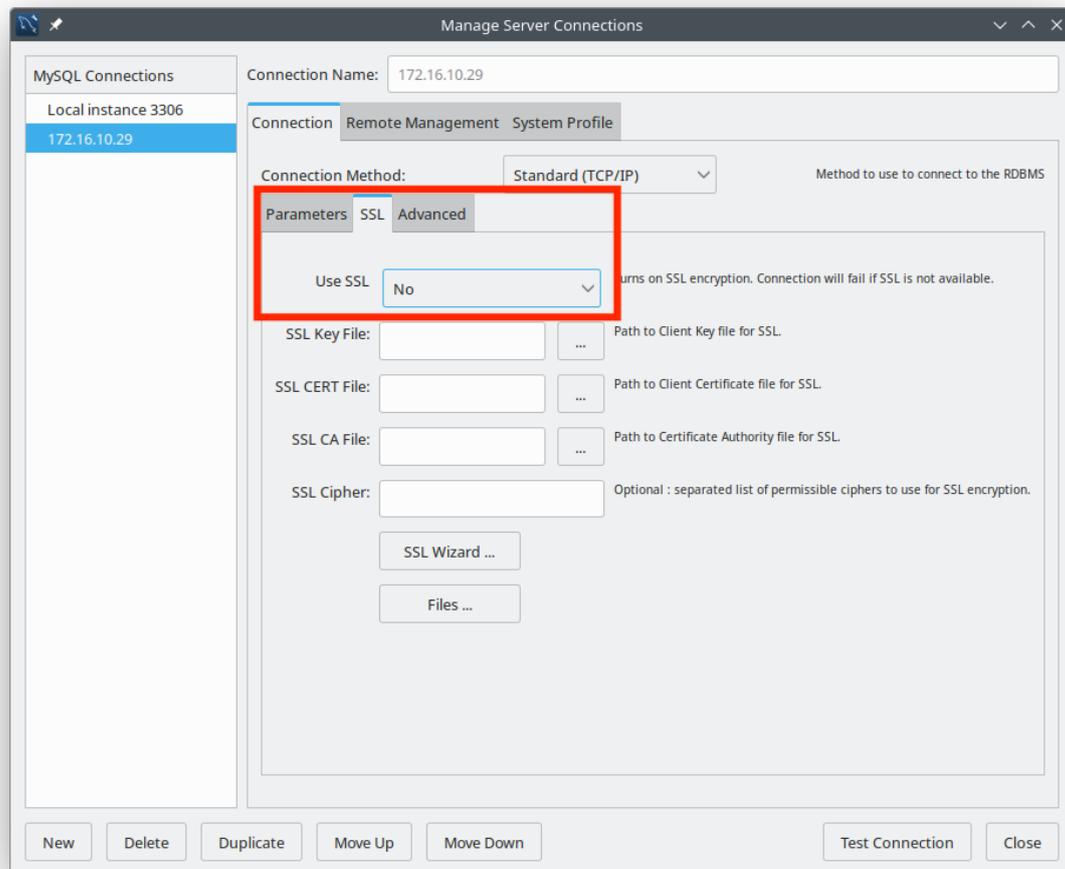
Los parámetros de conexión que tenemos que configurar son los siguientes:

- **Connection name:** Nombre que queramos asignarle a la conexión.
- **Connection method:** Usaremos [Standard \(TCP/IP\)](#).
- **Hostname:** Será la dirección IP de la máquina remota.
- **Port:** El puerto de MySQL Server por defecto es el 3306.
- **Username:** Nombre de usuario.
- **Password:** Contraseña del usuario.

La siguiente imagen muestra los valores de la conexión que tenemos que configurar en la pestaña [Parameters](#).



En la pestaña [SSL](#) tendremos que indicar que no queremos utilizar SSL.

