Percona XtraBackup Documentation

8.2.0-1 (2023-12-11)

Percona Technical Documentation Team

Percona LLC and/or its affiliates © 2023 — Cookie Consent

Table of contents

Percona LLC and/or its affiliates © 2023 — Cookie Consent

Percona LLC and/or its affiliates © 2023 — Cookie Consent

1. Percona XtraBackup 8.2 Documentation

This documentation is for the latest release: Percona XtraBackup 8.2 (Release Notes). This is an Innovation release. This type of release is only supported for a short time and is designed to be used in an environment with fast upgrade cycles. Developers and DBAs are exposed to the latest features and improvements.

Percona XtraBackup is an open source hot backup utility for MySQL-based servers that keep your database fully available during planned maintenance windows.

Whether it is a 24x7 highly loaded server or a low-transaction-volume Percona XtraBackup is designed to make backups seamless without disrupting the server's performance in a production environment. Percona XtraBackup (PXB) is a 100% open source backup solution with commercial support available for organizations who want to benefit from comprehensive, responsive, and cost-flexible database support for MySQL.

This is an Innovation release. This type of release is only supported for a short time and is designed to be used in an environment with fast upgrade cycles. Developers and DBAs are exposed to the latest features and improvements.

1.1 Supported storage engines

Percona XtraBackup can back up data from InnoDB, XtraDB, MyISAM, MyRocks tables on MySQL 8.2 servers and Percona Server for MySQL with XtraDB, Percona Server for MySQL 8.2, and Percona XtraDB Cluster 8.2.

Percona XtraBackup 8.2 supports the MyRocks storage engine. An incremental backup on the MyRocks storage engine does not determine if an earlier full or incremental backup contains duplicate files. Percona XtraBackup copies all MyRocks files each time it takes a backup.

1.1.1 Limitations

Percona XtraBackup 8.2 does not support making backups of databases created in versions before 8.2 of MySQL, Percona Server for MySQL or Percona XtraDB Cluster.

1.2 Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-11-02

Percona LLC and/or its affiliates © 2023 — Cookie Consent

2. Release notes

2.1 Percona XtraBackup 8.2 release notes index

- Percona XtraBackup 8.2.0-1 (2023-12-11)

2.1.1 Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum **Get a Percona Expert**

Last update: 2023-11-07

2.2 Percona XtraBackup 8.2.0-1 (2023-12-11)

Get started with Quickstart Guide for Percona XtraBackup.

Percona XtraBackup (PXB) is a 100% open source backup solution for all versions of Percona Server for MySQL and MySQL® that performs online non-blocking, tightly compressed, highly secure full backups on transactional systems. Maintain fully available applications during planned maintenance windows with Percona XtraBackup.

This is an Innovation release. This type of release is only supported for a short time and is designed to be used in an environment with fast upgrade cycles. Developers and DBAs are exposed to the latest features and improvements. Patches and security fixes are included in the next Innovation release instead of a patch release or fix release within an Innovation release. To keep your environment current with the latest patches or security fixes, upgrade to the latest release.

2.2.1 Release highlights

Percona XtraBackup 8.2.0-1 is based on MySQL 8.2 and fully supports the Percona Server for MySQL 8.2 Innovation series and the MySQL 8.2 Innovation series. This release allows taking backups of Percona Server 8.2.0-1 and MySQL 8.2.

Use the Percona XtraBackup 8.0 series to take backups of Percona Server for MySQL 8.0.x or MySQL 8.0.x. Percona XtraBackup 8.2.0-1 does not take a backup of the Percona Server for MySQL 8.0 or the MySQL 8.0.x series.

2.2.2 Bug fixes

- PXB-3003: Percona XtraBackup discovering redo logs to parse and the server purging redo logs at the same time could cause a race condition.
- PXB-3147: XtraBackup failed to execute the `DO innodb_redo_log_consumer_register("PXB");` query.

Percona LLC and/or its affiliates © 2023 — Cookie Consent

- PXB-3168: Under high write load, the backup might fail with the “log block numbers mismatch” error.
- PXB-3173: The xbcloud binary was refactored to use multiple threads with a global list of files. This list did not allow users to download a subset of files.
- PXB-3181: The use of the `innodb-use-native-aio=true` option resulted in printing duplicate timestamps.

2.2.3 Useful links

Install Percona XtraBackup 8.2

The Percona XtraBackup GitHub repository

Download product binaries, packages, and tarballs at Percona Product Downloads

2.2.4 Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum **Get a Percona Expert**

Last update: 2023-11-07

Percona LLC and/or its affiliates © 2023 — Cookie Consent

3. Discover Percona XtraBackup

3.1 About Percona XtraBackup

Percona XtraBackup is the world's only open source, free MySQL hot backup software that performs non-blocking backups for InnoDB and XtraDB databases.

Percona XtraBackup has the following benefits:

- Complete a backup quickly and reliably
- Process transactions uninterrupted during a back up
- Save on disk space and network bandwidth
- Verify backup automatically

Percona XtraBackup makes hot backups for Percona Server for MySQL and MySQL, takes streaming, compressed, and incremental MySQL backups, and supports encryption.

Percona's enterprise-grade commercial MySQL Support contracts include support for Percona XtraBackup. We recommend support for critical production deployments.

3.1.1 Supported storage engines

Percona XtraBackup can back up data from InnoDB, XtraDB, MyISAM, and MyRocks tables on MySQL 8.2 servers as well as Percona Server for MySQL with XtraDB, and Percona XtraDB Cluster.

Percona XtraBackup supports the MyRocks storage engine. An incremental backup on the MyRocks storage engine does not determine if an earlier full or incremental backup contains the same files. Percona XtraBackup copies all MyRocks files each time it takes a backup. Percona XtraBackup does not support the TokuDB storage engine.

3.1.2 Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum **Get a Percona Expert**

Last update: 2023-10-12

Percona LLC and/or its affiliates © 2023 — Cookie Consent

3.2 Percona XtraBackup features

The following is a short list of the Percona XtraBackup features:

- Creates hot InnoDB backups without pausing your database
- Makes incremental backups of MySQL
- Streams compressed MySQL backups to another server
- Moves tables between MySQL servers on-line
- Creates new MySQL replication replicas easily
- Backs up MySQL without adding load to the server
- Performs throttling based on the number of IO operations per second
- Skips secondary index pages and recreates them when a compact backup is prepared
- Exports individual tables from a full InnoDB backup

Percona XtraBackup automatically uses backup locks, a lightweight alternative to `FLUSH TABLES WITH READ LOCK` available in Percona Server, to copy non-InnoDB data. This operation avoids blocking DML queries that modify InnoDB tables.

See also

How Percona XtraBackup works

3.2.1 Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-10-04

3.3 Limitations

Percona XtraBackup 8.2 does not support making backups of databases created in versions prior to 8.2 of MySQL, Percona Server for MySQL or Percona XtraDB Cluster.

Additional information

The InnoDB tables are locked while copying non-InnoDB data.

Percona LLC and/or its affiliates © 2023 — Cookie Consent

3.3.1 Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum **Get a Percona Expert**

Last update: 2023-10-12

3.4 How Percona XtraBackup works

Percona XtraBackup is based on InnoDB's crash-recovery functionality. It copies your InnoDB data files, which results in data that is internally inconsistent; but then it performs crash recovery on the files to make them a consistent, usable database again.

This works because InnoDB maintains a redo log, also called the transaction log. This contains a record of every change to InnoDB data. When InnoDB starts, it inspects the data files and the transaction log, and performs two steps. It applies committed transaction log entries to the data files, and it performs an undo operation on any transactions that modified data but did not commit.

The `--register-redo-log-consumer` parameter is disabled by default. When enabled, this parameter lets Percona XtraBackup register as a redo log consumer at the start of the backup. The server does not remove a redo log that Percona XtraBackup (the consumer) has not yet copied. The consumer reads the redo log and manually advances the log sequence number (LSN). The server blocks the writes during the process. Based on the redo log consumption, the server determines when it can purge the log.

Percona XtraBackup remembers the LSN when it starts, and then copies the data files. The operation takes time, and the files may change, then LSN reflects the state of the database at different points in time. Percona XtraBackup also runs a background process that watches the transaction log files, and copies any changes. Percona XtraBackup does this continually. The transaction logs are written in a round-robin fashion, and can be reused.

Percona XtraBackup uses Backup locks

where available as a lightweight alternative to `FLUSH TABLES WITH READ LOCK`. MySQL 8.2 allows acquiring an instance level backup lock via the `LOCK INSTANCE FOR BACKUP` statement.

Locking is only done for MyISAM and other non-InnoDB tables after Percona XtraBackup finishes backing up all InnoDB/XtraDB data and logs. Percona XtraBackup uses this automatically to copy non-InnoDB data to avoid blocking DML queries that modify InnoDB tables.

Important

The `BACKUP_ADMIN` privilege is required to query the `performance_schema_log_status` for either `LOCK INSTANCE FOR BACKUP` or `LOCK TABLES FOR BACKUP`.

Percona LLC and/or its affiliates © 2023 — Cookie Consent

xtrabackup tries to avoid backup locks and `FLUSH TABLES WITH READ LOCK` when the instance contains only InnoDB tables. In this case, xtrabackup obtains binary log coordinates from `performance_schema.log_status`. `FLUSH TABLES WITH READ LOCK` is still required in MySQL 8.2 when xtrabackup is started with the `--slave-info`. The `log_status` table in Percona Server for MySQL 8.2 is extended to include the relay log coordinates, so no locks are needed even with the `--slave-info` option.

See also

MySQL Documentation: LOCK INSTANCE FOR BACKUP

When backup locks are supported by the server, xtrabackup first copies InnoDB data, runs the `LOCK TABLES FOR BACKUP` and then copies the MyISAM tables. Once this is done, the backup of the files will begin. It will backup .frm, .MRG, .MYD, .MYI, .CSM, .CSV, `.sdi` and `.par` files.

After that xtrabackup will use `LOCK BINLOG FOR BACKUP` to block all operations that might change either binary log position or `Exec_Master_Log_Pos` or `Exec_Gtid_Set` (i.e. source binary log coordinates corresponding to the current SQL thread state on a replication replica) as reported by `SHOW MASTER/SLAVE STATUS`. xtrabackup will then finish copying the REDO log files and fetch the binary log coordinates. After this is completed xtrabackup will unlock the binary log and tables.

Finally, the binary log position will be printed to `STDERR` and xtrabackup will exit returning 0 if all went OK.

Note that the `STDERR` of xtrabackup is not written in any file. You will have to redirect it to a file, for example, `xtrabackup OPTIONS 2> backupout.log`.

It will also create the following files in the directory of the backup.

During the `prepare` phase, Percona XtraBackup performs crash recovery against the copied data files, using the copied transaction log file. After this is done, the database is ready to restore and use.

The backed-up MyISAM and InnoDB tables will be eventually consistent with each other, because after the prepare (recovery) process, InnoDB's data is rolled forward to the point at which the backup completed, not rolled back to the point at which it started. This point in time matches where the `FLUSH TABLES WITH READ LOCK` was taken, so the MyISAM data and the prepared InnoDB data are in sync.

The xtrabackup offers many features not mentioned in the preceding explanation. The functionality of each tool is explained in more detail further in this manual. In brief, though, the tools enable you to do operations such as streaming and incremental backups with various combinations of copying the data files, copying the log files, and applying the logs to the data.

3.4.1 Restoring a backup

To restore a backup with xtrabackup you can use the `--copy-back` or `--move-back` options.

xtrabackup will read from the `my.cnf` the variables datadir, innodb_data_home_dir, innodb_data_file_path, innodb_log_group_home_dir and check that the directories exist.

It will copy the MyISAM tables, indexes, etc. (.MRG, .MYD, .MYI, .CSM, .CSV, `.sdi`, and `par` files) first, InnoDB tables and indexes next and the log files at last. It will preserve file's attributes when copying them, you may have to change the files' ownership to `mysql` before starting the database server, as they will be owned by the user who created the backup.

Percona and/or its affiliates © 2023 — Cookie Consent

Alternatively, the `--move-back` option may be used to restore a backup. This option is similar to `--copy-back` with the only difference that instead of copying files it moves them to their target locations. As this option removes backup files, it must be used with caution. It is useful in cases when there is not enough free disk space to hold both data files and their backup copies.

3.4.2 Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum **Get a Percona Expert**

Last update: 2023-10-12

3.5 Understand version numbers

A version number identifies the innovtion product release. The product contains the latest features, improvements, and bug fixes at the time of that release.

8.2.0	-1
Base version	Minor build version

Percona uses semantic version numbering, which follows the pattern of base version and build version. Percona assigns unique, non-negative integers in increasing order for each version release. The version number combines the base MySQL 8.2 version number and the minor build version.

The version numbers for Percona XtraBackup 8.2.0-1 define the following information:

- Base version - the leftmost numbers indicate the MySQL 8.2 version used as a base. An increase in base version resets the minor build version to 0.
- Minor build version - an internal number that denotes the version of the software. A build version increases by one each time the Percona XtraBackup is released.

3.5.1 Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum **Get a Percona Expert**

Last update: 2023-10-17

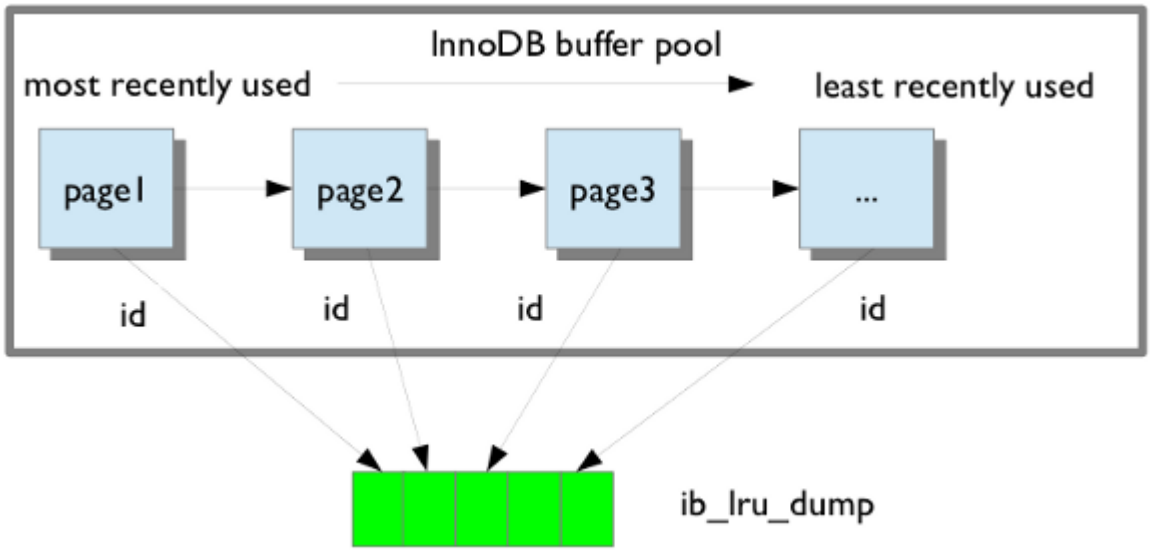
Percona LLC and/or its affiliates © 2023 — Cookie Consent

4. Features

4.1 Backup features

4.1.1 LRU dump backup

Percona XtraBackup includes a saved buffer pool dump into a backup to enable reducing the warm up time. It restores the buffer pool state from `ib_buffer_pool` file after restart. *Percona XtraBackup* discovers `ib_buffer_pool` and backs it up automatically.



If the buffer restore option is enabled in `my.cnf`, buffer pool will be in the warm state after backup is restored.

Find the information on how to save and restore the buffer pool dump in Saving and Restoring the Buffer Pool State.

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-10-17

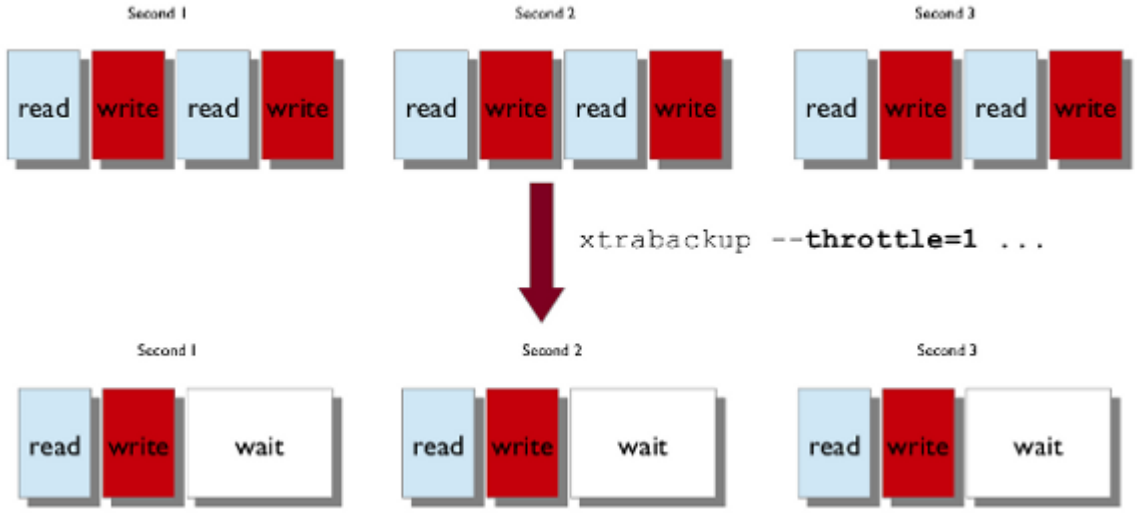
4.1.2 Throttling backups

Although *xtrabackup* does not block your database's operation, any backup can add load to the system being backed up. On systems that do not have much spare I/O capacity, it might be helpful

Percona and/or its affiliates © 2023 — Cookie Consent

to throttle the rate at which *xtrabackup* reads and writes data. You can do this with the `--throttle` option. This option limits the number of chunks copied per second. The chunk +size is *10 MB*.

The image below shows how throttling works when `--throttle` is set to `1`.



When specified with the `--backup` option, this option limits the number of pairs of read-and-write operations per second that *xtrabackup* will perform. If you are creating an incremental backup, then the limit is the number of read I/O operations per second.

By default, there is no throttling, and *xtrabackup* reads and writes data as quickly as it can. If you set too strict of a limit on the IOPS, the backup might be so slow that it will never catch up with the transaction logs that InnoDB is writing, so the backup might never complete.

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-06-12

4.1.3 Store backup history on the server

Percona XtraBackup supports storing the backups history on the server. Storing backup history on the server was implemented to provide users with additional information about backups that are being taken. Backup history information will be stored in the `PERCONA_SCHEMA.XTRABACKUP_HISTORY` table.

Percona and/or its affiliates © 2023 — Cookie Consent

To use this feature the following options are available:

- `--history` = : This option enables the history feature and allows the user to specify a backup series name that will be placed within the history record.
- `--incremental-history-name` = : This option allows an incremental backup to be made based on a specific history series by name. *xtrabackup* will search the history table looking for the most recent (highest `to_lsn`) backup in the series and take the `to_lsn` value to use as it's starting lsn. This is mutually exclusive with `--incremental-history-uuid`, `--incremental-basedir` and `--incremental-lsn` options. If no valid LSN can be found (no series by that name) *xtrabackup* will return with an error.
- `--incremental-history-uuid` = : Allows an incremental backup to be made based on a specific history record identified by UUID. *xtrabackup* will search the history table looking for the record matching UUID and take the `to_lsn` value to use as it's starting LSN. This options is mutually exclusive with `--incremental-basedir`, `--incremental-lsn` and `--incremental-history-name` options. If no valid LSN can be found (no record by that UUID or missing `to_lsn`), *xtrabackup* will return with an error.

Note

Backup that's currently being performed will **NOT** exist in the xtrabackup_history table within the resulting backup set as the record will not be added to that table until after the backup has been taken.

If you want access to backup history outside of your backup set in the case of some catastrophic event, you will need to either perform a `mysqldump`, partial backup or `SELECT` * on the history table after *xtrabackup* completes and store the results with you backup set.

For the necessary privileges, see Permissions and Privileges Needed.

PERCONA_SCHEMA.XTRABACKUP_HISTORY TABLE

This table contains the information about the previous server backups. Information about the backups will only be written if the backup was taken with `--history` option.

Column Name	Description
uuid	Unique backup id
name	User provided name of backup series. There may be multiple entries with the same name used to identify related backups in a series.
tool_name	Name of tool used to take backup
tool_command	Exact command line given to the tool with –password and –encryption_key obfuscated
tool_version	Version of tool used to take backup
ibbackup_version	Version of the xtrabackup binary used to take backup
server_version	Server version on which backup was taken
start_time	Time at the start of the backup
end_time	Time at the end of the backup
lock_time	Amount of time, in seconds, spent calling and holding locks for `FLUSH TABLES WITH READ LOCK`
binlog_pos	Binlog file and position at end of `FLUSH TABLES WITH READ LOCK`

Percona and/or its affiliates © 2023 — Cookie Consent

Column Name	Description
innodb_from_lsn	LSN at beginning of backup which can be used to determine prior backups
innodb_to_lsn	LSN at end of backup which can be used as the starting lsn for the next incremental
partial	Is this a partial backup, if `N` that means that it's the full backup
incremental	Is this an incremental backup
format	Description of result format (`xbstream`)
compact	Is this a compact backup
compressed	Is this a compressed backup
encrypted	Is this an encrypted backup

LIMITATIONS

- `--history` option must be specified only on the command line and not within a configuration file in order to be effective.
- `--incremental-history-name` and `--incremental-history-uuid` options must be specified only on the command line and not within a configuration file in order to be effective.

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-10-12

4.2 Prepare features

4.2.1 Dictionary cache

Percona XtraBackup is based on how [`crash recovery`] works. Percona XtraBackup copies the InnoDB data files, which results in data that is internally inconsistent; then the `prepare` phase performs crash recovery on the files to make a consistent, usable database again.

The `--prepare` phase has the following operations:

- Applies the redo log
- Applies the undo log

As a physical operation, the changes from the redo log modifications are applied to a page. In the redo log operation, there is no concept of row or transaction. The redo apply operation does not make the database consistent with a transaction. An uncommitted transaction can be flushed or written to the redo log by the server. Percona XtraBackup applies changes recorded in the redo log.

Percona and/or its affiliates © 2023 — Cookie Consent

Percona XtraBackup physically modifies a specific offset of a page within a tablespace (IBD file) when applying a redo log.

As a logical operation, Percona XtraBackup applies the undo log on a specific row. When the redo log completes, XtraBackup uses the undo log to roll back changes from uncommitted transactions during the backup.

Undo log

There are two types of undo log records:

- INSERT
- UPDATE

An undo log record contains a `table_id`. Percona XtraBackup uses this `table_id` to find the table definition, and then uses that information to parse the records on an index page. The transaction rollback reads the undo log records and applies the changes.

After initializing the data dictionary engine and the data dictionary cache, the storage engine can request the `table_id` and uses this ID to fetch the table schema. An index search tuple (key) is created from the table schema and key fields from the undo log record. The server finds the record using the search tuple (key) and performs the undo operation.

For example, InnoDB uses the `table_id` (also known as the `se_private_id`) for a table definition. Percona XtraBackup does not behave like a server and does not have access to the data dictionary. XtraBackup initializes the InnoDB engine and uses the `InnoDB table object` (`dict_table_t`) when needed. XtraBackup relies on Serialized Dictionary Information (SDI) that is stored in the tablespace. SDI is a JSON representation of a table.

In Percona XtraBackup 8.2.0-1, tables are loaded as `evictable`. XtraBackup scans the B-tree index of the data dictionary tables `mysql.indexes` and `mysql.index_partitions` to establish a relationship between the `table_id` and the `tablespace(space_id)`. XtraBackup uses this relationship during transaction rollback. XtraBackup does not load user tables unless there is a transaction rollback on them.

A background thread or a Percona XtraBackup main thread handles the cache eviction when the cache size limit is reached.

This design provides the following benefits during the `--prepare` phase:

- Uses less memory
- Uses less IO
- Improves the time taken in the `--prepare` phase
- Completes successfully, even if the `--prepare` phase has a huge number of tables
- Completes the Percona XtraDB Cluster SST process faster

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Percona and/or its affiliates © 2023 — Cookie Consent

Community Forum Get a Percona Expert

Last update: 2023-10-12

4.3 Restore features

4.3.1 Point-in-time recovery

Recovering up to particular moment in database's history can be done with xtrabackup and the binary logs of the server.

Note that the binary log contains the operations that modified the database from a point in the past. You need a full datadir as a base, and then you can apply a series of operations from the binary log to make the data match what it was at the point in time you want.

```
$ xtrabackup --backup --target-dir=/path/to/backup
$ xtrabackup --prepare --target-dir=/path/to/backup
```

For more details on these procedures, see Creating a backup and Preparing a backup.

Now, suppose that some time has passed, and you want to restore the database to a certain point in the past, having in mind that there is the constraint of the point where the snapshot was taken.

To find out what is the situation of binary logging in the server, execute the following queries:

```
mysql> SHOW BINARY LOGS;
```

Expected output

```
+------------------+-----------+
| Log_name         | File_size |
+------------------+-----------+
| mysql-bin.000001 |       126 |
| mysql-bin.000002 |      1306 |
| mysql-bin.000003 |       126 |
| mysql-bin.000004 |       497 |
+------------------+-----------+
```

and

```
mysql> SHOW MASTER STATUS;
```


Percona and/or its affiliates © 2023 — Cookie Consent

Expected output

```
+------------------+----------+--------------+------------------+
| File             | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+------------------+----------+--------------+------------------+
| mysql-bin.000004 |      497 |              |                  |
+------------------+----------+--------------+------------------+
```

The first query will tell you which files contain the binary log and the second one which file is currently being used to record changes, and the current position within it. Those files are stored usually in the datadir (unless other location is specified when the server is started with the `--log-bin=` option).

To find out the position of the snapshot taken, see the `xtrabackup_binlog_info` at the backup's directory:

```
$ cat /path/to/backup/xtrabackup_binlog_info
```

Expected output

```
mysql-bin.000003     57
```

This will tell you which file was used at moment of the backup for the binary log and its position. That position will be the effective one when you restore the backup:

```
$ xtrabackup --copy-back --target-dir=/path/to/backup
```

As the restoration will not affect the binary log files (you may need to adjust file permissions, see Restoring a Backup), the next step is extracting the queries from the binary log with mysqlbinlog starting from the position of the snapshot and redirecting it to a file

```
$ mysqlbinlog /path/to/datadir/mysql-bin.000003 /path/to/datadir/mysql-bin.000004 \
    --start-position=57 > mybinlog.sql
```

Note that if you have multiple files for the binary log, as in the example, you have to extract the queries with one process, as shown above.

Inspect the file with the queries to determine which position or date corresponds to the point-in-time wanted. Once determined, pipe it to the server. Assuming the point is `11-12-25 01:00:00`:

```
$ mysqlbinlog /path/to/datadir/mysql-bin.000003 /path/to/datadir/mysql-bin.000004 \
    --start-position=57 --stop-datetime="11-12-25 01:00:00" | mysql -u root -p
```

and the database will be rolled forward up to that Point-In-Time.

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Percona and/or its affiliates © 2023 — Cookie Consent

Community Forum Get a Percona Expert

Last update: 2023-08-17

4.3.2 Restore individual tables

Percona XtraBackup can export a table that is contained in its own .ibd file. With Percona XtraBackup, you can export individual tables from any InnoDB database, and import them into Percona Server for MySQL with XtraDB or MySQL 8.2. The source doesn't have to be XtraDB or MySQL 8.2, but the destination does. This method only works on individual .ibd files.

The following example exports and imports the following table:

```
CREATE TABLE export_test (
a int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Export the table

To generate an .ibd file in the target directory, create the table using the `innodb_file_per_table` mode:

```
$ find /data/backups/mysql/ -name export_test.*
/data/backups/mysql/test/export_test.ibd
```

During the `--prepare` step, add the `--export` option to the command. For example:

```
$ xtrabackup --prepare --export --target-dir=/data/backups/mysql/
```

When restoring an encrypted InnoDB tablespace table, add the keyring file:

```
$ xtrabackup --prepare --export --target-dir=/tmp/table \
--keyring-file-data=/var/lib/mysql-keyring/keyring
```

The following files are the only files required to import the table into a server running Percona Server for MySQL with XtraDB or MySQL 8.2. If the server uses InnoDB Tablespace Encryption, add the .cfp file, which contains the transfer key and an encrypted tablespace key.

The files are located in the target directory:

```
/data/backups/mysql/test/export_test.ibd
/data/backups/mysql/test/export_test.cfg
```

Import the table

On the destination server running Percona Server for MySQL with XtraDB or MySQL 8.2, create a table with the same structure, and then perform the following steps:

Percona and/or its affiliates © 2023 — Cookie Consent

0. Run the `ALTER TABLE test.export_test DISCARD TABLESPACE;` command. If you see the following error message:

   Error message

   ```
   ERROR 1809 (HY000): Table 'test/export_test' in system tablespace
   ```

   enable `innodb_file_per_table` option on the server and create the table again.

   ```
   $ set global innodb_file_per_table=ON;
   ```

0. Copy the exported files to the `test/` subdirectory of the destination server's data directory.

0. Run `ALTER TABLE test.export_test IMPORT TABLESPACE;`

   The table is imported, and you can run a `SELECT` to see the imported data.

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-10-12

4.4 Encrypted InnoDB tablespace backups

InnoDB supports data encryption for InnoDB tables stored in file-per-table tablespaces. This feature provides an at-rest encryption for physical tablespace data files.

For an authenticated user or an application to access an encrypted F tablespace, InnoDB uses the master encryption key to decrypt the tablespace key. The master encryption key is stored in a keyring.

Percona XtraBackup supports the following keyring components and plugins:

- keyring_vault component
- keyring_file plugin
- keyring_file component
- Key Management Interoperability Protocol (KMIP)
- Amazon Key Management Service (AWS KMS)

These components are stored in the `plugin` directory.

Percona and/or its affiliates © 2023 — Cookie Consent

Note

Enable only one keyring plugin or one keyring component at a time for each server instance. Enabling multiple keyring plugins or keyring components or mixing keyring plugins or keyring components is not supported and may result in data loss.

4.4.1 Use the keyring vault component

The `keyring_vault` component extends the server capabilities. The server uses a manifest to load the component and the component has its own configuration file.

The following example is a global manifest file that does not use local manifests:

```
{
 "read_local_manifest": false,
 "components": "file:///component_keyring_vault"
}
```

The following is an example of a global manifest file that points to a local manifest file:

```
{
 "read_local_manifest": true
}
```

The following is an example of a local manifest file:

```
{
 "components": "file:///component_keyring_vault"
}
```

The configuration settings are either in a global configuration file or a local configuration file.

Example of a configuration file in JSON format

```
{
 "timeout": 15,
 "vault_url": "https://vault.public.com:8202",
 "secret_mount_point": "secret",
 "secret_mount_point_version": "AUTO",
 "token": "58a20c08-8001-fd5f-5192-7498a48eaf20",
 "vault_ca": "/data/keyring_vault_confs/vault_ca.crt"
}
```

Find more information on configuring the `keyring_vault` component in Use the keyring vault component.

The component has no special requirements for backing up a database that contains encrypted InnoDB tablespaces.

The following command creates a backup in the `/data/backup` directory:

Percona and/or its affiliates © 2023 — Cookie Consent

```
$ xtrabackup --backup --target-dir=/data/backup --user=root
```

After xtrabackup completes the action, the following message confirms the action:

Confirmation message

```
xtrabackup: Transaction log of lsn (5696709) to (5696718) was copied.
160401 10:25:51 completed OK!
```

Prepare the backup with the keyring vault component

To prepare the backup, xtrabackup must access the keyring. xtrabackup does not communicate with the MySQL server or read the default `my.cnf` configuration file. Specify the keyring settings in the command line:

```
$ xtrabackup --prepare --target-dir=/data/backup --component-keyring-config==/etc/
vault.cnf
```

After xtrabackup completes the action, the following message confirms the action:

Confirmation message

```
InnoDB: Shutdown completed; log sequence number 5697064
160401 10:34:28 completed OK!
```

Restore the backup

As soon as the backup is prepared, you can restore it with the `--copy-back` option:

```
$ xtrabackup --copy-back --target-dir=/data/backup --datadir=/data/mysql \
--transition-key=MySecretKey --generate-new-master-key \
--component-keyring-config=/etc/vault.cnf
```

4.4.2 Use the keyring file plugin

Warning

The `keyring_file` plugin should not be used for regulatory compliance.

In order to back up and prepare a database containing encrypted InnoDB tablespaces, specify the path to a keyring file as the value of the `--keyring-file-data` option.

