

Actividad 1 (Parte B) - Instalación y configuración de un servidor MySQL e interacción con un servidor de aplicaciones

Índice

1. Introducción.....	1
2. Servidor MySQL.....	1
2.1 Instalación del servicio.....	2
2.2 Usuario de administración.....	2
2.3 Estructura de directorios y ficheros.....	3
2.3.1 Directivas del fichero mysqld.cnf.....	3
3. Stack LEMP.....	4
3.1 Topología física/lógica.....	4
3.2 Creación de usuarios para la aplicación.....	5
3.3 Instalación del módulo php-MySQL.....	7
4. Trabajo a realizar.....	8
5. Referencias.....	8

1. Introducción

A lo largo de la práctica analizaremos las principales arquitecturas que podemos implementar dentro del **stack LEMP** centrándonos en la instalación y configuración del SGBD y su acceso desde el servidor de aplicaciones **PHP-FPM**.

2. Servidor MySQL

MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual; Licencia pública general y comercial por la empresa Oracle Corporation. Actualmente, está considerada como la base de datos de código abierto más popular, sobre todo en entornos web.

MySQL fue inicialmente desarrollado por la compañía MySQL AB, pero su compra por parte de Oracle, propició la aparición de MariaDb, una bifurcación directa de MySQL que asegura la existencia de una versión de este producto con licencia GPL, ya que su creador estaba convencido de que el **único interés de Oracle en MySQL** era reducir la competencia que suponía para el mayor proveedor de bases de datos relacionales del mundo, que es Oracle.

2.1 Instalación del servicio

En distribuciones **linux** la instalación del SGBD MySQL es muy sencilla, se lleva a cabo mediante el gestor de paquetes:

```
$ sudo apt install mysql-server
```

Este comando no solo instalará el servidor MySQL, sino que también **instalará** los paquetes del **cliente de línea de comandos** de MySQL, que nos permitirán acceder y gestionar las diferentes bases de datos.

Para gestionar el servicio, utilizaremos el comando **systemctl** o su correspondiente **service**.

```
systemctl [start|stop|restart] mysql
```

```
service mysql [start|stop|restart]
```

2.2 Usuario de administración

En los sistemas operativos **ubuntu server**, la autenticación del usuario **root** del SGBD, se establece por defecto mediante `auth_socket` en lugar del tradicional `usuario/password`, lo que proporciona una mayor seguridad.

Este modo de autenticación se basa en crear y abrir un **socket unix** y asignar qué usuarios del sistema operativo host pueden leer y escribir en él (en nuestro caso será el **usuario root**). De esta forma, cuando queramos conectarnos, solo deberemos ejecutar el cliente MySQL como usuario root y tendremos acceso directo a la gestión del SGBD.

```
$ sudo mysql
```



Debemos tener en cuenta que la instalación por defecto de MySQL solo permite conectarnos al SGBD desde **localhost** y a través del usuario root utilizando `auth_socket`, por lo que **no podremos** establecer la conexión **desde PHP**.

En cualquier caso, **nunca deberíamos utilizar un usuario administrador** para establecer la conexión desde las diferentes aplicaciones que vayamos a desplegar, sino que cada aplicación debe tener un usuario específico con los permisos mínimos que necesite.

2.3 Estructura de directorios y ficheros

La instalación de **MySQL** establece el directorio `/etc/mysql` como base para el establecimiento de las **configuraciones globales del servicio**. El archivo principal lo encontramos en `/etc/mysql/mysql.cnf`. No obstante, con el fin de favorecer la legibilidad y mantenibilidad del servicio, se establece una estructura modular de archivos de configuración, que podemos encontrar a través del path `/etc/mysql/conf.d/`. Podemos destacar los siguientes archivos:

fichero/directorio	Descripción
<code>/etc/mysql/conf.d/mysqld.cnf</code>	<p>Contiene las configuraciones globales que queremos aplicar a nuestra instancia de mysql.</p> <p>Con el fin de mantener retro-compatibilidad hacia atrás, encontramos el mismo fichero con las configuraciones por defecto en el path <code>/etc/mysql/mysql.conf.d/mysql.cnf</code></p>
<code>/etc/mysql/conf.d/mysqldump.cnf</code>	Permite establecer configuraciones globales para la herramienta de backups de mysql (mysqldump)

