

# Percona Xtrabackup From Zero to Hero

Percona Live Denver – May 2023

## Marcelo Altmann

- Senior Software Engineer @ Percona
- Working on Xtrabackup Project
- Author of key features
  - FIFO Datasink / xbcloud multi-thread
  - Memory Estimation
  - ZSTD Compression Support
  - Xbcloud exponential Backoff
  - Xbcloud instance profile
  - KMIP & KMS Keyring Component Integration



## Vinicius Grippa

- Senior Support Engineer @ Percona



# Content

[Before we start](#)

[Full Backup](#)

[Incremental Backup](#)

[Streaming](#)

[Bonus - Inspect xstream raw chunk](#)

[Encryption & Compression](#)

[Single Table Backup / Partial Restore](#)

[Scenario 1 - Backup and restore a single table](#)

[Scenario 2 - Restore a single table from a full backup](#)

[Point In Time Recovery](#)

[Restore process](#)

## Before we start

- Launch tutorial docker container:

Unset

```
docker run -d -t -i --privileged --name percona_live -p 9091:9091 \
  altmannmarcelo/pxb_pl:latest
```

- Check container is up and running:

Unset

```
docker logs -n 1 percona_live
```

We expect to see: **Everything is up and running.**

- Enter container ( launch at least two sessions):

Unset

```
docker exec -it percona_live bash
```

- MySQL **root** user is installed with **auth\_socket** (no password)
- There is a **run\_load.sh** script to simulate load during backups
- We will make usage of debug sync points:
  - Way of stopping the program at a specific point in the code:
  - Widely used on test suite to make certain conditions deterministically
  - Only available on Debug builds
  - **NOT SUITABLE FOR PRODUCTION**
  - Example:  
[https://github.com/percona/percona-xtrabackup/blob/percona-xtrabackup-8.0.32-26/storage/innobase/xtrabackup/src/backup\\_mysql.cc#L1663](https://github.com/percona/percona-xtrabackup/blob/percona-xtrabackup-8.0.32-26/storage/innobase/xtrabackup/src/backup_mysql.cc#L1663)

## Full Backup

- Check there is no employee named **Roy**:

Unset

```
mysql> SELECT * FROM employees.employees WHERE first_name = 'Roy'\G  
Empty set (0.09 sec)
```

- Start to simulate load:

Unset

```
run_load.sh run
```

- Take a full backup:

Unset

```
xtrabackup --backup --target-dir=/backups/full \  
--register-redo-log-consumer --debug_sync='log_status_get'
```

We should see the backup has been stopped at **log\_status\_get** sync point and a kill command to resume the backup:

```
2023-04-28T15:22:40.613320-00:00 0 [Note] [MY-011825] [Xtrabackup] Selecting LSN and  
binary log position from p_s.log_status  
2023-04-28T15:22:40.613388-00:00 0 [Note] [MY-011825] [Xtrabackup] DEBUG: Suspending  
at debug sync point 'log_status_get'Resume with 'kill -SIGCONT 173'
```

```
[1]+  Stopped                  xtrabackup --backup --target-dir=/backups/full  
--debug_sync='log_status_get'
```

- Check table employees has already been copied:

Unset

```
ls -l /backups/full/employees/employees.ibd
```

- Roy was hired. Add him to the database:

Unset

```
mysql> INSERT INTO employees.employees VALUES (500000, '1988-05-21',  
'Roy', 'Trenneman', 'M', '2023-05-22');
```

- Resume backup:

Unset

```
kill -SIGCONT 173
```

Wait for Xtrabackup to complete

- Stop load script
- Stop MySQL and remove datadir:

Unset

```
mysql -e shutdown  
rm -rf /var/lib/mysql
```

- Prepare backup

Unset

```
xtrabackup --prepare --target-dir=/backups/full
```

- Copy backup to datadir and adjust folder ownership:

Unset

```
xtrabackup --copy-back --target-dir=/backups/full  
chown -R mysql:mysql /var/lib/mysql
```

- Start MySQL

Unset

```
mysqld --user=mysql &
```

- Check if Roy is an employee:

Unset

```
mysql> SELECT * FROM employees.employees WHERE first_name = 'Roy'\G
***** 1. row *****
      emp_no: 500000
 birth_date: 1988-05-21
first_name: Roy
      last_name: Trenneman
        gender: M
      hire_date: 2023-05-22
1 row in set (0.09 sec)
```

## Incremental Backup

- Install page tracking component

Unset

```
mysql> INSTALL COMPONENT "file://component_mysqlbackup";
```

- Check there is no employee named **Maurice** nor **Jen**

Unset

```
mysql> SELECT * FROM employees.employees WHERE first_name = 'Maurice'
OR first_name = 'Jen'\G
Empty set (0.09 sec)
```

- Start to simulate load:

Unset

```
run_load.sh run
```

- Take a full backup

Unset

```
xtrabackup --backup --page-tracking --target-dir=/backups/inc_full
```

Wait for backup to complete

- Maurice was hired. Add him to the database:

Unset

```
mysql> INSERT INTO employees.employees VALUES (500001, '1988-05-21',  
'Maurice', 'Moss', 'M', '2023-05-22');
```

- Take the first incremental backup

Unset

```
xtrabackup --backup --page-tracking --target-dir=/backups/inc_1 \  
--incremental-basedir=/backups/inc_full
```

- Jen was hired. Add her to the database



Unset

```
mysql> INSERT INTO employees.employees VALUES (500002, '1988-05-21',  
'Jen', 'Barber', 'F', '2023-05-22');
```

- Take the second incremental backup

Unset

```
xtrabackup --backup --page-tracking --target-dir=/backups/inc_2 \  
--incremental-basedir=/backups/inc_1
```

- Stop load script
- Stop MySQL and remove datadir:

Unset

```
mysql -e shutdown  
rm -rf /var/lib/mysql
```

- Prepare **inc\_full** backup

Unset

```
xtrabackup --prepare --apply-log-only --target-dir=/backups/inc_full
```

- Prepare **inc\_1** backup

Unset

```
xtrabackup --prepare --apply-log-only --target-dir=/backups/inc_full \  
--incremental-dir=/backups/inc_1
```

- Prepare **inc\_2** backup

Unset

```
xtrabackup --prepare --target-dir=/backups/inc_full \  
--incremental-dir=/backups/inc_2
```

- Copy backup to datadir and adjust folder ownership:

Unset

```
xtrabackup --copy-back --target-dir=/backups/inc_full  
chown -R mysql:mysql /var/lib/mysql
```

- Start MySQL

Unset

```
mysqld --user=mysql &
```

- Check there is employee named **Maurice** nor **Jen**

Unset

```
mysql> SELECT * FROM employees.employees WHERE first_name = 'Maurice'  
OR first_name = 'Jen'\G
```

## Streaming

- Check MinIO console access: <http://127.0.0.1:9091> (U: **admin** P: **password**)

- Check S3 API access:

Unset

```
aws --endpoint-url http://127.0.0.1:9090 s3 ls
2023-05-03 00:59:18 perconalive
```

- Check there is no employee named **Douglas**

Unset

```
mysql> SELECT * FROM employees.employees WHERE first_name =
'Douglas'\G
Empty set (0.09 sec)
```

- Start to simulate load:

Unset

```
run_load.sh run
```

- Take a backup streaming to S3:

Unset

```
xtrabackup --backup --page-tracking --stream=xbstream \
--extra-lsdir=/backups/meta_full | xbcloud put --storage=s3 \
--s3-endpoint=http://127.0.0.1:9090 --s3-access-key=admin \
--s3-secret-key=password --s3-bucket=perconalive full_backup
```

- Douglas was hired. Add her to the database

Unset

```
mysql> INSERT INTO employees.employees VALUES (500003, '1988-05-21',  
'Douglas', 'Reynholm', 'M', '2023-05-22');
```

- Take an incremental backup streaming to S3:

Unset

```
xtrabackup --backup --stream=xbstream \  
--extra-lsdir=/backups/meta_inc \  
--incremental-basedir=/backups/meta_full | xbccloud put \  
--storage=s3 --s3-endpoint=http://127.0.0.1:9090 \  
--s3-access-key=admin --s3-secret-key=password \  
--s3-bucket=perconalive inc_backup
```

- Stop load script
- Stop MySQL and remove datadir:

Unset

```
mysql -e shutdown  
rm -rf /var/lib/mysql
```

- List folder on **perconalive** bucket:

Unset

```
aws --endpoint-url http://127.0.0.1:9090 s3 ls s3://perconalive  
      PRE full_backup/  
      PRE inc_backup/
```

- Create a folder, download and extract full backup

Unset

```
mkdir /backups/stream_full
xbcloud get --storage=s3 --s3-endpoint=http://127.0.0.1:9090 \
--s3-access-key=admin --s3-secret-key=password \
--s3-bucket=perconalive full_backup | xbstream -x -C \
/backups/stream_full
```

- Create a folder, download and extract full backup

Unset

```
mkdir /backups/stream_inc
xbcloud get --storage=s3 --s3-endpoint=http://127.0.0.1:9090 \
--s3-access-key=admin --s3-secret-key=password \
--s3-bucket=perconalive inc_backup | xbstream -x -C \
/backups/stream_inc
```

- Prepare full backup

Unset

```
xtrabackup --prepare --apply-log-only \
--target-dir=/backups/stream_full
```

- Prepare incremental backup

Unset

```
xtrabackup --prepare --target-dir=/backups/stream_full \
--incremental-dir=/backups/stream_inc
```

- Copy backup to datadir and adjust folder ownership:

Unset

```
xtrabackup --copy-back --target-dir=/backups/stream_full  
chown -R mysql:mysql /var/lib/mysql
```

- Start MySQL

Unset

```
mysqld --user=mysql &
```

- Check if Douglas is an employee:

Unset

```
mysql> SELECT * FROM employees.employees WHERE first_name =  
'Douglas'\G
```

## Bonus – Inspect xstream raw chunk

- Xstream layout is available at <https://bit.ly/xstream-format>
- Download xstream chunk

Unset

```
FILE=xtrabackup_checkpoints.00000000000000000000  
aws --endpoint-url http://127.0.0.1:9090 s3 cp \  
s3://perconalive/full_backup/${FILE} /tmp/${FILE}  
  
FILE=xtrabackup_checkpoints.00000000000000000001
```

```
aws --endpoint-url http://127.0.0.1:9090 s3 cp \  
s3://perconalive/full_backup/${FILE} /tmp/${FILE}
```

- Inspect Payload chunk

Unset

```
xbstream_viewer.sh /tmp/xtrabackup_checkpoints.00000000000000000000  
Magic: XBSTCK01  
Type: P (Payload)  
Path Length:          22  
Path: xtrabackup_checkpoints  
Payload Size:          137  
Payload Offset:        0  
Checksum: 2788090455  
*****Payload*****  
backup_type = full-backup  
from_lsn = 0  
to_lsn = 367403611  
last_lsn = 367403611  
flushed_lsn = 366904398  
redo_memory = 0  
redo_frames = 0
```

- Inspect End Of File chunk

Unset

```
xbstream_viewer.sh /tmp/xtrabackup_checkpoints.00000000000000000001  
Magic: XBSTCK01  
Type: E (End of File)  
Path Length:          22
```

Path: `xtrabackup_checkpoints`

## Encryption & Compression

**NOTE:** Encryption & compression are not tied together, you can only encrypt or only compress if you want to.

- Generate a random key to be used for encryption

Unset

```
echo -n $(openssl rand --base64 24) > /backups/enc_key
```

- Take a compressed(**ZSTD**) and encrypted (**AES256**) backup

Unset

```
xtrabackup --backup --encrypt=AES256 \  
--encrypt-key-file=/backups/enc_key --compress=zstd \  
--target-dir=/backups/enc_comp_full
```

- Attempt to read **xtrabackup\_info** from first backup example (`/backups/full/xtrabackup_info`)

Unset

```
cat /backups/full/xtrabackup_info
```



- Attempt to read **xtrabackup\_info.zst.xbcrypt** from encrypted backup

Unset

```
cat /backups/enc_comp_full/xtrabackup_info.zst.xbcrypt
```

- Decrypt and decompress the backup

Unset

```
xtrabackup --remove-original --decompress --decrypt=AES256 \  
--encrypt-key-file=/backups/enc_key \  
--target-dir=/backups/enc_comp_full/
```

