[**Percona XtraBackup**](https://www.cnblogs.com/kerrycode/p/9236574.html)**使用手册**

目录

[1. XtraBackup介绍 3](#_Toc18499486)

[2. Percona  XtraBackup的特征 4](#_Toc18499487)

[3. XtraBackup完整备份 6](#_Toc18499488)

[3.1 备份的权限 6](#_Toc18499489)

[3.2 完整备份 7](#_Toc18499490)

[4. 如何使用全备恢复还原 11](#_Toc18499491)

[Step 1：Prepare the Backup。 11](#_Toc18499492)

[Step 2：关闭MySQL服务，备份数据目录或删除数据目录 12](#_Toc18499493)

[Step 3: 恢复数据，将备份数据文件拷贝到数据目录 12](#_Toc18499494)

[Step 4: 测试数据的完整性： 13](#_Toc18499495)

[5. XtraBackup增量备份 13](#_Toc18499496)

[6. 增量备份的还原 16](#_Toc18499497)

[Step 1：完整备份 16](#_Toc18499498)

[Step 2：增量备份1 17](#_Toc18499499)

[Step 3：增量备份2: 18](#_Toc18499500)

[Step 1：恢复完全备份 20](#_Toc18499501)

[Step 2：增量备份还原 22](#_Toc18499502)

[Step 3：增量备份还原 22](#_Toc18499503)

[Step 4： 关闭MySQL服务并移除数据目录。 23](#_Toc18499504)

[Step 5： 将备份文件拷贝回MySQL数据目录。 23](#_Toc18499505)

[Step 6： 检查验证数据数据恢复情况。 23](#_Toc18499506)

[7. XtraBackup备份高级选项 24](#_Toc18499507)

[7.1 Compact Backups 24](#_Toc18499508)

[7.2 Partial Backups 25](#_Toc18499509)

[7.2.1准备部分备份（Preparing Partial Backups） 27](#_Toc18499510)

[7.2.2恢复部分备份（Restoring Partial Backups） 27](#_Toc18499511)

[7.3 基于tar格式备份 29](#_Toc18499512)

[8. Innobackupex部分参数介绍 29](#_Toc18499513)

[innobackupex的的部分参数 30](#_Toc18499514)

[9. XtraBackup问题集合 37](#_Toc18499515)

[1：没有开启二进制日志，那么Xtrabackup有什么问题没？ 37](#_Toc18499516)

[2：xtrabackup:  Check 'to\_lsn' of the target and 'from\_lsn' of the incremental. 37](#_Toc18499517)

[3：版本不支持的情况，例如 Percona-XtraBackup-2.3.4 不支持MySQL 5.7.21 39](#_Toc18499518)

# XtraBackup介绍

  XtraBackup是Percona公司的开源项目，用以实现类似Innodb官方的热备份工具InnoDB Hot Backup的功能，它支持在线热备份(备份时不影响数据读写)。

XtraBackup有很多功能和优点：例如支持全备、增量备份、部分备份；支持压缩备份；备份不影响数据读写、事务等，但是也有缺陷不足：例如不支持脱机备份、不支持直接备份到磁带设备、不支持Cloud Back，MyISAM的备份也会阻塞。不过这些小瑕疵不影响XtraBackup成为一款流行的MySQL备份工具。另外，请注意XtraBackup只支Linux平台，不支持Windows平台。

下面是XtraBackup的官方文档的介绍资料：

Percona XtraBackup is the world’s only open-source, free MySQL hot backup software that performs non-blocking backups for InnoDB and XtraDB databases. With Percona XtraBackup, you can achieve the following benefits:

* Backups that complete quickly and reliably
* Uninterrupted transaction processing during backups
* Savings on disk space and network bandwidth
* Automatic backup verification
* Higher uptime due to faster restore time

Percona XtraBackup makes MySQL hot backups for all versions of Percona Server for MySQL, and MySQL. It performs streaming, compressed, and incremental MySQL backups.

**Important:**

With the introduction of Percona XtraBackup 8.0, Percona XtraBackup 2.4 will continue to support MySQL and Percona Server 5.6 and 5.7 databases. Due to the new MySQL redo log and data dictionary formats the Percona XtraBackup 8.0.x versions will only be compatible with MySQL 8.0.x and the upcoming Percona Server for MySQL 8.0.x

Percona XtraBackup works with MySQL and Percona Server for MySQL. It supports completely non-blocking backups of InnoDB, XtraDB, and MyRocks storage engines. In addition, it can back up the following storage engines by briefly pausing writes at the end of the backup: MyISAM, [Merge](https://www.percona.com/doc/percona-xtrabackup/LATEST/glossary.html#term-mrg), and [Archive](https://www.percona.com/doc/percona-xtrabackup/LATEST/glossary.html#term-arm), including partitioned tables, triggers, and database options.