```
$ xtrabackup --backup --target-dir=/data/backup/ --user=root \
--keyring-file-data=/var/lib/mysql-keyring/keyring
```

After xtrabackup takes the backup, the following message confirms the action:

Percona LLC and/or its affiliates © 2023 — Cookie Consent

Confirmation message

```
xtrabackup: Transaction log of lsn (5696709) to (5696718) was copied.
160401 10:25:51 completed OK!
```

Warning

xtrabackup does not copy the keyring file into the backup directory. To prepare the backup, you must copy the keyring file manually.

Prepare the backup with the keyring file plugin

To prepare the backup specify the keyring-file-data.

```
$ xtrabackup --prepare --target-dir=/data/backup \
--keyring-file-data=/var/lib/mysql-keyring/keyring
```

After xtrabackup takes the backup, the following message confirms the action:

Confirmation message

```
InnoDB: Shutdown completed; log sequence number 5697064
160401 10:34:28 completed OK!
```

The backup is now prepared and can be restored with the `--copy-back` option. In case the keyring has been rotated, you must restore the keyring used when the backup was taken and prepared.

4.4.3 Use the keyring file component

The `keyring_file` component is part of the component-based MySQL infrastructure which extends the server capabilities.

The server uses a manifest to load the component and the component has its own configuration file. See the MySQL documentation on the component installation for more information.

An example of a manifest and a configuration file follows:

An example of `./bin/mysqld.my`:

```
{
   "components": "file://component_keyring_file"
}
```

An example of `/lib/plugin/component_keyring_file.cnf`:

```
{
   "path": "/var/lib/mysql-keyring/keyring_file", "read_only": false
}
```


Percona and/or its affiliates © 2023 — Cookie Consent

For more information, see Keyring Component Installation and Using the keyring_file File-Based Keyring Plugin.

With the appropriate privilege, you can `SELECT` on the performance_schema.keyring_component_status table to view the attributes and status of the installed keyring component when in use.

The component has no special requirements for backing up a database that contains encrypted InnoDB tablespaces.

```
$ xtrabackup --backup --target-dir=/data/backup --user=root
```

After xtrabackup completes the action, the following message confirms the action:

Confirmation message

```
xtrabackup: Transaction log of lsn (5696709) to (5696718) was copied.
160401 10:25:51 completed OK!
```

Warning

xtrabackup does not copy the keyring file into the backup directory. To prepare the backup, you must copy the keyring file manually.

Prepare the backup with the keyring_file component

xtrabackup reads the `keyring_file` component configuration from `xtrabackup_component_keyring_file.cnf`. You must specify the keyring_file data path if the `--component-keyring-config` is not located in the attribute `PATH` from the `xtrabackup_component_keyring_file.cnf`.

The following is an example of adding the location for the –component-keyring-config:

```
$ xtrabackup --prepare --target-dir=/data/backup \
--component-keyring-config=/var/lib/mysql-keyring/keyring
```

xtrabackup attempts to read `xtrabackup_component_keyring_file.cnf`. You can assign another keyring file component configuration by passing the `--component-keyring-config` option.

After xtrabackup completes preparing the backup, the following message confirms the action:

Confirmation message

```
InnoDB: Shutdown completed; log sequence number 5697064
160401 10:34:28 completed OK!
```

The backup is prepared. To restore the backup use the `--copy-back` option. If the keyring has been rotated, you must restore the specific keyring used to take and prepare the backup.

Percona LLC and/or its affiliates © 2023 — Cookie Consent

4.4.4 Incremental encrypted InnoDB tablespace backups with keyring file plugin

The process of taking incremental backups with InnoDB tablespace encryption is similar to taking the Incremental Backups with unencrypted tablespace.

4.4.5 Create an incremental backup

To make an incremental backup, begin with a full backup. The xtrabackup binary writes a file called `xtrabackup_checkpoints` into the backup's target directory. This file contains a line showing the `to_lsn`, which is the database's LSN at the end of the backup. First you need to create a fullbackup with the following command:

```
$ xtrabackup --backup --target-dir=/data/backups/base \
--keyring-file-data=/var/lib/mysql-keyring/keyring
```

In order to prepare the backup, you must make a copy of the keyring file yourself. xtrabackup does not copy the keyring file into the backup directory. Restoring the backup after the keyring has been changed causes errors like `ERROR 3185 (HY000): Can't find master key from keyring, please check keyring plugin is loaded.` when the restore process tries accessing an encrypted table.

If you look at the `xtrabackup_checkpoints` file, you should see the output similar to the following:

Expected output

```
backup_type = full-backuped
from_lsn = 0
to_lsn = 7666625
last_lsn = 7666634
compact = 0
recover_binlog_info = 1
```

Now that you have a full backup, you can make an incremental backup based on it. Use a command such as the following:

```
$ xtrabackup --backup --target-dir=/data/backups/inc1 \
--incremental-basedir=/data/backups/base \
--keyring-file-data=/var/lib/mysql-keyring/keyring
```

To prepare the backup, you must copy the keyring file manually. xtrabackup does not copy the keyring file into the backup directory.

If the keyring has not been rotated you can use the same as the one you've backed-up with the base backup. If the keyring has been rotated, or you have upgraded the plugin to a component, you must back up the keyring file. Otherwise, you cannot prepare the backup.

The `/data/backups/inc1/` directory should now contain delta files, such as `ibdata1.delta` and `test/table1.ibd.delta`. These represent the changes since the `LSN 7666625`. If you examine the `xtrabackup_checkpoints` file in this directory, you should see the output similar to the following:

Percona LLC and/or its affiliates © 2023 — Cookie Consent

Expected output

```
backup_type = incremental
from_lsn = 7666625
to_lsn = 8873920
last_lsn = 8873929
compact = 0
recover_binlog_info = 1
```

Use this directory as the base for yet another incremental backup:

```
$ xtrabackup --backup --target-dir=/data/backups/inc2 \
--incremental-basedir=/data/backups/inc1 \
--keyring-file-data=/var/lib/mysql-keyring/keyring
```

Prepare incremental backups

The `--prepare` step for incremental backups is not the same as for normal backups. In normal backups, two types of operations are performed to make the database consistent: committed transactions are replayed from the log file against the data files, and uncommitted transactions are rolled back. You must skip the rollback of uncommitted transactions when preparing a backup, because transactions that were uncommitted at the time of your backup may be in progress, and it's likely that they will be committed in the next incremental backup. You should use the `--apply-log-only` option to prevent the rollback phase.

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

Beginning with the full backup you created, you can prepare it and then apply the incremental differences to it. Recall that you have the following backups:

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

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

```
$ xtrabackup --prepare --apply-log-only --target-dir=/data/backups/base \
--keyring-file-data=/var/lib/mysql-keyring/keyring
```

Expected output

```
InnoDB: Shutdown completed; log sequence number 7666643
InnoDB: Number of pools: 1
160401 12:31:11 completed OK!
```

To apply the first incremental backup to the full backup, you should use the following command:

Percona LLC and/or its affiliates © 2023 — Cookie Consent

```
$ xtrabackup --prepare --apply-log-only --target-dir=/data/backups/base \
--incremental-dir=/data/backups/inc1 \
--keyring-file-data=/var/lib/mysql-keyring/keyring
```

The backup should be prepared with the keyring file and type that was used when backup was being taken. This means that if the keyring has been rotated, or you have upgraded from a plugin to a component between the base and incremental backup that you must use the keyring that was in use when the first incremental backup has been taken.

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 \
--keyring-file-data=/var/lib/mysql-keyring/keyring
```

Use `--apply-log-only` when merging all incremental backups except the last one. That's why the previous line does not contain the `--apply-log-only` option. Even if the `--apply-log-only` was used on the last step, backup would still be consistent but in that case server would perform the rollback phase.

The backup is now prepared and can be restored with `--copy-back` option. In case the keyring has been rotated you'll need to restore the keyring which was used to take and prepare the backup.

4.4.6 Restore a backup when the keyring is not available

While the described restore method works, this method requires access to the same keyring that the server is using. It may not be possible if the backup is prepared on a different server or at a much later time, when keys in the keyring are purged, or, in the case of a malfunction, when the keyring vault server is not available at all.

The `--transition-key=<passphrase>` option should be used to make it possible for xtrabackup to process the backup without access to the keyring vault server. In this case, xtrabackup derives the AES encryption key from the specified passphrase and will use it to encrypt tablespace keys of tablespaces that are being backed up.

Create a backup with a passphrase

The following example illustrates how the backup can be created in this case:

```
$ xtrabackup --backup --user=root -p --target-dir=/data/backup \
--transition-key=MySecretKey
```

If `--transition-key` is specified without a value, xtrabackup will ask for it.

xtrabackup scrapes `--transition-key` so that its value is not visible in the `ps` command output.

Prepare a backup with a passphrase

The same passphrase should be specified for the prepare command:

Percona LLC and/or its affiliates © 2023 — Cookie Consent

```
$ xtrabackup --prepare --target-dir=/data/backup \
--transition-key=MySecretKey
```

There are no `--keyring-vault...`,``–keyring-file...``, or `--component-keyring-config` options here, because xtrabackup does not talk to the keyring in this case.

Restore a backup with a generated key

When restoring a backup you need to generate a new master key.

The example for `keyring_file` plugin:

```
$ xtrabackup --copy-back --target-dir=/data/backup --datadir=/data/mysql \
--transition-key=MySecretKey --generate-new-master-key \
--keyring-file-data=/var/lib/mysql-keyring/keyring
```

The example for `keyring_file` component:

```
$ xtrabackup --copy-back --target-dir=/data/backup --datadir=/data/mysql \
--transition-key=MySecretKey --generate-new-master-key \
--component-keyring-config=/var/lib/mysql-keyring/keyring
```

The example for `keyring_vault` component:

```
$ xtrabackup --copy-back --target-dir=/data/backup --datadir=/data/mysql \
--transition-key=MySecretKey --generate-new-master-key \
--component-keyring-config=/etc/vault.cnf
```

xtrabackup generates a new master key, stores it in the target keyring vault server and re-encrypts the tablespace keys using this key.

Make a backup with a stored transition key

Finally, there is an option to store a transition key in the keyring. In this case, xtrabackup will need to access the same keyring file or vault server during prepare and copy-back but does not depend on whether the server keys have been purged.

In this scenario, the three stages of the backup process look as follows.

- Back up

Percona and/or its affiliates © 2023 — Cookie Consent

```
$ xtrabackup --backup --user=root -p --target-dir=/data/backup \
--generate-transition-key
```

- Prepare
  - `keyring_file` plugin variant:

```
$ xtrabackup --prepare --target-dir=/data/backup \
--keyring-file-data=/var/lib/mysql-keyring/keyring
```

  - `keyring_file` component variant:

```
$ xtrabackup --prepare --target-dir=/data/backup \
--component-keyring-config=/var/lib/mysql-keyring/keyring
```

  - `keyring_vault` component variant:

```
$ xtrabackup --prepare --target-dir=/data/backup \
--component-keyring-config=/etc/vault.cnf
```

- Copy-back
  - `keyring_file` plugin variant:

```
$ xtrabackup --copy-back --target-dir=/data/backup --datadir=/data/mysql \
--generate-new-master-key --keyring-file-data=/var/lib/mysql-keyring/keyring
```

  - `keyring_file` component variant:

```
$ xtrabackup --copy-back --target-dir=/data/backup --datadir=/data/mysql \
--generate-new-master-key --component-keyring-config=/var/lib/mysql-keyring/keyring
```

  - `keyring_vault` component variant:

```
$ xtrabackup --copy-back --target-dir=/data/backup --datadir=/data/mysql \
--generate-new-master-key --component-keyring-config=/etc/vault.cnf
```

4.4.7 Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum **Get a Percona Expert**

Last update: 2023-10-13

Percona and/or its affiliates © 2023 — Cookie Consent

4.5 FLUSH TABLES WITH READ LOCK option

The `FLUSH TABLES WITH READ LOCK` option does the following with a global read lock:

- Closes all open tables
- Locks all tables for all databases

Release the lock with `UNLOCK TABLES`.

Note

`FLUSH TABLES WITH READ LOCK` does not prevent inserting rows into the log tables.

To ensure consistent backups, use the `FLUSH TABLES WITH READ LOCK` option before taking a non-InnoDB file backup. The option does not affect long-running queries.

Enabling `FLUSH TABLES WITH READ LOCK` when the server has long-running queries can leave the server in a read-only mode until the queries finish. If the server is in either the `Waiting for table flush` or the `Waiting for master to send event` state, stopping the `FLUSH TABLES WITH READ LOCK` operation does not help. Stop any long-running queries to return to normal operation.

To prevent the server staying in a read-only mode until the queries finish, xtrabackup does the following:

- checks if any queries run longer than specified in `--ftwrl-wait-threshold`. If xtrabackup finds such queries, xtrabackup waits for one second and checks again. If xtrabackup waits longer than specified in `--ftwrl-wait-timeout`, the backup is aborted. As soon as xtrabackup finds no queries running longer than specified in `--ftwrl-wait-threshold`, xtrabackup issues the global lock.
- kills all queries or only the SELECT queries which prevent the global lock from being acquired.

Note

All operations described in this section have no effect when Backup locks are used.

Percona XtraBackup uses Backup locks where available as a lightweight alternative to `FLUSH TABLES WITH READ LOCK`. This operation automatically copies non-InnoDB data and avoids blocking DML queries that modify InnoDB tables.

4.5.1 Wait for queries to finish

You should issue a global lock when no long queries are running. Waiting to issue the global lock for extended period of time is not a good method. The wait can extend the time needed for backup to take place. The `–ftwrl-wait-timeout` option can limit the waiting time. If it cannot issue the lock during this time, xtrabackup stops the option, exits with an error message, and backup is not be taken.

The option's default value is zero (0), which turns off the option.

Another possibility is to specify the type of query to wait on. In this case `--ftwrl-wait-query-type`.

Percona LLC and/or its affiliates © 2023 — Cookie Consent

The possible values are `all` and `update`. When `all` is used xtrabackup will wait for all long running queries (execution time longer than allowed by `--ftwrl-wait-threshold`) to finish before running the `FLUSH TABLES WITH READ LOCK`. When `update` is used xtrabackup will wait on `UPDATE/ALTER/REPLACE/INSERT` queries to finish.

The time needed for a specific query to complete is hard to predict. We assume that the long-running queries will not finish in a timely manner. Other queries which run for a short time finish quickly. xtrabackup uses the value of `–ftwrl-wait-threshold option to specify the long-running queries`
`and will block a global lock. To use this option`
`xtrabackup user should have` PROCESS `and` CONNECTION_ADMIN` privileges.

4.5.2 Kill the blocking queries

The second option is to kill all the queries which prevent from acquiring the global lock. In this case, all queries which run longer than `FLUSH TABLES WITH READ LOCK` are potential blockers. Although all queries can be killed, additional time can be specified for the short running queries to finish using the `--kill-long-queries-timeout` option. This option specifies a query time limit. After the specified time is reached, the server kills the query. The default value is zero, which turns this feature off.

The `--kill-long-query-type` option can be used to specify all or only `SELECT` queries that are preventing global lock from being acquired. To use this option xtrabackup user should have `PROCESS` and `CONNECTION_ADMIN` privileges.

4.5.3 Options summary

- `--ftwrl-wait-timeout` (seconds) - how long to wait for a good moment. Default is 0, not to wait.
- `--ftwrl-wait-query-type` - which long queries should be finished before `FLUSH TABLES WITH READ LOCK` is run. Default is all.
- `--ftwrl-wait-threshold` (seconds) - how long query should be running before we consider it long running and potential blocker of global lock.
- `--kill-long-queries-timeout` (seconds) - how many time we give for queries to complete after `FLUSH TABLES WITH READ LOCK` is issued before start to kill. Default if `0`, not to kill.
- `--kill-long-query-type` - which queries should be killed once `kill-long-queries-timeout` has expired.

Example

Running the xtrabackup with the following options will cause xtrabackup to spend no longer than 3 minutes waiting for all queries older than 40 seconds to complete.

```
$  xtrabackup --backup --ftwrl-wait-threshold=40 \
--ftwrl-wait-query-type=all --ftwrl-wait-timeout=180 \
--kill-long-queries-timeout=20 --kill-long-query-type=all \
--target-dir=/data/backups/
```

After `FLUSH TABLES WITH READ LOCK` is issued, xtrabackup will wait for 20 seconds for lock to be acquired. If lock is still not acquired after 20 seconds, it will kill all queries which are running longer that the `FLUSH TABLES WITH READ LOCK`.

Percona and/or its affiliates © 2023 — Cookie Consent

4.5.4 Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-10-30

4.6 Improved log statements

Percona XtraBackup is an open-source command-line utility. Command-line tools have limited interaction with the background operations and the logs provide the progress of an operation or more information about errors.

The error logs did not have a standard structure and the log statements varied in the following ways:

- The backup log statement header has the name of the module, `xtrabackup`, which generated the statement but no timestamp:

```
 $ xtrabackup: recognized client arguments: --parallel=4 --target-dir=/data/backups/
--backup=1
```

The output should be similar to the following:

Expected output

```
./bin/xtrabackup version 8.2.0-1 based on MySQL server 8.2 Linux (x86_64) (revision id:
b0f75188ca3)
```

- The copy-back log statement has a timestamp but no module name. The timestamp is a mix of UTC and the local timezone.

```
220322 19:05:13 [01] Copying undo_001 to /data/backups/undo_001
```

- The following prepare log statements do not have header information, which makes diagnosing an issue more difficult.

```
Completed space ID check of 1008 files.
Initializing buffer pool, total size = 128.000000M, instances = 1, chunk size
=128.000000M
Completed initialization of buffer pool
If the mysqld execution user is authorized, page cleaner thread priority can be
changed. See the man page of setpriority().
```


Percona and/or its affiliates © 2023 — Cookie Consent

4.6.1 Log statement structure

The improved log structure is displayed in the backup, prepare, move-back/copy-back error logs.

Each log statement has the following attributes:

- Timestamp - a timestamp for when the event occurred in a UTC format.
- Severity - the severity level of a statement indicates the importance of an event.
- ID - this identifier is currently not used but may be used in future versions.
- Context - the name of the module that issued the log statement, such as XtraBackup, InnoDB, or Server.
- Message - a description of the event generated by the module.

An example of a `prepare` log that is generated with the improved structure. The uniformity of the headers makes it easier to follow an operation's progress or review the log to diagnose issues.

Expected output

```
2022-03-22T19:15:36.142247+05:30 0 [Note] [MY-011825] [Xtrabackup] This target seems to be not
prepared yet.
2022-03-22T19:15:36.142792+05:30 0 [Note] [MY-013251] [InnoDB] Number of pools: 1
2022-03-22T19:15:36.149212+05:30 0 [Note] [MY-011825] [Xtrabackup] xtrabackup_logfile detected:
size=8388608, start_lsn=(33311656)
2022-03-22T19:15:36.149998+05:30 0 [Note] [MY-011825] [Xtrabackup] using the following InnoDB
configuration for recovery:
2022-03-22T19:15:36.150023+05:30 0 [Note] [MY-011825] [Xtrabackup] innodb_data_home_dir = .
2022-03-22T19:15:36.150036+05:30 0 [Note] [MY-011825] [Xtrabackup] innodb_data_file_path =
ibdata1:12M:autoextend
2022-03-22T19:15:36.150078+05:30 0 [Note] [MY-011825] [Xtrabackup] innodb_log_group_home_dir = .
2022-03-22T19:15:36.150095+05:30 0 [Note] [MY-011825] [Xtrabackup] innodb_log_files_in_group = 1
2022-03-22T19:15:36.150111+05:30 0 [Note] [MY-011825] [Xtrabackup] innodb_log_file_size = 8388608
2022-03-22T19:15:36.151667+05:30 0 [Note] [MY-011825] [Xtrabackup] inititialize_service_handles
suceeded
2022-03-22T19:15:36.151903+05:30 0 [Note] [MY-011825] [Xtrabackup] using the following InnoDB
configuration for recovery:
2022-03-22T19:15:36.151926+05:30 0 [Note] [MY-011825] [Xtrabackup] innodb_data_home_dir = .
2022-03-22T19:15:36.151954+05:30 0 [Note] [MY-011825] [Xtrabackup] innodb_data_file_path =
ibdata1:12M:autoextend
2022-03-22T19:15:36.151976+05:30 0 [Note] [MY-011825] [Xtrabackup] innodb_log_group_home_dir = .
2022-03-22T19:15:36.151991+05:30 0 [Note] [MY-011825] [Xtrabackup] innodb_log_files_in_group = 1
2022-03-22T19:15:36.152004+05:30 0 [Note] [MY-011825] [Xtrabackup] innodb_log_file_size = 8388608
2022-03-22T19:15:36.152021+05:30 0 [Note] [MY-011825] [Xtrabackup] Starting InnoDB instance for
recovery.
2022-03-22T19:15:36.152035+05:30 0 [Note] [MY-011825] [Xtrabackup] Using 104857600 bytes for
buffer pool (set by --use-memory parameter)
```

4.6.2 Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Percona and/or its affiliates © 2023 — Cookie Consent

Last update: 2023-10-23

4.7 lock-ddl-per-table option improvements

To block DDL statements on an instance, Percona Server implemented LOCK TABLES FOR BACKUP. Percona XtraBackup uses this lock for the duration of the backup. This lock does not affect DML statements.

Percona XtraBackup has also implemented `--lock-ddl-per-table`, which blocks DDL statements by using metadata locks (MDL).

The following procedures describe a simplified backup operation when using `--lock-ddl-per-table`:

1. Parse and copy all redo logs after the checkpoint mark
2. Fork a dedicated thread to continue following new redo log entries
3. List the tablespaces required to copy
4. Iterate through the list. The following steps occur with each listed tablespace:
5. Query INFORMATION_SCHEMA.INNODB_TABLES to find which tables belong to the tablespace ID and take an MDL on the underlying table or tables in case there is a shared tablespace.
6. Copy the tablespace `.ibd` files.

The backup process may encounter a redo log event, generated by the bulk load operations, which notifies backup tools that data file changes have been omitted from the redo log. This event is an `MLOG_INDEX_LOAD`. If this event is found by the redo follow thread, the backup continues and assumes the backup is safe because the MDL protects tablespaces already copied and the MLOG_INDEX_LOAD event is for a tablespace that is not copied.

These assumptions may not be correct and may lead to inconsistent backups.

4.7.1 `--lock-ddl-per-table` redesign

The `--lock-ddl-per-table` option has been redesigned to minimize inconsistent backups.

The following procedure reorders the steps:

- The MDL lock acquired at the beginning of the backup
- Scan the redo logs. An `MLOG_INDEX_LOAD` event may be recorded if a `CREATE INDEX` statement has occurred before the backup starts. At this time, the backup process is safe and can parse and accept the event.
- After the first scan has completed, the follow redo log thread is initiated. This thread stops the backup process if an `MLOG_INDEX_LOAD` event is found.
- Gather the tablespace list to copy
- Copy the `.ibd` files.

Percona LLC and/or its affiliates © 2023 — Cookie Consent

4.7.2 Other improvements

The following improvements have been added:

- If the `.ibd` file belongs to a temporary table, the `SELECT` query is skipped.
- For a FullText Index, an MDL is acquired on the base table.
- A `SELECT` query that acquires an MDL does not retrieve any data.

4.7.3 Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum **Get a Percona Expert**

Last update: 2023-10-10

4.8 Smart memory estimation

The Smart memory estimation is tech preview feature. Before using Smart memory estimation in production, we recommend that you test restoring production from physical backups in your environment and also use the alternative backup method for redundancy.

Percona XtraBackup supports the Smart memory estimation feature. With this feature, Percona XtraBackup computes the memory required for `prepare` phase, while copying redo log entries during the `backup` phase. Percona XtraBackup also considers the number of InnoDB pages to be fetched from the disk.

Percona and/or its affiliates © 2023 — Cookie Consent

Percona XtraBackup performs the backup procedure in two steps:

- Creates a backup

  To create a backup, Percona XtraBackup copies your InnoDB data files. While copying the files, Percona XtraBackup runs a background process that watches the InnoDB redo log, also called the transaction log, and copies changes from it.

- Prepares a backup

  During the `prepare` phase, Percona XtraBackup performs crash recovery against the copied data files using the copied transaction log file. Percona XtraBackup reads all the redo log entries into memory, categorizes them by space id and page id, reads the relevant pages into memory, and checks the log sequence number (LSN) on the page and on the redo log record. If the redo log LSN is more recent than the page LSN, Percona XtraBackup applies the redo log changes to the page.

  To `prepare` a backup, Percona Xtrabackup uses InnoDB Buffer Pool memory. Percona Xtrabackup reserves memory to load 256 pages into the buffer pool. The remaining memory is used for hashing/categorizing the redo log entries.

  The available memory is controlled by the `--use-memory` option. If the available memory on the buffer pool is insufficient, the work is performed in multiple batches. After the batch is processed, the memory is freed to release space for the next batch. This process greatly impacts performance as an InnoDB page holds data from multiple rows. If a change on a page happens in different batches, that page is fetched and evicted numerous times.

4.8.1 How does Smart memory estimation work

To run `prepare`, Percona XtraBackup checks the server's available free memory and uses that memory up to the limit specified in the `--use-free-memory-pct` option. Due to backward compatibility, the default value for the `--use-free-memory-pct` option is 0 (zero), which defines the option as disabled. For example, if you set `--use-free-memory-pct=50`, then 50% of the free memory is used to `prepare` a backup.

You can enable or disable the memory estimation during the `backup` phase with the `--estimate-memory` option. The default value is `OFF`. Enable the memory estimation with `--estimate-memory=ON`:

```
$ xtrabackup --backup --estimate-memory=ON --target-dir=/data/backups/
```

In the `prepare` phase, enable the `--use-free-memory-pct` option by specifying the percentage of free memory to be used to `prepare` a backup. The `--use-free-memory-pct` value must be larger than 0.

For example:

```
$ xtrabackup --prepare --use-free-memory-pct=50 --target-dir=/data/backups/
```

4.8.2 Example of Smart memory estimation usage

The examples of how Smart memory estimation can improve the time spent on `prepare` in Percona XtraBackup:

We back up 16, 32, and 64 tables using sysbench. Each set contains 1M rows. In the `backup` phase, we enable Smart memory estimation with `--estimate-memory=ON`. In the `prepare` phase, we set `--use-free-memory-pct=50`, and Percona XtraBackup uses 50% of the free memory to prepare a backup. The

Percona and/or its affiliates © 2023 — Cookie Consent

backup is run on an ec2 c4.8xlarge instance (36 vCPUs / 60GB memory / General Purpose SSD (gp2)).

During each `--backup`, the following sysbench is run:

```
sysbench --db-driver=mysql --db-ps-mode=disable --mysql-user=sysbench --mysql-
password=sysbench --table_size=1000000 --tables=${NUM_OF_TABLES} --threads=24 --
time=0 --report-interval=1 /usr/share/sysbench/oltp_write_only.lua run
```

The following table shows the backup details (all measurements are in Gigabytes):

	Used memory	Size of XtraBackup log	Size of backup
16 tables	3.375	0.7	4.7
32 tables	8.625	2.6	11
64 tables	18.5	5.6	22

- Used memory - the amount of memory required by Percona XtraBackup with `--use-free-memory-pct=50`
- Size of XtraBackup log - the size of Percona XtraBackup log file (redo log entries copied during the backup)
- Size of backup - the size of the resulting backup folder

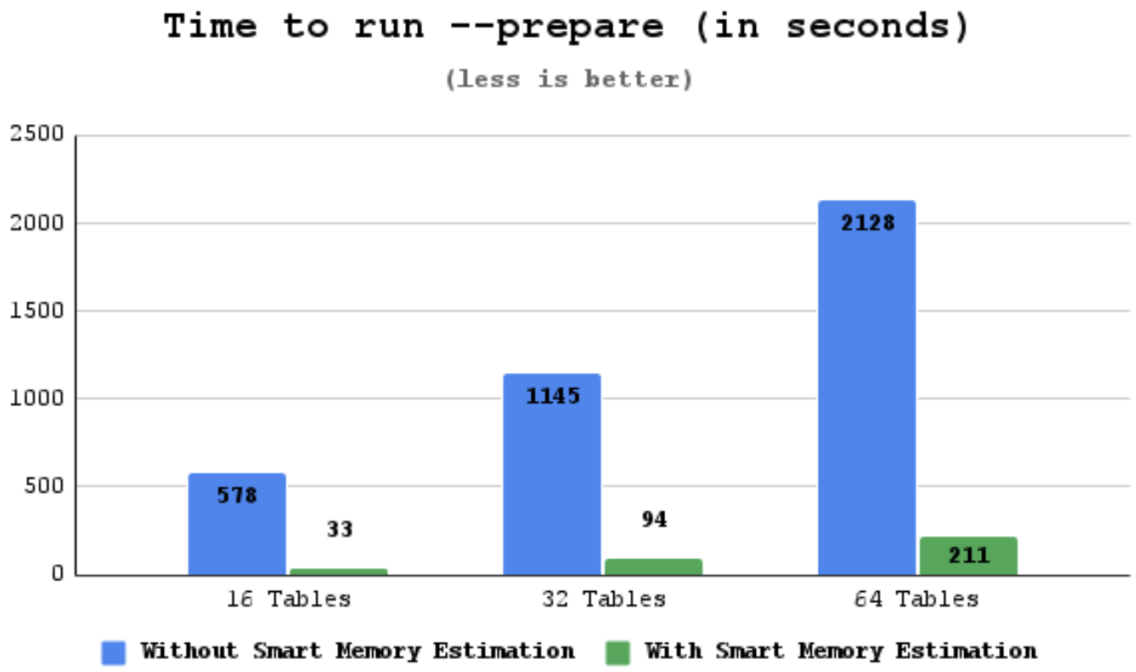
`Prepare` executed without Smart memory estimation uses the default of 128MB for the buffer pool.

The results are the following:

Note

The following results are based on tests in a specific environment. Your results may vary.

Percona and/or its affiliates © 2023 — Cookie Consent



- 16 tables result - prepare time dropped to ~5.7% of the original time. An improvement in recovery time of about 17x.
- 32 tables result - prepare time dropped to ~8,2% of the original time. An improvement in recovery time of about 12x.
- 64 tables result - prepare time dropped to ~9.9% of the original time. An improvement in recovery time of about 10x.

