

## Introducción

En este laboratorio, comenzaremos a controlar el funcionamiento y uso de nuestro servidor de bases de datos. Para esto, utilizaremos directamente los recursos que nos proporciona el propio sistema de bases de datos, aunque también instalaremos una herramienta un poco más sofisticada que nos facilitará nuestra tarea. De esta forma, podremos observar todas las acciones que hemos ido ejecutando en el servidor, así como comprobar la configuración de la política de seguridad que hemos establecido.

## Objetivos

Los objetivos de este laboratorio son:

- Aprender a consultar la configuración de seguridad del servidor.

- Conocer algunas opciones de configuración, especialmente en todo lo relacionado al registro de sucesos.

- Instalar y utilizar una herramienta que nos facilita la configuración de la política de seguridad y auditoría.

## Requisitos previos

Para realizar este laboratorio utilizaremos el servidor de bases de datos que obtuvimos como resultado del Laboratorio 1. El servidor tiene que estar correctamente configurado para ser accesible remotamente.

## Procedimiento a seguir

A continuación, se describen los pasos a realizar suponiendo que el servidor de bases de datos y la conexión de red están correctamente configurados.

### 1 *Antes de comenzar*

Una de las principales responsabilidades del administrador de bases de datos es mantener los programas del sistema actualizados. En nuestro caso, puede suceder que desde que instalamos la máquina hayan aparecido actualizaciones, o incluso que sea necesario reiniciar el servidor para que alguna actualización se pueda aplicar.

```
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-1029-aws x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage

System information as of Tue Feb 23 09:48:51 UTC 2021

System load:  0.05          Processes:            113
Usage of /:   10.1% of 29.02GB Users logged in:          0
Memory usage: 55%          IPv4 address for eth0: 172.31.18.109
Swap usage:   0%

* Introducing self-healing high availability clusters in MicroK8s.
  Simple, hardened, Kubernetes for production, from RaspberryPi to DC.

  https://microk8s.io/high-availability

25 updates can be installed immediately.
8 of these updates are security updates.
To see these additional updates run: apt list --upgradable

3 updates could not be installed automatically. For more details,
see /var/log/unattended-upgrades/unattended-upgrades.log

Last login: Tue Feb 23 09:48:26 2021 from 158.227.0.241
ubuntu@ip-172-31-18-109:~$
```

Siempre que nos encontremos con actualizaciones pendientes, tenemos que actualizar el software de nuestra máquina. En primer lugar, actualizaremos los repositorios:

```
:~$ sudo apt-get update
```

```
ubuntu@ip-172-31-18-109:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security InRelease [109 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [815 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main Translation-en [197 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 c-n-f Metadata [12.7 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [745 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/universe Translation-en [154 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/universe amd64 c-n-f Metadata [16.0 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 Packages [17.2 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/multiverse Translation-en [5216 B]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 c-n-f Metadata [560 B]
Get:14 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [497 kB]
Get:15 http://security.ubuntu.com/ubuntu focal-security/main amd64 c-n-f Metadata [7060 B]
Get:16 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [541 kB]
Get:17 http://security.ubuntu.com/ubuntu focal-security/universe Translation-en [77.3 kB]
Get:18 http://security.ubuntu.com/ubuntu focal-security/universe amd64 c-n-f Metadata [10.3 kB]
Fetched 3419 kB in 1s (2932 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-18-109:~$
```

A continuación, actualizaremos el software de nuestra máquina (esta operación podría llevar varios minutos si hay bastantes paquetes por actualizar):

```
:~$ sudo apt-get upgrade
```

```
ubuntu@ip-172-31-18-109:~$ sudo apt-get upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages have been kept back:
  linux-aws linux-headers-aws linux-image-aws
The following packages will be upgraded:
  bind9-dnsutils bind9-host bind9-libs dirmngr friendly-recovery gnupg gnupg-l10n gnupg-utils gpg
  gpg-agent gpg-wks-client gpg-wks-server gpgconf gpgsm gpgv grub-common grub-pc grub-pc-bin grub2-common
  libldap-2.4-2 libldap-common libssl1.1 openssl pollinate python3-software-properties
  software-properties-common
26 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
Need to get 9599 kB of archives.
After this operation, 4096 B of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 gpg-wks-client amd64 2.2.19-3
```

## 2 Operaciones de auditoría simples

Sin necesidad de instalar ninguna aplicación ni consultar la configuración del servidor, podemos auditar parte de la configuración de seguridad de nuestro sistema. En concreto, una de las primeras acciones a realizar consiste en comprobar los privilegios asignados a cada usuario, especialmente la de aquellos usuarios que tienen la posibilidad de propagarlos.

La comprobación de los permisos asignados a cada usuario puede ser realizada consultando la tabla “*user\_privileges*” de “*information\_schema*”. Por ejemplo, desde cualquier cliente de *MySQL* usando el usuario “*root*”(o cualquier otro usuario con privilegios de administración):

```
mysql> select * from information_schema.user_privileges;
```

```
mysql> select * from information_schema.user_privileges;
```

GRANTEE	TABLE_CATALOG	PRIVILEGE_TYPE	IS_GRANTABLE
'admDBCoin'@'localhost'	def	CREATE	NO
'admDBCoin'@'localhost'	def	RELOAD	NO
'admDBCoin'@'localhost'	def	ALTER	NO
'admDBet'@'localhost'	def	RELOAD	NO
'admDBet'@'localhost'	def	CREATE USER	NO
'debian-sys-maint'@'localhost'	def	SELECT	YES
'debian-sys-maint'@'localhost'	def	INSERT	YES
'debian-sys-maint'@'localhost'	def	UPDATE	YES
'debian-sys-maint'@'localhost'	def	DELETE	YES
'debian-sys-maint'@'localhost'	def	CREATE	YES
'debian-sys-maint'@'localhost'	def	DROP	YES
'debian-sys-maint'@'localhost'	def	RELOAD	YES
'debian-sys-maint'@'localhost'	def	SHUTDOWN	YES

Pero, si además queremos hacer énfasis en aquellos permisos que pueden ser propagados, podemos utilizar la siguiente sentencia:

```
mysql> select * from information_schema.user_privileges
        where is_grantable = 'YES';
```

```
mysql> select * from information_schema.user_privileges where is_grantable = 'YES';
```

GRANTEE	TABLE_CATALOG	PRIVILEGE_TYPE	IS_GRANTABLE
'debian-sys-maint'@'localhost'	def	SELECT	YES
'debian-sys-maint'@'localhost'	def	INSERT	YES
'debian-sys-maint'@'localhost'	def	UPDATE	YES
'debian-sys-maint'@'localhost'	def	DELETE	YES
'debian-sys-maint'@'localhost'	def	CREATE	YES
'debian-sys-maint'@'localhost'	def	DROP	YES
'debian-sys-maint'@'localhost'	def	RELOAD	YES
'debian-sys-maint'@'localhost'	def	SHUTDOWN	YES
'debian-sys-maint'@'localhost'	def	PROCESS	YES
'debian-sys-maint'@'localhost'	def	FILE	YES
'debian-sys-maint'@'localhost'	def	REFERENCES	YES
'debian-sys-maint'@'localhost'	def	INDEX	YES
'debian-sys-maint'@'localhost'	def	ALTER	YES

De manera alternativa, los permisos asignados a un usuario también se pueden consultar a través de la interfaz web de *phpMyAdmin*, en la sección “Cuentas de usuarios”:

Servidor: localhost:3306

Bases de datos SQL Estado actual Cuentas de usuarios Exportar Importar Configuración Registro binario

Vista global de las cuentas de usuario Grupos de usuario

### Vista global de las cuentas de usuario

<input type="checkbox"/>	Nombre de usuario	Nombre del servidor	Contraseña	Privilegios globales ⓘ	Grupo de usuario	Conceder	Acción
<input type="checkbox"/>	LUser1	%	Si	USAGE		No	Editar privilegios  Exportar
<input type="checkbox"/>	LUser2	%	Si	USAGE		No	Editar privilegios  Exportar
<input type="checkbox"/>	Mlser1	%	Si	USAGE		No	Editar privilegios  Exportar
<input type="checkbox"/>	Mlser2	%	Si	USAGE		No	Editar privilegios  Exportar
<input type="checkbox"/>	admDBCoin	localhost	Si	CREATE, RELOAD, ALTER		No	Editar privilegios  Exportar
<input type="checkbox"/>	admDBet	localhost	Si	RELOAD, CREATE USER		No	Editar privilegios  Exportar
<input type="checkbox"/>	debian-sys-maint	localhost	Si	ALL PRIVILEGES		Si	Editar privilegios  Exportar
<input type="checkbox"/>	dummy	%	Si	USAGE		No	Editar privilegios  Exportar
<input type="checkbox"/>	lel	localhost	Si	USAGE		No	Editar privilegios  Exportar
<input type="checkbox"/>	mysql.infoschema	localhost	Si	SELECT		No	Editar privilegios  Exportar
<input type="checkbox"/>	mysql.session	localhost	Si	SHUTDOWN, SUPER		No	Editar privilegios  Exportar
<input type="checkbox"/>	mysql.sys	localhost	Si	USAGE		No	Editar privilegios  Exportar
<input type="checkbox"/>	phpmyadmin	localhost	Si	USAGE		No	Editar privilegios  Exportar
<input type="checkbox"/>	root	localhost	Si	ALL PRIVILEGES		Si	Editar privilegios  Exportar
<input type="checkbox"/>	sh test user	%	Si	USAGE		No	Editar privilegios  Exportar

*MySQL* también proporciona comandos que muestran información del sistema y, entre ellos, los permisos asignados a usuarios. Para consultar los permisos del usuario actual se pueden utilizar las siguientes sentencias (son equivalentes):

```
mysql> show grants;
```

```
mysql> show grants for current_user;
```

```
mysql>
mysql> show grants for current_user;
+-----+
| Grants for root@localhost |
+-----+
| GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, RELOAD, SHUTDOWN, PROCESS, FILE, REFERENCES, INDEX, ALTER, SHOW DATABASES, SUPER, CREATE TEMPORARY TABLES, LOCK TABLES, EXECUTE, REPLICATION SLAVE, REPLICATION CLIENT, CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, CREATE USER, EVENT, TRIGGER, CREATE TABLESPACE, CREATE ROLE, DROP ROLE ON *.* TO 'root'@'localhost' WITH GRANT OPTION |
| GRANT APPLICATION_PASSWORD_ADMIN, AUDIT_ADMIN, BACKUP_ADMIN, BINLOG_ADMIN, BINLOG_ENCRYPTION_ADMIN, CLONE_ADMIN, CONNECTION_ADMIN, ENCRYPTION_KEY_ADMIN, FLUSH_OPTIMIZER_COSTS, FLUSH_STATUS, FLUSH_TABLES, FLUSH_USER_RESOURCES, GROUP_REPLICATION_ADMIN, INNODB_REDO_LOG_ARCHIVE, INNODB_REDO_LOG_ENABLE, PERSIST_RO_VARIABLES_ADMIN, REPLICATION_APPLIER, REPLICATION_SLAVE_ADMIN, RESOURCE_GROUP_ADMIN, RESOURCE_GROUP_USER, ROLE_ADMIN, SERVICE_CONNECTION_ADMIN, SESSION_VARIABLES_ADMIN, SET_USER_ID, SHOW_ROUTINE, SYSTEM_USER, SYSTEM_VARIABLES_ADMIN, TABLE_ENCRYPTION_ADMIN, XA_RECOVER_ADMIN ON *.* TO 'root'@'localhost' WITH GRANT OPTION |
| GRANT PROXY ON ''@'' TO 'root'@'localhost' WITH GRANT OPTION |
+-----+
```



También se pueden consultar los permisos asignados a otro usuario:

```
mysql> show grants for dummy;
```

```
mysql> show grants for dummy;
```

```
+-----+  
| Grants for dummy@% |  
+-----+  
| GRANT USAGE ON *.* TO `dummy`@`%` |  
+-----+  
1 row in set (0.00 sec)  
  
mysql> █
```

Otros comandos interesantes son los siguientes:

```
mysql> show status;
```

Proporciona información acerca de algunos parámetros del sistema o de las últimas operaciones realizadas.

```
mysql> show status;
```

Variable_name	Value
Aborted_clients	0
Aborted_connects	0
Acl_cache_items_count	0
Binlog_cache_disk_use	0
Binlog_cache_use	2
Binlog_stmt_cache_disk_use	0
Binlog_stmt_cache_use	0
Bytes_received	621
Bytes_sent	33124
Caching_sha2_password_rsa_public_key	-----BEGIN PUBLIC KEY----- MIB1JANBqkqhki09w0RGAOACAQAMII8CpKCAQEA8B7JUN7dDRnyQ1Z8fg WZPlle7Eegk1ap3U5Ury30eAeA2wWTEW44vORhNqEK1K01POTFpaFfRo0e+TY1 b+71E/T87hb/hZk4U5V23yM10tLXtakoU8D+HFBW8KCBAY5sOXNDYARmo8B wBvdIMioR8Sd1Hr1emq180T1a94bXTCp2m8QW6G1u+U93nte0112toq488kgb QcVdSdu0vFLRw0kMy1S40ur1uFMShuRKEVt4a32aonVE4sCfp027N28Wfc cC1ALM0cCB410SRyDVNkqB1FQNT3xx1GTUTC5q20p1YMKqpFX4GM1CUHMBP1 OwIDAQAB -----END PUBLIC KEY-----
Com_admin_commands	0
Com_assign_to_keycache	0
Com_alter_db	0
Com_alter_event	0
Com_alter_function	0
Com_alter_instance	0

```
mysql> show databases;
```

Muestra las bases de datos que existen en el sistema.

```
mysql> show databases;
+-----+
| Database |
+-----+
| DBCoin   |
| DBet     |
| WallStreet |
| example  |
| information_schema |
| mysql     |
| performance_schema |
| phpmyadmin |
| sbtest    |
| sys       |
+-----+
10 rows in set (0.01 sec)

mysql> █
```

```
mysql> show tables;
```

Muestra las tablas que existen en una base de datos del sistema (previamente seleccionada mediante “*use*”).

```
mysql> use DBCoin;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_DBCoin |
+-----+
| Coin              |
| Owner             |
| Transfer          |
+-----+
3 rows in set (0.00 sec)

mysql>
```

### **3 Configuración de características de *MySQL* para auditoría**

Como ya discutimos en clase, una de las principales fuentes de información para auditar un sistema de bases de datos es el diario. Al margen del diario que usa el sistema operativo, *MySQL* utiliza 3 diarios que resultan útiles para auditoría:

El *diario general*, donde se registran las operaciones generales del servidor de bases de datos.