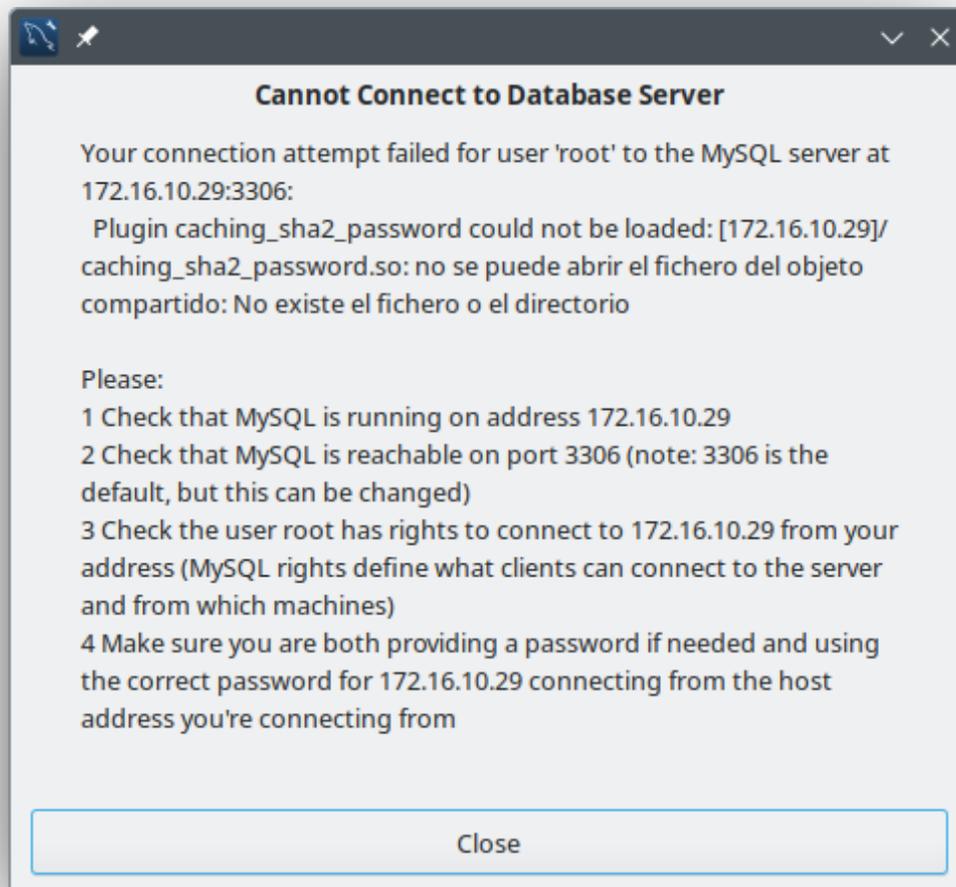


Después de configurar estos parámetros podemos pulsar sobre el botón **Test Connection** para comprobar que la conexión se realiza de forma correcta.

En las instalaciones de **MySQL Workbench** que tenemos en el aula, podemos encontrarnos con el siguiente error cuando intentamos conectarnos a MySQL Server.

```
1 Plugin caching_sha2_password cannot be loaded
```

Este error ocurre porque la versión de **MySQL Workbench** que tenemos instalada no tiene soporte para la autenticación `caching_sha2_password`.



Para solucionarlo, podemos actualizar la contraseña de nuestro usuario y modificar el plugin que se utilizará para cifrarla. En este caso utilizaremos el plugin `mysql_native_password` que sí es compatible con nuestra versión, aunque debe tener en cuenta que es menos seguro que `caching_sha2_password`.

Desde el terminal de la máquina remota donde se está ejecutando **MySQL Server** ejecutamos el siguiente comando:

```
1 sudo mysql
```

Una vez que hemos accedido a la consola de MySQL, modificamos la contraseña utilizando el plugin `mysql_native_password`.

```
1 ALTER USER 'root'@ '%' IDENTIFIED WITH mysql_native_password BY 'root';
```

Observe que estamos actualizando la contraseña del usuario `root@ '%'` y que como contraseña le hemos asig-

nado el valor `root`.

## 3.10 Cómo iniciar, parar y consultar el estado de MySQL

### 3.10.1 Método 1. `systemctl`

```
1 sudo systemctl start mysql
2 sudo systemctl stop mysql
3 sudo systemctl restart mysql
4 sudo systemctl status mysql
```

### 3.10.2 Método 2. `/etc/init.d/mysql`

```
1 sudo /etc/init.d/mysql start
2 sudo /etc/init.d/mysql stop
3 sudo /etc/init.d/mysql restart
4 sudo /etc/init.d/mysql reload
5 sudo /etc/init.d/mysql force-reload
6 sudo /etc/init.d/mysql status
```

## 3.11 Archivos de configuración de MySQL

```
1 /etc/mysql/mysql.cnf
```

También podemos encontrar archivos de configuración en los directorios:

```
1 /etc/mysql/conf.d/
2 /etc/mysql/mysql.conf.d/
```

## 3.12 Archivos de log de MySQL

```
1 /var/log/mysql/error.log
```

## 4 Intérprete de comandos de MySQL

### 4.1 Conexión con un MySQL local

Para conectarnos a un MySQL que se está ejecutando en nuestra máquina local (`localhost`). Con el parámetro `-u` indicamos el nombre del usuario con el que queremos conectar y con el parámetro `-p` nos pedirá el `password` del usuario.

```
1 mysql -u root -p
```

### 4.2 Conexión con un MySQL remoto

Para conectarnos a un MySQL remoto que se está ejecutando en otra máquina, utilizamos el parámetro `-h` con la IP de la máquina donde nos queremos conectar.

```
1 mysql -u root -p -h 192.168.20.102
```

