MySQL NDB Cluster 8.4 Release Notes

Abstract

This document contains release notes for the changes in each release of MySQL NDB Cluster that uses version 8.4 of the `NDB` (`NDBCLUSTER`) storage engine.

Each NDB Cluster 8.4 release is based on a mainline MySQL Server release and a particular version of the `NDB` storage engine, as shown in the version string returned by executing `SELECT VERSION()` in the `mysql` client, or by executing the `ndb_mgm` client `SHOW` or `STATUS` command; for more information, see MySQL NDB Cluster 8.4.

For general information about features added in NDB Cluster 8.4, see What is New in MySQL NDB Cluster 8.4. For a complete list of all bug fixes and feature changes in MySQL NDB Cluster, please refer to the changelog section for each individual NDB Cluster release.

For additional MySQL 8.4 documentation, see the MySQL 8.4 Reference Manual, which includes an overview of features added in MySQL 8.4 that are not specific to NDB Cluster (What Is New in MySQL 8.4 since MySQL 8.0), and discussion of upgrade issues that you may encounter for upgrades from MySQL 8.3 to MySQL 8.4 (Changes in MySQL 8.4). For a complete list of all bug fixes and feature changes made in MySQL 8.4 that are not specific to `NDB`, see MySQL 8.4 Release Notes.

Updates to these notes occur as new product features are added, so that everybody can follow the development process. If a recent version is listed here that you cannot find on the download page (https://dev.mysql.com/downloads/), the version has not yet been released.

The documentation included in source and binary distributions may not be fully up to date with respect to release note entries because integration of the documentation occurs at release build time. For the most up-to-date release notes, please refer to the online documentation instead.

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: 2024-07-30 (revision: 28793)

Table of Contents

Preface and Legal Notices

This document contains release notes for the changes in each release of MySQL NDB Cluster that uses version 8.4 of the `NDB` storage engine.

Legal Notices

Copyright © 1997, 2024, Oracle and/or its affiliates.

License Restrictions

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

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at
http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support for Accessibility

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit
http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Changes in MySQL NDB Cluster 8.4.2 (2024-07-23, LTS Release)

MySQL NDB Cluster 8.4.2 is a new LTS release of NDB 8.4, based on MySQL Server 8.4 and including features in version 8.4 of the `NDB` storage engine, as well as fixing recently discovered bugs in previous NDB Cluster releases.

Obtaining MySQL NDB Cluster 8.4.   NDB Cluster 8.4 source code and binaries can be obtained from https://dev.mysql.com/downloads/cluster/.

For an overview of major changes made in NDB Cluster 8.4, see What is New in MySQL NDB Cluster 8.4.

This release also incorporates all bug fixes and changes made in previous NDB Cluster releases, as well as all bug fixes and feature changes which were added in mainline MySQL 8.4 through MySQL 8.4.2 (see Changes in MySQL 8.4.2 (2024-07-23, LTS Release)).

This release contains no functional changes specific to MySQL NDB Cluster, and is published to align with and include changes made in MySQL Server 8.4.2.

Changes in MySQL NDB Cluster 8.4.1 (2024-07-02, LTS Release)

MySQL NDB Cluster 8.4.1 is a new LTS release of NDB 8.4, based on MySQL Server 8.4 and including features in version 8.4 of the `NDB` storage engine, as well as fixing recently discovered bugs in previous NDB Cluster releases.

Obtaining MySQL NDB Cluster 8.4.   NDB Cluster 8.4 source code and binaries can be obtained from https://dev.mysql.com/downloads/cluster/.

For an overview of major changes made in NDB Cluster 8.4, see What is New in MySQL NDB Cluster 8.4.

This release also incorporates all bug fixes and changes made in previous NDB Cluster releases, as well as all bug fixes and feature changes which were added in mainline MySQL 8.4 through MySQL 8.4.1 (see Changes in MySQL 8.4.1 (2024-07-01, LTS Release)).

