

Administración de MySQL

Noviembre 2005

Jordi Llonch
jordi@laigu.net



MySQL®



<http://creativecommons.org/licenses/by-sa/2.0/>

Índice



- Conceptos de MySQL
- Instalación
- Configuración
- Tipos de campos
- Introducción a la administración
- Monitor (*aplicación cliente*)
- Permisos
- Tuning
- Monitoraje (mytop)
- Chequear y reparar tablas
- Backup
- Replicar
- phpMyAdmin

Conceptos de MySQL



- MySQL es un motor de base de datos SQL con las siguientes características:
 - Muy rápido.
 - Multi-thread.
 - Multiusuario.
- Disponemos de dos tipos de licencia:
 - Open Source/Software libre (GNU General Public License).
 - Standard commercial license.

Conceptos de MySQL: Motores y tipos de tablas



- ISAM
 - Es el motor original. En *deprecated* a partir de la versión 4.1.
- MyISAM
 - Reemplazo de ISAM.
 - Tipo de tabla por defecto.
 - No soporta transacciones.
- HEAP
 - Tablas en memoria.
 - Actualmente se conoce como MEMORY.
 - No soporta transacciones.
- MERGE
 - Permite que una colección de tablas MyISAM sean tratadas como una única tabla.
 - No soporta transacciones.

Conceptos de MySQL: Motores y tipos de tablas



- BDB (Berkeley DB)
 - Soporta transacciones.
 - No funciona en todas las plataformas.
- InnoDB
 - Soporta transacciones (ACID compliant).
 - Bloqueo a nivel de fila (incrementa la capacidad de concurrencia y rendimiento).
 - Se puede mezclar InnoDB con otros tipos de tablas en la misma bbdd.
- NDB Cluster
 - Soporta tablas que se encuentran particionadas en diferentes ordenadores.

Instalación



- Supondremos un sistema Fedora Core 3:
 - Instalaremos los siguientes paquetes:
 - `rpm -i mysql-3.23.58-13.i386.rpm`
 -
 - Si queremos habilitar el soporte de PHP para MySQL:
 - `rpm -i php-mysql-4.3.9-3.i386.rpm`
 -
- También podríamos instalar los binarios que podemos descargar en la web oficial (<http://dev.mysql.com/downloads/>).

Configuración



- Este documento explica algunas directivas de configuración de MySQL.
- Archivo de configuración:
 - *my.cnf* (plataforma Linux)
 - *my.ini* (plataforma Windows)

Configuración



- *[client]*
 - *password* = *my_password*
 - *contraseña general de acceso para los clientes (es aconsejable usar el sistema de autenticación de MySQL)*
 - *port* = *3306*
 - *puerto dónde se conectarán los clientes (acceso via TCP/IP)*
 - *socket* = */var/run/mysqld/mysqld.sock*
 - *acceso mediante sockets Unix (localhost)*

Configuración



- *[safe_mysqld]*
 - *err-log* = */var/log/mysql/mysql.err*
 - *archivo dónde se registrarán los errores del servidor*
- *[mysqld]*
 - *user* = *mysql*
 - *usuario bajo el cual se ejecuta el servidor*
 - *pid-file* = */var/run/mysqld/mysqld.pid*
 - *Fichero en el que el servidor guarda el ID del proceso demonio de escucha (daemon).*
 - *socket* = */var/run/mysql/mysqld.sock*
 - *Fichero en el que el servidor crea el socket*
 - *port* = *3306*
 - *Puerto desde donde se aceptaran conexiones*

Configuración



- *[mysqld]*
 - *log* = */var/log/mysql.log*
 - *Archivo donde se registra el log*
 - *basedir* = */usr*
 - *Directorio donde se encuentra MySQL.*
 - *datadir* = */var/lib/mysql*
 - *Directorio donde están los datos*
 - *tmpdir* = */tmp*
 - *Directorio temporal (para crear peq. tablas temporales)*
 - *language* = */usr/share/mysql/english*
 - *Archivo de mensajes*
 - *skip-locking*
 - *No usar sistema de bloqueo (para hacer chequeos se tendrá que parar el servidor)*
 - *skip-networking*
 - *No escucha conexiones TCP/IP.*

Configuración



- *[mysql.server]*
 - *key_buffer=16M*
 - *tamaño de buffer de los índices*
 - *max_allowed_packet=1M*
 - *tamaño máximo de un paquete. Se tendrá que incrementar si se quiere utilizar columnas BLOB grandes.*
 - *thread_stack=128K*
 - *tamaño de cada hilo*
 - *max_connections=650*
 - *número máximo de conexiones simultáneas*

Configuración



- *[mysql.server]*
 - *wait_timeout=9600*
 - *segundos que el servidor esperará actividad en conexiones no interactivas antes de cerrar*
 - *connect_timeout=10*
 - *segundos que el servidor esperará un paquete de conexión antes de responder con Bad handshake.*
 - *max_connect_errors=10*
 - *si hay más de este número de conexiones interrumpidas por el servidor, el servidor es bloqueado*
 - *log-slow-queries = /var/log/mysql/mysql-slow.log*
 - *registra las consultas lentas*
 - *skip-innodb*
 - *no utiliza bases de datos tipo innodb*

Tipos de campos



- Tipos numéricos:
 - TINYINT[(M)] [UNSIGNED] [ZEROFILL]
 - Entero muy pequeño (-128 a 127 con signo / 0 a 255 sin signo).
 - BIT/BOOL/BOOLEAN
 - Sinónimo de TINYINT(1).
 - Añadido en MySQL 4.1.0.
 - Cero es falso. Valores no zero son considerados verdaderos.
 - SMALLINT[(M)] [UNSIGNED] [ZEROFILL]
 - Entero pequeño (-32768 a 32767 con signo / 0 a 65535 sin signo).
 - MEDIUMINT[(M)] [UNSIGNED] [ZEROFILL]
 - Entero medio (-8388608 a 8388607 con signo / 0 a 16777215 sin signo).
 - INT[(M)] [UNSIGNED] [ZEROFILL]
 - Entero (-2147483648 a 2147483647 con signo / 0 to 4294967295 sin signo).
 - INTEGER[(M)] [UNSIGNED] [ZEROFILL]

Sinónimo de INT.

Tipos de campos



- Tipos numéricos:
 - BIGINT[(M)] [UNSIGNED] [ZEROFILL]
 - Entero grande (-9223372036854775808 a 9223372036854775807 con signo / 0 to 18446744073709551615 sin signo)
 - FLOAT[(M,D)] [UNSIGNED] [ZEROFILL]
 - Número coma flotante pequeño.
 - -3.402823466E+38 a -1.175494351E-38, 0, y 1.175494351E-38 a 3.402823466E+38
 - DOUBLE[(M,D)] [UNSIGNED] [ZEROFILL]
 - Número coma flotante normal.
 - -1.7976931348623157E+308 a -2.2250738585072014E-308, 0, y 2.2250738585072014E-308 a 1.7976931348623157E+308
 - DOUBLE PRECISION[(M,D)] [UNSIGNED] [ZEROFILL]
 - REAL[(M,D)] [UNSIGNED] [ZEROFILL]
 - Sinónimos de DOUBLE.

