Mariadb数据库备份恢复系列(一)：mysqldump逻辑备份工具实战

实验一：利用mysqldump+二进制日志实现备份恢复数据库

1、查看原始数据

```
MariaDB [(none)]> USE testdb;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [testdb]> SHOW TABLES;
+------------------+
| Tables_in_testdb |
+------------------+
| courses          |
| students         |
+------------------+
2 rows in set (0.01 sec)

MariaDB [testdb]> SELECT * FROM students;
+------+----------+--------+------+------+
| sid  | name     | gender | age  | cid  |
+------+----------+--------+------+------+
|    6 | sun sir1 | M      |   28 |    3 |
|    8 | nwc123   | M      |   22 |    3 |
+------+----------+--------+------+------+
2 rows in set (0.00 sec)
```

初始时的数据信息

2、mysqldump执行备份

```
[root@node72 ~]# mysqldump -uroot -hlocalhost --single-transaction --routines --triggers --master-data=2 --flush-logs --databases testdb > /tmp/dbback1.sql
[root@node72 ~]#
[root@node72 ~]# cat /tmp/dbback1.sql
-- MySQL dump 10.14  Distrib 5.5.44-MariaDB, for Linux (x86_64)
--
-- Host: localhost    Database: testdb
-- ------------------------------------------------------
-- Server version       5.5.44-MariaDB-log

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Position to start replication or point-in-time recovery from
--
```

表示备份testdb库，--databases后可跟多个库名，表示备份多个库，如果备份全部的库，则要用--all-databases，如果只想备份某个库的某个表，则可以直接写上库名后跟表名

--single-transaction表示对InnoDB存储引擎执行热备时的选项

--routines表示备份存储过程和存储函数

--triggers 表示备份触发器

--master-data=2表示在备份出来的文件中指明备份时刻的二进制文件的位置

--flush-logs表示备份那一刻进行日志刷写，用于实现二进制日志滚动

在备份出来的文件中，显示了备份时刻的

```
-- CHANGE MASTER TO MASTER_LOG_FILE='binary.000006', MASTER_LOG_POS=245;
--
-- Current Database: `testdb`
--

CREATE DATABASE /*!32312 IF NOT EXISTS*/ `testdb` /*!40100 DEFAULT CHARACTER SET latin1 */;

USE `testdb`;

--
-- Table structure for table `courses`
--

DROP TABLE IF EXISTS `courses`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `courses` (
  `courseid` int(11) DEFAULT NULL,
  `cname` varchar(30) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `courses`
--

LOCK TABLES `courses` WRITE;
/*!40000 ALTER TABLE `courses` DISABLE KEYS */;
INSERT INTO `courses` VALUES (2,'Computer'),(5,'zabbix'),(6,'zabbix'),(3,'Linux OPS');
/*!40000 ALTER TABLE `courses` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `students`
--

DROP TABLE IF EXISTS `students`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `students` (
  `sid` tinyint(3) unsigned DEFAULT NULL,
  `name` char(20) DEFAULT NULL,
  `gender` enum('M','F') DEFAULT NULL,
  `age` tinyint(3) unsigned DEFAULT NULL,
  `cid` tinyint(3) unsigned DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `students`
--

LOCK TABLES `students` WRITE;
/*!40000 ALTER TABLE `students` DISABLE KEYS */;
INSERT INTO `students` VALUES (6,'sun sir1','M',28,3),(8,'nwc123','M',22,3);
/*!40000 ALTER TABLE `students` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Dumping routines for database 'testdb'
--
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;
```

正使用的二进制日志文件及其对应的位置，因此如果利用二进制日志进行还原时，可以基于此位置开始进行还原

有创建数据库的语句，因为用--databases指明了备份的库，如果用只备份表的方式进行备份，则不会有创建库的语句存在

创建表的语句

插入数据的语句

创建表的语句

插入数据的语句

3、模拟在备份后修改数据，模拟利用二进制日志还原的场景

```
MariaDB [(none)]> use testdb;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [testdb]> SELECT * FROM students;
+------+----------+--------+------+------+
| sid  | name     | gender | age  | cid  |
+------+----------+--------+------+------+
|    6 | sun sir1 | M      |   28 |    3 |
|    8 | nwc123   | M      |   22 |    3 |
+------+----------+--------+------+------+
2 rows in set (0.01 sec)

MariaDB [testdb]> INSERT INTO students VALUES (1,'nwc1','M',18,2),(2,'newbee','F',20,1);
Query OK, 2 rows affected (0.01 sec)
Records: 2  Duplicates: 0  Warnings: 0

MariaDB [testdb]> SELECT * FROM students;
+------+----------+--------+------+------+
| sid  | name     | gender | age  | cid  |
+------+----------+--------+------+------+
|    6 | sun sir1 | M      |   28 |    3 |
|    8 | nwc123   | M      |   22 |    3 |
|    1 | nwc1     | M      |   18 |    2 |
|    2 | newbee   | F      |   20 |    1 |
+------+----------+--------+------+------+
4 rows in set (0.02 sec)
```