2.3.1 Directivas del fichero `mysqld.cnf`

Si analizamos el fichero `/etc/mysql/mysql.conf.d/mysqld.cnf`, Podemos ver que se han establecido una serie de directivas por defecto como son:

Directiva	Descripción
user <code>user= mysql</code>	Establece el usuario del S.O. con el que se ejecutará el proceso de mysql
port <code>port= 3306</code>	Puerto por defecto en el que el servidor se mantendrá a la escucha
bind-address <code>bind-address = 127.0.0.1</code>	Establece la interfaz de red para la que atenderá las peticiones el servidor mysql.
log_error <code>log_error = /var/log/mysql/error.log</code>	Fichero de log por defecto del servicio
slow_query_log <code>slow_query_log = 1</code>	Activación / desactivación del registro de consultas pesadas que se están llevando a cabo en el servidor

Puedes consultar una lista completa de **directivas/variables** del sistema en el [siguiente](#) documento.

Actividad 1 (0.5 puntos)

- ¿Qué directiva nos permite especificar una **política de expiración de contraseñas**? Pon un ejemplo para que por defecto asigne 6 meses a todas las cuentas creadas.
- ¿Cuál es el valor por defecto de la directiva **bind-address**? ¿Desde que interfaces se mantendrá a la escucha el servidor si no se especifica la directiva?
- ¿Que función provee la directiva **collation_server**?

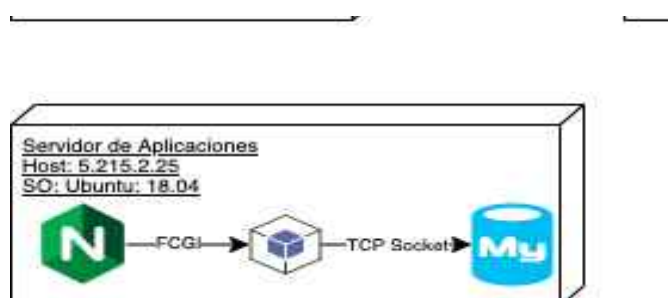
3. Stack LEMP

Como ya se ha comentado, el **stack LEMP** es una de las arquitecturas más comunes y utilizadas hoy en día. Se trata de la combinación de un sistema operativo **Linux**, un servidor web **Nginx**, la tecnología **PHP** y el servidor **MySql** como Sistema Gestor de Bases de Datos Relacionales.

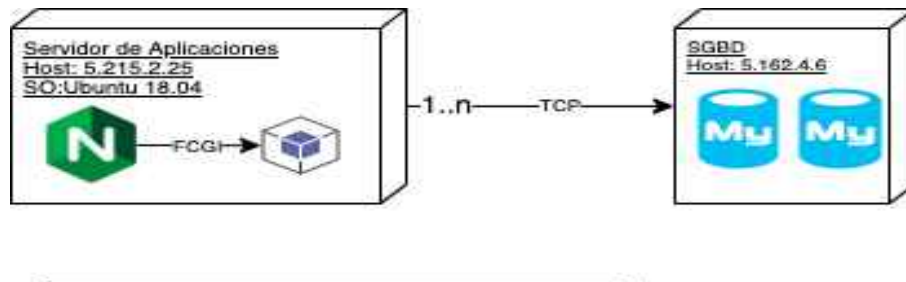
3.1 Topología física/lógica

Desde el punto de vista de la capa de datos, podemos encontrarnos diferentes topologías físicas y/o lógicas.

- **SGBD local:** El **Sistema gestor de bases de datos** y el **servidor de aplicaciones** se encuentran en la **misma máquina** por lo que no es necesaria una conexión de red remota en cada petición. Esta topología es válida cuando la aplicación que se va a desarrollar es pequeña y debe atender poco tráfico.