Tipos de campos



- Tipos numéricos:
 - DECIMAL[(M[,D])] [UNSIGNED] [ZEROFILL]
 - Número con decimal fijo.
 - Se comporta com un CHAR.
 - M es el total de dígitos y D es el número de decimales.
 - DEC[(M[,D])] [UNSIGNED] [ZEROFILL]
 - NUMERIC[(M[,D])] [UNSIGNED] [ZEROFILL]
 - FIXED[(M[,D])] [UNSIGNED] [ZEROFILL]
 - Sinónimos de DECIMAL

Tipos de campos



- Tipos fecha y hora:
 - DATE
 - Fecha. Rango de '1000-01-01' a '9999-12-31'.
 - Formato: 'YYYY-MM-DD'.
 - DATETIME
 - Combinación de fecha y hora. Rango de '1000-01-01 00:00:00' a '9999-12-31 23:59:59'.
 - Formato: 'YYYY-MM-DD HH:MM:SS'.
 - TIMESTAMP[(M)]
 - Rango de '1970-01-01 00:00:00' a aproximadamente el año 2037.
 - TIME
 - Tiempo. Rango de '-838:59:59' a '838:59:59'.
 - Formato: 'HH:MM:SS'.
 - YEAR[(2|4)]
 - Año en 2 dígitos o 4 dígitos. Rango de 1901 to 2155, and 0000 para 4 dígitos y de 70 a 69 (de 1970 a 2069) para 2 dígitos.

Tipos de campos



- Tipos string:
 - [NATIONAL] CHAR(M) [BINARY | ASCII | UNICODE]
 - Cadena de caracteres de longitud fija.
 - M de 0 a 255.
 - CHAR
 - Sinónimo de CHAR(1).
 - [NATIONAL] VARCHAR(M) [BINARY]
 - Cadena de caracteres de longitud variable.
 - M de 0 a 255.
 - TINYBLOB/TINYTEXT
 - Una columna BLOB o TEXT de como máximo 255 caracteres.
 - BLOB/TEXT
 - Una columna BLOB o TEXT de como máximo 65,535 caracteres.
 - MEDIUMBLOB/MEDIUMTEXT
 - Una columna BLOB o TEXT de como máximo 16,777,215 caracteres.

Tipos de campos



- Tipos string:
 - `LONGBLOB/LONGTEXT`
 - Una columna BLOB o TEXT de como máximo 4,294,967,295 caracteres.
 - `ENUM('value1','value2',...)`
 - Enumeración.
 - Cadena de caracteres que sólo puede tener un valor dado en la lista.
 - `SET('value1','value2',...)`
 - Lista que puede tener valor cero o más valores. Deben estar en la lista y como máximo pueden haber 64 miembros.

Introducción a la administración



- El sistema de base de datos MySQL está formado por diversos componentes:
 - El servidor MySQL:
 - *mysqld*: es el demonio principal, realiza todas las manipulaciones de bdd y tablas.
 - *mysqld_safe*: usado para iniciar el servidor, monitorizar y reiniciar en caso de parada (antes de MySQL 4, *safe_mysqld*).
 - *mysqld_multi*: si se ejecutan múltiples servidores en el mismo host.
 - El cliente MySQL y utilidades:
 - *mysql*: cliente interactivo que permite enviar instrucciones SQL
 - *mysqladmin*: programa de administración
 - *mysqlcheck*, *myisamchk*: utilidades de análisis, optimización y reparación de tablas.
 - *mysqldump*, *mysqlhotcopy*: herramientas de backup o copia de base de datos a otros servidores.
 -
 - (Aunque muchas operaciones se podrán realizar con la utilidad *mysqladmin* o alguna otra aplicación de administración es preferible conocer el lenguaje SQL)

Introducción a la administración



- El directorio de datos de MySQL:
 - Es el directorio dónde se almacenan las bases de datos y archivos.
 - Es importante conocer:
 - Dónde están situados los archivos que representan las bbdd.
 - Archivos de log.
 - Evitar que el sistema de archivos esté demasiado lleno.

Introducción a la administración



- Administración General:
 - Iniciar y parar el servidor.
 - Se debe conocer cómo iniciar y parar el servidor manualmente.
 - Qué hacer si no se inicia correctamente o se para inesperadamente.
 - Mantenimiento de las cuentas de usuario.
 - Se debe entender la diferencia entre las cuentas UNIX y las MySQL.
 - Cómo crear y mantener cuentas de usuario.
 - De que forma se deben conectar los nuevos usuarios al servidor.
 - Cómo resetear un password perdido.
 - Mantenimiento del archivo de log.
 - Se deben entender los tipos de archivos de log que genera MySQL.
 - Rotar y expirar los archivos de log para prevenir que se llene el sistema de archivos.
 - Backup y copia de base de datos.
 - La copia de seguridad es una tarea crucial.
 - Se debe saber como restaurar una bbdd.

Introducción a la administración



- Administración General:
 - Replicación de base de datos.
 - Hacer un backup o copia de una bdd requiere tiempo. Una opción es usar replicación.
 - Configuración y tuning del servidor.
 - Se deben conocer que parámetros de la configuración permiten ajustar el rendimiento del servidor.
 - La localización (juego de caracteres) y la zona horaria también debe estar bien configurada.
 - Servidores múltiples.
 - En ocasiones puede resultar útil configurar servidores simultáneos en el mismo host.
 - Actualizaciones.
 - Las nuevas *releases* aparecen frecuentemente.
 - Se debe estar al día de los bugs, agujeros de seguridad y nuevas características.

Monitor (*aplicación cliente*)