Percona’s enterprise-grade commercial [MySQL Support](http://www.percona.com/mysql-support/) contracts include support for Percona XtraBackup. We recommend support for critical production deployments.

# Percona  XtraBackup特性

| **Features** | **Percona XtraBackup** | **MySQL Enterprise backup** |
| --- | --- | --- |
| License | GPL | Proprietary |
| Price | Free | Included in subscription at $5000 per Server |
| Streaming and encryption formats | Open source | Proprietary |
| Supported MySQL flavors | MySQL, Percona Server for MySQL, Percona XtraDB Cluster, | MySQL |
| Supported operating systems | Linux | Linux, Solaris, Windows, OSX, FreeBSD. |
| Non-blocking InnoDB backups [[1]](https://www.percona.com/doc/percona-xtrabackup/LATEST/intro.html#n-1) | Yes | Yes |
| Blocking MyISAM backups | Yes | Yes |
| Incremental backups | Yes | Yes |
| Full compressed backups | Yes | Yes |
| Incremental compressed backups | Yes |  |
| Fast incremental backups [[2]](https://www.percona.com/doc/percona-xtrabackup/LATEST/intro.html#n-2) | Yes |  |
| Incremental backups with archived logs feature in Percona Server | Yes |  |
| Incremental backups with REDO log only |  | Yes |
| Backup locks [[7]](https://www.percona.com/doc/percona-xtrabackup/LATEST/intro.html#n-8) | Yes (LOCK TABLE FOR BACKUP) | Yes (LOCK INSTANCE FOR BACKUP) |
| Encrypted backups | Yes | Yes [[3]](https://www.percona.com/doc/percona-xtrabackup/LATEST/intro.html#n-3) |
| Streaming backups | Yes | Yes |
| Parallel local backups | Yes | Yes |
| Parallel compression | Yes | Yes |
| Parallel encryption | Yes | Yes |
| Parallel apply-log | Yes |  |
| Parallel copy-back |  | Yes |
| Partial backups | Yes | Yes |
| Partial backups of individual partitions | Yes |  |
| Throttling [[4]](https://www.percona.com/doc/percona-xtrabackup/LATEST/intro.html#n-4) | Yes | Yes |
| Backup image validation |  | Yes |
| Point-in-time recovery support | Yes | Yes |
| Safe slave backups | Yes |  |
| Compact backups [[5]](https://www.percona.com/doc/percona-xtrabackup/LATEST/intro.html#n-5) | Yes |  |
| Buffer pool state backups | Yes |  |
| Individual tables export | Yes | Yes [[6]](https://www.percona.com/doc/percona-xtrabackup/LATEST/intro.html#n-6) |
| Individual partitions export | Yes |  |
| Restoring tables to a different server | Yes | Yes |
| Data & index file statistics | Yes |  |
| InnoDB secondary indexes defragmentation | Yes |  |
| rsync support to minimize lock time | Yes |  |
| Improved FTWRL handling | Yes |  |
| Backup history table | Yes | Yes |
| Backup progress table |  | Yes |
| Offline backups |  | Yes |
| Backup to tape media managers |  | Yes |
| Cloud backups support |  | Amazon S3 |
| External graphical user interfaces to backup/recovery | Zmanda Recovery Manager for MySQL | MySQL Workbench, MySQL Enterprise Monitor |

# XtraBackup完整备份

## 3.1 备份的权限

  使用XtraBack做备份，需要连接到数据库服务器并在服务器上有相关目录操作权限，所以必须要有数据库的相关操作权限以及相关目录执行READ、WRITE以及EXECUTE（执行innobackupex 或xtrabackup 命令）的系统权限。

下面我们来简单介绍一下：

数据库所需的一些权限请参考下面表格，细节部分可以参考官方文档。此处不详细阐述！

|  |  |
| --- | --- |
| MySQL 数据库权限 | 权限描述 |
| RELOAD/ LOCK TABLES | RELOAD权限是管理员级别权限，用来执行flush-hosts, flush-logs, flush-privileges, flush-status, flush-tables,flush-threads, refresh, reload等命令的权限。XtraBackup在复制文件之前，必须先执行FLUSH TABLES WITH READ LOCK和 FLUSH ENGINE LOGS等 |
| REPLICATION CLIENT | 此权限是为了获取二进制日志位置（Position） |
| CREATE TABLESPACE | 此权限是为了导入表 |
| PROCESS | 此权限是因为要运行SHOW ENGINE INNODB STATUS命令，以及查看服务器上所有运行的线程。 |
| SUPER | 此权限是为了开启、关闭复制环境中的slave threads |
| CREATE | 不是必须，Percon Server数据库才需要的。创建PERCONA\_SCHEMA.xtrabackup\_history的需要 |
| INSERT | 不是必须，跟上面权限一样，需要往PERCONA\_SCHEMA.xtrabackup\_history插入记录 |
| SELECT | 不是必须，跟上面权限一样，需要查询ERCONA\_SCHEMA.xtrabackup\_history中的记录 |

