

TEMA 3: MySQL Server

ADMINISTRACIÓN DE SERVICIOS

ADMINISTRACIÓN DE SISTEMAS OPERATIVOS

Curso 2020/2021

Cristian Martín Fernández

Contenido

- Servidor MySQL
- Instalación
- Configuraciones Ubuntu
- mysqladmin, mysql, mysqldump
- Administración web de MySQL: phpmyadmin
- MySQL Tuner

SERVIDOR MySQL

- Uno de los servidores de base de datos más populares y que forma parte de distribución LAMP.
- *“MySQL is a fast, multi-threaded, multi-user, and robust SQL database server. It is intended for mission-critical, heavy-load production systems and mass-deployed software.”*

Instalación MySQL Server

- Para la instalación (requiere de al menos 247MB):
 - `$ sudo apt install mysql-server`
- Una vez completada la instalación, el servidor MySQL debe iniciarse automáticamente. Para verificarlo:
 - `$ sudo service mysql status`

Convecciones Ubuntu

- `/etc/mysql`
 - Este directorio contiene los archivos principales de configuración para MySQL.
- `/etc/mysql/my.cnf`
 - Este es el núcleo del archivo de configuración de MySQL, incluye directorios. Los usuarios pueden crear de forma individual sus propias configuraciones y guardarlas en `~/.my.cnf`
- `/var/lib/mysql/`
 - En este directorio encontraremos todos los archivos de la base de datos utilizados en la base de datos activa.

Convecciones Ubuntu

- `/etc/mysql/mysql.conf.d/mysqld.cnf`
- Fichero principal de configuración para MySQL. Se puede cambiar el puerto de escucha, el fichero de log, etc.
- Pero lo más importante es configurar la IP del servidor para que pueda atender a peticiones de otros equipos en la red. Por defecto, sólo escucha en localhost:
 - `bind-address = 127.0.0.1`
- Hay que reiniciar el servicio MySQL después de cambiar esta configuración:
 - `$ sudo systemctl restart mysql.service`

Convecciones Ubuntu

- `/var/log/syslog`
- Ubuntu ajusta MySQL para que escriba en el fichero estándar del registro del sistema en vez de hacerlo en `/var/log/mysql.log` o en el directorio `/var/log/mysql`.
- De esta forma para visualizar solo los registros relacionados con mysql podemos usar el comando `grep`:
 - ```
$ sudo grep mysql /var/log/syslog
```
- Para consultar los logs de este servicio `<mysql>` o los demás disponibles (`apache2`, `ssh`, `bind9`) se puede utilizar también la herramienta centralizada `journalctl`:
  - ```
$ journalctl -u mysql
```
- También se puede filtrar por fecha:
 - ```
$ journalctl -u mysql --since "2020-10-04 17:15:00"
```

# Procesos y servicios de MySQL

## ■ /etc/init.d/mysql

- Es el script de inicio de MySQL. Está configurado para ejecutarse automáticamente en el inicio del sistema.

## ■ mysqld

- Es el proceso servidor de MySQL

## ■ mysqld\_safe

- Es la forma recomendada de iniciar mysqld. Agrega algunas características de seguridad, como reiniciar el servidor cuando se produce un error y registrar la información de tiempo de ejecución en un registro de errores.

## ■ mysqladmin

- Herramienta de administración

## ■ mysql

- Cliente en línea de comandos. Se pueden ejecutar transacciones SQL.

## ■ mysqldump

- Herramienta para crear copias de seguridad



# mysqladmin

- Ubuntu incluye la herramienta mysqladmin para la administración de MySQL.
- Por ejemplo: Para ver el estado actual del proceso MySQL:
  - `$ sudo mysqladmin status`
- El programa devolverá información como el tiempo de actividad, cuantos hilos están activos, cuantas consultas lentas están ejecutándose y el porcentaje de consultas que se procesas por segundo.
- Cuando entramos con sudo no es necesario la contraseña por defecto.
- La opción “-p” le indica a mysqladmin que nos pregunte la contraseña en la línea de comandos
  - `$ sudo mysqladmin -u root -p status`
- Comprobar que mysql está corriendo:
  - `$ sudo mysqladmin -u root -p ping`

# mysqladmin

- Algunos de los comandos más usados son:

- create y drop databases:

- `$ mysqladmin -p create <databasename>`
- `$ mysqladmin -p drop <databasename>`

- Estos comandos reciben un nombre de base de datos como argumento y crean o borran respectivamente una base de datos de nuestra instancia de MySQL.

- flush-\*:

- Existe un cierto número de comandos que liberan cachés u otros parámetros en MySQL.

- flush-hosts: Limpia la información sobre hosts en la caché de hosts
- flush-tables: Limpia todas las tablas.
- flush-threads: Limpia todos los hilos caché.
- flush-logs: Limpia la información de logs.

# mysqladmin

- El comando `extended-status` proporciona un conjunto más complejo de información de estado que `status`
- El comando `processlist` lista todas las tareas del servidor activas junto con su ID. Este ID nos puede servir como referencia si necesitamos cerrar un proceso en particular.
- `kill` Termina todos los id de procesos que le pasemos como argumento. Si se especifican múltiples procesos, los separamos por coma.

