

# CONFIGURACIÓN DEL SERVIDOR

---

## 1. Estructura del archivo de configuración

El archivo de configuración está organizado en secciones (o grupos). Una sección está compuesta por un nombre (en general, el nombre del programa correspondiente) colocado entre dos corchetes, por opciones y termina por el comienzo de otra sección o el final del archivo. De esta forma, el nombre de la sección del servidor MySQL es [mysqld]. Sin embargo, es posible utilizar [server]. El nombre de la sección del cliente en texto por defecto es [mysql] (no confundir con [mysqld]). Siguiendo el mismo principio, podemos usar la sección [mysqldump], que permite configurar el cliente texto para hacer respaldos mysqldump. También existe la sección [client], que permite agrupar la información de configuraciones de todos los clientes (mysql, mysqldump, mysqladmin...).

La sintaxis de las opciones es la siguiente: nombre\_opción=valor. Las opciones compuestas por varias palabras pueden separarse indistintamente por un guión «-» o un guión bajo «\_». Algunas opciones son ejecutables: la funcionalidad se activa con solo poner el nombre de la opción. Por ejemplo, la opción enable-federated permite activar el motor FEDERATED. Además, podemos poner comentarios en el archivo de configuración: basta con añadir el prefijo almohadilla «#» o un punto y coma «;».

Ejemplo de archivo de configuración

```
$ cat /etc/mysql/my.cnf
# Sección global a todos los clientes MySQL
[cliente]
socket = /tmp/mysql.sock # los clientes se conectan con el socket
# Sección del cliente texto mysql
[mysql]
prompt = \r:\m \U$d> # la línea de comandos del cliente texto mysql
tiene el formato HH:MM user$schema>
# Sección del servidor MySQL
[mysqld]
slow_query_log # activación del registro de peticiones lentas
slow_query_log_file = mysql-slow.log # Nombre del registro de
peticiones lentas
largo-query-time = 2 # Las demandas cuya duración sea superior a 2 segundos
son registradas
enable-federated # Activa el motor FEDERATED
```

## 2. Configuración al arrancar mysqld

Podemos configurar el servidor pasando las opciones de configuración al ejecutar el programa mysqld. Estas últimas son las mismas utilizadas en el archivo de configuración pero, precedidas por dos guiones «--».

```
$ ./mysqld --defaults-file=/home/daz/mysql/my.cnf
--basedir=/usr/local/mysql/ --datadir=/home/daz/mysql/data
--log-error=/home/daz/mysql/data/mysql.err
--pid-file=/home/daz/mysql/data/mysql.pid
--socket=/tmp/mysql.sock --port=3306 --console &
```

### 3. Configuración dinámica del servidor

Algunas opciones se pueden cambiar de forma dinámica (en caliente) mediante el comando SET siempre que el usuario tenga el derecho SUPER (ver el capítulo Seguridad y gestión de los usuarios). Esto es especialmente práctico para perfeccionar la configuración del servidor MySQL en caso de emergencia. El cambio puede realizarse de dos maneras, bien a nivel local en la conexión, es decir, solo para la sesión de un cliente en curso mediante la cláusula SESSION, o bien de forma global en el servidor, empleando la cláusula GLOBAL. La mayoría de las opciones tienen un alcance de sesión y global (como la opción `sql_mode` que veremos más adelante en este capítulo), pero este no es el caso de todas. Por ejemplo, la opción `sql_log_bin`, que permite desactivar el registro binario, solo tiene un alcance de sesión, mientras que la opción `query_cache_size`, que permite determinar el tamaño de la caché de peticiones, solo tiene alcance global.

#### a. Cambio para la sesión

Usando el comando SET SESSION `parámetro_sesion=valor`, el valor del parámetro cambiará solo para la sesión del cliente en la que se ejecutó el comando.

```
mysql> SHOW GLOBAL VARIABLES LIKE 'sort_buffer_size';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| sort_buffer_size | 2097144 |
+-----+-----+

mysql> SHOW SESSION VARIABLES LIKE 'sort_buffer_size';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| sort_buffer_size | 2097144 |
+-----+-----+

mysql> SET SESSION sort_buffer_size=1048576;

mysql> SHOW GLOBAL VARIABLES LIKE 'sort_buffer_size';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| sort_buffer_size | 2097144 |
+-----+-----+

mysql> SHOW SESSION VARIABLES LIKE 'sort_buffer_size';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| sort_buffer_size | 1048576 |
+-----+-----+
```

#### b. Cambio global

Con el comando `SET GLOBAL parámetro_global=valor`, el cambio es efectivo para todos los clientes que se conectan al servidor después de que se ejecute el comando. Sin embargo, este nuevo valor no se tiene en cuenta para las sesiones de los clientes conectados antes del cambio (sesiones en curso), que conservan el valor inicial del parámetro. Lo mismo sucede con la sesión del cliente en la que se ha ejecutado el comando.