- Aplicación cliente que nos conecta al servidor.
- Se le suele proporcionar: usuario, password y servidor.
 - ```
shell> mysql -h host -u user -p
```
  - ```
Enter password: *****
```
 - ```
Welcome to the MySQL monitor. Commands end with ; or \g.
```
  - ```
Your MySQL connection id is 25338 to server version: 4.0.14-log
```
 - ```
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```
  - ```
mysql>
```
 - Ahora podemos introducir comandos SQL en el cliente.
-
- Para desconectar:
 - ```
mysql> QUIT
```
  - Bye
  - En Linux/Unix... podemos usar CTRL+D

# Permisos

---



- Securizando una nueva instalación.
- Recuperar root perdido.
- Gestión de cuentas de usuario.





## Permisos: Securizando una nueva instalación

---

- La instalación de MySQL crea dos bases de datos:
  - `mysql`: contiene las tablas de permisos.
  - `test`: para uso de pruebas.
- En la primera instalación los permisos en la bdd *mysql* estan de forma que permiten la conexión sin password.
- Si se actualiza la versión de MySQL no es necesario la resecurización.



# Permisos: Securizando una nueva instalación

- Veamos un ejemplo de cómo securizar el servidor cobra.snake.net:
  - Inicialmente existen dos tipos de cuentas:
    - Cuenta que tiene root como usuario. Cuenta de superusuario para administración. Dispone de todos los privilegios.
    - Cuenta que no tiene asociado ningún usuario. Cuenta "anonymous", útil para test porque permite la conexión sin tener una cuenta explícita. Tiene muy pocos privilegios.
  - Ninguna de estas cuentas tiene asociado password. Se espera que el administrador los proporcione.
  - Tabla user de la bbdd mysql:

| Host            | User | Password | Superuser Privileges |
|-----------------|------|----------|----------------------|
| localhost       | root |          | All                  |
| cobra.snake.net | root |          | All                  |
| localhost       |      |          | None                 |
| cobra.snake.net |      |          | None                 |



# Permisos: Securizando una nueva instalación

- Estas entradas permiten las siguientes conexiones:
  - `% mysql -h localhost -u root`
  - `% mysql -h cobra.snake.net -u root`
  -
- Entradas anónimas:
  - `% mysql -h localhost`
  - `% mysql -h cobra.snake.net`
  -
- Los usuarios anónimos no tienen privilegios de administrador. Pero en la tabla *db*, se especifica que pueden usar la bbdd test o cualquier bbdd que tenga un nombre que comience por test.
- 
- *Nota:* El símbolo % en host permite la conexión desde cualquier remoto.



# Permisos: Securizando una nueva instalación

- Establecer passwords para las cuentas iniciales:
  - Según el método usado se tendrá que pedir a MySQL que vuelva a leer las tablas de permisos.
    - 1) mysqladmin
      - » % mysqladmin -h localhost -u root password "rootpass"
      - » % mysqladmin -h cobra.snake.net -u root password "rootpass"
    - 2) SET PASSWORD
      - » % mysql -u root
      - » mysql> SET PASSWORD FOR 'root'@'localhost' = PASSWORD('rootpass');
      - » mysql> SET PASSWORD FOR 'root'@'cobra.snake.net' = PASSWORD('rootpass');
    - 3) Modificar directamente las tablas
      - » % mysql -u root
      - » mysql> USE mysql;
      - » mysql> UPDATE user SET Password=PASSWORD('rootpass') WHERE User='root';
      - » mysql> FLUSH PRIVILEGES;



# Permisos: Securizando una nueva instalación

- Después de establecer el password deberemos conectarnos usando:
  - `% mysql -p -u root`
  - Enter password: rootpass
  - `mysql>`
  -
- Si queremos eliminar el acceso anónimo:
  - `% mysql -p -u root`
  - Enter password: rootpass
  - `mysql> USE mysql;`
  - `mysql> DELETE FROM user WHERE User = '';`
  - `mysql> DELETE FROM db WHERE User = '';`
  - `mysql> FLUSH PRIVILEGES;`
- 
- Otra forma de permitir el acceso anónimo sería con password:
  - `mysql> SET PASSWORD FOR ''@'localhost' = PASSWORD('anonpass');`



# Permisos: Recuperar root perdido

- Pasos a seguir:
  - 1) Parar el servidor MySQL
    - `/etc/init.d/mysql stop`
    - Si no funciona usar: `kill -TERM PID` (primero matar `mysqld_safe` y luego el resto)
    - Si tampoco funciona usar: `kill -9 PID`
  - 2) Reiniciar el servidor con la opción `skip-grant-tables`
    - Modificar `my.cnf` y añadir `skip-grant-tables` (sección `[mysqld]`)
    - Reiniciar.
  - 3) Entrar en el monitor y ejecutar `FLUSH PRIVILEGES`
  - 4) Modificar los privilegios según nos convenga con `GRANT` o `SET PASSWORD`.
  - 5) No olvidar volver a dejar `my.cnf` como estaba.



## Permisos: Gestión de cuentas de usuario

---

- El administrador establecerá los usuarios que se pueden conectar a MySQL y con qué privilegios.
- Disponemos de dos comandos para ello:
  - GRANT: crea un cuenta MySQL y especifica privilegios.
  - REVOKE: borra privilegios de una cuenta existente.
- También existe la posibilidad de:
  - Modificar directamente las tablas de permisos.
  - Usar los scripts *mysqlaccess* y *mysql\_setpermission*.



# Permisos: Gestión de cuentas de usuario

- GRANT y REVOKE afectan cuatro tablas:
  - *user*: usuarios que pueden conectarse al servidor y sus privilegios globales.
  - *db*: privilegios a nivel de bbdd.
  - *tables\_priv*: privilegios a nivel de tabla.
  - *columns\_priv*: privilegios a nivell de columna.
  - Existe una quina tabla (*host*) pero no está afectada por los comandos *GRANT* o *REVOKE*.





# Permisos: Gestión de cuentas de usuario

- Crear un nuevo usuario y establecer privilegios:
  - GRANT privileges (columns)
  - ON what
  - TO account IDENTIFIED BY 'password'
  - WITH grant
  - REQUIRE encryption requirements;
  - *privileges*: privilegios que se le asignaran a la cuenta (SELECT, SHUTDOWN, ALL PRIVILEGES...)
  - *columns*: (opcional) SELECT, INSERT y UPDATE.
  - *what*: nivel al que seran aplicados los privilegios. A nivel de bbdd, tabla o columna (ON db\_name.\*, ON \*.\*)
  - *account*: usuario y host ('user\_name'@'host\_name'). Se pueden usar *wildcards*: user\_name@'144.155.166.%'
  - *password*: (opcional) contraseña de la cuenta.
  - *WITH*: (opcional) permite dar permisos a otros usuarios.
  - *REQUIRE*: (opcional, MySQL 4.0.0) establece que la conexión debe estar securizada por SSL.



## Permisos: Gestión de cuentas de usuario

- La forma mas sencilla de configurar una cuenta es hacerse las siguientes preguntas:
  - ¿Quién se puede conectar y desde dónde?
  - ¿Qué tipo de acceso debería tener la cuenta?
  - ¿Se requiere que la conexión sea segura?
  - ¿Debería poder administrar privilegios?
  - ¿Deberían limitarse el consumo de recursos?



# Permisos: Gestión de cuentas de usuario

- ¿Quién se puede conectar y desde dónde?
  - Limitar a un sólo host:
    - `GRANT ALL ON sampdb.* TO 'boris'@'localhost' IDENTIFIED BY 'ruby';`
    - `GRANT ALL ON sampdb.* TO 'fred'@'ares.mars.net' IDENTIFIED BY 'quartz';`
  - Permitir conexiones desde todas partes:
    - `GRANT ALL ON sampdb.* TO 'max'@'%' IDENTIFIED BY 'diamond';`
  - Permitir conexiones desde ciertas redes:
    - `GRANT ALL ON sampdb.* TO 'mary'@'%.snake.net' IDENTIFIED BY 'topaz';`
    - `GRANT ALL ON sampdb.* TO 'joe'@'192.168.128.3' IDENTIFIED BY 'water';`
    - `GRANT ALL ON sampdb.* TO 'ardis'@'192.168.128.%' IDENTIFIED BY 'snow';`
    - `GRANT ALL ON sampdb.* TO 'rex'@'192.168.128.0/255.255.128.0' IDENTIFIED BY 'lice';`



# Permisos: Gestión de cuentas de usuario

- ¿Qué tipo de acceso debería tener la cuenta?
  - Hay diversos tipos de privilegios:
    - CREATE TEMPORARY TABLES: Crear tablas temporales.
    - EXECUTE: Ejecutar procedimientos almacenados (MySQL 5.0.0)
    - FILE: Leer y escribir archivos en el servidor.
    - GRANT OPTION: Dar los privilegios a otras cuentas.
    - LOCK TABLES: Bloquear tablas.
    - PROCESS: Ver información sobre la ejecución de threads en el servidor.
    - RELOAD: Recargar la tabla de privilegios o flush de logs y cachés.
    - REPLICATION CLIENT: Preguntar sobre localización de servidor master y slave.
    - REPLICATION SLAVE: Actuar como una réplica de servidor slave.
    - SHOW DATABASES: Mostrar bbdd.
    - SHUTDOWN: Parar el servidor.
    - SUPER: Matar threads y otras operaciones de supervisor.
    - ALTER: Modificar tablas e índices.



# Permisos: Gestión de cuentas de usuario

- ¿Qué tipo de acceso debería tener la cuenta?
  - Hay diversos tipos de privilegios:
    - CREATE: Crear bbdd y tablas.
    - DELETE: Borrar registros de una tabla.
    - DROP: Eliminar bbdd y tablas.
    - INDEX: Crear o eliminar índices.
    - INSERT: Insertar nuevos registros en tablas.
    - REFERENCES: (reservado para futuras versiones).
    - SELECT: Consultar registros en tablas.
    - UPDATE: Modificar registros de tablas.
    - ALL: Todas las operaciones (excepto GRANT); sinónimo de ALL PRIVILEGES.
    - USAGE: Privilegio especial "no privilegios".



# Permisos: Gestión de cuentas de usuario

- ¿Qué tipo de acceso debería tener la cuenta?
  - Se pueden establecer privilegios a diferentes niveles.
  - Se usa la cláusula ON:
    - ON \*.\* privilegios globales, todas las bdd, todas las tablas.
    - ON \* privilegios globales si no se ha seleccionado bdd sinó privilegios en la bdd actual.
    - ON db\_name.\* privilegios a nivel de bdd; todas las tablas de la bdd.
    - ON db\_name.tbl\_name privilegios a nivel de tabla; todas las columnas en la tabla.
    - ON tbl\_name privilegios a nivel de tabla; todas las columnas en la tabla por defecto.



## Permisos: Gestión de cuentas de usuario

- ¿Se requiere que la conexión sea segura?
  - A partir de MySQL 4, se pueden securizar las conexiones usando SSL.
  - Tiene un coste en tiempo de proceso.
  - Se puede obligar el acceso seguro:
    - `GRANT ALL ON sampdb.* TO 'eladio'@'%.snake.net' IDENTIFIED BY 'flint' REQUIRE SSL;`
  - Incluso requerir un certificado X509:
    - `GRANT ALL ON sampdb.* TO 'eladio'@'%.snake.net' IDENTIFIED BY 'flint' REQUIRE X509;`



## Permisos: Gestión de cuentas de usuario

- ¿Debería poder administrar privilegios?
  - Podríamos permitir al dueño de la bdd controlar los accesos cediendo todos los privilegios en la bdd y especificando WITH GRANT OPTION.
    - GRANT ALL ON sales.\*
    - TO 'alicia'@'%.big-corp.com' IDENTIFIED BY 'shale'
    - WITH GRANT OPTION;





# Permisos: Gestión de cuentas de usuario

- ¿Deberían limitarse el consumo de recursos?
  - A partir de MySQL 4.0.2, se pueden poner límites:
  - Número de veces por hora que se pueden hacer conexiones.
  - Consultas o actualizaciones por hora.
    - `GRANT ALL ON sampdb.* TO 'spike'@'localhost' IDENTIFIED BY 'pyrite'`
    - `WITH`
    - `MAX_CONNECTIONS_PER_HOUR 10`
    - `MAX_QUERIES_PER_HOUR 200`
    - `MAX_UPDATES_PER_HOUR 50;`
  - `FLUSH USER_RESOURCES` o `FLUSH PRIVILEGES` resetan los valores contados a 0.



# Permisos: Gestión de cuentas de usuario

- Revocar privilegios y eliminar usuarios
  - REVOKE privileges (columns) ON what FROM account;
  - what: debe coincidir con el usuario y el host
  - Se pueden dar permisos y luego revocar algunos:
    - GRANT ALL ON sampdb.\* TO 'boris'@'localhost' IDENTIFIED BY 'ruby';
    - REVOKE DELETE,UPDATE ON sampdb.\* FROM 'boris'@'localhost';
  - REVOKE no elimina totalmente una cuenta, para eliminarla:
    - % mysql -u root
    - mysql> USE mysql;
    - mysql> DELETE FROM user
    - -> WHERE User = 'mary' and Host = '%.snake.net';
    - mysql> FLUSH PRIVILEGES;



# Permisos: Gestión de cuentas de usuario

- Cambiar passwords o resetear password perdidos
  - 1) Método 1:
    - `mysql> UPDATE user SET Password=PASSWORD('silicon')`
    - `-> WHERE User='boris' AND Host='localhost';`
    - `mysql> FLUSH PRIVILEGES;`
  - 2) Método 2:
    - `mysql> SET PASSWORD FOR 'boris'@'localhost' =  
PASSWORD('silicon');`
  - 3) Método 3:
    - `mysql> GRANT USAGE ON *.* TO 'boris'@'localhost'  
IDENTIFIED BY 'silicon';`

# Tuning

---



- Optimizaciones de hardware y software.
- Parámetros de configuración.
- Cómo usar el comando OPTIMIZE TABLE.
- Cómo usar el comando EXPLAIN.
- Cómo usar el comando FLUSH para limpiar tablas, caches y archivos de log.
- Cómo usar el comando SHOW para obtener información sobre base de datos, tablas e índices.
- Cómo usar el comando SHOW para obtener el estado del sistema.

# Tuning

---



- Optimizaciones de hardware y software:
  - CPU
    - Simple. Lo más rápido posible.
    - Hyper-threading, duales, quads...
  - Memoria
    - Nunca hay bastante.
  - Disco duro
    - Tiempo de acceso!
    - Suele ser el “embudo”.
  - Sistema operativo
    - Escoger un sistema operativo que no malgaste recursos en ventanas con bonitos colores...
    - Existen diferencias de rendimiento entre diferentes SO. Se pueden encontrar *benchmarks* que los comparan.

# Tuning



- Optimizaciones de hardware y software:
  - Uso de la función *benchmark()*
    - Ejecuta una expresión N veces.
    - Ejecutada en diversas ocasiones del día nos da una idea del rendimiento del servidor.
    - 
    - `mysql> select benchmark(10000000, 10+10);`
    - `+-----+`
    - `| benchmark(10000000, 10+10) |`
    - `+-----+`
    - `| 0 |`
    - `+-----+`
    - `1 row in set (0.44 sec)`

# Tuning

---



- Parámetros de configuración:
  - MySQL AB nos proporciona información para afinar los parámetros de nuestro servidor.
  - Con la instalación disponemos de unos archivos de configuración de muestra:
    - ***my-small.cnf***: Para sistemas con menos de 64Mb de RAM, dónde MySQL se usa de forma ocasional.
    - ***my-medium.cnf***: Para sistemas con menos de 64Mb de RAM, dónde MySQL es la actividad principal, o en sistemas con hasta 128Mb de RAM, donde MySQL uno de los procesos en el servidor. (configuración común de un servidor web con tráfico moderado).
    - ***my-large.cnf***: Para sistemas de 128Mb hasta 512Mb de RAM, dónde MySQL es el proceso principal.
    - ***my-huge.cnf***: Para sistemas de 1Gb a 2 Gb de RAM, dónde MySQL es el proceso principal.

# Tuning



- Parámetros de configuración:
  - Comparativa archivos de configuración de guía:

## my-small.cnf

