18. MARIADB

18.1.- Introducción

MariaDB es un proyecto OpenSource derivado de MySQL que se ha adoptado como gestor de BBDD relacionales por defecto en muchas distribuciones Linux como Fedora, RHEL, CentOS, Debian, OpenSuSe, …

Dada la mala gestión que hizo Oracle de MySQL, dejando de actualizar los repositorios públicos, utilizando componentes no libres, etc., el creador de MySQL, Michael Widenius, junto con la comunidad desarrolló MariaDB para ser una alternativa libre a MySQL. Actualmente el proyecto MariaDB es mantenido por la *MariaDB Foundation*.

El funcionamiento de MariaDB es igual a MySQL, mismos conectores, mismo archivo de configuración, mismas órdenes, interfaces, APIs, bibliotecas … pudiendo realizarse un reemplazo transparente de MySQL por MariaDB. No es necesario convertir las BBDD.

Una BBDD relacional es una forma de persistir datos de forma estructurada. Los datos se organizan en tablas (entidades) con filas (registros) y columnas (atributos) donde las tablas se relacionan entre sí.

18.2. Instalación

MariaDB en CentOS 7 se proporciona en dos grupos de paquetes que contienen los siguientes paquetes:

- ***mariabd***: *mariadb-server*, *mariadb-bench* y *mariadb-test* siendo el primer paquete obligatorio y los otros dos opcionales.
- ***mariadb-client***: *mariadb*, *mysql-python*, *mysql-connector-odbc*, *libdbi-dbd-mysql*, *mysql-connector-java* y *perl-DBD-MySQL* siendo el primer paquete obligatorio, los dos siguientes por defecto y los tres últimos optativos.

Con el comando:

yum groupinstall mariadb mariadb-client ó yum install @mariadb @mariadb-client instalaríamos ambos grupos de paquetes en el sistema que vaya a configurarse como servidor de MariaBD. En los sistemas configurados como clientes, sólo se instala el grupo de paquetes *mariadb-client.*

En el servidor, tenemos:

- */etc/my.cn*f: archivo de configuración principal.
- */etc/my.cnf.d/: archivos* de configuración extras con extensión *.cn*f.
- */var/log/mariadb/mariabd.log:* archivos de logs.

Nada más instalar el servidor, arrancamos el servicio *mariadb* y debemos securizar la instalación utilizando el comando:

```
mysql_secure_installation
```

Con este comando le proporcionamos password al usuario *root* de *MariaDB*, y además podemos eliminar que root sea accesible en remoto, eliminar la cuenta anónima para conectar y eliminar la BBDD de test si así lo deseamos.

18.3. Puesta en marcha del servidor MariaDB

Tras instalar los paquetes correspondientes en el servidor, el servicio ya se ha arrancado para hacer la securización y sólo falta activar el servicio *mariabd* con:

```
systemctl enable mariadb
```

Podemos verificar su funcionamiento con:

```
ss -tulpn | grep mysql
```

Veremos que por defecto está escuchando en el puerto *3306* de todos los interfaces del sistema.

18.4. Configurar el networking en el servidor MariaDB

Tenemos dos posibles formas de funcionamiento:

- *acceso sólo local*: cuando las BBDD sólo van a ser accedidas desde aplicaciones que están en el mismo sistema que el servidor de MariaDB. Tiene la desventaja de que el servidor comparte recursos con las aplicaciones y esto va a impactar en el rendimiento de las bases de datos. No es necesario abrir el *firewall*, con la ventaja de la seguridad que esto implica.
- *acceso remoto permitido*: las aplicaciones que utilizan las BBDD están en otros sistemas y hay que abrir el firewall para permitirlo. La ventaja es que el servidor no comparte recursos con las aplicaciones pero hay que abrir un puerto más en el firewall, lo cual puede implicar problemas de seguridad.