# mysql

- mysql es un cliente que permite interactuar con el servidor MySQL y generar consultas
- Por defecto, la autenticación se realiza utilizando el plugin `auth_socket`.
- Si se accede como usuario `root`, no es necesario introducir contraseña (se comprueba el ID del usuario)
- Para volver al método de autenticación tradicional hay que realizar los siguientes pasos:

```
■$ sudo mysql
■$ select user, host, plugin from mysql.user;
■$ alter user 'root'@'localhost' identified with mysql_native_password by
'newPass';
```

`$ mysql -u root -p → OK`

- También se puede cambiar con `mysqladmin`:
  - `$ sudo mysqladmin -p password <newpassword>`

# Comando útiles mysql

- Una vez conectados a mysql podemos consultar las bases de datos activas:
  - `$ mysql> show databases;`
- Crear una base de datos:
  - `$ mysql> CREATE DATABASE database1;`
- Crear usuario en MySQL que se pueda conectar desde localhost:
  - `$ mysql> CREATE USER 'newuser'@'localhost' IDENTIFIED BY 'user_password';`
- Crear usuario en MySQL que se pueda conectar desde cualquier host:
  - `$ mysql> CREATE USER 'newuser'@'%' IDENTIFIED BY 'user_password';`
- Una vez creado el usuario hay que darle permisos sobre la BBDD creada (database1):
  - `$ mysql> GRANT ALL PRIVILEGES ON database1.* TO 'newuser'@'localhost';`
- Mostrar permisos para un usuario (newuser):
  - `SHOW GRANTS FOR 'newuser'@'localhost';`
- Eliminar usuario:
  - `$ mysql> DROP USER 'user'@'localhost'`

# mysqldump

- MySQL incluye la herramienta `mysqldump` para realizar **copias de seguridad lógicas**, produciendo un conjunto de instrucciones SQL que se pueden ejecutar para reproducir la base de datos original.
- Para backups de grandes bases de datos se recomienda realizar backup físicos.
- Se puede utilizar de tres formas:
  - `mysqldump [opciones] nombre_bd [nombre_tabla1 nombre_tabla2...]`
  - `mysqldump [opciones] --databases nombre_bd1 nombre_bd2 ...`
  - `mysqldump [opciones] -all-databases > seguridad.sql`
- Con la opción `--no-data` se evitar copiar los datos de las tablas.
- Más información de las opciones en: <https://dev.mysql.com/doc/refman/8.0/en/mysqldump.html>

# mysqldump

- Vamos a seleccionar la base de datos database1 creada anteriormente con mysql:
  - `$ mysql> use database1;`
- Dentro de la base de datos podemos crear una tabla de ejemplo:
  - `$ mysql> CREATE TABLE pet (name VARCHAR(20), owner VARCHAR(20), species VARCHAR(20), sex CHAR(1), birth DATE, death DATE);`
- Comprobamos que la tabla se ha creado:
  - `$ mysql> show tables;`
- Vamos a crear un backup lógico de la base de datos “database1”
  - `$ mysqldump -u root -p --databases database1 > database1.sql`
- Ahora eliminamos la base de datos, la volvemos a crear y la recuperamos con:
  - `$ mysql -u root -p database1 < database1.sql`

## ■ Administración web de MySQL: phpmyadmin

- Para instalar phpmyadmin y seleccionar la opción de Apache:

- `$ sudo apt install phpmyadmin`

- phpmyadmin se puede servir en Apache. Una vez completada la instalación del paquete, añadir la siguiente línea al final del fichero `/etc/apache2/apache2.conf` y reiniciar apache para que lo pueda servir:

```
Include /etc/phpmyadmin/apache.conf
```



# Administración web de MySQL: phpmyadmin

- Para acceder debemos usar la ruta /phpmyadmin.
- Supongamos que tenemos nuestro servidor Apache con una redirección de puertos en el puerto 8888, entonces la URL sería:
  - <http://localhost:8888/phpmyadmin>
- Es importante tener en cuenta que phpmyadmin no mantiene ninguna cuenta de usuarios por si mismo, simplemente pasa el usuario y la contraseña directamente a MySQL.
- Una vez que nos hayamos autenticado, veremos la pantalla principal de configuración y desde aquí podemos afinar los parámetros de nuestro servidor MySQL, crear y modificar bases de datos, sus tablas y acceder a la propia documentación de phpmyadmin

# MySQL Tuner

- MySQL Tuner se conecta a una instancia de MySQL en ejecución y ofrece sugerencias de configuración para optimizar la base de datos.
- Para instalación:
  - `$ sudo apt install mysqltuner`
- Funcionará mejor cuando MySQL lleve tiempo ejecutándose. Como resultado proporcionará un informe con mejoras para el fichero principal de configuración de MySQL.

# Bibliografía

- MySQL Doc: <https://dev.mysql.com/doc/>
- MySQL Ubuntu Server: <https://ubuntu.com/server/docs/databases-mysql>
- ApacheMySQLPHP: <https://help.ubuntu.com/community/ApacheMySQLPHP>