```
...
[mysqld]
port = 3306
socket = /var/run/...
skip-locking
key_buffer = 16K
max_allowed_packet = 1M
table_cache = 4
sort_buffer_size = 64K
read_buffer_size = 256K
read_rnd_buffer_size = 256K
net_buffer_length = 2K
thread_stack = 64K
```

```
[myisamchk]
key_buffer = 8M
sort_buffer_size = 8M
...
```

## my-medium.cnf

```
...
[mysqld]
port = 3306
socket = /var/run/...
skip-locking
key_buffer = 16M
max_allowed_packet = 1M
table_cache = 64
sort_buffer_size = 512K
read_buffer_size = 256K
read_rnd_buffer_size = 512K
net_buffer_length = 8K
```

```
myisam_sort_buffer_size = 8M
```

```
[myisamchk]
key_buffer = 20M
sort_buffer_size = 20M
read_buffer = 2M
write_buffer = 2M
...
```

## my-large.cnf

```
...
[mysqld]
port = 3306
socket = /var/run/...
skip-locking
key_buffer = 256M
max_allowed_packet = 1M
table_cache = 256
sort_buffer_size = 1M
read_buffer_size = 1M
read_rnd_buffer_size = 4M
```

```
myisam_sort_buffer_size = 64M
thread_cache = 8
query_cache_size= 16M
Try number of CPU's*2
thread_concurrency = 8
```

```
[myisamchk]
key_buffer = 128M
sort_buffer_size = 128M
read_buffer = 2M
write_buffer = 2M
...
```

## my-huge.cnf

```
...
[mysqld]
port = 3306
socket = /var/run/...
skip-locking
key_buffer = 384M
max_allowed_packet = 1M
table_cache = 512
sort_buffer_size = 2M
read_buffer_size = 2M
read_rnd_buffer_size = 8M
```

```
myisam_sort_buffer_size = 64M
thread_cache = 8
query_cache_size = 32M
Try number of CPU's*2
thread_concurrency = 8
```

```
[myisamchk]
key_buffer = 256M
sort_buffer_size = 256M
read_buffer = 2M
write_buffer = 2M
...
```



# Tuning



- Parámetros de configuración (*key*):
  - Existen dos parámetros principales que afectan al sistema:
    - **key\_buffer\_size**
      - Tamaño de buffer usado en los índices
      - Verificar el rendimiento actual de los buffers:
        - » Usando el comando *SHOW STATUS*:
        - » `mysql> show status;`
        - » Se muestra una larga lista de valores. Buscar los siguientes:
        - »
        - » | Key\_read\_requests | 602843 |
        - » | Key\_reads | 151 |
        - » | Key\_write\_requests | 1773 |
        - » | Key\_writes | 805 |
        - »
        - » Si se divide *key\_read* por *key\_reads\_requests*, el resultado debería ser menor que 0.01.
        - » También si se divide *key\_write* por *key\_writes\_requests*, el resultado debería ser menor que 1.
        - » En este ejemplo los resultados son 0.000250479809834401 y 0.454032712915962, respectivamente. Aceptables.

# Tuning

---



- Parámetros de configuración (key):
- **table\_cache**
  - Número de tablas abiertas para todos los threads.
  - Usando SHOW STATUS, mirar open\_tables. Si el número es grande se deberá incrementar table\_cache.

# Tuning



- Cómo usar el comando OPTIMIZE TABLE:
  - Se usa para optimizar la estructura de la tabla (no tiene nada que ver con el buen diseño de tablas).
  - Reclama el espacio no usado después del borrado o modificación de estructura.
  - Se debe tener en cuenta que durante el proceso se bloquea la tabla.

```
mysql> optimize table grocery_inventory;
+-----+-----+-----+-----+
| Table | Op | Msg_type | Msg_text |
+-----+-----+-----+-----+
| testDB.grocery_inventory | optimize | status | OK |
+-----+-----+-----+-----+
1 row in set (0.08 sec)
mysql> optimize table grocery_inventory;
+-----+-----+-----+-----+
| Table | Op | Msg_type | Msg_text |
+-----+-----+-----+-----+
| testDB.grocery_inventory | optimize | status | Table is already up to date |
+-----+-----+-----+-----+
1 row in set (0.03 sec)
```

# Tuning

---



- Cómo usar el comando EXPLAIN:
  - Optimización de consultas.
  - Uso correcto de los índices.
  - Especialmente útil para analizar consultas complejas que implican *JOINS*.
  - La salida del comando contiene las siguientes columnas:
    - *table*: nombre de tabla
    - *type*: tipo de union (join)
    - *possible\_keys*: indica que índices podrían usarse para encontrar los registros. Si es *NULL*, no se podrán usar índices.
    - *key*: el índice usado en esta consulta. Si es *NULL* no se usa índice.
    - *key\_len*: la longitud del índice usado.
    - *ref*: columnas usadas con el índice.
    - *rows*: número de registros que se deben examinar para ejecutar la consulta.
    - *extra*: información adicional (*using index, where...*).

# Tuning



- Cómo usar el comando EXPLAIN:

- Ejemplo:

```

•
• mysql> explain select * from grocery_inventory;
• +-----+-----+-----+-----+-----+-----+
 +-----+
• | table | type | possible_keys | key | key_len| ref |
 rows| Extra|
• +-----+-----+-----+-----+-----+-----+
 +-----+
• | grocery_inventory | ALL | NULL | NULL| NULL| NULL|
 6| |
• +-----+-----+-----+-----+-----+
 +-----+
• 1 row in set (0.00 sec)

```

- 
- Aquí no se pueden hacer demasiadas optimizaciones excepto añadir una clausula WHERE con la clave primaria.

# Tuning



- Cómo usar el comando FLUSH para limpiar tablas, caches y archivos de log:
  - FLUSH TABLES:
    - Cierra todas las tablas abiertas o en uso.
    - Esencialmente da al servidor un respiro de un milisegundo antes de volver al trabajo. Cuando las cachés están vacías, MySQL puede utilizar mejor la memoria disponible.
      - `mysql> flush tables;`
      - `Query OK, 0 rows affected (0.21 sec)`
  - FLUSH HOSTS:
    - Actua sobre la caché del servidor.
    - Si no se puede conectar al servidor, lo habitual es que se hayan superado el número máximo de conexiones para un servidor particular.
      - `mysql> flush hosts;`
      - `Query OK, 0 rows affected (0.00 sec)`
  - FLUSH LOGS:
    - Cierra y reabre todos los archivos de log creando un nuevo archivo.
      - `mysql> flush logs;`
      - `Query OK, 0 rows affected (0.04 sec)`



# Tuning

- SHOW para obtener información sobre base de datos, tablas e índices.
  - Algunos comandos dependen de los privilegios que dispongamos.
  - SHOW GRANTS FOR user
    - Muestra los privilegios de un usuario en un host determinado.

```

-
- mysql> show grants for joe@localhost;
- +-----+
- +-----+
- | Grants for joeuser@localhost
- |
- +-----+
- +-----+
- | GRANT USAGE ON *.* TO 'joeuser'@'localhost'
- IDENTIFIED BY PASSWORD '34f3a6996d856efd'
- |
- | GRANT ALL PRIVILEGES ON testDB.* TO
- 'joeuser'@'localhost' |
- +-----+
- +-----+

```

# Tuning



- SHOW DATABASES [LIKE something]

-

- Listado de bases de datos.

•

•

• `mysql> show databases;`

• `+-----+`

• `| Database |`

• `+-----+`

• `| testDB |`

• `| mysql |`

• `+-----+`

• `2 rows in set (0.00 sec)`

•



# Tuning



- SHOW [OPEN] TABLES [FROM database\_name] [LIKE something]
  - Muestra tablas de la base de datos seleccionada.

```
mysql> show tables;
+-----+
| Tables_in_testDB |
+-----+
| grocery_inventory |
| email |
| master_name |
| myTest |
| testTable |
+-----+
5 rows in set (0.01 sec)

mysql> show open tables;
+-----+-----+-----+-----+
+-----+
| Database | Table | In_use |
| Name_locked | | |
+-----+-----+-----+-----+
+-----+
dvddatabase	tdp	0	
	0		
dvddatabase	purchase_request	0	
	0		
mysql	time_zone	0	
	0		
dvddatabase	user_collection	0	
	0		
mysql	func	0	
```

# Tuning



- SHOW CREATE TABLE table\_name
- 

- Muestra el SQL usado para la creación de una tabla.

