
MySQL Enterprise Backup 4.1 Release Notes

Abstract

This document lists the changes to the MySQL Enterprise Backup 4.1 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 4.1 documentation, see the [MySQL Enterprise Backup User's Guide \(Version 4.1.4\)](#).

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: 2021-03-05 (revision: 22081)

Table of Contents

Preface and Legal Notices	1
Changes in MySQL Enterprise Backup 4.1.5 (Not yet released)	3
Changes in MySQL Enterprise Backup 4.1.4 (2020-01-24)	3
Changes in MySQL Enterprise Backup 4.1.3 (2019-02-15)	4
Changes in MySQL Enterprise Backup 4.1.2 (2018-07-27)	6
Changes in MySQL Enterprise Backup 4.1.1 (2018-04-27)	8
Changes in MySQL Enterprise Backup 4.1.0 (2017-03-03)	9
Index	11

Preface and Legal Notices

This document lists the changes to the MySQL Enterprise Backup 4.1 product, beginning with the most recent release.

Legal Notices

Copyright © 2003, 2021, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software

documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

This documentation is NOT distributed under a GPL license. Use of this documentation is subject to the following terms:

You may create a printed copy of this documentation solely for your own personal use. Conversion to other formats is allowed as long as the actual content is not altered or edited in any way. You shall not publish or distribute this documentation in any form or on any media, except if you distribute the documentation in a manner similar to how Oracle disseminates it (that is, electronically for download on a Web site with the software) or on a CD-ROM or similar medium, provided however that the documentation is disseminated together with the software on the same medium. Any other use, such as any dissemination of printed copies or use of this documentation, in whole or in part, in another publication, requires the prior written consent from an authorized representative of Oracle. Oracle and/or its affiliates reserve any and all rights to this documentation not expressly granted above.

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 4.1.5 (Not yet released)

Version 4.1.5 has no release notes, or they have not been published because the product version has not been released.

Changes in MySQL Enterprise Backup 4.1.4 (2020-01-24)

- [Functionality Added or Changed](#)
- [Bugs Fixed](#)

Functionality Added or Changed

- When `mysqlbackup` fails to connect to a server, the warning returned by `mysqlbackup` now includes the hostname and port number for TCP connections, and the socket information for socket connections. This is particularly helpful for a Group Replication setup, for which `mysqlbackup` might attempt to connect to more than one host. (Bug #30040027)
- Before the current release, when backing up a server that used the `keyring_okv` plugin for InnoDB table encryption, `mysqlbackup` must not be run by a sudo user of its operating system. This restriction has now been removed. (Bug #29020232)
- A new option, `--lock-wait-retry-count`, can now be used to specify the maximum number of retries to be attempted by `mysqlbackup` after the `FLUSH TABLES WITH READ LOCK` statement, issued during the final stage of a backup to temporarily put the database into a read-only state, fails due to a timeout. See the description of the option for details. (Bug #26008480)
- The `--uncompress` option is now supported for the `extract` operation: Files from a compressed single-file backup can now be extracted and uncompressed with a single command.

Bugs Fixed

- A backup failed for a server containing encrypted InnoDB tables when `mysqlbackup` was connected to it with `--socket` option. It was because `mysqlbackup` did not pass the value of `--keyring-migration-socket` to the keyring migration server; this fix makes `mysqlbackup` pass the option whenever it connects to the server to be backed up with a Unix socket or Windows named pipe. (Bug #30082039)
- A compressed and encrypted single-file backup was sometimes corrupted if the backed-up server was configured with `innodb_file_per_table=OFF`. (Bug #29798621)
- Backups on busy servers failed due to redo log overwrites, even when the redo log size was big. It was actually caused by a costly but unnecessary check for relay log overwrites, which has been removed by this fix. (Bug #29526337)
- When tables were altered in some ways during backup (for example, renamed, deleted and then replaced by tables of the same names, and so on), `mysqlbackup` reported success for the backup, but the backup could not be restored. This fix makes sure all those cases of table alterations are properly handled by `mysqlbackup`, so the backup can be restored. (Bug #29462340)
- A back up failed with `mysqlbackup` complaining that `Error Log scan was only able to reach...` if the server to be backed up was started with the system variable setting `innodb_checksum_algorithm=none`. (Bug #29285006)