当然所需的最小权限如下所示，注意：这个仅仅是最小权限，实际情况中最好根据实际需求授予最小权限。不要放大不必要的权限。

mysql> use mysql;

Database changed

mysql> CREATE USER 'backuser'@'localhost' IDENTIFIED BY 'oracle';

Query OK, 0 rows affected (0.00 sec)

mysql> GRANT RELOAD,LOCK TABLES,PROCESS,REPLICATION CLIENT ON \*.\* TO 'backuser'@'localhost';

Query OK, 0 rows affected (0.00 sec)

mysql> FLUSH PRIVILEGES;

Query OK, 0 rows affected (0.00 sec)

如果上面的权限不够的话，可以使用下面授权语句。

GRANT RELOAD, LOCK TABLES,PROCESS, REPLICATION CLIENT, CREATE TABLESPACE,SUPER ON \*.\* TO 'backuser'@'localhost';

系统账号授权相关目录的操作权限。如下所示：

[root@DB-Server ~]# chown -R mysql:mysql /data

[root@DB-Server ~]# su - mysql

-bash-3.2$

## 3.2 完整备份

 常用的完整备份（全备）命令如下所示

innobackupex --defaults-file=/etc/my.cnf --socket=/tmp/mysql.sock --user=backuser --password=oracle /backup/full20190904/

参数--no-timestamp 表示不生成带时间戳的目录

innobackupex --defaults-file=/etc/my.cnf --socket=/tmp/mysql.sock --user=backuser --password=oracle /backup/full20190904\_2/ --no-timestamp

  注意：如果你指定了参数--defaults-file，那么此参数就必须位于第一个，否则就会报“--defaults-file must be specified first on the command line”.

 先在测试数据库MyDB创建一个测试表，插入几条数据，后面做还原测试验证使用。

mysql> use MyDB;

Database changed

mysql> create table test(id  int primary key);

Query OK, 0 rows affected (0.13 sec)

mysql> insert into test values(1000);

Query OK, 1 row affected (0.01 sec)

mysql> insert into test values(2000);

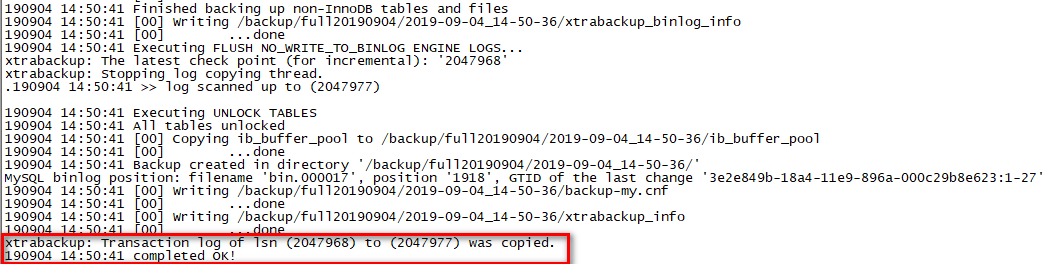
Query OK, 1 row affected (0.01 sec)

mysql> insert into test values(3000);

Query OK, 1 row affected (0.01 sec)

使用下面命令对MySQL做了一个完整备份，那么默认就会在备份目录下生成一个日期时间格式的全备文件

innobackupex --defaults-file=/etc/my.cnf --socket=/tmp/mysql.sock --user=backuser --password=oracle /backup/full20190904/



 那么备份出来的文件以及内容是什么样的呢？ 如下所示，你可以检查xtrabackup\_checkpoints、xtrabackup\_binlog\_info、xtrabackup\_info三个文件了解备份的一些详细信息，例如备份类型、是否压缩等等。

