

BACKUP MySQL®



No Backup ?





Agenda

1

mysqldump

2

mydumper/myloader

3

innobackupex/xtraBackup

4

mysqlbinlog

01

mysqldump

<https://dev.mysql.com/doc/refman/5.6/en/mysqldump.html>

<https://dev.mysql.com/doc/refman/5.6/en/using-mysqldump.html>

mysqldump



Logical Backup



Single Thread(<5.7)



Official

Format Options

- + `--default-character-set`
- + `--result-file=file_name, -r file_name`
- + `--no-data, -d`
- + `--tab=dir_name, -T dir_name`

This options are used with the `--tab` option and has the same meaning as the corresponding clause for **SELECT INTO OUTFILE**

- ✓ `--fields-enclosed-by`
- ✓ `--fields-escaped-by`
- ✓ `--fields-optionally-enclosed-by`
- ✓ `--fields-terminated-by`
- ✓ `--lines-terminated-by`

Filtering Options

+ --events, -E & --routines, -R & --triggers

+ --all-databases, -A

+ --databases, -B

+ --tables

Usage: mysqldump [--tables] dbname tbl_1 tbl_2

+ --ignore-table=db_name.tbl_name

To ignore multiple tables, use this option multiple times

+ --where='where_condition', -w 'where_condition'

Transactional Options



--lock-all-tables, -x

Lock all tables across all databases by acquiring a global read lock for the duration of the whole dump



--lock-tables, -l

For each dumped database, lock all tables to be dumped before dumping them



--single-transaction

This option sets the transaction isolation mode to **REPEATABLE READ** and sends a **START TRANSACTION** SQL statement to the server before dumping data. It is useful only with transactional tables such as InnoDB

Performance Options

+ --quick, -q

Retrieving rows for a table row-by-row rather than retrieving the entire row set and buffering it in memory before writing it out

+ --extended-insert, -e

Write INSERT statements using multiple-row syntax that includes several VALUES lists

+ --disable-keys , -K

For each table, surround the INSERT statements with */*!40000 ALTER TABLE tbl_name DISABLE KEYS */;* and */*!40000 ALTER TABLE tbl_name ENABLE KEYS */;*

+ --opt

Combination of *--add-drop-table* *--add-locks* *--create-options* *--disable-keys* *--extended-insert* *--lock-tables* *--quick* *--set-charset*

Replication Issues

+ --skip-opt

Combination of *--add-drop-table* *--add-locks* *--create-options* *--disable-keys*
--extended-insert *--lock-tables* *--quick* *--set-charset*

```
/*!40101 SET @saved_cs_client      = @@character_set_client */;  
/*!40101 SET character_set_client = utf8 */;  
CREATE TABLE `w1x` (  
  `id` int(10) unsigned NOT NULL,  
  `f1` varchar(32) NOT NULL DEFAULT '',  
  `f2` varchar(40) NOT NULL DEFAULT '',  
  PRIMARY KEY (`id`)  
);  
/*!40101 SET character_set_client = @saved_cs_client */;  
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;
```

```
DROP TABLE IF EXISTS `w1x`;  
/*!40101 SET @saved_cs_client      = @@character_set_client */;  
/*!40101 SET character_set_client = utf8 */;  
CREATE TABLE `w1x` (  
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `f1` varchar(32) NOT NULL DEFAULT '',  
  `f2` varchar(40) NOT NULL DEFAULT '',  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=1822 DEFAULT CHARSET=utf8;  
/*!40101 SET character_set_client = @saved_cs_client */;  
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;
```

Replication Options



`--master-data[=value]`

SHOW MASTER STATUS

Used to dump a **master** replication server and the output includes a **CHANGE MASTER TO** statement that indicates the binary log coordinates (file name and position) of the dumped server

- ✓ 1: the statement will take effect when the dump file is reloaded
- ✓ 2: the statement is written as an SQL



`--dump-slave[=value]`

SHOW SLAVE STATUS

Similar to `--master-data` except that it is used to dump a replication **slave** server