- A restore operation failed with the error that the binary log index file could not be opened if the binary log base name for the backed-up server was a substring of the word “index”. (Bug #29273947)
- When backing up a server that used the [keyring_okv](#) plugin for InnoDB table encryption, if the `--host`, `--user`, and `--port` options were not specified with the `mysqlbackup` command via the command line or a configuration file, the backup failed. It was because in that case, `mysqlbackup` had no values for those options it could use to connect to the server that took care of keyring operations. With this fix, default values are now set, so that `mysqlbackup` connects to the server on `localhost` as `root` and on port `3306` for keyring operations when those options are not specified. (Bug #29015923)
- A restore operation for a TTS backup failed if the backed-up server has `ANSI_QUOTES` as one of its SQL modes, as specified in its system variable `sql-mode`. (Bug #28979134)
- Attempts to extract binary log files from a compressed backup failed with a `No such file or directory` error when the size of the binary log files on the backed up server was greater than 16MB. (Bug #28787312)
- After restoring an incremental backup taken from a MySQL Community Server with encrypted InnoDB tables, the keyring file of the restored server became corrupted, so the server could not be started. (Bug #28422191)
- During the InnoDB buffer pool dump in a backup operation, `mysqlbackup` sometimes reported failure for the dump while it was actually still in progress. The fix prevents the problem by improving the way `mysqlbackup` checks for the status of the dump. (Bug #27185901)
- `mysqlbackup` tried to connect to a remote host specified by the `--host` option, while it was supposed to ignore the option (see [Connection Options](#) for details). With this fix, the option is now ignored.



Note

As a side-effect of this change, on Unix-like platforms, `mysqlbackup` command that used the `--host` option now needs to use the `--protocol=TCP` option to indicate explicitly that `mysqlbackup` is to connect to `localhost` using TCP/IP.

(Bug #25911987)

- Backups failed for database containing tables created with [transparent page compression](#), with complaints of page corruption. It was because `mysqlbackup` did not perform decompression before data validation, and that has been corrected with this fix. (Bug #24366571)
- A backup failed at the step when `mysqlbackup` applied the `FLUSH TABLES tbl_name [, tbl_name] ... WITH READ LOCK` statement on all non-InnoDB tables if any table names contained reserved words or special characters. It was because `mysqlbackup` did not enclose table names in backticks when issuing the statement, and this fix makes sure that is done. (Bug #19709505, Bug #74144)

Changes in MySQL Enterprise Backup 4.1.3 (2019-02-15)

- [Functionality Added or Changed](#)
- [Bugs Fixed](#)

Functionality Added or Changed

- In addition to the requirement that the target data directory for a restore specified by the `--datadir` option must be non-existent or empty, `mysqlbackup` now enforces the same rule for the `--`

`innodb_data_home_dir`, `--innodb_log_group_home_dir`, and `--innodb_undo_directory` options (the `--force` option cannot be used to override the requirement on the three options).

Bugs Fixed

- MySQL Enterprise Backup 4.1 failed to backup a MySQL Server that has been backed up by MySQL Enterprise Backup 4.0 before. It was due to the way the `backup_history` table was queried, which has now been fixed. (Bug #29208386)
- When working with a Group Replication cluster, `mysqlbackup` might quit unexpectedly near the end of a backup operation when, in order to write to the `backup_history` table, it tried to connect with an unencrypted connection to one of the nodes on which the backup user had not logged on before. It happened when the backup user was created with the `caching_sha2_password` plugin, so that it must log on with an encrypted connection when it connected to the server for the first time; the attempt to log on thus failed, and `mysqlbackup` could not handle the failure. With this fix, at such failures, `mysqlbackup` quits gracefully with the warning that the backup operation is finished without updates to the backup history. (Bug #28893180)
- Restoring an incremental backup image using the `copy-back-and-apply-log` command failed with `mysqlbackup` complaining that the server repository configuration (including, for example, value of `innodb_data_file_path`) was unknown for the target server, unless the configuration is explicitly supplied to `mysqlbackup`. With this fix, `mysqlbackup` gets the required information from the `backup-my.cnf` file already restored with the base backup of the incremental backup. (Bug #28411028)
- A restore operation could corrupt a backup when, by mistake, a user specified the source directory to become the target directory for restoring some files (for example, specifying what was the backup's `--backup_innodb_data_home_dir` value as the restore's `--innodb_data_home_dir` value). This fix prevents the problem by having `mysqlbackup` throw an error when the command options make the source and target file paths the same for any file copying during a restore. (Bug #28376873)
- On FreeBSD platforms, using the `--show-progress` option did not make `mysqlbackup` print progress reports. (Bug #28350122)
- After encrypted InnoDB tables have been restored, sometimes the restored server could not be started, or the encrypted InnoDB tables could not be opened after the server had been started.

This fix not only resolves the aforementioned issues, but also two other problems: the failure to restore a backup containing encrypted InnoDB tables that were row-compressed, and the failure to complete a backup when an encrypted InnoDB table was created in the middle of the backup operation. (Bug #28301281)

References: See also: Bug #28360241, Bug #27168458.

- While MySQL Server interprets the system variable setting `--innodb_checksum_algorithm=0` to mean `--innodb_checksum_algorithm=crc32`, a `mysqlbackup` operation (except for backup) failed when `--innodb_checksum_algorithm=0` was set as a configuration option on the backed up server. With this fix, `mysqlbackup` now takes `--innodb_checksum_algorithm=0` as valid and interprets it as `--innodb_checksum_algorithm=crc32`. (Bug #28295519)
- The Windows version of MySQL Enterprise Backup did not display its build ID when invoked. (Bug #27916702)
- `mysqlbackup` quit unexpectedly when backing up a MySQL Server of release 4.1.2 or later for the first time if the `ALTER` privilege on the `mysql.backup_history_new` table had not been granted to the MySQL user with which `mysqlbackup` connected to the server. With this fix, `mysqlbackup` quits gracefully in the situation after throwing the proper error.

Also, the `CREATE`, `INSERT`, and `DROP` privileges on `mysql.backup_history_old` and `CREATE`, `INSERT`, `DROP`, and `ALTER` privileges on `mysql.backup_history_new` are now required only for backing up for the first time a MySQL Server that has been upgraded from 5.7.22 or earlier and has been backed up by MySQL Enterprise Backup before. (Bug #27879530, Bug #28546256)

- When the `--show-progress=table` option was used, `mysqlbackup` gave a warning in the error log on an aborted connection to the server near the end of the operation. It was because the connection to the server for writing to the `backup_progress` table had remained open. With this fix, the connection is properly closed after the `mysqlbackup` operation is finished. (Bug #27647283)
- A restore operation for an incremental backup failed when individual tablespaces with relative file paths were involved. (Bug #26442994)
- When the option `--only-innodb-with-frm` or `--no-locking` was used during a backup operation, the backup sometimes failed with `mysqlbackup` complaining that the highest LSN was larger in a copied page than on the backed-up server. It was because `mysqlbackup` did not perform a log flushing before copying the redo log when either of the mentioned options was used. With this fix, log flushing was always performed to prevent the error. (Bug #25412655)
- Partial backups sometimes failed because full-text index files had their file names matched by the regular expression provided by the `--include-tables` option, and the files were then handled as ordinary tablespace files by `mysqlbackup`. With this fix, `mysqlbackup` excludes any full-text index files from backups. (Bug #25044900)
- If, when a backup was in progress and `mysqlbackup` was reading the binary log (or the relay log) index file and the server tried to modify the index file (because, for example, a log flush or log purge just took place), the binary logging (or relay logging) stopped; the server also quit unexpectedly on Windows platforms. It was because `mysqlbackup` did not handle well the file sharing violation. With this fix, `mysqlbackup` now reads the index file using the local file system API, which handles the file sharing violation gracefully; also, `mysqlbackup` now copies the index file into its buffer and then closes it, instead of keeping it open for long, so the server can modify or delete the index file more easily. (Bug #22914974, Bug #26047119)

