

# 第1章 备份恢复

## 1.备份恢复的职责

1. 备份、恢复策略的设计。
2. 备份周期、备份工具、备份方式、恢复方式全部流程化
3. 日常备份检查
4. 日志、备份内容
5. 定期的恢复演练
6. 数据故障时，利用现有的资源，快速恢复
7. 数据迁移、升级。

# 第2章 备份工具介绍

## 1.逻辑备份

- 1 mysqldump / source \*\*\*\*\*
- 2 mysqlbinlog / source
- 3 mydumper / myloader
- 4 select into outfile / load data infile
- 5 binlog2sql
- 6 myflashback

## 2.物理备份

- 1 Percona Xtrabackup (PXB,XBK) \*\*\*\*\*

## 3.选型

- 1 100G 以内： 逻辑
- 2 100G 以上： 物理

# 第3章 mysqldump工具使用

## 1.介绍

- 1 mdp数据逻辑备份工具。(Create database\ create table \ insert)
- 2 MySQL 自带的客户端命令。可以实现远程和本地备份。

## 2.参数

### 2.1 连接参数

- 1 -u
- 2 -p
- 3 -S
- 4 -h
- 5 -P

### 2.2 备份参数

- 1 # -A 全备
- 2 mkdir /data/backup
- 3 mysqldump -uroot -p123 -A >/data/backup/full.sql
- 4
- 5 # -B 单库或多库
- 6 mysqldump -uroot -p123 -B world gtdb test >/data/backup/db.sql
- 7

```
8 # 备份单表或多表
9 mysqldump -uroot -p123 world t1 country >/data/backup/tab.sql
10
11 # --master-data=2
12 1.自动记录备份时的binlog信息（注释）
13 2.自动锁定所有表，自动解锁（global read lock）。最好配合--single-transaction
    参数，减少锁表时间。
14 mysqldump -uroot -p123 -A --master-data=2 >/data/backup/full.sql
15
16 # --single-transaction
17 对于InnoDB表，开启独立事务，通过快照备份表数据，不锁表备份，可以理解为热备。
18 mysqldump -uroot -p123 -A --master-data=2 --single-transaction
    >/data/backup/full.sql
19
20 # --max_allowed_packet=64M 最大允许的数据包大小
21 mysqldump -uroot -p123 -A --master-data=2 --single-transaction --
    max_allowed_packet=64M >/data/backup/full.sql
22
23 # -R -E --triggers 备份特殊对象使用
24 mysqldump -uroot -p123 -A --master-data=2 --single-transaction --
    max_allowed_packet=64M -R -E --triggers >/data/backup/full.sql
25
26 # 按日期备份定义文件名
27 mysqldump -uroot -p -A --master-data=2 --single-transaction --
    max_allowed_packet=64M -R -E --triggers >/data/backup/full_`date
    +%F`.sql
```

## 3.故障恢复演练(mysql dump+binlog)

### 3.1 模拟环境

```
1 create database mdp charset utf8mb4;
2 use mdp
3 create table t1(id int);
4 insert into t1 values(1),(2),(3);
5 commit;
```

## 3.2 模拟周一23:00 全备

备份命令：

```
1 mysqldump -uroot -p -A --master-data=2 --single-transaction --
  max_allowed_packet=64M -R -E --triggers >/data/backup/full_`date
  +%F`.sql
```

查看GTID相关信息，GTID截取起点

```
1 SET @@GLOBAL.GTID_PURGED='9b52b744-eb82-11ea-986c-000c294983f8:1-6';
```

查看pos号，备份开始时binlog位置点信息

```
1 -- CHANGE MASTER TO MASTER_LOG_FILE='mysql-bin.000006',
  MASTER_LOG_POS=1507;
```

## 3.3 模拟周二白天数据变化

```
1 use mdp;
2 create table t2 (id int);
3 insert into t2 values(1),(2),(3);
4 commit;
```

### 3.4 模拟周二下午2点，误删除了mdb核心库

```
1 mysql> drop database mdp;
```

### 3.5 恢复数据

a.恢复全备到周一晚23:00

```
1 # 检查全备:
2 vim /data/backup/full_2020-09-14.sql
3
4 # 查看 GTID相关信息 : GTID截取起点。
5 SET @@GLOBAL.GTID_PURGED='9b52b744-eb82-11ea-986c-000c294983f8:1-6';
6
7 # 查看pos号, 备份开始时binlog位置点信息。
8 -- CHANGE MASTER TO MASTER_LOG_FILE='mysql-bin.000006',
   MASTER_LOG_POS=1507;
```