El *diario de error*, donde se registran las paradas o puestas en marcha del servidor, así como los errores críticos que han sucedido durante la ejecución del servidor de bases de datos.

El *diario de consultas lentas*, donde se registran las consultas que requieren un tiempo de ejecución mayor que el establecido (*long\_query\_time*).

Además de estos 3, existe un 4º diario que se utiliza principalmente para recuperación llamado *diario binario*. Se presentará con detalle en el 4º laboratorio de la asignatura.

Por defecto, solo el diario de error se encuentra activado (y no conviene desactivarlo). El resto de diarios vienen desactivados para ahorrar espacio en disco.

Los diarios generales y de consultas lentas se pueden consultar desde la consola de “*mysql*”, aunque también se pueden leer directamente desde los ficheros que los contienen.

Las opciones de *log* (así como el resto de opciones de configuración de *MySQL*) se pueden establecer mediante directivas al proceso *mysqld*, que es el proceso general de *MySQL*. Sin embargo, es más recomendable modificar el fichero de configuración de *MySQL*, para que estas opciones puedan ser consultadas y utilizadas en cualquier momento. El fichero de configuración de *MySQL* es:

`/etc/mysql/my.cnf`

Para consultar este fichero, podéis utilizar el siguiente comando desde la consola de Linux:

`:~$ cat /etc/mysql/my.cnf`

```
ubuntu@ip-172-31-18-109:~$ cat /etc/mysql/my.cnf
#
# The MySQL database server configuration file.
#
# You can copy this to one of:
# - "/etc/mysql/my.cnf" to set global options,
# - "~/.my.cnf" to set user-specific options.
#
# One can use all long options that the program supports.
# Run program with --help to get a list of available options and with
# --print-defaults to see which it would actually understand and use.
#
# For explanations see
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html
#
# * IMPORTANT: Additional settings that can override those from this file!
#   The files must end with '.cnf', otherwise they'll be ignored.
#
!includedir /etc/mysql/conf.d/
!includedir /etc/mysql/mysql.conf.d/
```

Como recordaréis del laboratorio anterior, este fichero *my.cnf* sirve para importar la configuración desde 2 carpetas. De estas dos, el fichero que necesitamos editar es:

`/etc/mysql/mysql.conf.d/mysqld.cnf`

En este fichero, podemos activar/desactivar los diarios, así como indicar los ficheros de registro (o *log*) en los cuales queremos que se guarden. Como en otros ficheros de configuración, el valor 0 se corresponde con desactivado y 1 con activado. Además, el símbolo *#* sirve para comentar una línea. Existe un *flag* (opción de configuración, variable global) para activar/desactivar el diario general y el de consultas lentas.

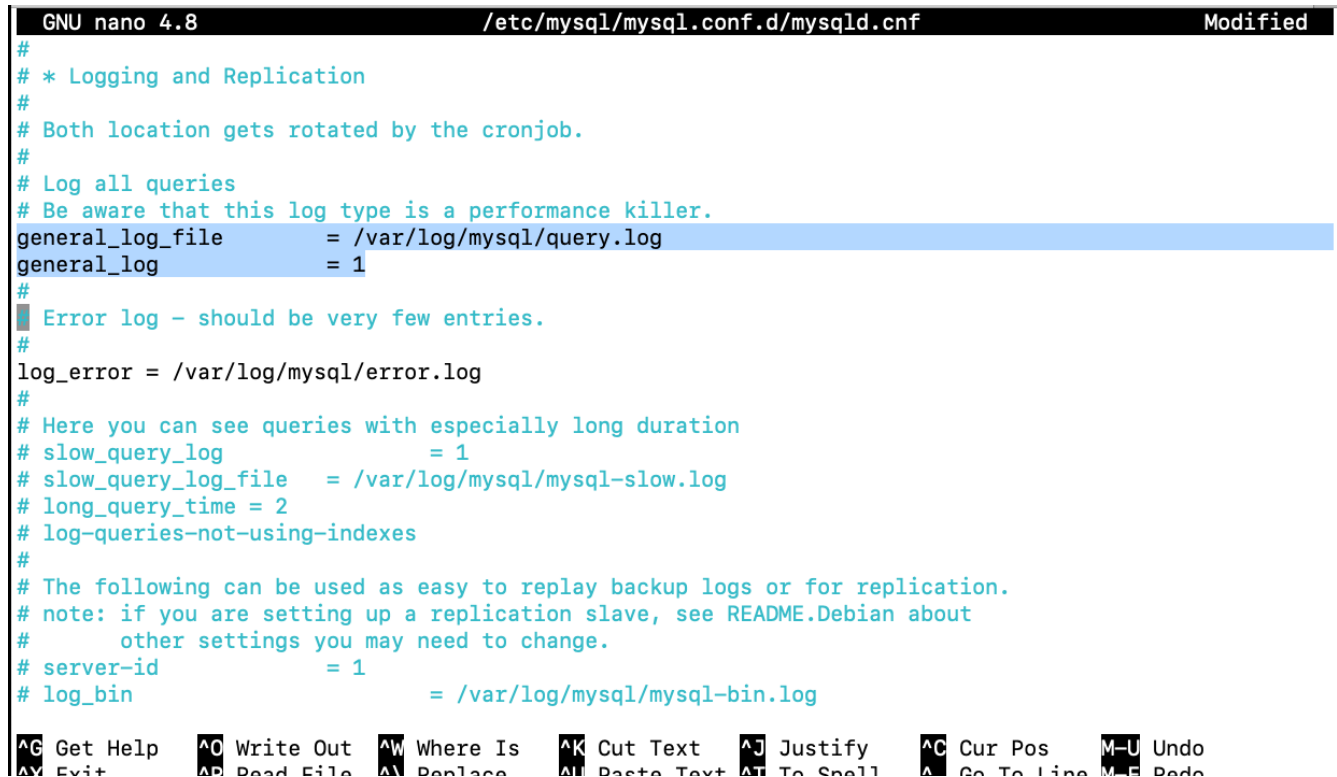
```
#
# * Logging and Replication
#
# Both location gets rotated by the cronjob.
#
# Log all queries
# Be aware that this log type is a performance killer.
# general_log_file      = /var/log/mysql/query.log
# general_log           = 1
#
# Error log - should be very few entries.
#
log_error = /var/log/mysql/error.log
#
# Here you can see queries with especially long duration
# slow_query_log        = 1
# slow_query_log_file   = /var/log/mysql/mysql-slow.log
# long_query_time = 2
# log-queries-not-using-indexes
#
# The following can be used as easy to replay backup logs or for replication.
# note: if you are setting up a replication slave, see README.Debian about
#       other settings you may need to change.
# server-id             = 1
# log_bin               = /var/log/mysql/mysql-bin.log
# binlog_expire_logs_seconds = 2502000
```

Consultando el fichero de configuración, podemos comprobar que los ficheros de *log* se almacenan por defecto en el directorio:

`/var/log/mysql/`

Para activar el diario general, descomentamos las siguientes líneas del fichero:

```
general_log_file = /var/log/mysql/query.log
general_log = 1
```



