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

Apuntes de BD para DAW, DAM y ASIR

José Juan Sánchez Hernández

Curso 2023/2024

Índice

1	SGBDRs más usados actualmente	1
2	MySQL Server	2
2.1	Instalación y configuración de MySQL Server en Ubuntu Server	2
2.2	Configuración de la interfaz de red	2
2.3	Configuración de la interfaz de red en Ubuntu 20.04 LTS	2
2.4	Configuración en versiones previas a Ubuntu 18.04 LTS Server	3
2.5	Configuración del proxy	4
3	MySQL	5
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)	6
3.5	Configuración de MySQL	7
3.5.1	Consultar los usuarios creados en MySQL Server	8
3.6	Asignando privilegios a un usuario para conectarnos desde una máquina remota	9
3.6.1	Para versiones de MySQL <= 5.7	9
3.6.2	Para todas las versiones de MySQL	10
3.7	Comprobamos que podemos conectarnos a MySQL	12
3.8	Cómo iniciar, parar y consultar el estado de MySQL	12
3.8.1	Método 1. systemctl	12
3.8.2	Método 2. /etc/init.d/mysql	12
3.9	Archivos de configuración de MySQL	12
3.10	Archivos de log de MySQL	12
4	Intérprete de comandos de MySQL	14
4.1	Conexión con un MySQL local	14
4.2	Conexión con un MySQL remoto	14
4.3	Algunos comandos útiles para MySQL desde consola	14
4.3.1	Listar todas las bases de datos disponibles	14
4.3.2	Crear una nueva base de datos	15
4.3.3	Borrar una base de datos	15
4.3.4	Seleccionar una base de datos	15
4.3.5	Listar las tablas que contiene una base de datos	15
4.3.6	Mostrar la estructura de una tabla	15
4.3.7	Cerrar la sesión y salir	15

4.4	Ejecutar un script .sql desde la consola	15
4.5	Usuarios y permisos en MySQL desde consola	16
4.5.1	Tipos de permisos que podemos aplicar	16
4.5.2	Crear un nuevo usuario	16
4.5.3	Eliminar un usuario	16
4.5.4	Asignar permisos a un usuario	16
4.5.5	Eliminar permisos a un usuario	17
4.5.6	Consultar los usuarios creados en MySQL	17
5	Licencia	18

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

Vamos a crear 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 Configuración de la interfaz de red

Tenga en cuenta que la tarjeta de red de la máquina virtual tiene que estar configurada en modo **adaptador puente**, que debemos asignarle una **dirección IP estática manualmente** y que tenemos que configurar la máquina para que use el **proxy** que tenemos en clase.

2.3 Configuración de la interfaz de red en Ubuntu 20.04 LTS

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
```

2.4 Configuración en versiones previas a Ubuntu 18.04 LTS Server

Este paso sólo será necesario realizarlo si hemos instalado una versión anterior a la 18.04.

Para configurar la interfaz de red manualmente lo haremos editando el archivo `/etc/network/interfaces`.

```
1 sudo nano /etc/network/interfaces
```

Configuramos la interfaz con los siguientes parámetros, teniendo 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.

```
1 auto enp0s3
2 iface enp0s3 inet static
3     address 192.168.21.101
4     netmask 255.255.255.0
5     gateway 192.168.21.99
6     dns-nameservers 8.8.8.8 8.8.4.4
```


Reiniciamos el servicio de red:

```
1 sudo /etc/init.d/networking restart
```

También podemos desactivar la interfaz **enp0s3** y volver a activarla para que se apliquen los cambios de la nueva configuración a esa interfaz de red.

```
1 sudo ifdown enp0s3
2 sudo ifup enp0s3
```

2.5 Configuración del proxy

Este paso sólo será necesario realizarlo si estamos haciendo uso del proxy del aula.

Para poder tener conexión a Internet en nuestra máquina virtual tenemos que configurarla para que use el proxy que tenemos en el aula. Para esto editamos el archivo `/etc/environment`:

```
1 sudo nano /etc/environment
```

Y añadimos las siguientes variables de entorno al final del archivo:

```
1 http_proxy=http://192.168.21.99:3128
2 https_proxy=https://192.168.21.99:3128
```

3 MySQL

3.1 Instalación de MySQL server

En primer lugar actualizamos la lista de paquetes del repositorio:

```
1 sudo apt-get update
```

Realizamos la instalación de `mysql-server`:

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

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. En primer lugar vamos a iniciar 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.

Si queremos cambiar la contraseña de `root` tendremos que cambiar el método de autenticación a `mysql_native_password` o `caching_sha2_password`. 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';
```

Opción `caching_sha2_password` (Nueva)

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

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/mysql` 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

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.