Cuando hayamos conectado nos aparecerá la siguiente consola de comandos SQL.

```
1 Welcome to the MySQL monitor.  Commands end with ; or \g.
2 Your MySQL connection id is 5
3 Server version: 5.7.19-0ubuntu0.16.04.1 (Ubuntu)
4
5 Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights reserved.
6
7 Oracle is a registered trademark of Oracle Corporation and/or its
8 affiliates. Other names may be trademarks of their respective
9 owners.
10
11 Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
12
13 mysql>
```

### 4.3 Algunos comandos útiles para MySQL desde consola

#### 4.3.1 Listar todas las bases de datos disponibles

```
1 SHOW DATABASES;
```

### 4.3.2 Crear una nueva base de datos

```
1 CREATE DATABASE <database>;
```

### 4.3.3 Borrar una base de datos

```
1 DROP DATABASE <database>;
```

### 4.3.4 Seleccionar una base de datos

```
1 USE <database>;
```

### 4.3.5 Listar las tablas que contiene una base de datos

```
1 SHOW TABLES;
```

### 4.3.6 Mostrar la estructura de una tabla

```
1 DESCRIBE <table>;
```

### 4.3.7 Cerrar la sesión y salir

```
1 exit
```

```
1 quit
```

## 4.4 Ejecutar un script .sql desde la consola

Desde la consola de Linux:

```
1 mysql -u root -p < script.sql
```

Desde la consola de MySQL:

```
1 mysql> source script.sql
```

## 4.5 Usuarios y permisos en MySQL desde consola

### 4.5.1 Tipos de permisos que podemos aplicar

- **ALL PRIVILEGES**: permite a un usuario de MySQL acceder a todas las bases de datos asignadas en el sistema.
- **CREATE**: permite crear nuevas tablas o bases de datos.
- **DROP**: permite eliminar tablas o bases de datos
- **DELETE**: permite eliminar registros de tablas.
- **INSERT**: permite insertar registros en tablas.
- **SELECT**: permite leer registros en las tablas.
- **UPDATE**: permite actualizar registros seleccionados en tablas.
- **GRANT OPTION**: permite remover privilegios de usuarios

### 4.5.2 Crear un nuevo usuario

Habrás que reemplazar *nombre\_usuario* y *contraseña* por los datos del nuevo usuario que desea crear:

```
1 CREATE USER 'nombre_usuario'@'localhost' IDENTIFIED BY 'contraseña';
```

Una vez que hemos creado el usuario hay que asignarle permisos para que pueda acceder a la/s base/s de datos que queramos.

### 4.5.3 Eliminar un usuario

```
1 DROP USER 'nombre_usuario'@'localhost';
```

### 4.5.4 Asignar permisos a un usuario

```
1 GRANT [permiso] ON [nombre_base_de_datos].[nombre_tabla] TO 'nombre_usuario'@'localhost';
```

Por ejemplo:

```
1 GRANT ALL PRIVILEGES ON *.* TO 'nombre_usuario'@'localhost';
```

En este comando, los asteriscos indican que estamos aplicando el permiso **ALL PRIVILEGES** al usuario *nombre\_usuario* para todas las tablas de cada una de las bases de datos. bases de datos y tablas.

Después de este comando habrá que ejecutar el siguiente comando para refrescar todos los privilegios a los usuarios.

```
1 FLUSH PRIVILEGES;
```

### 4.5.5 Eliminar permisos a un usuario

```
1 REVOKE [permiso] ON [nombre_base_de_datos].[nombre_tabla] FROM 'nombre_usuario'
   @'localhost';
```

### 4.5.6 Consultar los usuarios creados en MySQL

Los usuarios de MySQL se almacenan en la tabla `mysql.user`. La clave primaria de esta tabla está formada por los valores `user` y `host`, de modo que cada fila vendrá identificada por un nombre de usuario y el host desde el que puede conectarse.

La siguiente consulta nos devuelve el listado de usuarios que tenemos en MySQL y desde qué host pueden conectarse:

```
1 SELECT user,host FROM mysql.user;
```

En nuestra caso la consulta anterior devuelve el siguiente resultado:

```
1 +-----+-----+
2 | user          | host          |
3 +-----+-----+
4 | root          | %             |
5 | root          | localhost     |
6 | debian-sys-maint | localhost     |
7 | phpmyadmin    | localhost     |
8 | mysql.session | localhost     |
9 | mysql.sys     | localhost     |
10 +-----+-----+
```

También podemos consultar qué permisos específicos tiene un determinado usuario. La siguiente consulta nos devuelve los permisos que tiene el usuario `root`:

```
1 SHOW GRANTS FOR root;
```

```
1 +-----+
2 | Grants for root@% |
3 +-----+
4 | GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' |
5 +-----+
```

## **5 Licencia**

Esta página forma parte del curso Bases de Datos de José Juan Sánchez Hernández y su contenido se distribuye bajo una licencia Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional.