MySQL Enterprise Backup 4.0 Release Notes

Abstract

This document lists the changes to the MySQL Enterprise Backup 4.0 product, beginning with the most recent release. Each release section covers added or changed functionality, bug fixes, and known issues, if applicable. For information about changes in a different MySQL Enterprise Backup series, see the release notes for that series.

For additional MySQL Enterprise Backup 4.0 documentation, see the MySQL Enterprise Backup User's Guide (Version 4.0.3).

For legal information, see the Legal Notices.

For help with using MySQL, please visit the MySQL Forums, where you can discuss your issues with other MySQL users.

Document generated on: 2021-10-21 (revision: 23564)

Table of Contents

Preface and Legal Notices

This document lists the changes to the MySQL Enterprise Backup 4.0 product, beginning with the most recent release.

Legal Notices

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at
https://www.oracle.com/corporate/accessibility/.

Access to Oracle Support for Accessibility

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit
https://www.oracle.com/corporate/accessibility/learning-support.html#support-tab.

Changes in MySQL Enterprise Backup 4.0.3 (2016-08-05)

Version 4.0.3 is the latest release in the MySQL Enterprise Backup 4.0 series, including such features like support for MySQL 5.7 and ability to backup and restore encrypted InnoDB tables. See What's New in MySQL Enterprise Backup 4.0? for details.

Bugs Fixed

- MySQL Enterprise Backup 4.0.2 did not work with MySQL 5.7.10 or earlier, quitting unexpectedly during its operations. (Bug #24373967)

Changes in MySQL Enterprise Backup 4.0.2 (2016-07-26)

Functionality Added or Changed

- When there were no tables matching the regular expression specified with the `--include-tables` option during a backup operation, `mysqlbackup` still created a backup, which contained an empty folder for each database on the server. `mysqlbackup` now throws an error when `--include-tables` selects no tables to be backed up. (Bug #18114353)
- MySQL Enterprise Backup can now backup and restore encrypted InnoDB tables. See Working with Encrypted InnoDB Tables and Options for Working with Encrypted InnoDB Tablespaces for details.

Bugs Fixed

- When trying to restore a compressed image backup of a server that had separate undo tablespaces residing in the data directory with the `copy-back-and-apply-log` command, the operation failed at the apply-log phase, as `mysqlbackup` could not load the undo tablespaces. (Bug #23583961)
- Attempts to restore an image backup from the cloud using the `--skip-binlog` option failed with a “global tail magic mismatch” error. This was because `mysqlbackup` failed to perform a non-sequential read from the cloud with gaps caused by the skipping of the binary logs. This fix makes sure `mysqlbackup` can perform such reads. (Bug #23534700)
- When a compressed backup was being restored, if the undo logs had been put into separate tablespaces outside of the data directory on the backed up server, they got restored twice, once mistakenly as general tablespaces with the `.ibd` extension, and once as undo tablespaces without a file extension. This fix makes sure they are restored normally as undo tablespaces only. (Bug #23179194)
- An `extract` operation for an image backup failed with a checksum mismatch error in cases when, during the backup, an InnoDB tablespace file kept growing in size, and `mysqlbackup` failed to put the correct file size in its file header. (Bug #22905984)

  References: This issue is a regression of: Bug #22613568.
- During a `mysqlbackup` operation on a compressed backup (that is, the `--uncompress` option was used), `mysqlbackup`, in some situations, wrote to the log file multiple instances of the message “ERROR: InnoDB: file write at offset > 4 GB,” even though the operation was actually successful. (Bug #22733760)
- Occasionally, some files were missing from an image backup created by the `--backup-to-image` command. It was due to an internal race condition, which this fix eliminates. (Bug #19600687)

Changes in MySQL Enterprise Backup 4.0.1 (2016-05-26)

Functionality Added or Changed

- There are two enhancements on how you can configure SSL host authenticate for cloud operations of MySQL Enterprise Backup:
  - A CA certificate directory, in addition to the default folder of the system, can now be specified with the new `--cloud-ca-path` option.

  - `mysqlbackup` now supports authentication using a CA bundle file, whose path is specified by the new `--cloud-ca-info` option.