> ⚠ **Important**
>
> This release is no longer available for download. It was removed due to a critical issue that could stop the server from restarting following the creation of a very large number of tables (8001, or more). Please upgrade to MySQL Cluster 8.4.2 instead.

- Functionality Added or Changed

- Bugs Fixed

Functionality Added or Changed

- **Important Change:** Now, when the removal of a data node file or directory fails with a file does not exist (`ENOENT`) error, this is treated as a successful removal.

- **ndbinfo Information Database:** Added a `type` column to the `transporter_details` table in the `ndbinfo` information database. This column shows the type of connection used by the transporter, which is either of `TCP` or `SHM`.

- **NDB Client Programs:** Added the `--CA-days` option to `ndb_sign_keys` to make it possible to specify a certificate's lifetime. (Bug #36549567)

- **NDB Client Programs:** When started, `ndbd` now produces a warning in the data node log like this one:

```
2024-05-28 13:32:16 [ndbd] WARNING  -- Running ndbd with a single thread of
signal execution.  For multi-threaded signal execution run the ndbmtd binary.
```

  (Bug #36326896)

Bugs Fixed

- **NDB Replication:** When subscribing to changes in the `mysql.ndb_apply_status` table, different settings were used depending on whether `ndb_log_apply_status` was `ON` or `OFF`. Since `ndb_log_apply_status` can be changed at runtime and subscriptions are not recreated at that time, changing these settings at runtime did not have the desired effect.

  The difference between enabling `ndb_log_apply_status` dynamically at runtime and doing so from the start of the MySQL proccess was in the format used when writing the ndb_apply_status updates to the binary log. When `ndb_log_apply_status` was enabled at runtime, writes were still done using the `UPDATE` format when `WRITE` was intended.

  To fix this inconsistency and make the behavior more distinct, we now always use `WRITE` format in such cases; using the `WRITE` format also makes the binary log image slightly smaller and is thus preferred. In addition, the cleanup of old events has been improved, which improves the cleanup of failed attempts to create tables and events. (Bug #36453684)

- **NDB Replication:** The binary log index purge callback was skipped for the replica applier, which caused orphan rows to be left behind in the `ndb_binlog_index` table. (Bug #20573020, Bug #35847745, Bug #36378551, Bug #36420628, Bug #36423593, Bug #36485220, Bug #36492736)

- **NDB Cluster APIs:** It was possible to employ the following NDB API methods without them being used as `const`, although this alternative usage had long been deprecated (and was not actually documented):

  - `Dictionary::listEvents()`

  - `Dictionary::listIndexes()`

  - `Dictionary::listObjects()`

- `NdbOperation::getNdbErrorLine()`

Now, each of these methods must always be invoked as `const`. (Bug #36165876)

- **NDB Client Programs:** `ndb_redo_log_reader` could not read data from encrypted files. (Bug #36313482)

- **NDB Client Programs:** `ndb_redo_log_reader` exited with `Record type = 0 not implemented` when reaching an unused page, all zero bytes, or a page which was only partially used (typically a page consisting of the page header only). (Bug #36313259)

- **NDB Client Programs:** `ndb_restore` did not restore a foreign key whose columns differed in order from those of the parent key.

  Our thanks to Axel Svensson for the contribution. (Bug #114147, Bug #36345882)

- The destructor for `NDB_SCHEMA_OBJECT` makes several assertions about the state of the schema object, but the state was protected by a mutex, and the destructor did not acquire this mutex before testing the state.

  We fix this by acquiring the mutex within the destructor. (Bug #36568964)

- `NDB` now writes a message to the MySQL server log before and after logging an incident in the binary log. (Bug #36548269)

- Removed a memory leak in `/util/NodeCertificate.cpp`. (Bug #36537931)

- Removed a memory leak from `src/ndbapi/NdbDictionaryImpl.cpp`. (Bug #36532102)

- The internal method `CertLifetime::set_set_cert_lifetime(X509 *cert)` should set the not-before and not-after times in the certificate to the same as those stored in the `CertLifetime` object, but instead it set the not-before time to the current time, and the not-after time to be of the same duration as the object. (Bug #36514834)

- Removed a possible use-after-free warning in `ConfigObject::copy_current()`. (Bug #36497108)

- When a thread acquires and releases the global schema lock required for schema changes and reads, the associated log message did not identify who performed the operation.