# more xtrabackup\_checkpoints

backup\_type = full-backuped

from\_lsn = 0

to\_lsn = 2047968

last\_lsn = 2047977

compact = 0

recover\_binlog\_info = 0# more xtrabackup\_binlog\_info

bin.000017 1918 3e2e849b-18a4-11e9-896a-000c29b8e623:1-27# more xtrabackup\_info

uuid = 4f9189c1-cee0-11e9-87b6-000c29b8e623

name =

tool\_name = innobackupex

tool\_command = --defaults-file=/etc/my.cnf --socket=/tmp/mysql.sock --user=backuser --password=... /backup/full20190904/

tool\_version = 2.4.11

ibbackup\_version = 2.4.11

server\_version = 5.7.21-log

start\_time = 2019-09-04 14:50:36

end\_time = 2019-09-04 14:50:41

lock\_time = 0

binlog\_pos = filename 'bin.000017', position '1918', GTID of the last change '3e2e849b-18a4-11e9-896a-000c29b8e623:1-27'

innodb\_from\_lsn = 0

innodb\_to\_lsn = 2047968

partial = N

incremental = N

format = file

compact = N

compressed = N

encrypted = N

xtrabackup\_checkpoints:

备份类型信息（完全备份或增量备份）、LSN(日志序列号)范围信息、备份是否压缩.

 xtrabackup\_binlog\_info:

二进制日志信息,MySQL当前正在使用的二进制日志文件及至备份这一刻为止二进制日志事件的位置。如果没有开启二进制日志，那么就不会有xtrabackup\_binlog\_info这个文件。

 xtrabackup\_info:

xtrabackup工具的版本信息、详细命令、参数以及备份的开始、结束时间。以及备份详细信息等。

 backup-my.cnf :备份命令用到的配置选项信息

# 如何使用全备恢复还原

## 4.1 全库备份

innobackupex --defaults-file=/etc/my.cnf --socket=/tmp/mysql.sock --user=backuser --password=oracle /backup/full20190904/

## 4.2 恢复准备

关闭MySQL服务，备份数据目录或删除数据目录

# systemctl stop mysql

# mv /data/mysql/ /data/mysql\_20190904\_bak

## 4.3 全库恢复

恢复数据，将备份数据文件拷贝到数据目录

innobackupex--defaults-file=/etc/my.cnf--copy-back /backup/full20190904/2019-09-04\_14-50-36

如果执行上面命令是在mysql用户下，并且有权限创建对应目录，那么就不需要做任何操作，如果是root账号，那么就必须修改对应目录的Owner，否则启动MySQL服务会报错。

chown -R mysql:mysql /data/

chown -R mysql:mysql /undolog/

 注意，如果这里权限没有处理好，有时候会出现一些莫名其妙的错误，例如，设置文件权限时，由于上面脚本处理的是/data/mysql下面的文件，没有处理/data/mysql这个目录的权限，启动MySQL的时候就报这个错误

# ls -lrt

drwxr-x--- 8 root  root  4096 Jun 25 12:09 mysql

# systemctl stop mysql

Starting MySQL.. ERROR! The server quit without updating PID file (/data/mysql/DB-Server.localdomain.pid).

# chown -R mysql:mysql /data/mysql

# systemctl stop mysql

Starting MySQL... SUCCESS!

## 4.4 测试数据的完整性：

systemctl stop mysql

[root@DB-Server ~]# mysql -u root -p

mysql> use MyDB;

Database changed

mysql> select \* from test;

+------+

| id   |

+------+

| 1000 |

| 2000 |

| 3000 |

+------+

3 rows in set (0.00 sec)

# XtraBackup增量备份

在阐述Xtrabackup增量备份前，我们先来看看下面这个增量备份，你能看出其问题吗？这个备份有啥不妥呢？ 当然在做增量备份前，都有对数据库进行了一些测试操作。

完整备份

innobackupex --defaults-file=/etc/my.cnf --socket=/tmp/mysql.sock --user=backuser --password=oracle /backup/full20190904/

增量备份1

innobackupex --defaults-file=/etc/my.cnf--socket=/tmp/mysql.sock --user=root --password=oracle --incremental--incremental-basedir=/backup/full20190904/2019-09-04\_14-50-36 /backup/incr/

增量备份2

innobackupex --defaults-file=/etc/my.cnf--socket=/tmp/mysql.sock --user=root --password=oracle --incremental--incremental-basedir=/backup/full20190904/2019-09-04\_14-50-36 /backup/incr/

其实第一次完整备份时，生成了备份目录2019-09-04\_14-50-36； 第一次增量备份时，生成了增量备份目录2019-09-04\_15-24-07,2019-09-04\_15-25-16，关键在于第二次增量备份，--incremental-basedir这个参数的值不应该是第一次完成备份的目录，而应是增量备份1的备份目录2017-07-24\_09-40-11，如下所示，如果使用了错误的参数，增量备份2其实包含了增量备份1的变化部分。

