

NSD DBA1 DAY05

1 数据备份与恢复

1.1 问题

本案例要求熟悉MySQL的备份与恢复，完成以下任务操作：

- 逻辑备份工具 mysqldump
- 使用mysql 恢复数据库

1.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：使用mysqldump进行逻辑备份

1) 备份MySQL服务器上的所有库

将所有的库备份为mysql-all.sql文件：

```
01. [root@dbsvr1 ~]# my sql dump - u root - p -- all- databases > /root/all db. sql
02. Enter password: //验证口令
03. [root@dbsvr1 my sql]# file /root/all db. sql //确认备份文件类型
04. /root/all db. sql: UTF- 8 Unicode English text, with very long lines
```

查看备份文件alldb.sql的部分内容：

```
01. [root@dbsvr1 ~]# grep - v E ' ^ | ^ $ ' /root/all db. sql | head - 15
02. CREATE DATA BASE /*! 32312 IF NOT EXISTS */ `home` /*! 40100 DEFAULT CHARACTER SI
03. USE `home`;
04. DROP TABLE IF EXISTS `biao01`;
05. CREATE TABLE `biao01` (
06.   `id` int( 2) NOT NULL,
07.   `name` varchar( 8) DEFAULT NULL
08. ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
09. LOCK TABLES `biao01` WRITE;
10. UNLOCK TABLES;
11. DROP TABLE IF EXISTS `biao02`;
12. CREATE TABLE `biao02` (
13.   `id` int( 4) NOT NULL,
14.   `name` varchar( 8) DEFAULT NULL,
15.   PRIMARY KEY ( `id` )
16. ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

[Top](#)

17. ...

注意：若数据库都使用MyISAM存储引擎，可以采用冷备份的方式，直接复制对应的数据库目录即可；恢复时重新复制回来就行。

2) 只备份指定的某一个库

将userdb库备份为userdb.sql文件：

```
01. [root@dbsvr1 ~] # mysqldump -u root -p userdb > userdb.sql
02. Enter password: //验证口令
```

查看备份文件userdb.sql的部分内容：

```
01. [root@dbsvr1 ~] # grep -vE '^/|^$' /root/userdb.sql
02. DROP TABLE IF EXISTS `stu_info`;
03. CREATE TABLE `stu_info` (
04.   `name` varchar(12) NOT NULL,
05.   `gender` enum('boy','girl') DEFAULT 'boy',
06.   `age` int(3) NOT NULL
07. ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
08. LOCK TABLES `stu_info` WRITE;
09. ...
```

3) 同时备份指定的多个库

同时备份mysql、userdb库，保存为mysql+userdb.sql文件：

```
01. [root@dbsvr1 ~] # mysqldump -u root -p -B mysql userdb > mysql+test+userdb.sql
02. Enter password: //验证口令
```

查看备份文件userdb.sql的部分内容：

```
01. [root@dbsvr1 ~] # grep '^CREATE DATA' /root/mysql+userdb.sql
02. CREATE DATABASE /*! 32312 IF NOT EXISTS*/ `mysql` /*! 40100 DEFAULT CHARACTER S
03. CREATE DATABASE /*! 32312 IF NOT EXISTS*/ `userdb` /*! 40100 DEFAULT CHARACTER :
```

[Top](#)

步骤二：使用mysql命令从备份中恢复数据库、表

以恢复userdb库为例，可参考下列操作。通常不建议直接覆盖旧库，而是采用建立新库并导入逻辑备份的方式执行恢复，待新库正常后即可废弃或删除旧库。

1) 创建名为userdb2的新库

```
01. mysql> CREATE DATABASE userdb2;
02. Query OK, 1 row affected (0.00 sec)
```

2) 导入备份文件，在新库中重建表及数据

```
01. [root@dbsvr1 ~]# mysql -u root -p userdb2 < /root/userdb.sql
02. Enter password: //验证口令
```

3) 确认新库正常，启用新库