  To fix this issue, we now do the following:

  - Prepend the message in the log with the identification of the NDB Cluster component or user session responsible.

  - Provide information about the related Performance Schema thread so that it can be traced.

  (Bug #36446730)

  References: See also: Bug #36446604.

- Metadata changes were not logged with their associated thread IDs. (Bug #36446604)

  References: See also: Bug #36446730.

- When building `NDB` using `lld`, the build terminated prematurely with the error message `ld.lld: error: version script assignment of 'local' to symbol 'my_init' failed: symbol not defined` while attempting to link `libndbclient.so`. (Bug #36431274)

- TLS did not fail cleanly on systems which used OpenSSL 1.0, which is unsupported. Now in such cases, users get a clear error message advising that an upgrade to OpenSSL 1.1 or later is required to use TLS with NDB Cluster. (Bug #36426461)

- The included `libxml2` library was updated to version 2.9.13. (Bug #36417013)

- NDB Cluster's pushdown join functionality expects pushed conditions to filter exactly, so that no rows that do not match the condition must be returned, and all rows that do match the condition must returned. When the condition contained a BINARY value compared to a `BINARY` column this was not always true; if the value was shorter than the column size, it could compare as equal to a column value despite having different lengths, if the condition was pushed down to `NDB`.

  Now, when deciding whether a condition is pushable, we also make sure that the `BINARY` value length exactly matches the `BINARY` column's size. In addition, when binary string values were used in conditions with `BINARY` or `VARBINARY` columns, the actual length of a given string value was not used but rather an overestimate of its length. This is now changed; this should allow more conditions comparing short string values with `VARBINARY` columns to be pushed down than before this fix was made. (Bug #36390313, Bug #36513270)

  References: See also: Bug #36399759, Bug #36400256. This issue is a regression of: Bug #36364619.

- Setting `AutomaticThreadConfig` and `NumCPUs` when running single-threaded data nodes (`ndbd`) sometimes led to unrecoverable errors. Now `ndbd` ignores settings for these parameters, which are intended to apply only to multi-threaded data nodes (`ndbmtd`). (Bug #36388981)

- Improved the error message returned when trying to add a primary key to an `NDBCLUSTER` table using `ALGORITHM=INPLACE`. (Bug #36382071)

  References: See also: Bug #30766579.

- The handling of the `LQH` operation pool which occurs as part of TC takeover skipped the last element in either of the underlying physical pools (static or dynamic). If this element was in use, holding an operation record for a transaction belonging to a transaction coordinator on the failed node, it was not returned, resulting in an incomplete takeover which sometimes left operations behind. Such operations interfered with subsequent transactions and the copying process (`CopyFrag`) used by the failed node to recover.

  To fix this problem, we avoid skipping the final record while iterating through the `LQH` operation records during TC takeover. (Bug #36363119)

- The `libssh` library was updated to version 0.10.4. (Bug #36135621)

- When distribution awareness was not in use, the cluster tended to choose the same data node as the transaction coordinator repeatedly. (Bug #35840020, Bug #36554026)

- In certain cases, management nodes were unable to allocate node IDs to restarted data and SQL nodes. (Bug #35658072)

- Setting `ODirect` in the cluster's configuration caused excess logging when verifying that `ODirect` was actually settable for all paths. (Bug #34754817)

- In some cases, when trying to perform an online add index operation on an `NDB` table with no explicit primary key (see Limitations of NDB online operations), the resulting error message did not make the nature of the problem clear. (Bug #30766579)

  References: See also: Bug #36382071.