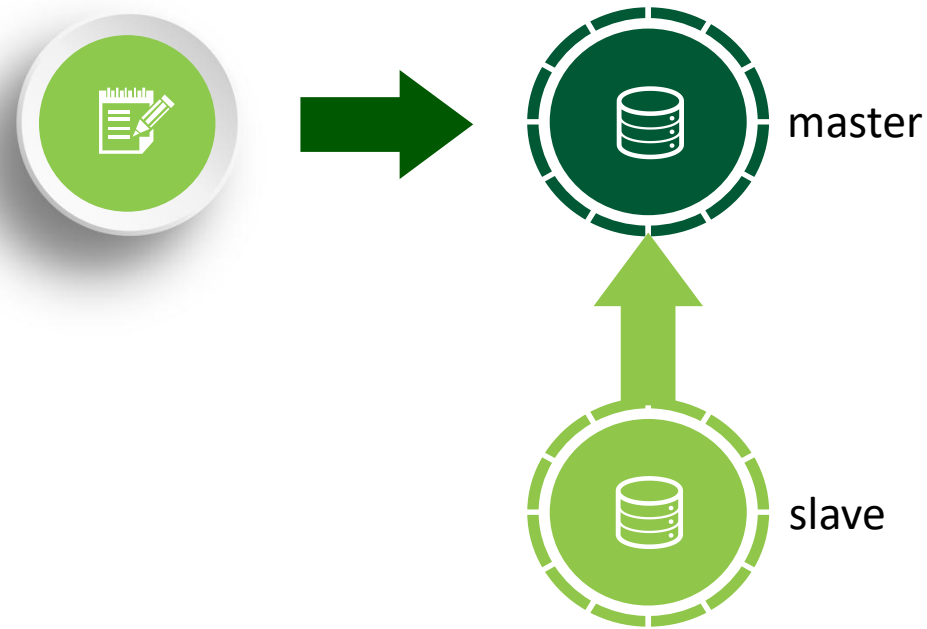
`--set-gtid-purged=value`

- ✓ OFF: "SET @@SESSION.SQL_LOG_BIN=0;" is not added to the output
- ✓ ON: "SET @@SESSION.SQL_LOG_BIN=0;" is added to the output
- ✓ AUTO: "SET @@SESSION.SQL_LOG_BIN=0;" is added to the output if GTIDs are enabled on the server you are backing up

Replication Issues

+ --set-gtid-purged=ON

- ✓ OFF: "SET @@SESSION.SQL_LOG_BIN=0;" is not added to the output
- ✓ ON: "SET @@SESSION.SQL_LOG_BIN=0;" is added to the output



Process



What about non-transaction Tables ?



Charset Issues

+ `mysqldump db1 t1 > t1.sql`

Do not use `'>'` or `'<'`, since it might screw up encoding



+ `"-r"` and `"source xxx.sql"`

The `-r` or, the same, `--result-file` option, will create the output in ASCII format

Goal: Client & Connection & Filesystem & MySQL & Databases & Tables all have exactly the same charset