Changes in MySQL Enterprise Backup 4.1.2 (2018-07-27)

- [Functionality Added or Changed](#)
- [Bugs Fixed](#)

Functionality Added or Changed

- **Important Change:** Starting from release 4.1.2, the storage engine of the `mysql.backup_history` table on a backed-up server has switched from CSV to InnoDB, and a new column for server UUIDs has been added to the table. See [Backup History Table Update](#) for the new user privileges required by `mysqlbackup` due to this change.
- **Important Change:** When working with a [Group Replication](#) setup, `mysqlbackup` now makes the backup history available to all members of the server group by making sure that the `backup_history` table is updated on a primary node after each `mysqlbackup` operation. See [Using MySQL Enterprise Backup with Group Replication](#) for details.

With the implementation of this feature, the new user privilege of `SELECT` on `performance_schema.replication_group_members` is now required by `mysqlbackup` to work with any server, even when it does not belong to a Group Replication setup. See [Grant MySQL Privileges to Backup Administrator](#) for details.

- The relay log for a backed-up slave server, instead of being restored always to the data directory on the target slave server, is now restored by default to the same location it was found on the backed-up slave server. It can also be restored to a different location specified with the new `--relay-log` option. (Bug #25141738, Bug #83927)
- The binary log for a backed-up server, instead of being restored always to the data directory on the target server, is now restored by default to the same location it was found on the backed-up server. It can also be restored to a different location specified with the new `--log-bin` option. (Bug #25141738, Bug #83927)
- The `backup_history` table now includes a `server_uuid` column, which stores the value of the `server_uuid` of the backed up server.
- OAuth is now supported for Oracle Cloud Storage client authentication. Two new options, `--cloud-storage-url` and `--cloud-oauth-token`, have been introduced for the purpose. See [Cloud Storage Options](#) for details.

Bugs Fixed

- During an `--apply-incremental-backup` operation, `mysqlbackup` attempted to delete the binary log of the backed-up, running server. (Bug #28377502)
- A `mysqlbackup` operation on an image stored on an OpenStack cloud storage service sometimes failed with a segmentation fault or a bad URL error. It was because of a race condition caused by an uninitiated variable, which has been eliminated by this fix. (Bug #28189239, Bug #28183729)
- Backups failed when the `--log-bin` option on the backed-up server contained a full-path pointing to a location under the server's data directory. (Bug #28044530)
- A `mysqlbackup` operation on an image stored on an OpenStack cloud storage service failed with a `401 Unauthorized` error when the operation took a long time and the authentication token for the cloud access expired. With this fix, a separate thread in `mysqlbackup` requests a new token from the OpenStack cloud service in that situation, so that the operation can continue. (Bug #27893174)
- The maximum value that could be set for the `--safe-slave-backup-timeout` option was 2700 (seconds), which automatically replaced any larger value. With this fix, there is no longer a maximum limit, even though a high value is not recommended; see the description of `--safe-slave-backup-timeout` for details. (Bug #27883020)
- When `mysqlbackup` performed sanity checks on InnoDB tablespaces and found a space ID mismatch for an FSP header and a page header, the name of the problematic tablespace was not given in the error report. (Bug #27752703)
- `mysqlbackup` issued a warning whenever the number of files specified in the system variable `innodb_data_file_path` of the server to be backed up exceeded 100. With this fix, a warning is issued only if the number of InnoDB data files to be opened exceeds the number specified by the system variable `innodb_open_files`. (Bug #27701402)
- An `apply-incremental-backup` operation failed when individual tablespaces with relative file paths were involved. (Bug #27278876)
- Backups created with the `---skip-unused-pages` option failed validation by the `mysqlbackup validate` command. (Bug #27102697)
- `mysqlbackup` quit unexpectedly when the `--disable-manifest` option was used in a backup operation. (Bug #26861475)