Changes in MySQL NDB Cluster 8.4.0 (2024-04-30, LTS Release)

MySQL NDB Cluster 8.4.0 is a new development release of NDB 8.4, based on MySQL Server 8.4 and including features in version 8.4 of the `NDB` storage engine, as well as fixing recently discovered bugs in previous NDB Cluster releases.

Obtaining MySQL NDB Cluster 8.4. NDB Cluster 8.4 source code and binaries can be obtained from https://dev.mysql.com/downloads/cluster/.

For an overview of major changes made in NDB Cluster 8.4, see What is New in MySQL NDB Cluster 8.4.

This release also incorporates all bug fixes and changes made in previous NDB Cluster releases, as well as all bug fixes and feature changes which were added in mainline MySQL 8.4 through MySQL 8.4.0 (see Changes in MySQL 8.4.0 (2024-04-30, LTS Release)).

- Deprecation and Removal Notes

- ndbinfo Information Database

- Functionality Added or Changed

- Bugs Fixed

Deprecation and Removal Notes

- **Packaging; Linux:** Removed the deprecated tool `/usr/bin/pathfix.py` from packages for Fedora 39. (Bug #35997178)

- The unused `INFORMATION_SCHEMA.TABLESPACES` table, deprecated in MySQL 8.0.22, has now been removed.

  The Information Schema `FILES` table provides tablespace-related information for `NDB` tables. (WL #14065)

ndbinfo Information Database

- The `ndbinfo transporter_details` table, introduced in NDB 8.0, provides information about individual transporters used in an NDB Cluster, rather than aggregate data as shown by the `transporters` table.

  This release adds the following columns to `transporter_details`:

  - `sendbuffer_used_bytes`: Number of bytes of signal data currently stored pending send using this transporter.

  - `sendbuffer_max_used_bytes`: Historical maximum number of bytes of signal data stored pending send using this transporter. Reset when the transporter connects.

  - `sendbuffer_alloc_bytes`: Number of bytes of send buffer currently allocated to store pending send bytes for this transporter. Send buffer memory is allocated in large blocks which may be sparsely used.

  - `sendbuffer_max_alloc_bytes`: Historical maximum number of bytes of send buffer allocated to store pending send bytes for this transporter.

  for more information, see The ndbinfo transporter_details Table. . (WL #7662)

Functionality Added or Changed

- **Packaging:** Added support for Fedora 40 and Ubuntu 24.04.

- **NDB Replication:** Previously, when SQL nodes performing binary logging had `log_replica_updates=OFF`, replicated updates applied on a replica NDB cluster were still sent to the SQL nodes performing binary logging. Such updates, as well as any updates that do not trigger logging, are no longer sent, in order to decrease network traffic and resource consumption. (WL #15407)

- **ndbinfo Information Database:** Added the `transporter_details` table to the `ndbinfo` information database. This table is similar to the `transporters` table, but provides information about individual transporters rather than in the aggregate.

For more information, see The ndbinfo transporter_details Table. (Bug #113163, Bug #36031560)

- **NDB Client Programs:** Added the `--verbose` option to the `ndb_waiter` test program to control the verbosity level of the output. (Bug #34547034)

- Improved logging related to purging of the binary log, including start and completions times, and whether it is the injector which has initiated the purge. (Bug #36176983)

Bugs Fixed

- **NDB Replication:** Replication of an `NDB` table stopped under the following conditions:

  - The table had no explicit primary key

  - The table contained `BIT` columns

  - A hash scan was used to find the rows to be updated or deleted

  To fix this issue, we now make sure that the hash keys for the table match on the source and the replica. (Bug #34199339)

- **NDB Cluster APIs:** TLS connection errors were printed even though TLS was not specified for connections.

