

Creacion y restauracion de respaldos
dentro de contenedores

Examen

Segundo parcial

Base de datos

RAFAEL ALEJANDRO RAYGOSA MENA

Contenido

Objetivo 2

Alcance 3

Dump 4

Binario 5

Completo 5

 Incremental 6

Restauración completa 7

 Base de datos vacía 7

 Base de datos tras la restauración 7

Conclusión 8

Objetivo

El objetivo de este documento es explicar el proceso necesario para la creación de respaldos de distintos tipos, los cuales son: Dump, binario completo y binario incremental. Todo esto se mostrará con la serie de comandos necesarios, así como el proceso de restauración según cada tipo de respaldo.

Alcance

Primeramente, es necesario tener instalado el servicio de Docker en tu dispositivo, con el cual se montó el contenedor del archivo yml en el cual está la base de datos a respaldar. Esta base puede ser accedida y modificada utilizando un administrador de BD como Workbench o HeidiSQL. Para crear y restaurar los respaldos, es necesario seguir una serie de comandos en CMD mientras se ejecuta Docker.

Dump

En la carpeta donde se creo el contenedor ejecutamos el siguiente comando:

```
C:\clase>docker exec -i clase_basedatos_1 mysqldump -u root -p123456 --default-character-set=utf8 --routines --skip-triggers --add-drop-table=false myfirstdb > respaldos.sql
Warning: World-writable config file '/etc/mysql/conf.d/my.cnf' is ignored
```

Esto hará que se cree un archivo nombrado “respaldos.sql” en la carpeta:

respaldos.sql	17/10/2020 19:28	Archivo de origen ...	2 KB
respaldos.sql	9/11/2020 21:51	Archivo de origen ...	8 KB

Para restaurar la base de datos se debe ejecutar el siguiente comando:

```
C:\clase>docker exec -i clase_basedatos_1 mysql myfirstdb < respaldos.sql -p123456 -u root
Warning: World-writable config file '/etc/mysql/conf.d/my.cnf' is ignored
```

Binario

Completo

Para crear el respaldo primero se ejecuta el siguiente comando, tras el cual aparecerán una serie de mensajes como en la imagen:

```
C:\clase>docker exec -i clase_basedatos_1 mariabackup --backup --target-dir=dbbackups --user=root --password=123456
Warning: World-writable config file '/etc/mysql/conf.d/my.cnf' is ignored
Warning: World-writable config file '/etc/mysql/conf.d/my.cnf' is ignored
Warning: World-writable config file '/etc/mysql/conf.d/my.cnf' is ignored
[00] 2020-11-10 04:09:29 Connecting to MySQL server host: localhost, user: root, password: set, port: not set,
socket: /run/mysqld/mysqld.sock
[00] 2020-11-10 04:09:29 Using server version 10.5.5-MariaDB-1:10.5.5+maria~focal
mariabackup based on MariaDB server 10.5.5-MariaDB debian-linux-gnu (x86_64)
[00] 2020-11-10 04:09:29 uses posix_fadvise().
[00] 2020-11-10 04:09:29 cd to /var/lib/mysql/
[00] 2020-11-10 04:09:29 open files limit requested 0, set to 1048576
[00] 2020-11-10 04:09:29 mariabackup: using the following InnoDB configuration:
[00] 2020-11-10 04:09:29 innodb_data_home_dir =
[00] 2020-11-10 04:09:29 innodb_data_file_path = ibdata1:12M:autoextend
[00] 2020-11-10 04:09:29 innodb_log_group_home_dir = ./
[00] 2020-11-10 04:09:29 InnoDB: Using Linux native AIO
```

Esto creará el directorio dbbackups dentro del contenedor. Para extraerlo utilizamos el siguiente comando:

```
C:\clase>docker cp clase_basedatos_1:/dbbackups dbbackups
```

Así tendremos el directorio dentro de la carpeta donde creamos el contenedor



Incremental

Para crear el respaldo primero se ejecuta el siguiente comando:

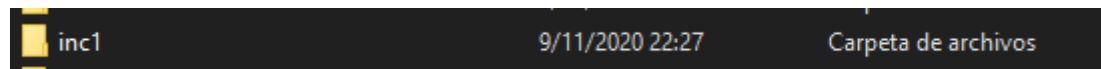
```
C:\clase>docker exec -i clase_basedatos_1 mariabackup --backup --target-dir=inc --incremental-basedir=dbbackup
s --user=root --password=123456
```

```
C:\clase>docker exec -i clase_basedatos_1 mariabackup --backup --target-dir=dbbackups --user=root --password=1
23456
Warning: World-writable config file '/etc/mysql/conf.d/my.cnf' is ignored
Warning: World-writable config file '/etc/mysql/conf.d/my.cnf' is ignored
Warning: World-writable config file '/etc/mysql/conf.d/my.cnf' is ignored
[00] 2020-11-10 04:09:29 Connecting to MySQL server host: localhost, user: root, password: set, port: not set,
socket: /run/mysqld/mysqld.sock
[00] 2020-11-10 04:09:29 Using server version 10.5.5-MariaDB-1:10.5.5+maria~focal
mariabackup based on MariaDB server 10.5.5-MariaDB debian-linux-gnu (x86_64)
[00] 2020-11-10 04:09:29 uses posix_fadvise().
[00] 2020-11-10 04:09:29 cd to /var/lib/mysql/
[00] 2020-11-10 04:09:29 open files limit requested 0, set to 1048576
[00] 2020-11-10 04:09:29 mariabackup: using the following InnoDB configuration:
[00] 2020-11-10 04:09:29 innodb_data_home_dir =
[00] 2020-11-10 04:09:29 innodb_data_file_path = ibdata1:12M:autoextend
[00] 2020-11-10 04:09:29 innodb_log_group_home_dir = ./
[00] 2020-11-10 04:09:29 InnoDB: Using Linux native AIO
```

Esto creará el directorio inc1 dentro del contenedor. Para extraerlo utilizamos el siguiente comando:

```
C:\clase>docker cp clase_basedatos_1:/inc/ inc1
```

Así tendremos el directorio dentro de la carpeta donde creamos el contenedor



Restauración completa

Primero se debe preparar el directorio que contenga el respaldo utilizando los siguientes comandos:

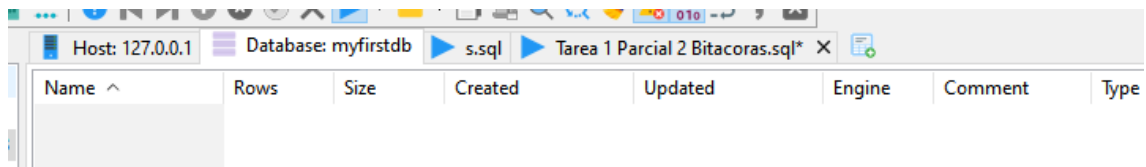
```
C:\clase>docker exec -i clase_basedatos_1 mariabackup --prepare --target-dir=/dbbackups/  
mariabackup based on MariaDB server 10.5.5-MariaDB debian-linux-gnu (x86_64)  
  
C:\clase>docker exec -i clase_basedatos_1 mariabackup --prepare --target-dir=/dbbackups/ --incremental-dir=/inc/  
mariabackup based on MariaDB server 10.5.5-MariaDB debian-linux-gnu (x86_64)
```

Después para restaurar la información ejecutamos el siguiente comando, que mostrará varios mensajes como en la siguiente imagen, seguido por una confirmación de éxito:

```
C:\clase>docker exec -i clase_basedatos_1 mariabackup --copy-back --target-dir=/dbbackups/  
Warning: World-writable config file '/etc/mysql/conf.d/my.cnf' is ignored  
Warning: World-writable config file '/etc/mysql/conf.d/my.cnf' is ignored  
Warning: World-writable config file '/etc/mysql/conf.d/my.cnf' is ignored  
mariabackup based on MariaDB server 10.5.5-MariaDB debian-linux-gnu (x86_64)  
[01] 2020-11-10 05:40:12 Copying ibdata1 to /var/lib/mysql/ibdata1
```

Antes de ejecutar una restauración completa es necesario asegurarse que el directorio destino esté vacío, por lo que si no está así hay que acceder al directorio /var/lib/mysql dentro del contenedor y ejecutar el comando “rm -R *”.