4.8.3 Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum **Get a Percona Expert**

Last update: 2023-10-10

4.9 Work with binary logs

The `xtrabackup` binary integrates with the `log_status table`. This integration enables `xtrabackup` to print out the backup's corresponding binary log position, so that you can use this binary log position to provision a new replica or perform point-in-time recovery.

Percona LLC and/or its affiliates © 2023 — Cookie Consent

4.9.1 Find the binary log position

You can find the binary log position corresponding to a backup after the backup has been taken. If your backup is from a server with binary logging enabled, `xtrabackup` creates a file named `xtrabackup_binlog_info` in the target directory. This file contains the binary log file name and position of the exact point when the backup was taken.

Expected output during the backup stage

```
210715 14:14:59 Backup created in directory '/backup/'
MySQL binlog position: filename 'binlog.000002', position '156'
. . .
210715 14:15:00 completed OK!
```

4.9.2 Point-in-time recovery

To perform a point-in-time recovery from an `xtrabackup` backup, you should prepare and restore the backup, and then replay binary logs from the point shown in the `xtrabackup_binlog_info` file.

Find a more detailed procedure in the Point-in-time recovery document.

4.9.3 Set up a new replication replica

To set up a new replica, you should prepare the backup, and restore it to the data directory of your new replication replica. In the CHANGE_REPLICATION_SOURCE_TO with the appropriate options command, use the binary log filename and position shown in the `xtrabackup_binlog_info` file to start replication.

A more detailed procedure is found in How to setup a replica for replication in 6 simple steps with Percona XtraBackup.

4.9.4 Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum **Get a Percona Expert**

Last update: 2023-10-12

Percona and/or its affiliates © 2023 — Cookie Consent

5. Binaries

5.1 Percona XtraBackup binaries overview

Percona XtraBackup contains a set of the following binaries:

- xtrabackup - a compiled C binary that provides functionality to backup a whole MySQL database instance with MyISAM, InnoDB, and XtraDB tables.
- xbcrypt - a utility used for encrypting and decrypting backup files.
- xbstream - a utility that allows streaming and extracting files to or from the xbstream format.
- xbcloud - a utility used for downloading and uploading full or part of xbstream archive from or to cloud.

The recommended way to take the backup is by using the xtrabackup script. For more information on script options, see xtrabackup.

5.1.1 Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum **Get a Percona Expert**

Last update: 2023-10-10

5.2 The xtrabackup binary

5.2.1 The xtrabackup binary overview

The *xtrabackup* binary is a compiled C program that is linked with the *InnoDB* libraries and the standard *MySQL* client libraries.

xtrabackup enables point-in-time backups of *InnoDB* / *XtraDB* tables together with the schema definitions, *MyISAM* tables, and other portions of the server.

The *InnoDB* libraries provide the functionality to apply a log to data files. The *MySQL* client libraries are used to parse command-line options and configuration file.

The tool runs in either `--backup` or `--prepare` mode, corresponding to the two main functions it performs. There are several variations on these functions to accomplish different tasks, and there is one less commonly used mode, `--print-param`.

Percona LLC and/or its affiliates © 2023 — Cookie Consent

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-09-07

5.2.2 The xtrabackup option reference

Here you can find all of the command-line options for the xtrabackup binary.

Modes of operation

You invoke xtrabackup in one of the following modes:

- `--backup` mode to make a backup in a target directory
- `--prepare` mode to restore data from a backup (created in `--backup` mode)
- `--copy-back` to copy data from a backup to the location that contained the original data; to move data instead of copying use the alternate `--move-back` mode.

When you intend to run xtrabackup in any of these modes, use the following syntax:

```
$ xtrabackup [--defaults-file=#] --backup|--prepare|--copy-back| [OPTIONS]
```

For example, the `--prepare` mode is applied as follows:

```
$ xtrabackup --prepare --target-dir=/data/backup/mysql/
```

For all modes, the default options are read from the xtrabackup and mysqld configuration groups from the following files in the given order:

1. `/etc/my.cnf`
2. `/etc/mysql/my.cnf`
3. `/usr/etc/my.cnf`
4. `~/.my.cnf`.

Percona and/or its affiliates © 2023 — Cookie Consent

As the first parameter to xtrabackup (in place of the `--defaults-file`, you may supply one of the following:

- `--print-defaults` to have xtrabackup print the argument list and exit.
- `--no-defaults` to forbid reading options from any file but the login file.
- `--defaults-file` to read the default options from the given file.
- `--defaults-extra-file` to read the specified additional file after the global files have been read.
- `--defaults-group-suffix` to read the configuration groups with the given suffix. The effective group name is constructed by concatenating the default configuration groups (xtrabackup and mysqld) with the given suffix.
- `--login-path` to read the given path from the login file.

INNODB OPTIONS

There is a large group of InnoDB options that are normally read from the `my.cnf` configuration file, so that xtrabackup boots up its embedded InnoDB in the same configuration as your current server. You normally do not need to specify them explicitly. These options have the same behavior in InnoDB and XtraDB. See `--innodb-miscellaneous` for more information.

Options

–APPLY-LOG-ONLY()

This option causes only the redo stage to be performed when preparing a backup. It is very important for incremental backups.

–BACKUP()

Make a backup and place it in `--target-dir`. See Creating a backup.

–BACKUP-LOCK-TIMEOUT()

The timeout in seconds for attempts to acquire metadata locks.

–BACKUP-LOCK-RETRY-COUNT()

The number of attempts to acquire metadata locks.

–BACKUP-LOCKS()

This option controls if backup locks should be used instead of `FLUSH TABLES WITH READ LOCK` on the backup stage. The option has no effect when backup locks are not supported by the server. This option is enabled by default, disable with `--no-backup-locks`.

–CHECK-PRIVILEGES()

This option checks if Percona XtraBackup has all required privileges. If a missing privilege is required for the current operation, it will terminate and print out an error message. If a missing privilege is not required for the current operation, but may be necessary for some other XtraBackup operation, the process is not aborted and a warning is printed.

```
xtrabackup: Error: missing required privilege LOCK TABLES on *.*
xtrabackup: Warning: missing required privilege REPLICATION CLIENT on *.*
```


Percona LLC and/or its affiliates © 2023 — Cookie Consent

–CLOSE-FILES()

Do not keep files opened. When xtrabackup opens tablespace it normally doesn't close its file handle in order to handle the DDL operations correctly. However, if the number of tablespaces is really huge and can not fit into any limit, there is an option to close file handles once they are no longer accessed. Percona XtraBackup can produce inconsistent backups with this option enabled. Use at your own risk.

–COMPRESS()

This option tells xtrabackup to compress all output data, including the transaction log file and meta data files, using either the `ZSTD` or `lz4` compression algorithm. `ZSTD` is the default compression algorithm.

`--compress` produces `\*.zst` files. You can extract the contents of these files by using the `--decompress` option. You can specify `ZSTD` compression level with the `--compress-zstd-level(=#)` option.

`--compress=lz4` produces `\*.lz4` files. You can extract the contents of these files by using `lz4` program.

–COMPRESS-CHUNK-SIZE(=#)

Size of working buffer(s) for compression threads in bytes. The default value is 64K.

–COMPRESS-THREADS(=#)

This option specifies the number of worker threads used by xtrabackup for parallel data compression. This option defaults to `1`. Parallel compression (`--compress-threads`) can be used together with parallel file copying (`--parallel`). For example, `--parallel=4 --compress --compress-threads=2` will create 4 I/O threads that will read the data and pipe it to 2 compression threads.

–COMPRESS-ZSTD-LEVEL(=#)

This option specifies `ZSTD` compression level. Compression levels provide a trade-off between the speed of compression and the size of the compressed files. A lower compression level provides faster compression speed but larger file sizes. A higher compression level provides lower compression speed but smaller file sizes. For example, set level 1 if the compression speed is the most important for you. Set level 19 if the size of the compressed files is the most important.

The default value is 1. Allowed range of values is from 1 to 19.

–COPY-BACK()

Copy all the files in a previously made backup from the backup directory to their original locations. This option will not copy over existing files unless `--force-non-empty-directories` option is specified.

–CORE-FILE()

Write core on fatal signals.

–DATABASES(=#)

This option specifies a list of databases and tables that should be backed up. The option accepts the list of the form `"databasename1[.table_name1] databasename2[.table_name2] . . ."`.

Percona LLC and/or its affiliates © 2023 — Cookie Consent

–DATABASES-EXCLUDE(=NAME)

Excluding databases based on name, Operates the same way as `--databases`, but matched names are excluded from backup. Note that this option has a higher priority than `--databases`.

–DATABASES-FILE(=#)

This option specifies the path to the file containing the list of databases and tables that should be backed up. The file can contain the list elements of the form `databasename1[.table_name1]`, one element per line.

–DATADIR(=DIRECTORY)

The source directory for the backup. This should be the same as the datadir for your MySQL server, so it should be read from `my.cnf` if that exists; otherwise you must specify it on the command line.

When combined with the `--copy-back` or `--move-back` option, `--datadir` refers to the destination directory.

Once connected to the server, in order to perform a backup you will need `READ` and `EXECUTE` permissions at a filesystem level in the server's datadir.

–DEBUG-SLEEP-BEFORE-UNLOCK(=#)

This is a debug-only option used by the xtrabackup test suite.

–DEBUG-SYNC(=NAME)

The debug sync point. This option is only used by the xtrabackup test suite.

–DECOMPRESS()

Decompresses all files in a backup previously made with the `--compress` option. The `--parallel` option will allow multiple files to be decrypted simultaneously. Percona XtraBackup does not automatically remove the compressed files. In order to clean up the backup directory users should use `--remove-original` option.

–DECOMPRESS-THREADS(=#)

Force xbstream to use the specified number of threads for decompressing.

–DECRYPT(=ENCRYPTION-ALGORITHM)

Decrypts all files with the `.xbcrypt` extension in a backup previously made with `--encrypt` option. The `--parallel` option will allow multiple files to be decrypted simultaneously. Percona XtraBackup doesn't automatically remove the encrypted files. In order to clean up the backup directory users should use `--remove-original` option.

–DEFAULTS-EXTRA-FILE(=[MY.CNF])

Read this file after the global files are read. Must be given as the first option on the command-line.

–DEFAULTS-FILE(=[MY.CNF])

Only read default options from the given file. Must be given as the first option on the command-line. Must be a real file; it cannot be a symbolic link.

–DEFAULTS-GROUP(=GROUP-NAME)

This option is to set the group which should be read from the configuration file. This is used by xtrabackup if you use the `--defaults-group` option. It is needed for `mysqld_multi` deployments.

Percona and/or its affiliates © 2023 — Cookie Consent

–DEFAULTS-GROUP-SUFFIX(=#)

Also reads groups with concat(group, suffix).

–DUMP-INNODB-BUFFER-POOL()

This option controls whether or not a new dump of buffer pool content should be done.

With `--dump-innodb-buffer-pool`, xtrabackup makes a request to the server to start the buffer pool dump (it takes some time to complete and is done in background) at the beginning of a backup provided the status variable `innodb_buffer_pool_dump_status` reports that the dump has been completed.

```
$ xtrabackup --backup --dump-innodb-buffer-pool --target-dir=/home/user/backup
```

By default, this option is set to OFF.

If `innodb_buffer_pool_dump_status` reports that there is running dump of buffer pool, xtrabackup waits for the dump to complete using the value of `--dump-innodb-buffer-pool-timeout`

The file `ib_buffer_pool` stores tablespace ID and page ID data used to warm up the buffer pool sooner.

–DUMP-INNODB-BUFFER-POOL-TIMEOUT()

This option contains the number of seconds that xtrabackup should monitor the value of `innodb_buffer_pool_dump_status` to determine if buffer pool dump has completed.

This option is used in combination with `--dump-innodb-buffer-pool`. By default, it is set to 10 seconds.

–DUMP-INNODB-BUFFER-POOL-PCT()

This option contains the percentage of the most recently used buffer pool pages to dump.

This option is effective if `--dump-innodb-buffer-pool` option is set to ON. If this option contains a value, xtrabackup sets the MySQL system variable `innodb_buffer_pool_dump_pct`. As soon as the buffer pool dump completes or it is stopped (see `--dump-innodb-buffer-pool-timeout`), the value of the MySQL system variable is restored.

–ENCRYPT(=ENCRYPTION_ALGORITHM)

This option instructs xtrabackup to encrypt backup copies of InnoDB data files using the algorithm specified in the ENCRYPTION_ALGORITHM. Currently supported algorithms are: `AES128`, `AES192` and `AES256`

–ENCRYPT-KEY(=ENCRYPTION_KEY)

A proper length encryption key to use. It is not recommended to use this option where there is uncontrolled access to the machine as the command line and thus the key can be viewed as part of the process info.

–ENCRYPT-KEY-FILE(=ENCRYPTION_KEY_FILE)

The name of a file where the raw key of the appropriate length can be read from. The file must be a simple binary (or text) file that contains exactly the key to be used.

It is passed directly to the xtrabackup child process. See the xtrabackup documentation for more details.

Percona and/or its affiliates © 2023 — Cookie Consent

–ENCRYPT-THREADS(=#)

This option specifies the number of worker threads that will be used for parallel encryption/decryption. See the xtrabackup documentation for more details.

–ENCRYPT-CHUNK-SIZE(=#)

This option specifies the size of the internal working buffer for each encryption thread, measured in bytes. It is passed directly to the xtrabackup child process.

To adjust the chunk size for encrypted files, use `--read-buffer-size` and `--encrypt-chunk-size`.

–ESTIMATE-MEMORY(=#)

This option is in tech preview. Before using this option in production, we recommend that you test restoring production from physical backups in your environment, and also use the alternative backup method for redundancy.

Implemented in Percona XtraBackup 8.0.32-26, the option lets you enable or disable the Smart memory estimation feature. The default value is OFF. Enable the feature by setting `--estimate-memory=ON` in the backup phase and setting the `--use-free-memory-pct` option in the `--prepare` phase. If the `--estimate-memory` setting is disabled, the `--use-free-memory-pct` setting is ignored.

An example of how to enable the Smart memory estimation feature:

```
$ xtrabackup --backup --estimate-memory=ON --target-dir=/data/backups/
```

```
$ xtrabackup --prepare --use-free-memory-pct=50 --target-dir=/data/backups/
```

–EXPORT()

Create files necessary for exporting tables. See Restoring Individual Tables.

–EXTRA-LSNDIR(=DIRECTORY)

(for –backup): save an extra copy of the `xtrabackup_checkpoints` and `xtrabackup_info` files in this directory.

–FORCE-NON-EMPTY-DIRECTORIES()

When specified, it makes `--copy-back` and `--move-back` option transfer files to non-empty directories. No existing files will be overwritten. If files that need to be copied/moved from the backup directory already exist in the destination directory, it will still fail with an error.

–FTWRL-WAIT-TIMEOUT(=SECONDS)

This option specifies time in seconds that xtrabackup should wait for queries that would block `FLUSH TABLES WITH READ LOCK` before running it. If there are still such queries when the timeout expires, xtrabackup terminates with an error. Default is `0`, in which case it does not wait for queries to complete and starts `FLUSH TABLES WITH READ LOCK` immediately. Where supported xtrabackup will automatically use Backup Locks as a lightweight alternative to `FLUSH TABLES WITH READ LOCK` to copy non-InnoDB data to avoid blocking DML queries that modify InnoDB tables.

–FTWRL-WAIT-THRESHOLD(=SECONDS)

This option specifies the query run time threshold which is used by xtrabackup to detect long-running queries with a non-zero value of `--ftwrl-wait-timeout`. `FLUSH TABLES WITH READ LOCK` is not started until such long-running queries exist. This option has no effect if `--ftwrl-wait-timeout` is `0`.

Percona and/or its affiliates © 2023 — Cookie Consent

Default value is `60` seconds. Where supported xtrabackup will automatically use Backup Locks as a lightweight alternative to `FLUSH TABLES WITH READ LOCK` to copy non-InnoDB data to avoid blocking DML queries that modify InnoDB tables.

–FTWRL-WAIT-QUERY-TYPE(=ALL|UPDATE)

This option specifies which types of queries are allowed to complete before xtrabackup will issue the global lock. Default is `all`.

–GALERA-INFO()

This option creates the `xtrabackup_galera_info` file which contains the local node state at the time of the backup. Option should be used when performing the backup of Percona XtraDB Cluster. It has no effect when backup locks are used to create the backup.

–GENERATE-NEW-MASTER-KEY()

Generate a new master key when doing a copy-back.

–GENERATE-TRANSITION-KEY()

xtrabackup needs to access the same keyring file or vault server during prepare and copy-back but it should not depend on whether the server keys have been purged.

`--generate-transition-key` creates and adds to the keyring a transition key for xtrabackup to use if the master key used for encryption is not found because it has been rotated and purged.

–GET-SERVER-PUBLIC-KEY()

Get the server public key

–HELP()

When run with this option or without any options xtrabackup displays information about how to run the program on the command line along with all supported options and variables with default values where appropriate.

–HISTORY(=NAME)

This option enables the tracking of backup history in the `PERCONA_SCHEMA.xtrabackup_history` table. An optional history series name may be specified that will be placed with the history record for the current backup being taken.

–HOST(=HOST)

This option accepts a string argument that specifies the host to use when connecting to the database server with TCP/IP. It is passed to the mysql child process without alteration. See mysql –help for details.

–INCREMENTAL-BASEDIR(=DIRECTORY)

When creating an incremental backup, this is the directory containing the full backup that is the base dataset for the incremental backups.

–INCREMENTAL-DIR(=DIRECTORY)

When preparing an incremental backup, this is the directory where the incremental backup is combined with the full backup to make a new full backup.

Percona and/or its affiliates © 2023 — Cookie Consent

–INCREMENTAL-FORCE-SCAN()

When creating an incremental backup, force a full scan of the data pages in that instance.

–INCREMENTAL-HISTORY-NAME(=NAME)

This option specifies the name of the backup series stored in the `PERCONA_SCHEMA.xtrabackup_history` history record to base an incremental backup on. xtrabackup will search the history table looking for the most recent (highest `innodb_to_lsn`), successful backup in the series and take the to_lsn value to use as the starting `lsn` for the incremental backup. This will be mutually exclusive with `--incremental-history-uuid`, `--incremental-basedir` and `--incremental-lsn`. If no valid lsn can be found (no series by that name, no successful backups by that name) xtrabackup will return with an error. It is used with the `--incremental` option.

–INCREMENTAL-HISTORY-UUID(=NAME)

This option specifies the UUID of the specific history record stored in the `PERCONA_SCHEMA.xtrabackup_history` to base an incremental backup on. `--incremental-history-name`, `--incremental-basedir` and `--incremental-lsn`. If no valid lsn can be found (no success record with that UUID) xtrabackup will return with an error. It is used with the –incremental option.

–INCREMENTAL-LSN(=LSN)

When creating an incremental backup, you can specify the log sequence number (LSN) instead of specifying `--incremental-basedir`. For databases created in 5.1 and later, specify the LSN as a single 64-bit integer.

> **Important**
>
> If a wrong LSN value is specified (a user error which Percona XtraBackup does not detect), the backup is unusable. Be careful!

–INNODB([=NAME])

This option is ignored for MySQL option compatibility.

–INNODB-MISCELLANEOUS()

There is a large group of InnoDB options that are normally read from the `my.cnf` configuration file, so that xtrabackup boots up its embedded InnoDB in the same configuration as your current server. You normally do not need to specify these explicitly. These options have the same behavior in InnoDB and XtraDB:

–KEYRING-FILE-DATA(=FILENAME)

The path to the keyring file. Combine this option with `--xtrabackup-plugin-dir`.

–KILL-LONG-QUERIES-TIMEOUT(=SECONDS)

This option specifies the number of seconds xtrabackup waits between starting `FLUSH TABLES WITH READ LOCK` and killing those queries that block it. Default is 0 seconds, which means xtrabackup will not attempt to kill any queries. In order to use this option xtrabackup user should have the `PROCESS` and `SUPER` privileges. Where supported, xtrabackup automatically uses Backup locks as a lightweight alternative to `FLUSH TABLES WITH READ LOCK` to copy non-InnoDB data to avoid blocking DML queries that modify InnoDB tables.

Percona and/or its affiliates © 2023 — Cookie Consent

–KILL-LONG-QUERY-TYPE(=ALL|SELECT)

This option specifies which types of queries should be killed to unblock the global lock. Default is “select”.

–LOCK-DDL()

Issue `LOCK TABLES FOR BACKUP` if it is supported by server (otherwise use `LOCK INSTANCE FOR BACKUP`) at the beginning of the backup to block all DDL operations.

Note

Prior to Percona XtraBackup 8.0.22-15.0, using a safe-slave-backup stops the SQL replica thread after the InnoDB tables and before the non-InnoDB tables are backed up.

As of Percona XtraBackup 8.0.22-15.0, using a safe-slave-backup option stops the SQL replica thread before copying the InnoDB files.

–LOCK-DDL-PER-TABLE()

Lock DDL for each table before xtrabackup starts to copy it and until the backup is completed.

Note

As of Percona XtraBackup 8.0.15, the `–lock-ddl-per-table` option is deprecated. Use the `–lock-ddl` option instead.

–LOCK-DDL-TIMEOUT()

If `LOCK TABLES FOR BACKUP` or `LOCK INSTANCE FOR BACKUP` does not return within given timeout, abort the backup.

–LOG()

This option is ignored for MySQL

–LOG-BIN()

The base name for the log sequence.

–LOG-BIN-INDEX(=NAME)

File that holds the names for binary log files.

–LOG-COPY-INTERVAL(=#)

This option specifies the time interval between checks done by the log copying thread in milliseconds (default is 1 second).

–LOGIN-PATH()

Read the given path from the login file.

–MOVE-BACK()

Move all the files in a previously made backup from the backup directory to their original locations. As this option removes backup files, it must be used with caution.

–NO-BACKUP-LOCKS()

Explicity disables the `--backup-locks` option which is enabled by default.

Percona and/or its affiliates © 2023 — Cookie Consent

–NO-DEFAULTS()

The default options are only read from the login file.

–NO-LOCK()

Disables table lock with `FLUSH TABLES WITH READ LOCK`. Use it only if all your tables are InnoDB and you do not care about the binary log position of the backup. This option shouldn't be used if there are any `DDL` statements being executed or if any updates are happening on non-InnoDB tables (this includes the system MyISAM tables in the mysql database), otherwise it could lead to an inconsistent backup. Where supported xtrabackup will automatically use Backup locks as a lightweight alternative to `FLUSH TABLES WITH READ LOCK` to copy non-InnoDB data to avoid blocking DML queries that modify InnoDB tables. If you are considering to use this because your backups are failing to acquire the lock, this could be because of incoming replication events are preventing the lock from succeeding. Please try using `--safe-slave-backup` to momentarily stop the replication replica thread, this may help the backup to succeed and you do not need to use this option.

–NO-SERVER-VERSION-CHECK()

Implemented in Percona XtraBackup 8.0.21.

The `--no-server-version-check` option disables the server version check.

The default behavior runs a check that compares the source system version to the Percona XtraBackup version. If the source system version is higher than the XtraBackup version, the backup is aborted with a message.

Adding the option overrides this check, and the backup proceeds, but there may be issues with the backup.

See Server Version and Backup Version Comparison for more information.

–NO-VERSION-CHECK()

This option disables the version check. If you do not pass this option, the automatic version check is enabled implicitly when xtrabackup runs in the `--backup` mode. To disable the version check, you should pass explicitly the `--no-version-check` option when invoking xtrabackup.

When the automatic version check is enabled, xtrabackup performs a version check against the server on the backup stage after creating a server connection. xtrabackup sends the following information to the server:

- MySQL flavour and version
- Operating system name
- Percona Toolkit version
- Perl version

Each piece of information has a unique identifier. This is a MD5 hash value that Percona Toolkit uses to obtain statistics about how it is used. This is a random UUID; no client information is either collected or stored.

–OPEN-FILES-LIMIT(=#)

The maximum number of file descriptors to reserve with setrlimit().

Percona LLC and/or its affiliates © 2023 — Cookie Consent

–PARALLEL(=#)

This option specifies the number of threads to use to copy multiple data files concurrently when creating a backup. The default value is 1 (i.e., no concurrent transfer). In Percona XtraBackup 2.3.10 and newer, this option can be used with the `--copy-back` option to copy the user data files in parallel (redo logs and system tablespaces are copied in the main thread).

–PASSWORD(=PASSWORD)

This option specifies the password to use when connecting to the database. It accepts a string argument. See mysql –help for details.

–PLUGIN-LOAD()

List of plugins to load.

–PORT(=PORT)

This option accepts a string argument that specifies the port to use when connecting to the database server with TCP/IP. It is passed to the mysql child process without alteration. See mysql –help for details.

–PREPARE()

Makes xtrabackup perform a recovery on a backup created with `--backup`, so that it is ready to use. See preparing a backup.

–PRINT-DEFAULTS()

Print the program argument list and exit. Must be given as the first option on the command-line.

–PRINT-PARAM()

Makes xtrabackup print out parameters that can be used for copying the data files back to their original locations to restore them.

–READ-BUFFER-SIZE()

Set the read buffer size. The given value is scaled up to page size. The default size is 10MB. Use this option to increase the xbcloud/xbstream chunk size from the default size. To adjust the chunk size for encrypted files, use `--read-buffer-size` and `--encrypt-chunk-size`.

–REBUILD-INDEXES()

Rebuilds indexes in a compact backup. This option only has effect when the `--prepare` and `--rebuild-threads` options are provided.

–REBUILD-THREADS(=#)

Uses the given number of threads to rebuild indexes in a compact backup. This option only has effect with the `--prepare` and `--rebuild-indexes` options.

–REGISTER-REDO-LOG-CONSUMER()

The `--register-redo-log-consumer` parameter is disabled by default. When enabled, this parameter lets Percona XtraBackup register as a redo log consumer at the start of the backup. The server does not remove a redo log that Percona XtraBackup (the consumer) has not yet copied. The consumer reads the redo log and manually advances the log sequence number (LSN). The server blocks the writes during the process. Based on the redo log consumption, the server determines when it can purge the log.

Percona LLC and/or its affiliates © 2023 — Cookie Consent

–REMOVE-ORIGINAL()

This option when specified will remove `.qp`, `.xbcrypt` and `.qp.xbcrypt` files after decryption and decompression.

–ROCKSDB-DATADIR()

RocksDB data directory

–ROCKSDB-WAL-DIR()

RocksDB WAL directory.

–ROCKSDB-CHECKPOINT-MAX-AGE()

The checkpoint cannot be older than this number of seconds when the backup completes.

–ROCKSDB-CHECKPOINT-MAX-COUNT()

Complete the backup even if the checkpoint age requirement has not been met after this number of checkpoints.

–ROLLBACK-PREPARED-TRX()

Force rollback prepared InnoDB transactions.

–RSYNC()

Uses the rsync utility to optimize local file transfers. When this option is specified, xtrabackup uses rsync to copy all non-InnoDB files instead of spawning a separate cp for each file, which can be much faster for servers with a large number of databases or tables. This option cannot be used together with `--stream`.

–SAFE-SLAVE-BACKUP()

When specified, xtrabackup will stop the replica SQL thread just before running `FLUSH TABLES WITH READ LOCK` and wait to start backup until `Slave_open_temp_tables` in `SHOW STATUS` is zero. If there are no open temporary tables, the backup will take place, otherwise the SQL thread will be started and stopped until there are no open temporary tables. The backup will fail if `Slave_open_temp_tables` does not become zero after `--safe-slave-backup-timeout` seconds. The replication SQL thread will be restarted when the backup finishes. This option is implemented in order to deal with replicating temporary tables and isn't necessary with Row-Based-Replication.

Note

Using a safe-slave-backup option stops the SQL replica thread before copying the InnoDB files.

–SAFE-SLAVE-BACKUP-TIMEOUT(=SECONDS)

How many seconds `--safe-slave-backup` should wait for `Slave_open_temp_tables` to become zero. Defaults to 300 seconds.

–SECURE-AUTH()

Refuse client connecting to server if it uses old (pre-4.1.1) protocol. (Enabled by default; use –skip-secure-auth to disable.)

–SERVER-ID(=#)

The server instance being backed up.

Percona LLC and/or its affiliates © 2023 — Cookie Consent

–SERVER-PUBLIC-KEY-PATH()

The file path to the server public RSA key in the PEM format.

–SKIP-TABLES-COMPATIBILITY-CHECK()

See `--tables-compatibility-check`.

–SLAVE-INFO()

This option is useful when backing up a replication replica server. It prints the binary log position of the source server. It also writes the binary log coordinates to the `xtrabackup_slave_info` file as a `CHANGE MASTER` command. A new replica for this source can be set up by starting a replica server on this backup and issuing a `CHANGE MASTER` command with the binary log position saved in the `xtrabackup_slave_info` file.

–SOCKET()

This option accepts a string argument that specifies the socket to use when connecting to the local database server with a UNIX domain socket. It is passed to the mysql child process without alteration. See mysql –help for details.

–SSL()

Enable secure connection.

–SSL-CA()

Path of the file which contains list of trusted SSL CAs.

–SSL-CAPATH()

Directory path that contains trusted SSL CA certificates in PEM format.

–SSL-CERT()

Path of the file which contains X509 certificate in PEM format.

–SSL-CIPHER()

List of permitted ciphers to use for connection encryption.

–SSL-CRL()

Path of the file that contains certificate revocation lists. MySQL server documentation.

–SSL-CRLPATH()

Path of directory that contains certificate revocation list files.

–SSL-FIPS-MODE()

SSL FIPS mode (applies only for OpenSSL); permitted values are: OFF, ON, STRICT.

–SSL-KEY()

Path of file that contains X509 key in PEM format.

–SSL-MODE()

Security state of connection to server.

Percona and/or its affiliates © 2023 — Cookie Consent

–SSL-VERIFY-SERVER-CERT()

Verify server certificate Common Name value against host name used when connecting to server.

–STREAM(=FORMAT)

Stream all backup files to the standard output in the specified format. Currently, this option only supports the xbstream format.

–STRICT()

If this option is specified, xtrabackup fails with an error when invalid parameters are passed.

–TABLES(=NAME)

A regular expression against which the full tablename, in `databasename.tablename` format, is matched. If the name matches, the table is backed up. See partial backups.

–TABLES-COMPATIBILITY-CHECK()

Enables the engine compatibility warning. The default value is ON. To disable the engine compatibility warning use `--skip-tables-compatibility-check`.

–TABLES-EXCLUDE(=NAME)

Filtering by regexp for table names. Operates the same way as `--tables`, but matched names are excluded from backup. Note that this option has a higher priority than `--tables`.

–TABLES-FILE(=NAME)

A file containing one table name per line, in databasename.tablename format. The backup will be limited to the specified tables.

–TARGET-DIR(=DIRECTORY)

This option specifies the destination directory for the backup. If the directory does not exist, xtrabackup creates it. If the directory does exist and is empty, xtrabackup will succeed. xtrabackup will not overwrite existing files, however; it will fail with operating system error 17, `file exists`.

If this option is a relative path, it is interpreted as being relative to the current working directory from which xtrabackup is executed.

In order to perform a backup, you need `READ`, `WRITE`, and `EXECUTE` permissions at a filesystem level for the directory that you supply as the value of `--target-dir`.