```
01. mysql> USE userdb2; //切换到新库
02. Reading table information for completion of table and column names
03. You can turn off this feature to get a quicker startup with -A
04.
05. Database changed
06. mysql> SELECT sn,username,uid,gid,homedir //查询数据，确认可用
07. -> FROM userlist LIMIT 10;
08. +---+-----+---+---+-----+
09. | sn | username | uid | gid | homedir |
10. +---+-----+---+---+-----+
11. | 1 | root | 0 | 0 | /root |
12. | 2 | bin | 1 | 1 | /bin |
13. | 3 | daemon | 2 | 2 | /sbin |
14. | 4 | adm | 3 | 4 | /var/adm |
15. | 5 | lp | 4 | 7 | /var/spool/lpd |
16. | 6 | sync | 5 | 0 | /sbin |
17. | 7 | shutdown | 6 | 0 | /sbin |
18. | 8 | halt | 7 | 0 | /sbin |
19. | 9 | mail | 8 | 12 | /var/spool/mail |
20. | 10 | operator | 11 | 0 | /root |
21. +---+-----+---+---+-----+
22. 10 rows in set (0.00 sec)
```

[Top](#)

4) 废弃或删除旧库

```
01.  my sql> DROP DATABASE userdb;
02.  Query OK, 2 rows affected ( 0.09 sec)
```

2 使用binlog日志

2.1 问题

利用binlog恢复库表，要求如下：

- 启用binlog日志
- 创建db1库tb1表，插入3条记录
- 删除tb1表中刚插入的3条记录
- 使用mysqlbinlog恢复删除的3条记录

2.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：启用binlog日志

1) 调整/etc/my.cnf配置，并重启服务

```
01.  [ root@dbsvr1 ~] # vim /etc/my.cnf
02.  [ my sql d]
03.  ... ..
04.  log- bin- index= my sql- bin           //启用二进制日志，并指定前缀
05.  server_id=1
06.  binlog_ format= STATEMENT
07.  //在My sql5.7中，binlog日志格式默认为ROW，但它不记录sql语句上下文相关信息。需要
08.  ... ..
09.  [ root@dbsvr1 ~] # systemctl restart mysql. service
```

2) 确认binlog日志文件

新启用binlog后，每次启动MySQL服务都会新生成一份日志文件：

```
01.  [ root@dbsvr1 ~] # ls /var/lib/mysql/mysql- bin. *
02.  /var/lib/mysql/mysql- bin.000001 /var/lib/mysql/mysql- bin.index
```

[Top](#)

其中mysql-bin.index文件记录了当前保持的二进制文件列表：

```

01. [ root@db1 ~] # cat /var/lib/mysql/mysql-bin.index
02. ./mysql-bin.000001

```

重启MySQL服务程序，或者执行SQL操作“FLUSH LOGS;”，会生成一份新的日志：

```

01. [ root@db1 ~] # ls /var/lib/mysql/mysql-bin.*
02. /var/lib/mysql/mysql-bin.000001 /var/lib/mysql/mysql-bin.index
03. /var/lib/mysql/mysql-bin.000002
04.
05. [ root@db1 ~] # cat /var/lib/mysql/mysql-bin.index
06. ./mysql-bin.000001
07. ./mysql-bin.000002

```

步骤二：利用binlog日志重做数据库操作

1) 执行数据库表添加操作

创建db1库tb1表，表结构自定义：

```

01. mysql> CREATE DATABASE db1;
02. Query OK, 1 row affected (0.05 sec)
03.
04. mysql> USE db1;
05. Database changed
06. mysql> CREATE TABLE tb1(
07.     -> id int(4) NOT NULL, name varchar(24)
08.     -> );
09. Query OK, 0 rows affected (0.28 sec)

```

插入3条表记录：

```

01. mysql> INSERT INTO tb1 VALUES
02.     -> (1, 'Jack'),
03.     -> (2, 'Kenth'),
04.     -> (3, 'Bob');
05. Query OK, 3 rows affected (0.12 sec)
06. Records: 3 Duplicates: 0 Warnings: 0

```

[Top](#)

确认插入的表记录数据：

```

01.  my sql> SELECT * FROM tb1;
02.  +---+-----+
03.  | id | name |
04.  +---+-----+
05.  | 1 | Jack |
06.  | 2 | Kenthy |
07.  | 3 | Bob |
08.  +---+-----+
09.  3 rows in set (0.00 sec)

```

2) 删除前一步添加的3条表记录

执行删除所有表记录操作：

```

01.  my sql> DELETE FROM tb1;
02.  Query OK, 3 rows affected (0.09 sec)

```

确认删除结果：

```

01.  my sql> SELECT * FROM tb1;
02.  Empty set (0.00 sec)

```

步骤三：通过binlog日志恢复表记录

binlog会记录所有的数据库、表更改操作，所以可在必要的时候重新执行以前做过的一部分数据操作，但对于启用binlog之前已经存在的库、表数据将不适用。

根据上述“恢复被删除的3条表记录”的需求，应通过mysqlbinlog工具查看相关日志文件，找到删除这些表记录的时间点，只要恢复此前的SQL操作（主要是插入那3条记录的操作）即可。

1) 查看mysql-bin.000002日志内容