Una excepción a esta regla se refiere a los parámetros que solo tienen un alcance global. La actualización se llevará a cabo también en la sesión del cliente que realiza el comando `SET GLOBAL`.

```
mysql> SHOW GLOBAL VARIABLES LIKE 'sort_buffer_size';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| sort_buffer_size | 2097144 |
+-----+-----+

mysql> SHOW SESSION VARIABLES LIKE 'sort_buffer_size';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| sort_buffer_size | 1048576 |
+-----+-----+

mysql> SET GLOBAL sort_buffer_size=5242880;

mysql> SHOW GLOBAL VARIABLES LIKE 'sort_buffer_size';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| sort_buffer_size | 5242880 |
+-----+-----+

mysql> SHOW SESSION VARIABLES LIKE 'sort_buffer_size';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| sort_buffer_size | 1048576 |
+-----+-----+
```

Es muy importante tener en cuenta que estos cambios en caliente no son persistentes. Están almacenados en el espacio de memoria utilizado por el servidor MySQL y no sobreviven, por lo tanto, a un reinicio del servidor. Los cambios realizados con el comando `SET SESSION` no resisten tampoco al reinicio del cliente, pues esto inicia una nueva sesión. Si quiere hacer que los cambios globales sean persistentes, debe ponerlos en el archivo de configuración (o también pasarlos como parámetros a `mysqld`).

## Visualización de la configuración

---

El Administrador tiene varias posibilidades para ver la configuración del servidor MySQL. La manera más obvia es pasar por el archivo de configuración. Sin embargo, es posible que algunas de las opciones que contiene sean redefinidas al pasarlas por parámetro a `mysqld`. También debemos pensar en mirar el estado del proceso `mysqld`; por ejemplo, con el comando `ps -ef | grep mysqld` en Linux.

Pero estos dos primeros métodos tienen algunas limitaciones. En un servidor mal administrado, puede haber varios archivos de configuración, que el servidor no utilice necesariamente, y que además no tienen por qué llamarse my.cnf (o my.ini). Es más, los parámetros no suelen aparecer todos en el archivo de configuración y los que sí aparecen pueden redefinirse en caliente, como sucede con el parámetro de mysqld.

Para estar seguro de ver la configuración actual del servidor, debemos utilizar el comando SHOW GLOBAL VARIABLES, es decir, las tablas GLOBAL\_VARIABLES del esquema virtual PERFORMANCE\_SCHEMA o el comando SELECT @@global.nombre\_variable.

```
mysql> select * from performance_schema.global_variables where  
variable_name = 'datadir';
```

| VARIABLE_NAME | VARIABLE_VALUE        |
|---------------|-----------------------|
| DATADIR       | /home/daz/mysql/data/ |

```
mysql> SHOW GLOBAL VARIABLES LIKE 'datadir';
```

| Variable_name | Value                 |
|---------------|-----------------------|
| datadir       | /home/daz/mysql/data/ |

```
mysql> SELECT @@global.datadir;
```

| @@global.datadir      |
|-----------------------|
| /home/daz/mysql/data/ |

Un error común es ejecutar el comando SHOW VARIABLES en lugar de SHOW GLOBAL VARIABLES. Si omitimos la cláusula GLOBAL, el comando se ejecuta implícitamente con la cláusula de SESSION. Lo que tiene el efecto de mostrar el estado de las variables para la sesión del cliente en la que se ejecuta el comando. En INFORMATION\_SCHEMA, las opciones de configuración relacionadas con la sesión en curso se encuentran en la tabla SESSION\_VARIABLES.

```
mysql> SHOW GLOBAL VARIABLES LIKE 'character_set_connection';
```