02

mydumper/myloader

<https://github.com/maxbube/mydumper>
<https://launchpad.net/mydumper/0.9/0.9.1>

mydumper | myloader



Logical Backup



Multi-Threaded

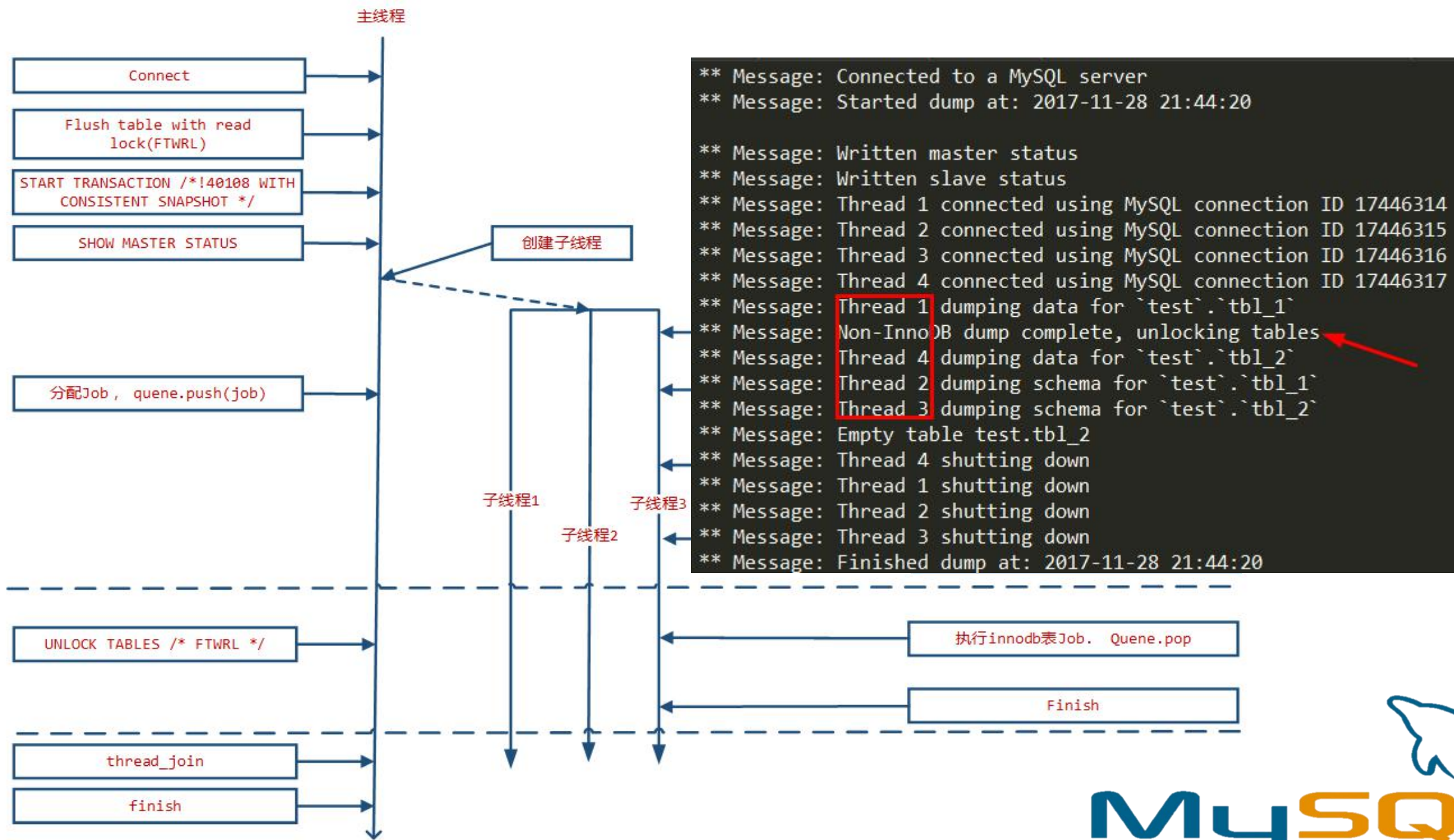


Output Compression(gzip)



Opensource & C Language

Process



Filtering Options : mydumper

- + -B, --database=dbname
- + -T, --tables-list=t1,t2
- + -x, --regex=db.table
- + -m, --no-schemas
- + -d, --no-data
- + -e, --build-empty-files
- + -c, --compress
- + -t, --threads
- + -o, --outputdir
- + -r, --rows | -F, --chunk-filesize
- + -G, --triggers & -E, --events & -R, --routines
- + -v, --verbose *[0=silent, 1=errors, 2=warnings, 3=info, default 2]*

```
# mydumper -B mydb -T t1,t2,t3,t4 -c -t 4 -v 3 -o /data/mydb_backup/
```

```
# mydumper --regex='^(?!mysql|test)' -c -t 4 -v 3 -o /data/mydb_backup/
```

```
# mydumper --regex=order.* -c -t 4 -v 3 -o /data/mydb_backup/
```

```
# mydumper -B mydb -T t1 -r 10000 -c -t 4 -v 3 -o /data/mydb_backup/
```