```

01.  [root@dbsvr1 ~]# mysqlbinlog /var/lib/mysql/mysql-bin.000002
02.  /*!50530 SET @@SESSION.PSEUDO_SLAVE_MODE=1*/;
03.  /*!50003 SET @@OLD_COMPLETION_TYPE=@@COMPLETION_TYPE,COMPLETION_TYPE=0*/;
04.  DELIMITER /*!*/;
05.  # at 4 Top
06.  #170412 12:05:32 server id 1 end_log_pos 123 CRC32 0x6d8c069c Start: binlog v 4, serve
07.  # Warning: this binlog is either in use or was not closed properly.

```

```

08. ROLLBACK/*! */;
09. BINLOG '
10. jKftWA8BAAAAdwAAAHsAAAAABAAQANS43LjE3LWxvZwAAAAAAAAAAAAAAAAAAAA
11. AAAAAAAAAAAAAAAAAAAACmp+1YEzgNAAGAEgAEBAQEEgAAXwAEGggAAAAICAgCAAAAC
12. AZwGjGO=
13. '/*! */;
14. # at 123
15. #170412 12: 05: 32 server id 1 end_log_pos 154 CRC32 0x17f50164 Previous GTIDs
16. # [ empty ]
17. # at 154
18. #170412 12: 05: 59 server id 1 end_log_pos 219 CRC32 0x4ba5a976 Anonymous_GTID last
19. SET @@SESSION.GTID_NEXT= 'ANONYMOUS'/*! */;
20. # at 219
21. #170412 12: 05: 59 server id 1 end_log_pos 310 CRC32 0x5b66ae13 Query thread_id=3
22. SET TIMESTAMP=1491969959/*! */;
23. SET @@session.pseudo_thread_id=3/*! */;
24. SET @@session.foreign_key_checks=1, @@session.sql_auto_is_null=0, @@session.unique_cl
25. SET @@session.sql_mode=1436549152/*! */;
26. SET @@session.auto_increment_increment=1, @@session.auto_increment_offset=1/*! */;
27. /*! \C utf8 *//*! */;
28. SET @@session.character_set_client=33, @@session.collation_connection=33, @@session.col
29. SET @@session.lc_time_names=0/*! */;
30. SET @@session.collation_database=DEFAULT/*! */;
31. CREATE DATABASE db1
32. /*! */;
33. # at 310
34. #170412 12: 06: 23 server id 1 end_log_pos 375 CRC32 0x2967cc28 Anonymous_GTID last
35. SET @@SESSION.GTID_NEXT= 'ANONYMOUS'/*! */;
36. # at 375
37. #170412 12: 06: 23 server id 1 end_log_pos 502 CRC32 0x5de09aae Query thread_id=3
38. use `db1`/*! */;
39. SET TIMESTAMP=1491969983/*! */;
40. CREATE TABLE tb1(
41. id int( 4) NOT NULL, name varchar( 24)
42. )
43. /*! */;
44. # at 502
45. #170412 12: 06: 55 server id 1 end_log_pos 567 CRC32 0x0b8cd418 Anonymous_GTID last
46. SET @@SESSION.GTID_NEXT= 'ANONYMOUS'/*! */;
47. # at 567
48. #170412 12: 06: 55 server id 1 end_log_pos 644 CRC32 0x7e8f2fa0 Query thread_id=3

```

[Top](#)

