

Guion para respaldar MySQL

Para bases de datos pequeñas, de 1 a 3 gigabytes, la mejor manera de respaldar es **de manera lógica** ([sobre PostgreSQL tenemos un artículo](#) en detalle con respecto a ese tema). Es decir, se extrae los datos en el formato deseado y se puede importar a otro servidor, llegado el caso. Acá giramos hacia realizar respaldos totales comprimidos de una o más bases de datos, de manera local o remota, *veamos*.

Los ramanes lo hacen todo por triplicado

Somos fanáticos de la *ciencia ficción dura* y una de las mejores novelas es «[Cita con Rama](#)», escrita por Arthur C. Clarke en 1972. Allí expresan la "manía" de los ramanes -extraterrestres que construyeron la nave espacial que se acerca al sistema solar- de realizar todo por triplicado.

Pues no es mala idea realizar eso, comencemos por allí, directo al grano:

1. Este guion extrae los datos de MySQL y los comprime y añade la funcionalidad para que **rsync** pueda manejar mejor dicha información. Este primer respaldo es local, con ciertas medidas de seguridad básica.
2. En la misma red de área local establecemos un servidor de archivos que, como dijimos, mantiene sincronizadas las carpetas con **rsync**. Este sería el segundo respaldo.
3. Fuera de las instalaciones físicas de la red de área local también subiremos los respaldos, este sería el tercer respaldo.

Este guion cubre el punto número uno, los otros dos puntos son material para otro artículo. Ahora sí, pasemos a los detalles.

Repositorio en GitHub

En nuestra cuenta gratuita en GitHub hemos decidido publicar un repositorio para todas las utilidades que usamos día a día en GNU/Linux **pero escritas de manera genérica para que puedan ser usadas por gran cantidad de personas**. Está amparada bajo la Licencia GNU 3 y [este es el enlace directo al guion](#).

Código fuente

Explicación línea a línea

- En la línea 45 y 46 colocaremos las credenciales del usuario que tiene derechos de lectura sobre la base de datos. Bien se puede colocar la contraseña directamente y en texto plano en el guion *pero recomendamos encarecidamente utilicen un método alterno para ello*.
 - Una manera es usar el complemento de autenticación "auth_socket" para que coincida las credenciales del usuario en el sistema GNU/Linux. El inconveniente de este método es que solamente puede ser usado en conexiones locales; este guion no tiene manera de saber el método de autenticación empleado en MySQL. **El guion tampoco devuelve un resultado para evaluar si el respaldo fue exitoso o no, ojo con eso.**
 - Otra manera de autenticar es guardar la contraseña en el archivo de configuración de MySQL llamado **my.cnf** en la sección **[client]**. Según [este extenso artículo](#), dicho archivo de configuración puede estar ubicado en varios lugares distintos: /etc/my.cnf /etc/mysql/my.cnf /var/lib/mysql/my.cnf y deben ser editados con derechos de superusuario, pero no se necesitan de dichos derechos para ejecutar **mysql** y **mysqldump** (ambas son herramientas para la terminal). La desventaja es que siempre introducirá la misma contraseña para todos los usuarios: de manera curiosa si nos conectamos remoto a otro servidor MySQL también "pasará" la contraseña colocada en el fichero de configuración. **También se puede especificar ficheros diferentes de configuración:** esencialmente usando la opción `--defaults-file = nombre_de_fichero` (dicho fichero debe tener atributos de solo lectura y modificación solamente por el usuario en cuestión, todo está explicado en detalle -en idioma inglés- [en este artículo](#)).
- Este guion asume entonces que, si no establecemos contraseña en el guion, es que habremos configurado todo lo anterior de manera adecuada.
- En la línea 49 establecemos dónde está el servidor MySQL a respaldar. Volviendo al tema de la seguridad, recomendamos que sea hecho de manera local, es decir, en la misma máquina donde se ejecuta MySQL.
- En las líneas 54, 57 y 58 configuramos la ubicación y formato del respaldo comprimido.
- En la línea 61 podemos especificar las bases de datos a respaldar, si se deja en blanco o vacío se asume que son todas las bases de datos *donde el usuario tenga derechos de lectura*.
- En la línea 63 podemos colocar las bases de datos a excluir, hemos colocado algunos de

manera predeterminada; tengan en cuenta que si una misma base de datos se incluye en ambas listas, *dicha base de datos no será respaldada*.