- A restore operation for an incremental backup failed when individual tablespaces with relative file paths were involved. (Bug #26442994)

Changes in MySQL Enterprise Backup 4.1.1 (2018-04-27)

- [Functionality Added or Changed](#)
- [Bugs Fixed](#)

Functionality Added or Changed

- *For MySQL Server 5.7.21 and later:* Servers' use of the [keyring_encrypted_file](#) and [keyring_aws](#) plugins is now supported. See [Working with Encrypted InnoDB Tables](#) for details. (Bug #27127898)
- Information on the executed GTIDs is now included in the [mysqlbackup](#) output and the backup log when the backed-up server has GTIDs enabled. (Bug #25978803)
- The buffer size for cloud transfers can now be specified using the new `--cloud-buffer-size` option. See [Cloud Storage Options](#) for details.
- HTTP Basic Authentication and non-chunked transfer are now supported for backup and restore using OpenStack Swift-compatible object storage services. Two new options, `--cloud-basicauth-url` and `--cloud-chunked-transfer`, have been introduced for these purposes. See [Cloud Storage Options](#) for details.

Bugs Fixed

- [mysqlbackup](#) failed to restore a TTS (transportable tablespace) backup if the row format of any TTS tables did not match the InnoDB default row format on the target server. This was because the row format of a table was not included in the backup if the table was created with the InnoDB default row format. With this fix, the row format is now always written to the backup, so that tables are always restored with the right row format. (Bug #27445546)
- [mysqlbackup](#) failed to backup to an Amazon S3 cloud storage. (Bug #27231229)
- [mysqlbackup](#) could not establish a connection to the server with a Unix socket using the option `--protocol=SOCKET`. (Bug #26977679)
- [mysqlbackup](#) did not indicate in its output whether an established connection with the server was encrypted or not. With this fix, the crucial information is now provided. (Bug #26919023)
- An `apply-log` operation failed when more than one process threads were used for the operation and the backup contained encrypted InnoDB tables. (Bug #26495013)
- On macOS, [mysqlbackup](#) failed to determine the relay log file name correctly if the relay log index file name was not specified. (Bug #25574605)
- [mysqlbackup](#) only accepts values for `--ssl-mode` in upper case. With this fix, the option's value has become case insensitive. (Bug #25548088)
- After restoring an incremental backup of a slave server, the file `ibbackup_slave_info` was missing from the `meta` folder under the backup directory on the target server. (Bug #25097753)
- When a compressed backup was restored with the `--innodb_data_home_dir` option pointing to a restore location outside of the data directory, `.ibd` files were still being copied into the data directory, causing an exception to be thrown at the attempt to start the restored server. (Bug #24826986)

Changes in MySQL Enterprise Backup 4.1.0 (2017-03-03)

- [Security Notes](#)
- [Functionality Added or Changed](#)
- [Bugs Fixed](#)

Security Notes

- The linked OpenSSL library for MySQL Enterprise Backup 4.1 has been updated to version 1.0.2k. (Bug #25615449)

Functionality Added or Changed

- A backup became corrupted if, during the backup process, a DDL operation that took advantage of MySQL server's [online DDL feature](#) occurred. This was because `mysqlbackup` did not support the server feature—and it still does not. This fix avoids the error by having `mysqlbackup` turn the server's system variable `old_alter_table` to “1” at the beginning of a backup if it is “0,” so that any DDL operations that take place during the backup are handled with the old table copy method. `mysqlbackup` then turns the variable back to “0” near the end of the backup operation.



Important

Notice that in cases where `mysqlbackup` quits unexpectedly and does not turn `old_alter_table` back to its original value, the user will have to turn the value back to “0” manually on the server, in order to return the server to its original configuration. This should be performed if the statement “Server system variable 'old_alter_table' was set to '0'. Setting it to '1'” appears in the early output of `mysqlbackup`, but the statement “Setting server system variable 'old_alter_table' back to '0'” does not appear before `mysqlbackup` quits.