```

49.  SET TIMESTAMP=1491970015/*! */;
50.  BEGIN
51.  /*! */;
52.  # at 644
53.  #170412 12: 06: 55 server id 1 end_log_pos 772 CRC32 0x4e3f 728e Query thread_id=3
54.  SET TIMESTAMP=1491970015/*! */;
55.  INSERT INTO tb1 VALUES( 1,'Jack'),( 2,'Kenthly'),( 3,'Bob')
56.  /*! */;
57.  # at 772
58.  #170412 12: 06: 55 server id 1 end_log_pos 803 CRC32 0x6138b21f Xid = 10
59.  //确认事务的时间点
60.  COMMIT/*! */;
61.  # at 803
62.  #170412 12: 07: 24 server id 1 end_log_pos 868 CRC32 0xbef 3f 472 Anonymous_GTID last_
63.  SET @@SESSION.GTID_NEXT = 'ANONYMOUS'/*! */;
64.  # at 868
65.  #170412 12: 07: 24 server id 1 end_log_pos 945 CRC32 0x5684e92c Query thread_id=3
66.  SET TIMESTAMP=1491970044/*! */;
67.  BEGIN
68.  /*! */;
69.  # at 945
70.  #170412 12: 07: 24 server id 1 end_log_pos 1032 CRC32 0x4c1c75fc Query thread_id=
71.  SET TIMESTAMP=1491970044/*! */;
72.  DELETE FROM tb1
73.  /*! */;
74.  # at 1032
75.  #170412 12: 07: 24 server id 1 end_log_pos 1063 CRC32 0xccf 549b2 Xid = 12
76.  COMMIT/*! */;
77.  SET @@SESSION.GTID_NEXT = 'AUTOMATIC' /* added by mysqlbinlog */ /*! */;
78.  DELIMITER ;
79.  # End of log file
80.  /*! 50003 SET COMPLETION_TYPE=@OLD_COMPLETION_TYPE*/;
81.  /*! 50530 SET @@SESSION.PSEUDO_SLAVE_MODE=0*/;

```

2) 执行指定Pos节点范围内的sql命令恢复数据

根据上述日志分析，只要恢复从2014.01.12 20:12:14到2014.01.12 20:13:50之间的操作即可。可通过mysqlbinlog指定时间范围输出，结合管道交给mysql命令执行导入重做：

[Top](#)

```

01.  [root@dbsvr1 ~]# mysqlbinlog \
02.  -- start-datetime="2017- 04- 12 12: 06: 55" \

```



```

03.      -- stop-datetime="2017-04-12 12:07:23" \
04.      /var/lib/mysql/mysql-bin.000002 | mysql -u root -p
05.      Enter password:                //验证口令

```

3) 确认恢复结果

```

01.      mysql> SELECT * FROM db1.tb1;
02.      +----+-----+
03.      | id | name |
04.      +----+-----+
05.      |  1 | Jack |
06.      |  2 | Kenthy |
07.      |  3 | Bob |
08.      +----+-----+
09.      3 rows in set (0.00 sec)

```

3 innobackupex备份工具

3.1 问题

- 安装percona软件包
- innobackupex完整备份、增量备份操作。
- 恢复数据

1.

3.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：安装XtraBackup软件包

1) 了解软件包描述信息