–INNODB-TEMP-TABLESPACES-DIR(=DIRECTORY)

Directory where temp tablespace files live, this path can be absolute.

–THROTTLE(=#)

This option limits the number of chunks copied per second. The chunk size is 10 MB.

To limit the bandwidth to 10 MB/s, set the option to 1.

–TLS-CIPHERSUITES()

TLS v1.3 cipher to use.

–TLS-VERSION()

TLS version to use, permitted values are: TLSv1, TLSv1.1, TLSv1.2, TLSv1.3.

Percona and/or its affiliates © 2023 — Cookie Consent

–TMPDIR(=NAME)

Specify the directory that will be used to store temporary files during the backup

–TRANSITION-KEY(=NAME)

This option is used to enable processing the backup without accessing the keyring vault server. In this case, xtrabackup derives the AES encryption key from the specified passphrase and uses it to encrypt tablespace keys of tablespaces being backed up.

If `--transition-key` does not have any value, xtrabackup will ask for it. The same passphrase should be specified for the `--prepare` command.

–USE-FREE-MEMORY-PCT()

The `--use-free-memory-pct` is a tech preview option. Before using this option in production, we recommend that you test restoring production from physical backups in your environment, and also use the alternative backup method for redundancy.

Implemented in Percona XtraBackup 8.0.30-23, this option lets you configure the Smart memory estimation feature. The option controls the amount of free memory that can be used to `--prepare` a backup. The default value is 0 (zero) which defines the option as disabled. For example, if you set `--use-free-memory-pct=50`, then 50% of the free memory is used to `prepare` a backup. The maximum allowed value is 100.

This option works, only if `--estimate-memory` option is enabled. If the `--estimate-memory` option is disabled, the `--use-free-memory-pct` setting is ignored.

An example of how to enable the Smart memory estimation feature:

```
$ xtrabackup --backup --estimate-memory=ON --target-dir=/data/backups/
```

```
$ xtrabackup --prepare --use-free-memory-pct=50 --target-dir=/data/backups/
```

–USE-MEMORY()

This option affects how much memory is allocated and is similar to `innodb_buffer_pool_size`. This option is only relevant in the `--prepare` phase. The default value is 100MB. The recommended value is between 1GB to 2GB. Multiple values are supported if you provide the unit (for example, 1MB, 1M, 1GB, 1G).

–USER(=USERNAME)

This option specifies the MySQL username used when connecting to the server, if that's not the current user. The option accepts a string argument. See mysql –help for details.

-V()

See `--version`

–VERSION()

This option prints xtrabackup version and exits.

–XTRABACKUP-PLUGIN-DIR(=DIRNAME)

The absolute path to the directory that contains the `keyring` plugin.

Percona LLC and/or its affiliates © 2023 — Cookie Consent

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-10-17

5.2.3 xtrabackup implementation details

This page contains notes on various internal aspects of the *xtrabackup* tool's operation.

File permissions

xtrabackup opens the source data files in read-write mode, although it does not modify the files. This means that you must run *xtrabackup* as a user who has permission to write the data files. The reason for opening the files in read-write mode is that *xtrabackup* uses the embedded *InnoDB* libraries to open and read the files, and *InnoDB* opens them in read-write mode because it normally assumes it is going to write to them.

Tune the OS buffers

Because *xtrabackup* reads large amounts of data from the filesystem, it uses `posix_fadvise()` where possible, to instruct the operating system not to try to cache the blocks it reads from disk. Without this hint, the operating system would prefer to cache the blocks, assuming that `xtrabackup` is likely to need them again, which is not the case. Caching such large files can place pressure on the operating system's virtual memory and cause other processes, such as the database server, to be swapped out. The `xtrabackup` tool avoids this with the following hint on both the source and destination files:

```
posix_fadvise(file, 0, 0, POSIX_FADV_DONTNEED)
```

In addition, xtrabackup asks the operating system to perform more aggressive read-ahead optimizations on the source files:

```
posix_fadvise(file, 0, 0, POSIX_FADV_SEQUENTIAL)
```

Copy data files

When copying the data files to the target directory, *xtrabackup* reads and writes 1 MB of data at a time. This is not configurable. When copying the log file, *xtrabackup* reads and writes 512 bytes at a time. This is also not possible to configure, and matches InnoDB's behavior (workaround exists in *Percona Server for MySQL* because it has an option to tune `innodb_log_block_size` for *XtraDB*, and in that case *Percona XtraBackup* will match the tuning).

Percona and/or its affiliates © 2023 — Cookie Consent

After reading from the files, `xtrabackup` iterates over the 1MB buffer a page at a time, and checks for page corruption on each page with InnoDB's `buf_page_is_corrupted()` function. If the page is corrupt, it re-reads and retries up to 10 times for each page. It skips this check on the doublewrite buffer.

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-06-12

5.2.4 Configure xtrabackup

All the xtrabackup configuration is done through options, which behave exactly like standard MySQL program options: they can be specified either at the command-line, or through a file such as `/etc/my.cnf`.

The xtrabackup binary reads the `[mysqld]` and `[xtrabackup]` sections from any configuration files, in that order. That is so that it can read its options from your existing MySQL installation, such as the datadir or some of the InnoDB options. If you want to override these, just specify them in the `[xtrabackup]` section, and because it is read later, it will take precedence.

You don't need to put any configuration in your `my.cnf`. You can specify the options on the command-line. Normally, the only thing you may find convenient to place in the `[xtrabackup]` section of your `my.cnf` file is the `target_dir` option. This options adds a default path to the directory for backups.

The following is an example:

```
[xtrabackup]
target_dir = /data/backups/
```

This manual assumes you do not have any file-based configuration for xtrabackup and shows the command-line options.

Please see the option and variable reference for details on all the configuration options.

The xtrabackup binary does not accept exactly the same syntax in the `my.cnf` file as the mysqld server binary does. For historical reasons, the mysqld server binary accepts parameters with a `--set-variable=<variable>=<value>` syntax, which xtrabackup does not understand. If your `my.cnf` file has such configuration directives, you should rewrite them in the `--variable=value` syntax.

System configuration and NFS volumes

The xtrabackup tool requires no special configuration on most systems. However, the storage where the `--target-dir` is located must behave properly when `fsync()` is called. In particular, we have noticed that if you mount the NFS volume without the `sync` option the NFS volume does not sync the

Percona LLC and/or its affiliates © 2023 — Cookie Consent

data. As a result, if you back up to an NFS volume mounted with the async option, and then try to prepare the backup from a different server that also mounts that volume, the data might appear to be corrupt. Use the `sync` mount option to avoid this issue.

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-08-17

5.3 The xbcrypt binary overview

To support encryption and decryption of the backups, a new tool `xbcrypt` was introduced to *Percona XtraBackup*.

The `XBCRYPT_ENCRYPTION_KEY` environment variable is only used in place of the `--encrypt_key=name` option. You can use the environment variable or command line option. If you use both, the command line option takes precedence over the value specified in the environment variable.

This utility has been modeled after The xbstream binary to perform encryption and decryption outside of *Percona XtraBackup*. `xbcrypt` has following command line options:

-d(, –decrypt()

Decrypt data input to output.

-i(, –input(=name)

Optional input file. If not specified, input will be read from standard input.

-o(, –output(=name)

Optional output file. If not specified, output will be written to standard output.

-a(, –encrypt-algo(=name)

Encryption algorithm.

-k(, –encrypt-key(=name)

Encryption key.

-f(, –encrypt-key-file(=name)

File which contains encryption key.

Percona LLC and/or its affiliates © 2023 — Cookie Consent

-s(, –encrypt-chunk-size(=#)

Size of working buffer for encryption in bytes. The default value is 64K.

–encrypt-threads(=#)

This option specifies the number of worker threads that will be used for parallel encryption/decryption.

-v(, –verbose()

Display verbose status output.

5.3.1 Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-08-28

5.4 The xbstream binary

5.4.1 The xbstream binary overview

To support simultaneous compression and streaming, a new custom streaming format called xbstream was introduced to Percona XtraBackup in addition to the TAR format. That was required to overcome some limitations of traditional archive formats such as tar, cpio and others which did not allow streaming dynamically generated files, for example dynamically compressed files. Other advantages of xbstream over traditional streaming/archive format include ability to stream multiple files concurrently (so it is possible to use streaming in the xbstream format together with the –parallel option) and more compact data storage.

Percona LLC and/or its affiliates © 2023 — Cookie Consent

This utility has a tar-like interface:

- with the `-x` option it extracts files from the stream read from its standard input to the current directory unless specified otherwise with the `-c` option. The parallel extraction is supported with the `--parallel` option.
- with the `-c` option it streams files specified on the command line to its standard output.
- with the `--decrypt=ALGO` option specified xbstream will automatically decrypt encrypted files when extracting input stream. Supported values for this option are: `AES128`, `AES192`, and `AES256`. Either `--encrypt-key` or `--encrypt-key-file` options must be specified to provide encryption key, but not both.
- with the `--encrypt-threads` option you can specify the number of threads for parallel data encryption. The default value is `1`.
- the `--encrypt-key` option is used to specify the encryption key that will be used. It can't be used with `--encrypt-key-file` option because they are mutually exclusive.
- the `--encrypt-key-file` option is used to specify the file that contains the encryption key. It can't be used with `--encrypt-key` option. because they are mutually exclusive.

The utility also tries to minimize its impact on the OS page cache by using the appropriate `posix_fadvise()` calls when available.

When compression is enabled with *xtrabackup* all data is being compressed, including the transaction log file and meta data files, using the specified compression algorithm. Percona XtraBackup supports the following compression algorithms:

Zstandard (ZSTD)

The Zstandard (ZSTD) is a fast lossless compression algorithm that targets real-time compression scenarios and better compression ratios. `ZSTD` is the default compression algorithm for the `--compress` option.

To compress files using the `ZSTD` compression algorithm, use the `--compress` option:

```
$ xtrabackup --backup --compress --target-dir=/data/backup
```

The resulting files have the `\*.zst` format.

You can specify `ZSTD` compression level with the `--compress-zstd-level(=#)` option. The default value is `1`.

```
$ xtrabackup –backup –compress –compress-zstd-level=1 –target-dir=/data/backup
```

lz4

To compress files using the `lz4` compression algorithm, set the `--compress` option to `lz4`:

```
$ xtrabackup --backup --compress=lz4 --target-dir=/data/backup
```

The resulting files have the `\*.lz4` format.

To decompress files, use the `--decompress` option.

Percona LLC and/or its affiliates © 2023 — Cookie Consent

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-10-10

5.4.2 Take a streaming backup

Percona XtraBackup supports streaming mode. Streaming mode sends a backup to `STDOUT` in the *xbstream* format instead of copying the files to the backup directory.

This method allows you to use other programs to filter the output of the backup, providing greater flexibility for storage of the backup. For example, compression is achieved by piping the output to a compression utility. One of the benefits of streaming backups and using Unix pipes is that the backups can be automatically encrypted.

To use the streaming feature, you must use the `--stream`, providing the format of the stream (`xbstream`) and where to store the temporary files:

```
$ xtrabackup --stream=xbstream --target-dir=/tmp
```

xtrabackup uses *xbstream* to stream all of the data files to `STDOUT`, in a special `xbstream` format. After it finishes streaming all of the data files to `STDOUT`, it stops xtrabackup and streams the saved log file too.

When compression is enabled, *xtrabackup* compresses the output data, except for the meta and non-InnoDB files which are not compressed, using the specified compression algorithm. Percona XtraBackup supports the following compression algorithms:

Zstandard (ZSTD)

The Zstandard (ZSTD) is a fast lossless compression algorithm that targets real-time compression scenarios and better compression ratios. `ZSTD` is the default compression algorithm for the `--compress` option.

To compress files using the `ZSTD` compression algorithm, use the `--compress` option:

```
$ xtrabackup --backup --compress --target-dir=/data/backup
```

The resulting files have the `\*.zst` format.

You can specify `ZSTD` compression level with the `--compress-zstd-level(=#)` option. The default value is `1`.

```
$ xtrabackup –backup –compress –compress-zstd-level=1 –target-dir=/data/backup
```


Percona and/or its affiliates © 2023 — Cookie Consent

lz4

To compress files using the `lz4` compression algorithm, set the `--compress` option to `lz4`:

```
$ xtrabackup --backup --compress=lz4 --target-dir=/data/backup
```

The resulting files have the `\*.lz4` format.

To decompress files, use the `--decompress` option.

Using *xbstream* as a stream option, backups can be copied and compressed in parallel. This option can significantly improve the speed of the backup process. In case backups were both compressed and encrypted, they must be decrypted before they are uncompressed.

Task	Command
Stream the backup into an archived named `backup.xbstream`	`$ xtrabackup --backup --stream=xbstream --target-dir=./ > backup.xbstream`
Stream the backup into a compressed archive named `backup.xbstream`	`$ xtrabackup --backup --stream=xbstream --compress --target-dir=./ > backup.xbstream`
Encrypt the backup	`$ xtrabackup --backup --stream=xbstream ./ > backup.xbstream gzip -`` | openssl des3 -salt -k “password” backup.xbstream.gz.des3`
Unpack the backup to the current directory	`$ xbstream -x <  backup.xbstream`
Send the backup compressed directly to another host and unpack it	`$ xtrabackup --backup --compress --stream=xbstream --target-dir=./ | ssh user@otherhost "xbstream -x"`

Send the backup to another server using `netcat`	On the destination host:
	`$ nc -l 9999 | cat - > /data/backups/backup.xbstream`
	
	On the source host:
	`$ xtrabackup --backup --stream=xbstream ./ | nc desthost 9999`

Send the backup to another server using a one-liner	`$ ssh user@desthost “( nc -l 9999 > /data/backups/backup.xbstream & )” && xtrabackup --backup --stream=xbstream ./ | nc desthost 9999`
Throttle the throughput to 10MB/sec using the pipe viewer tool	`$ xtrabackup --backup --stream=xbstream ./ | pv -q -L10m ssh user@desthost “cat - > /data/backups/backup.xbstream”`

Checksum the backup during the streaming	On the destination host:
	`$ nc -l 9999 | tee >(sha1sum > destination_checksum) > /data/backups/backup.xbstream`
	
	On the source host:
	`$ xtrabackup --backup --stream=xbstream ./ | tee >(sha1sum > source_checksum) | nc desthost 9999`
	
	Compare the checksums on the source host:
	`$ cat source_checksum 65e4f916a49c1f216e0887ce54cf59bf3934dbad`
	
	Compare the checksums on the destination host:
	`$ cat destination_checksum 65e4f916a49c1f216e0887ce54cf59bf3934dbad`

Percona LLC and/or its affiliates © 2023 — Cookie Consent

Task	Command
Parallel compression with parallel copying backup	`$ xtrabackup --backup --compress --compress-threads=8 --stream=xbstream --parallel=4 --target-dir=./ > backup.xbstream`

Important

The streamed backup must be prepared before restoration. Streaming mode does not prepare the backup.

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-10-12

5.4.3 Accelerate the backup process

Copy with the `--parallel` and `--compress-threads` options

When making a local or streaming backup with xbstream binary, multiple files can be copied at the same time when using the `--parallel` option. This option specifies the number of threads created by xtrabackup to copy data files.

To take advantage of this option either the multiple tablespaces option must be enabled (innodb_file_per_table) or the shared tablespace must be stored in multiple ibdata files with the innodb_data_file_path option. Having multiple files for the database (or splitting one into many) doesn't have a measurable impact on performance.

As this feature is implemented at the file level, concurrent file transfer can sometimes increase I/O throughput when doing a backup on highly fragmented data files, due to the overlap of a greater number of random read requests. You should consider tuning the filesystem also to obtain the maximum performance (e.g. checking fragmentation).

If the data is stored on a single file, this option has no effect.

To use this feature, simply add the option to a local backup, for example:

```
$ xtrabackup --backup --parallel=4 --target-dir=/path/to/backup
```

By using the xbstream in streaming backups, you can additionally speed up the compression process with the `--compress-threads` option. This option specifies the number of threads created by xtrabackup for for parallel data compression. The default value for this option is 1.

To use this feature, simply add the option to a local backup, for example:

Percona LLC and/or its affiliates © 2023 — Cookie Consent

```
$ xtrabackup --backup --stream=xbstream --compress --compress-threads=4 --target-
dir=./ > backup.xbstream
```

Before applying logs, compressed files will need to be uncompressed.

The `--rsync` option

In order to speed up the backup process and to minimize the time `FLUSH TABLES WITH READ LOCK` is blocking the writes, the option `--rsync` should be used. When this option is specified, xtrabackup uses `rsync` to copy all non-InnoDB files instead of spawning a separate `cp` for each file, which can be much faster for servers with a large number of databases or tables. xtrabackup will call the `rsync` twice, once before the `FLUSH TABLES WITH READ LOCK` and once during to minimize the time the read lock is being held. During the second `rsync` call, it will only synchronize the changes to non-transactional data (if any) since the first call performed before the `FLUSH TABLES WITH READ LOCK`. Note that Percona XtraBackup will use Backup locks where available as a lightweight alternative to `FLUSH TABLES WITH READ LOCK`.

Percona XtraBackup uses these locks automatically to copy non-InnoDB data to avoid blocking Data manipulation language (DML) queries that modify InnoDB tables.

Note

This option cannot be used together with the `--stream` option.

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-10-12

5.4.4 Encrypt backups

Percona XtraBackup supports encrypting and decrypting local and streaming backups with xbstream option adding another layer of protection. The encryption is implemented using the `libgcrypt` library from GnuPG.

Percona and/or its affiliates © 2023 — Cookie Consent

Create encrypted backups

To make an encrypted backup the following options need to be specified (options `--encrypt-key` and `--encrypt-key-file` are mutually exclusive, i.e. just one of them needs to be provided):

- `--encrypt`
- `--encrypt-key`
- `--encrypt-key-file`

Both the `--encrypt-key` option and `--encrypt-key-file` option can be used to specify the encryption key. An encryption key can be generated with a command like `openssl rand -base64 24`

```
U2FsdGVkX19VPN7VM+lwNI0fePhjgnhgqmDBqbF3Bvs=
```

This value then can be used as the encryption key

THE `--ENCRYPT-KEY` OPTION

Example of the xtrabackup command using the `--encrypt-key` should look like this:

```
$  xtrabackup --backup --encrypt=AES256 --encrypt-
key="U2FsdGVkX19VPN7VM+lwNI0fePhjgnhgqmDBqbF3Bvs=" --target-dir=/data/backup
```

THE `--ENCRYPT-KEY-FILE` OPTION

Use the `--encrypt-key-file` option as follows:

```
$ xtrabackup --backup --encrypt=AES256 --encrypt-key-file=/data/backups/keyfile --
target-dir=/data/backup
```

Note

Depending on the text editor that you use to make the `KEYFILE`, the editor can automatically insert the CRLF (end of line) character. This will cause the key size to grow and thus making it invalid. The suggested way to create the file is by using the command line: `echo -n "U2FsdGVkX19VPN7VM+lwNI0fePhjgnhgqmDBqbF3Bvs=" > /data/backups/keyfile`.

Optimize the encryption process

Two new options are available for encrypted backups that can be used to speed up the encryption process. These are `--encrypt-threads` and `--encrypt-chunk-size`. By using the `--encrypt-threads` option multiple threads can be specified to be used for encryption in parallel. Option `--encrypt-chunk-size` can be used to specify the size (in bytes) of the working encryption buffer for each encryption thread (default is 64K).

Decrypt encrypted backups

Backups can be decrypted with The xbcrypt binary. The following one-liner can be used to encrypt the whole folder:

Percona LLC and/or its affiliates © 2023 — Cookie Consent

```
$ for i in `find . -iname "*\.xbcrypt"`; do xbcrypt -d --encrypt-key-file=/root/
secret_key --encrypt-algo=AES256 < $i > $(dirname $i)/$(basename $i .xbcrypt) && rm
$i; done
```

Percona XtraBackup `--decrypt` option has been implemented that can be used to decrypt the backups:

```
$ xtrabackup --decrypt=AES256 --encrypt-
key="U2FsdGVkX19VPN7VM+lwNI0fePhjgnhgqmDBqbF3Bvs=" --target-dir=/data/backup/
```

Percona XtraBackup doesn't automatically remove the encrypted files. In order to clean up the backup directory users should remove the `\*.xbcrypt` files.

Note

`--parallel` can be used with `--decrypt` option to decrypt multiple files simultaneously.

When the files are decrypted, the backup can be prepared.

Prepare encrypted backups

After the backups have been decrypted, they can be prepared in the same way as the standard full backups with the `--prepare` option:

```
$ xtrabackup --prepare --target-dir=/data/backup/
```

Restore encrypted backups

xtrabackup offers the `--copy-back` option to restore a backup to the server's datadir:

```
$ xtrabackup --copy-back --target-dir=/data/backup/
```

It will copy all the data-related files back to the server's datadir, determined by the server's `my.cnf` configuration file. You should check the last line of the output for a success message:

Expected output

```
150318 11:08:13  xtrabackup: completed OK!
```

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Percona and/or its affiliates © 2023 — Cookie Consent

Last update: 2023-08-17

5.5 The xbcloud binary

5.5.1 The xbcloud binary overview

The purpose of xbcloud is to download from the cloud and upload to the cloud the full or part of an xbstream archive. xbcloud will not overwrite the backup with the same name. xbcloud accepts input via a pipe from xbstream so that it can be invoked as a pipeline with xtrabackup to stream directly to the cloud without needing a local storage.

Note

In a Bash shell, the `$?` parameter returns the exit code from the last binary. If you use pipes, the ${PIPESTATUS[x]} array parameter returns the exit code for each binary in the pipe string.

```
$ xtrabackup --backup --stream=xbstream --target-dir=/storage/backups/ | xbcloud put [options]
full_backup
...
$ ${PIPESTATUS[x]}
0 0
$ true | false
$ echo $?
1

# with PIPESTATUS
$ true | false
$ echo ${PIPESTATUS[0]} ${PIPESTATUS[1]}
0 1
```

The xbcloud binary stores each chunk as a separate object with a name `backup_name/database/table.ibd.NNN...`, where `NNN...` is a 0-padded serial number of chunk within a file. The size of chunk produced by xtrabackup and xbstream changed to 10M.

To adjust the chunk size use `--read-buffer-size`. To adjust the chunk size for encrypted files, use `--read-buffer-size` and `--encrypt-chunk-size`.

xbcloud has three essential operations: *put*, *get*, and *delete*. With these operations, backups are created, stored, retrieved, restored, and deleted. xbcloud operations clearly map to similar operations within the AWS Amazon S3 API.

The Exponential Backoff feature increases the chances for the completion of a backup or a restore operation. When taking a backup, a chunk upload or download may fail if you have an unstable network connection or other network issues. This feature adds an exponential backoff, a sleep time, and retries the operations.

With the FIFO data sink feature, users with a streaming capacity of 10Gbps (typically on a Local Area Network (LAN)) can benefit from faster backups by streaming data in parallel to object storage.

Important

To prevent intermittent backup failures, update the curl utility in Debian 10.

Percona and/or its affiliates © 2023 — Cookie Consent

Supported cloud storage types

The following cloud storage types are supported:

- OpenStack Object Storage (Swift) - see Using the xbcloud binary with Swift
- Amazon Simple Storage (S3) - see Using the xbcloud binary with Amazon S3
- Azure Cloud Storage - see Using the xbcloud binary with Microsoft Azure Cloud Storage
- Google Cloud Storage (gcs) - see Using the xbcloud binary with Google Cloud Storage
- MinIO - see Using the xbcloud binary with MinIO

In addition to OpenStack Object Storage (Swift), which has been the only option for storing backups in a cloud storage until Percona XtraBackup 2.4.14, xbcloud supports Amazon S3, MinIO, and Google Cloud Storage. Other Amazon S3-compatible storages, such as Wasabi or Digital Ocean Spaces, are also supported.

See also

OpenStack Object Storage("Swift")

Amazon Simple Storage Service

MinIO

Google Cloud Storage

Wasabi

Digital Ocean Spaces

Usage

The following sample command creates a full backup:

```
$ xtrabackup --backup --stream=xbstream --target-dir=/storage/backups/ --extra-
lsndirk=/storage/backups/| xbcloud \
put [options] full_backup
```

An incremental backup only includes the changes since the last backup. The last backup can be either a full or incremental backup.

The following sample command creates an incremental backup:

```
$ xtrabackup --backup --stream=xbstream --incremental-basedir=/storage/backups \
--target-dir=/storage/inc-backup | xbcloud  put [options] inc_backup
```

To prepare an incremental backup, you must first download the full backup with the following command:

```
$ xtrabackup get [options] full_backup | xbstream -xv -C /tmp/full-backup
```

You must prepare the full backup:

Percona LLC and/or its affiliates © 2023 — Cookie Consent

```
$ xtrabackup --prepare --apply-log-only --target-dir=/tmp/full-backup
```

After the full backup has been prepared, download the incremental backup:

```
xbcloud get [options] inc_backup | xbstream -xv -C /tmp/inc-backup
```

The downloaded backup is prepared by running the following command:

```
$ xtrabackup --prepare --target-dir=/tmp/full-backup --incremental-dir=/tmp/inc-
backup
```

You do not need the full backup to restore only a specific database. You can specify only the tables to be restored:

```
xbcloud get [options] ibdata1 sakila/payment.ibd /tmp/partial/partial.xbs
```

An example of the code:

```
xbstream -xv -C /tmp/partial < /tmp/partial/partial.xbs
```

Supplying parameters

Each storage type has mandatory parameters that you can supply on the command line, in a configuration file, and via environment variables.

CONFIGURATION FILES

The parameters the values of which do not change frequently can be stored in `my.cnf` or in a custom configuration file. The following example is a template of configuration options under the `[xbcloud]` group:

```
[xbcloud]
storage=s3
s3-endpoint=http://localhost:9000/
s3-access-key=minio
s3-secret-key=minio123
s3-bucket=backupsx
s3-bucket-lookup=path
s3-api-version=4
```

Note

If you explicitly use a parameter on the command line and in a configuration file, xbcloud uses the value provided on the command line.

ENVIRONMENT VARIABLES

If you explicitly use a parameter on the command line, in a configuration file, and the corresponding environment variable contains a value, xbcloud uses the value provided on the command line or in the configuration file.

Percona and/or its affiliates © 2023 — Cookie Consent

SHORTCUTS

For all operations (put, get, and delete), you can use a shortcut to specify the storage type, bucket name, and backup name as one parameter instead of using three distinct parameters (–storage, –s3-bucket, and backup name per se).

Note

Use the following format: `storage-type://bucket-name/backup-name`

In this example s3 refers to a storage type, operator-testing is a bucket name, and bak22 is the backup name.

```
$ xbcloud get s3://operator-testing/bak22 ...
```

This shortcut expands as follows:

```
$ xbcloud get --storage=s3 --s3-bucket=operator-testing bak22 ...
```

You can supply the mandatory parameters on the command line, configuration files, and in environment variables.

ADDITIONAL PARAMETERS

xbcloud accepts additional parameters that you can use with any storage type. The `--md5` parameter computes the MD5 hash value of the backup chunks. The result is stored in files that following the `backup_name.md5` pattern.

```
$ xtrabackup --backup --stream=xbstream \
--parallel=8 2>backup.log | xbcloud put s3://operator-testing/bak22 \
--parallel=8 --md5 2>upload.log
```

You may use the `--header` parameter to pass an additional HTTP header with the server side encryption while specifying a customer key.

An example of using the `--header` for AES256 encryption.

```
$ xtrabackup --backup --stream=xbstream --parallel=4 | \
xbcloud put s3://operator-testing/bak-enc/ \
--header="X-Amz-Server-Side-Encryption-Customer-Algorithm: AES256" \
--header="X-Amz-Server-Side-Encryption-Customer-Key: CuStoMerKey=" \
--header="X-Amz-Server-Side-Encryption-Customer-Key-MD5: CuStoMerKeyMd5==" \
--parallel=8
```

The `--header` parameter is also useful to set the access control list (ACL) permissions: `--header="x-amz-acl: bucket-owner-full-control`

Incremental backups

First, you need to make the full backup on which the incremental one is going to be based:

```
$ xtrabackup --backup --stream=xbstream --extra-lsndir=/storage/backups/ \
--target-dir=/storage/backups/ | xbcloud put \
--storage=swift --swift-container=test_backup \
--swift-auth-version=2.0 --swift-user=admin \
```


Percona LLC and/or its affiliates © 2023 — Cookie Consent

```
--swift-tenant=admin --swift-password=xoxoxoxo \
--swift-auth-url=http://127.0.0.1:35357/ --parallel=10 \
full_backup
```

Then you can make the incremental backup:

```
$ xtrabackup --backup --incremental-basedir=/storage/backups \
--stream=xbstream --target-dir=/storage/inc_backup | xbcloud put \
--storage=swift --swift-container=test_backup \
--swift-auth-version=2.0 --swift-user=admin \
--swift-tenant=admin --swift-password=xoxoxoxo \
--swift-auth-url=http://127.0.0.1:35357/ --parallel=10 \
inc_backup
```

PREPARING INCREMENTAL BACKUPS

To prepare a backup you first need to download the full backup:

```
$ xbcloud get --swift-container=test_backup \
--swift-auth-version=2.0 --swift-user=admin \
--swift-tenant=admin --swift-password=xoxoxoxo \
--swift-auth-url=http://127.0.0.1:35357/ --parallel=10 \
full_backup | xbstream -xv -C /storage/downloaded_full
```

Once you download the full backup it should be prepared:

```
$ xtrabackup --prepare --apply-log-only --target-dir=/storage/downloaded_full
```

After the full backup has been prepared you can download the incremental backup:

```
$ xbcloud get --swift-container=test_backup \
--swift-auth-version=2.0 --swift-user=admin \
--swift-tenant=admin --swift-password=xoxoxoxo \
--swift-auth-url=http://127.0.0.1:35357/ --parallel=10 \
inc_backup | xbstream -xv -C /storage/downloaded_inc
```

Once the incremental backup has been downloaded you can prepare it by running:

```
$ xtrabackup --prepare --apply-log-only \
--target-dir=/storage/downloaded_full \
--incremental-dir=/storage/downloaded_inc

$ xtrabackup --prepare --target-dir=/storage/downloaded_full
```

PARTIAL DOWNLOAD OF THE CLOUD BACKUP

If you do not want to download the entire backup to restore the specific database you can specify only the tables you want to restore:

```
$ xbcloud get --swift-container=test_backup
--swift-auth-version=2.0 --swift-user=admin \
--swift-tenant=admin --swift-password=xoxoxoxo \
--swift-auth-url=http://127.0.0.1:35357/ full_backup \
ibdata1 sakila/payment.ibd \
```


Percona and/or its affiliates © 2023 — Cookie Consent

```
> /storage/partial/partial.xbs

$ xbstream -xv -C /storage/partial < /storage/partial/partial.xbs
```

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-08-28

5.5.2 Use the xbcloud binary

Use the xbcloud binary with Amazon S3

CREATE A FULL BACKUP WITH AMAZON S3

```
$ xtrabackup --backup --stream=xbstream --extra-lsndir=/tmp --target-dir=/tmp | \
xbcloud put --storage=s3 \
--s3-endpoint='s3.amazonaws.com' \
--s3-access-key='YOUR-ACCESSKEYID' \
--s3-secret-key='YOUR-SECRETACCESSKEY' \
--s3-bucket='mysql_backups'
--parallel=10 \
$(date -I)-full_backup
```

The following options are available when using Amazon S3:

Option	Details
–s3-access-key	Use to supply the AWS access key ID
–s3-secret-key	Use to supply the AWS secret access key
–s3-bucket	Use supply the AWS bucket name
–s3-region	Use to specify the AWS region. The default value is **us-east-1**
–s3-api-version =	Select the signing algorithm. The default value is AUTO. In this case, xbcloud will probe.
–s3-bucket-lookup =	Specify whether to use bucket.endpoint.com or endpoint.com/bucket*style requests. The default value is AUTO. In this case, xbcloud will probe.
–s3-storage-class=	Specify the S3 storage class. The default storage class depends on the provider. The name options are the following: • STANDARD • STANDARD_IA • GLACIER

Percona LLC and/or its affiliates © 2023 — Cookie Consent

Option	Details
	NOTE If you use the GLACIER storage class, the object must be restored to S3 before restoring the backup. Also supports using custom S3 implementations such as MinIO or CephRadosGW.