Mydumper Outputs

- + metadata ①
- + *dbname-schema-create.sql.gz*
- + *dbname.tablename-schema.sql.gz* & *dbname.tablename.sql.gz*

```
① # cat metadata
  Started dump at: 2017-10-14 17:51:50
  SHOW MASTER STATUS:
    Log: 3306-binlog.000001
    Pos: 622194819
    GTID:a3194045-e108-11e5-88ef-6c92bf0f0dd1:1-1161559

  SHOW SLAVE STATUS:
    Host: xx.xx.xx.xx
    Log: 3306-binlog.000001
    Pos: 643101999
    GTID:a3194045-e108-11e5-88ef-6c92bf0f0dd1:1-1161559
  Finished dump at: 2017-10-14 17:51:50
```

Filtering Options : myloader

+ -s, --source-db=from_db

+ -B, --database=to_db

+ -q, --queries-per-transaction

+ -t, --threads

+ -o, --overwrite-tables

+ -d, --directory

+ -v, --verbose=[0|1|2|3]

+ -e, --enable-binlog

```
# myloader -s mydb -B mydb_new -d /data/mydb_backup/ --enable-binlog  
--queries-per-transactio=1000 -t 4 -o -v 3
```

Issues

- + Option parsing failed: Error parsing option -o, try -help ①
- + Error while loading shared libraries: libmysqlclient.so.18 ②
- + /usr/lib64/libmysqlclient.so.18: no version information available ③
mydumper: symbol lookup error: mydumper: undefined symbol: g_malloc_n
- + Error writing file '/tmp/MLrwMd0b' (Errcode: 28 - No space left on device) ④

- ① You need add a **blank space** between parameter and value
- ② In -s /usr/local/mysql/lib/libmysqlclient.so.18 /usr/lib64/libmysqlclient.so.18
- ③ glib2-devel, mysql-devel, zlib-devel, pcre-devel, openssl-devel, gcc-c++, gcc, cmake
- ④ Increase **binlog_cache_size** or decrease **--queries-per-transaction & --threads**

03

innobackupex/xtraBackup

<https://www.percona.com/doc/percona-xtrabackup/LATEST/index.html>

innobackupex | xtraBackup



Physical Backup



Multi-Threaded



Streaming & Compression &
Encryption



Opensource

Component

+ xtrabackup

a compiled C binary, which copies only **InnoDB** and **XtraDB** data

+ innobackupex

a *perl* script that provides functionality to backup a whole MySQL database instance with **MyISAM**, **InnoDB**, and **XtraDB** tables

Warning

The **innobackupex** program is deprecated. Please switch to **xtrabackup**.