- **SGBD remoto:** A medida que la aplicación crece y/o los servicios que ofrece la empresa son más numerosos, es una buena idea separar el SGBD y el servidor de aplicaciones de forma que cada uno pueda operar con sus propios recursos distribuyendo la carga derivada de responder las peticiones de los visitantes.



Como se ha comentado anteriormente, la instalación por defecto solo permite las conexiones desde la misma máquina. Si queremos establecer un SGBD remoto deberemos habilitar la escucha por las interfaces remotas estableciendo la directiva `bind-address 0.0.0.0`

3.2 Creación de usuarios para la aplicación

Cada **aplicación** debe disponer de un **usuario específico** con los **mínimos privilegios** que necesita para su funcionamiento, de esta forma mantendremos diferentes dominios de seguridad y una vulnerabilidad descubierta en una aplicación no afectará al resto.

En primer lugar, debemos crear la base de datos que nuestra aplicación va a utilizar:

```
mysql> CREATE DATABASE myDataBase;
```

Una vez creada la base de datos crearemos el usuario que utilizará la aplicación para el establecimiento de las conexiones.

```
mysql> CREATE USER 'user'@'localhost' IDENTIFIED BY 'password';
```

- **user:** Usuarios que utilizaremos para conectarnos con el servidor

- **localhost:** IP desde la cual dicho usuario podrá conectarse al servidor. (Si utilizamos '%' podrá conectarse desde cualquier IP)
- **password:** Password que utilizaremos para autenticarnos. (Podemos hacer uso de la herramienta openssl para generar un password seguro)

```
$ openssl rand -base64 32
```



En mysql los usuarios se identifican tanto por el nombre como por el origen de las conexiones, por lo que un usuario user@localhost será diferente de user@'%'

Podemos consultar los usuarios de mysql mediante una consulta al diccionario de datos del sistema.

```
mysql> SELECT user,plugin,host FROM mysql.user;
```

Por último, **asignaremos** los **permisos** que necesita nuestra aplicación **al usuario** y los aplicaremos

```
mysql> GRANT ALL PRIVILEGES ON myDataBase.* TO 'user'@'localhost';
mysql> FLUSH PRIVILEGES;
mysql> exit();
```

Actividad 2 (0.5 puntos)

- Con el comando anterior hemos asignado al usuario "**user@localhost**", todos los privilegios sobre la base de datos `myDatabase`. ¿Cómo establecerías solo permisos DML para un usuario? Puedes consultar la **Tabla** de privilegios de la documentación oficial.

3.3 Instalación del módulo php-MySQL

Ya tenemos instalados todos los componentes del **stack** de nuestra aplicación. Sin

embargo, todavía no podremos establecer comunicación con el SGBD desde el servidor de aplicaciones.

Aunque la **comunicación con el SGBD** se va a llevar a cabo a partir del lenguaje **SQL**, a un nivel menor de abstracción, **nuestra aplicación** necesitará poder establecer una conexión y comunicarse con la **API nativa** que ha proporcionado el **desarrollador del SGBD**. Esta tarea se lleva a cabo a partir de los **controladores** o **drivers** que, como podemos ver en la [página oficial](#), se encuentran disponibles para la mayor parte de tecnologías existentes y que utilizaremos a partir de un módulo específico de nuestro servidor de aplicaciones o **intérprete PHP (PHP-FPM)**.