```

01.      [root@dbsvr1 pub] # rpm -qpi percona-xtrabackup-24.2.4.6-2.el7.x86_64.rpm
02.      Name       : percona-xtrabackup-24
03.      Version    : 2.4.6
04.      Release    : 2.el7
05.      Architecture: x86_64
06.      Install Date: (not installed)
07.      Group      : Applications/Databases
08.      Size       : 32416340
09.      License    : GPLv2

```

[Top](#)

10. Signature : DSA/SHA1, 2017年02月27日 星期一 20时28分17秒, Key ID 1c4cbdcdd2ef d2c
11. Source RPM : percona-xtrabackup-24.2.4.6-2.el7.src.rpm
12. Build Date : 2017年02月27日 星期一 20时27分21秒
13. Build Host : vps-centos7-x64-01.ci.percona.com
14. Relocations : (not relocatable)
15. URL : http://www.percona.com/software/percona-xtrabackup
16. Summary : XtraBackup online backup for MySQL / InnoDB
17. Description :
18. Percona XtraBackup is OpenSource online (non- blockable) backup solution for InnoDB and

2) 安装依赖包perl-DBD-MySQL perl-Digest-MD5 libev

使用RHEL 7自带的即可 , yum方式安装 :

01. [root@dbvr1 pub] # yum -y install perl-DBD-MySQL perl-Digest-MD5
02. libev使用网上找的rpm包 libev-4.15-1.el6.rf.x86_64.rpm //该包由讲师提供
03. [root@dbvr1 pub] # rpm -ivh libev-4.15-1.el6.rf.x86_64.rpm

如果未安装这些依赖包 , 则直接安装percona-xtrabackup时会报错 :

[代码](#)

3) 安装percona-xtrabackup

01. [root@dbvr1 pub] # rpm -ivh percona-xtrabackup-*.rpm
02. 警告 : percona-xtrabackup-24.2.4.6-2.el7.x86_64.rpm: 头V4 DSA/SHA1 Signature, 密钥 |
03. 准备中... ##### [100%]
04. 正在升级/安装...
05. 1: percona-xtrabackup-24.2.4.6-2.el7##### [33%]
06. 2: percona-xtrabackup-test-24.2.4.6-##### [67%]
07. 3: percona-xtrabackup-24-debuginfo-2##### [100%]

4) 确认安装的主要程序/脚本

01. [root@dbvr1 pub] # rpm -ql percona-xtrabackup-24.2.4.6-2.el7.x86_64
02. /usr/bin/innobackupex
03. /usr/bin/xbcloud
04. /usr/bin/xbcloud_osenv

[Top](#)

05. `/usr/bin/xbcrypt`
06. `/usr/bin/xbstream`
07. `/usr/bin/xtrabackup`
08. `/usr/share/doc/percona-xtrabackup-24.2.4.6`
09. `/usr/share/doc/percona-xtrabackup-24.2.4.6/COPYING`
10. `/usr/share/man/man1/innobackupex.1.gz`
11. `/usr/share/man/man1/xbcrypt.1.gz`
12. `/usr/share/man/man1/xbstream.1.gz`
13. `/usr/share/man/man1/xtrabackup.1.gz`

步骤二：innobackupex完整备份、增量备份操作

--host 主机名

--port 3306

--user 用户名

--password 密码

--databases="库名"

--databases="库1 库2"

--databases="库.表"

--no-timestamp 不用日期命名备份文件存储的子目录，使用备份的数据库名做备份目录名

--no-timestmap 不使用日期命名备份目录名

1) 做一个完整备份

默认情况下，备份文件存储的子目录会用日期命名，

innobackupex作为客户端工具，以mysql协议连入mysqld，将数据备份到/backup文件夹：

01. `[root@dbvr1 ~]# innobackupex --user=root --password=1234567 /backup/mysql --no-t`
02. `170425 11:05:44 innobackupex: Starting the backup operation`
- 03.
04. `IMPORTANT: Please check that the backup run completes successfully.`
05. `At the end of a successful backup run innobackupex`
06. `prints "completed OK! "`
- 07.
08. `Unrecognized character '\x01' marked by <- HERE after <- HERE near column 1 at - line 1`
09. `170425 11:05:45 Connecting to MySQL server host: localhost, user: root, password: set,`
10. `Using server version 5.7.17`
11. `innobackupex version 2.4.6 based on MySQL server 5.7.13 Linux (x86_64) (revision id: 8ec`
12. `xtrabackup: uses posix_fadvise().`
13. `xtrabackup: cd to /var/lib/mysql`

[Top](#)