Para configurar el networking en MariaDB, dentro de su archivo de configuración principal */etc/my.cnf*, en la sección *[mysqld]* configuraremos las directivas:

- **bind-address=<valor>**: donde <*valor*> es:
  - hostname
  - dir IPv4 o IPv6
  - :: (disponible en todas las direcciones tanto IPv4 como IPv6)
  - vacío ó 0.0.0.0 (disponible en todas las direcciones IPv4)

Sólo se puede poner una opción de estas, este modo de funcionamiento de “o todo o nada”, hace que si queremos una configuración intermedia, p.e. que sólo escuchase por dos de las tres posibles direcciones del sistema, tendríamos que elegir la opción de todas y a nivel de *firewall* restringir que sólo puedan haber comunicaciones de MariaDB por esas dos direcciones usando una rich-rule.

- ***skip-networking*=1** para indicar que sólo vamos a dar servicio en local. De esta forma si ejecutamos *ss -tulpn | grep mysql* no veremos nada en la salida ya que la comunicación interna se hace vía *sockets* locales.
- **Port=<puerto>**. Puerto por el que escucha el servidor de MariaDB para conexiones TC/IP. No es necesario configurarlo si se va a utilizar el puerto por defecto, el 3306/tcp.

En el paquete *mariadb-server* existen ficheros de configuración de ejemplo que podemos encontrar ejecutando:

```
rpm -ql mariadb-server | grep my_
```

por ejemplo, el fichero *my-innodb-heavy-4G.cnf* muy completo y comentado.

→ Configuración en el firewall:

En el caso de usar el puerto por defecto para conexiones remotas, hay que abrir dicho puerto en el firewall ejecutando:

```
firewall-cmd --permanent --add-service=mariadb &&  firewall --reload.
```

Si se utiliza otro puerto distinto al 3306, p.e., 4406, el puerto se abre ejecutando:

```
firewall-cmd --permanent  --add-port 4406/tcp && firewall --reload
```

y asegurarnos de etiquetar ese puerto con la etiqueta SELinux *mysqld_port_t* de la forma:

```
semanage port -a -t mysqld_port_t -p tcp 4406
```

18.5. Operaciones básicas en MariaDB

Para poder realizar cualquier operación dentro de la línea de comandos de MariaDB, tenemos que utilizar el comando *mysql* que permite conectar con una BBDD local o remota, Dicho comando es proporcionado por el paquete *mariadb-client*. Es por eso que el grupo de paquetes *mariadb-client* también se suele instalar en el propio servidor.

El auto-completado para nombres de bases de datos, tablas y columnas dentro del *prompt* de MariaDB está desactivado por defecto. Para activarlo se puede hacer de tres formas:

- en el fichero de configuración */etc/my.cnf*: dentro de la sección *[mysql]* se debe añadir una línea con el contenido *auto-rehash* y reiniciar el servicio.

- en un fichero llamado **.***my.cnf* dentro del home del usuario con el que usemos el comando *mysql*.
- usando la opción --*auto-rehash* en el comando *mysql*.

Nos conectaremos con el servidor de MariaDB:

- si el servidor está en local: *myslq -u <usuario> -p*

- si el servidor está en remoto: *myslq -u <usuario> -h <hostname> -p*

donde:

<*usuario*> es el usuario de MariaDB, no de sistema, que ha sido dado de alta en la configuración de MariaDB con una password.

<*hostname*> es el hostname o IP del sistema que hace de servidor MariaDB.

Se nos pedirá la password del usuario.

Existe la posibilidad de introducir la password directamente con el comando *mysql*, justo después de la opción *-p*, sin espacios. Esto es útil para utilizarlo en scripts.

Una vez hemos hecho *login*, nos aparece el *prompt* de la línea de comandos de MariaDB de la forma: *MariaDB[none]* donde se introducen los comandos propios de MariaDB.

Los comandos utilizados en esta línea de comandos terminan en **;** y se pueden poner en mayúsculas o minúsculas, es indistinto. Se suelen poner en mayúsculas para, a golpe de vista, diferenciarlos de los nombres de las bases de datos, tablas y atributos que suelen ir en minúsculas. Pero esto sólo una recomendación. Los nombres que sí hay que respetar tal y como estén, son los nombres de bases de datos, tablas y atributos.