  To fix this issue, following an ignored TLS error, we explicitly reset the error condition in the management handle to **NO_ERROR**. (Bug #36354973)

- **NDB Cluster APIs:** The `NdbEventOperation` methods `hasError()` and `clearError()`, long deprecated, are effectively disabled: `hasError()` now returns a constant 0, and `clearError()` does nothing. To determine an event type, use `getEventType2()` instead.

- **NDB Client Programs:** In some cases, it was not possible to load cerificates generated using `ndb_sign_keys`. (Bug #36430004)

- **NDB Client Programs:** The following command-line options did not function correctly for the `ndb_redo_log_reader` utility program:

  - `--mbyte`

  - `--page`

  - `--pageindex`

  (Bug #36313427)

- **NDB Client Programs:** A certificate lifetime generated by `ndb_sign_keys` should consist of a fixed number of days, plus a random amount of extra time provided by the OpenSSL function `RAND_bytes()`, casting the result to a signed integer value. Because this value could sometimes be negative, this led to extra time being subtracted rather than added.

  We eliminate this problem by using an unsigned integer type to hold the value obtained from `RAND_bytes()`. (Bug #36270629)

- **NDB Client Programs:** Invoking ndb_mgmd with the --bind-address option could in some cases cause the program to terminate unexpectedly. (Bug #36263410)

- **NDB Client Programs:** Some `NDB` utilities such `ndb_show_tables` leaked memory from API connections when TLS was required by the data nodes, and with valid certificates. (Bug #36170703)

- **NDB Client Programs:** Work begun in NDB 8.0.18 and 8.0.20 to remove the unnecessary text `NDBT_ProgramExit ...` from the output of NDB programs is completed in this release. This message should no longer appear in the release binaries of any such programs. (Bug #36169823)

References: See also: Bug #27096741.

- **NDB Client Programs:** The output from `ndb_waiter --ndb-tls-search-path` was not correctly formatted. (Bug #36132430)

- **NDB Client Programs:** On Windows hosts, `ndb_sign_keys` could not locate the `ssh` program. (Bug #36053948)

- **NDB Client Programs:** `ndb_sign_keys` did not handle the `--CA-tool` option correctly on Windows. (Bug #36053908)

- **NDB Client Programs:** The use of a strict 80-character limit for `clang-format` on the file `CommandInterpreter.cpp` broke the formatting of the interactive help text in the `NDB` management client. (Bug #36034395)

- **NDB Client Programs:** Trying to start `ndb_mgmd` with `--bind-address=localhost` failed with the error `Illegal bind address`, which was returned from the MGM API when attempting to parse the bind adress to split it into host and port parts. `localhost` is now accepted as a valid address in such cases. (Bug #36005903)

- The included `libexpat` library was updated to version 2.5.0. (Bug #36324146)

- An implicit rollback generated when refusing to discover a table in an ongoing transaction caused the entire transaction to roll back. This could happen when a table definition changed while a transaction was active. We also checked at such times to see whether the table already existed in the data dictionary, which also meant that a subsequent read from same table within the same transaction would (wrongly) allow discovery.

  Now in such cases, we skip checking whether or not a given table already exists in the data dictionary; instead, we now always refuse discovery of a table that is altered while a transaction is ongoing and return an error to the user. (Bug #36191370)

- When a backup was restored using `ndb_restore` with `--disable-indexes` and `--restore-privilege-tables`, the ordered index of the primary key was lost on the `mysql.ndb_sql_metadata` table, and could not be rebuilt even with `--rebuild-indexes`. (Bug #36157626)

- `NDB` maintains both a local and a global pool of free send buffers. When send buffers cannot be allocated from the local pool NDB allocates one from the global pool; likewise, buffers are freed and returned to the global pool when the local pool has too many free buffers. Both of these allocations require a mutex to be locked.

  In order to reduce contention on this global mutex, we attempt to over-allocate buffers from the global pool when needed, keeping the excess buffers in the local pool, when releasing excess buffers to the global pool this was done only to the limit determined by `max_free`. After having released to the global pool, such that the `max_free` limit was met, it was likely that additional buffers would soon be released, once again exceeding `max_free`. This caused extra contention on the global pool mutex.