```

14. xtrabackup: open files limit requested 0, set to 1024
15. xtrabackup: using the following InnoDB configuration:
16. xtrabackup: innodb_data_home_dir = .
17. xtrabackup: innodb_data_file_path = ibdata1:12M:autoextend
18. xtrabackup: innodb_log_group_home_dir = ./
19. xtrabackup: innodb_log_files_in_group = 2
20. xtrabackup: innodb_log_file_size = 50331648
21. InnoDB: Number of pools: 1
22. 170425 11:05:45 >> log scanned up to ( 2543893)
23. xtrabackup: Generating a list of tablespaces
24. InnoDB: Allocated tablespace ID 2 for mysql/plugin, old maximum was 0
25. 170425 11:05:45 [ 01] Copying ./ibdata1 to /backup/ibdata1
26. 170425 11:05:45 [ 01] ...done
27. 170425 11:05:46 [ 01] Copying ./mysql/plugin.ibd to /backup/mysql/plugin.ibd
28. 170425 11:05:46 [ 01] ...done
29. 170425 11:05:46 [ 01] Copying ./mysql/servers.ibd to /backup/mysql/servers.ibd
30. 170425 11:05:46 [ 01] ...done
31. 170425 11:05:46 [ 01] Copying ./mysql/help_topic.ibd to /backup/mysql/help_topic.ibd
32. 170425 11:05:46 [ 01] ...done
33. 170425 11:05:46 >> log scanned up to ( 2543893)
34. ...
35. 170425 11:06:00 [ 01] Copying ./sys/x@0024waits_global_by_latency.frm to /backup/sys
36. 170425 11:06:00 [ 01] ...done
37. 170425 11:06:00 [ 01] Copying ./sys/session_ssl_status.frm to /backup/sys/session_ssl_s
38. 170425 11:06:00 [ 01] ...done
39. 170425 11:06:00 [ 01] Copying ./db1/db.opt to /backup/db1/db.opt
40. 170425 11:06:00 [ 01] ...done
41. 170425 11:06:00 [ 01] Copying ./db1/tb1.frm to /backup/db1/tb1.frm
42. 170425 11:06:00 [ 01] ...done
43. 170425 11:06:00 Finished backing up non- InnoDB tables and files
44. 170425 11:06:00 Executing FLUSH NO_WRITE_TO_BINLOG ENGINE LOGS...
45. xtrabackup: The latest check point ( for incremental ): '2543884'
46. xtrabackup: Stopping log copying thread.
47. 170425 11:06:00 >> log scanned up to ( 2543893)
48.
49. 170425 11:06:00 Executing UNLOCK TABLES
50. 170425 11:06:00 All tables unlocked
51. 170425 11:06:00 [ 00] Copying ib_buffer_pool to /backup/ib_buffer_pool
52. 170425 11:06:00 [ 00] ...done
53. 170425 11:06:00 Backup created in directory '/backup/'
54. 170425 11:06:00 [ 00] Writing backup-my.cnf

```

[Top](#)

55. 170425 11:06:00 [00] ...done
56. 170425 11:06:00 [00] Writing xtrabackup_info
57. 170425 11:06:00 [00] ...done
58. xtrabackup: Transaction log of lsn (2543884) to (2543893) was copied.
59. 170425 11:06:01 completed OK

确认备份好的文件数据：

01. [root@dbsvr1 ~] #ls /backup/
02. backup- my.cnf ib_buffer_pool mysql sys xtrabackup_info
03. db1 ibdata1 performance_schema xtrabackup_checkpoints xtrabackup_logfile

2) 做一个增量备份 (基于前一步的完整备份)

随意做一些新增或更改库表的操作，比如在db1库中新建一个mytb的表：

01. mysql> USE db1;
02. Database changed
03. mysql> CREATE TABLE mytb(id int(4), name varchar(24));
04. Query OK, 0 rows affected (0.38 sec)
05. mysql> INSERT INTO tb1 VALUES
06. -> (1,'bon'),
07. -> (2,'bo'),
08. Query OK, 2 rows affected (0.12 sec)
09. Records: 2 Duplicates: 0 Warnings: 0
10. mysql> SELECT * FROM tb1;
11. +-----+-----+
12. | id | name |
13. +-----+-----+
14. | 1 | bob |
15. | 2 | bo |
16. +-----+-----+
17. 2 rows in set (0.00 sec)

以前一次保存到/backup的完整备份为基础，做一个增量备份，保存到/incr01/，指定增量备份参照的基本目录 (完整备份目录) 需要用到选项--incremental-basedir。相关操作如下：

[Top](#)

01. [root@dbsvr1 ~] # innobackupex --user=root --password=12345678 --incremental /incr01

```

02. 170425 11: 30: 14 innobackupex: Starting the backup operation
03.
04. IMPORTANT: Please check that the backup run completes successfully .
05.     At the end of a successful backup run innobackupex
06.     prints "completed OK! ".
07.
08. Unrecognized character '\x01' marked by <- - HERE after <- - HERE near column 1 at - line 1
09. 170425 11: 30: 14 Connecting to MySQL server host: localhost, user: root, password: set,
10. Using server version 5.7.17
11. innobackupex version 2.4.6 based on MySQL server 5.7.13 Linux ( x86_64 ) ( revision id: 8ec
12. incremental backup from 2543884 is enabled.
13. xtrabackup: uses posix_fadvise( ) .
14. xtrabackup: cd to /var/lib/mysql
15. xtrabackup: open files limit requested 0, set to 1024
16. xtrabackup: using the following InnoDB configuration:
17. xtrabackup: innodb_data_home_dir = .
18. xtrabackup: innodb_data_file_path = ibdata1: 12M:autoextend
19. xtrabackup: innodb_log_group_home_dir = ./
20. xtrabackup: innodb_log_files_in_group = 2
21. xtrabackup: innodb_log_file_size = 50331648
22. InnoDB: Number of pools: 1
23. 170425 11: 30: 14 >> log scanned up to ( 2549933)
24. xtrabackup: Generating a list of tablespaces
25. InnoDB: Allocated tablespace ID 2 for mysql/plugin, old maximum was 0
26. xtrabackup: using the full scan for incremental backup
27. 170425 11: 30: 15 [ 01] Copying ./ibdata1 to /incr01/ibdata1.delta
28. 170425 11: 30: 15 [ 01]     ...done
29. 170425 11: 30: 15 >> log scanned up to ( 2549933)
30. 170425 11: 30: 15 [ 01] Copying ./mysql/plugin.ibd to /incr01/mysql/plugin.ibd.delta
31. 170425 11: 30: 15 [ 01]     ...done
32. ... ..
33. 170425 11: 30: 35 Executing UNLOCK TABLES
34. 170425 11: 30: 35 All tables unlocked
35. 170425 11: 30: 35 [ 00] Copying ib_buffer_pool to /incr01/ib_buffer_pool
36. 170425 11: 30: 35 [ 00]     ...done
37. 170425 11: 30: 35 Backup created in directory '/incr01/'
38. 170425 11: 30: 35 [ 00] Writing backup-my.cnf
39. 170425 11: 30: 35 [ 00]     ...done
40. 170425 11: 30: 35 [ 00] Writing xtrabackup_info
41. 170425 11: 30: 35 [ 00]     ...done
42. xtrabackup: Transaction log of lsn ( 2549924) to ( 2549933) was copied.

```

[Top](#)

43. 170425 11: 30: 35 completed OK!

确认备份好的文件数据：