→ *Comandos de MariaDB*:

- **SHOW DATABASES;**

  Ver todas las bases de datos existentes en el servidor. Tras la instalación aparecen 4: *information_schema*, *performance_schema*, *myslq* y *test*. La única que se puede borrar es la de *test*, el resto las utiliza el servidor para mantener sus propios datos; son de funcionamiento interno. Por ejemplo, en la base de datos *mysql*, en la tabla *user*, se almacenan los usuarios de MariaDB.

- **USE <base_datos>;**

  Conecta con la base de datos dada en <*base_datos*>.

- **SHOW TABLES;**

  Lista las tablas de la base de datos en uso.

- **DESCRIBE <tabla>;**

Muesta los atributos de la tabla dada en *<tabla>*.

- **CREATE DATABASE <nombre**>;

  Crea la base de datos dada en *<nombre>*.

- **DROP DATABASE <nombre>;**

  Elimina la base de datos dada en *<nombre>*, con todas sus tablas.

→ *Comandos SQL para interaccionar con las bases de datos*:

- **INSERT INTO <tabla> (<at1>,<at2>,...,<atN>) VALUES ('<v1>','<v2>', …,'<vN>');**

  Inserta un registro en la base de datos en uso, dentro de la tabla *<tabla>*, de forma que el atributo *<at1>* tiene valor *<v1>*, el atributo *<at2>* tiene valor *<v2>*, ...y el atributo *<atN>* tiene valor *<vN>*..

- **DELETE FROM <tabla> WHERE <clave>='<valor>';**

  Borra un registro de la tabla de la base de datos en uso accediendo a él por su clave principal. Si no se pone *WHERE* se borrarían todos los registros de la tabla, dejándola vacía.

- **UPDATE <tabla> SET <at1>='<v1>',<at2>='<v2>',...,<atN>='<vN>') WHERE <clave>='<valor>';**

  Actualiza un registro de la tabla en la base de datos en uso accediendo a él por su clave principal. Si no se pone WHERE se actualizarían todos los registros de la tabla.

- **SELECT (<at1>,<at2>,...,<atN>) FROM <tabla>;**

  Muestra el valor de los atributos dados para todos los regitros de la tabla *<tabla>*. Si se añade *WHERE* con una condición, mostrará los atributos datos de los registros que cumplan la condición.

NOTA: Para ejecutar comandos MariaDB sin necesidad de entrar en su *prompt*, podemos utilizar la opción *-e '<sentencia_sql1>;<sentencia_sql2>;...'* con el comando *mysql*.

18.6. Gestionar usuarios de MariaDB

Hay que gestionar la autenticación añadiendo el usuario a la tabla *user* de la base de datos *mysql* y gestionar la autorización en base a unos permisos para realizar acciones.

- Autenticación:

  El usuario está formado de dos partes que se separan con @. Un *username* y el *host* desde el que se podrá conectar. El host puede ser:

  - localhost, para accesos locales.
  - una IP en concreto.

- una subred, indicando el octeto de host con el carácter %, p.e., *10.11.1.%*.
- '%' para indicar que se puede conectar desde cualquier sistema remoto.

De esta forma se puede crear un usuario con el mismo *username* pero con distinto *host* y asignarle distintos privilegios según desde donde se conecte.

La password se proporciona al crear el usuario y se guarda encriptada como un campo más de la tabla *user*.

Operaciones con usuarios:

- *Añadir un usuario*: el usuario con el que me he conectado a MariaDB debe tener el privilegio CREATE USER/INSERT en la base de datos *mysql*. Comando:

  CREATE USER <username>@<host> IDENTIFIED BY '<password>';

- *Borrar un usuario*:

  DROP USER <username>@<host>;.

  En ambas sentencias, *CREATE USER* y *DROP USER* si no se pone la parte de *@<host>* se presupone *@'%'*.