增量备份注意事项

1：--incremental-basedir

第一次增量备份，--incremental-basedir肯定是全备目录，第二次增量备份，--incremental-basedir为第一次增量备份目录，依此类推！

2：增量备份仅能应用于InnoDB或XtraDB表，对于MyISAM表而言，执行增量备份时其实进行的是完全备份。

3：应用增量备份的时候只能按照备份的顺序来应用。如果应用顺序错误，那么备份就不可用。如果无法确定顺序，可以查看xtrabackup-checkpoints来确定顺序。

# 增量备份的还原

那么我们来用案例测试一下增量备份的还原，具体步骤如下所示

## 6.1 完整备份

执行完整备份

innobackupex --defaults-file=/etc/my.cnf --socket=/tmp/mysql.sock --user=backuser --password=oracle /backup/

## 6.2 增量备份1

注意 --incremental-basedir参数应该根据实际情况确定

innobackupex --defaults-file=/etc/my.cnf --socket=/tmp/mysql.sock --user=root --password=oracle --incremental--incremental-basedir=/backup/2019-09-04\_16-51-58 /backup/

## 6.3 增量备份2

mysql> create table big\_table (id  int);

Query OK, 0 rows affected (0.13 sec)

mysql> insert into big\_table values(10000);

Query OK, 1 row affected (0.00 sec)

mysql> insert into big\_table values(20000);

Query OK, 1 row affected (0.00 sec)

注意 --incremental-basedir参数应该根据实际情况确定

innobackupex --defaults-file=/etc/my.cnf --socket=/tmp/mysql.sock --user=root --password=oracle --incremental--incremental-basedir=/backup/2019-09-04\_16-54-45 /backup/

注意：Xtrabackup增量备份时，需要设置参数--incremental-basedir 其实这个参数不是必须的。可以用--incremental-lsn来替代。完整备份后，检查备份目录下的xtrabackup\_checkpoints文件，找到to\_lsn，然后使用下面脚本做增量备份。

# more xtrabackup\_checkpoints

backup\_type = full-backuped

from\_lsn = 0

to\_lsn = 1733142

last\_lsn = 1733142

compact = 0

recover\_binlog\_info = 0

#innobackupex --user=backuser--password=kkk123456--incremental --incremental-lsn=1733142 /u02/mysql\_backup/incr/

然后，我们先删除或DROP掉一些数据，模拟数据被误删的情况，然后还原测试验证。 如下所示：

mysql> drop table test;

Query OK, 0 rows affected (0.00 sec)

mysql> delete from big\_table where id=10000;

Query OK, 1 row affected (0.01 sec)

## 6.5 恢复完全备份

innobackupex --defaults-file=/etc/my.cnf --apply-log --redo-only /backup/2019-09-04\_16-51-58

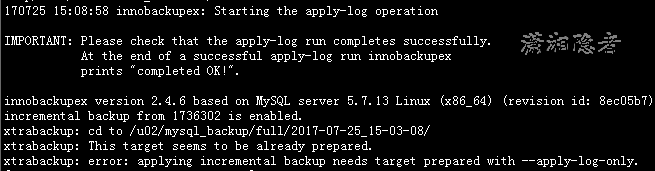
 注意:如果后续必须应用增量备份，则必须加上--redo-only. 否则后续不能应用增量备份。

 --redo-only

If you replay the committed transactions and rollback the uncommitted ones on the base backup, you will not be able to add the incremental ones. If you do this on an incremental one, you won’t be able to add data from that moment and the remaining increments。

Note: --redo-only should be used when merging all incrementals except the last one. That’s why the previous line doesn’t contain the --redo-only option. Even if the --redo-only was used on the last step, backup would still be consistent but in that case server would perform the rollback phase.

如下截图所示，如果在恢复全备时没有应用参数--redo-only ，后面应用增量备份时就会遇到下面错误。只有当最后一个增量备份时，才不需要--redo-only

[](https://images2018.cnblogs.com/blog/73542/201806/73542-20180627233332966-1484209564.png)

## 6.6 增量备份1还原

innobackupex --apply-log --redo-only --incremental /backup/full20190904/2019-09-04\_14-50-36--incremental-dir=/backup/incr/2019-09-04\_15-36-06

 注意：增量备份必须按一定顺序Apply，否则后面的增量备份都无效了！

## 6.7 增量备份2还原

innobackupex --apply-log--incremental /backup/full20190904/2019-09-04\_14-50-36 --incremental-dir=/backup/incr/2019-09-04\_15-38-00

innobackupex--defaults-file=/etc/my.cnf--copy-back /backup/incr/2019-09-04\_15-38-00