```
01. [root@dbsvr1 ~] # ls /incr01/
02. backup- my.cnf ib_buffer_pool ibdata1.meta performance_schema xtrabackup_checkpoi
03. db1          ibdata1.delta mysql      sys
```

对比完整备份、增量备份的大小：

```
01. [root@dbsvr1 ~] # du - sh /backup/ /incr01/
02. 142M /backup/ //完整备份的大小
03. 3.5M /incr01/ //增量备份的大小
```

步骤三：恢复数据

通过XtraBackup工具备份的数据库目录，若要恢复到另一个MySQL服务器，需要先做一个 “--apply-log --redo-only” 的准备操作。

1) 准备恢复 “完整备份”

完成准备以后，最终/backup可用来重建MySQL服务器。这种情况下，需要先做一个 “--apply-log --redo-only” 的准备操作，以确保数据一致性：

```
01. [root@dbsvr1 ~] #innobackupex --user=root --password=12345678 --apply-log --redo-
02. 170425 11: 42: 19 innobackupex: Starting the apply-log operation
03.
04. IMPORTANT: Please check that the apply-log run completes successfully.
05.           At the end of a successful apply-log run innobackupex
06.           prints "completed OK!".
07.
08. innobackupex version 2.4.6 based on MySQL server 5.7.13 Linux (x86_64) (revision id: 8ec
09. xtrabackup: cd to /backup/
10. xtrabackup: This target seems to be already prepared.
11. InnoDB: Number of pools: 1
12. xtrabackup: notice: xtrabackup_logfile was already used to '--prepare'.
13. xtrabackup: using the following InnoDB configuration for recovery:
14. xtrabackup: innodb_data_home_dir = .
15. xtrabackup: innodb_data_file_path = ibdata1: 12M:autoextend
```

[Top](#)

16. xtrabackup: innodb_log_group_home_dir = .
17. xtrabackup: innodb_log_files_in_group = 2
18. xtrabackup: innodb_log_file_size = 50331648
19. xtrabackup: using the following InnoDB configuration for recovery:
20. xtrabackup: innodb_data_home_dir = .
21. xtrabackup: innodb_data_file_path = ibdata1:12M:autoextend
22. xtrabackup: innodb_log_group_home_dir = .
23. xtrabackup: innodb_log_files_in_group = 2
24. xtrabackup: innodb_log_file_size = 50331648
25. xtrabackup: Starting InnoDB instance for recovery.
26. xtrabackup: Using 104857600 bytes for buffer pool (set by --use-memory parameter)
27. InnoDB: PUNCH HOLE support available
28. InnoDB: Mutexes and rw_locks use GCC atomic builtins
29. InnoDB: Uses event mutexes
30. InnoDB: GCC builtin __atomic_thread_fence() is used for memory barrier
31. InnoDB: Compressed tables use zlib 1.2.7
32. InnoDB: Number of pools: 1
33. InnoDB: Not using CPU crc32 instructions
34. InnoDB: Initializing buffer pool, total size = 100M, instances = 1, chunk size = 100M
35. InnoDB: Completed initialization of buffer pool
36. InnoDB: page_cleaner coordinator priority: - 20
37. InnoDB: Highest supported file format is Barracuda.
- 38.
39. xtrabackup: starting shutdown with innodb_fast_shutdown = 1
40. InnoDB: Starting shutdown...
41. InnoDB: Shutdown completed; log sequence number 2544177
42. InnoDB: Number of pools: 1
43. 170425 11: 42: 20 completed OK!

准备恢复 “增量备份”

01. [root@dbvr1 ~] #innobackupex --user=root --password=12345678 --apply-log --redo-
02. 170425 11: 42: 55 innobackupex: Starting the apply-log operation
- 03.
04. IMPORTANT: Please check that the apply-log run completes successfully.
05. At the end of a successful apply-log run innobackupex
06. prints "completed OK! ".
- 07.
08. innobackupex version 2.4.6 based on MySQL server 5.7.13 Linux (x86_64) (revision id: 8ec

[Top](#)