(Bug #25217215)

- MySQL Enterprise Backup now supports the `--ssl-mode` option, which enables you to specify the security state of the connection to the server. It replaces the client side `--ssl` and `--ssl-verify-server-cert` options, which are now deprecated. See the description of the `--ssl-mode` option in [MySQL 5.7 Reference Manual](#) for details. (Bug #23508228)
- A new option, `--skip-final-rescan`, makes `mysqlbackup` skip the final rescan for InnoDB tables that are modified by DDL operations after the database has been locked near the end of a backup operation. This potentially shortens the duration for the lock and reduces the backup's impact on the server's normal operation. See the description for `--skip-final-rescan` for details. (Bug #21094221)
- The output by `mysqlbackup`, which goes to the `stderr` stream and the message log, has now been improved to include the timestamps and thread IDs for all steps taken by `mysqlbackup`, in order to provide more information for debugging purposes. (Bug #20142619)
- During the final stage of a backup when MySQL Enterprise Backup tried to temporarily put the database into a read-only state using the `FLUSH TABLES WITH READ LOCK` statement in order to copy non-InnoDB files, if a long query was running on the server at the same time, the `FLUSH TABLES WITH READ LOCK` statement could be taking too long to finish, holding up further queries and eventually bringing down the server.

A new `mysqlbackup` option `--lock-wait-timeout` can now be used to specify the timeout in seconds for the `FLUSH TABLES WITH READ LOCK` statement. If the timeout is exceeded, the

statement is failed and the lock on the tables is released, so that queries held up by the lock can then be executed. `mysqlbackup` then retries the statement and continues with the backup. Default value for `--lock-wait-timeout` is 60 [seconds]. (Bug #14339483)

- The `backup_history` table now includes the following new columns:
 - `start_time_utc`
 - `end_time_utc`
 - `consistency_time_utc`
 - `meb_version`
- A full set of exit codes have now been implemented for MySQL Enterprise Backup. Also, a new `mysqlbackup` command, `print-message`, returns an exit message for any given exit code supplied with the new option `--error-code`. See [Exit codes of MySQL Enterprise Backup](#) for details.
- To increase the performance for hot backups, `mysqlbackup` now shortens the final phase of the backups by resizing the MyISAM key cache before it locks the database with a `FLUSH TABLES WITH READ LOCK` statement. The resize triggers a flush of the MyISAM key cache, which reduces the time it takes to run the `FLUSH TABLES WITH READ LOCK` statement. The MyISAM key cache size is changed back to its original value afterward.
- Apply-log operations can now be performed with multiple worker threads in parallel, which can improve performance for the operations. The number of threads to be used can be specified with the `--process-threads` option.
- The copying of redo log files into backups has been made faster, shortening the overall backup time in some cases and making it less likely that a backup fails because a redo log file has been overwritten before it is copied.
- MySQL Enterprise Backup now supports *optimistic incremental backup*, in which `mysqlbackup` scans only those InnoDB data files that have been modified since the last backup for changed pages and then saves them into the incremental backup. It potentially makes incremental backups faster, and is performed by specifying `--incremental=optimistic`. See [Full-scan versus Optimistic Incremental Backup](#) for details.
- In order to minimize the impact of a hot backup on the MySQL server, the copying of the buffer pool dump files and some of the metadata files is now performed before the final phase of the backup in which the server instance is locked. This shortens the duration for the lock and reduces the backup's impact on the server's normal operation.

Also, to minimize the resource used on a backup, the copying of the buffer pool dump files is no longer performed for partial and offline backups, for which the buffer pool dump is usually not very useful.

Bugs Fixed

- The [Release](#) number for the RPM packages of MySQL Enterprise Backup always said “1,” instead of giving the release number of the Linux distribution for which the package was built. (Bug #25538798)
- An `apply-log` or `apply-incremental-backup` operation failed when, during the time the backup was created, any tables have been dropped and then recreated with the same name for more than once. (Bug #25334929)
- When creating a full image backup with the `--no-locking` option, `mysqlbackup` failed to write the binary log information to the backup history table and the `backup_variables.txt` file. The result was

that when creating an incremental image backup based on the full backup, the attempt to copy the binary log files from the server into the incremental image backup (which is the default behavior) would fail, causing the incremental backup to stop. With this fix, the binary log information is no longer missing after the full backup, so the incremental image backup no longer fails." (Bug #25296324)

References: See also: Bug #19915713.