| Variable_name            | Value |
|--------------------------|-------|
| character_set_connection | utf8  |

```
mysql> select * from performance_schema.global_variables where  
variable_name like 'character_set_connection';
```

| VARIABLE_NAME            | VARIABLE_VALUE |
|--------------------------|----------------|
| CHARACTER_SET_CONNECTION | utf8           |

```
+-----+-----+
mysql> SHOW VARIABLES LIKE 'character_set_connection';
+-----+-----+
| Variable_name          | Value |
+-----+-----+
| character_set_connection | utf8  |
+-----+-----+

mysql> SHOW SESSION VARIABLES LIKE 'character_set_connection';
+-----+-----+
| Variable_name          | Value |
+-----+-----+
| character_set_connection | utf8  |
+-----+-----+

mysql> select * from performance_schema.session_variables where
variable_name like 'character_set_connection';
+-----+-----+
| VARIABLE_NAME          | VARIABLE_VALUE |
+-----+-----+
| CHARACTER_SET_CONNECTION | utf8           |
+-----+-----+
```

Otro comando que puede ser útil es STATUS. Proporciona, entre otras cosas, información sobre el estado del servidor e información sobre los juegos de caracteres utilizados:

```
mysql> STATUS
-----
mysql Ver 14.14 Distrib 5.6.7-rc, for linux2.6 (x86_64) using
EditLine wrapper
...
SSL: Not in use
Current pager: stdout
Using outfile: ''
Using delimiter: ;
...
Server version:          5.6.7-rc MySQL Community Server (GPL)
Protocol version:        10
Connection: Localhost via UNIX socket
Server charset: utf8
Db charset: utf8
Client charset: utf8
Conn. charset: utf8
UNIX socket: /tmp/mysql.sock
...
```

## LOS ARCHIVOS DE LOG

El servidor MySQL utiliza cuatro tipos de registros, cada uno con sus especialidades: el registro binario (binary log o binlog), el registro de peticiones lentas (slow query log), el registro general (general query log) y el registro de errores (error log), el único de los cuatro que se activará por defecto.

## 1. El registro binario

El registro binario o binlog se encarga de almacenar en un formato binario todas las peticiones que modifican los objetos de la base de datos (**INSERT, UPDATE, DELETE, DROP, CREATE, ALTER...**). Es el elemento central de la replicación MySQL. También es muy útil para la restauración de los datos. Para activar el registro binario, use la opción **log-bin**, y para definir su archivo de índice, configure la opción **log-bin-index**. Este archivo, que contiene la lista de todos los registros binarios desde la última purga, le permite al servidor conocer el nombre del archivo actual que se obtiene con el comando **SHOW MASTER STATUS** y también mostrar la lista de todos los registros binarios presentes en el servidor con **SHOW BINARY LOGS**. El registro binario se puede desactivar en caliente, pero solo para la sesión de un cliente, con la opción **SQL\_LOG\_BIN**. Una configuración básica podría ser:

```
[mysqld]
log-bin = /var/lib/mysql/mysql-bin
```

El formato utilizado para el registro es un formato binario; es significa que las entradas del registro no pueden leerse de forma directa. Por lo tanto, necesitará recurrir a `mysqlbinlog` cada vez que tenga que leer el contenido de los registros binarios, especificando el archivo o los archivos que desee decodificar.

```
$ mysqlbinlog /var/lib/mysql/mysql-bin.000001
```

El registro binario es un registro incremental cuyo incremento administra el servidor. El salto es de 1. El primer archivo binario creado tiene el formato `nombre_archivo.000001`; el siguiente, `nombre_archivo.000002`, y así sucesivamente. El paso al siguiente se realiza al cerrar el archivo en curso. Este paso de testigo puede realizarse de tres modos diferentes:

- En caso de reinicio del servidor.

Si se alcanza el tamaño máximo del archivo especificado con la opción `max_binlog_size`.

- En caso de ejecutar el comando **FLUSH LOGS**.

Observe, sin embargo, que los antiguos archivos no se purgan por defecto. Tendremos varias posibilidades:

- Una eliminación automática por el servidor MySQL de los registros de más de N días con la opción **expire\_logs\_days = N**.
- Una eliminación manual:
  - Bien en comparación con un intervalo de tiempo: **PURGE BINARY LOGS BEFORE**
  - O bien en comparación con un número de registro: **PURGE BINARY LOGS TO**
  - O bien borrándolo todo y volviendo a empezar con el archivo .000001: **RESET MASTER**.