b.截取日志

```
1 # 起点:
2 mysql-bin.000006 9b52b744-eb82-11ea-986c-000c294983f8:7 或者 mysql-
  bin.000006 pos=1507
3
4 # 终点: 找到drop事件
5 [root@db-51 ~]# mysql -uroot -p123456 -e "show binlog events in
  'mysql-bin.000006'" | grep -B 1 "drop database mdp"
6 mysql: [Warning] Using a password on the command line interface can be
  insecure.
7 mysql-bin.000006          1929      Gtid      6          1994      SET
  @@SESSION.GTID_NEXT= '9b52b744-eb82-11ea-986c-000c294983f8:9'
8 mysql-bin.000006          1994      Query     6          2083      drop database
  mdp
9
10 # 截取日志
11 [root@db-51 ~]# mysqlbinlog --skip-gtids --include-gtids='9b52b744-
  eb82-11ea-986c-000c294983f8:7-8' /data/mysql_3306/logs/mysql-bin.000006
  >/data/backup/bin.sql
```

#### c.恢复

```
1 set sql_log_bin=0;
2 source /data/backup/full_2020-09-14.sql
3 source /data/backup/bin.sql
4 set sql_log_bin=1;
```

#### d.检查数据

```
1 use mdp
2 show tables;
3 select * from t1;
4 select * from t2;
```

## 4.mysqlDump多种备份策略和恢复策略介绍

### 4.1 场景

- 1 100G 全库数据 全库备份 30分钟-40分钟，恢复整库需要5倍时间2.5-3小时之间
- 2 一张表 1G 被误删除了。

### 4.2 备份策略

#### a. mdp full+ binlog 增量备份思路

1. 提取full全备中的故障表数据，恢复数据
- 2 

```
# sed -e '/./{H;$!d;}' -e 'x;/CREATE TABLE `t1`/*!d;q' full.sql>createtable.sql
```
- 3 

```
# grep -i 'INSERT INTO `t1`' full.sql >data.sql
```
- 4 2.binlog中截取全备到误删除t1之间对于这张表的修改

#### b.单库单表备份+binlog 增量思路

1. 恢复单表的备份
2. binlog中截取备份到误删除t1之间对于这张表的修改

### 4.3 模拟故障

#### a.模拟原始数据

- 1 

```
create database oldboy charset utf8mb4;
```
- 2 

```
use oldboy;
```
- 3 

```
create table oldguo (id int);
```
- 4 

```
insert into oldguo values(1),(2),(3);
```
- 5 

```
commit;
```

#### b.周一晚上全库备份

```
1 mysqldump -uroot -p -A --master-data=2 --single-transaction --  
max_allowed_packet=64M -R -E --triggers >/data/backup/full.sql
```

#### c.模拟周二白天的数据变化

```
1 use oldboy ;  
2 insert into oldguo values(11),(22),(33);  
3 commit;  
4 create table oldli(id int);  
5 insert into oldli values(1),(2),(3);  
6 commit;  
7 insert into oldguo values(111),(222),(333);  
8 commit;
```

#### d.模拟周二下午2点，误删除数据库

```
1 drop table oldguo;
```

## 4.4 模拟恢复

#### a.处理全备

```
1 [root@db-51 ~]# sed -n '/CREATE TABLE `oldguo` /,/\;/p'  
/data/backup/full.sql >/data/backup/create.sql  
2 [root@db-51 ~]# grep -i 'INSERT INTO `oldguo`' /data/backup/full.sql  
>/data/backup/insert.sql
```

#### b.binlog的截取

起点：9b52b744-eb82-11ea-986c-000c294983f8:13

```
1 SET @@GLOBAL.GTID_PURGED='9b52b744-eb82-11ea-986c-000c294983f8:1-12';
```



终点: 9b52b744-eb82-11ea-986c-000c294983f8:16

```
1 [root@db-51 ~]# mysql -uroot -p123456 -e "show binlog events in 'mysql-  
bin.000007'" |grep -B 1 'DROP TABLE\ `oldguo`'  
2 mysql: [Warning] Using a password on the command line interface can be  
insecure.  
3 mysql-bin.000007          1799      Gtid      6          1864      SET  
@@SESSION.GTID_NEXT= '9b52b744-eb82-11ea-986c-000c294983f8:17'  
4 mysql-bin.000007          1864      Query     6          1987      use `oldboy`;  
DROP TABLE `oldguo` /* generated by server */
```

gtid范围:

```
1 [root@db-51 ~]# mysqlbinlog --include-gtids='9b52b744-eb82-11ea-986c-  
000c294983f8:13-16' /data/mysql_3306/logs/mysql-bin.000007 |grep -B 16  
'oldguo'|grep "GTID_NEXT"  
2 SET @@SESSION.GTID_NEXT= '9b52b744-eb82-11ea-986c-000c294983f8:13'/*!*/;  
3 SET @@SESSION.GTID_NEXT= '9b52b744-eb82-11ea-986c-000c294983f8:16'/*!*/;
```