- Stop MySQL and remove datadir:

Unset

```
mysql -e shutdown  
rm -rf /var/lib/mysql
```

- Prepare backup

Unset

```
xtrabackup --prepare --target-dir=/backups/enc_comp_full
```

- Copy backup to datadir and adjust folder ownership:

Unset

```
xtrabackup --copy-back --target-dir=/backups/enc_comp_full
```

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

- Start MySQL

Unset

```
mysqld --user=mysql &
```

## Single Table Backup / Partial Restore

### Scenario 1 – Backup and restore a single table

- Backup **employees** table only

Unset

```
xtrabackup --backup --tables=employees.employees \  
--target-dir=/backups/emp
```

- Accidentally, run a **DELETE** without **WHERE**

Unset

```
mysql> DELETE FROM employees.employees;  
-- panic moment --  
mysql> SELECT * FROM employees.employees;
```

- Prepare the backup

Unset

```
xtrabackup --prepare --export --target-dir=/backups/emp/
```

- Discard Tablespace and import backup files (keep the session open)

Unset

```
mysql> SET FOREIGN_KEY_CHECKS=0;  
mysql> ALTER TABLE employees.employees DISCARD TABLESPACE;
```

- Copy backup files to datadir

Unset

```
cp /backups/emp/employees/employees.* /var/lib/mysql/employees/  
chown mysql:mysql /var/lib/mysql/employees/employees.*
```

- One mysql session, import the tablespace

Unset

```
mysql> ALTER TABLE employees.employees IMPORT TABLESPACE;  
mysql> SET FOREIGN_KEY_CHECKS=1;  
mysql> SELECT * FROM employees.employees;  
-- relief moment --
```

## Scenario 2 – Restore a single table from a full backup

- Accidentally, run a **DELETE** without **WHERE**

Unset

```
mysql> DELETE FROM employees.employees;  
-- panic moment --  
mysql> SELECT * FROM employees.employees;
```

- On our first backup – export all tablespaces

Unset

```
xtrabackup --prepare --export --target-dir=/backups/full/
```

- Discard Tablespace and import backup files (keep the session open)

Unset

```
mysql> SET FOREIGN_KEY_CHECKS=0;  
mysql> ALTER TABLE employees.employees DISCARD TABLESPACE;
```

- Copy backup files to datadir

Unset

```
cp /backups/full/employees/employees.* /var/lib/mysql/employees/  
chown mysql:mysql /var/lib/mysql/employees/employees.*
```

- On the previously open mysql session, import the tablespace

Unset

```
mysql> ALTER TABLE employees.employees IMPORT TABLESPACE;  
mysql> SET FOREIGN_KEY_CHECKS=1;  
mysql> SELECT * FROM employees.employees;  
-- relief moment --
```

## Point In Time Recovery

**Note:** For PITR we need a copy of binary logs. Xtrabackup does not backup binary logs, you need to backup your binlogs. In our example we will use a local copy of binlogs. For production environments, please check [Using mysqlbinlog to Back Up Binary Log Files](#)

Note: We will be using the fake master approach. Check more details [here](#)

- Take a full backup

Unset

```
xtrabackup --backup --target-dir=/backups/pitr
```

- Employee Richmond was hired as Database Administrator with a salary of \$80k during his probationary period

Unset

```
mysql> INSERT INTO employees.employees VALUES (500005, '1988-05-21',  
'Richmond', 'Avenal', 'M', '2023-05-22');  
mysql> INSERT INTO employees.titles VALUES
```

```
(500005, 'Database Administrator', '2023-05-22', NULL);
mysql> INSERT INTO employees.salaries VALUES (500005, 80000,
'2023-05-22', '2023-08-21');
```

- Check current **Richmond** full record

Unset

```
mysql> SELECT emp_no, first_name, last_name, ANY_VALUE(title) AS
title, MAX(salary) AS salary FROM employees.employees JOIN
employees.titles USING (emp_no) JOIN employees.salaries USING (emp_no)
WHERE emp_no = 500005;
```

emp_no	first_name	last_name	title	salary
500005	Richmond	Avenal	Database Administrator	80000