模拟在完全备份之后，又进行了数据库的相关操作，以演示用二进制日志还原的目的

4、还原mysqldump备份的数据

```
[root@node73 ~]# scp 10.1.32.72:/tmp/dbback1.sql /tmp
root@10.1.32.72's password:
dbback1.sql                                   100% 3027     3.0KB/s   00:00
[root@node73 ~]#
[root@node73 ~]# mysql
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 2
Server version: 5.5.44-MariaDB-log MariaDB Server

Copyright (c) 2000, 2015, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
MariaDB [(none)]> SHOW GLOBAL VARIABLES LIKE '%log_bin%';
+---------------------------------+-------+
| Variable_name                   | Value |
+---------------------------------+-------+
| log_bin                         | ON    |
| log_bin_trust_function_creators | OFF   |
| sql_log_bin                     | ON    |
+---------------------------------+-------+
3 rows in set (0.01 sec)

MariaDB [(none)]>
MariaDB [(none)]> SET sql_log_bin=OFF;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> SHOW GLOBAL VARIABLES LIKE '%log_bin%';
+---------------------------------+-------+
| Variable_name                   | Value |
+---------------------------------+-------+
| log_bin                         | ON    |
| log_bin_trust_function_creators | OFF   |
| sql_log_bin                     | ON    |
```

模拟在另一个节点上实现还原操作，先将备份的文件拷贝到要还原的节点上

将此会话级别的二进制日志文件的记录功能关闭，因为还原时，也会产生大量的二进制日志事件信息，而此时产生的二进制日志事件是没有任何意义的，因此要关闭

```
+---------------------------------+-------+
3 rows in set (0.00 sec)

MariaDB [(none)]> SHOW VARIABLES LIKE '%log_bin%';
+---------------------------------+-------+
| Variable_name                   | Value |
+---------------------------------+-------+
| log_bin                         | ON    |
| log_bin_trust_function_creators | OFF   |
| sql_log_bin                     | OFF   |
+---------------------------------+-------+
3 rows in set (0.00 sec)

MariaDB [(none)]>
MariaDB [(none)]> SOURCE /tmp/dbback1.sql;
Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

MariaDB [testdb]> SHOW TABLES;
+------------------+
| Tables_in_testdb |
+------------------+
| courses          |
| students         |
+------------------+
2 rows in set (0.00 sec)

MariaDB [testdb]> SELECT * FROM students;
+------+----------+--------+------+------+
| sid  | name     | gender | age  | cid  |
+------+----------+--------+------+------+
|    6 | sun sir1 | M      |   28 |    3 |
|    8 | nwc123   | M      |   22 |    3 |
+------+----------+--------+------+------+
2 rows in set (0.00 sec)

MariaDB [testdb]>
```

导入备份的二进制日志文件，实现备份恢复的目的

查看还原出来的数据，发现并没有此备份数据之后产生的新数据，因此要用二进制日志进行还原新产生的数据

5、导出二进制日志的事件

```
[root@node72 ~]#
[root@node72 ~]# mysqlbinlog --start-position=245 /var/lib/mysql/binary.000006 > /tmp/2jz.sql
[root@node72 ~]#
[root@node72 ~]# cat /tmp/2jz.sql
/*!50530 SET @@SESSION.PSEUDO_SLAVE_MODE=1*/;
/*!40019 SET @@session.max_insert_delayed_threads=0*/;
/*!50003 SET @OLD_COMPLETION_TYPE=@@COMPLETION_TYPE,COMPLETION_TYPE=0*/;
DELIMITER /*!*/;
# at 4
#160823 16:01:45 server id 1  end_log_pos 245   Start: binlog v 4, server v 5.5.44-MariaDB-log created 160823
 16:01:45
BINLOG '
6QK8Vw8BAAAA8QAAAPUAAAAAAAQANS41LjQ0LU1hcmlhREItbG9nAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAEzgNAAgAEgAEBAQEEgAA2QAEGggAAAAICAgCAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAA4TTf7g==
'/*!*/;
# at 245
#160823 16:15:08 server id 1  end_log_pos 315   Query   thread_id=8     exec_time=0     error_code=0
SET TIMESTAMP=1471940108/*!*/;
SET @@session.pseudo_thread_id=8/*!*/;
SET @@session.foreign_key_checks=1, @@session.sql_auto_is_null=0, @@session.unique_checks=1, @@session.autoco
mmit=1/*!*/;
```

在此前的服务器上，将完全备份那一刻开始产生的二进制日志文件的数据进行导出，指明从哪个文件的哪个位置开始导出

如果二进制日志发生过滚动，则将剩余的产生的其他二进制日志文件也要写到命令行中，如：mysql-binlog --start-position=333 binary.000006 binary.000007 binary.000008 代表复制binlog.000006的从333位置开始复制，复制binary.000007的全部事件，复制binary.000008的全部事件，此日志文件的整个内容

6、还原二进制日志事件

```
[root@node73 ~]# scp 10.1.32.72:/tmp/2jz.sql /tmp
root@10.1.32.72's password:
2jz.sql                                       100% 1821     1.8KB/s   00:00
[root@node73 ~]#
[root@node73 ~]#

MariaDB [testdb]>
MariaDB [testdb]> SOURCE /tmp/2jz.sql;
Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

MariaDB [testdb]> SELECT * FROM students;
+------+----------+--------+------+------+
| sid  | name     | gender | age  | cid  |
+------+----------+--------+------+------+
|    6 | sun sir1 | M      |   28 |    3 |
|    8 | nwc123   | M      |   22 |    3 |
|    1 | nwc1     | M      |   18 |    2 |
|    2 | newbee   | F      |   20 |    1 |
+------+----------+--------+------+------+
4 rows in set (0.02 sec)
```

将备份出来的二进制文件的文件，拷贝到需要还原的节点上

导入二进制日志的事件，完成还原

查看利用二进制日志还原后的数据，发现已经还原了mysqldump备份之后新增的数据