- A backup operation sometimes failed due to corruption of the copied redo log, and the corruption occurred because checksum validation for redo logs was not activated for `mysqlbackup`. This fix restores the checksum function, so that the validation is performed when `innodb_log_checksums=ON` on the backed up server. When a chunk of redo log data fails the validation, `mysqlbackup` tries to reread the data from the server.

However, redo log checksum validation is not performed for offline backups, and online backups fail when the value of `innodb_log_checksums` is switched off on the server when the backup is in progress. (Bug #25057394)

- `mysqlbackup` logged double entries with wrong information into the `backup_history` table for optimistic backups. (Bug #24502876)
- Backups for a slave server in a multi-source replication setup using the `--slave-info` option failed. (Bug #24444950)

References: See also: Bug #21830316.

- If a backed up server was configured with `innodb_undo_tablespaces` being nonzero, during a backup, `mysqlbackup` emitted meaningless warning messages about the tablespaces already existing in the cache. This fix removes those messages. (Bug #24400230)
- During an `apply-log` or a `copy-back-and-apply-log` operation on a backup with encrypted InnoDB tables, if no value was specified for the `--encrypt-password` option used, the operation failed with the complaint that no encryption password was specified. With this fix, `mysqlbackup` asks the user for the encryption password when it is in the same situation. (Bug #24364442)
- An `apply-log` operation failed for a backup of a server that contained tablespaces outside of its data directory. It was because those tablespaces, unlike tablespaces in the data directory, were not loaded by `mysqlbackup` before the `apply-log` operation started. This fix ensures all tablespaces are loaded. (Bug #22026145)
- When running the `backup-and-apply-log` command without a connection to the MySQL server, `mysqlbackup` could not know the correct binary log file name and binary log position for the backup; yet, at the end of the `backup-and-apply-log` operation, `mysqlbackup` still printed out some values for the binary log file name and position, which were random in nature. (Bug #21623089)

Index

Symbols

`--disable-manifest`, 6
`--include-tables`, 4
`--innodb_checksum_algorithm`, 4
`--lock-wait-retry-count`, 3
`--lock-wait-timeout`, 3
`--no-locking`, 4, 9
`--only-innodb-with-frm`, 4
`--show-progress`, 4

--skip-unused-pages, 6
--uncompress, 3

A

altered tables, 3
Amazon S3, 8
ANSI_QUOTES, 3
apply-incremental-backup, 6
apply-log, 8

B

backticks, 3
backup_history table, 9
binary log, 3, 4, 6, 9

C

cloud operations, 6
cloud-basicauth-url, 8
cloud-buffer-size, 8
cloud-chunked-transfer, 8
cloud-oauth-token, 6
cloud-storage-url, 6
compressed backup, 3
compression, 8

E

encrypted backup, 3
encrypted InnoDB tables, 3
extract, 3

F

failed connection, 3
FLSUH TABLES, 3

G

Group Replication, 4
GTID, 8

I

ibbackup_slave_info, 8
Important Change, 6
incremental backup, 4, 6, 8
InnoDB cluster, 4
innodb_checksum_algorithm, 3
innodb_data_home, 4
innodb_file_per_table, 3
innodb_log_group_home_dir, 4
innodb_open_files, 6
innodb_undo_directory, 4
innodb_data_file_path, 4
innodb_data_file_path, 6
innodb_data_home_dir, 8

K

keyring_okv, 3

L

log-bin, 6

O

OAuth, 6

online DDL, 9

OpenSSL, 9

OpenStack, 6

P

privileges, 4

R

redo log overwrite, 3

relay log, 4, 6

S

server_uuid, 6

sql_mode, 3

ssl-mode, 8

T

tablespace ID check, 6

transparent page compression, 3

V

validate, 6

W

warnings, 3