  See descriptions for the two new options in Cloud Storage Options for more information. (Bug #22761313)

- `mysqlbackup` used to sync all data from the buffer cache to the hard disk before closing all tables at the end of a backup operation. However, for systems with slow storage devices and databases with a huge number of tables, the sync would increase the backup time significantly. To shorten the backup time for those and other cases, starting with this release, the sync is no longer performed automatically. Users who want the sync to be performed at the end of a backup have to use the new `--free-os-buffers` option. (Bug #22561345)

- MySQL Enterprise Backup now supports backup of slave servers in a multi-source replication setup. (Bug #21830316, Bug #22283631)

- To avoid completing a backup of a slave server when temporary tables are still open on the slave, which will cause the restored slave server to be in an inconsistent replication state, `mysqlbackup` now has a new mechanism for ensuring that all temporary tables have already been closed before finishing a slave backup. See Temporary tables on statement-based replication (SBR) slave for details. A new option, `--safe-slave-backup-timeout`, has been created for specifying the time `mysqlbackup` will wait for all temporary tables to be closed before it times out. (Bug #19158516)

- The compression options can now be used with the `backup-and-apply-log` operation to create a directory backup that is prepared and compressed; the backup can then be restored using the `copy-back` operation and the `--uncompress` option. (Bug #18913565)

- During a `copy-back-and-apply-log` or a `copy-back` operation, `mysqlbackup` now checks that the specified values for the `innodb_log_files_in_group` and `innodb_log_file_size` options match those recorded in the backup's `backup-my.cnf` file, and throws an error if the values do not match. This prevents `mysqlbackup` from restoring the backup with the wrong parameters, which would result in a restored server that cannot be started. (Bug #14751027)

- Values for `MASTER_USER` and `MASTER_PORT` are now included in the `CHANGE MASTER TO` statement in the slave information file (`meta/ibbackup_slave_info`) when the `--slave-info` option is used for backing up a slave server. (Bug #14213115)

Bugs Fixed

- **Microsoft Windows:** On Windows platforms, `mysqlbackup` crashed instead of quitting gracefully when the `--datadir` option was not specified for a `copy-back` or `copy-back-and-apply-log` operation. (Bug #22069093)

- An `apply-log` operation failed when a RENAME TABLE statement within the redo log was applied to the backup even when the renamed table was already included into the backup. This fix prevents the problem by making `mysqlbackup` ignore “dirty” (intermediate) log records when (a) the new table name is already in used, or (b) if the tablespace has not been loaded into memory with the old table name. (Bug #23068440)

- An incremental or compressed backup might fail with an end-of-file error if there are large data files that kept growing during the backup. It was because, as the data files expanded, the write process altered the file sizes, which confused the read process for the same files. With this fix, file sizes and information on them is now properly handled. (Bug #23048004)

  References: This issue is a regression of: Bug #19149210.

- Restoring a cloud backup sometimes failed with `Error 18: Transferred a partial file`. It was because `mysqlbackup` created wrong range headers for its REST requests for partial downloads. (Bug #23035334)

- `mysqlbackup` crashed when a `validate` operation was performed on an incremental backup that contained an undo tablespace but not a system tablespace. It was because `mysqlbackup` did not handle data files for undo tablespaces properly, and this fix corrects that. (Bug #22960185)
- Attempts to restore a compressed image backup using the `copy-back-and-apply-log` command failed with "`ERROR: InnoDB: Missing MLOG_FILE_NAME or MLOG_FILE_DELETE for redo log record ...`." It was because the undo tablespaces were not loaded, so that the log operations on them failed. This fix ensures that the undo tablespaces are loaded. (Bug #22914556)
- If a table had been renamed during a backup operation, it sometimes got copied twice into the backup, once under its old table name and another time under its new name. It was because `mysqlbackup`, when checking for tables that have been changed during the backup process, only checked tables by their `.ibd` file names. With this fix, the tablespace ID is also checked, so that a renamed table will be recognized as such and will not be copied twice. (Bug #22859445)
- Sometimes, when sizes of the data files kept growing during a single-file backup, a broken image backup might be produced, causing subsequent commands on it (for example, `validate` or `copy-back-and apply-log`) to fail. It was due to a mismatch of the file size given in the image file's header and the actual file size on disk, which is now prevented by this fix. (Bug #22613568)
- Offline backups sometimes failed, with occasional crashes of `mysqlbackup`. (Bug #22595461)
- For a backup of a slave server, the file name of the master server's binary log and the binary log position for starting replication, which were stored in the file `backup_varaibles.txt` in the backup as `masterlog_file` and `masterlog_pos`, got corrupted when an `apply-log` or `copy-back-and-apply-log` operation was applied to the backup. (Bug #22329306)
- When `mysqlbackup` came across a file of an unknown file type and its path name contained characters that `mysqlbackup` could not convert to the file system character set, it threw an error. With this fix, `mysqlbackup` continues with its operation in the situation after giving a warning. (Bug #22098742)
- A backup failed if, towards the end of the backup process, `mysqlbackup` found the binary log file that was current at the beginning of the backup had been purged. With this fix, `mysqlbackup` now ignores the fact that the file has been purged, resets the log position to the now current binary log file, and continues with the backup without raising any issues. (Bug #21655145)
- During a backup, `mysqlbackup` performed, by default, an SQL query to get storage engine information that was to be put into the `backup_history` table. Because the query caused all table files on the server to be scanned, it consumed a great amount of IO resources when there were many tables on the server, resulting sometimes in serious performance issues. With this fix, only tables included in the backup are scanned, thus reducing the IO stress on the server. (Bug #21098174)
- When creating a compressed backup, `mysqlbackup` threw an error if a table on the server was dropped in the middle of the process. With this fix, the dropped table is ignored (as it does not need to be restored) and `mysqlbackup` finishes without throwing an error. (Bug #21087079)