PERMISSIONS SETUP

Following the principle of “least-privilege”, these are the minimum bucket permissions needed for xbcloud to write backups to S3: ***LIST/PUT/GET/DELETE***.

The following example shows the policy definition for writing to the `xbcloud-testing` bucket on the AWS S3 storage.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket"
            ],
            "Resource": "arn:aws:s3:::xbcloud-testing"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:PutObjectAcl",
                "s3:GetObject",
                "s3:GetObjectAcl",
                "s3:DeleteObject"
            ],
            "Resource": "arn:aws:s3:::xbcloud-testing/*"
        }
    ]
}
```

ENVIRONMENT VARIABLES

The following environment variables are recognized. xbcloud maps them automatically to corresponding parameters applicable to the selected storage.

- AWS_ACCESS_KEY_ID (or ACCESS_KEY_ID)
- AWS_SECRET_ACCESS_KEY (or SECRET_ACCESS_KEY)
- AWS_DEFAULT_REGION (or DEFAULT_REGION)
- AWS_ENDPOINT (or ENDPOINT)
- AWS_CA_BUNDLE

RESTORE WITH S3

```
$ xbcloud get s3://operator-testing/bak22 \
--s3-endpoint=https://storage.googleapis.com/ \
--parallel=10 2>download.log | xbstream -x -C restore --parallel=8
```


Percona LLC and/or its affiliates © 2023 — Cookie Consent

GET EXPERT HELP

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-06-12

Use the xbcloud binary with an IAM instance profile

You can use the IAM instance profile when running xbcloud from an EC2 instance.

An authentication system has two elements:

- Who am I?
- What can I do?

A role defines “what can I do.” A role provides a method to define a collection of permissions. Roles are assigned to users, services and EC2 instances, the “who am I” element.

The IAM instance profile is the “who” for an EC2 instance and assumes the IAM role, which has permissions. The instance profile has the same name as the IAM role.

An IAM instance profile does not need the `--s3-secret-key` nor the `--s3-access-key` if they are running `xbcloud` from an Amazon EC2 instance. To configure or attach an instance metadata to an EC2 instance, see How can I grant my Amazon EC2 instance access to an Amazon S3 bucket.

An example of the command:

```
$ xtrabackup ... | xbcloud put --storage=s3 --s3-bucket=bucket-name backup-name
```

The xbcloud binary outputs a connect message when successful.

Expected output

```
221121 13:16:26 Using instance metadata for access and secret key
221121 13:16:26 xbcloud: Successfully connected.
```

An important consideration is that the instance metadata has a time to live (TTL) of 6 hours. A backup that takes more than that time contains Expired token errors. Use Exponential Backoff to retry the upload after fetching new keys from the instance metadata.

Percona LLC and/or its affiliates © 2023 — Cookie Consent

Output when keys have expired

```
221121 13:04:52 xbcloud: S3 error message: The provided token has expired.
221121 13:04:52 xbcloud: Sleeping for 2384 ms before retrying test/mysql.ibd.
00000000000000000002 [1]
221121 13:04:55 xbcloud: S3 error message: The provided token has expired.
221121 13:04:55 xbcloud: Sleeping for 2887 ms before retrying test/mysql.ibd.
00000000000000000003 [1]
221121 13:04:58 xbcloud: S3 error message: The provided token has expired.
221121 13:04:58 xbcloud: Sleeping for 2778 ms before retrying test/undo_002.00000000000000000000
[1]
221121 13:05:00 xbcloud: S3 error message: The provided token has expired.
221121 13:05:00 xbcloud: Sleeping for 2916 ms before retrying test/undo_002.00000000000000000001
[1]
221121 13:05:03 xbcloud: S3 error message: The provided token has expired.
221121 13:05:03 xbcloud: Sleeping for 2794 ms before retrying test/undo_002.00000000000000000002
[1]
221121 13:05:06 xbcloud: S3 error message: The provided token has expired.
221121 13:05:06 xbcloud: Sleeping for 2336 ms before retrying test/undo_001.00000000000000000000
[1]
221121 13:05:09 xbcloud: successfully uploaded chunk: test/mysql.ibd.00000000000000000002, size:
5242923
221121 13:05:09 xbcloud: successfully uploaded chunk: test/mysql.ibd.00000000000000000003, size:
23
221121 13:05:09 xbcloud: successfully uploaded chunk: test/undo_002.00000000000000000000, size:
10485802
221121 13:05:09 xbcloud: successfully uploaded chunk: test/undo_002.00000000000000000001, size:
6291498
221121 13:05:09 xbcloud: successfully uploaded chunk: test/undo_002.00000000000000000002, size:
22
221121 13:05:09 xbcloud: successfully uploaded chunk: test/undo_001.00000000000000000000, size:
10485802
221121 13:05:10 xbcloud: successfully uploaded chunk: test/undo_001.00000000000000000001, size:
6291498
221121 13:05:10 xbcloud: successfully uploaded chunk: test/undo_001.00000000000000000002, size:
22
. . .
221121 13:05:18 xbcloud: successfully uploaded chunk: test/xtrabackup_tablespaces.
00000000000000000001, size: 36
221121 13:05:19 xbcloud: Upload completed.
```

GET EXPERT HELP

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-09-07

Use the xbcloud binary with Swift

CREATE A FULL BACKUP WITH SWIFT

The following example shows how to make a full backup and upload it to Swift.

Percona LLC and/or its affiliates © 2023 — Cookie Consent

```
$ xtrabackup --backup --stream=xbstream --extra-lsndir=/tmp --target-dir=/tmp | \
xbcloud put --storage=swift \
--swift-container=test \
--swift-user=test:tester \
--swift-auth-url=http://192.168.8.80:8080/ \
--swift-key=testing \
--parallel=10 \
full_backup
```

The following OpenStack environment variables are also recognized and mapped automatically to the corresponding swift parameters (`--storage=swift`):

- OS_AUTH_URL
- OS_TENANT_NAME
- OS_TENANT_ID
- OS_USERNAME
- OS_PASSWORD
- OS_USER_DOMAIN
- OS_USER_DOMAIN_ID
- OS_PROJECT_DOMAIN
- OS_PROJECT_DOMAIN_ID
- OS_REGION_NAME
- OS_STORAGE_URL
- OS_CACERT

RESTORE WITH SWIFT

```
$ xbcloud get [options] <name> [<list-of-files>] | xbstream -x
```

The following example shows how to fetch and restore the backup from Swift:

```
$ xbcloud get --storage=swift \
--swift-container=test \
--swift-user=test:tester \
--swift-auth-url=http://192.168.8.80:8080/ \
--swift-key=testing \
full_backup | xbstream -xv -C /tmp/downloaded_full

$ xbcloud delete --storage=swift --swift-user=xtrabackup \
--swift-password=xtrabackup123! --swift-auth-version=3 \
--swift-auth-url=http://openstack.ci.percona.com:5000/ \
--swift-container=mybackup1 --swift-domain=Default
```

COMMAND-LINE OPTIONS

xbcloud has the following command line options:

–storage(=[swift|s3|google])

Cloud storage option. *xbcloud* supports Swift, MinIO, and AWS S3. The default value is `swift` .

Percona LLC and/or its affiliates © 2023 — Cookie Consent

–swift-auth-url()

The URL of the Swift cluster

–swift-storage-url()

The xbcloud tries to get the object-store URL for a given region (if any are specified) from the keystone response. You can override that URL by passing –swift-storage-url=URL argument.

–swift-user()

The Swift username (X-Auth-User, specific to Swift)

–swift-key()

The Swift key/password (X-Auth-Key, specific to Swift)

–swift-container()

The container to back up into (specific to Swift)

–parallel(=N)

The maximum number of concurrent upload/download requests. The default value is `1`.

–cacert()

The path to the file with CA certificates

–insecure()

Do not verify server's certificate

Swift authentication options

The Swift specification describes several authentication options. The *xbcloud* tool can authenticate against keystone with API version 2 and 3.

–swift-auth-version()

Specifies the swift authentication version. The possible values are: `1.0` - TempAuth, `2.0` - Keystone v2.0, and `3` - Keystone v3. The default value is `1.0`.

For v2 additional options are:

–swift-tenant()

Swift tenant name

–swift-tenant-id()

Swift tenant ID

–swift-region()

Swift endpoint region

–swift-password()

Swift password for the user

For v3 additional options are:

Percona and/or its affiliates © 2023 — Cookie Consent

–swift-user-id()

Swift user ID

–swift-project()

Swift project name

–swift-project-id()

Swift project ID

–swift-domain()

Swift domain name

–swift-domain-id()

Swift domain ID

GET EXPERT HELP

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-06-12

Use the xbcloud binary with Google Cloud Storage

CREATE A FULL BACKUP WITH GOOGLE CLOUD STORAGE

The support for Google Cloud Storage is implemented using the interoperability mode. This mode was especially designed to interact with cloud services compatible with Amazon S3.

See also

Cloud Storage Interoperability

```
$ xtrabackup --backup --stream=xbstream --extra-lsndir=/tmp --target-dir=/tmp | \
xbcloud put --storage=google \
--google-endpoint=`storage.googleapis.com` \
--google-access-key='YOUR-ACCESSKEYID' \
--google-secret-key='YOUR-SECRETACCESSKEY' \
--google-bucket='mysql_backups'
--parallel=10 \
$(date -I)-full_backup
```


Percona and/or its affiliates © 2023 — Cookie Consent

The following options are available when using Google Cloud Storage:

- –google-access-key =
- –google-secret-key =
- –google-bucket =
- –google-storage-class=name

Note

The Google storage class name options are the following:

- STANDARD
- NEARLINE
- COLDLINE
- ARCHIVE

See also

Google storage classes - the default Google storage class depends on the storage class of the bucket

GET EXPERT HELP

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-06-12

Use the xbcloud binary with Microsoft Azure Cloud Storage

The xbcloud binary adds support for the Microsoft Azure Cloud Storage using the REST API.

OPTIONS

The following are the options, environment variables, and descriptions for uploading a backup to Azure using the REST API. The environment variables are recognized by xbcloud, which maps them automatically to the corresponding parameters:

Option name	Environment variables	Description
–azure-storage-account=name	AZURE_STORAGE_ACCOUNT	An Azure storage account is a unique namespace to access and store your Azure data objects.
–azure-container-name=name	AZURE_CONTAINER_NAME	A container name is a valid DNS name that conforms to the Azure naming rules
	AZURE_ACCESS_KEY	

Percona and/or its affiliates © 2023 — Cookie Consent

Option name	Environment variables	Description
–azure-access-key=name		A generated key that can be used to authorize access to data in your account using the Shared Key authorization.
–azure-endpoint=name	AZURE_ENDPOINT	The endpoint allows clients to securely access data
–azure-tier-class=name	AZURE_STORAGE_CLASS	Cloud tier can decrease the local storage required while maintaining the performance. When enabled, this feature has the following categories: Hot - Frequently accessed or modified data Cool - Infrequently accessed or modified data Archive - Rarely accessed or modified data

Test your Azure applications with the Azurite open-source emulator. For testing purposes, the xbcloud binary adds the `--azure-development-storage` option that uses the default `access_key` and `storage account` of azurite and `testcontainer` for the container. You can overwrite these options, if needed.

USAGE

All the available options for xbcloud, such as parallel, max-retries, and others, can be used. For more information, see the xbcloud binary overview.

EXAMPLES

An example of an xbcloud backup.

```
$ xtrabackup --backup --stream=xbstream  |
xbcloud put backup_name --azure-storage-account=pxbtesting --azure-access-
key=$AZURE_KEY --azure-container-name=test --storage=azure
```

An example of restoring a backup from xbcloud.

```
$ xbcloud get backup_name  --azure-storage-account=pxbtesting
--azure-access-key=$AZURE_KEY --azure-container-name=test --storage=azure --
parallel=10 2>download.log | xbstream -x -C restore
```

An example of deleting a backup from xbcloud.

```
$ xbcloud delete backup_name --azure-storage-account=pxbtesting
--azure-access-key=$AZURE_KEY --azure-container-name=test --storage=azure
```

An example of using a shortcut restore.

```
$ xbcloud get azure://operator-testing/bak22 ...
```


Percona LLC and/or its affiliates © 2023 — Cookie Consent

GET EXPERT HELP

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-08-17

Use the xbcloud binary with MinIO

CREATE A FULL BACKUP WITH MINIO

```
$ xtrabackup --backup --stream=xbstream --extra-lsndir=/tmp --target-dir=/tmp | \
xbcloud put --storage=s3 \
--s3-endpoint='play.minio.io:9000' \
--s3-access-key='YOUR-ACCESSKEYID' \
--s3-secret-key='YOUR-SECRETACCESSKEY' \
--s3-bucket='mysql_backups'
--parallel=10 \
$(date -I)-full_backup
```

GET EXPERT HELP

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-06-12

5.5.3 Update curl utility

Update the curl utility in Debian 10

The default curl version, 7.64.0, in Debian 10 has known issues when attempting to reuse an already closed connection. This issue directly affects `xbcloud` and users may see intermittent backup failures.

For more details, see curl #3750 or curl #3763.

Follow these steps to upgrade curl to version 7.74.0:

0. Edit the `/etc/apt/sources.list` to add the following:

```
deb http://ftp.de.debian.org/debian buster-backports main
```


Percona and/or its affiliates © 2023 — Cookie Consent

0. Refresh the `apt` sources:

```
$ sudo apt update
```

0. Install the version from `buster-backports`:

```
$ sudo apt install curl/buster-backports
```

0. Verify the version number:

```
$ curl --version
```

Expected output

```
curl 7.74.0 (x86_64-pc-linux-gnu) libcurl/7.74.0
```

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-10-04

5.5.4 FIFO data sink

The feature is in tech preview.

Percona XtraBackup implements a data sink that uses the first in, first out (FIFO) method. With the `FIFO` data sink feature, users with a streaming capacity of 10Gbps (typically on a Local Area Network (LAN)) can benefit from faster backups by streaming data in parallel to an object storage.

FIFO data sink options

Percona XtraBackup implements the following options:

- `--fifo-streams=#` - specifies the number of FIFO files to use for parallel data stream. To disable FIFO data sink and send stream to STDOUT, set `--fifo-streams=1`. The default value is `1` (STDOUT) to ensure the backward compatibility. The `--fifo-streams` value must match on both the XtraBackup and xbcloud sides.
- `--fifo-dir=name` - specifies a directory to write Named Pipe.
- `--fifo-timeout=#` - specifies the number of seconds to wait for the other end to open the stream for reading. The default value is `60` seconds.

Percona LLC and/or its affiliates © 2023 — Cookie Consent

How to enable FIFO data sink

To use FIFO data sink, you can either run two commands in separate terminal sessions or run xtrabackup in the background.

For example, run the following commands in separate terminal sessions:

```
$ xtrabackup --backup --stream --fifo-streams=2 --fifo-dir=/tmp/fifo
```

```
$ xbcloud put --fifo-streams=2 --fifo-dir=/tmp/fifo full
```

Run xtrabackup in the background with the following commands:

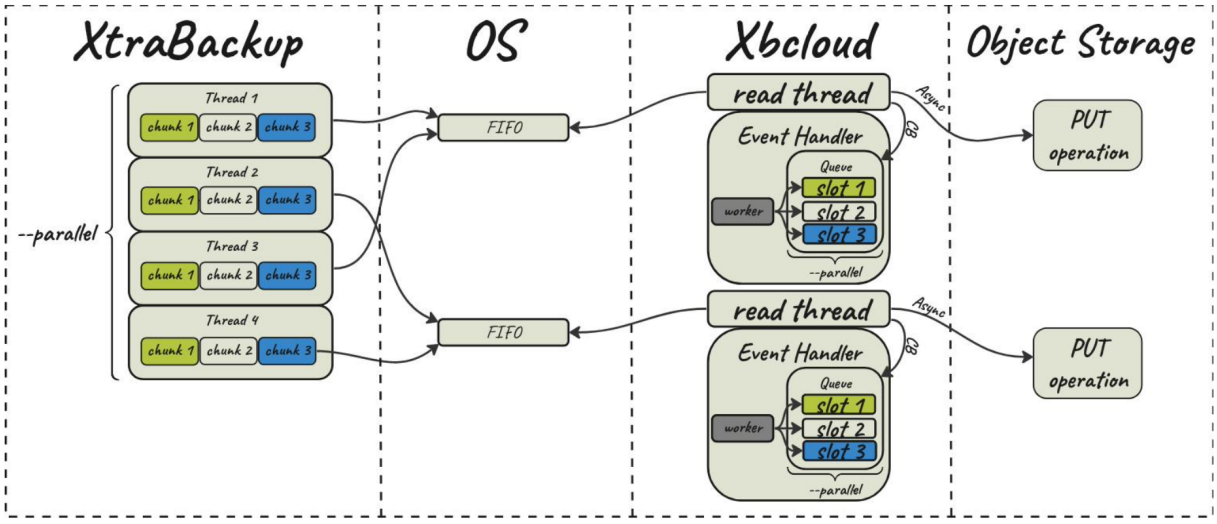
```
$ xtrabackup --backup --stream --fifo-streams=2 --fifo-dir=/tmp/fifo &
$ xbcloud put --fifo-streams=2 --fifo-dir=/tmp/fifo full
```

Stream to an object storage

When taking a backup, you can save the files locally or stream the files to either a different server or an object storage.

When you stream backups to Amazon S3 compatible storage using LAN with a streaming capacity of 10Gbps, XtraBackup can use multiple FIFO streams to stream the backups faster.

XtraBackup spawns multiple copy threads and each copy thread reads a data chunk from a specific file. Then multiple FIFO files are created to store the data from XtraBackup. Each XtraBackup copy thread writes the data chunks to a specific FIFO file. Xbcloud reads from the FIFO streams and uploads data to an object storage using an async request. The xbcloud event handler executes the callback depending on the response from the object storage (success or failure).



Performance improvement

Consider an example of using a FIFO data sink compared to the STDOUT method.

The database has 1TB of data in multiple tables. The link speed between the source server and destination server using MinIO is ~ 9.2 Gbps.

Percona LLC and/or its affiliates © 2023 — Cookie Consent

Both STDOUT and FIFO data sink scenarios push 1TB of data from two servers.

For the FIFO data sink we configure 8 parallel streams with `--fifo-streams=8` both for XtraBackup and xbcloud.

The results are the following:

- The `STDOUT` method takes 01:25:24 to push 1TB of data using 239 MBps (1.8 Gbps).
- The `FIFO` method, using 8 streams, takes 00:16:01 to push 1TB of data using 1.15 GBps (9.2 Gbps).

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-08-30

5.5.5 Exponential backoff

Exponential backoff increases the chances for the completion of a backup or a restore operation. For example, a chunk upload or download may fail if you have an unstable network connection or other network issues. This feature adds an exponential backoff, or sleep, time and then retries the upload or download.

When a chunk upload or download operation fails, xbcloud checks the reason for the failure. This failure can be a CURL error or an HTTP error, or a client-specific error. If the error is listed in the Retriable errors list, xbcloud pauses for a calculated time before retrying the operation until that time reaches the `--max-backoff` value.

The operation is retried until the `--max-retries` value is reached. If the chunk operation fails on the last retry, xbcloud aborts the process.

The default values are the following:

- –max-backoff = 300000 (5 minutes)
- –max-retries = 10

You can adjust the number of retries by adding the `--max-retries` parameter and adjust the maximum length of time between retries by adding the `--max-backoff` parameter to an xbcloud command.

Since xbcloud does multiple asynchronous requests in parallel, a calculated value, measured in milliseconds, is used for `max-backoff`. This algorithm calculates how many milliseconds to sleep before the next retry. A number generated is based on the combining the power of two (2), the number of retries already attempted and adds a random number between 1 and 1000. This number avoids network congestion if multiple chunks have the same backoff value. If the default values are used, the final retry attempt to be approximately 17 minutes after the first try. The number is no longer

Percona and/or its affiliates © 2023 — Cookie Consent

calculated when the milliseconds reach the `--max-backoff` setting. At that point, the retries continue by using the `--max-backoff` setting until the `max-retries` parameter is reached.

Retriable errors

We retry for the following CURL operations:

- CURLE_GOT_NOTHING
- CURLE_OPERATION_TIMEOUT
- CURLE_RECV_ERROR
- CURLE_SEND_ERROR
- CURLE_SEND_FAIL_REWIND
- CURLE_PARTIAL_FILE
- CURLE_SSL_CONNECT_ERROR

We retry for the following HTTP operation status codes:

- 503
- 500
- 504
- 408

Each cloud provider may return a different CURL error or an HTTP error, depending on the issue. Add new errors by setting the following variables `--curl-retriable-errors` or `--http-retriable-errors` on the command line or in `my.cnf` or in a custom configuration file under the [xbcloud] section. You can add multiple errors using a comma-separated list of codes.

The error handling is enhanced when using the `--verbose` output. This output specifies which error caused xbcloud to fail and what parameter a user must add to retry on this error.

The following is an example of a verbose output:

Expected output

```
210701 14:34:23 /work/pxb/ins/8.2/bin/xbcloud: Operation failed. Error: Server returned nothing
(no headers, no data)
210701 14:34:23 /work/pxb/ins/8.2/bin/xbcloud: Curl error (52) Server returned nothing (no
headers, no data) is not configured as retriable. You can allow it by adding --curl-retriable-
errors=52 parameter
```

Example

The following example adjusts the maximum number of retries and the maximum time between retries.

```
xbcloud [options] --max-retries=5 --max-backoff=10000
```


Percona LLC and/or its affiliates © 2023 — Cookie Consent

The following list describes the process using `--max-backoff=10000`:

- The chunk `xtrabackup_logfile.00000000000000000006` fails to upload the first time and sleeps for 2384 milliseconds.
- The same chunk fails for the second time and the sleep time is increased to 4387 milliseconds.
- The same chunk fails for the third time and the sleep time is increased to 8691 milliseconds.
- The same chunk fails for the fourth time. The `max-backoff` parameter has been reached. All retries sleep the same amount of time after reaching the parameter.
- The same chunk is successfully uploaded.

An example of the output for this setting

```
210702 10:07:05 /work/pxb/ins/8.2/bin/xbcloud: Operation failed. Error: Server returned nothing
(no headers, no data)
210702 10:07:05 /work/pxb/ins/8.2/bin/xbcloud: Sleeping for 2384 ms before retrying backup3/
xtrabackup_logfile.00000000000000000006
. . .
210702 10:07:23 /work/pxb/ins/8.2/bin/xbcloud: Operation failed. Error: Server returned nothing
(no headers, no data)
210702 10:07:23 /work/pxb/ins/8.2/bin/xbcloud: Sleeping for 4387 ms before retrying backup3/
xtrabackup_logfile.00000000000000000006
. . .
210702 10:07:52 /work/pxb/ins/8.2/bin/xbcloud: Operation failed. Error: Failed sending data to
the peer
210702 10:07:52 /work/pxb/ins/8.2/bin/xbcloud: Sleeping for 8691 ms before retrying backup3/
xtrabackup_logfile.00000000000000000006
. . .
210702 10:08:47 /work/pxb/ins/8.2/bin/xbcloud: Operation failed. Error: Failed sending data to
the peer
210702 10:08:47 /work/pxb/ins/8.2/bin/xbcloud: Sleeping for 10000 ms before retrying backup3/
xtrabackup_logfile.00000000000000000006
. . .
210702 10:10:12 /work/pxb/ins/8.2/bin/xbcloud: successfully uploaded chunk: backup3/
xtrabackup_logfile.00000000000000000006, size: 8388660
```

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-10-17

Percona and/or its affiliates © 2023 — Cookie Consent

6. Quickstart Guide

6.1 Quickstart Guide for Percona XtraBackup 8.2

Percona XtraBackup (PXB) is a 100% open source backup solution for all versions of Percona Server for MySQL and MySQL® that performs online non-blocking, tightly compressed, highly secure full backups on transactional systems. Maintain fully available applications during planned maintenance windows with Percona XtraBackup.

6.1.1 Install Percona XtraBackup:

You can install Percona XtraBackup using different methods:

- Use the Percona Repositories
- Use APT
- Use YUM
- Use binary tarballs
- Use Docker

6.1.2 For superior and optimized performance

Percona Server for MySQL (PS) is a freely available, fully compatible, enhanced, and open source drop-in replacement for any MySQL database. It provides superior and optimized performance, greater scalability and availability, enhanced backups, increased visibility, and instrumentation. Percona Server for MySQL is trusted by thousands of enterprises to provide better performance and concurrency for their most demanding workloads.

Install Percona Server for MySQL.

6.1.3 For high availability

Percona XtraDB Cluster (PXC) is a 100% open source, enterprise-grade, highly available clustering solution for MySQL multi-master setups based on Galera. PXC helps enterprises minimize unexpected downtime and data loss, reduce costs, and improve performance and scalability of your database environments supporting your critical business applications in the most demanding public, private, and hybrid cloud environments.

Install Percona XtraDB Cluster.

6.1.4 For Monitoring and Management

Percona Monitoring and Management (PMM) monitors and provides actionable performance data for MySQL variants, including Percona Server for MySQL, Percona XtraDB Cluster, Oracle MySQL Community Edition, Oracle MySQL Enterprise Edition, and MariaDB. PMM captures metrics and data for the InnoDB, XtraDB, and MyRocks storage engines, and has specialized dashboards for specific engine details.

Install PMM and connect your MySQL instances to it.

Percona and/or its affiliates © 2023 — Cookie Consent

6.1.5 Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum **Get a Percona Expert**

Last update: 2023-10-17

Percona and/or its affiliates © 2023 — Cookie Consent

7. Install

7.1 Install overview

We recommend that you install Percona XtraBackup 8.2 from the official Percona software repositories using the appropriate package manager for your system:

- Use an APT repo to install Percona XtraBackup
- Use a YUM repo to install Percona XtraBackup

7.1.1 Installation alternatives

Percona also provides the following methods:

- Use DEB downloaded packages to install Percona XtraBackup
- Use RPM downloaded packages to install Percona XtraBackup
- Install Percona XtraBackup from a Binary Tarball with all the required files and binaries for a manual installation
- Compile and install Percona XtraBackup from source code
- Run Percona XtraBackup in a Docker container

Before installing Percona XtraBackup with any of these methods, we recommend that you review the release notes and Server version and backup version comparison.

7.1.2 Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum **Get a Percona Expert**

Last update: 2023-11-02

7.2 Before you start

7.2.1 Server version and backup version comparison

A MySQL change to a feature, such as the structure of a redo log record, can cause older versions of Percona XtraBackup to fail. To ensure that you can back up and restore your data, use a Percona XtraBackup version that is equal to your source server major version. This means if you use Percona XtraBackup 8.2.x, you can safely back up the source server from 8.2.x to 8.2.xx.

Percona LLC and/or its affiliates © 2023 — Cookie Consent

See also

How XtraBackup works

The `--no-server-version-check` option performs the following test. Before the backup/prepare starts, XtraBackup compares the source server version to the Percona XtraBackup version. If the source server version is greater than the XtraBackup major version, XtraBackup stops the backup and returns an error message. This comparison prevents a failed backup or a corrupted backup due to source server changes.

The parameter checks for the following scenarios:

- The source server and the PXB major version are the same, the backup proceeds
- The source server is greater than the PXB major version, and the parameter is not overridden, the backup is stopped and returns an error message
- The source server is less than the PXB major version, and the parameter is not overridden, the backup is stopped and returns an error message
- The source server is greater than the PXB major version up to the last release in the LTS series, and the parameter is overridden, the backup proceeds

Explicitly adding the `--no-server-version-check` parameter, like the example, overrides the parameter and the backup proceeds.