En sistemas operativos Unix y Like-Unix podemos hacer uso del gestor de paquetes para su instalación.

```
sudo apt install php-mysql
```

Como podemos observar, en la siguiente imagen, se instalará y cargará el driver nativo de mysql (**mysqlnd**) y las librerías de **php** nativas **pdo** y **mysqli** para gestionar la comunicación con el servidor de aplicaciones.

```
[sudo] password for alecogi:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  php7.2-mysql
Se instalarán los siguientes paquetes NUEVOS:
  php-mysql php7.2-mysql
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 49 no actualizados.
Se necesita descargar 119 kB de archivos.
Se utilizarán 461 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] S
Des:1 http://es.archive.ubuntu.com/ubuntu bionic-updates/main amd64 php7.2-mysql amd64 7.2.24-0ubuntu0.18.04.1 [117 kB]
Des:2 http://es.archive.ubuntu.com/ubuntu bionic/main amd64 php-mysql all 1:7.2+60ubuntu1 [2.004 B]
Descargados 119 kB en 1s (137 kB/s)
Seleccionando el paquete php7.2-mysql previamente no seleccionado.
(Leyendo la base de datos ... 104999 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../php7.2-mysql 7.2.24-0ubuntu0.18.04.1_amd64.deb ...
Desempaquetando php7.2-mysql (7.2.24-0ubuntu0.18.04.1) ...
Seleccionando el paquete php-mysql previamente no seleccionado.
Preparando para desempaquetar .../php-mysql_1%3a7.2+60ubuntu1_all.deb ...
Desempaquetando php-mysql (1:7.2+60ubuntu1) ...
Configurando php7.2-mysql (7.2.24-0ubuntu0.18.04.1) ...
Creating config file /etc/php/7.2/mods-available/mysqlnd.ini with new version
Creating config file /etc/php/7.2/mods-available/mysqli.ini with new version
Creating config file /etc/php/7.2/mods-available/pdo_mysql.ini with new version
```

Actividad 3 (0.5 puntos)

Busca información sobre las librerías nativas cargadas. ¿Qué funcionalidad nos proporciona cada una de ellas? ¿Qué función realizan los ficheros .ini que se han

creado?

4. Trabajo a realizar

Actividad 4 (1 punto)

- Instala un **servidor mysql** local al servidor de aplicaciones. Entra a la consola de mysql y haz una captura de pantalla.
- Crea un fichero **info.php** que contenga una referencia a la función `phpinfo()` y comprueba que los módulos **pdo**, **mysqli** y **mysqlnd** se encuentren cargados.
Pega una captura de pantalla que permita verificarlo
- Lleva a cabo el despliegue de la [siguiente aplicación web](#) (copiándola al directorio raíz del server block de nginx). Para ello, antes deberás importar la base de datos `crm_db` adjunta a esta práctica y crear un usuario (tendrá tu nombre) que solo sea accesible de forma local y con permisos DML sobre dicha base de datos. Tendrás también que cambiar los parámetros de la base de datos (busca el fichero de configuración dentro de la aplicación) y cambiar el usuario y contraseña por el que has creado nuevo.
- Para finalizar, habilita la conexión remota al servidor mysql y crea un usuario llamado `admin_db` que tenga todos los privilegios sobre la base de datos `crm_db` que permita la conexión desde cualquier host.
- Instala la aplicación [MySQL Workbench](#) según este [enlace](#) y establece una conexión con el usuario de administración creado. Aporta una captura de pantalla del estado del servidor desde MySQL Workbench.

5. Referencias

- Digital Ocean. Instalar un entorno LEMP.
["https://www.digitalocean.com/community/tutorials/how-to-install-linux-nginx-mysql-php-lemp-stack-ubuntu-18-04"](https://www.digitalocean.com/community/tutorials/how-to-install-linux-nginx-mysql-php-lemp-stack-ubuntu-18-04)

- dev.mysql.com. Documentación oficial Mysql.
["https://dev.mysql.com/doc/refman/5.7/en/"](https://dev.mysql.com/doc/refman/5.7/en/)
- php.net."Documentación oficial Mysql Native Driver de PHP".
<https://www.php.net/manual/en/book.mysqlnd.php>
- Digital Ocean.Como establecer una SBD remoto para optimizar el rendimiento mediante mysql. "<https://www.digitalocean.com/community/tutorials/how-to-set-up-a-remote-database-to-optimize-site-performance-with-mysql-on-ubuntu-18-04>