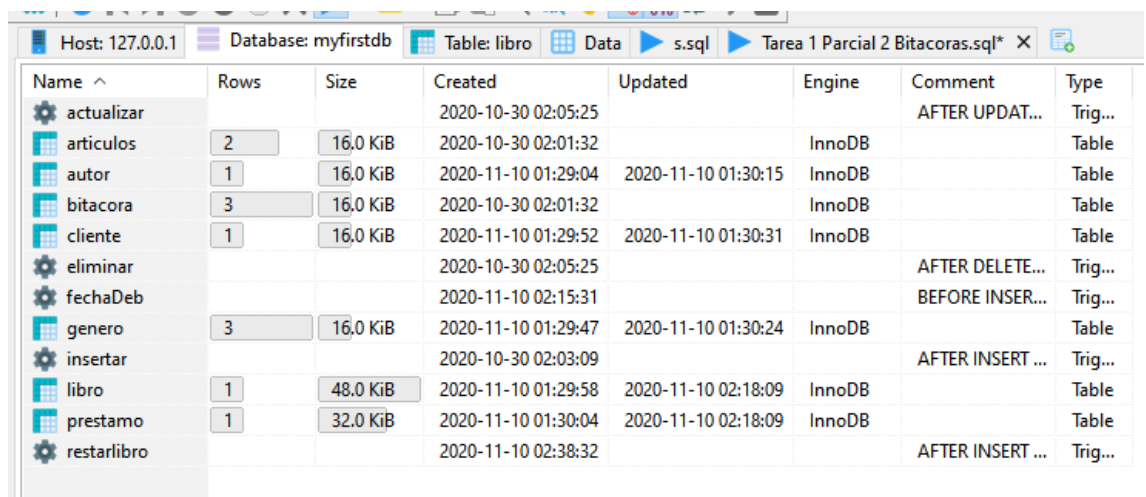
Base de datos vacía



The screenshot shows a database management interface with a table named 's.sql' in the 'myfirstdb' database. The table is empty, and the columns are: Name, Rows, Size, Created, Updated, Engine, Comment, and Type.

Name	Rows	Size	Created	Updated	Engine	Comment	Type
------	------	------	---------	---------	--------	---------	------

Base de datos tras la restauración



The screenshot shows a database management interface with a table named 'libro' in the 'myfirstdb' database. The table is restored with data, and the columns are: Name, Rows, Size, Created, Updated, Engine, Comment, and Type.

Name	Rows	Size	Created	Updated	Engine	Comment	Type
actualizar			2020-10-30 02:05:25			AFTER UPDAT...	Trig...
articulos	2	16.0 KiB	2020-10-30 02:01:32		InnoDB		Table
autor	1	16.0 KiB	2020-11-10 01:29:04	2020-11-10 01:30:15	InnoDB		Table
bitacora	3	16.0 KiB	2020-10-30 02:01:32		InnoDB		Table
cliente	1	16.0 KiB	2020-11-10 01:29:52	2020-11-10 01:30:31	InnoDB		Table
eliminar			2020-10-30 02:05:25			AFTER DELETE...	Trig...
fechaDeb			2020-11-10 02:15:31			BEFORE INSE...	Trig...
genero	3	16.0 KiB	2020-11-10 01:29:47	2020-11-10 01:30:24	InnoDB		Table
insertar			2020-10-30 02:03:09			AFTER INSERT ...	Trig...
libro	1	48.0 KiB	2020-11-10 01:29:58	2020-11-10 02:18:09	InnoDB		Table
prestamo	1	32.0 KiB	2020-11-10 01:30:04	2020-11-10 02:18:09	InnoDB		Table
restarlibro			2020-11-10 02:38:32			AFTER INSERT ...	Trig...

Conclusión

Hacer respaldos de la información es un proceso muy importante para la seguridad de un sistema o programa. Nunca se sabe cuándo se puede ocasionar un daño los elementos que contiene o se dañe el dispositivo donde está almacenado, por lo que es muy recomendable hacer estos de forma regular.