```
$ xtrabackup --backup --no-server-version-check --target-dir=$mysql/backup1
```

Overriding the `--no-server-version-check` parameter allows taking backups using a Percona XtraBackup version that is equal to a version of your source server up to the last release in the LTS series. This means if you use Percona XtraBackup 8.2.x, you can back up the source server from 8.2.x to 8.4.xx.

When you override the parameter, the following events can happen:

- Backup fails
- Creates a corrupted backup
- Backup successful

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-10-12

Percona LLC and/or its affiliates © 2023 — Cookie Consent

7.2.2 Connection and privileges needed

Percona XtraBackup needs to be able to connect to the database server and perform operations on the server and the datadir when creating a backup, when preparing in some scenarios and when restoring it. In order to do so, there are privileges and permission requirements on its execution that must be fulfilled.

Privilege refers to the operations that a system user is permitted to do in the database server. They are set at the database server and only apply to users in the database server.

Permissions are those which permits a user to perform operations on the system, like reading, writing or executing on a certain directory or start/stop a system service. They are set at a system level and only apply to system users.

When xtrabackup is used, there are two actors involved: the user invoking the program - a system user - and the user performing action in the database server - a database user. Note that these are different users in different places, even though they may have the same username.

All the invocations of xtrabackup in this documentation assume that the system user has the appropriate permissions, and you are providing the relevant options for connecting the database server - besides the options for the action to be performed - and the database user has adequate privileges.

Connect to the server

The database user used to connect to the server and its password are specified by the `--user` and `--password` option:

```
$ xtrabackup --user=DVADER --password=14MY0URF4TH3R --backup \
--target-dir=/data/bkps/
```

If you don't use the `--user` option, Percona XtraBackup will assume the database user whose name is the system user executing it.

OTHER CONNECTION OPTIONS

According to your system, you may need to specify one or more of the following options to connect to the server:

Option	Description
-port	Use this port when connecting to the database with TCP/IP
-socket	Use this socket when connecting to the local database.
-host	Use this host when connecting to the database server with TCP/IP

These options are passed to the mysql child process without alteration, see `mysql --help` for details.

Note

In case of multiple server instances, the correct connection parameters (port, socket, host) must be specified in order for xtrabackup to talk to the correct server.

Percona and/or its affiliates © 2023 — Cookie Consent

Privileges needed

Once connected to the server, in order to perform a backup you need `READ` and `EXECUTE` permissions at a filesystem level in the server's datadir.

The database user needs the following privileges to back up tables or databases:

- `RELOAD` and `FLUSH_TABLES` in order to run `FLUSH TABLES WITH READ LOCK`.
- The `BACKUP_ADMIN` privilege is needed to query the performance_schema.log_status table, and run `LOCK INSTANCE FOR BACKUP`, `LOCK BINLOG FOR BACKUP`, or `LOCK TABLES FOR BACKUP`. The `BACKUP_ADMIN` privilege is required to use the Page tracking feature.
- `REPLICATION CLIENT` in order to obtain the binary log position,
- `CREATE TABLESPACE` in order to import tables (see Restoring Individual Tables),
- `PROCESS` in order to run `SHOW ENGINE INNODB STATUS` (which is mandatory), and optionally to see all threads which are running on the server (see FLUSH TABLES WITH READ LOCK option),
- `REPLICATION_SLAVE_ADMIN` in order to start/stop the replication threads in a replication environment,
- `CREATE` privilege in order to create the PERCONA_SCHEMA.xtrabackup_history database and table,
- `ALTER` privilege in order to upgrade the PERCONA_SCHEMA.xtrabackup_history database and table,
- `INSERT` privilege in order to add history records to the PERCONA_SCHEMA.xtrabackup_history table,
- `SELECT` privilege in order to use `--incremental-history-name` or `--incremental-history-uuid` in order for the feature to look up the `innodb_to_lsn` values in the PERCONA_SCHEMA.xtrabackup_history table.
- `SELECT` privilege on the keyring_component_status table to view the attributes and status of the installed keyring component when in use.
- `SELECT` privilege on the replication_group_members table to validate if the instance is part of group replication cluster.

A SQL example of creating a database user with the minimum privileges required to take full backups would be:

```
mysql> CREATE USER 'bkpuser'@'localhost' IDENTIFIED BY 's3cr%T';
mysql> GRANT BACKUP_ADMIN, PROCESS, RELOAD, LOCK TABLES, REPLICATION CLIENT ON *.*
TO 'bkpuser'@'localhost';
mysql> GRANT SELECT ON performance_schema.log_status TO 'bkpuser'@'localhost';
mysql> GRANT SELECT ON performance_schema.keyring_component_status TO
bkpuser@'localhost';
mysql> GRANT SELECT ON performance_schema.replication_group_members TO
bkpuser@'localhost';
mysql> FLUSH PRIVILEGES;
```

QUERY THE PRIVILEGES

To query the privileges that your database user has been granted at the console of the server execute:

```
mysql> SHOW GRANTS;
```

or for a particular user with:

Percona LLC and/or its affiliates © 2023 — Cookie Consent

```
mysql> SHOW GRANTS FOR 'db-user'@'host';
```

It will display the privileges using the same format as for the GRANT statement.

Note that privileges may vary across versions of the server. To list the exact list of privileges that your server support (and a brief description of them) execute:

```
mysql> SHOW PRIVILEGES;
```

See also

Permissions needed

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-10-19

7.2.3 Permissions needed

We will be referring to permissions to the ability of a user to access and perform changes on the relevant parts of the host's filesystem, starting/stopping services and installing software.

By privileges, we refer to the abilities of a database user to perform different kinds of actions on the database server.

There are many ways for checking the permission on a file or directory. For example, `ls -ls /path/to/file` or `stat /path/to/file | grep Access` will do the job:

```
$ stat /etc/mysql | grep Access
```

The result could look like this:

Expected output

```
Access: (0755/drwxr-xr-x)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2011-05-12 21:19:07.129850437 -0300
$ ls -ld /etc/mysql/my.cnf
-rw-r--r-- 1 root root 4703 Apr  5 06:26 /etc/mysql/my.cnf
```


Percona and/or its affiliates © 2023 — Cookie Consent

As in this example, `my.cnf` is owned by `root` and not writable for anyone else. Assuming that you do not have `root`'s password, you can check what permissions you have on these types of files with `sudo -l`:

```
$ sudo -l
```

The results could look like this:

Expected output

```
Password:
You may run the following commands on this host:
(root) /usr/bin/
(root) NOPASSWD: /etc/init.d/mysqld
(root) NOPASSWD: /bin/vi /etc/mysql/my.cnf
(root) NOPASSWD: /usr/local/bin/top
(root) NOPASSWD: /usr/bin/ls
(root) /bin/tail
```

Being able to execute with `sudo` scripts in `/etc/init.d/`, `/etc/rc.d/` or `/sbin/service` is the ability to start and stop services.

Also, If you can execute the package manager of your distribution, you can install or remove software with it. If not, having `rwx` permission over a directory will let you do a local installation of software by compiling it there. This is a typical situation in many hosting companies' services.

There are other ways for managing permissions, such as using PolicyKit, Extended ACLs or SELinux, which may be preventing or allowing your access. You should check them in that case.

See also

Connection and privileges needed

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-08-17

Percona and/or its affiliates © 2023 — Cookie Consent

7.3 Use APT

7.3.1 Install with an APT repository

Ready-to-use packages are available from the Percona XtraBackup software repositories and the download page.

Specific information on the supported platforms, products, and versions is described in Percona Release Lifecycle Overview.

Install Percona XtraBackup through percona-release

Percona XtraBackup, like many other Percona products, is installed with the *percona-release* package configuration tool.

0. Download a `DEB` package for *percona-release* the repository packages from Percona web:

```
$ wget https://repo.percona.com/apt/percona-release_latest.$(lsb_release -
sc)_all.deb
```

0. Install the downloaded package with **dpkg**. To do that, run the following commands as root or with **sudo**: `dpkg -i percona-release_latest.$(lsb_release -sc)_all.deb`

   Once you install this package the Percona repositories should be added. You can check the repository setup in the `/etc/apt/sources.list.d/percona-release.list` file.

0. Enable the repository: `percona-release enable-only tools release`

   If *Percona XtraBackup* is intended to be used in combination with the upstream MySQL Server, you only need to enable the `tools` repository: `percona-release enable-only tools`.

0. Refresh the local cache to update the package information:

```
$ sudo apt update
```

0. Install the `percona-xtrabackup-82` package:

```
$ sudo apt install percona-xtrabackup-82
```

0. To decompress backups made using `LZ4` or `ZSTD` compression algorithm, install the corresponding package:

Install the `lz4` package **Install the `zstd` package**

```
$ sudo apt install lz4

$ sudo apt install zstd
```


Percona LLC and/or its affiliates © 2023 — Cookie Consent

Note

For AppArmor profile information, see Working with AppArmor.

See also

To install Percona XtraBackup using downloaded deb packages, see Install Percona XtraBackup 8.2.

To uninstall Percona XtraBackup, see Uninstall Percona XtraBackup 8.2

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-10-17

7.3.2 Files in the DEB package

The following tables show what data each `DEB` package contains.

Package	Contains
percona-xtrabackup-82	The latest Percona XtraBackup GA binaries and associated files
percona-xtrabackup-dbg-82	The debug symbols for binaries in `percona-xtrabackup-82`
percona-xtrabackup-test-82	The test suite for Percona XtraBackup
percona-xtrabackup	The older version of the Percona XtraBackup

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-10-17

7.3.3 Install using downloaded DEB packages

Download `DEB` packages of the desired series for your architecture from Percona Product Downloads. This method requires you to resolve all dependencies and install any missing packages.

Percona and/or its affiliates © 2023 — Cookie Consent

The following example downloads Percona XtraBackup 8.2.0-1 release package for Ubuntu 20.04:

```
$ wget https://downloads.percona.com/downloads/Percona-XtraBackup-LATEST/Percona-
XtraBackup-8.2.0-1/binary/debian/focal/x86_64/percona-
xtrabackup-82_8.2.0-1.focal_amd64.deb
```

Install Percona XtraBackup by using `dpkg`. Run this command as root or use the sudo command:

```
$ sudo dpkg -i percona-xtrabackup-82_8.2.0-1.focal_amd64.deb
```

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-10-17

7.3.4 Apt pinning packages

In some cases you might need to `pin` the selected packages to avoid the upgrades from the distribution repositories.

The pinning takes place in the `preference` file. To pin a package, set the `Pin-Priority` to higher numbers.

Make a new file `/etc/apt/preferences.d/00percona.pref`. For example, add the following to the preference file:

```
Package:
Pin: release o=Percona Development Team
Pin-Priority: 1001
```

For more information about the pinning, check the official debian wiki.

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-10-23

Percona and/or its affiliates © 2023 — Cookie Consent

7.3.5 Work with AppArmor

The Linux Security Module implements mandatory access controls (MAC) with AppArmor. Debian and Ubuntu systems install AppArmor by default. AppArmor uses profiles which define which files and permissions are needed for application.

Percona XtraBackup does not have a profile and is not confined by AppArmor.

For a list of common AppArmor commands, see Percona Server for MySQL - AppArmor.

Develop a profile

Download the profile from:

https://github.com/percona/percona-xtrabackup/tree/trunk/packaging/percona/apparmor/apparmor.d

The following profile sections should be updated with your system information, such as location of the backup destination directory.

Expected output

```
# enable storing backups only in /backups directory
# /backups/** rwk,

# enable storing backups anywhere in caller user home directory
/@{HOME}/** rwk,


# enable storing backups only in /backups directory
# /backups/** rwk,

# enable storing backups anywhere in caller user home directory
/@{HOME}/** rwk,
}

# enable storing backups only in /backups directory
# /backups/** rwk,

# enable storing backups anywhere in caller user home directory
/@{HOME}/** rwk,
}
```

Move the updated file:

```
$ sudo mv usr.sbin.xtrabackup /etc/apparmor.d/
```

Install the profile with the following command:

```
$ sudo apparmor_parser -r -T -W /etc/apparmor.d/usr.sbin.xtrabackup
```

Run the backup as usual.

No additional AppArmor-related actions are required to restore a backup.

Percona LLC and/or its affiliates © 2023 — Cookie Consent

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-10-12

7.3.6 Uninstall Percona XtraBackup 8.2 on Debian and Ubuntu

To completely uninstall Percona XtraBackup, remove all the installed packages:

```
$ sudo apt remove percona-xtrabackup-82
```

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-10-17

7.4 Use YUM

7.4.1 Use a YUM repository to install Percona XtraBackup

Ready-to-use packages are available from the Percona XtraBackup software repositories and the download page. The Percona yum repository supports popular `RPM`-based operating systems, including the `Amazon Linux AMI`.

The easiest way to install the Percona Yum repository is to install an `RPM` that configures yum and installs the Percona GPG key.

Specific information on the supported platforms, products, and versions is described in Percona Software and Platform Lifecycle.

Install Percona XtraBackup from Percona `yum` repository

To install Percona XtraBackup from Percona `yum` repository, do the following steps:

Percona LLC and/or its affiliates © 2023 — Cookie Consent

0. Install the Percona yum repository by running the following command as the `root` user or with **sudo**:

```
$ sudo yum install \
https://repo.percona.com/yum/percona-release-latest.\
noarch.rpm
```

0. Enable the repository:

```
$ sudo percona-release enable-only tools release
```

If *Percona XtraBackup* is intended to be used in combination with the upstream MySQL Server, you only need to enable the `tools repository:

```
$ sudo percona-release enable-only tools
```

0. Install *Percona XtraBackup* by running:

```
$ sudo yum install percona-xtrabackup-82
```

Warning

Make sure that you have the `libev` package installed before installing *Percona XtraBackup* on CentOS 6. For this operating system, the `libev` package is available from the EPEL repositories.

0. To decompress backups made using `LZ4` or `ZSTD` compression algorithm, install the corresponding package:

Install the `lz4` package **Install the `zstd` package**

```
$ sudo yum install lz4

$ sudo yum install zstd
```

See also

To install Percona XtraBackup using downloaded rpm packages, see Install with package manager.

To uninstall Percona XtraBackup, see Uninstall Percona XtraBackup

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

<a href="https://www.percona.com/about">Percona LLC and/or its affiliates © 2023 — Cookie Consent

Last update: 2023-10-23

7.4.2 Files in the RPM package built for Percona XtraBackup 8.2

The following tables show what data each `RPM` package contains:

Package	Contains
percona-xtrabackup-82	The latest Percona XtraBackup GA binaries and associated files
percona-xtrabackup-82-debuginfo	The debug symbols for binaries in percona-xtrabackup-82
percona-xtrabackup-test-82	The test suite for Percona XtraBackup
percona-xtrabackup	The older version of the Percona XtraBackup

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-10-17

7.4.3 Install Percona XtraBackup 8.2 using downloaded RPM packages

Download `RPM` packages of the desired series for your architecture from the download page.

The following example downloads *Percona XtraBackup* 8.2.0-1 release package for *CentOS* 7:

```
$ wget https://www.percona.com/downloads/XtraBackup/Percona-XtraBackup-8.2.0-1/
binary/redhat/7/x86_64/percona-xtrabackup-82-8.2.0-1.el7.x86_64.rpm
```

Install Percona XtraBackup by running:

```
$ yum localinstall percona-xtrabackup-82-8.2.0-1.el7.x86_64.rpm
```

When installing packages manually like this, you'll need to make sure to resolve all the dependencies and install missing packages yourself.

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Percona LLC and/or its affiliates © 2023 — Cookie Consent

Last update: 2023-10-17

7.4.4 Work with SELinux

Percona XtraBackup is installed as an unconfined process running in an undefined domain. SELinux allows unconfined processes almost all access and the processes only use Discretionary Access Control (DAC) rules.

You find the current state of the Percona XtraBackup file with the following command:

```
$ ls -Z /usr/bin | grep xtrabackup
```

Expected output

```
-rwxr-xr-x. root root   system_u:object_r:bin_t:s0       xtrabackup
```

The SELinux context is the following:

- user (root)
- role (object_r)
- type (bin_t)
- level (s0)

The unconfined domain supports the network-facing services, which are protected by SELinux. These domains are not exposed. In this configuration, SELinux protects against remote intrusions but local intrusions, which require local access, are not confined.

Percona XtraBackup works locally. The service is not network-facing and cannot be exploited externally. The service interacts only with the local user, who provides the parameters. *Percona XtraBackup* requires access to the `target-dir` location.

Confine XtraBackup

You can modify your security configuration to confine *Percona XtraBackup*. The first question is where to store the backup files. The service requires read and write access to the selected location.

You can use either of the following methods:

- Allow *Percona XtraBackup* to write to any location. The user provides any path to the `target-dir` parameter.
- Allow *Percona XtraBackup* to write to a specific location, such as /backups or the user's home directory.

The first option opens the entire system to read and write. Select the second option to harden your security.

Percona LLC and/or its affiliates © 2023 — Cookie Consent

Install SELinux tools

To work with policies, you must install the SELinux tools. To find which package provides the `semanage` command and install the package. The following is an example on CentOS 7.

```
$ yum provides *bin/semanage
```

The result should list the packages.

Expected output

```
...
policycoreutils-python-2.5-34.el7.x86_64 : SELinux policy core python utilities
...
```

To install missing packages, run the following:

```
$ sudo yum install -y policycoreutils-python
```

The following is an example on CentOS 8:

```
$ yum provides *bin/semanage
```

The result should list the missing packages.

Expected output

```
...
policycoreutils-python-utils-2.8-16.1.el8.noarch : SELinux policy core python utilities
```

Run the following to install the missing packages:

```
$ sudo yum install -y policycoreutils-python-utils
```

Create a policy

Use a modular approach to create an SELinux policy. Create a policy module to manage XtraBackup. You must create a `.te` file for type enforcement, and an optional `.fc` file for the file contexts.

Use `$ ps -efZ | grep xtrabackup` to verify the service is not confined by SELinux.

Create the `xtrabackup.fc` file and add content. This file defines the security contexts.

```
/usr/bin/xtrabackup    -- gen_context(system_u:object_r:xtrabackup_exec_t,s0)
/usr/bin/xbcrypt    -- gen_context(system_u:object_r:xtrabackup_exec_t,s0)
/usr/bin/xbstream    -- gen_context(system_u:object_r:xtrabackup_exec_t,s0)
/usr/bin/xbcloud    -- gen_context(system_u:object_r:xtrabackup_exec_t,s0)
/backups(/.*)?       system_u:object_r:xtrabackup_data_t:s0
```


<a href="https://www.percona.com/about">Percona LLC and/or its affiliates © 2023 — Cookie Consent

Note

If you are using the `/backups` directory you must have the last line. If you are storing the backups in the user's home directory, you can omit this line.

Download the `xtrabackup.te` file from the following location:

https://github.com/percona/percona-xtrabackup/tree/trunk/packaging/percona/selinx

Note

In the file, the sections in bold should be modified for your system. The fc file can also be downloaded from the same location.

Compile the policy module:

```
$ make -f /usr/share/selinux/devel/Makefile xtrabackup.pp
```

Install the module:

```
$ semodule -i xtrabackup.pp
```

Tag the PXB binaries with the proper SELinux tags, such as `xtrabackup_exec_t`.

```
$ restorecon -v /usr/bin/*
```

If you store your backups at `/backups`, restore the tag in that location:

```
$ restorecon -v /backups
```

Note

Remember to add the standard Linux DAC permissions for this directory.

Perform the backup in the standard way.

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

<a href="https://www.percona.com/about">Percona LLC and/or its affiliates © 2023 — Cookie Consent

7.4.5 Uninstall Percona XtraBackup 8.2 on Red Hat Enterprise Linux and CentOS

To completely uninstall Percona XtraBackup, remove all the installed packages:

```
$ yum remove percona-xtrabackup
```

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-10-12

7.5 Use binary tarballs

7.5.1 Install from a binary tarball

Binary tarballs contain precompiled executable files, libraries, and other dependencies and are compressed `tar` archives. Extract the binary tarballs to any path.

Download the binary tarballs from the `Linux - Generic` section on Percona Product Downloads.

The following example downloads the a tarball:

```
$ wget https://downloads.percona.com/downloads/Percona-XtraBackup-8.2/Percona-
XtraBackup-8.2.0-1/binary/tarball/percona-xtrabackup-8.2.0-1-Linux-
x86_64.glibc2.17.tar.gz
```

The output displays the following information:

Expected output

```
--2023-10-20 05:56:30--  https://downloads.percona.com/downloads/Percona-XtraBackup-8.2/Percona-
XtraBackup-8.2.0-1/binary/tarball/percona-xtrabackup-8.2.0-1-Linux-x86_64.glibc2.17.tar.gz
Resolving downloads.percona.com (downloads.percona.com)... 2604:2dc0:200:69f::2, 147.135.54.159
Connecting to downloads.percona.com (downloads.percona.com)|2604:2dc0:200:69f::2|:443...
connected.
HTTP request sent, awaiting response... 200 OK
Length: 470358258 (449M) [application/gzip]
Saving to: ‘percona-xtrabackup-8.2.0-1-Linux-x86_64.glibc2.17.tar.gz’

percona-xtrabackup   0%[                    ]   3.17M  1.54MB/s
```


<a href="https://www.percona.com/about">Percona LLC and/or its affiliates © 2023 — Cookie Consent

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-10-17

7.5.2 Binary tarball file names available

Download the binary tarballs from Percona Product Downloads.

The following table lists the tarball types available in `Linux - Generic`. Select the *Percona XtraBackup* 8.2 version, the software or the operating system, and the type of tarball for your installation. Binary tarballs support all distributions.

After you have downloaded the binary tarballs, extract the tarball in the file location of your choice.

Type	Name	Operating systems	Description
Full	percona-xtrabackup-8.2.0-1-Linux.x86_64.glibc2.12.tar.gz	Built for CentOS 6	Contains binary files, libraries, test files, and debug symbols
Minimal	percona-xtrabackup-8.2.0-1-Linux.x86_64.glibc2.12-minimal.tar.gz	Built for CentOS 6	Contains binary files, and libraries but does not include test files, or debug symbols
Full	percona-xtrabackup-8.2.0-1-Linux.x86_64.glibc2.17.tar.gz	Compatible with any operating system except for CentOS 6	Contains binary files, libraries, test files, and debug symbols
Minimal	percona-xtrabackup-8.2.0-1-Linux.x86_64.glibc2.17-minimal.tar.gz	Compatible with any operating system except for CentOS 6	Contains binary files, and libraries but does not include test files, or debug symbols

Select a different software, such as Ubuntu 20.04 (Focal Fossa), for a tarball for that operating system. You can download the packages together or separately.

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

<a href="https://www.percona.com/about">Percona LLC and/or its affiliates © 2023 — Cookie Consent

Community Forum Get a Percona Expert

Last update: 2023-10-17

7.6 Compile from source

7.6.1 Compile and install from source

The following instructions install Percona XtraBackup 8.2.

1. Install Percona XtraBackup from the Git Source Tree

Percona uses the Github revision control system for development. To build the latest Percona Server for MySQL from the source tree, you will need `git` installed on your system.

You can now fetch the latest Percona XtraBackup 8.2 sources:

```
$ git clone https://github.com/percona/percona-xtrabackup.git
$ cd percona-xtrabackup
$ git checkout trunk
$ git submodule update --init --recursive
```

2. Installation prerequisites

The following packages and tools must be installed to compile *Percona XtraBackup* from source. These might vary from system to system.

Important

To build **Percona XtraBackup 8.2 from source, you must use `cmake` version 3. To check which version is currently installed, run `cmake --version` at a command prompt. If the version is not `3`, install `cmake3`.

Percona LLC and/or its affiliates © 2023 — Cookie Consent

This `cmake` version may be available in your distribution as a separate package `cmake3`. For more information, see cmake.org.

Debian or Ubuntu using `apt` **CentOS or Red Hat using `yum`**

```
sudo apt install bison pkg-config cmake devscripts debconf \
debhelper automake bison ca-certificates libprocps-dev \
libcurl4-openssl-dev cmake debhelper libaio-dev \
libncurses-dev libssl-dev libtool libz-dev libgcrypt-dev libev-dev libprocps-dev \
lsb-release build-essential rsync libdbd-mysql-perl \
libnuma1 socat librtmp-dev libtinfo5 vim-common \
liblz4-tool liblz4-1 liblz4-dev zstd python-docutils
```

To install the man pages, install the python3-sphinx package first:

```
$ sudo apt install python3-sphinx
```

Percona Xtrabackup requires GCC version 5.3 or higher. If the version of GCC installed on your system is lower then you may need to install and enable the Developer Toolset on `RPM`-based distributions to make sure that you use the latest GCC compiler and development tools. Then, install `cmake` and other dependencies:

```
$ sudo yum install cmake openssl-devel libaio libaio-devel automake autoconf \
bison libtool ncurses-devel libgcrypt-devel libev-devel libcurl-devel zlib-devel \
zstd vim-common procps-ng-devel
```

To install the man pages, install the python3-sphinx package first:

```
$ sudo yum install python3-sphinx
```

3. Generate the build pipeline

At this step, you have `cmake` run the commands in the `CMakeList.txt` file to generate the build pipeline, i.e., a native build environment that will be used to compile the source code).

0. Change to the directory where you cloned the Percona XtraBackup repository

   ```
   $ cd percona-xtrabackup
   ```

0. Create a directory to store the compiled files and then change to that directory:

   ```
   $ mkdir build
   $ cd build
   ```

0. Run cmake or cmake3. In either case, the options you need to use are the same.

<a href="https://www.percona.com/about">Percona LLC and/or its affiliates © 2023 — Cookie Consent

Note

You can build *Percona XtraBackup* with man pages but this requires `python-sphinx` package which isn't available from that main repositories for every distribution. If you installed the `python-sphinx` package you need to remove the `-DWITH_MAN_PAGES=OFF` from previous command.

```
$ cmake -DWITH_BOOST=PATH-TO-BOOST-LIBRARY -DDOWNLOAD_BOOST=ON \
-DBUILD_CONFIG=xtrabackup_release -DWITH_MAN_PAGES=OFF -B ..
```

PARAMETER INFORMATION

Parameter	Description
`-DWITH_BOOST`	For the `-DWITH_BOOST` parameter, specify the name of a directory to download the boost library to. This directory is created automatically in your current directory.
`-DWITH_MAN_PAGES`	To build **Percona XtraBackup** man pages, use `ON` or remove this parameter from the command line (it is `ON` by default). To install the man pages, install the python3-sphinx package first.
`-B` (–build)	**Percona XtraBackup** is configured to forbid generating the build pipeline for make in the same directory where you store your sources. The `-B` parameter refers to the directory that contains the source code. In this example, we use the relative path to the parent directory (..).

Important

CMake Error at CMakeLists.txt:367 (MESSAGE): Please do not build in-source. Out-of source builds are highly recommended: you can have multiple builds for the same source, and there is an easy way to do cleanup, simply remove the build directory (note that 'make clean' or 'make distclean' does `not` work)

You `can` force in-source build by invoking cmake with `-DFORCE_INSOURCE_BUILD=1`.

4. Compile the source code

To compile the source code in your `build` directory, use the `make` command.

0. Change to the `build` directory (created at Step 2: Generating the build pipeline).
0. Run the `make` command. This command may take a long time to complete.

   ```
   $ make
   ```

   To use all CPU threads and make compilation faster please use:

   ```
   $ make -j$(nproc --all)
   ```

5. Install on the target system

The following command installs all *Percona XtraBackup* binaries *xtrabackup* and tests to default location on the target system: `/usr/local/xtrabackup`.

<a href="https://www.percona.com/about">Percona LLC and/or its affiliates © 2023 — Cookie Consent

Run `make install` to install *Percona XtraBackup* to the default location.

```
$ sudo make install
```

INSTALL TO A NON-DEFAULT LOCATION

You may use the `DESTDIR` parameter with `make install` to install *Percona XtraBackup* to another location. Make sure that the effective user is able to write to the destination you choose.

```
$ sudo make DESTDIR=<DIR_NAME> install
```

In fact, the destination directory is determined by the installation layout (`-DINSTALL_LAYOUT`) that `cmake` applies (see Step 2: Generating the build pipeline). In addition to the installation directory, this parameter controls a number of other destinations that you can adjust for your system.

By default, this parameter is set to `STANDALONE`, which implies the installation directory to be `/usr/local/xtrabackup`.

See also

MySQL Documentation: -DINSTALL_LAYOUT

6. Run Percona XtraBackup

After *Percona XtraBackup* is installed on your system, you may run it by using the full path to the `xtrabackup` command:

```
$ /usr/local/xtrabackup/bin/xtrabackup
```

Update your PATH environment variable if you would like to use the command on the command line directly.

```
$# Setting $PATH on the command line
$ PATH=$PATH:/usr/local/xtrabackup/bin/xtrabackup

$# Run xtrabackup directly
$ xtrabackup
```

Alternatively, you may consider placing a soft link (using `ln -s`) to one of the locations listed in your `PATH` environment variable.

To view the documentation with `man`, update the `MANPATH` variable.

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Percona and/or its affiliates © 2023 — Cookie Consent

Last update: 2023-10-12

7.7 Docker

7.7.1 Run a Docker container

Docker allows you to run applications in a lightweight unit called a container.

You can run Percona XtraBackup in a Docker container without installing the product. All required libraries are available in the container. Being a lightweight execution environment, Docker containers enable creating configurations where each program runs in a separate container. You may run Percona Server for MySQL in one container and Percona XtraBackup in another. Docker images offer a range of options.

Create a Docker container based on a Docker image. Docker images for Percona XtraBackup are hosted publicly on Docker Hub.

```
$ sudo docker create ... percona/percona-xtrabackup --name xtrabackup ...
```

SCOPE OF THIS SECTION

This section demonstrates how to back up data on a Percona Server for MySQL running in another Dockers container.

1. Install Docker

Your operating system may already provide a package for docker. However, the versions of Docker provided by your operating system are likely to be outdated.

Use the installation instructions for your operating system available from the Docker site to set up the latest version of docker.

2. Connect to a Percona Server for MySQL container

Percona XtraBackup works in combination with a database server. When running a Docker container for Percona XtraBackup, you can make backups for a database server either installed on the host machine or running in a separate Docker container.

To set up a database server on a host machine or in Docker container, follow the documentation of the supported product that you intend to use with Percona XtraBackup.

See also

Docker volumes as container persistent data storage

More information about containers

```
$ sudo docker run -d --name percona-server-mysql \
-e MYSQL_ROOT_PASSWORD=root percona/percona-server:8.2
```


Percona and/or its affiliates © 2023 — Cookie Consent

As soon as Percona Server for MySQL runs, add some data to it. Now, you are ready to make backups with Percona XtraBackup.

> **Important**
>
> When running Percona XtraBackup from a container and connecting to a MySQLserver container, we recommend using the –user root option in the Docker command.

3. Create a Docker container from Percona XtraBackup image

You can create a Docker container based on Percona XtraBackup image with either `docker create` or the `docker run` command. `docker create` creates a Docker container and makes it available for starting later.

Docker downloads the Percona XtraBackup image from the Docker Hub. If it is not the first time you use the selected image, Docker uses the image available locally.

```
$ sudo docker create --name percona-xtrabackup --volumes-from percona-server-mysql \
percona/percona-xtrabackup  \
xtrabackup --backup --datadir=/var/lib/mysql/ --target-dir=/backup \
--user=root --password=mysql
```

With parameter name you give a meaningful name to your new Docker container so that you could easily locate it among your other containers.

The `volumes-from` flag refers to Percona Server for MySQL and indicates that you intend to use the same data as the Percona Server for MySQL container.

Run the container with exactly the same parameters that were used when the container was created:

```
$ sudo docker start -ai percona-xtrabackup
```

This command starts the percona-xtrabackup container, attaches to its input/output streams, and opens an interactive shell.

The `docker run` is a shortcut command that creates a Docker container and then immediately runs it.