截取方法1:

```
1 mysqlbinlog --skip-gtids --include-gtids='9b52b744-eb82-11ea-986c-  
000c294983f8:13-16' --exclude-gtids='9b52b744-eb82-11ea-986c-  
000c294983f8:14-15' /data/mysql_3306/logs/mysql-bin.000007  
>/data/backup/bin.sql
```

截取方法2:

```
1 mysqlbinlog --skip-gtids --include-gtids='9b52b744-eb82-11ea-986c-  
000c294983f8:13','9b52b744-eb82-11ea-986c-000c294983f8:16'  
/data/mysql_3306/logs/mysql-bin.000007 >/data/backup/bin1.sql
```

c.恢复数据

```
1 use oldboy;
2 set sql_log_bin=0;
3 source /data/backup/create.sql
4 source /data/backup/insert.sql
5 commit;
6 source /data/backup/bin.sql
7 set sql_log_bin=1;
```

## 5.mysqlDump实现单库单表备份

设置安全导出文件：

```
1 [root@db-51 ~]# vim /etc/my.cnf
2 [mysqld]
3 secure_file_priv=/tmp
4
5 [root@db-51 ~]# systemctl restart mysqld
```

构造备份语句脚本：

```
1 [root@db-51 ~]# mkdir -p /data/backup/single_bak
2 [root@db-51 ~]# mysql -uroot -p123
3 mysql> select concat("mysqldump -uroot -p123 -A --master-data=2 --
  single-transaction --max_allowed_packet=64M -R -E --triggers
  ",table_schema," ",table_name,"
  >/data/backup/single_bak/",table_schema,"_",table_name,".sql")
4 from information_schema.tables
5 where table_schema not in
  ('sys','information_schema','performance_schema')
6 into outfile '/tmp/single_bak.sh';
7 [root@db-51 ~]# sh /tmp/single_bak.sh &>/tmp/bak.log
```

# 第4章 Xtrabackup工具使用

## 1.介绍

- 1 percona公司研发
- 2 xtrabackup --》C C++
- 3 innobackupex --》perl语言
- 4 8.0之前, 2.4.x
- 5 8.0之后, 8.0
- 6 物理备份工具, 类似于cp文件。支持: 全备和增量备份

## 2.安装

### 2.1 安装依赖包

- 1 `wget -O /etc/yum.repos.d/epel.repo http://mirrors.aliyun.com/repo/epel-7.repo`
- 2 `yum -y install perl perl-devel libaio libaio-devel perl-Time-HiRes perl-DBD-MySQL libev`

### 2.2 下载软件并安装

- 1 `wget https://www.percona.com/downloads/XtraBackup/Percona-XtraBackup-2.4.12/binary/redhat/7/x86_64/percona-xtrabackup-24-2.4.12-1.el7.x86_64.rpm`
- 2 `yum -y install percona-xtrabackup-24-2.4.4-1.el7.x86_64.rpm`

## 3.全备

### 3.1 介绍

- 1 拷贝/data/mysql\_3306/data/下的数据文件。
- 2 InnoDB : 热备。拷贝ibdataN,UNDO000N ,ibtmpN ,ibd 。通过截取变化redo。
- 3 非InnoDB: FTWRL, 全局锁。拷贝非INNODB的文件frm\myi\myd\...
- 4 只能本地备份。

## 3.2 实现全备

修改配置文件并重启

```
1 [root@db-51 ~]# vim /etc/my.cnf
2 [client]
3 socket=/tmp/mysql.sock
4 [root@db-51 ~]# systemctl restart mysqld
```