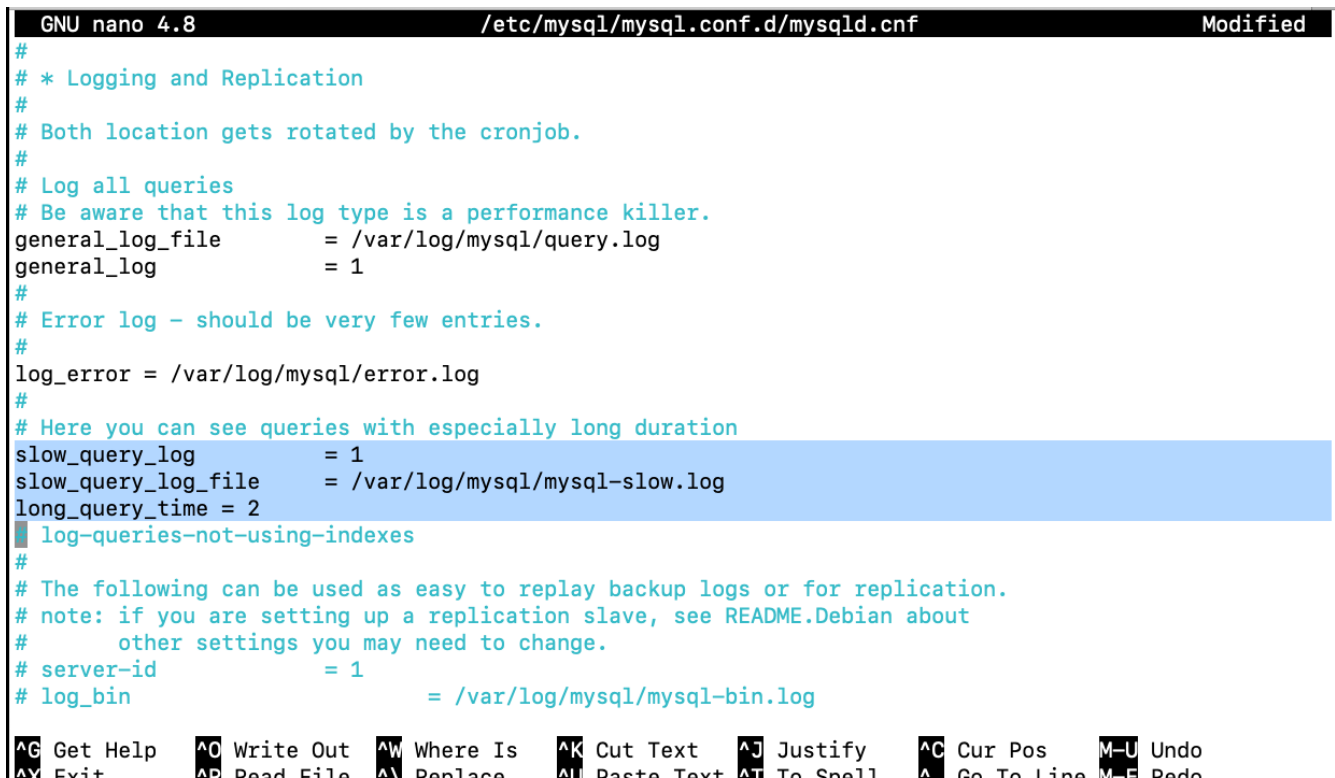
```
GNU nano 4.8 /etc/mysql/mysql.conf.d/mysqld.cnf Modified
#
# * Logging and Replication
#
# Both location gets rotated by the cronjob.
#
# Log all queries
# Be aware that this log type is a performance killer.
general_log_file = /var/log/mysql/query.log
general_log = 1
#
# Error log - should be very few entries.
#
log_error = /var/log/mysql/error.log
#
# Here you can see queries with especially long duration
# slow_query_log = 1
# slow_query_log_file = /var/log/mysql/mysql-slow.log
# long_query_time = 2
# log-queries-not-using-indexes
#
# The following can be used as easy to replay backup logs or for replication.
# note: if you are setting up a replication slave, see README.Debian about
#       other settings you may need to change.
# server-id = 1
# log_bin = /var/log/mysql/mysql-bin.log

^G Get Help  ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos    M-U Undo
^X Exit      ^P Read File  ^R Replace    ^U Paste Text ^T To Spell   ^_ Go To Line  M-E Redo
```



Y para activar el diario de consultas lentas, utilizaremos las siguientes opciones de configuración:

```
slow_query_log = 1
slow_query_log_file = /var/log/mysql/mysql-slow.log
long_query_time = 2
```



```
GNU nano 4.8 /etc/mysql/mysql.conf.d/mysqld.cnf Modified
#
# * Logging and Replication
#
# Both location gets rotated by the cronjob.
#
# Log all queries
# Be aware that this log type is a performance killer.
general_log_file      = /var/log/mysql/query.log
general_log           = 1
#
# Error log - should be very few entries.
#
log_error = /var/log/mysql/error.log
#
# Here you can see queries with especially long duration
slow_query_log        = 1
slow_query_log_file    = /var/log/mysql/mysql-slow.log
long_query_time = 2
log-queries-not-using-indexes
#
# The following can be used as easy to replay backup logs or for replication.
# note: if you are setting up a replication slave, see README.Debian about
#       other settings you may need to change.
# server-id           = 1
# log_bin              = /var/log/mysql/mysql-bin.log

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos   M-U Undo
^X Exit      ^R Read File  ^\ Replace  ^P Paste Text ^T To Spell  ^_ Go To Line M-F Redo
```

Una vez guardado este fichero, tendremos que reiniciar el servidor de *MySQL* utilizando el siguiente comando:

```
:~$ sudo service mysql restart
```

```
ubuntu@ip-172-31-18-109:~$ sudo service mysql restart
ubuntu@ip-172-31-18-109:~$
ubuntu@ip-172-31-18-109:~$
```

Ahora, podemos comprobar que en el *log* de error de *MySQL* se ha registrado el reinicio del servidor *MySQL*.

```
ubuntu@ip-172-31-18-109:~$ tail -n 10 /var/log/mysql/error.log
2021-02-23T09:46:18.654081Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready for connections. Version: '8.0.23-0ubuntu0.20.04.1' socket: '/var/run/mysqld/mysqld.sock' port: 3306 (Ubuntu).
2021-02-23T10:49:52.604881Z 0 [System] [MY-013172] [Server] Received SHUTDOWN from user <via user signal>. Shutting down mysqld (Version: 8.0.23-0ubuntu0.20.04.1).
2021-02-23T10:49:54.125931Z 0 [System] [MY-010910] [Server] /usr/sbin/mysqld: Shutdown complete (mysqld 8.0.23-0ubuntu0.20.04.1) (Ubuntu).
2021-02-23T10:49:54.642391Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (mysqld 8.0.23-0ubuntu0.20.04.1) starting as process 5762
2021-02-23T10:49:54.651358Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
2021-02-23T10:49:55.248847Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
2021-02-23T10:49:55.493585Z 0 [System] [MY-011323] [Server] X Plugin ready for connections. Bind-address: '::' port: 33060, socket: /var/run/mysqld/mysqldx.sock
2021-02-23T10:49:55.608292Z 0 [Warning] [MY-010068] [Server] CA certificate ca.pem is self signed.
2021-02-23T10:49:55.608473Z 0 [System] [MY-013602] [Server] Channel mysql_main configured to support TLS. Encrypted connections are now supported for this channel.
2021-02-23T10:49:55.655679Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready for connections. Version: '8.0.23-0ubuntu0.20.04.1' socket: '/var/run/mysqld/mysqld.sock' port: 3306 (Ubuntu).
ubuntu@ip-172-31-18-109:~$
```