  References: See also: Bug #18358912.
- `validate` operations for backup images and `backup-to-image` operations left over a temporary folder (`/tmp`) after the operations were over. (Bug #20912357)
- Backups failed for a server that had once been started with the `--log-bin` option and then restarted without it. It was because `mysqlbackup`, seeing the old binary log index file on the server, looked in vain for the current binary log files, reported that they could not be found, and then exited. With this fix, `mysqlbackup` checks if binary logging is enabled for the server; if it is not, `mysqlbackup` then skips the copying of the binary log into the backup. (Bug #20873010)
- A backup of a slave server failed if, during the backup, a relay log file got purged from the slave server (for example, due to a log file rotation). With this fix, backup continues even if `mysqlbackup` finds a relay log file missing. (Bug #20769891, Bug #76312, Bug #21655314, Bug #19255925)

- When the `trace` level of `mysqlbackup` messages was greater than “0,” if the operation command for `mysqlbackup` was invalid or missing, a stack trace and some error messages were printed, which made it look like `mysqlbackup` had crashed. With this fix, a new message is now shown before the stack trace, to better explain the situation. (Bug #20281022)

- If an incremental backup had already been applied to a directory backup with the `apply-incremental-backup` command and the up-to-date backup was then restored to a data directory, it was possible to restore the same incremental backup again to the data directory using the `copy-back-and-apply-log` command, potentially causing data inconsistencies. With this fix, the incremental data can be reapplied only when the `--force` option is used. Without the `--force` option, the `copy-back-and-apply-log` command skips the apply log operation if the incremental backup is a directory backup and throws an error if it is an image backup. (Bug #18004179)

Changes in MySQL Enterprise Backup 4.0.0 (2015-10-21)

Functionality Added or Changed

- MySQL Enterprise Backup now writes tape information onto the MySQL server when a backup is made to a tape using the System Backup to Tape (SBT) API. See What's New in MySQL Enterprise Backup 4.0? for details.

- MySQL Enterprise Backup 4.0 supports MySQL server 5.7 (starting with 5.7.9; for any earlier versions of MySQL 5.7 or 5.6, please use MySQL Enterprise Backup 3.12 instead), on which:

  - Tables in general tablespaces can be backed up and restored.

  - InnoDB tables with page sizes of 32 and 64K can be backed up and restored.

  - Partitioned tables can be selectively backed up or restored using the `--use-tts`, `--include-tables`, and `--exclude-tables` options.

  See What's New in MySQL Enterprise Backup 4.0? for details.

Bugs Fixed

- After an `apply-incremental-backup` operation on a full backup, `mysqlbackup` printed to the output stream and the message log file the old instead of the updated binary log position. (Bug #21822086)

- When a backup took a long time to perform and the binary logs were rotated in the middle of the process, `mysqlbackup` lost track of the binary log files it was copying, skipping the second last log file and attempting to copy the last one twice; that resulted in a file creation error, at which point `mysqlbackup` exited without releasing its lock on the tables in the database. With this fix, all binary log files are now copied properly, and the lock on the tables is released at the end of the backup process as usual. (Bug #20971763)

- When restoring an incremental backup image, if the binary log in the backup was larger than 16MB, the restored binary log would become corrupted, as `mysqlbackup` kept overwriting the same 16-MB file again and again with binary log contents. With this fix, the binary log is now correctly restored and has the same size as it did on the backed-up server. (Bug #20915642)

- When the `--password` option was used without an argument with the `copy-back-and-apply-log` command, `mysqlbackup` did not prompt user for a password, but either took the password from the defaults files, or took it to be an empty string when no value was specified in the defaults files. (Bug #20657939)

Index

T

V