```
mysql> show create table grocery_inventory;
+-----+
+-----+
| Table | Create Table
+-----+
+-----+
| grocery_inventory | CREATE TABLE 'grocery_inventory' (
| | 'id' int(11) NOT NULL
| | auto_increment,
| | 'item_name' varchar(50) NOT NULL
| | default "",
| | 'item_desc' text,
| | 'item_price' float NOT NULL default
| | '0',
| | 'curr_qty' int(11) NOT NULL default
| | '0',
| | PRIMARY KEY ('id')
| |) TYPE=MyISAM
+-----+
+-----+
1 row in set (0.00 sec)
```



# Tuning



- SHOW INDEX FROM table\_name [FROM database\_name]
  - Muestra información de todos los índices de una tabla.
  - Se obtiene una tabla con las siguientes columnas:
    - *Table*: Nombre de la tabla.
    - *Non\_unique*: 1=puede contener duplicado; 0=no puede contener duplicados
    - *Key\_name*: Nombre del índice.
    - *Seq\_in\_index*: Número de secuencia para el índice; comienza en 1.
    - *Column\_name*: Nombre de la columna.
    - *Collation*: Orden, A (ascendente) o NULL (no ordenado).
    - *Cardinality*: Número de valores únicos en el índice.
    - *Sub\_part*: Número de caracteres indexados o NULL si está toda la clave indexada.
    - *Packed*: El tamaño de las columnas numéricas.
    - *Null*: Si permite valores nulos.
    - *Index\_type*: Método usado de indexado (*BTREE*, *FULLTEXT*, *HASH*, *RTREE*).
    - *Comment*: Comentarios.

# Tuning



- Cómo usar el comando SHOW para obtener el estado del sistema.
  - SHOW TABLE STATUS [FROM db\_name] [LIKE something]
    - Muestra información completa de tablas.
    - Se obtiene una tabla con las siguientes columnas:
      - *Name*: Nombre de la tabla.
      - *Type*: Tipo de tabla (MyISAM, BDB, InnoDB o Gemini).
      - *Row\_format*: Formato de almacenaje de fila: fixed, dynamic o compressed.
      - *Rows*: Número de filas.
      - *Avg\_row\_length*: Longitud de fila media.
      - *Data\_length*: Longitud del archivo de datos.
      - *Max\_data\_length*: Máxima longitud del archivo de datos.
      - *Index\_length*: Longitud del archivo índice.
      - *Data\_free*: Bytes reservados y aún no usados.
      - *Auto\_increment*: Siguiete valor a usar en los campos auto\_increment.
      - *Create\_time*: Fecha y hora cuando se creó la tabla.
      - *Update\_time*: Fecha y hora cuando el archivo de datos fue actualizado por última vez.
      - *Check\_time*: Fecha y hora del último chequeo.
      - *Create\_options*: Opciones extra en la sentencia de creación

CREATE TABLE.

*Comment*: Comentarios

# Tuning

---



- SHOW STATUS [LIKE something]
  - Información del servidor.
  - Las variables más importantes:
    - - *Aborted\_connects*: Número de intentos fallidos de conexión al servidor. Puede ser debido a un username y password incorrectos o un número simultáneo de conexiones superior a las permitidas.
      - *Connections*: Suma de conexiones atendidas por el servidor en el periodo de uptime.
      - *Max\_used\_connections*: Número máximo de conexiones usadas simultáneamente durante el periodo de uptime.
      - *Slow\_queries*: Número de consultas que han tardado en ejecutarse más del valor de *long\_query\_time* (por defecto 10 s).
      - *Uptime*: Número total de segundos que el servidor ha estado funcionando en el periodo de uptime.

# Tuning

---



- SHOW VARIABLES [LIKE something]
  - Información de configuración del servidor.
  - Las variables más importantes:
    - *connect\_timeout*: Número de segundos que MySQL esperará durante una conexión antes de cerrarla.
    - *have\_innodb*: Si soporta las tablas InnoDB.
    - *have\_bdb*: Si soporta las tablas Berkeley DB.
    - *max\_connections*: El número de conexiones simultáneas permitidas.
    - *port*: Puerto dónde está funcionando MySQL.
    - *table\_type*: Tipo de tabla por defecto, suele ser:MyISAM.
    - *version*: Número de versión de MySQL.

## Monitoraje: mytop

---



- Utilidad al estilo de top de *Unix*.
- Útil para encontrar consultas problemáticas o para hacerse una idea de porqué el servidor está tan cargado.
- 
- Modos de visualizar datos:
  - Thread view (*defecto*)
  - Command view
  - Status view



# Monitoraje: mytop



- Thread view:
  - Visión general de los procesos del momento.

```

MySQL on db.example.com (4.0.15) up 2+22:52:17 [16:04:48]
Queries: 93.0M qps: 382 Slow: 2.1k Se/In/Up/De(%): 62/29/02/05
qps now: 12 Slow qps: 0.0 Threads: 26 (10/ 51) 36/15/20/25
Cache Hits: 135.0 Hits/s: 0.0 Hits now: 0.0 Ratio: 0.0% Ratio now: 0.0%
Key Efficiency: 99.5% Bps in/out: 15.4k/13.8k Now in/out: 10.0k/574.6k