```
$ sudo docker run --name percona-xtrabackup --volumes-from percona-server-mysql \
percona/percona-xtrabackup
xtrabackup --backup --data-dir=/var/lib/mysql --target-dir=/backup --user=root --
password=mysql
```

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Percona LLC and/or its affiliates © 2023 — Cookie Consent

Last update: 2023-10-17

Percona LLC and/or its affiliates © 2023 — Cookie Consent

8. How to

8.1 Take a full backup

8.1.1 Create a full backup

To create a backup, run xtrabackup with the `--backup` option. You also must specify the `--target-dir` option, which is where the backup is stored. If the InnoDB data or log files are not stored in the same directory, you must specify their location. If the target directory does not exist, xtrabackup creates it. If the directory does exist and is empty, xtrabackup succeeds.

xtrabackup does not overwrite existing files. it will fail with operating system error 17, `file exists`.

The following command starts the process:

```
$ xtrabackup --backup --target-dir=/data/backups/
```

This operation stores the backup at `/data/backups/`. If you specify a relative path, the target directory is relative to the current directory.

During the backup process, the output shows the copied data files, and a log file thread that scans and copies from the log files.

The following is an example of the output:

Expected output

```
160906 10:19:17 Finished backing up non-InnoDB tables and files
160906 10:19:17 Executing FLUSH NO_WRITE_TO_BINLOG ENGINE LOGS...
xtrabackup: The latest check point (for incremental): '62988944'
xtrabackup: Stopping log copying thread.
.160906 10:19:18 >> log scanned up to (137343534)
160906 10:19:18 Executing UNLOCK TABLES
160906 10:19:18 All tables unlocked
160906 10:19:18 Backup created in directory '/data/backups/'
160906 10:19:18 [00] Writing backup-my.cnf
160906 10:19:18 [00]        ...done
160906 10:19:18 [00] Writing xtrabackup_info
160906 10:19:18 [00]        ...done
xtrabackup: Transaction log of lsn (26970807) to (137343534) was copied.
160906 10:19:18 completed OK!
```

The process ends with the following statement; the value of the `<LSN>` depends on your system:

```
$ xtrabackup: Transaction log of lsn (<LSN>) to (<LSN>) was copied.
```


Percona and/or its affiliates © 2023 — Cookie Consent

Note

Log copying thread checks the transactional log every second to see if there were any new log records written that need to be copied, but there is a chance that the log copying thread might not be able to keep up with the amount of writes that go to the transactional logs, and will hit an error when the log records are overwritten before they could be read.

After the backup is finished, the target directory will contain files such as the following, assuming you have a single InnoDB table `test.tbl1` and you are using MySQL's innodb_file_per_table option:

```
$ ls -lh /data/backups/
```

The result should look like this:

Expected output

```
total 182M
drwx------  7 root root 4.0K Sep  6 10:19 .
drwxrwxrwt 11 root root 4.0K Sep  6 11:05 ..
-rw-r-----  1 root root  387 Sep  6 10:19 backup-my.cnf
-rw-r-----  1 root root  76M Sep  6 10:19 ibdata1
drwx------  2 root root 4.0K Sep  6 10:19 mysql
drwx------  2 root root 4.0K Sep  6 10:19 performance_schema
drwx------  2 root root 4.0K Sep  6 10:19 sbtest
drwx------  2 root root 4.0K Sep  6 10:19 test
drwx------  2 root root 4.0K Sep  6 10:19 world2
-rw-r-----  1 root root  116 Sep  6 10:19 xtrabackup_checkpoints
-rw-r-----  1 root root  433 Sep  6 10:19 xtrabackup_info
-rw-r-----  1 root root 106M Sep  6 10:19 xtrabackup_logfile
```

The backup can take a long time, depending on how large the database is. It is safe to cancel at any time, because xtrabackup does not modify the database.

Next step

Prepare the backup →

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-10-04

Percona LLC and/or its affiliates © 2023 — Cookie Consent

8.1.2 Prepare a full backup

After creating a backup with the `--backup` option, you need to prepare the backup and then restore it. Data files are not point-in-time consistent until they are prepared, because they were copied at different times as the program ran, and they might have been changed while this was happening.

If you try to start InnoDB with these data files, it will detect corruption and stop working to avoid running on damaged data. The `--prepare` step makes the files perfectly consistent at a single instant in time, so you can run InnoDB on them.

You can run the prepare operation on any machine; it does not need to be on the originating server or the server to which you intend to restore. You can copy the backup to a utility server and prepare it there.

Note that Percona XtraBackup 8.2 can only prepare backups of MySQL 8.2 and Percona Server for MySQL 8.2 databases. Releases prior to 8.2 are not supported.

During the prepare operation, xtrabackup boots up a kind of modified embedded InnoDB (the libraries xtrabackup was linked against). The modifications are necessary to disable InnoDB standard safety checks, such as complaining about the log file not being the right size. This warning is not appropriate for working with backups. These modifications are only for the xtrabackup binary; you do not need a modified InnoDB to use xtrabackup for your backups.

The prepare step uses this “embedded InnoDB” to perform crash recovery on the copied data files, using the copied log file. The `prepare` step is very simple to use: you simply run xtrabackup with the `--prepare` option and tell it which directory to prepare, for example, to prepare the previously taken backup run:

```
$ xtrabackup --prepare --target-dir=/data/backups/
```

When this finishes, you should see an `InnoDB shutdown` with a message such as the following, where again the value of LSN will depend on your system:

Expected output

```
InnoDB: Shutdown completed; log sequence number 137345046
160906 11:21:01 completed OK!
```

All following prepares will not change the already prepared data files, you’ll see that output says:

Expected output

```
xtrabackup: This target seems to be already prepared.
xtrabackup: notice: xtrabackup_logfile was already used to '--prepare'.
```

It is not recommended to interrupt xtrabackup process while preparing backup because it may cause data files corruption and backup will become unusable. Backup validity is not guaranteed if prepare process was interrupted.

Percona and/or its affiliates © 2023 — Cookie Consent

Note

If you intend the backup to be the basis for further incremental backups, you should use the `--apply-log-only` option when preparing the backup, or you will not be able to apply incremental backups to it. See the documentation on preparing incremental backups for more details.

Next step

Restore the backup →

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-10-12

8.2 Take an incremental backup

8.2.1 Create an incremental backup

xtrabackup supports incremental backups, which means that they can copy only all the data that has changed since the last backup.

Note

Incremental backups on the MyRocks storage engine do not determine if an earlier full backup or incremental backup contains the same files. Percona XtraBackup copies all the MyRocks files each time it takes a backup.

You can perform many incremental backups between each full backup, so you can set up a backup process such as a full backup once a week and an incremental backup every day, or full backups every day and incremental backups every hour.

Incremental backups work because each InnoDB page contains a log sequence number, or LSN. The LSN is the system version number for the entire database. Each page's LSN shows how recently it was changed.

An incremental backup copies each page which LSN is newer than the previous incremental or full backup's LSN. An algorithm finds the pages that match the criteria. The algorithm reads the data pages and checks the page LSN.

Percona LLC and/or its affiliates © 2023 — Cookie Consent

Create an incremental backup

To make an incremental backup, begin with a full backup as usual. The xtrabackup binary writes a file called `xtrabackup_checkpoints` into the backup's target directory. This file contains a line showing the `to_lsn`, which is the database's LSN at the end of the backup. Create the full backup with a following command:

```
$ xtrabackup --backup --target-dir=/data/backups/base
```

If you look at the `xtrabackup_checkpoints` file, you should see similar content depending on your LSN nuber:

Expected output

```
backup_type = full-backuped
from_lsn = 0
to_lsn = 1626007
last_lsn = 1626007
compact = 0
recover_binlog_info = 1
```

Now that you have a full backup, you can make an incremental backup based on it. Use the following command:

```
$ xtrabackup --backup --target-dir=/data/backups/inc1 \
--incremental-basedir=/data/backups/base
```

The `/data/backups/inc1/` directory should now contain delta files, such as `ibdata1.delta` and `test/table1.ibd.delta`. These represent the changes since the `LSN 1626007`. If you examine the `xtrabackup_checkpoints` file in this directory, you should see similar content to the following:

Expected output

```
backup_type = incremental
from_lsn = 1626007
to_lsn = 4124244
last_lsn = 4124244
compact = 0
recover_binlog_info = 1
```

`from_lsn` is the starting LSN of the backup and for incremental it has to be the same as `to_lsn` (if it is the last checkpoint) of the previous/base backup.

It's now possible to use this directory as the base for yet another incremental backup:

```
$ xtrabackup --backup --target-dir=/data/backups/inc2 \
--incremental-basedir=/data/backups/inc1
```

This folder also contains the `xtrabackup_checkpoints`:

Percona and/or its affiliates © 2023 — Cookie Consent

Expected output

```
backup_type = incremental
from_lsn = 4124244
to_lsn = 6938371
last_lsn = 7110572
compact = 0
recover_binlog_info = 1
```

Note

In this case you can see that there is a difference between the `to_lsn` (last checkpoint LSN) and `last_lsn` (last copied LSN), this means that there was some traffic on the server during the backup process.

Next step

Prepare the backup →

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-10-04

8.2.2 Prepare an incremental backup

The `--prepare` step for incremental backups is not the same as for full backups. In full backups, two types of operations are performed to make the database consistent: committed transactions are replayed from the log file against the data files, and uncommitted transactions are rolled back. You must skip the rollback of uncommitted transactions when preparing an incremental backup, because transactions that were uncommitted at the time of your backup may be in progress, and it's likely that they will be committed in the next incremental backup. You should use the `--apply-log-only` option to prevent the rollback phase.

Warning

If you do not use the `--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.

Beginning with the full backup you created, you can prepare it, and then apply the incremental differences to it. Recall that you have the following backups:

Percona and/or its affiliates © 2023 — Cookie Consent

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

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

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

The output should end with text similar to the following:

Expected output

```
InnoDB: Shutdown completed; log sequence number 1626007
161011 12:41:04 completed OK!
```

The log sequence number should match the `to_lsn` of the base backup, which you saw previously.

Warning

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.

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
```

This applies the delta files to the files in `/data/backups/base`, which rolls them forward in time to the time of the incremental backup. It then applies the redo log as usual to the result. The final data is in `/data/backups/base`, not in the incremental directory. You should see an output similar to:

Expected output

```
incremental backup from 1626007 is enabled.
xtrabackup: cd to /data/backups/base
xtrabackup: This target seems to be already prepared with --apply-log-only.
xtrabackup: xtrabackup_logfile detected: size=2097152, start_lsn=(4124244)
...
xtrabackup: page size for /tmp/backups/inc1/ibdata1.delta is 16384 bytes
Applying /tmp/backups/inc1/ibdata1.delta to ./ibdata1...
...
161011 12:45:56 completed OK!
```

Again, the LSN should match what you saw from your earlier inspection of the first incremental backup. If you restore the files from `/data/backups/base`, you should see the state of the database as of the first incremental backup.

<a href="https://www.percona.com/about">Percona LLC and/or its affiliates © 2023 — Cookie Consent

Warning

Percona XtraBackup does not support using the same incremental backup directory to prepare two copies of backup. Do not run `--prepare` with the same incremental backup directory (the value of –incremental-dir) more than once.

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
```

Note

`--apply-log-only` should be used when merging the incremental backups except the last one. That's why the previous line does not contain the `--apply-log-only` option. Even if the `--apply-log-only` was used on the last step, backup would still be consistent but in that case server would perform the rollback phase.

Next step

Restore the backup →

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-10-04

8.2.3 Take an incremental backup using page tracking

To create an incremental backup with page tracking, Percona XtraBackup uses the MySQL `mysqlbackup` component. This component provides a list of pages modified since the last backup, and Percona XtraBackup copies only those pages. This operation removes the need to scan the pages in the database. If the majority of pages have not been modified, the page tracking feature can improve the speed of incremental backups.

Install the component

To start using the page tracking functionality, do the following:

0. Install the `mysqlbackup` component and enable it on the server:

Percona LLC and/or its affiliates © 2023 — Cookie Consent

```
$ INSTALL COMPONENT "file://component_mysqlbackup";
```

0. Check whether the `mysqlbackup` component is installed successfully:

```
$ SELECT COUNT(1) FROM mysql.component WHERE component_urn='file://
component_mysqlbackup';
```

Use page tracking

You can enable the page tracking functionality for the full and incremental backups with the `--page-tracking` option.

The option has the following benefits:

- Resets page tracking to the start of the backup. This reset allows the next incremental backup to use page tracking.
- Allows the use of page tracking for an incremental backup if the page tracking data is available from the backup's start checkpoint LSN.

Percona XtraBackup processes a list of all the tracked pages in memory. If Percona XtraBackup does not have enough available memory to process this list, the process throws an error and exits. For example, if an incremental backup uses 200GB, Percona XtraBackup can use an additional 100MB of memory to process and store the page tracking data.

The examples of creating full and incremental backups using the `--page-tracking` option:

Full backup **Incremental backup**

```
$ xtrabackup --backup --target-dir=$FULL_BACK --page-tracking

$ xtrabackup --backup --target-dir=$INC_BACKUP
--incremental-basedir=$FULL_BACKUP --page-tracking
```

After enabling the functionality, the next incremental backup finds changed pages using page tracking.

The first full backup using page tracking, Percona XtraBackup may have a delay. The following is an example of the message:

Expected output

```
xtrabackup: pagetracking: Sleeping for 1 second, waiting for checkpoint lsn 17852922 /
to reach to page tracking start lsn 21353759
```

Enable page tracking before creating the first backup to avoid this delay. This method ensures that the page tracking log sequence number (LSN) is higher than the checkpoint LSN of the server.

Percona LLC and/or its affiliates © 2023 — Cookie Consent

Start page tracking manually

After the mysqlbackup component is loaded and active on the server, you can start page tracking manually with the following option:

```
$ SELECT mysqlbackup_page_track_set(true);
```

Check the LSN value

Check the LSN value starting from which changed pages are tracked with the following option:

```
$ SELECT mysqlbackup_page_track_get_start_lsn();
```

Stop page tracking

To stop page tracking, use the following command:

```
$ SELECT mysqlbackup_page_track_set(false);
```

Purge page tracking data

When you start page tracking, it creates a file under the server's datadir to collect data about changed pages. This file grows until you stop the page tracking. If you stop the server and then restart it, page tracking creates a new file but also keeps the old one. The old file continues to grow until you stop the page tracking explicitly.

If you purge the page tracking data, you should create a full backup afterward. To purge the page tracking data, do the following steps:

```
$ SELECT mysqlbackup_page_track_set(false);
$ SELECT mysqlbackup_page_track_purge_up_to(9223372036854775807);
/* Specify the LSN up to which you want to purge page tracking data. /
9223372036854775807 is the highest possible LSN which purges all page tracking
files.*/
$ SELECT mysqlbackup_page_track_set(true);
```

Known issue

If the index is built in place using an exclusive algorithm and then is added to a table after the last LSN checkpoint, you may generate a bad incremental backup using page tracking. For more details see PS-8032.

Uninstall the mysqlbackup component

To uninstall the mysqlbackup component, use the following statement:

```
$ UNINSTALL COMPONENT "file://component_mysqlbackup"
```


Percona LLC and/or its affiliates © 2023 — Cookie Consent

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-10-17

8.3 Take a compressed backup

8.3.1 Create a compressed backup

Percona XtraBackup supports compressed backups. To make a compressed backup, use the `--compress` option along with the `--backup` and `--target-dir` options. A local or streaming backup can be compressed or decompressed with xbstream.

By default, the `--compress` option uses the `zstandard` tool that you can install with the `percona-release` package configuration tool as follows:

```
$ sudo percona-release enable tools
$ sudo apt update
$ sudo apt install zstandard
```

Note

Enable the repository: `percona-release enable-only tools release`.

If Percona XtraBackup is intended to be used in combination with the upstream MySQL Server, you only need to enable the `tools` repository: `percona-release enable-only tools`.

Percona XtraBackup supports the following compression algorithms:

Zstandard (ZSTD)

The Zstandard (ZSTD) is a fast lossless compression algorithm that targets real-time compression scenarios and better compression ratios. `ZSTD` is the default compression algorithm for the `--compress` option.

To compress files using the `ZSTD` compression algorithm, use the `--compress` option:

```
$ xtrabackup --backup --compress --target-dir=/data/backup
```

The resulting files have the `\*.zst` format.

You can specify `ZSTD` compression level with the `--compress-zstd-level(=#)` option. The default value is `1`.

<a href="https://www.percona.com/about">Percona LLC and/or its affiliates © 2023 — Cookie Consent

```
$ xtrabackup –backup –compress –compress-zstd-level=1 –target-dir=/data/backup
```

lz4

To compress files using the `lz4` compression algorithm, set the `--compress` option to `lz4`:

```
$ xtrabackup --backup --compress=lz4 --target-dir=/data/backup
```

The resulting files have the `\*.lz4` format.

If you want to speed up the compression you can use the parallel compression, which can be enabled with `--compress-threads` option. Following example will use four threads for compression:

```
$ xtrabackup --backup --compress --compress-threads=4 \
--target-dir=/data/compressed/
```

Expected output

```
...
170223 13:00:38 [01] Compressing ./test/sbtest1.frm to /tmp/compressed/test/sbtest1.frm.qp
170223 13:00:38 [01]        ...done
170223 13:00:38 [01] Compressing ./test/sbtest2.frm to /tmp/compressed/test/sbtest2.frm.qp
170223 13:00:38 [01]        ...done
...
170223 13:00:39 [00] Compressing xtrabackup_info
170223 13:00:39 [00]        ...done
xtrabackup: Transaction log of lsn (9291934) to (9291934) was copied.
170223 13:00:39 completed OK!
```

Next step

Prepare the backup →

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-10-04

Percona LLC and/or its affiliates © 2023 — Cookie Consent

8.3.2 Prepare the backup

Before you can prepare the backup you'll need to uncompress all the files. Percona XtraBackup has implemented `--decompress` option that can be used to decompress the backup.

```
$ xtrabackup --decompress --target-dir=/data/compressed/
```

Note

`--parallel` can be used with `--decompress` option to decompress multiple files simultaneously.

Percona XtraBackup does not automatically remove the compressed files. In order to clean up the backup directory you should use `--remove-original` option. Even if they're not removed these files will not be copied/moved over to the datadir if `--copy-back` or `--move-back` are used.

When the files are uncompressed you can prepare the backup with the `--prepare` option:

```
$ xtrabackup --prepare --target-dir=/data/compressed/
```

Confirmation message

```
InnoDB: Starting shutdown...
InnoDB: Shutdown completed; log sequence number 9293846
170223 13:39:31 completed OK!
```

Now the files in `/data/compressed/` are ready to be used by the server.

Next step

Restore the backup →

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-10-04

Percona LLC and/or its affiliates © 2023 — Cookie Consent

8.4 Take a partial backup

8.4.1 Create a partial backup

xtrabackup supports taking partial backups when the `innodb_file_per_table` option is enabled.

> **Warning**
>
> Do not copy back the prepared backup.
>
> Restoring partial backups should be done by importing the tables. We do not by using the `–copy-back` option. This operation may lead to database inconsistencies.
>
> We do not recommend running incremental backups after taking a partial backup.

The xtrabackup binary fails if you delete any of the matched or listed tables during the backup.

There are multiple ways of specifying which part of the whole data is backed up:

- Use the `--tables` option to list the table names
- Use the `--tables-file` option to list the tables in a file
- Use the `--databases` option to list the databases
- Use the `--databases-file` option to list the databases

Percona LLC and/or its affiliates © 2023 — Cookie Consent

The following examples assume a database named `test` that contains tables named `t1` and `t2`.

`--tables` option `--tables-file` option `--databases` option `--databases-file` option

The first method involves the xtrabackup `–tables` option. The option's value is a regular expression that is matched against the fully-qualified database name and table name using the `databasename.tablename` format.

To back up only tables in the `test` database, use the following command:

```
$ xtrabackup --backup --datadir=/var/lib/mysql --target-dir=/data/backups/ \
--tables="^test[.].*"
```

To back up only the `test.t1` table, use the following command:

```
$ xtrabackup --backup --datadir=/var/lib/mysql --target-dir=/data/backups/ \
--tables="^test[.]t1"
```

The `--tables-file` option specifies a file that can contain multiple table names, one table name per line in the file. Only the tables named in the file will be backed up. Names are matched exactly, case-sensitive, with no pattern or regular expression matching. The table names must be fully-qualified in `databasename.tablename` format.

```
$ echo "mydatabase.mytable" > /tmp/tables.txt
$ xtrabackup --backup --tables-file=/tmp/tables.txt
```

The ` –databases` option accepts a space-separated list of the databases and tables to backup in the `databasename[.tablename]` format. In addition to this list, make sure to specify the `mysql`, `sys`, and

`performance_schema` databases. These databases are required when restoring the databases using xtrabackup –copy-back.

Note

Tables processed during the –prepare step may also be added to the backup even if they are not explicitly listed by the parameter if they were created after the backup started.

```
$ xtrabackup --databases='mysql sys performance_schema test ...'
```

The –databases-file option specifies a file that can contain multiple databases and tables in the `databasename[.tablename]` format, one element name per line in the file. Names are matched exactly, case-sensitive, with no pattern or regular expression matching.

Note

Tables processed during the –prepare step may also be added to the backup even if they are not explicitly listed by the parameter if they were created after the backup started.

<a href="https://www.percona.com/about">Percona LLC and/or its affiliates © 2023 — Cookie Consent

Next step

Prepare the backup →

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-10-04

8.4.2 Prepare a partial backup

The procedure is analogous to restoring individual tables: apply the logs and use the `--export` option:

```
$ xtrabackup --prepare --export --target-dir=/path/to/partial/backup
```

When you use the `--prepare` option on a partial backup, you will see warnings about tables that don't exist. This is because these tables exist in the data dictionary inside InnoDB, but the corresponding .ibd files don't exist. They were not copied into the backup directory. These tables will be removed from the data dictionary, and when you restore the backup and start InnoDB, they will no longer exist and will not cause any errors or warnings to be printed to the log file.

Could not find any file associated with the tablespace ID: 5

Use `--innodb-directories` to find the tablespace files. If that fails then use `-–innodb-force-recovery=1` to ignore this and to permanently lose all changes to the missing tablespace(s).

Next step

Restore the partition from the backup →

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-10-04

Percona LLC and/or its affiliates © 2023 — Cookie Consent

8.5 Back up individual partitions

8.5.1 Create an individual partitions backup

Percona XtraBackup lets you back up individual partitions because partitions are regular tables with specially formatted names. The only requirement for this feature is to enable the `innodb_file_per_table` option on the server.

There is one caveat about using this kind of backup: you can not copy back the prepared backup. Restoring partial backups should be done by importing the tables.

There are three ways of specifying which part of the whole data will be backed up: regular expressions (–tables), enumerating the tables in a file (–tables) or providing a list of databases (–databases).

The regular expression provided to this option will be matched against the fully qualified database name and table name, in the form of `database-name.table-name`.

If the partition 0 is not backed up, Percona XtraBackup cannot generate a .cfg file. MySQL 8.0 stores the table metadata in partition 0.

For example, this operation takes a back-up of the partition `p4` from the table `name` located in the database `imdb`:

```
$ xtrabackup --tables=^imdb[.]name#p#p4 --backup
```

If partition 0 is not backed up, the following errors may occur:

The error message

```
xtrabackup: export option not specified
xtrabackup: error: cannot find dictionary record of table imdb/name#p#p4
```

Note that this option is passed to `xtrabackup --tables` and is matched against each table of each database, the directories of each database will be created even if they are empty.

Next step

Prepare an individual partitions backup →

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

<a href="https://www.percona.com/about">Percona LLC and/or its affiliates © 2023 — Cookie Consent

Last update: 2023-10-04

8.5.2 Prepare an individual partitions backup

For preparing partial backups, the procedure is analogous to restoring individual tables. Apply the logs and use xtrabackup –export:

```
$ xtrabackup --apply-log --export /mnt/backup/2012-08-28_10-29-09
```

You may see warnings in the output about tables that do not exist. This happens because InnoDB-based engines stores its data dictionary inside the tablespace files. xtrabackup removes the missing tables (those that haven't been selected in the partial backup) from the data dictionary in order to avoid future warnings or errors.

Next step

Restore the partition from the backup →

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-10-04

8.6 Restore a backup

8.6.1 Restore a backup

Warning

Backup needs to be prepared before it can be restored.

For convenience, *xtrabackup* binary has the `--copy-back` option to copy the backup to the datadir of the server:

```
$ xtrabackup --copy-back --target-dir=/data/backups/
```

If you don't want to save your backup, you can use the `--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.

Percona LLC and/or its affiliates © 2023 — Cookie Consent

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 cannot restore to a datadir of a running mysqld instance (except when importing a partial backup).

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 are preserved, in most cases you must change the files' ownership to `mysql` before starting the database server, as the files are owned by the user who created the backup:

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

Data is now restored, and you can start the server.

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-08-17

8.6.2 Restore a partial backup

Restoring should be done by restoring individual tables in the partial backup to the server.

It can also be done by copying back the prepared backup to a "clean" datadir (in that case, make sure to include the `mysql` database) to the datadir you are moving the backup to. A system database can be created with the following:

```
$ sudo mysql --initialize --user=mysql
```

Once you start the server, you may see mysql complaining about missing tablespaces:

Expected output

```
2021-07-19T12:42:11.077200Z 1 [Warning] [MY-012351] [InnoDB] Tablespace 4, name 'test1/t1', file
'./d2/test1.ibd' is missing!
2021-07-19T12:42:11.077300Z 1 [Warning] [MY-012351] [InnoDB] Tablespace 4, name 'test1/t1', file
'./d2/test1.ibd' is missing!
```


Percona LLC and/or its affiliates © 2023 — Cookie Consent

In order to clean the orphan database from the data dictionary, you must manually create the missing database directory and then `DROP` this database from the server.

Example of creating the missing database:

```
$ mkdir /var/lib/mysql/test1/d2
```

Example of dropping the database from the server:

```
mysql> DROP DATABASE d2;
```

Expected output

```
Query OK, 2 rows affected (0.5 sec)
```

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-06-12

8.6.3 Restore the partition from the backup

Restoring should be done by importing the tables in the partial backup to the server.

First step is to create new table in which data will be restored.

```
mysql> CREATE TABLE `name_p4` (
`id` int(11) NOT NULL AUTO_INCREMENT,
`name` text NOT NULL,
`imdb_index` varchar(12) DEFAULT NULL,
`imdb_id` int(11) DEFAULT NULL,
`name_pcode_cf` varchar(5) DEFAULT NULL,
`name_pcode_nf` varchar(5) DEFAULT NULL,
`surname_pcode` varchar(5) DEFAULT NULL,
PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=2812744 DEFAULT CHARSET=utf8
```

Note

Generate a .cfg metadata file by running `FLUSH TABLES ... FOR EXPORT`. The command can only be run on a table, not on the individual table partitions. The file is located in the table schema directory and is used for schema verification when importing the tablespace.

Percona LLC and/or its affiliates © 2023 — Cookie Consent

To restore the partition from the backup, the tablespace must be discarded for that table:

```
mysql> ALTER TABLE name_p4 DISCARD TABLESPACE;
```

The next step is to copy the `.ibd` file from the backup to the MySQL data directory:

```
cp /mnt/backup/2012-08-28_10-29-09/imdb/name#P#p4.ibd /var/lib/mysql/imdb/name_p4.ibd
```

Note

Make sure that the copied files can be accessed by the user running MySQL.

The last step is to import the tablespace:

```
mysql> ALTER TABLE name_p4 IMPORT TABLESPACE;
```

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-10-12

8.7 Replicate

8.7.1 How to set up a replica for replication in 6 simple steps with Percona XtraBackup

Data is, by far, the most valuable part of a system. Having a backup done systematically and available for a rapid recovery in case of failure is admittedly essential to a system. However, it is not common practice because of its costs, infrastructure needed or even the boredom associated to the task. Percona XtraBackup is designed to solve this problem.

You can have almost real-time backups in 6 simple steps by setting up a replication environment with Percona XtraBackup.

Things you need

Setting up a replica for replication with *Percona XtraBackup* is a straightforward procedure. In order to keep it simple, here is a list of the things you need to follow the steps without hassles:

`Source`

<a href="https://www.percona.com/about">Percona LLC and/or its affiliates © 2023 — Cookie Consent

A system with a *MySQL*-based server installed, configured and running. This system is called `Source`. The `Source` server stores your data and can be replicated. We assume the following about this system:

- the *MySQL* server is able to communicate with others by the standard TCP/IP port;
- the *SSH* server is installed and configured;
- you have a user account in the system with the appropriate permissions;
- you have a MySQL's user account with appropriate privileges.
- server has binlogs enabled and server-id set up to 1.

`Replica`

Another system, with a *MySQL*-based server installed on it. We refer to this machine as `Replica` and assume the same things we did about `Source`, except that the server-id on `Replica` is 2.

`Percona XtraBackup`

We use this backup tool. Install Percona XtraBackup on both computers for convenience.

Note

It is not recommended to mix MySQL variants (Percona Server, MySQL) in your replication setup. This may produce incorrect `xtrabackup_slave_info` file when adding a new replica.

1. Make a backup on the `Source` and prepare it

At the `Source`, issue the following to a shell:

```
$ xtrabackup --backup --user=yourDBuser --password=MaGiCdB1 --target-dir=/path/to/
backupdir
```

After this is finished you should get:

Expected output

```
xtrabackup: completed OK!
```

This operation makes a copy of your *MySQL* data dir to the `/path/to/backupdir` directory. You have told *Percona XtraBackup* to connect to the database server using your database user and password, and do a hot backup of all your data in it (all *MyISAM*, *InnoDB* tables and indexes in them).

In order for snapshot to be consistent, prepare the data on the source:

```
$ xtrabackup --prepare --target-dir=/path/to/backupdir
```

Select the path where your snapshot has been taken. Apply the transaction logs to the data files and your data files are ready to be used by the MySQL server.

Percona LLC and/or its affiliates © 2023 — Cookie Consent

Percona XtraBackup reads the my.cnf file to locate your data. If you have your configuration file in a non-standard place, you should use the flag `--defaults-file =/location/of/my.cnf`.

If you want to skip writing the username and password every time you want to access *MySQL*, you can set it up in `.mylogin.cnf` as follows:

```
mysql_config_editor set --login-path=client --host=localhost --user=root --password
```

For more information, see MySQL Configuration Utility.

This statement provides root access to MySQL.

2. Copy backed up data to the Replica

On the Source, use rsync or scp to copy the data from the Source to the Replica. If you are syncing the data directly to replica's data directory, we recommend that you stop the `mysqld` there.

```
$ rsync -avpP -e ssh /path/to/backupdir Replica:/path/to/mysql/
```

After data is copied, you can back up the original or previously installed *MySQL* datadir.

Note

Make sure mysqld is shut down before you move the contents of its datadir, or move the snapshot into its datadir.

Run the following commands on the Replica:

```
$ mv /path/to/mysql/datadir /path/to/mysql/datadir_bak
```

and move the snapshot from the `Source` in its place:

```
$ xtrabackup --move-back --target-dir=/path/to/mysql/backupdir
```

After you copy data over, make sure the Replica *MySQL* has the proper permissions to access them.

```
$ chown mysql:mysql /path/to/mysql/datadir
```

If the ibdata and iblog files are located in directories outside the datadir, you must put these files in their proper place after the logs have been applied.

3. Configure the Source's MySQL server

On the source, run the following command to add the appropriate grant. This grant allows the replica to be able to connect to source:

```
mysql> GRANT REPLICATION SLAVE ON *.*  TO 'repl'@'$replicaip'
IDENTIFIED BY '$replicapass';
```


<a href="https://www.percona.com/about">Percona LLC and/or its affiliates © 2023 — Cookie Consent

Also make sure that firewall rules are correct and that the `Replica` can connect to the `Source`. Run the following command on the Replica to test that you can run the mysql client on `Replica`, connect to the `Source`, and authenticate.

```
mysql> mysql --host=Source --user=repl --password=$replicapass
```

Verify the privileges.

```
mysql> SHOW GRANTS;
```

4. Configure the Replica's MySQL server

Copy the `my.cnf` file from the `Source` to the `Replica`:

```
$ scp user@Source:/etc/mysql/my.cnf /etc/mysql/my.cnf
```

and change the following options in /etc/mysql/my.cnf:

```
server-id=2
```

and start/restart mysqld on the `Replica`.

In case you're using init script on Debian-based system to start mysqld, be sure that the password for `debian-sys-maint` user has been updated, and it's the same as that user's password on the `Source`. Password can be seen and updated in `/etc/mysql/debian.cnf`.

5. Start the replication

On the `Replica`, review the content of the `xtrabackup_binlog_info` file:

```
$ cat /var/lib/mysql/xtrabackup_binlog_info
```

The results should resemble the following:

Expected output

```
Source-bin.000001     481
```

Do the following on a MySQL console and use the username and password you've set up in STEP 3:

Use the `CHANGE_REPLICATION_SOURCE_TO` statement

```
CHANGE REPLICATION SOURCE TO
    SOURCE_HOST='$sourceip',
    SOURCE_USER='repl',
    SOURCE_PASSWORD='$replicapass',
    SOURCE_LOG_FILE='Source-bin.000001',
    SOURCE_LOG_POS=481;
```


Percona and/or its affiliates © 2023 — Cookie Consent

Start the replica:

```
START REPLICA;
```

6. Check

On the `Replica`, check that everything went OK with:

```
SHOW REPLICA STATUS\G
```

The result shows the status:

Expected output

```
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
Seconds_Behind_Master: 13
```

Both `IO` and `SQL` threads need to be running. The `Seconds_Behind_Master` means the `SQL` currently being executed has a `current_timestamp` of 13 seconds ago. It is an estimation of the lag between the `Source` and the `Replica`. Note that at the beginning, a high value could be shown because the `Replica` has to "catch up" with the `Source`.

Adding more replicas to the Source

You can use this procedure with slight variation to add new replicas to a source. We will use *Percona XtraBackup* to clone an already configured replica. We will continue using the previous scenario for convenience, but we will add a `NewReplica` to the plot.

At the `Replica`, do a full backup:

```
$ xtrabackup --user=yourDBuser --password=MaGiCiGaM \
   --backup --slave-info --target-dir=/path/to/backupdir
```

By using the `--slave-info` *Percona XtraBackup* creates additional file called `xtrabackup_slave_info`.

Apply the logs:

```
$ xtrabackup --prepare --use-memory=2G --target-dir=/path/to/backupdir/
```

Note

In the `prepare` phase, the `--use-memory` parameter speeds up the process if the amount of RAM assigned to the option is available. Use the parameter only in the `prepare` phase. In the other phases the parameter makes the application lazy allocate this memory (reserve) but does not affect database pages.

Copy the directory from the `Replica` to the `NewReplica`:

<a href="https://www.percona.com/about">Percona LLC and/or its affiliates © 2023 — Cookie Consent

Note

Make sure mysqld is shut down on the `NewReplica` before you copy the contents the snapshot into its datadir.

