xtrabackup 恢复 restore

```
# ./xtrabackup --prepare --target-dir=/mydata/mybak/base
# ./xtrabackup --prepare --target-dir=/mydata/mybak/base --incremental-dir=/mydata/mybak/delta
```

上述过程，首先将全部文件备份到/mydata/mybak/base目录下，增量备份产生的文件备份到/mydata/mybak/delta目录下。在恢复过程中，首先指定全量备份的路径，然后将增量备份应用于该全量备份。
将/mydata/mybak/base目录下的文件拷贝到datadir目录下，即可。同时还需要数据表结构文件。

1. 停掉mysql服务
2. 删除mysql 数据目录下所有文件
3. #innobackupex --defaults-file=/opt/mysql/my.cnf --user=root --password=123456 --apply-log /opt/mysql/backup/2016-08-05_10-14-42 看到100313 10:51:44 innobackupex: completed OK! 表示成功 #innobackupex --defaults-file=/opt/mysql/my.cnf --user=root --password=123456 --copy-back /opt/mysql/backup/2016-08-05_10-14-42 看到100313 10:58:44 innobackupex: completed OK!表示成功
4. 启动mysql服务

xtrabackup没有什么功能来还原备份，可以直接通过rsync，cp来还原数据库
注意：注意保持datadir必须是空的，并且mysql服务是停止的。不能还原到已经在运行的mysql服务中。
通过rsync还原：

```
$ rsync -avrP /data/backup/ /var/lib/mysql/
```

还原后注意修改所有者

```
$ chown -R mysql:mysql /var/lib/mysql
```

注意：xtrabackup只备份innodb数据文件，不会备份其他引擎的表，和frm文件。如果要对整个库备份还原可以使用innodbbackupex

```
[root@BkpV5Mysql01 xtrabackup]# more xtrarestore_2017-08-21.sh
#预备全备，不回滚未提交的事物 ：
xtrabackup --prepare --apply-log-only --target-dir=/srv/xtrabackup/wift_bcp/2017-08-21
#xtrabackup --prepare --target-dir=/srv/xtrabackup/wift_bcp/2017-08-21
#应用第一个增量备份 ：
xtrabackup --prepare --apply-log-only --target-dir=/srv/xtrabackup/wift_bcp/2017-08-21 --incremental-dir=/srv/xtrabackup/wift_bcp/2017-08-21-1038
#应用增量备份 ：
xtrabackup --prepare --apply-log-only --target-dir=/srv/xtrabackup/wift_bcp/2017-08-21 --incremental-dir=/srv/xtrabackup/wift_bcp/2017-08-21-1040
[root@BkpV5Mysql01 xtrabackup]#
```

```
xtrabackup --user=xtrabkp --password=Xtrabkp_6 --datadir=/var/lib/mysql/ --backup --parallel=1 --databases=wift_arch --target-dir=/srv/xtrabackup/wift_arch/$(date +'%F')
```

```
/data/backups/base
/data/backups/inc1
/data/backups/inc2
```

1. To prepare the base backup, you need to run xtrabackup --prepare as usual, but prevent the rollback phase:

```
$ xtrabackup --prepare --apply-log-only --target-dir=/data/backups/base
```

If you do not use the xtrabackup --apply-log-only option to prevent the rollback phase, then your incremental backups will be useless. After transactions have been rolled back, further incremental backups cannot be applied.

This backup is actually safe to restore as-is now, even though the rollback phase has been skipped. If you restore it and start *MySQL*, *InnoDB* will detect that the rollback phase was not performed, and it will do that in the background, as it usually does for a crash recovery upon start. It will notify you that the database was not shut down normally.

2. To apply the first incremental backup to the full backup, run the following command:

```
$ xtrabackup --prepare --apply-log-only --target-dir=/data/backups/base \
--incremental-dir=/data/backups/inc1
```

3. Preparing the second incremental backup is a similar process: apply the deltas to the (modified) base backup, and you will roll its data forward in time to the point of the second incremental backup:

```
$ xtrabackup --prepare --target-dir=/data/backups/base \
--incremental-dir=/data/backups/inc2
```

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

Once prepared incremental backups are the same as the full backups and they can be restored the same way.

4. For convenience **xtrabackup** binary has an xtrabackup --copy-back option, which will copy the backup to the server’ s datadir:

```
$ xtrabackup --copy-back --target-dir=/data/backups/
## Use chmod to correct the permissions, if necessary!
```

If you don’ t want to save your backup, you can use the xtrabackup --move-back option which will move the backed up data to the datadir.

If you don’ t want to use any of the above options, you can additionally use **rsync** or **cp** to restore the files.

Example of the **rsync** command that can be used to restore the backup can look like this:

```
$ rsync -avrP /data/backup/ /var/lib/mysql/
```

You should check that the restored files have the correct ownership and permissions.

As files’ attributes will be preserved, in most cases you will need to change the files’ ownership to mysql before starting the database server, as they will be owned by the user who created the backup:

```
$ chown -R mysql:mysql /var/lib/mysql
```

Note: The datadir must be empty before restoring the backup. Also it’ s important to note that MySQL server needs to be shut down before restore is performed. You can’ t restore to a datadir of a running mysqld instance (except when importing a partial backup).

eg :

```
xtrabackup --prepare --apply-log-only --target-dir=/srv/backup/wift_arch/2017-08-22

xtrabackup --prepare --apply-log-only --target-dir=/srv/backup/wift_arch/2017-08-22 --incremental-dir=/srv/backup/wift_arch/2017-08-23
```

报错 :

```
[root@MysqlRestore 2017-08-24]# xtrabackup --prepare --target-dir=/srv/backup/wift_arch/2017-08-23/ --incremental-dir=/srv/backup/wift_arch/2017-08-24/
xtrabackup version 2.4.8 based on MySQL server 5.7.13 Linux (x86_64) (revision id: 97330f7)
incremental backup from 92876680058 is enabled.
xtrabackup: cd to /srv/backup/wift_arch/2017-08-23/
xtrabackup: This target seems not to have correct metadata...
[root@MysqlRestore 2017-08-24]#
```

恢复步骤 :

prepare

拷贝数据库目录下数据文件(ibd),表结构文件(frm)到数据目录下

拷贝ibdata1覆盖原来的ibdata1文件到数据目录下