- Simulate incident by running **incident.sh** at bash terminal

Unset

```
incident.sh
```

- **Richmond** has passed probation period and got a \$10k raise

Unset

```
mysql> INSERT INTO employees.salaries VALUES (500005, 90000,
'2023-08-22', '2024-08-21');
```

- Check current **Richmond** full record

Unset

```
mysql> SELECT emp_no, first_name, last_name, ANY_VALUE(title) AS  
title, MAX(salary) AS salary FROM employees.employees JOIN  
employees.titles USING (emp_no) JOIN employees.salaries USING (emp_no)  
WHERE emp_no = 500005;  
ERROR 1146 (42S02): Table 'employees.titles' doesn't exist
```

## Restore process

- Stop MySQL

Unset

```
mysql -e shutdown
```

- Prepare backup

Unset

```
xtrabackup --prepare --target-dir=/backups/pitr
```

- Check backup binlog coordinations

Unset

```
cat /backups/pitr/xtrabackup_binlog_info  
binlog.000017 157
```

- Copy all binlogs after **binlog.000017**

Unset

```
# cd /var/lib/mysql
# ls binlog.0*
binlog.000014 binlog.000015 binlog.000016 binlog.000017
# mkdir /backups/binlogs/
# cp binlog.000017 /backups/binlogs/
```

- Remove old datadir and prepare backup and copy backup to datadir

Unset

```
rm -rf /var/lib/mysql/*
xtrabackup --prepare --target-dir=/backups/pitr
xtrabackup --copy-back --target-dir=/backups/pitr
```

- Copy backup binlog, adjust relay index and folder ownership:

Unset

```
cp /backups/binlogs/binlog.000017 /var/lib/mysql/relay.000017
echo "relay.000017" > /var/lib/mysql/relay.index
chown -R mysql:mysql /var/lib/mysql
```

- Start MySQL with below configuration



Unset

```
mysqld --replicate-same-server-id --log-replica-updates=OFF \  
--skip-slave-start --relay-log=relay --relay_log_index=relay.index \  
--user=mysql &
```

- Inspect binlog, identify position of **DROP** statement

Unset

```
root@7dc31004f432:/var/lib/mysql# mysqlbinlog -vvvv  
--start-position=157 /var/lib/mysql/relay.000017 | grep -B 15 DROP  
#230506 17:50:48 server id 1  end_log_pos 1086 CRC32 0x5632eed7  Xid =  
289  
COMMIT/*!*/;  
# at 1086  
#230506 17:50:58 server id 1  end_log_pos 1163 CRC32 0x77ce2cc5  
Anonymous_GTID  last_committed=3 sequence_number=4      rbr_only=no  
original_committed_timestamp=1683395458393853  
immediate_commit_timestamp=1683395458393853 transaction_length=218  
# original_commit_timestamp=1683395458393853 (2023-05-06  
17:50:58.393853 UTC)  
# immediate_commit_timestamp=1683395458393853 (2023-05-06  
17:50:58.393853 UTC)  
/*!80001 SET  
@@session.original_commit_timestamp=1683395458393853/*!*/;  
/*!80014 SET @@session.original_server_version=80032/*!*/;  
/*!80014 SET @@session.immediate_server_version=80032/*!*/;  
SET @@SESSION.GTID_NEXT= 'ANONYMOUS'/*!*/;  
# at 1163  
#230506 17:50:58 server id 1  end_log_pos 1304 CRC32 0x3ef8eb55  Query  
thread_id=16      exec_time=0      error_code=0      Xid = 292  
use `employees`/*!*/;  
SET TIMESTAMP=1683395458/*!*/;
```

```
SET @@session.pseudo_thread_id=16/!*/;  
DROP TABLE `titles` /* generated by server */
```

On above example, DROP happened at position 1163, we will want to execute up to previous position ( #at 1086 )

- Check current **Richmond** full record

Unset

```
mysql> SELECT emp_no, first_name, last_name, ANY_VALUE(title) AS  
title, MAX(salary) AS salary FROM employees.employees JOIN  
employees.titles USING (emp_no) JOIN employees.salaries USING (emp_no)  
WHERE emp_no = 500005;
```

```
+-----+-----+-----+-----+-----+  
| emp_no | first_name | last_name | title | salary |  
+-----+-----+-----+-----+-----+  
|  NULL | NULL      | NULL      | NULL  |  NULL  |  
+-----+-----+-----+-----+-----+  
1 row in set (0.02 sec)
```