Probad a realizar distintas operaciones con las bases de datos creadas para el Laboratorio 2 e id comprobando cómo las operaciones se van registrando en los diarios, en particular en el diario general:

```
:~$ sudo cat /var/log/mysql/query.log
```

```
ubuntu@ip-172-31-18-109:~$ sudo cat /var/log/mysql/query.log
/usr/sbin/mysqld, Version: 8.0.23-0ubuntu0.20.04.1 ((Ubuntu)). started with:
Tcp port: 3306 Unix socket: /var/run/mysqld/mysqld.sock
Time          Id Command      Argument
2021-02-23T10:53:39.408904Z      8 Connect    root@localhost on  using Socket
2021-02-23T10:53:39.409360Z      8 Query       select @@version_comment limit 1
2021-02-23T10:53:52.254748Z      8 Query       show databases
2021-02-23T10:54:02.512021Z      8 Query       SELECT DATABASE()
2021-02-23T10:54:02.512312Z      8 Init DB     DBCoin
2021-02-23T10:54:02.513256Z      8 Query       show databases
2021-02-23T10:54:02.513972Z      8 Query       show tables
2021-02-23T10:54:02.519139Z      8 Field List   Coin
2021-02-23T10:54:02.526199Z      8 Field List   Owner
2021-02-23T10:54:02.527704Z      8 Field List   Transfer
2021-02-23T10:54:09.631717Z      8 Query       select * from Owner
2021-02-23T10:54:11.073820Z      8 Quit
ubuntu@ip-172-31-18-109:~$
```

¿Qué operaciones se han realizado en el sistema según la información en la siguiente figura?

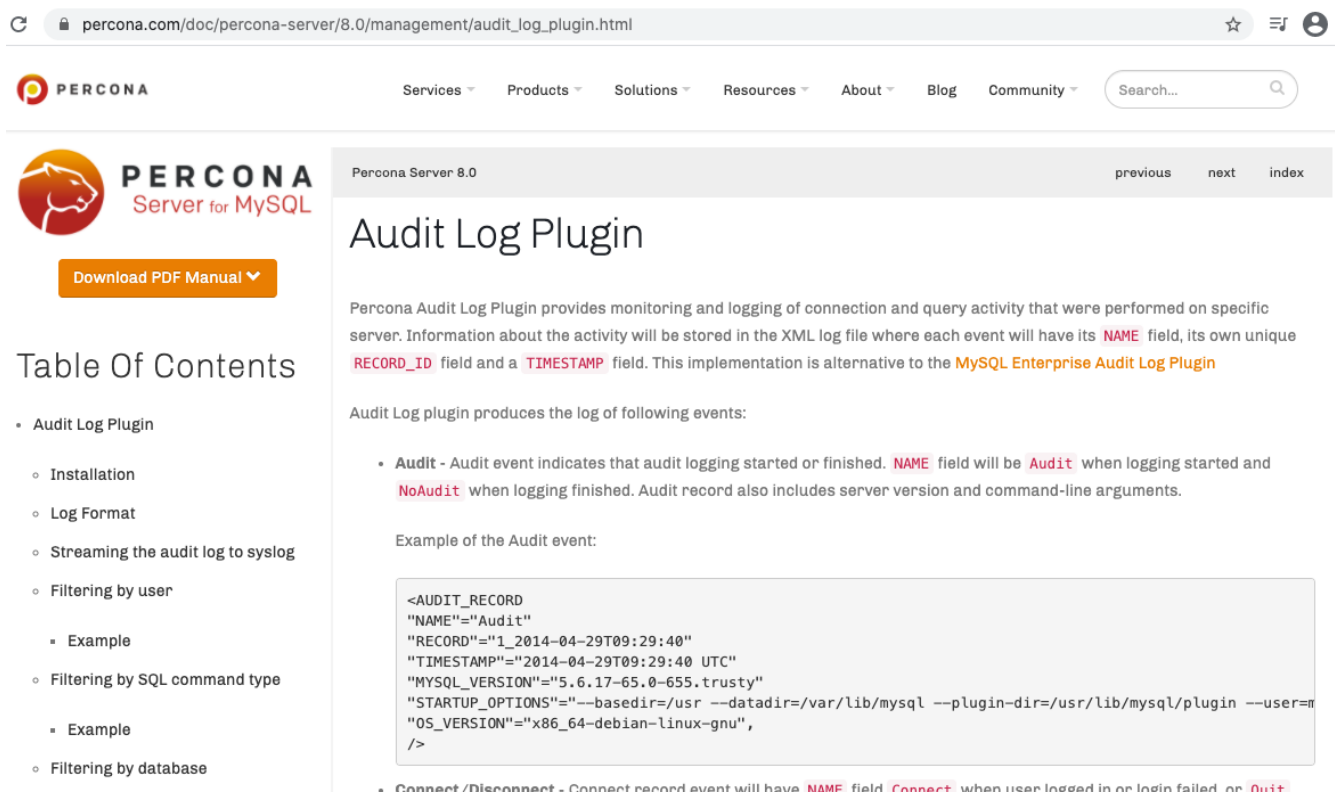
## 4 Auditoría en MySQL con un plug-in externo

Hasta ahora, nos hemos limitado a registrar y a consultar información acerca del funcionamiento de nuestro servidor de bases de datos. Sin embargo, una de las principales ventajas del uso de herramientas específicas de auditoría es que éstas facilitan el análisis de la información.

Desgraciadamente, el *plugin* para auditoría de MySQL solo está disponible en la versión comercial: <https://dev.mysql.com/doc/refman/8.0/en/audit-log.html>

Por eso, para esta sección vamos a utilizar el plugin de auditoría de MySQL de Percona, un proveedor de un SGBD y *plugins* compatible con MySQL. Toda la información sobre la última versión de este *plugin* está en su web oficial:

[https://www.percona.com/doc/percona-server/8.0/management/audit\\_log\\_plugin.html](https://www.percona.com/doc/percona-server/8.0/management/audit_log_plugin.html)



percona.com/doc/percona-server/8.0/management/audit\_log\_plugin.html

PERCONA

Services Products Solutions Resources About Blog Community Search...

PERCONA Server for MySQL

Download PDF Manual

Table Of Contents

- Audit Log Plugin
  - Installation
  - Log Format
  - Streaming the audit log to syslog
  - Filtering by user
    - Example
  - Filtering by SQL command type
    - Example
  - Filtering by database

Percona Server 8.0

# Audit Log Plugin

Percona Audit Log Plugin provides monitoring and logging of connection and query activity that were performed on specific server. Information about the activity will be stored in the XML log file where each event will have its **NAME** field, its own unique **RECORD\_ID** field and a **TIMESTAMP** field. This implementation is alternative to the **MySQL Enterprise Audit Log Plugin**

Audit Log plugin produces the log of following events:

- Audit** - Audit event indicates that audit logging started or finished. **NAME** field will be **Audit** when logging started and **NoAudit** when logging finished. Audit record also includes server version and command-line arguments.