+ Xbcrypt

encrypting and decrypting backup files

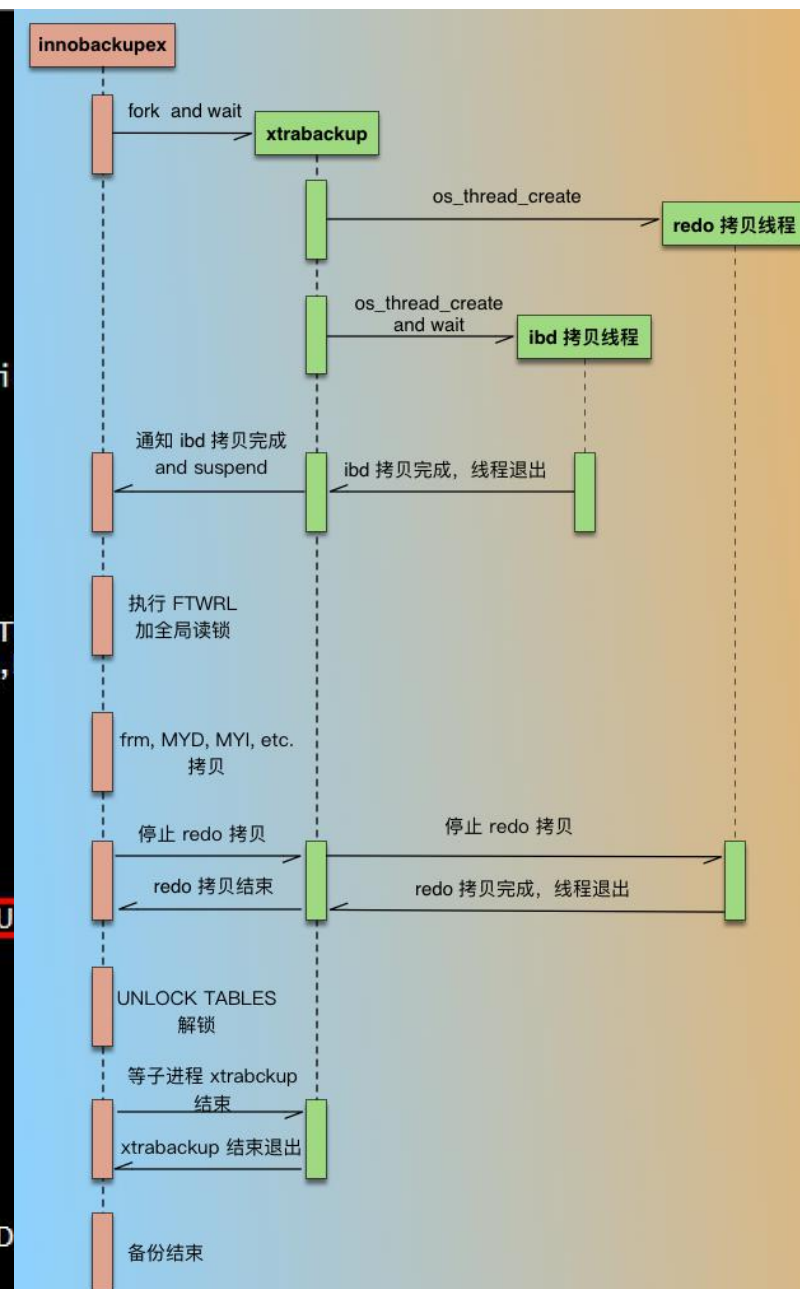
+ xbstream

streaming and extracting files to/from the xbstream format



Process

```
innobackupex_56: Created backup directory /data1/backup/mysql3306
171130 11:42:26 innobackupex_56: Starting ibbackup with command: xtrabackup ...
innobackupex_56: Suspend file '/data1/backup/mysql3306/xtrabackup_suspended_2'
xtrabackup: uses posix_fadvise().
xtrabackup: cd to /data1/mysql3306/
xtrabackup: open files limit requested 65535, set to 65535
xtrabackup: using the following InnoDB configuration: ...
xtrabackup: using O_DIRECT
[01] Copying /data1/mysql3306/ibdata1 to /data1/backup/mysql3306/ibdata1
[01] Copying ./mysql/slave_master_info.ibd to /data1/backup/mysql3306/mysql/slave_master_info.ibd
[01] Copying ./test/wlx.ibd to /data1/backup/mysql3306/test/wlx.ibd
xtrabackup: Creating suspend file '/data1/backup/mysql3306/xtrabackup_suspended_2'
171130 11:42:49 innobackupex_56: Continuing after ibbackup has suspended
171130 11:42:49 innobackupex_56: Executing FLUSH TABLES WITH READ LOCK on 5.5
171130 11:42:49 innobackupex_56: All tables locked and flushed to disk
171130 11:42:49 innobackupex_56: Starting to backup non-InnoDB tables and files
innobackupex_56: Backing up files '/data1/mysql3306//mysql/*.{frm,isl,MYD,MYI,MAD,MAI,MRG,TRG,T
innobackupex_56: Backing up files '/data1/mysql3306//performance_schema/*.{frm,isl,MYD,MYI,MAD,
innobackupex_56: Backing up file '/data1/mysql3306//test/db.opt'
innobackupex_56: Backing up file '/data1/mysql3306//test/wlx.frm'
innobackupex_56: Backing up file '/data1/mysql3306//test/wlx1.frm'
innobackupex_56: Backing up file '/data1/mysql3306//test/wlx1.MYI'
innobackupex_56: Backing up file '/data1/mysql3306//test/wlx1.MYD'
171130 11:42:55 innobackupex_56: Finished backing up non-InnoDB tables and files
171130 11:42:55 innobackupex_56: Failed to get master binlog coordinates from SHOW SLAVE STATUS
171130 11:42:55 innobackupex_56: Executing FLUSH NO_WRITE_TO_BINLOG ENGINE LOGS...
171130 11:42:55 innobackupex_56: Waiting for log copying to finish
xtrabackup: The latest check point (for incremental): '5052672850'
xtrabackup: Stopping log copying thread.
xtrabackup: Creating suspend file '/data1/backup/mysql3306/xtrabackup_log_copied'
xtrabackup: Transaction log of lsn (5052672683) to (5052672850) was copied.
171130 11:42:56 innobackupex_56: All tables unlocked
innobackupex_56: MySQL binlog position: filename '3306-binlog.000005', position 213917707, GTID
171130 11:42:56 innobackupex_56: Connection to database server closed
171130 11:42:56 innobackupex_56: completed OK!
```