```

09. incremental backup from 2543884 is enabled.
10. xtrabackup: cd to /backup/
11. xtrabackup: This target seems to be already prepared with --apply-log-only.
12. InnoDB: Number of pools: 1
13. xtrabackup: xtrabackup_logfile detected: size=8388608, start_lsn=(2549924)
14. xtrabackup: using the following InnoDB configuration for recovery:
15. xtrabackup: innodb_data_home_dir = .
16. xtrabackup: innodb_data_file_path = ibdata1:12M:autoextend
17. xtrabackup: innodb_log_group_home_dir = /incr01/
18. xtrabackup: innodb_log_files_in_group = 1
19. xtrabackup: innodb_log_file_size = 8388608
20. xtrabackup: Generating a list of tablespaces
21. InnoDB: Allocated tablespace ID 2 for mysql/plugin, old maximum was 0
22. xtrabackup: page size for /incr01/ibdata1 delta is 16384 bytes
23. Applying /incr01/ibdata1.delta to ./ibdata1..
24. ... ..
25. 170425 11:43:09 [01] Copying /incr01/performance_schema/global_status.frm to ./perf
26. 170425 11:43:09 [01] ...done
27. 170425 11:43:09 [01] Copying /incr01/performance_schema/session_status.frm to ./perf
28. 170425 11:43:09 [01] ...done
29. 170425 11:43:09 [00] Copying /incr01/xtrabackup_info to ./xtrabackup_info
30. 170425 11:43:09 [00] ...done
31. 170425 11:43:10 completed OK!

```

2) 关闭mysql服务，并将/var/lib/mysql/下的文件删除，假设数据被删除。

```

01. [root@dbsvr1 ~]# systemctl stop mysqld
02. [root@dbsvr1 ~]# rm -rf /var/lib/mysql

```

3) 恢复“完整备份+增量备份”

完成准备以后，最终仍然是/backup用来重建MySQL服务器，但这种情况下需提前合并相关增量备份的数据

```

01. [root@dbsvr1 ~]# innobackupex --user=root --password=12345678 --copy-back /backup/
02. ... ..
03. 170425 11:51:39 [01] Copying ./performance_schema/global_status.frm to /var/lib/mysql
04. 170425 11:51:39 [01] ...done
05. 170425 11:51:39 [01] Copying ./performance_schema/session_status.frm to /var/lib/mysql

```

06. 170425 11: 51: 39 [01] ...done
07. 170425 11: 51: 39 [01] Copying ./ib_buffer_pool to /var/lib/mysql/ib_buffer_pool
08. 170425 11: 51: 39 [01] ...done
09. 170425 11: 51: 39 [01] Copying ./ibtmp1 to /var/lib/mysql/ibtmp1
10. 170425 11: 51: 39 [01] ...done
11. 170425 11: 51: 39 [01] Copying ./xtrabackup_info to /var/lib/mysql/xtrabackup_info
12. 170425 11: 51: 39 [01] ...done
13. 170425 11: 51: 39 completed OK!

4) 修改/var/lib/mysql/下文件属主与属组,查看数据 :

恢复后 , /var/lib/mysql下文件属组与属主皆为root , 需要更改为mysql

01. [root@dbsvr1 ~] #chown - R mysql:mysql /var/lib/mysql
02. [root@dbsvr1 ~] #systemctl start mysqld.service
03. [root@dbsvr1 ~] #mysql -uroot -p12345678 -e "select * from db1.tb1"
04. mysql: [Warning] Using a password on the command line interface can be insecure.
05. +-----+-----+
06. | id | name |
07. +-----+-----+
08. | 1 | bob |
09. | 2 | bo |
10. +-----+-----+

[Top](#)