Example of the Audit event:

```
<AUDIT_RECORD
  "NAME"="Audit"
  "RECORD"="1_2014-04-29T09:29:40"
  "TIMESTAMP"="2014-04-29T09:29:40 UTC"
  "MYSQL_VERSION"="5.6.17-65.0-655.trusty"
  "STARTUP_OPTIONS"="--basedir=/usr --datadir=/var/lib/mysql --plugin-dir=/usr/lib/mysql/plugin --user=
  "OS_VERSION"="x86_64-debian-linux-gnu",
/>
```

- Connect/Disconnect** - Connect record event will have **NAME** field **Connect** when user logged in or login failed, or **Quit**

Para instalar este *plugin* en nuestro servidor hay que copiar un fichero llamado “audit\_log.so” en un directorio concreto de nuestra instalación de MySQL. Este fichero se incluye con la instalación completa de Percona pero no se distribuye de manera separada. Por ello y para facilitar su uso en este laboratorio, hemos creado un repositorio con el fichero extraído: <https://github.com/ulopeznovoa/ABD-DBK-Lab3-Plugin>

El fichero audit\_log.so debe incluirse en la carpeta `/usr/lib/mysql/plugin` de la instalación local para que MySQL lo reconozca. Con el siguiente comando se puede descargar el fichero directamente desde el repositorio y guardarlo en la carpeta objetivo (es necesario utilizar “sudo” para acceder a la carpeta “/usr/...”):

```
:~$ sudo wget -P /usr/lib/mysql/plugin https://github.com/ulopeznovoa/ABD-DBK-Lab3-Plugin/raw/main/audit_log.so
```

```
ubuntu@ip-172-31-18-109:~$ sudo wget -P /usr/lib/mysql/plugin https://github.com/ulopeznovoa/ABD-DBK-Lab3-Plugin/raw/main/audit_log.so
--2021-02-23 12:03:45-- https://github.com/ulopeznovoa/ABD-DBK-Lab3-Plugin/raw/main/audit_log.so
Resolving github.com (github.com)... 140.82.113.4
Connecting to github.com (github.com)|140.82.113.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://raw.githubusercontent.com/ulopeznovoa/ABD-DBK-Lab3-Plugin/main/audit_log.so [following]
--2021-02-23 12:03:45-- https://raw.githubusercontent.com/ulopeznovoa/ABD-DBK-Lab3-Plugin/main/audit_log.so
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 185.199.111.133, 185.199.110.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 84080 (82K) [application/octet-stream]
Saving to: '/usr/lib/mysql/plugin/audit_log.so'

audit_log.so          100%[=====>] 82.11K  --.-KB/s    in 0.002s

2021-02-23 12:03:46 (53.0 MB/s) - '/usr/lib/mysql/plugin/audit_log.so' saved [84080/84080]

ubuntu@ip-172-31-18-109:~$
```

Una vez que el fichero esté descargado en el directorio indicado, se debe activar desde la consola de MySQL con el siguiente comando:

```
mysql> INSTALL PLUGIN audit_log SONAME 'audit_log.so';
```

Y después verificar que aparece entre los plugins reconocidos en MySQL:

```
mysql> SHOW PLUGINS;
```

```
mysql> INSTALL PLUGIN audit_log SONAME 'audit_log.so';
Query OK, 0 rows affected (0.02 sec)

mysql> SHOW PLUGINS;
```

Name	Status	Type	Library	License
binlog	ACTIVE	STORAGE ENGINE	NULL	GPL
mysql_native_password	ACTIVE	AUTHENTICATION	NULL	GPL
sha256_password	ACTIVE	AUTHENTICATION	NULL	GPL
caching_sha2_password	ACTIVE	AUTHENTICATION	NULL	GPL
audit_log	ACTIVE	AUDIT	audit_log.so	GPL

```

46 rows in set (0.00 sec)

mysql>
```

Tras esto, el plugin registra las acciones que se vayan realizando en la base de datos y las escribe en el fichero `/var/lib/mysql/audit.log`.

Probad a hacer algunas operaciones desde la consola de MySQL y consultad el fichero (es necesario utilizar “sudo” para ello).

En la imagen siguiente se muestran 2 entradas de ejemplo del log:

```
ubuntu@ip-172-31-18-109:~$ sudo tail -n 28 /var/lib/mysql/audit.log
<AUDIT_RECORD
  NAME="Query"
  RECORD="30_2021-02-24T08:47:23"
  TIMESTAMP="2021-02-24T09:08:55Z"
  COMMAND_CLASS="select"
  CONNECTION_ID="11"
  STATUS="0"
  SQLTEXT="select * from Bet"
  USER="admDBet[admDBet] @ localhost []"
  HOST="localhost"
  OS_USER=""
  IP=""
  DB=""
/>
<AUDIT_RECORD
  NAME="Quit"
  RECORD="31_2021-02-24T08:47:23"
  TIMESTAMP="2021-02-24T09:08:59Z"
  CONNECTION_ID="11"
  STATUS="0"
  USER="admDBet"
  PRIV_USER="admDBet"
  OS_LOGIN=""
  PROXY_USER=""
  HOST="localhost"
  IP=""
  DB="DBet"
/>
ubuntu@ip-172-31-18-109:~$ █
```

¿Qué operaciones se han realizado? ¿Con qué usuario?

Por defecto, el plugin de Percona utiliza XML como formato para registrar cada entrada del log. Sin embargo, existen otros 3 formatos que se pueden utilizar para generar ficheros de logs adaptados a nuestras necesidades:

- OLD: El formato XML que se utiliza por defecto, utilizando 1 único elemento con atributos.
- NEW: Un formato XML diferente, utiliza sub-elementos en lugar de atributos.
- JSON: Formato JSON<sup>1</sup>, útil para uso con navegadores y sistemas NoSQL.
- CSV: Formato CSV, como valores separados con comas.

El formato de log que el plugin utiliza se controla mediante la variable *audit\_log\_format*. Se puede cambiar su valor modificándolo en el fichero de configuración de MySQL (p.e. para utilizar CSV como formato):

```
GNU nano 4.8 /etc/mysql/mysql.conf.d/mysqld.cnf Modified
# binlog_ignore_db      = include_database_name
#
#
# * MySQL Validation Password plugin
#
validate_password.policy=LOW
#
# * Percona plugin configuration
#
audit_log_format=CSV
█

^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Paste Text ^T To Spell   ^_ Go To Line
```

Una vez hecho el cambio en el fichero es imprescindible reiniciar el servicio de MySQL. Después, entrar a la consola MySQL como root y verificar que la variable ha cambiado su valor:

```
mysql> show global variables like 'audit_log_format';
```

```
mysql> show global variables like 'audit_log_format';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| audit_log_format | CSV |
+-----+-----+
1 row in set (0.00 sec)

mysql> █
```

<sup>1</sup> Más información sobre JSON: <https://developer.mozilla.org/es/docs/Learn/JavaScript/Objects/JSON>



Una vez verificado que el formato del log ha cambiado, probad a hacer algunas operaciones desde la consola de MySQL y consultad el fichero de nuevo.