Innobackupex: Full Backup

+ \$ innobackupex --defaults-file=/tmp/other-my.cnf /path/to/BACKUP-DIR

+ \$ innobackupex --apply-log --use-memory=4G /path/to/BACKUP-DIR

Innobackupex started the prepare process by reading the configuration from the *backup-my.cnf* and calling *xtrabackup --prepare* twice.

+ \$ innobackupex --copy-back /path/to/BACKUP-DIR

+ \$ chown -R mysql:mysql /data/mysql

Innobackupex: Incremental Backup

- + \$ innobackupex --defaults-file=/data/mysql/my.cnf /data/backups
- + \$ innobackupex --incremental /data/backups --incremental-basedir= *BASE-DIR*
- + \$ innobackupex --incremental /data/backups --incremental-basedir= *INC-DIR-1*
- + \$ innobackupex --apply-log --redo-only *BASE-DIR*
xtrabackup --prepare --apply-log-only
- + \$ innobackupex --apply-log --redo-only *BASE-DIR* --incremental-dir= *INC-DIR-1*
- + \$ innobackupex --apply-log *BASE-DIR* --incremental-dir= *INC-DIR-2*
- + \$ innobackupex --apply-log *BASE-DIR*
- + \$ innobackupex --copy-back *BASE-DIR*

Innobackupex: Partial Backups

- + `$ innobackupex --include='^mydatabase[.]mytable' /path/to/backup`
- + `$ innobackupex --tables-file=/tmp/tables.txt /path/to/backup`
- + `$ innobackupex --databases="mydatabase.mytable mysql" /path/to/backup`
- + `$ innobackupex --apply-log --export /path/to/partial/backup`

```
mysql> CREATE TABLE mytable (...) ENGINE=InnoDB;  
mysql> ALTER TABLE mydatabase.mytable DISCARD TABLESPACE;  
### copy .ibd、.exp (which contains InnoDB dictionary dump)  
mysql> ALTER TABLE mydatabase.mytable IMPORT TABLESPACE;
```

Innobackupex: Compression & Encryption

+ `--compress --compress-threads=#`

+ `--decompress --parallel=#`

Before proceeding you'll need to make sure that **qpress** has been installed

+ `--encrypt=AES128/AES192/AES256 --encrypt-key-file=KEYFILE`

+ `--encrypt-threads=# --encrypt-chunk-size=#`

```
$ openssl rand -out file -base64 24  
$ echo -n `openssl rand -base64 24` > file
```

+ `--decrypt=AES128/AES192/AES256 --encrypt-key=KEY --parallel=#`

Innobackupex: Streaming

+ \$ innobackupex --stream=**xbstream** ./ | ssh user@otherhost \
"xbstream -x -C /data/backups/"

+ \$ innobackupex --stream=**tar** ./ | ssh user@destination \
"cat - > /data/backups/backup.tar"

Streaming **tar** backups not support encryption/decryption, and will store the complete backup directly to a tar archive

When compression is enabled, xtrabackup compresses([quicklz](#)) all output data, except the meta and non-InnoDB files which are not compressed

```
$ innobackupex --parallel=8 --stream=xbstream --compress --compress-  
threads=8 --encrypt=AES256 --encrypt-key-file=./keyfile.txt --encrypt-threads=8  
/tmp/ | sudo -u wlx ssh wlx@xx.xx.xx.xx "xbcrypt -d -a AES256 -f  
/backup/mysql3306/keyfile.txt | xbstream -x -C /backup/mysql3306/"
```