- *Ver usuarios existentes*:

  SELECT User,Host FROM mysql.user;

- Autorización:

  Se otorgan privilegios, permisos que un usuario podrá tener en MariaDB. Para ello el usuario con el que me conecto al servidor debe tener el privilegio GRANT OPTION. Los privilegios están organizados en varios tipos:

  - *globales*: para administración del propio server MariaDB: *CREATE USER, SHOW DATABASES, GRANT OPTION,…*
  - *de bases de datos*: crear/trabajar con bases de datos en el servidor, alto nivel: *CREATE, ALTER, DROP.*
  - *de tabla*: crear tablas y manipular datos de la base de datos: *CREATE, INSERT, UPDATE, DROP, SELECT*
  - *de columna*: sólo en una columna de la tabla.
  - *otros*: ALL PRIVILEGES

Con la sentencia *GRANT* daremos estos privilegios a los usuarios con la sintáxis:

GRANT <privilegio> ON <basedatos>.<tabla> TO <username>@<host>;

Se pueden quitar privilegios usando la sentencia *REVOKE* usada de forma parecida a *GRANT:*

REVOKE <privilegio> ON <basedatos>.<tabla> FROM <username>@<host>;

Cuidado al hacer *REVOKE ALL PRIVILEGES ON *.* TO <username>@<host>;* ya que también quitamos el acceso al usuario a MariaDB.

Tras añadir/quitar privilegios se deben recargar los privilegios usando la sentencia:

```
FLUSH PRIVILEGES;
```

Para consultar los privilegios de un usuario:

```
SHOW GRANTS FOR <username>@<host>;
```

si no se añade la parte de @<*host*>, se presupone *@'%'*.

18.7. Backup de las bases de datos

Las bases de datos suelen contener información crítica y es importante el obtener y recuperar las copias de seguridad. Deben realizarse copias con regularidad.

Hay dos formas de crear un *backup* en MariaDB:

- *lógica*: exportando la información y los registros a archivos planos.
  - Ventajas: muy portable, se puede recuperar en otro gestor de base de datos, se hacen cuando el servidor está *online*.
  - Desventajas: es más lento que el *backup* físico ya que tiene que acceder a toda la base de datos y pasarlo a texto. No incluyen información de configuración ni de *logs*.
- física (raw): guardando el contenido del directorio donde se almacena la información en el sistema.
  - Ventajas: son más rápidos que los backups lógicos, incluyen archivos de configuración y *logs* y la salida es más compacta.
  - Desventajas: son menos portables, el sistema debe tener unas características HW y SW muy parecidas, han de hacerse cuando el servidor está *offline* o bloquear las tablas para que no se puedan hacer modificaciones.

18.7.1. Backup lógico

Para crear un *backup* lógico se utiliza el comando *mysqldump* con un usuario de MariaDB que tenga privilegios de *SELECT* con el formato:

```
mysqldump -u <usuario> -p <opciones> <base_datos> > archivo.dump
```

La opciones son:

--*add-drop-tables*: antes de cada sentencia CREATE hace un DROP.

--*no-data*: para hacer *backup* sólo de la estructura de la base de datos, no de los datos.

--*lock-all-tables*: para bloquear las tablas mientras se produce el backup.

--all-databases: para hacer *backup* de todas las bases de datos existentes en el servidor. En este caso no se proporciona <*base_datos*>. Esto incluye las bases de datos internas de MariaDB.

--add-drop-databases: antes de todas las sentencias *CREATE TABLE* hace un *DROP*.

Para recuperar un *backup* lógico nos conectaremos al servidor de MariaDB como *root* y simplemente redirigiremos el archivo a esta conexion:

```
mysql -u root -p <base_datos> < <archivo>.dump
```

NOTA: los archivos de *backup* lógico también se les suele poner la extensión *.sql* ya que contienen las sentencias sql necesarias para recrear toda la base de datos y los datos que contienen.

18.7.2. Backup físico

Para crear un *backup* físico simplemente hay que sacar una copia del directorio donde están los datos de MariaDB que por defecto es */var/lib/mysql*. Se puede consultar el directorio usado con:

```
mysqladmin variables | grep datadir
```

Para sacar esta copia se puede utilizar distintas herramientas como pueden ser c*p, rsync, ibbackkup, mysqlhotcopy* o incluso si el directorio usado está sobre un volumen lógico de LVM, se puede sacar un snapshot de dicho volumen lógico y usar este snapshot como *backup*.