 Id User Host/IP DB Time Cmd Query or State
 --- -
317134 client www02 catalog 0 Query SELECT COUNT(DISTINCT OR
317108 client www03 mysql 0 Query show full processlist
276987 client www01 catalog 2 Sleep
276969 client www02 test 3 Sleep
317129 client www01 catalog 3 Sleep
317131 client www01 mysql 3 Sleep
317133 jzawodn www01 catalog 4 Sleep
317132 client www01 catalog 5 Sleep
317130 root www01 catalog 5 Sleep
314252 readonly www02 catalog 6 Sleep
234961 readonly www01 catalog 6 Sleep
276910 client www02 catalog 9 Sleep
317062 root localhost test 10 Sleep
317048 client www02 catalog 10 Sleep
273970 root www02 catalog 10 Sleep
317127 root www01 catalog 16 Sleep
315979 repl db1 catalog 878 Binlog Slave: waiting for binlo
299354 client www03 catalog 12075 Sleep
265235 repl db14 Binlog 33981 Binlog Slave: waiting for binlo
259924 repl db25 Binlog 37581 Binlog Slave: waiting for binlo
237800 repl db22 Binlog 54927 Binlog Slave: waiting for binlo
237799 repl db24 Binlog 54927 Binlog Slave: waiting for binlo
237796 repl db21 Binlog 54930 Binlog Slave: waiting for binlo
142404 repl db13 Binlog 124119 Binlog Slave: waiting for binlo
40678 repl db12 Binlog 215526 Binlog Slave: waiting for binlo
18 repl db11 Binlog 255122 Binlog Slave: waiting for binlo
-- paused. press any key to resume --

```

# Monitoraje: mytop



- Command view:
  - Número de veces que se consulta al servidor determinados

COR

| Command            | Total    | Pct | Last | Pct |
|--------------------|----------|-----|------|-----|
| select             | 60030319 | 61% | 1342 | 78% |
| insert             | 17013017 | 17% | 10   | 0%  |
| replace            | 11600046 | 11% | 55   | 3%  |
| delete             | 5158335  | 5%  | 31   | 1%  |
| update             | 2143269  | 2%  | 279  | 16% |
| set option         | 469303   | 0%  | 0    | 0%  |
| commit             | 469302   | 0%  | 0    | 0%  |
| delete multi       | 49079    | 0%  | 0    | 0%  |
| admin commands     | 21241    | 0%  | 0    | 0%  |
| show status        | 16443    | 0%  | 1    | 0%  |
| lock tables        | 15276    | 0%  | 0    | 0%  |
| show processlist   | 10583    | 0%  | 0    | 0%  |
| show slave status  | 6916     | 0%  | 0    | 0%  |
| show master status | 6196     | 0%  | 0    | 0%  |
| unlock tables      | 5128     | 0%  | 0    | 0%  |
| change db          | 4694     | 0%  | 0    | 0%  |
| show tables        | 4397     | 0%  | 0    | 0%  |
| check              | 1029     | 0%  | 0    | 0%  |
| show variables     | 745      | 0%  | 0    | 0%  |
| drop table         | 512      | 0%  | 0    | 0%  |
| create table       | 474      | 0%  | 0    | 0%  |
| show fields        | 305      | 0%  | 0    | 0%  |
| rename table       | 118      | 0%  | 0    | 0%  |
| show databases     | 95       | 0%  | 0    | 0%  |
| create index       | 72       | 0%  | 0    | 0%  |
| repair             | 42       | 0%  | 0    | 0%  |
| show slave hosts   | 20       | 0%  | 0    | 0%  |
| show innodb status | 12       | 0%  | 0    | 0%  |
| insert select      | 8        | 0%  | 0    | 0%  |
| alter table        | 3        | 0%  | 0    | 0%  |
| show binlogs       | 3        | 0%  | 0    | 0%  |
| purge              | 3        | 0%  | 0    | 0%  |
| show create        | 2        | 0%  | 0    | 0%  |

-- paused, press any key to resume --

# Monitoraje: mytop



- Status view:
  - Resumen de los contadores de los “no comandos”.

| Counter                   | Total      | Change |
|---------------------------|------------|--------|
| Aborted_clients:          | 33307      | 0      |
| Aborted_connects:         | 7          | 0      |
| Bytes_received:           | 1690019594 | 156708 |
| Bytes_sent:               | 1013917210 | 23539  |
| Connections:              | 23193      | 0      |
| Created_tmp_disk_tables:  | 3890       | 0      |
| Created_tmp_tables:       | 3894       | 0      |
| Created_tmp_files:        | 0          | 0      |
| Delayed_insert_threads:   | 0          | 0      |
| Delayed_writes:           | 0          | 0      |
| Delayed_errors:           | 0          | 0      |
| Flush_commands:           | 1          | 0      |
| Handler_commit:           | 72886      | 19     |
| Handler_delete:           | 1613306    | 0      |
| Handler_read_first:       | 38         | 0      |
| Handler_read_key:         | 3581942    | 110    |
| Handler_read_next:        | 83880457   | 88     |
| Handler_read_prev:        | 0          | 0      |
| Handler_read_rnd:         | 2745412    | 57     |
| Handler_read_rnd_next:    | 36566543   | 0      |
| Handler_rollback:         | 5787       | 0      |
| Handler_update:           | 2154433    | 57     |
| Handler_write:            | 5975014    | 267    |
| Key_blocks_used:          | 375052     | 0      |
| Key_read_requests:        | 50594275   | 444    |
| Key_reads:                | 412307     | 0      |
| Key_write_requests:       | 12568078   | 88     |
| Key_writes:               | 7517370    | 40     |
| Max_used_connections:     | 108        | 0      |
| Not_flushed_key_blocks:   | 0          | 0      |
| Not_flushed_delayed_rows: | 0          | 0      |
| Open_tables:              | 512        | 0      |
| Open_files:               | 964        | 0      |
| Open_streams:             | 0          | 0      |

## Chequear y reparar tablas

---



- Algunas veces las tablas tipo MyISAM contienen errores.
- Los síntomas son consultas que abortan inesperadamente y observación de errores como:
  - `tbl\_name.frm' is locked against change
  - Can't find file `tbl\_name.MYI' (Errcode: ####)
  - Unexpected end of file
  - Record file is crashed
  - Got error #### from table handler

# Chequear y reparar tablas



- Chequeo de una tabla:
  - Usaremos el comando *myisamchk*:
    - `myisamchk tbl_name`
      - Encontrará el 99.99% de los errores. No busca corrupción en los datos (es muy inusual)
    - `myisamchk -m tbl_name`
      - Encontrará el 99.999% de los errores
    - `myisamchk -e tbl_name`
      - Chequeo completo.
      - Muy lento para tablas grandes.

# Chequear y reparar tablas



- Reparación de una tabla:
  - Debemos parar el servidor MySQL antes de reparar las tablas.
  - Usaremos el comando *myisamchk*:
    - Primero probaremos `myisamchk -r -q tbl_name` (“quick recovery mode”)
      - Intentará reparar el archivo índice sin tocar el archivo de datos.
    - Si no funciona el siguiente paso será:
      - Hacer backup de los archivos de datos.
      - Usar `myisamchk -r tbl_name` (“recovery mode”).
        - » Borrará registros incorrectos y borrados del archivo de datos y reconstruirá el archivo índice.
    - El siguiente paso en caso que no funcione el precedente es:
      - Usar `myisamchk --safe-recover tbl_name`
        - » Método antiguo de recuperación que en algunos casos recupera lo que el modo normal no puede.
        - » Es muy lento.

(aún existen más pasos, consultar al manual)

# Backup

---



- Se dispone de dos opciones:
  - Volcar datos (a nivel de SQL o en *csv*)
  - Copiando los archivos del directorio de la base de datos.
- Backups de como mínimo una tabla.
- Antes de iniciar el proceso se debe bloquear la escritura en las tablas y vaciar las cachés.
  - LOCK TABLE
  - FLUSH TABLES