04

mysqlbinlog

<https://dev.mysql.com/doc/refman/5.6/en/mysqlbinlog-backup.html>

mysqlbinlog



Binary log backup



Keep Alive until the
connection ends



Since MySQL 5.6

Filtering Options

+ --read-from-remote-server★

+ --raw★

+ --stop-never

+ --stop-never-slave-server-id=#

+ -r, --result-file=name

<u>--result-file</u> Option	Output File Names
<u>--result-file=x</u>	xbinlog.000999 and up
<u>--result-file=/tmp/</u>	/tmp/binlog.000999 and up
<u>--result-file=/tmp/x</u>	/tmp/xbinlog.000999 and up

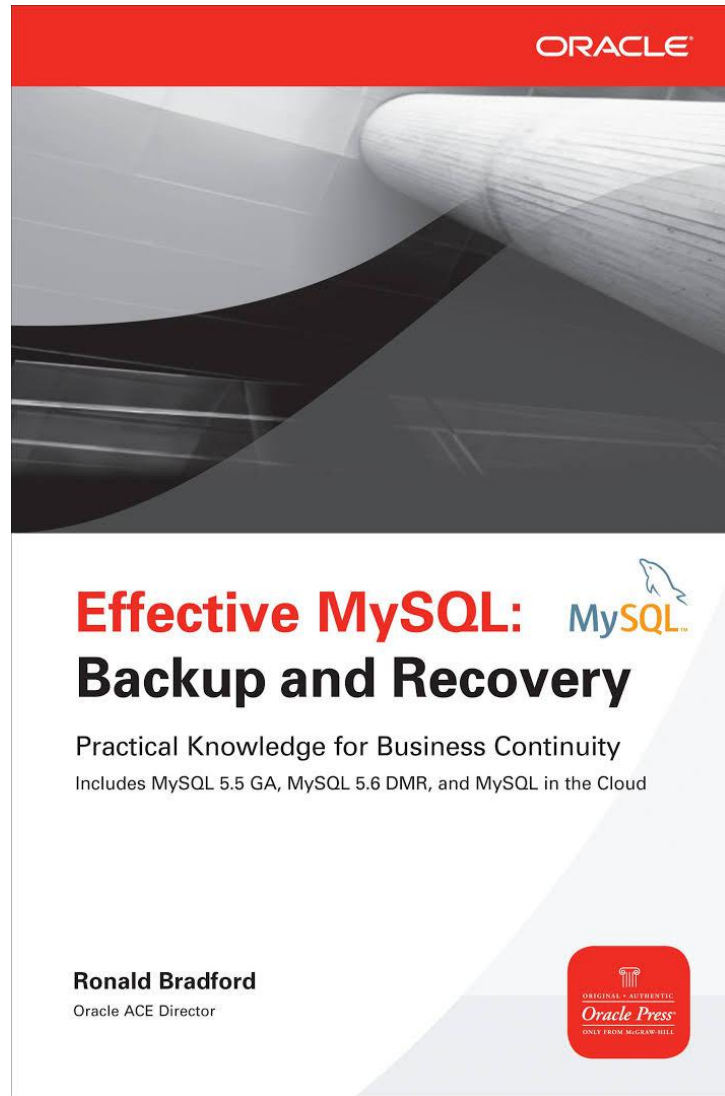
```
# mysqlbinlog --user=root --host=host_name --port=3306 --password=xxxx -  
-read-from-remote-server --raw --stop-never --result-file=/data/binlog/3306/  
binlog.000009
```

Contrast

Tools	Target	Model	Threads	Compression	Encryption	Free Space	Extra
mysqldump	data	logical	Single(Mul)	No(Yes)	no	yes	5.7
mydumper	data	logical	multi	yes	no	yes	
innobackupex	data	physical	multi	yes	yes	no	streaming
mysqlbinlog	binlog	logical	single	no	no		

Just enough

Resources



PERCONA
XTRABACKUP





Thanks for
Listening