全备命令:

```
1 innobackupex --user=root --password=123456 /data/backup/test
```

查看备份完成的目录:

```
1 [root@db-51 ~]# ll /data/backup/test/2020-09-14_22-06-11/
2 总用量 12348
3 -rw-r----- 1 root root      487 9月  14 22:06 backup-my.cnf
4 drwxr-x--- 2 root root        48 9月  14 22:06 gtdb
5 -rw-r----- 1 root root    10056 9月  14 22:06 ib_buffer_pool
6 -rw-r----- 1 root root 12582912 9月  14 22:06 ibdata1
7 drwxr-x--- 2 root root        52 9月  14 22:06 ku
8 drwxr-x--- 2 root root        52 9月  14 22:06 linux5
9 drwxr-x--- 2 root root        76 9月  14 22:06 mdp
10 drwxr-x--- 2 root root     4096 9月  14 22:06 mysql
11 drwxr-x--- 2 root root        90 9月  14 22:06 oldboy
12 drwxr-x--- 2 root root       134 9月  14 22:06 oldya
13 drwxr-x--- 2 root root     8192 9月  14 22:06 performance_schema
14 drwxr-x--- 2 root root       160 9月  14 22:06 school
```

15	drwxr-x---	2	root	root	8192	9月	14	22:06	sys
16	drwxr-x---	2	root	root	54	9月	14	22:06	test
17	drwxr-x---	2	root	root	144	9月	14	22:06	world
18	-rw-r-----	1	root	root	63	9月	14	22:06	xtrabackup_binlog_info
19	-rw-r-----	1	root	root	117	9月	14	22:06	xtrabackup_checkpoints
20	-rw-r-----	1	root	root	546	9月	14	22:06	xtrabackup_info
21	-rw-r-----	1	root	root	2560	9月	14	22:06	xtrabackup_logfile

目录文件介绍:

```

1 1.xtrabackup_binlog_info
2 记录binlog位置点, 截取binlog起点位置
3
4 2.xtrabackup_checkpoints
5 from_lsn = 0          # 一般增量备份会关注, 一般上次备份的to_lsn的位置
6 to_lsn = 180881595    # CKPT-LSN 最近的内存数据落地到磁盘上的LSN号
7 last_lsn = 180881604 # xtrabackup_logfile LSN
8
9 3.xtrabackup_info
10 备份总览信息
11
12 4.xtrabackup_logfile
13 备份期间产生的redo变化

```

自定义备份目录

```

1 innobackupex --user=root --password=123 --no-timestamp
  /data/backup/xbk/full_`date +%F`

```

### 3.3 全备恢复应用

模拟删除

```
1 pkill mysqld
2 rm -rf /data/mysql_3306/*
```

使用全备恢复数据

a.prepare 准备备份阶段

```
1 innobackupex --apply-log /data/backup/xbk/full
```

b.copy-back 恢复

方法1:

```
1 cp -a /data/backup/test/2020-09-14_22-06-11/* /data/mysql_3306/
2 mkdir /data/mysql_3306/logs/
3 touch /data/mysql_3306/logs/mysql.err
4 chown -R mysql:mysql /data/*
```

方法2:

```
1 innobackupex --copy-back /data/backup/test/2020-09-14_22-06-11/
2 innobackupex --move-back /data/backup/test/2020-09-14_22-06-11/
```

## 4.增量备份功能

### 4.1 介绍

- 1 自带的功能。
- 2 每次增量一般是将最近一次备份作为参照物。
- 3 自动读取参照物cat xtrabackup\_checkpoints中to\_lsn值，与当前CKPT的LSN对比，备份变化过page。
- 4 备份期间新的数据变化，通过redo自动备份。
- 5 恢复数据时，需要把所有需要的增量合并到FULL中。无法通过增量单独恢复数据，依赖与全备。

## 4.2 增量备份演练 (FULL(周日)+inc1(周一)+inc2(周二)+inc3(周三))

### 1.备份前数据准备:

```
1 create database xbk charset utf8mb4;
2 use xbk
3 create table full (id int);
4 insert into full values(1),(2),(3);
5 commit;
```

### 2.模拟周日 23:00 全备

```
1 innobackupex --user=root --password=123 --no-timestamp
/data/backup/full_`date +%F`
```

### 3.模拟周一白天数据变化

```
1 use xbk
2 create table inc1 (id int);
3 insert into inc1 values(1),(2),(3);
4 commit;
```

### 4.模拟周一23:00增量备份

```
1 innobackupex --user=root --password=123 --no-timestamp --incremental
--incremental-basedir=/data/backup/full_2020-09-15
/data/backup/inc1_`date +%F`
```

### 5.模拟周二白天数据变化

```
1 use xbk
2 create table inc2 (id int);
3 insert into inc2 values(1),(2),(3);
4 commit;
```

#### 6.模拟周二23:00增量备份

```
1 innobackupex --user=root --password=123 --no-timestamp --incremental
--incremental-basedir=/data/backup/inc1_2020-09-15
/data/backup/inc2_`date +%F`
```