En la imagen siguiente se muestran 3 entradas de ejemplo del log en formato CSV:

```
ubuntu@ip-172-31-18-109:~$ sudo tail -n 3 /var/lib/mysql/audit.log
"Query","10111_2021-02-24T09:24:52","2021-02-24T09:31:45Z","update","13",0,"update Gambler set
gEmail=""betty@ehu.eus"" where gName=""Betty""","admDBet[admDBet] @ localhost []", "localhost"
,"", "", ""
"Query","10112_2021-02-24T09:24:52","2021-02-24T09:31:58Z","select","13",0,"select * from Gamb
ler where gName=""Betty""","admDBet[admDBet] @ localhost []", "localhost", "", "", ""
"Quit","10113_2021-02-24T09:24:52","2021-02-24T09:32:01Z","13",0,"admDBet","admDBet","", "", "lo
calhost", "", "DBet"
ubuntu@ip-172-31-18-109:~$
```

¿Qué operaciones se han realizado? ¿Con qué usuario?

Para finalizar, podéis acceder a la web del plugin Percona (indicada al comienzo de esta sección) para consultar todas las opciones de configuración posibles.

## 5 *Herramienta sysbench*

En este laboratorio vamos a utilizar una herramienta adicional, ajena a MySQL, para generar datos sintéticos que llenen una base de datos de pruebas. El objetivo es tener una BBDD con suficientes datos como para generar consultas lentas, sin tener que escribir miles de tuplas de forma manual.

Esta herramienta es sysbench, un software orientado a hacer pruebas de rendimiento en bases de datos. Es bastante conocido en la comunidad de usuarios de MySQL por su efectividad y simplicidad de uso. Podéis obtener más información en su repositorio oficial: <https://github.com/akopytov/sysbench>

Sysbench está disponible en los repositorios de Debian/Ubuntu y se puede instalar en nuestro sistema con el siguiente comando:

```
:~$ sudo apt-get install sysbench
```

```
ubuntu@ip-172-31-18-109:~$ sudo apt-get install sysbench
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  sysbench
0 upgraded, 1 newly installed, 0 to remove and 3 not upgraded.
Need to get 0 B/107 kB of archives
```

Una vez instalado, sysbench se utiliza desde línea de comandos y es capaz de ejecutar diferentes tipos de pruebas en una base de datos MySQL. En este apartado se describe cómo utilizar sysbench para llenar tablas con datos sintéticos.

Antes de ejecutar sysbench, es necesario tener lo siguiente en nuestra base de datos MySQL para que la prueba se ejecute correctamente:

- Crear una nueva base de datos.
- Crear un nuevo usuario con ubicación ‘%’ y permisos Create, Insert e Index a nivel de sistema.

Una vez hecho esto, se puede utilizar sysbench de la siguiente forma desde la terminal Linux (aunque aparezca en 4 líneas es un único comando):

```
:~$ sysbench --db-driver=mysql --mysql-db=<DATABASE> --mysql-user=<USER>
--mysql-password=<PASSWORD> --mysql-host=<IP-ADDRESS> --mysql-port=3306
--tables=<NUM-TABLES> --table-size=<NUM-ROWS>
/usr/share/sysbench/oltp_read_write.lua prepare
```

En este comando hay que configurar las siguientes variables:

- <DATABASE>: Nombre de la base de datos recién creada en MySQL.
- <USER>: Nombre del usuario recién creado.
- <PASSWORD>: Contraseña de acceso para el usuario indicado en <USER>.
- <IP-ADDRESS>: Dirección IP o nombre del servidor MySQL.
- <TABLE>: Número de tablas de prueba a crear.
- <NUM-ROWS>: Número de filas a insertar.

En la siguiente imagen se ve un ejemplo de uso de sysbench para crear 2 tablas y llenar cada de ellas con 10000 líneas de datos sintéticos en una base de datos llamada “Lab3DB” utilizando un usuario “Lab3user”:

Después de realizar esta operación, desde la consola de MySQL con el usuario “root” se puede verificar que sysbench ha creado 2 tablas nuevas llamadas “sbtest1” y “sbtest2”, y que contienen datos sintéticos.

```
mysql> use Lab3DB;
Database changed
mysql> show tables;
+-----+
| Tables_in_Lab3DB |
+-----+
| sbtest1          |
| sbtest2          |
+-----+
2 rows in set (0.00 sec)

mysql> describe sbtest1;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra           |
+-----+-----+-----+-----+-----+-----+
| id    | int    | NO   | PRI | NULL    | auto_increment |
| k     | int    | NO   | MUL | 0        |                 |
| c     | char(120) | NO   |     |         |                 |
| pad   | char(60)  | NO   |     |         |                 |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select id, k, substring(c, 1, 15), substring(pad, 1, 15) from sbtest1 limit 3;
+-----+-----+-----+-----+
| id | k   | substring(c, 1, 15) | substring(pad, 1, 15) |
+-----+-----+-----+-----+
| 1  | 4993 | 83868641912-287    | 67847967377-480      |
| 2  | 5020 | 38014276128-252    | 23183251411-362      |
| 3  | 5044 | 33973744704-805    | 38615512647-914      |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

## 6 Ejercicio

Una vez el *plugin Audit* de *Percona* está instalado y correctamente configurado, y una vez también los diarios *general* y *slow\_queries* están activados, vamos a ejecutar distintas operaciones en *MySQL* para después auditarlas.

Como resultado de las tareas, deberías encontrar unas entradas parecidas a las siguientes en el *log* generado por el *plugin* de *Percona* y en el *log* de consultas lentas:

```
ubuntu@ip-172-31-18-109:~$ sudo tail -n 3 /var/lib/mysql/audit.log
"Field List","10423_2021-02-24T09:24:52","2021-02-24T12:10:06Z","show_fields","23",0,"","root[root] @ localhost []","localhost","","",""
"Query","10424_2021-02-24T09:24:52","2021-02-24T12:10:08Z","select","23",0,"select min(sbtest1.k) from sbtest1 inner join sbtest2 on sbtest1.k = sbtest2.k","root[root] @ localhost []","localhost","","",""
"Quit","10425_2021-02-24T09:24:52","2021-02-24T12:10:16Z","23",0,"root","root","","","localhost","","Lab3DB"
ubuntu@ip-172-31-18-109:~$ sudo tail -n 10 /var/log/mysql/mysql-slow.log
# User@Host: root[root] @ localhost [] Id: 21
# Query_time: 4.959380 Lock_time: 0.019598 Rows_sent: 1 Rows_examined: 15615619
use Lab3DB;
SET timestamp=1614168567;
select count(*) from sbtest1 inner join sbtest2 on sbtest1.k = sbtest2.k;
# Time: 2021-02-24T12:10:12.482236Z
# User@Host: root[root] @ localhost [] Id: 23
# Query_time: 4.178087 Lock_time: 0.000174 Rows_sent: 1 Rows_examined: 15615619
SET timestamp=1614168608;
select min(sbtest1.k) from sbtest1 inner join sbtest2 on sbtest1.k = sbtest2.k;
```

En cada grupo de 2 personas, tendréis que repartiros las siguientes tareas.

## Tarea AB

- a) Utilizando el usuario *root* y en tu propio servidor de base de datos:
  1. Crea la base de datos *auditingAB*.
  2. Crea el usuario *auditorAB* para que sea utilizado por vuestro/a compañero/a de grupo y asígnale permisos Create, Insert, Index y Select a nivel de sistema.
  
- b) Utilizando el usuario *auditorAB* que vuestro/a compañero/a os ha creado:
  1. Utiliza sysbench (bien desde tu instancia GCP o desde tu equipo local si tienes Linux) para crear 2 tablas con 10.000 filas en la base de datos *auditingAB*.
  2. Ejecuta las siguientes consultas:

```
select count(*) from sbtest1      inner      join      sbtest2      on
sbtest1.k = sbtest2.k;
```