```
rsync -avprP -e ssh /path/to/backupdir NewReplica:/path/to/mysql/datadir
```

For example, to set up a new user, `user2`, you add another grant on the Source:

```
> GRANT REPLICATION SLAVE ON *.*  TO 'user2'@'$newreplicaip'
 IDENTIFIED BY '$replicapass';
```

On the `NewReplica`, copy the configuration file from the `Replica`:

```
$ scp user@Replica:/etc/mysql/my.cnf /etc/mysql/my.cnf
```

Make sure you change the server-id variable in `/etc/mysql/my.cnf` to 3 and disable the replication on start:

```
skip-slave-start
server-id=3
```

After setting `server_id`, start mysqld.

Fetch the source_log_file and source_log_pos from the file `xtrabackup_slave_info`, execute the statement for setting up the source and the log file for the NewReplica:

```
CHANGE REPLICATION SOURCE TO
    SOURCE_HOST='$sourceip',
    SOURCE_USER='repl',
    SOURCE_PASSWORD='$replicapass',
    SOURCE_LOG_FILE='Source-bin.000001',
    SOURCE_LOG_POS=481;
```

Start the replica:

```
> START REPLICA;
```

If both IO and SQL threads are running when you check the `NewReplica`, server is replicating the `Source`.

See also

How to create a new (or repair a broken) GTID based slave

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Percona LLC and/or its affiliates © 2023 — Cookie Consent

Community Forum Get a Percona Expert

Last update: 2023-10-19

8.7.2 How to create a new (or repair a broken) GTID-based replica

Percona XtraBackup automatically stores the `GTID` value in the `xtrabackup_binlog_info` when doing the backup of MySQL and Percona Server for MySQL 8.2 with the `GTID` mode enabled. This information can be used to create a new (or repair a broken) `GTID` -based replica.

1. Take a backup from any server on the replication environment, source or replica

The following command takes a backup and saves it in the `/data/backups/$TIMESTAMP` folder:

```
$ xtrabackup --backup --target-dir=/data/backups/
```

In the destination folder, there will be a file with the name `xtrabackup_binlog_info`. This file contains both binary log coordinates and the `GTID` information.

```
$ cat xtrabackup_binlog_info
```

The result could look like this:

Expected output

```
mysql-bin.000002    1232        c777888a-b6df-11e2-a604-080027635ef5:1-4
```

That information is also printed by xtrabackup after taking the backup:

Expected output

```
xtrabackup: MySQL binlog position: filename 'mysql-bin.000002', position 1232, GTID of the last
change 'c777888a-b6df-11e2-a604-080027635ef5:1-4'
```

2. Prepare the backup

The backup will be prepared with the following command on the Source:

```
$ xtrabackup --prepare --target-dir=/data/backup
```

You need to select the path where your snapshot has been taken, for example `/data/backups/2023-05-07_08-33-33`. If everything is ok you should get the same OK message. Now, the transaction logs are applied to the data files, and new ones are created: your data files are ready to be used by the MySQL server.

<a href="https://www.percona.com/about">Percona LLC and/or its affiliates © 2023 — Cookie Consent

3. Move the backup to the destination server

Use rsync or scp to copy the data to the destination server. If you are synchronizing the data directly to the already running replica's data directory it is advised to stop the MySQL server there.

```
$ rsync -avprP -e ssh /path/to/backupdir/$TIMESTAMP NewSlave:/path/to/mysql/
```

After you copy the data over, make sure MySQL has proper permissions to access them.

```
$ chown mysql:mysql /path/to/mysql/datadir
```

4. Configure and start replication

Set the `gtid_purged` variable to the `GTID` from `xtrabackup_binlog_info`. Then, update the information about the source node and, finally, start the replica.

Note

The example above is applicable to Percona XtraDB Cluster. The `wsrep_on` variable is set to 0 before resetting the source (`RESET MASTER`). The reason is that Percona XtraDB Cluster will not allow resetting the source if `wsrep_on=1`.

```
# Using the mysql shell
 > SET SESSION wsrep_on = 0;
 > RESET MASTER;
 > SET SESSION wsrep_on = 1;
 > SET GLOBAL gtid_purged='<gtid_string_found_in_xtrabackup_binlog_info>';
 > CHANGE REPLICATION SOURCE TO
             SOURCE_HOST="$masterip",
             SOURCE_USER="repl",
             SOURCE_PASSWORD="$slavepass",
             SOURCE_AUTO_POSITION = 1;
 > START REPLICA;
```

5. Check the replication status

The following command returns the replica status:

```
mysql> SHOW REPLICA STATUS\G
```

The results should be similar to the following:

Expected output

```
[..]
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
[...]
Retrieved_Gtid_Set: c777888a-b6df-11e2-a604-080027635ef5:5
Executed_Gtid_Set: c777888a-b6df-11e2-a604-080027635ef5:1-5
```


Percona and/or its affiliates © 2023 — Cookie Consent

We can see that the replica has retrieved a new transaction with step 5, so transactions from 1 to 5 are already on the replica.

We have created a new replica in our `GTID` based replication environment.

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-10-19

8.7.3 Make backups in replication environments

There are options specific to back up from a replication replica.

The `--slave-info` option

This option is useful when backing up a replication replica server. It prints the binary log position and name of the source server. It also writes this information to the `xtrabackup_slave_info` file as a `CHANGE MASTER` statement.

This option is useful for setting up a new replica for this source. You can start a replica server with this backup and issue the statement saved in the `xtrabackup_slave_info` file. More details of this procedure can be found in How to setup a replica for replication in 6 simple steps with Percona XtraBackup.

The `--safe-slave-backup` option

In order to assure a consistent replication state, this option stops the replication SQL thread and waits to start backing up until `Slave_open_temp_tables` in `SHOW STATUS` is zero. If there are no open temporary tables, the backup will take place, otherwise the SQL thread will be started and stopped until there are no open temporary tables. The backup will fail if `Slave_open_temp_tables` does not become zero after `--safe-slave-backup-timeout` seconds (defaults to 300 seconds). The replication SQL thread will be restarted when the backup finishes.

Note

Using a safe-slave-backup option stops the SQL replica thread before copying the InnoDB files.

Using this option is always recommended when taking backups from a replica server.

<a href="https://www.percona.com/about">Percona LLC and/or its affiliates © 2023 — Cookie Consent

Warning

Make sure your replica is a true replica of the source before using it as a source for backup. A good tool to validate a replica is pt-table-checksum.

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-10-12

8.7.4 Verify backups with replication and pt-checksum

One way to verify if the backup is consistent is by setting up the replication and running pt-table-checksum. This can be used to verify any type of backups, but before setting up replication, backup should be prepared and be able to run (this means that incremental backups should be merged to full backups, encrypted backups decrypted etc.).

Set up the replication

How to set up a replica for replication in 6 simple steps with Percona XtraBackup guide provides a detailed instructions on how to take the backup and set up the replication.

For checking the backup consistency you can use either the original server where the backup was taken, or another test server created by using a different backup method (such as cold backup, mysqldump or LVM snapshots) as the source server in the replication setup.

Use pt-table-checksum

This tool is part of the *Percona Toolkit*. It performs an online replication consistency check by executing checksum queries on the source, which produces different results on replicas that are inconsistent with the source.

After you confirmed that replication has been set up successfully, you can install or download *pt-table-checksum*. This example shows downloading the latest version of *pt-table-checksum*:

```
$ wget percona.com/get/pt-table-checksum
```

Note

In order for pt-table-checksum to work correctly `libdbd-mysql-perl` will need to be installed on *Debian/Ubuntu* systems or `perl-DBD-MySQL` on *RHEL/CentOS*. If you installed the *percona-toolkit* package from the Percona repositories package manager should install those libraries automatically.

Percona LLC and/or its affiliates © 2023 — Cookie Consent

After this command has been run, *pt-table-checksum* will be downloaded to your current working directory.

Running the *pt-table-checksum* on the source will create `percona` database with the `checksums` table which will be replicated to the replicas as well. Example of the *pt-table-checksum* will look like this:

```
$ ./pt-table-checksum
```

The results are similar to the following:

Expected output

```
TS ERRORS  DIFFS     ROWS  CHUNKS SKIPPED    TIME TABLE
04-30T11:31:50      0      0   633135       8       0   5.400 exampledb.aka_name
04-30T11:31:52      0      0   290859       1       0   2.692 exampledb.aka_title
Checksumming exampledb.user_info:  16% 02:27 remain
Checksumming exampledb.user_info:  34% 01:58 remain
Checksumming exampledb.user_info:  50% 01:29 remain
Checksumming exampledb.user_info:  68% 00:56 remain
Checksumming exampledb.user_info:  86% 00:24 remain
04-30T11:34:38      0      0 22187768     126       0 165.216 exampledb.user_info
04-30T11:38:09      0      0        0       1       0   0.033 mysql.time_zone_name
04-30T11:38:09      0      0        0       1       0   0.052 mysql.time_zone_transition
04-30T11:38:09      0      0        0       1       0   0.054 mysql.time_zone_transition_type
04-30T11:38:09      0      0        8       1       0   0.064 mysql.user
```

If all the values in the `DIFFS` column are 0 that means that backup is consistent with the current setup.

In case backup was not consistent *pt-table-checksum* should spot the difference and point to the table that does not match. Following example shows adding new user on the backed up replica in order to simulate the inconsistent backup:

```
mysql> grant usage on exampledb.* to exampledb@localhost identified by
'thisisnewpassword';
```

If we run the *pt-table-checksum* now difference should be spotted

```
$ ./pt-table-checksum
```

The results are similar to the following:

<a href="https://www.percona.com/about">Percona LLC and/or its affiliates © 2023 — Cookie Consent

Expected output

```
TS ERRORS  DIFFS     ROWS  CHUNKS SKIPPED    TIME TABLE
04-30T11:31:50      0      0   633135       8       0   5.400 exampledb.aka_name
04-30T11:31:52      0      0   290859       1       0   2.692 exampledb.aka_title
Checksumming exampledb.user_info:  16% 02:27 remain
Checksumming exampledb.user_info:  34% 01:58 remain
Checksumming exampledb.user_info:  50% 01:29 remain
Checksumming exampledb.user_info:  68% 00:56 remain
Checksumming exampledb.user_info:  86% 00:24 remain
04-30T11:34:38      0      0 22187768     126       0 165.216 exampledb.user_info
04-30T11:38:09      0      0        0       1       0   0.033 mysql.time_zone_name
04-30T11:38:09      0      0        0       1       0   0.052 mysql.time_zone_transition
04-30T11:38:09      0      0        0       1       0   0.054 mysql.time_zone_transition_type
04-30T11:38:09      1      0        8       1       0   0.064 mysql.user
```

This output shows that source and the replica aren't in consistent state and that the difference is in the `mysql.user` table.

More information on different options that pt-table-checksum provides can be found in the *pt-table-checksum* documentation.

Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum **Get a Percona Expert**

Last update: 2023-10-10

<a href="https://www.percona.com/about">Percona LLC and/or its affiliates © 2023 — Cookie Consent

9. Troubleshoot

9.1 xtrabackup exit codes

The *xtrabackup* binary exits with the traditional success value of 0 after a backup when no error occurs. If an error occurs during the backup, the exit value is 1.

In certain cases, the exit value can be something other than 0 or 1, due to the command-line option code included from the *MySQL* libraries. An unknown command-line option, for example, will cause an exit code of 255.

9.1.1 Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum **Get a Percona Expert**

Last update: 2023-06-12

<a href="https://www.percona.com/about">Percona LLC and/or its affiliates © 2023 — Cookie Consent

10. Reference

10.1 Index of files created by Percona XtraBackup

- Information related to the backup and the server

File Name	Description
`backup-my.cnf`	This file contains information to start the mini instance of InnoDB during the `--prepare`. This **not** a backup of the original `my.cnf`. The InnoDB configuration is read from the file `backup-my.cnf` created by **xtrabackup** when the backup was made. The `--prepare` uses InnoDB configuration from `backup-my.cnf` by default, or from `--defaults-file`, if specified. The InnoDB's configuration in this context means server variables that affect dataformat, i.e., `innodb_page_size` option, `innodb_log_block_size`, etc. Location-related variables, like `innodb_log_group_home_dir` or `innodb_data_file_path` are always ignored by `--prepare`, so preparing a backup always works with data files from the back directory, rather than any external ones.

`xtrabackup_checkpoints`	The type of the backup (for example, full or incremental), its state (for example, prepared) and the LSN range contained in it. This information is used for incremental backups. Example of the `xtrabackup_checkpoints` after taking a full backup:
	`backup_type = full-backuped`
	`from lsn= 0`
	`to_lsn = 15188961605`
	`last_lsn = 15188961605`
	Example of the `xtrabackup_checkpoints` after taking an incremental backup:
	`backup_type = incremental`
	`from_lsn = 15188961605`
	`to_lsn = 15189350111`
	`last_lsn = 15189350111`

`xtrabackup_binlog_info`	The binary log file used by the server and its position at the moment of the backup. A result of the following query: `SELECT server_uuid, local, replication, storage_engines FROM performance_schema.log_status;`
`xtrabackup_binlog`	The **xtrabackup** binary used in the process.
`xtrabackup_logfile`	Contains data needed for running the: `--prepare`. The bigger this file is the `--prepare` process will take longer to finish.

<a href="https://www.percona.com/about">Percona LLC and/or its affiliates © 2023 — Cookie Consent

`<table_name>.delta.meta`	This file is going to be created when performing the incremental backup. It contains the per-table delta metadata: page size, size of compressed page (if the value is 0 it means the tablespace isn't compressed) and space id. Example of this file:
	`page_size = 16384`
	`zip_size = 0`
	`space_id = 0`

- Information related to the replication environment (if using the `--slave-info` option):

  `xtrabackup_slave_info`

  The `CHANGE MASTER` statement needed for setting up a replica.

- Information related to the *Galera* and *Percona XtraDB Cluster* (if using the `--galera-info` option):

  `xtrabackup_galera_info`

  Contains the values of `wsrep_local_state_uuid` and `wsrep_last_committed` status variables

10.1.1 Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum **Get a Percona Expert**

Last update: 2023-10-17

10.2 Frequently asked questions

10.2.1 Does Percona XtraBackup 8.2 support making backups of databases in versions prior to 8.2?

Percona XtraBackup 8.2 does not support making backups of databases created in versions prior to 8.2 of MySQL, Percona Server for MySQL or Percona XtraDB Cluster.

10.2.2 Are you aware of any web-based backup management tools (commercial or not) built around Percona XtraBackup*?

ZRM Community is a community tool that uses Percona XtraBackup for Non-Blocking Backups:

"ZRM provides support for non-blocking backups of MySQL using Percona XtraBackup. ZRM with \Percona XtraBackup provides resource utilization management by providing throttling based on the number of IO operations per second. Percona XtraBackup based backups also allow for table level recovery even though the backup was done at the database level (needs the recovery database server to be Percona Server for MySQL with XtraDB)."*

<a href="https://www.percona.com/about">Percona LLC and/or its affiliates © 2023 — Cookie Consent

10.2.3 xtrabackup binary fails with a floating point exception

In most of the cases this is due to not having installed the required libraries (and version) by xtrabackup. Installing the GCC suite with the supporting libraries and recompiling xtrabackup solves the issue. See Compiling and Installing from Source Code for instructions on the procedure.

10.2.4 How xtrabackup handles the ibdata/ib_log files on restore if they aren't in mysql datadir?

In case the `ibdata` and `ib_log` files are located in different directories outside the datadir, you will have to put them in their proper place after the logs have been applied.

10.2.5 Backup fails with Error 24: 'Too many open files'

This usually happens when database being backed up contains large amount of files and Percona XtraBackup can't open all of them to create a successful backup. In order to avoid this error the operating system should be configured appropriately so that Percona XtraBackup can open all its files. On Linux, this can be done with the `ulimit` command for specific backup session or by editing the `/etc/security/limits.conf` to change it globally (NOTE: the maximum possible value that can be set up is `1048576` which is a hard-coded constant in the Linux kernel).

10.2.6 How to deal with skipping of redo logs for DDL operations?

To prevent creating corrupted backups when running DDL operations, Percona XtraBackup aborts if it detects that redo logging is disabled. In this case, the following error is printed:

```
[FATAL] InnoDB: An optimized (without redo logging) DDL operation has been
performed. All modified pages may not have been flushed to the disk yet.
Percona XtraBackup will not be able to take a consistent backup. Retry the backup
operation.
```

Note

- Redo logging is disabled during a sorted index build. To avoid this error, Percona XtraBackup can use metadata locks on tables while they are copied:
- To block all DDL operations, use the `--lock-ddl` option that issues `LOCK TABLES FOR BACKUP`.

10.2.7 Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum Get a Percona Expert

Last update: 2023-10-12

<a href="https://www.percona.com/about">Percona LLC and/or its affiliates © 2023 — Cookie Consent

10.3 Glossary

10.3.1 .CSM

Each table with the CSV Storage Engine has `.CSM` file which contains the metadata of it.

10.3.2 .CSV

Each table with the CSV Storage engine has `.CSV` file which contains the data of it (which is a standard Comma Separated Value file).

10.3.3 .exp

Files with the `.exp` extension are created by Percona XtraBackup per each InnoDB tablespace when the `--export` option is used on prepare. See restore individual tables.

10.3.4 .frm

For each table, the server will create a file with the `.frm` extension containing the table definition (for all storage engines).

10.3.5 General availability (GA)

A finalized version of the product which is made available to the general public. It is the final stage in the software release cycle.

10.3.6 .ibd

On a multiple tablespace setup ([innodb_file_per_table] enabled), MySQL will store each newly created table on a file with a `.ibd` extension.

10.3.7 .MRG

Each table using the MERGE storage engine, besides of a `.frm` file, will have `.MRG` file containing the names of the MyISAM tables associated with it.

10.3.8 .MYD

Each MyISAM table has `.MYD` (MYData) file which contains the data on it.

10.3.9 .MYI

Each MyISAM table has `.MYI` (MYIndex) file which contains the table's indexes.

10.3.10 .opt

MySQL stores options of a database (like charset) in a file with a `.opt` extension in the database directory.

<a href="https://www.percona.com/about">Percona LLC and/or its affiliates © 2023 — Cookie Consent

10.3.11 .par

Each partitioned table has `.par` file which contains metadata about the partitions.

10.3.12 .TRG

The file contains the triggers associated with a table, for example, `\mytable.TRG`. With the `.TRN` file, they represent all the trigger definitions.

10.3.13 .TRN

The file contains the names of triggers that are associated with a table, for example, `\mytable.TRN`. With the `.TRG` file, they represent all the trigger definitions.

10.3.14 backup

The process of copying data or tables to be stored in a different location.

10.3.15 compression

The method that produces backups in a reduced size.

10.3.16 configuration file

The file that contains the server startup options.

10.3.17 crash

An unexpected shutdown which does not allow the normal server shutdown cleanup activities.

10.3.18 crash recovery

The actions that occur when MySQL is restarted after a crash.

10.3.19 data dictionary

The metadata for the tables, indexes, and table columns stored in the InnoDB system tablespace.

10.3.20 datadir

The directory in which the database server stores its data files. Most Linux distribution use `/var/lib/mysql` by default.

10.3.21 full backup

A backup that contains the complete source data from an instance.

Percona and/or its affiliates © 2023 — Cookie Consent

10.3.22 ibdata

The default prefix for tablespace files. For example, `ibdata1` is a 10MB auto-extensible file that MySQL creates for a shared tablespace by default.

10.3.23 incremental backup

A backup stores data from a specific point in time.

10.3.24 InnoDB

Storage engine which provides ACID-compliant transactions and foreign key support, among others improvements over MyISAM. It is the default engine for MySQL 8.2.

10.3.25 innodb_buffer_pool_size

The size in bytes of the memory buffer to cache data and indexes of InnoDB's tables. This aims to reduce disk access to provide better performance.

```
[mysqld]
innodb_buffer_pool_size=8MB
```

10.3.26 innodb_data_home_dir

The directory (relative to `datadir`) where the database server stores the files in a shared tablespace setup. This option does not affect the location of `innodb\_file\_per\_table`. For example:

```
[mysqld]
innodb_data_home_dir = ./
```

10.3.27 innodb_data_file_path

Specifies the names, sizes and location of shared tablespace files:

```
[mysqld]
innodb_data_file_path=ibdata1:50M;ibdata2:50M:autoextend
```

10.3.28 innodb_file_per_table

By default, InnoDB creates tables and indexes in a file-per-tablespace. If the `innodb_file_per_table` variable is disabled, you can enable the variable in your configuration file:

```
[mysqld]
innodb_file_per_table
```

or

start the server with `--innodb_file_per_table`.

<a href="https://www.percona.com/about">Percona LLC and/or its affiliates © 2023 — Cookie Consent

10.3.29 innodb_log_group_home_dir

Specifies the location of the InnoDB log files:

```
[mysqld]
innodb_log_group_home=/var/lib/mysql
```

10.3.30 logical backup

A backup which contains a set of SQL statements. The statements can be used to recreate the databases.

10.3.31 LSN

Each InnoDB page contains a log sequence number(LSN). The LSN is the system version number for the database. Each page's LSN shows how recently it was changed.

10.3.32 my.cnf

The database server's main configuration file. Most Linux distributions place it as `/etc/mysql/my.cnf` or `/etc/my.cnf`, but the location and name depends on the particular installation. Note that this method is not the only way of configuring the server, some systems rely on the command options.

10.3.33 MyISAM

The MySQL default storage engine until version 5.5. It doesn't fully support transactions but in some scenarios may be faster than InnoDB. Each table is stored on disk in 3 files: `.frm`, `.MYD`, `.MYI`.

10.3.34 physical backup

A backup that copies the data files.

10.3.35 point in time recovery

This method restores the data into the state it was at any selected point of time.

10.3.36 prepared backup

A consistent set of backup data that is ready to be restored.

10.3.37 restore

Copies the database backups taken using the backup command to the original location or a different location. A restore returns data that has been either lost, corrupted, or stolen to the original condition at a specific point in time.

10.3.38 Tech preview

A tech preview item can be a feature, a variable, or a value within a variable. Before using this feature in production, we recommend that you test restoring production from physical backups in your

<a href="https://www.percona.com/about">Percona LLC and/or its affiliates © 2023 — Cookie Consent

environment and also use an alternative backup method for redundancy. A tech preview item is included in a release for users to provide feedback. The item is either updated and released as general availability(GA) or removed if not useful. The functionality can change from tech preview to GA.

10.3.39 xbcrypt

To support the encryption and the decryption of the backups. This utility has been modeled after the xbstream binary to perform encryption and decryption outside Percona XtraBackup.

10.3.40 xbstream

To support simultaneous compression and streaming, Percona XtraBackup uses the xbstream format. For more information see xbstream

10.3.41 XtraDB

Percona XtraDB is an enhanced version of the InnoDB storage engine, designed to better scale on modern hardware. Percona XtraDB includes features which are useful in a high performance environment. It is fully backward-compatible, and is a drop-in replacement for the standard InnoDB storage engine. For more information, see The Percona XtraDB Storage Engine.

10.3.42 Zstandard (ZSTD)

`ZSTD` is a fast lossless compression algorithm that targets real-time compression scenarios and better compression ratios.

10.3.43 Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum **Get a Percona Expert**

Last update: 2023-10-12

10.4 Percona Toolkit version checking

Some Percona software contains “version checking” functionality which is a feature that enables Percona software users to be notified of available software updates to improve your environment security and performance. Alongside this, the version check functionality also provides Percona with information relating to which software versions you are running, coupled with the environment confirmation which the software is running within. This helps enable Percona to focus our development effort accordingly based on trends within our customer community.

The purpose of this document is to articulate the information that is collected, as well as to provide guidance on how to disable this functionality if desired.

<a href="https://www.percona.com/about">Percona LLC and/or its affiliates © 2023 — Cookie Consent

10.4.1 Usage

Version Check was implemented in *Percona Toolkit* 2.1.4, and was enabled by default in version 2.2.1. Currently, it is supported as a `--[no]version-check` option by a number of tools in Percona Toolkit, *Percona XtraBackup*, and *Percona Monitoring and Management* (PMM).

When launched with Version Check enabled, the tool that supports this feature connects to a Percona's *version check service* via a secure HTTPS channel. It compares the locally installed version for possible updates, and also checks versions of the following software:

- Operating System
- Percona Monitoring and Management (PMM)
- MySQL
- Perl
- MySQL driver for Perl (DBD::mysql)
- Percona Toolkit

Then it checks for and warns about versions with known problems if they are identified as running in the environment.

Each version check request is logged by the server. Stored information consists of the checked system unique ID followed by the software name and version. The ID is generated either at installation or when the *version checking* query is submitted for the first time.

Note

Prior to version 3.0.7 of *Percona Toolkit*, the system ID was calculated as an MD5 hash of a hostname, and starting from *Percona Toolkit* 3.0.7 it is generated as an MD5 hash of a random number. *Percona XtraBackup* continues to use hostname-based MD5 hash.

As a result, the content of the sent query is as follows:

Expected output

```
85624f3fb5d2af8816178ea1493ed41a;DBD::mysql;4.044
c2b6d625ef3409164cbf8af4985c48d3;MySQL;MySQL Community Server (GPL) 5.7.22-log
85624f3fb5d2af8816178ea1493ed41a;OS;Manjaro Linux
85624f3fb5d2af8816178ea1493ed41a;Percona::Toolkit;3.0.11-dev
85624f3fb5d2af8816178ea1493ed41a;Perl;5.26.2
```

10.4.2 Disabling version check

Although the *version checking* feature does not collect any personal information, you might prefer to disable this feature, either one time or permanently. To disable it one time, use `--no-version-check` option when invoking the tool from a Percona product which supports it. Here is a simple example which shows running pt-diskstats tool from the *Percona Toolkit* with *version checking* turned off:

```
pt-diskstats --no-version-check
```


Percona LLC and/or its affiliates © 2023 — Cookie Consent

Disabling *version checking* permanently can be done by placing `no-version-check` option into the configuration file of a Percona product (see correspondent documentation for exact file name and syntax). For example, in case of *Percona Toolkit* this can be done in a global configuration file `/etc/percona-toolkit/percona-toolkit.conf`:

```
# Disable Version Check for all tools:
no-version-check
```

In case of *Percona XtraBackup* this can be done in its configuration file in a similar way:

```
[xtrabackup]
no-version-check
```

10.4.3 Frequently asked questions

Why is this functionality enabled by default?

We believe having this functionality enabled improves security and performance of environments running Percona Software and it is good choice for majority of the users.

Why not rely on Operating System's built in functionality for software updates?

In many environments the Operating Systems repositories may not carry the latest version of software and newer versions of software often installed manually, so not being covered by operating system wide check for updates.

Why do you send more information than just the version of software being run as a part of version check service?

Compatibility problems can be caused by versions of various components in the environment, for example problematic versions of Perl, DBD or MySQL could cause operational problems with Percona Toolkit.

10.4.4 Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum **Get a Percona Expert**

Last update: 2023-07-10

10.5 Trademark policy

This Trademark Policy is to ensure that users of Percona-branded products or services know that what they receive has really been developed, approved, tested and maintained by Percona.

<a href="https://www.percona.com/about">Percona LLC and/or its affiliates © 2023 — Cookie Consent

Trademarks help to prevent confusion in the marketplace, by distinguishing one company's or person's products and services from another's.

Percona owns a number of marks, including but not limited to Percona, XtraDB, Percona XtraDB, XtraBackup, Percona XtraBackup, Percona Server, and Percona Live, plus the distinctive visual icons and logos associated with these marks. Both the unregistered and registered marks of Percona are protected.

Use of any Percona trademark in the name, URL, or other identifying characteristic of any product, service, website, or other use is not permitted without Percona's written permission with the following three limited exceptions.

First, you may use the appropriate Percona mark when making a nominative fair use reference to a bona fide Percona product.

Second, when Percona has released a product under a version of the GNU General Public License ("GPL"), you may use the appropriate Percona mark when distributing a verbatim copy of that product in accordance with the terms and conditions of the GPL.

Third, you may use the appropriate Percona mark to refer to a distribution of GPL-released Percona software that has been modified with minor changes for the sole purpose of allowing the software to operate on an operating system or hardware platform for which Percona has not yet released the software, provided that those third party changes do not affect the behavior, functionality, features, design or performance of the software. Users who acquire this Percona-branded software receive substantially exact implementations of the Percona software.

Percona reserves the right to revoke this authorization at any time in its sole discretion. For example, if Percona believes that your modification is beyond the scope of the limited license granted in this Policy or that your use of the Percona mark is detrimental to Percona, Percona will revoke this authorization. Upon revocation, you must immediately cease using the applicable Percona mark. If you do not immediately cease using the Percona mark upon revocation, Percona may take action to protect its rights and interests in the Percona mark. Percona does not grant any license to use any Percona mark for any other modified versions of Percona software; such use will require our prior written permission.

Neither trademark law nor any of the exceptions set forth in this Trademark Policy permit you to truncate, modify or otherwise use any Percona mark as part of your own brand. For example, if XYZ creates a modified version of the Percona Server, XYZ may not brand that modification as "XYZ Percona Server" or "Percona XYZ Server", even if that modification otherwise complies with the third exception noted above.

In all cases, you must comply with applicable law, the underlying license, and this Trademark Policy, as amended from time to time. For instance, any mention of Percona trademarks should include the full trademarked name, with proper spelling and capitalization, along with attribution of ownership to Percona LLC and/or its affiliates. For example, the full proper name for XtraBackup is Percona XtraBackup. However, it is acceptable to omit the word "Percona" for brevity on the second and subsequent uses, where such omission does not cause confusion.

In the event of doubt as to any of the conditions or exceptions outlined in this Trademark Policy, please contact trademarks@percona.com for assistance and we will do our very best to be helpful.

Percona LLC and/or its affiliates © 2023 — Cookie Consent

10.5.1 Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum **Get a Percona Expert**

Last update: 2023-06-24

10.6 Copyright and licensing information

10.6.1 Documentation licensing

Percona XtraBackup documentation is (C)2009-2023 Percona LLC and/or its affiliates and is distributed under the Creative Commons Attribution 4.0 International License.

10.6.2 Get expert help

If you need assistance, visit the community forum for comprehensive and free database knowledge, or contact our Percona Database Experts for professional support and services.

Community Forum **Get a Percona Expert**

Last update: 2023-02-15

<a href="https://www.percona.com/about">Percona LLC and/or its affiliates © 2023 — Cookie Consent