Lo que si es importante antes de realizar el *backup* físico, es bloquear las tablas para que no se introduzcan nuevos datos mientras se produce la copia conectando al servidor MariaDB con:

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

y hacer

```
FLUSH TABLES WITH READ LOCK;
```

y no cerrar esta sesión hasta haber completado la copia. Tras la realización de la copia, desbloqueamos las tablas con *UNLOCK TABLES;* y ya podemos cerrar la sesión abierta.

Para recuperar un *backup* físico, dependerá de la herramienta utilizada para realizarlo.

18.8. Gestionar la contraseña de root de MariaDB

La contraseña de administración de MariaDB hay que asignarla nada más instalar con el comando *mysql_secure_installation*. Si por algún motivo no se hizo, se puede ejecutar:

```
mysqladmin -u root password '<contraseña_deseada>'
```

Para cambiar la contraseña de root:

```
mysqladmin -u root -p<contraseña_actual> password '<contraseña_nueva>'
```

Si no se conoce la contraseña que tiene actualmente y se quiere recuperar, hay que seguir el procedimiento:

1. Parar el servicio mariadb:

```
systemctl stop mariadb
```

2. Arrancar en modo *safe*:

   ```
   mysqld_safe --skip-grant-tables –skip-networking&
   ```

   Con la opción *--skip-grant-tables* se iniciará sin emplear el sistema de privilegios, por lo que tendrás acceso ilimitado a todas las bases de datos y la opción *--skip-networking* hará que se dejen de escuchar conexiones TCP/IP externas, lo que evitará problemas externos durante el cambio.

3. Conectar como root sin contraseña:

   ```
   mysql -u root
   ```

4. Ir a la BBDD *mysql*:

   ```
   USE mysql;
   ```

5. Modificar el registro de *root* en la tabla *user*:

   ```
   UPDATE user SET password=PASSWORD('nueva') WHERE user='root';
   flush privileges;
   exit
   ```

   donde *nueva* es la nueva contraseña de *root* de MariaDB.

6. Parar el modo *safe*:

   Con el comando *jobs*, veo el número de job y ejecuto *kill %n* , donde *n* es el número de job otra opción es:

   ```
   mysqladmin -p shutdown
   ```

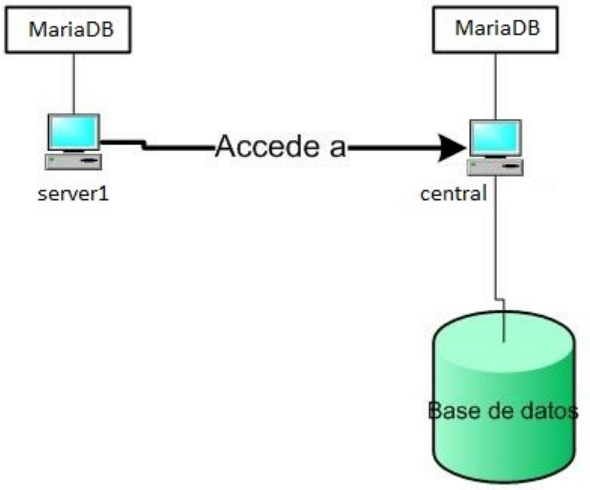
7. Arranco MariaDB:

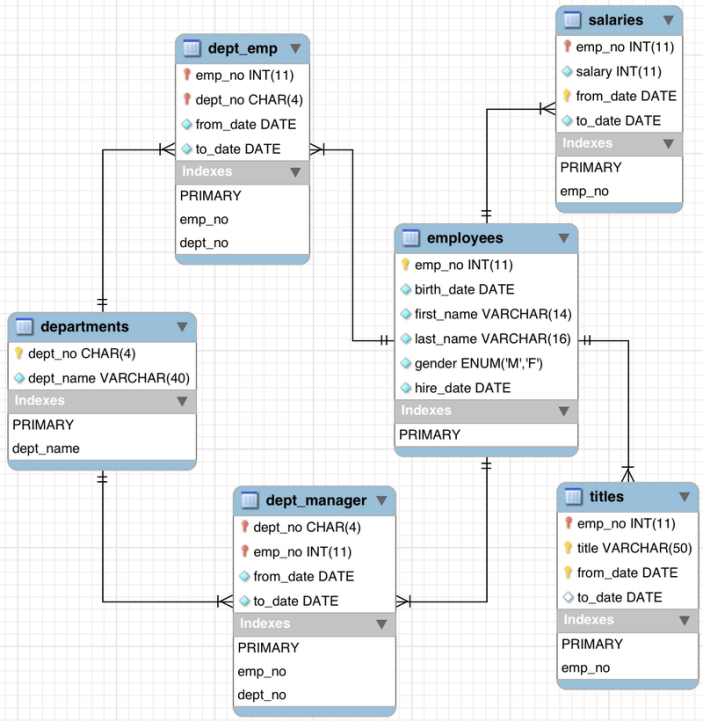
   ```
   systemctl start mariadb
   ```

18.9. Caso práctico

Vamos a configurar un servidor *MariaDB* en **central.** El usuario administrador de MariaDB tiene la contraseña de administración *mariadb* y sólo puede acceder en local. Sólo estará escuchando por la IP *10.11.1.254* de **central**.

En este servidor, habrá una base de datos de ejemplo llamada *employees* donde **server1** accederá como cliente para realizar consultas en dicha base de datos.





Esquema de la base de datos employees

Hay un *backup* lógico de la base de datos *employees* en los archivos que están en la máquina física, accesibles en http://192.168.1.122/test_db-master.zip.

Se van a configurar usuarios de MariaDB, uno por departamento: *desarrollo*, *redes*, *seguridad* y *sistemas* con el nombre del departamento y que se puedan conectar desde cualquier sistema de la subred 10.11.1.0/24. La contraseña para los usuarios serán las tres primeras letras del nombre del departamento seguido de la cadena *pass* (*despass*, *redpass*, *segpass y sispass*). Los permisos sobre la base de datos *employees* son:

- *desarrollo*: permisos CRUD sobre todas las tablas.
- *redes* y *seguridad*: consulta sobre la tabla *employees* y *departments.*
- *sistemas*: tendrá permisos totales sobre la base de datos de *employees*.

RESOLUCIÓN

- Instalamos en **central** los paquetes necesarios para que sea el servidor de MariaDB, arrancamos y activamos el servicio *mariadb*:

```
[root@central ~]# yum groupinstall mariadb mariadb-client
[root@central ~]# systemctl start mariadb && systemctl enable mariadb
```

- Securizamos la instalación con la contraseña de administración *mariadb*, eliminando el acceso remoto a root, el acceso anónimo y la base de datos de *test*:

[root@central ~]# mysql_secure_installation

- Verificamos que MariaDB está escuchando en todos los interfaces:

[root@central ~]# ss -tulpn | grep mysql

tcp LISTEN 0 50 *:3306 *:* users:(("mysqld",pid=1254,fd=14))

- Configuramos el *networking* y el auto-completado y reiniciamos el servicio *mariadb*:

[root@central ~]# vi /etc/my.cnf

```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
bind-address=10.11.1.254
# Disabling symbolic-links is recommended to prevent assorted security risks
symbolic-links=0
# Settings user and group are ignored when systemd is used.
# If you need to run mysqld under a different user or group,
# customize your systemd unit file for mariadb according to the
# instructions in http://fedoraproject.org/wiki/Systemd
[mysqld_safe]
log-error=/var/log/mariadb/mariadb.log
pid-file=/var/run/mariadb/mariadb.pid

[mysql]
auto-rehash
#
# include all files from the config directory
#
!includedir /etc/my.cnf.d
```

[root@central ~]# systemctl restart mariadb

- Verificamos que ahora sólo escucha por la IP:

[root@central ~]# ss -tulpn | grep mysql

tcp LISTEN 0 50 10.11.1.254:3306 *:* users:(("mysqld",pid=4073,fd=14))