```
select min(sbtest1.k) from sbtest1      inner      join      sbtest2      on
sbtest1.k = sbtest2.k;
```
  3. Pídele a tu compañero/a que compruebe si las consultas anteriores se han registrado correctamente en cada uno de los *log* que hemos visto, incluyendo el generado por el plugin de Percona. Si las consultas del paso b.2 no se han registrado en el diario *slow\_queries*, pídele a tu compañero que elimine las tablas “sbtest”. Después, utiliza sysbench para crear nuevas tablas pero con un mayor número de filas. Repite este paso hasta que haya suficientes filas como para que las consultas del paso b.2 se registren en *slow\_queries*.

## Tarea CD

- c) Utilizando el usuario *root* y en tu propio servidor de base de datos:
1. Crea la base de datos *auditingCD*.
  2. Crea el usuario *auditorCD* para que sea utilizado por vuestro/a compañero/a de grupo y asígnale permisos Create, Insert, Index y Select a nivel de sistema.
- d) Utilizando el usuario *auditorCD* que vuestro/a compañero/a os ha creado:
1. Utiliza sysbench (bien desde tu instancia GCP o desde tu equipo local si tienes Linux) para crear 2 tablas con 10.000 filas en la base de datos *auditingCD*.
  2. Ejecuta las siguientes consultas:

```
select count(*) from sbtest2      inner      join      sbtest1      on  
sbtest2.k = sbtest1.k;
```

```
select min(sbtest2.k) from sbtest2      inner      join      sbtest1      on  
sbtest2.k = sbtest1.k;
```

3. Pídele a tu compañero/a que compruebe si las consultas anteriores se han registrado correctamente en cada uno de los *log* que hemos visto, incluyendo el generado por el plugin de Percona. Si las consultas del paso b.2 no se han registrado en el diario *slow\_queries*, pídele a tu compañero que elimine las tablas “sbtest”. Después, utiliza sysbench para crear nuevas tablas pero con un mayor número de filas. Repite este paso hasta que haya suficientes filas como para que las consultas del paso d.2 se registren en *slow\_queries*.

Para copiar los *logs* desde vuestro servidor en GCP a vuestra máquina, podéis utilizar el comando *scp* con el fichero *.pem* de la siguiente forma:

```
:~$ scp -i ficheroClaves.pem usuario@direccionIP:/rutaEnOrigen
rutaEnDestino
```

En la siguiente imagen se muestra un ejemplo para copiar un fichero desde la carpeta */home/ubuntu* en la instancia GCP a la carpeta */tmp* de la máquina local:

```
unai@MBP AWS-Educate % scp -i ABD-keypair.pem ubuntu@54.89.141.160:/home/ubuntu/mysql-slow.log /tmp
mysql-slow.log 100% 1630 18.1KB/s 00:00
unai@MBP AWS-Educate % █
```

Si alguno de los ficheros a copiar necesita permisos de “root” para su acceso, no se podrá copiar directamente usando *scp*. Antes, tenéis que realizar una copia del fichero y (a) cambiar el propietario de la copia utilizando *chown* o (b) cambiar los permisos de la copia utilizando *chmod*. Después podréis acceder a la copia del fichero con *scp*.

En la siguiente imagen se muestra un ejemplo de cómo crear una copia del fichero “mysql-slow.log” en el directorio */home/ubuntu* y se cambia el propietario al usuario “ubuntu” para que sea accesible por *scp*:

```
ubuntu@ip-172-31-18-109:~$ sudo cp /var/log/mysql/mysql-slow.log /home/ubuntu
ubuntu@ip-172-31-18-109:~$ ls -lh /home/ubuntu/mysql-slow.log
-rw-rw-rw- 1 root root 1.8K Feb 24 17:23 /home/ubuntu/mysql-slow.log
ubuntu@ip-172-31-18-109:~$ sudo chown ubuntu:ubuntu /home/ubuntu/mysql-slow.log
ubuntu@ip-172-31-18-109:~$ ls -lh /home/ubuntu/mysql-slow.log
-rw-rw-rw- 1 ubuntu ubuntu 1.8K Feb 24 17:23 /home/ubuntu/mysql-slow.log
```

Si alguno de los ficheros de log supera 2 MB (u otro tamaño que consideréis excesivo) podéis escribir las últimas líneas que creáis relevantes a un fichero nuevo utilizando el “>” de la terminal Linux de la siguiente forma:

```
:~$ tail -n <NUM-LINEAS> ficheroGrande > nuevoFichero
```

En la siguiente imagen se ve un ejemplo de cómo extraer las últimas 20 líneas del fichero *audit.log* (que ocupa 101 MB) en un nuevo fichero *extracto.log* (que ocupa 3 KB).



```

ubuntu@ip-172-31-18-109:~$ sudo ls -lh /var/lib/mysql/audit.log
-rw-r----- 1 mysql mysql 101M Feb 24 12:22 /var/lib/mysql/audit.log
ubuntu@ip-172-31-18-109:~$ sudo tail -n 20 /var/lib/mysql/audit.log > /home/ubuntu/extracto.log
ubuntu@ip-172-31-18-109:~$ ls -lh /home/ubuntu/extracto.log
-rw-rw-r-- 1 ubuntu ubuntu 3.0K Feb 24 12:39 /home/ubuntu/extracto.log
ubuntu@ip-172-31-18-109:~$ █

```

**IMPORTANTE:** Una vez que hayáis acabado de trabajar con el ejercicio o con los logs en general, es muy recomendable desactivar los *logs* de consultas generales y de consultas lentas que se activan al comienzo del laboratorio, y también el *plugin* de *Percona*.

Para desactivar los *logs*, basta con volver a comentar las líneas correspondientes en el fichero de configuración y reiniciar el servicio de MySQL. Tal y como se especifica en el propio fichero de configuración, dejar los *logs* activos sin necesidad afecta severamente al rendimiento de MySQL.

Para desactivar el *plugin* de *Percona*, es necesario utilizar el siguiente comando desde la consola de MySQL como usuario “root”:

```
mysql> UNINSTALL PLUGIN audit_log;
```

Y después verificar que su estado es “DELETED” en el listado de plugins de MySQL:

```
mysql> SHOW PLUGINS;
```

```

mysql> UNINSTALL PLUGIN audit_log;
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> SHOW PLUGINS;
+-----+-----+-----+-----+-----+
| Name          | Status | Type          | Library      | License |
+-----+-----+-----+-----+-----+
| binlog        | ACTIVE | STORAGE ENGINE | NULL         | GPL     |
+-----+-----+-----+-----+-----+
...
| audit_log     | DELETED | AUDIT          | audit_log.so | GPL     |
| auth_socket   | ACTIVE  | AUTHENTICATION | auth_socket.so | GPL     |
+-----+-----+-----+-----+-----+
46 rows in set (0.00 sec)

```

Si habéis cambiado el formato del *log* de *Percona*, el último paso es eliminar o comentar la línea que define la variable *audit\_log\_format* en el fichero de configuración de MySQL:

```
GNU nano 4.8 /etc/mysql/mysql.conf.d/mysqld.cnf Modified
#
# * Percona plugin configuration
#
#audit_log_format=CSV
█
^G Get Help      ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify      ^C Cur Pos
^X Exit          ^R Read File    ^\ Replace      ^U Paste Text   ^T To Spell     ^  Go To Line
```