## 6.8 检查验证数据数据恢复情况。

[root@DB-Server mysql]# service mysql start

Starting MySQL..........[  OK  ]

mysql> select \* from test;

+------+-------+

| id   | name  |

+------+-------+

| 1000 | kerry |

| 2000 | ken   |

+------+-------+

2 rows in set (0.00 sec)

mysql> select \* from big\_table;

+-------+

| id    |

+-------+

| 10000 |

| 20000 |

+-------+

2 rows in set (0.00 sec)

# XtraBackup备份高级选项

## 7.1 Compact Backups

1:紧凑型备份（compact backups）指在备份InnoDB tables时， 可以不备份二级索引（Secondary index)数据页，这样使得备份更加紧凑，从而可以减少备份的大小。缺点是备份准备过程（Prepare）需要更长的时间才能重建这些二级索引(secondary indexes)。

2：系统表空间不支持紧凑型备份（Compact backups）,所以如果要使用Compact backups，那么就必须设置变量innodb\_file\_per\_table为ON

 innobackupex --user=backuser--password=kkk123456--compact  /u02/mysql\_backup/full/

 如下所示，你检查xtrabackup\_checkpoints 就会发现compact为1， xtrabackup\_info里面compact值为Y

**注:**发现Xtrabackup有个有意思的现象，例如下面备份脚本，参数 --compat弄错了，执行也不报错，只是紧凑型备份没有生效。

## 7.2 Partial Backups

 Percona Xtrabackup提供部分备份，这意味着你可以只备份一些特定表或数据库，不过部分备份（Partial Backups）是有诸多限制和前提的。

1： 如果要使用部分备份（Partial Backups），前提条件是你备份的表必须在单独的表空间中。你必须在服务器上启动innodb\_file\_per\_table。

2： 部分备份（Partial Backups）是不支持--stream选项，即不支持将数据通过管道传输给其它程序进行处理。

3： 还原部分备份跟还原完整备份也有所不同，即你不能通过简单地将prepared的部分备份使用--copy-back选项直接复制回数据目录，而是要通过导入表的方向来实现还原。尽管有时简单的拷贝备份文件可以成功，但是这种方法很容易导致数据库的不一致，因此不推荐大家这么做。

**创建部分备份（Creating Partial Backups）**部分备份共有三种方式，分别是：1. 用正则表达式表示要备份的库名及表名（参数为--include）；2. 将要备份的表名或库名都写在一个文本文件中（参数为--tables-file）以及 3. 将要备份表名或库名完整的写在命令行中（参数为：--databases）。（译者注：不管你备份哪个库或是哪张表，强烈推荐把mysql库也一起备份，恢复的时候要用。）

对数据库进行部分备份有三种方式：正则表达式(--include), 枚举表文件(--tables-file)和列出要备份的数据库(--databases)

方法1：使用正则表达式

innobackupex --defaults-file=/usr/my.cnf--user=backuser--password=kkk123456--include='^MyDB[.]kkk'   /u02/mysql\_backup/part/

方法2：枚举表文件(--tables-file)

# cat >back\_table.txt

MyDB.test

MyDB.kkk

innobackupex --defaults-file=/usr/my.cnf--user=backuser--password=kkk123456 --tables-file=/u02/mysql\_backup/back\_table.txt  /u02/mysql\_backup/part

方法3： 通过参数--database 指定需要备份的表

–database选项可以直接指定需要备份的”表”和”库”名,或指定一个包含所需备份表名的列表文件。

innobackupex --defaults-file=/usr/my.cnf--user=backuser--password=kkk123456 --databases="MyDB.test MyDB.kkk"  /u02/mysql\_backup/part

### 7.2.1准备部分备份（Preparing Partial Backups）

innobackupex--apply-log --export /u02/mysql\_backup/part/2018-06-27\_16-49-59

### 7.2.2恢复部分备份（Restoring Partial Backups）

innobackupex--apply-log --export /u02/mysql\_backup/part/2018-06-27\_16-49-59

此时就会生成对应表的exp文件

[](https://images2018.cnblogs.com/blog/73542/201806/73542-20180627233337337-1693977191.png)

导入表操作：

假设由于误操作，将表TEST的数据TRUNCATE掉了。

mysql> select \* from test;

+------+-------+

| id   | name  |

+------+-------+

| 1000 | kerry |

| 2000 | ken   |

+------+-------+

2 rows in set (0.00 sec)

mysql> truncate table test;

Query OK, 0 rows affected (0.12 sec)

2:我们discard tablespace

mysql> alter table test discard tablespace;