- Abrimos el firewall para el servicio mysql para permitir que se conecte **server1**:

[root@central ~]# firewall-cmd --permanent --add-service=mysql --zone=internal

[root@central ~]# firewall-cmd --reload

- Obtenemos de la máquina física, ubicada en IP *192.168.122.1*, el *backup* lógico:

[root@central ~]# wget -P /tmp/ http://192.168.122.1/test_db-master.zip

[root@central ~]# yum install -y wget unzip

[root@central ~]# unzip /tmp/test_db-master.zip

- Creamos la base de datos *employees* vacía donde recuperar el backup y lo volcamos:

[root@central ~]# mysql -u root -pmariadb -e 'CREATE DATABASE employees'

```
[root@central ~]#  mysql -u root -pmariadb employees < /tmp/test_db-master/employees.sql
INFO
CREATING DATABASE STRUCTURE
INFO
storage engine: InnoDB
INFO
LOADING departments
INFO
LOADING employees
INFO
LOADING dept_emp
INFO
LOADING dept_manager
INFO
LOADING titles
INFO
LOADING salaries
data_load_time_diff
NULL
```

- Verificamos que hay contenido en la base de datos *employees*:

```
[root@central ~]#  mysql -u root -pmariadb -e 'USE employees;SHOW TABLES'
+------------------------------+
| Tables_in_employees          |
+------------------------------+
| current_dept_emp             |
| departments                  |
| dept_emp                     |
| dept_emp_latest_date         |
| dept_manager                 |
| employees                    |
| salaries                     |
| titles                       |
+------------------------------+
```

- Creamos el usuario de MariaDB para el departamento de desarrollo con permisos CRUD sobre todas las tablas de la base de datos *employees*:

```
[root@central ~]# mysql -u root -pmariadb -e “CREATE USER desarrollo@'10.11.1.%'
    IDENTIFIED BY 'despass';GRANT SELECT, UPDATE, INSERT, DELETE ON
    employees.* TO desarrollo@'10.11.1.%';”
```

-Creamos el usuario de MariaDB para el departamento de redes con permisos sólo de consulta sobre las tablas *employees* y *departments*:

```
[root@central ~]# mysql -u root -pmariadb -e “CREATE USER redes@'10.11.1.%'
    IDENTIFIED BY 'redpass';GRANT SELECT ON employees.employees TO
    redes@'10.11.1.%';GRANT SELECT ON employees.departments TO
    redes@'10.11.1.%';”
```

-Creamos el usuario de MariaDB para el departamento de seguridad con permisos sólo de consulta sobre las tablas *employees* y *departments*:

[root@central ~]# mysql -u root -pmariadb -e “CREATE USER seguridad@'10.11.1.%' IDENTIFIED BY 'segpass';GRANT SELECT ON employees.employees TO seguridad@'10.11.1.%';GRANT SELECT ON employees.departments TO seguridad@'10.11.1.%'; ”

- Y por último el usuario del departamento de sistemas con permisos totales:

[root@central ~]# mysql -u root -pmariadb -e "CREATE USER sistemas@'10.11.1%' IDENTIFIED BY 'sispass';GRANT ALL PRIVILEGES ON employees.* TO sistemas@'10.11.1.%';"

- Verificamos los usuarios creados:

[root@central ~]# mysql -u root -pmariadb -e "select User,Host from mysql.user"

```
+-----------------+------------------+
| User            | Host             |
+-----------------+------------------+
| root            | 127.0.0.1        |
| root            | ::1              |
| root            | localhost        |
| desarrollo      | 10.11.1.%        |
| redes           | 10.11.1.%        |
| seguridad       | 10.11.1.%        |
| sistemas        | 10.11.1.%        |
+-----------------+------------------+
```

- Verificamos los privilegios:

[root@central ~]# mysql -u root -pmariadb -e "SHOW GRANTS FOR desarrollo@'10.11.1.%'; "

```
+-----------------------------------------------------------------------------------------------------------------+
| Grants for desarrollo@10.11.1.%                                                                                 |
+-----------------------------------------------------------------------------------------------------------------+
| GRANT USAGE ON *.* TO 'desarrollo'@'10.11.1.%' IDENTIFIED BY PASSWORD '*E3A0B486B3A4A0F6998505BC230BEDC08BABC11E' |
| GRANT SELECT, INSERT, UPDATE, DELETE ON `employees`.* TO 'desarrollo'@'10.11.1.%'                               |
+-----------------------------------------------------------------------------------------------------------------+
```

[root@central ~]# mysql -u root -pmariadb -e "SHOW GRANTS FOR redes@'10.11.1.%';"

```
+-----------------------------------------------------------------------------------------------------------------+
| Grants for redes@10.11.1.%                                                                                      |
+-----------------------------------------------------------------------------------------------------------------+
| GRANT USAGE ON *.* TO 'redes'@'10.11.1.%' IDENTIFIED BY PASSWORD '*13A0D5D9C0EBF7E6FBD683725F8D13DDEC05A717'    |
| GRANT SELECT ON `employees`.`employees` TO 'redes'@'10.11.1.%'                                                  |
| GRANT SELECT ON `employees`.`departments` TO 'redes'@'10.11.1.%'                                                |
+-----------------------------------------------------------------------------------------------------------------+
```

[root@central ~]# mysql -u root -pmariadb -e "SHOW GRANTS FOR seguridad@'10.11.1.%'; "

```
+-----------------------------------------------------------------------------------------------------------------+
| Grants for seguridad@10.11.1.%                                                                                  |
+-----------------------------------------------------------------------------------------------------------------+
```

```
| GRANT USAGE ON *.* TO 'seguridad'@'10.11.1.%' IDENTIFIED BY PASSWORD
'*15CD24727239A8582A59262AAE74C6B7D34C15C2'                                    |
| GRANT SELECT ON `employees`.`departments` TO 'seguridad'@'10.11.1.%'          |
| GRANT SELECT ON `employees`.`employees` TO 'seguridad'@'10.11.1.%'            |
+-------------------------------------------------------------------------------------------------------------------+
```

[root@central ~]# mysql -u root -pmariadb -e "SHOW GRANTS FOR sistemas@'10.11.1.%'; "

```
+-------------------------------------------------------------------------------------------------------------------+
| Grants for sistemas@10.11.1.%                                                                                     |
+-------------------------------------------------------------------------------------------------------------------+
| GRANT USAGE ON *.* TO 'sistemas'@'10.11.1.%' IDENTIFIED BY PASSWORD
'*F72E0A0DCA1A64E8A9915CFC929B1F8C071105A0'                                     |
| GRANT ALL PRIVILEGES ON `employees`.* TO 'sistemas'@'10.11.1.%'                |
+-------------------------------------------------------------------------------------------------------------------+
```

- Instalamos en **server1** la parte cliente de MariaDB:

[root@server1 ~]# yum groupinstall -y mariadb-client

- Nos conectamos desde **server1** para comprobar si puedo hacer un *select* con el usuario *desarrollo*:

[root@server1 ~]# mysql -u desarrollo -pdespass -h central -e 'USE employees;

SELECT COUNT(first_name) FROM employees WHERE first_name LIKE "Patricia";'

```
+--------------------------+
| COUNT(first_name) |
+--------------------------+
|            215           |
+--------------------------+
```

- Vemos si puedo hacer *select* a otras tablas:

[root@server1 ~]# mysql -u desarrollo -pdespass -h central -e 'USE employees;SELECT COUNT(emp_no) FROM salaries'

```
+------------------------+
| COUNT(emp_no) |
+------------------------+
|      2844047          |
+---------------------- --+
```

- Comprobamos que con el usuario *redes* no puedo acceder a la tabla *salaries*:

[root@server1 ~]# mysql -u redes -predpass -h central -e 'USE employees;SELECT COUNT(emp_no) FROM salaries'

ERROR 1142 (42000) at line 1: SELECT command denied to user 'redes'@'server1' for table 'salaries'