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

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

```
1 sudo /etc/init.d/mysql restart
```

3.5.1 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:

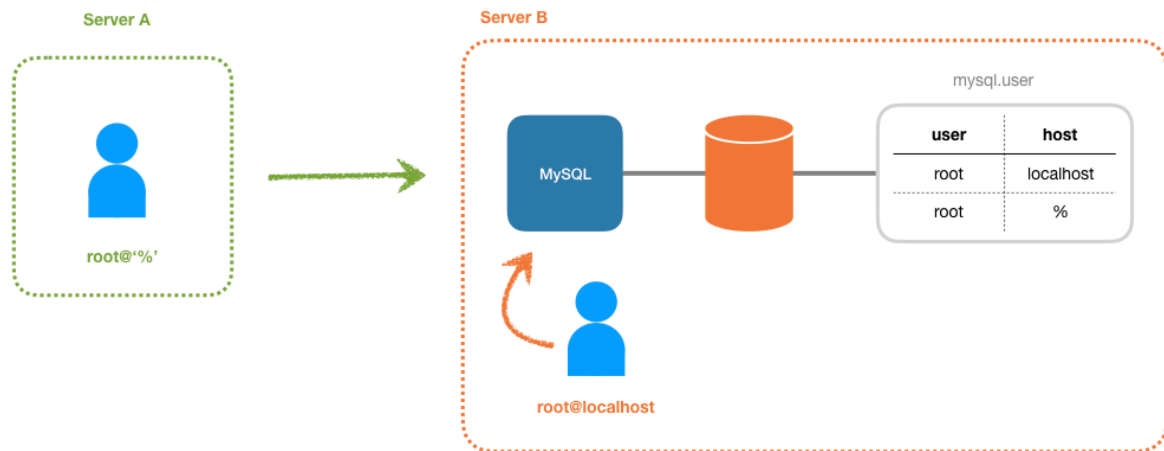
```
1 SELECT user,host FROM mysql.user;
```

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.

3.6 Asignando privilegios a un usuario para conectarnos desde una máquina remota

Ahora tenemos que asignar privilegios al usuario de MySQL que vamos a utilizar para conectarnos desde una máquina remota.

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

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

3.6.1 Para versiones de MySQL <= 5.7

Una vez que hemos conectado con MySQL Server vamos a crear un usuario que pueda conectarse desde cualquier máquina. Recuerde que los usuarios en MySQL están identificados por un nombre de usuario y un host, en este caso vamos a utilizar el comodín `%` como el host, para indicar que este usuario se puede conectar desde cualquier dirección IP.

La sentencia SQL para realizar esta acción sería la siguiente:

```
1 mysql> GRANT ALL PRIVILEGES ON DATABASE.* to USERNAME@'%' IDENTIFIED BY '
  PASSWORD';
2 mysql> FLUSH PRIVILEGES;
```

Tenga en cuenta que tendrá que reemplazar los valores `DATABASE`, `USERNAME` y `IP-SERVIDOR-HTTP` por los valores que necesite.

Por ejemplo, si queremos crear un usuario `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 mysql> GRANT ALL PRIVILEGES ON *.* to root@'%' IDENTIFIED BY 'root';
2 mysql> FLUSH PRIVILEGES;
```

Una vez hecho esto comprobamos cuál es la lista de usuarios que existen actualmente 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 | mysql.session | localhost     |
8 | mysql.sys     | localhost     |
9 +-----+-----+
```

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 +-----+-----+
```

3.6.2 Para todas las versiones de 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.

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.

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

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;
```


3.7 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.

3.8 Cómo iniciar, parar y consultar el estado de MySQL

3.8.1 Método 1. `systemctl`

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

3.8.2 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
```

3.9 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/
```

3.10 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

Habrá 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.