## 2. El registro de peticiones lentas

El **registro de peticiones lentas** (slow query log) tiene como principal función ayudar al administrador de la base de datos a identificar las peticiones cuyo tiempo de ejecución es un problema. Su activación se

realiza en caliente o en frío con la opción **slow\_query\_log**, y todas las peticiones cuyo tiempo de ejecución supere el valor de la opción **long\_query\_time** (en segundos) serán registradas. Ejemplo de configuración

```
[mysqld]
slow_query_log = ON
slow_query_log_file = /var/lib/mysql/mysql-slow.log
long-query-time = 1
```

El objetivo principal de un servidor de bases de datos es ejecutar lo antes posible las peticiones. El registro de peticiones lentas es una mina de oro para quien desea mejorar el rendimiento. Una primera forma de utilizar este registro es colocar la variable **long\_query\_time** con un valor alto (por ejemplo, 5 o 10) para identificar las peticiones extremadamente lentas. Otra forma de proceder es poner **long\_query\_time = 0** para grabar un histórico de todas las peticiones realizadas por el servidor. En este caso, se aconseja vigilar el espacio en disco, ya que un servidor cargado puede escribir 1 GB por minuto en el registro de peticiones lentas.

### 3. El registro de errores

El registro de errores (error log) es el único de los cuatro registros que se activa por defecto. Se reconoce a través de su extensión **.err**. El prefijo es por defecto el nombre del equipo. Su ubicación se obtiene con el comando **SHOW GLOBAL VARIABLES LIKE 'log\_error'**. Puede cambiarse, así como el nombre del archivo, poniéndolo en la opción **log-error**. En caso de fallo o de comportamiento anormal, es la herramienta preferida del DBA en curso de investigación. Este registro contiene la información relativa a las advertencias y errores del servidor MySQL. Aquí encontraremos también los mensajes relacionados con los motores de almacenamiento o las funcionalidades avanzadas, como la replicación o el programador de eventos (event scheduler).

Elección de la ubicación del registro de errores en el archivo de configuración:

```
[mysqld]
log-error = /usr/local/mysql/logs/mysql-error.err
```

**4. El registro general** El registro general (general log) registra los acontecimientos recibidos por mysqld, es decir, las peticiones válidas o no enviadas por los clientes y la información de conexión/desconexión de los clientes. Al igual que el registro de peticiones lentas, lo podemos activar y desactivar en caliente o en frío con la opción **general\_log**. También podemos almacenar la información, ya sea en un archivo o en una tabla, pero como para el registro de peticiones lentas, se debe siempre optar por usar un archivo en la medida de lo posible.

A diferencia del registro de peticiones lentas, que almacena las peticiones después de su ejecución, el registro general almacena las peticiones antes de su ejecución. Por lo tanto, el registro general no cuenta con ninguna información disponible en el registro de peticiones lentas. Solo sirve para mantener una constancia de las peticiones o conexiones con error.

Por lo tanto, es raro tener que activar el registro general, ya que puede llenar el disco con mucha rapidez y tener un impacto importante sobre el rendimiento.

### 5. Mejores prácticas

## **a. Configuración**

Con un uso convencional de MySQL, la configuración estándar consiste en:

- Activar el registro binario. Es necesario para restaurar los datos modificados desde la última copia. Además, será obligatorio para la replicación.
- Activar el registro de peticiones lentas. Aquí encontraremos las peticiones que se deben optimizar.
- No activar el registro general, salvo necesidades específicas (auditoría, por ejemplo).

## **b. Supervisar el uso del disco**

Implemente procedimientos para supervisar los espacios de almacenamiento. En función de la actividad del servidor (volumen y carga), algunos archivos pueden crecer de forma rápida. Este es concretamente el caso del registro general.

## **c. Impacto sobre el rendimiento**

La activación de los registros tiene un impacto negativo sobre el rendimiento, que no es fácil de evaluar, ya que depende de nuestra aplicación y de los registros activados. Nos corresponde encontrar el equilibrio entre lo que aportan y las prestaciones que requerimos. Observe, sin embargo, el registro en las tablas del sistema (registro general y registro de peticiones lentas), ya que es mucho más pesado que el registro en archivos.