Query OK, 0 rows affected (0.03 sec)

3: copy test表的ibd、cfg、exp文件

#cp /u02/mysql\_backup/part/2018-06-27\_16-49-59/MyDB/test.ibd  /data/mysql/MyDB

#cp /u02/mysql\_backup/part/2018-06-27\_16-49-59/MyDB/test.cfg  /data/mysql/MyDB

#cp /u02/mysql\_backup/part/2018-06-27\_16-49-59/MyDB/test.exp  /data/mysql/MyDB

4： 导入表空间

mysql> alter table test import tablespace;

Query OK, 0 rows affected (0.03 sec)

5：验证测试是否成功

[](https://images2018.cnblogs.com/blog/73542/201806/73542-20180627233337947-301404490.png)

## 7.3 基于tar格式备份

流备份打包

 innobackupex --user=backuser --password=kkk123456 --stream=tar /tmp  > /u02/mysql\_backup/full/backup\_20170725.tar

xbstream流备份打包压缩

innobackupex --user=root--password=123456--stream=tar /tmp  | gzip >  /u02/mysql\_backup/full/backup\_20170725.tar.gz

# Innobackupex部分参数介绍

查看innobackupex与xtrabackup的参数，可以使用下面命令：

#innobackupex --help

innobackupex的的部分参数介绍如下（有些不常用参数没来及整理，后续再慢慢补充）

**-v, --version**

输出xtrabackup的版本信息

**-?, --help**

输出、显示帮助信息然后退出

**--defaults-file**

默认配置文件的路径，如果不指定该参数，xtrabackup将从依次从以下位置查找配置文件/etc/my.cnf、/etc/mysql/my.cnf、/usr/local/etc/my.cnf、~/.my.cnf，并读取配置文件中的[mysqld]和[xtrabackup]配置段。[mysqld]中只需要指定datadir、innodb\_data\_home\_dir、innodb\_data\_file\_path、innodb\_log\_group\_home\_dir、innodb\_log\_files\_in\_group、innodb\_log\_file\_size6个参数即可让xtrabackup正常工作。

注意：指定要备份的mysql实例的my.cnf文件，必须为第一个选项.

**--apply-log**

该参数是对xtrabackup的参数--prepare的封装。 一般情况下,在备份完成后，数据尚且不能用于恢复操作，因为备份的数据中可能会包含尚未提交的事务或已经提交但尚未同步至数据文件中的事务。因此，此时数据文件仍处理不一致状态。--apply-log的作用是通过回滚未提交的事务及同步已经提交的事务至数据文件使数据文件处于一致性状态

**--redo-only**

这个选项在prepare base full backup，往其中merge增量备份（但不包括最后一个）时候使用。This forces xtrabackup to skip the "rollback" phase and do a "redo" only

**--copy-back**

做数据恢复时将备份数据文件拷贝到MySQL服务器的datadir目录;

**--move-back**

将所有文件从以前创建的备份目录移动到事情数据目录， 使用此选项必须小心， 因为它会删除备份文件。

**--galera-info**

该选项创建xtrabackup\_galera\_info文件,其中包含备份当时的本地节点状态.执行Percona-XtraDB-Cluster备份时应使用此选项。

**--slave-info**

该选项表示对slave进行备份的时候使用，打印出master的名字和binlog pos，同样将这些信息以change master的命令写入xtrabackup\_slave\_info文件。可以通过基于这份备份启动一个从库。

**--incremental**

建立增量备份

**--no-lock**

该选项表示关闭FTWRL的表锁，只有在所有表都是Innodb表并且不关心backup的binlog pos点，如果有任何DDL语句正在执行或者非InnoDB正在更新时（包括mysql库下的表），都不应该使用这个选项，后果是导致备份数据不一致，如果考虑备份因为获得锁失败，可以考虑--safe-slave-backup立刻停止复制线程。

不允许使用flush tables with read lock表锁。如果你的所有表示INNODB并且你不关心二进制日志备份的位置。如果有任何DDL语句被执行或任何非INNODB表上的update操作，这个选项就不能使用

**--safe-slave-backup**

停止从SQL进程并等待启动备份直到slave\_open\_temp\_tables的值为0。如果没有打开临时表，备份会进行，否则SQL进程将启动并直到没有打开的临时表时停止。如果slave\_open\_temp\_tables在--safe-slave-backup-timeout秒后没有变成0，则备份会失败。备份结束后，从SQL进程将重新启动。

**--rsync**

表示通过rsync工具优化本地传输，当指定这个选项，innobackupex使用rsync拷贝非Innodb文件而替换cp，当有很多DB和表的时候会快很多，不能--stream一起使用。

**--force-non-empty-directories**

使用此选项，就可以通过--copy-back或--move-back 将文件拷贝或移动到非空目录。

This option, when specified, makes --copy-back or  --move-back transfer files to non-empty directories. Note that no existing files will be overwritten. If --copy-back or --nove-back has to copy a file from the backup directory which already exists in the destination directory, it will still fail with an error.