#### 7.模拟周三白天数据变化

```
1 use xbk
2 create table inc3(id int);
3 insert into inc3 values(1),(2),(3);
4 commit;
```

#### 8.模拟周三23:00增量备份

```
1 innobackupex --user=root --password=123 --no-timestamp --incremental
--incremental-basedir=/data/backup/inc2_2020-09-15
/data/backup/inc3_`date +%F`
```

#### 9.模拟周四白天的数据变化。

```
1 use xbk
2 create table inc4(id int);
3 insert into inc4 values(1),(2),(3);
4 commit;
```

#### 10.周四下午出现数据损坏。如何恢复到误删除之前。



```
1 pkill mysqld
2 rm -rf /data/mysql_3306/*
```

## 11.恢复思路

```
1 1.我们有什么?
2 备份:
3 full+inc1+inc2+inc3
4 binlog:
5 full以来全量的binlog
6
7 2.处理备份
8 需要将inc1\inc2\inc3按顺序依次合并到全备, 并进行prepare.
9 从官方角度: 基础全备和合并所有增量(排除最后一个)都需要此参数
10 原理角度: 使所有备份合并时, LSN必须是连续的
```

## 12.处理base\_full

```
1 innobackupex --apply-log --redo-only /data/backup/full_2020-09-15/
```

## 13.inc1合并到full中, 并且prepare

```
1 cd /data/backup/
2 innobackupex --apply-log --redo-only --incremental-dir=inc1_2020-09-15
  full_2020-09-15
```

## 检验合并结果

```
1 cat full_2020-09-15/xtrabackup_checkpoints |grep "to_lsn"
2 cat inc1_2020-09-15/xtrabackup_checkpoints |grep "to_lsn"
```

## 14.inc2合并到full中, 并且prepare

```
1 cd /data/backup/
2 innobackupex --apply-log --redo-only --incremental-dir=inc2_2020-09-15
  full_2020-09-15
```

#### 检验合并结果

```
1 cat full_2020-09-15/xtrabackup_checkpoints |grep "to_lsn"
2 cat inc2_2020-09-15/xtrabackup_checkpoints |grep "to_lsn"
```

#### 15.inc3合并到full中，并且prepare

```
1 cd /data/backup/
2 innobackupex --apply-log --redo-only --incremental-dir=inc3_2020-09-15
  full_2020-09-15
```

#### 检验合并结果

```
1 cat full_2020-09-15/xtrabackup_checkpoints |grep "to_lsn"
2 cat inc3_2020-09-15/xtrabackup_checkpoints |grep "to_lsn"
```

#### 16.将合并后全备再次prepare

```
1 innobackupex --apply-log /data/backup/full_2020-09-15
```

#### 17.恢复并启动

```
1 innobackupex --copy-back /data/backup/full_2020-09-15
2 mkdir /data/mysql_3306/logs/
3 touch /data/mysql_3306/logs/mysql.err
4 chown -R mysql:mysql /data/*
5 systemctl restart mysqld
```

#### 18.截取周三增量备份后到故障前所有的binlog日志并进行恢复

前提条件：binlog没有被删掉，如果binlog和数据目录放在一起，刚才的操作就把binlog也一起删掉了。

起点：2029

```
1 [root@db-51 ~]# cat /data/backup/inc3_2020-09-15/xtrabackup_binlog_info
2 mysql-bin.000001          2029      9b52b744-eb82-11ea-986c-000c294983f8:1-
  17,
3 cb0fd847-f6e3-11ea-af80-000c294983f8:1-9
```

终点：binlog 结尾 2423

```
1 mysql -uroot -p123456 -e "show binlog events in 'mysql-bin.000001';"
```

## 19.截取命令

```
1 mysqlbinlog --skip-gtids --start-position=2029 /binlog/mysql-bin.000001
>/data/backup/bin.sql
```

## 20.恢复binlog

```
1 mysql -uroot -p123
2 mysql> set sql_log_bin=0;
3 mysql> source /data/backup/bin.sql
4 mysql> set sql_log_bin=1;
```

## 21.xbk恢复完成后，清空所有日志

```
1 mysql> reset master;
```

## 21.立即再做个全备

```
1 innobackupex --user=root --password=123456 --no-timestamp  
/data/backup/full_bak_`date +%F`
```

深圳三  
木男孩  
教师