# Backup

---



- Backup volcando datos:

## 1) CSV (sólo datos):

- `mysql> SELECT INTO ... OUTFILE`
  - **Restaurar** con: `mysql> LOAD DATA INFILE 'data.txt' INTO TABLE db2.my_table`

## 2) SQL (datos y/o estructura):

- `shell> mysqldump`
  - `mysqldump [options] db_name [tables]`
  - `mysqldump [options] --databases DB1 [DB2 DB3...]`
  - `mysqldump [options] --all-databases`
    - » Ex: `mysqldump --all-databases > all_databases.sql`
  - **Restaurar** con: `shell> mysql database_name < database.sql`



# Backup

---



- Backup de archivos del directorio de base de datos:
    - **SOLO PARA LOS TIPOS MYISAM E ISAM** (NUNCA InnoDB)
    -
- 1) Copiar los archivos: \*.frm, \*.MYD, \*.MYI
  - 2) `mysql> BACKUP TABLE tbl_name [, tbl_name] ... TO  
'/path/to/backup/directory'`
  - 3) *Al restaurar se debería parar el servidor MySQL o bien ejecutar **REPAIR TABLE** sobre la tabla restaurada.*

# Replicar

---



- Sistema para copiar todos los datos de un servidor a otro.
- Por ejemplo para compartir datos con una oficina remota.
- La replicación permite configurar fácilmente un segundo servidor como *slave* de un *master*.
- 
- En esta documentación se tratará brevemente aunque la replicación es un tema extensa que merecería un curso entero.

# Replicar



- Problemas solventados con réplicas:
  - Distribución de datos:
    - Ej.
      - Mantener una copia de datos a 10.000 Km con una conexión decente.
  - Balanceo de carga:
    - Ej.
      - Servidor web dónde la mayoría de operaciones son *SELECT*
      - Podríamos utilizar *round-robin DNS*, *LVS*...
  - Backup y restauración:
    - Cuando la demanda del acceso es 24h la mejor solución es configurar un *slave*.
  - Alta disponibilidad:
    - Configurando la réplica se evita que *MySQL* sea un punto de fallada de nuestra aplicación.
    - Se pueden preparar mecanismos automáticos que en caso de fallada del *master* se desvien las conexiones al *slave*.

# Replicar

---



- Problemas no solventados con réplicas:
  - El mayor inconveniente de las réplicas es el rendimiento.
  - En aplicaciones que requieren un muy alto índice de escrituras en la bbdd, los servidores *slaves* deben de ser como mínimo tan potentes com el *master*. Para este caso el balanceo de carga con réplicas es una mala opción.
  - Si la carga en el *slave* es alta no se puede asegurar que los datos estarán sincronizados a cada instante.
  - El ancho de banda y latencia de la red también supone un inconveniente.
  - Dos ejemplos que no son fáciles de implementar con réplicas:
    - Transmisión de datos en tiempo real.
    - Peticiones online (ejecución de largas transacciones, ej: informes, estadísticas...).

# Replicar

---



- Configurando réplicas en 4 pasos:
  - 1. Crear una cuenta de réplica en cada servidor.
  - 2. Añadir entradas de configuración en my.cnf en cada servidor.
  - 3. Reiniciar el master y verificar la creación del log binario.
  - 4. Reiniciar el slave y verificar que la replicación funcione.
  -
- Se supone una nueva instalación si ya existían datos deberíamos copiarlos previamente al *slave* como *snapshot*.

# Replicar



- 1. Crear una cuenta de réplica en cada servidor:

- 
- `mysql> GRANT REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO`
- `repl@"192.168.1.0/255.255.255.0" IDENTIFIED BY 'c0pyIT!';`
- Query OK, 0 rows affected (0.00 sec)
- 

- Verificamos que sea correcto:

- `mysql> SHOW GRANTS FOR repl;`
- +-----+  
 -----+  
 |Grants for repl@"192.168.1.0/255.255.255.0"  
 |  
 +-----+  
 -----+  
 | GRANT REPLICATION SLAVE, REPLICATION CLIENT ON \*.\* TO 'repl'@'...' IDENTIFIED  
 BY ...|  
 +-----+  
 -----+  
 1 row in set (0.00 sec)  
 •  
 •

- ¿Porqué crear las cuentas en los dos servidores?

- Si falla el *master* configuraremos el *slave* como *master* y cuando se restaure el antiguo *master* tal vez queramos dejarlo configurado como *slave*

# Replicar

---



- 2. Añadir entradas de configuración en my.cnf en cada servidor:
  - En el *master*, en la sección *[mysqld]*:
    - 
    - log-bin
    - server-id = 1 #*¡cada servidor debe tener un identificador único!*
    - log-bin = /var/db/repl/log-

# Replicar

---



- 3. Reiniciar el *master* y verificar la creación del *log* binario:
  - Paramos y reiniciamos el *master*.
  - Verificamos que se cree el archivo `master-bin.001` (el formato del nombre es `hostname-bin.001`)
  - También podemos utilizar la utilidad `mysqlbinlog` para ver el contenido
    - `$ mysqlbinlog master-bin.001`
    - `# at 4`
    - `#020922 14:59:11 server id 1 log_pos 4 Start:  
binlog v 3, server v 4.0.4-beta-log created  
020922 14:59:11`



# Replicar

---



- 4. Reiniciar el *slave* y verificar que la replicación funcione:
  - Paramos y reiniciamos el *slave*.
  - Verificamos el log de errores:
    - 021103 13:58:10 Slave I/O thread: connected to master 'repl@master:3306', replication started in log 'log-bin.001' at position 4
  - Nos indica que funciona.
  - 
  - Ya podemos insertar algún registro en el *master* y ver el reflejo en el *slave*.

# Replicar



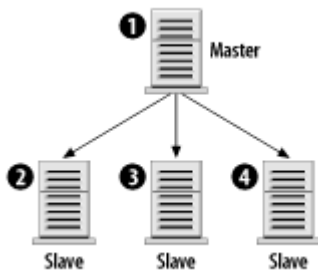
- Arquitecturas:

- Normas:

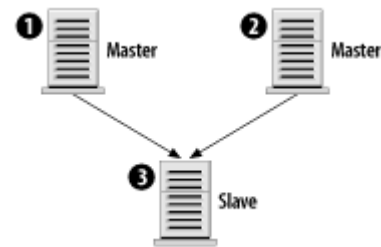
- Cada *slave* debe tener un único *ID*.
- Un *slave* puede tener sólo un *master*.
- Un *master* puede tener muchos *slaves*.
- Los *slaves* también pueden ser *master* de otros *slaves*.

-

- Master con slaves



- Slave con dos masters (!)



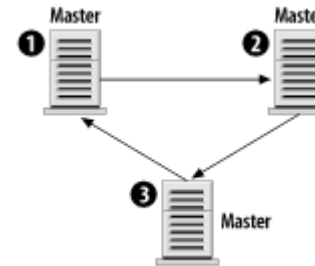
# Replicar



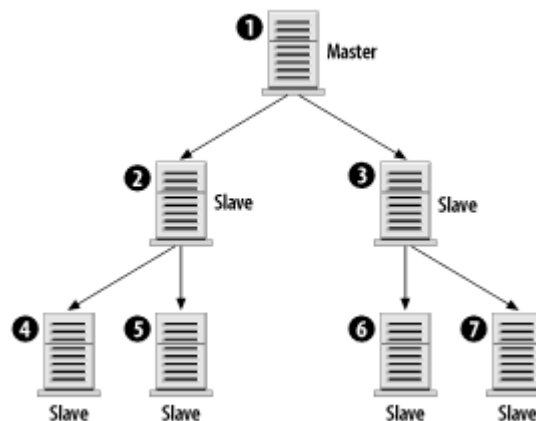
- Dual master



- Anillo de réplica



- Pirámide



## phpMyAdmin



- Herramienta de administración y desarrollo de MySQL en entorno web.

**Bienvenido a phpMyAdmin 2.6.4-pl2**  
MySQL 4.1.14-Debian\_6-log ejecutándose en localhost como root@localhost

Base de datos:  
(Bases de datos) ...

Seleccionar una base de datos

**MySQL**

- Crear nueva base de datos (Cotejamiento) **Crear**
- Mostrar información de tiempo de ejecución de MySQL
- Mostrar las variables del sistema MySQL
- Mostrar procesos
- Juego de caracteres y sus cotejamientos
- Motores de almacenamiento
- Reinicio de MySQL
- Privilegios
- Log binario
- Bases de datos
- Exportar

**phpMyAdmin**

- Language: Spanish (es-utf-8)
- Juegos de caracteres de MySQL: UTF-8 Unicode (utf8)
- Cotejamiento de las conexiones MySQL: utf8\_general\_ci
- Tema / Estilo: Darkblue/orange
- Documentación de phpMyAdmin
- Mostrar la información de PHP
- Página oficial de phpMyAdmin [ChangeLog] [CVS] [Lists]

# Recursos

---



- Links recomendados:
  - Documentación oficial:
    - <http://dev.mysql.com/doc/>



# Recursos

- Bibliografía recomendada:

- **MySQL, Second Edition**
- By Paul DuBois
- Publisher: Sams
- Pub Date: January 17, 2003
- ISBN: 0-7357-1212-3
- Pages: 1248
- 
- **Sams Teach Yourself PHP, MySQL® and Apache All in One**
- By Julie C. Meloni
- Publisher: Sams Publishing
- Pub Date: December 18, 2003
- ISBN: 0-672-32620-5
- Pages: 624
- 
- **High Performance MySQL**
- By Derek J. Balling, Jeremy Zawodny
- Publisher: O'Reilly
- Pub Date: April 2004
- ISBN: 0-596-00306-4
- Pages: 294
- 
- **MySQL Cookbook**
- By Paul Dubois
- Publisher: O'Reilly
- Pub Date: October 2002
- ISBN: 0-596-00145-2
- Pages: 1022
-