- De las líneas 67 a 69 son variables que es preferible no modificar. Noten que la variable **ksMYSQL** es usada en el nombre del fichero de respaldo comprimido, que si lo hacemos de manera local pues es el mismo origen (excelente si el ordenador [tiene el nombre completo](#) para sí identificar los respaldos a futuro). La variable *ksJustoAhora* tiene una [manera peculiar para nombrar](#) y que permite ser ordenado por nombre en el sistema operativo.
- En la línea 77 es creada, si no existe, las carpetas necesarias que albergarán los respaldos comprimidos. La idea que esta carpeta principal llamada «Respaldos» o en su defecto la llamada «Respaldos MySQL» sea sincronizada con otros equipos en la red de área local o en Internet con **rsync**.
- En la línea 80 obtenemos la lista de base de datos.
- De la línea 82 a la 124 es el bloque principal donde vamos iterando base de datos por base de datos, una en una.
- En la línea 85 comprobamos que la base de datos iterada esté en la lista a respaldar, si la lista está vacía pues simplemente establece a verdadero la variable *ksRespaldar*.
- En la línea 98 comprobamos si la base de datos iterada se encuentra en la lista de ignoradas, de nuevo utiliza la variable booleana *ksRespaldar* según sea el caso.
- En la línea 111 está el motor del guion: crea un nombre de archivo muy específico para identificar el origen, fecha y hora del respaldo en cuestión.
- Línea 120: prepara los argumentos para **mysqldump**, [este artículo indica](#) cómo evitar un mensaje de error (opción **--column-statistics = 0**).
- En la línea 121 de manera comentada (si la descomentamos) podemos hacer una corrida de depuración para visualizar la lista definitiva a respaldar (**nota, s edebe comentar así la siguiente línea**).
- En la línea 122, por fin, se conecta y obtiene la base de datos y se pasa directo por medio de comando tubería, al programa **gzip** que tiene la opción **-9** (mejor método de compresión) y **--rsyncable**, opción que facilita a **rsync** su trabajo.
- Este guion no cifra el archivo de respaldo comprimido, sin embargo existen muchos tutoriales en línea que pueden ayudar al respecto.
- Como somos sumamente prácticos proponemos que se establezca otro guion, esta vez con la cuenta superusuario, y se [agregue a una tarea programada diaria](#) para copiemos a otra carpeta local. De esta manera solamente el superusuario podrá abrir dichos *respaldos de los respaldos comprimidos*.

Restaurar la base de datos

- De un archivo comprimido, debemos usar **gzip -d -k nombre_fichero_comprimido**: la

opción **-d** indica descomprimir y **-k** para mantener, no borrar el archivo comprimido después de haber sido extraído.

- Ahora bien podemos conectar con **mysql** por línea de comandos, borrar o crear la base de datos -según sea el caso- y abrir dicha base de datos **USE *base_de_datos***; . A continuación usamos el comando "**source *ubicación_del_fichero/nombre_del_fichero***" y esperamos que finalice. [Este artículo](#), aunque explicado para ambiente Windows, ilustra bien sobre el tema.

No es el fin, es apenas el comienzo

Como decimos siempre, *esta es una maratón, no una carrera contra reloj*, evidentemente que tendremos algún u algunos errores a corregir **y muchas cosas que mejorar y/o ampliar**. Por eso, como todos nuestros artículos, queda abierto a partir de este punto para seguir en dicho trabajo.