**Note:** No **Richmond** in the database yet

- Start replica up to the position of DROP

Unset

```
mysql> CHANGE REPLICATION SOURCE TO RELAY_LOG_FILE='relay.000017',  
RELAY_LOG_POS=157, SOURCE_HOST='dummy';  
mysql> START REPLICATION UNTIL RELAY_LOG_FILE = 'relay.000017',  
RELAY_LOG_POS=1086;
```

- Monitor **SHOW REPLICA STATUS\G** validate **Relay\_Log\_File** and **Relay\_Log\_Pos** has reached the desired position:

Unset

```
mysql> SHOW REPLICA STATUS\G
***** 1. row *****
      Replica_IO_State:
            Source_Host: dummy
            Source_User:
            Source_Port: 3306
            Connect_Retry: 60
            Source_Log_File:
      Read_Source_Log_Pos: 4
            Relay_Log_File: relay.000017
            Relay_Log_Pos: 1086
```

- Check current **Richmond** full record

Unset

```
mysql> SELECT emp_no, first_name, last_name, ANY_VALUE(title) AS
title, MAX(salary) AS salary FROM employees.employees JOIN
employees.titles USING (emp_no) JOIN employees.salaries USING (emp_no)
WHERE emp_no = 500005;
+-----+-----+-----+-----+-----+
| emp_no | first_name | last_name | title | salary |
+-----+-----+-----+-----+-----+
| 500005 | Richmond  | Avenal    | Database Administrator | 80000 |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Note: **Richmond** is now in the database, but with an old salary.

- Now we want to skip the DROP event and continue to apply the next event onwards

Unset

```
root@7dc31004f432:/var/lib/mysql# mysqlbinlog -vvvv
--start-position=1086 /var/lib/mysql/relay.000017 | grep -A10 DROP
DROP TABLE `titles` /* generated by server */
/*!*/;
# at 1304
#230506 17:51:04 server id 1  end_log_pos 1383 CRC32 0x5879d1ea
Anonymous_GTID  last_committed=4 sequence_number=5      rbr_only=yes
original_committed_timestamp=1683395464024775
immediate_commit_timestamp=1683395464024775 transaction_length=293
/*!50718 SET TRANSACTION ISOLATION LEVEL READ COMMITTED*/*!*/;
# original_commit_timestamp=1683395464024775 (2023-05-06
17:51:04.024775 UTC)
# immediate_commit_timestamp=1683395464024775 (2023-05-06
17:51:04.024775 UTC)
/*!80001 SET
@@session.original_commit_timestamp=1683395464024775*/*!*/;
/*!80014 SET @@session.original_server_version=80032*/*!*/;
/*!80014 SET @@session.immediate_server_version=80032*/*!*/;
SET @@SESSION.GTID_NEXT= 'ANONYMOUS'/*!*/;
```

Note: The next valid event after DROP is #at 1304

- Reconfigure replication to that event

Unset

```
mysql> CHANGE REPLICATION SOURCE TO RELAY_LOG_FILE='relay.000017',
RELAY_LOG_POS=1304, SOURCE_HOST='dummy';
mysql> START REPLICA;
```

- Re check current **Richmond** full record

Unset

```
mysql> SELECT emp_no, first_name, last_name, ANY_VALUE(title) AS  
title, MAX(salary) AS salary FROM employees.employees JOIN  
employees.titles USING (emp_no) JOIN employees.salaries USING (emp_no)  
WHERE emp_no = 500005;
```

```
+-----+-----+-----+-----+-----+  
| emp_no | first_name | last_name | title | salary |  
+-----+-----+-----+-----+-----+  
| 500005 | Richmond | Avenal | Database Administrator | 90000 |  
+-----+-----+-----+-----+-----+  
1 row in set (0.00 sec)
```

**Note:** For GTID based replication the process is similar, but instead of adjusting the relay log position we will add empty transactions with the GTIDs we want to skip – check [Skipping Transactions With GTIDs](#)

**Note:** You can even do selective PITR by using replication filters to apply events to a single table only

**Note:** You can tune PITR by using multiple parallel workers