**--no-timestamp**

该选项可以表示要不要创建一个时间戳目录来存储备份，指定到自己想要的备份文件夹。指定了这个选项，备份会直接备份在BACKUP-DIR，不再创建时间戳文件夹

**--no-version-check**

禁止版本检查

**--no-backup-locks**

**--decompress**

**-u, --user=name**

该选项表示备份账号。

**-H, --host=**

该选项表示备份数据库的地址。

**-p, --password**

该选项表示备份账号的密码。

**-P, --port=#**

该选项表示当前需要备份数据库的使用的端口。

**-S, --socket**

此选项指定连接本地数据库时使用的套接字，此选项接受一个字符串参数。

**--incremental-history-name=name**

该选项表示存储在PERCONA\_SCHEMA.xtrabackup\_history基于增量备份的历史记录的名字。Percona Xtrabackup搜索历史表查找最近（innodb\_to\_lsn）成功备份并且将to\_lsn值作为增量备份启动出事lsn.与innobackupex--incremental-history-uuid互斥。如果没有检测到有效的lsn，xtrabackup会返回error。

**--incremental-history-uuid=name**

**--decrypt=name**

**--ftwrl-wait-query-type=name**

**--kill-long-query-type=name**

**--history[=name]**

**--include=name**

**--databases**

该选项表示需要备份的数据库,如果没有指定该参数，表示备份全部数据库。如果要指定多个数据库，彼此间需要以空格隔开；如："mydb1 mydb2"，同时，在指定某数据库时，也可以只指定其中的某张表。如："mydatabase.mytable"。该选项对innodb引擎表无效，还是会备份所有innodb表。此外，此选项也可以接受一个文件为参数，文件中每一行为一个要备份的对象。

**--kill-long-queries-timeout=#**

**--ftwrl-wait-timeout=#**

**--ftwrl-wait-threshold=#**

**--debug-sleep-before-unlock=#**

**--safe-slave-backup-timeout=#**

**--close-files**

**--compact**

该选项表示创建一份没有辅助索引的紧凑的备份。

指在备份InnoDB tables时， 可以不备份二级索引（Secondary index)数据页，这样使得备份更加紧凑，从而可以减少备份的大小。缺点是备份准备过程（Prepare）需要更长的时间才能重建这些二级索引(secondary indexes)。

**--compress[=name]**

对备份数据进行行压缩

**--compress-threads=#**

指定备份压缩的线程数量。

**--compress-chunk-size=#**

**--encrypt=name**

**--encrypt-key=name**

**--encrypt-key-file=name**

**--encrypt-threads=#**

**--encrypt-chunk-size=#**

**--export**

**--extra-lsndir=name**

**--incremental-basedir=name**

指定一个全库备份的目录作为增量备份的基础数据库.与--incremental同时使用。

**--incremental-dir=name**

该选项表示增量备份的目录。

**--incremental-force-scan**

该选项表示创建一份增量备份时，强制扫描所有增量备份中的数据页。

**--log-copy-interval=#**

**--incremental-lsn=name**

该选项表示指定增量备份的LSN，与--incremental选项一起使用。

**--parallel**

**--rebuild-indexes**

**--rebuild-threads=#**

**--stream**

该选项表示流式备份的格式，backup完成之后以指定格式到STDOUT，目前只支持tar和xbstream。

备份文件输出格式, tar时使用tar4ibd , 该文件可在XtarBackup binary文件中获得.如果备份时有指定--stream=tar, 则tar4ibd文件所处目录一定要在$PATH中(因为使用的是tar4ibd去压缩, 在XtraBackup的binary包中可获得该文件)。

在使用参数stream=tar备份的时候，你的xtrabackup\_logfile可能会临时放在/tmp目录下，如果你备份的时候并发写入较大的话xtrabackup\_logfile可能会很大(5G+)，很可能会撑满你的/tmp目录，可以通过参数--tmpdir指定目录来解决这个问题。

**--tables-file**

该选项表示指定含有表列表的文件，格式为database.table，该选项直接传给--tables-file。

**--throttle**

每秒IO次数,限制backup时使用的I/O操作量，使备份对数据库正常业务的影响最小化

**-t, --tmpdir=name**

**--use-memory**

指定备份所用内存大小。