  To address this issue, we now reduce the free buffers to 2/3 of the `max_free` limit in such cases. (Bug #36108639)

- `SSL_pending()` data from an SSL-enabled `NdbSocket` was not adequately checked for. (Bug #36076879)

- In certain cases, `ndb_mgmd` hung when attempting to sending a stop signal to `ndbmtd`. (Bug #36066725)

- Starting a replica to apply changes when NDB was not yet ready or had no yet started led to an unhelpful error message (`Fatal error: Failed to run 'applier_start' hook`). This happened when the replica started and the applier start hook waited for the number of seconds

specified by `--ndb-wait-setup` for NDB to become ready; if it was not ready by then, the start hook reported the failure. Now in such cases, we let processing continue, instead, and allow the error to be returned from NDB, which better indicates its true source. (Bug #36054134)

- A `mysqld` process took much longer than expected to shut down when all data nodes were unreachable. (Bug #36052113)

- Negated the need for handling in the `NDB` binary log injector thread for a failure to instantiate an injector transaction by removing a potential point of failure in that operation. (Bug #36048889)

- It was possible in certain cases for the `TRPMAN` block to operate on transporters outside its own receive thread. (Bug #36028782)

- Removed a possible race condition between `start_clients_thread()` and `update_connections()`, due to both of these seeing the same transporter in the `DISCONNECTING` state. Now we make sure that disconnection is in fact completed before we set indicating that that the transporter has disconnected, so that `update_connections()` cannot close the `NdbSocket` before it has been completely shut down. (Bug #36009860)

- When a transporter was overloaded, the send thread did not yield to the CPU as expected, instead retrying the transporter repeatedly until reaching the hard-coded 200 microsecond timeout. (Bug #36004838)

- A MySQL server disconnected from schema distribution was unable to set up event operations because the table columns could not be found in the event. This could be made to happen by using `ndb_drop_table` or another means to drop a table directly from `NDB` that had been created using the MySQL server.

  We fix this by making sure in such cases that we properly invalidate the `NDB` table definition from the dictionary cache. (Bug #35948153)

- The `ndb_sign_keys` utility's `--remote-openssl` option did not function as expected. (Bug #35853405)

- A replica could not apply a row change while handling a `Table definition changed` error. Now any such error is handled as a temporary error which can be retried multiple times. (Bug #35826145)

- Repeated incomplete incomplete attempts to perform a system restart in some cases left the cluster in a state from which it could not recover without restoring it from backup. (Bug #35801548)

- The event buffer used by the NDB API maintains an internal pool of free memory to reduce the interactions with the runtime and operating system, while allowing memory that is no longer needed to be returned for other uses. This free memory is subtracted from the total allocated memory to determine the memory is use which is reported and used for enforcing buffer limits and other purposes; this was represented using a 32-bit value, so that if it exceeded 4 GB, the value wrapped, and the amount of free memory appeared to be reduced. This had potentially adverse effects on event buffer memory release to the runtime and OS, free memory reporting, and memory limit handling.

  This is fixed by using a 64-bit value to represent the amount of pooled free memory. (Bug #35483764)

  References: See also: Bug #35655162, Bug #35663761.

- `START REPLICA`, `STOP REPLICA`, and `RESET REPLICA` statements are now written to `mysqld.log`. (Bug #35207235)

- `NDB` transporter handling in `mt.cpp` differentiated between neighbor transporters carrying signals between nodes in the same node group, and all other transporters. This sometimes led to issues with multiple transporters when a transporter connected nodes that were neighbors with nodes that were not. (Bug #33800633)

- Removed unnecessary warnings generated by transient disconnections of data nodes during restore operations. (Bug #33144487)

- During setup of utility tables, the schema event handler sometimes hung waiting for the global schema lock (GSL) to become available. This could happen when the physical tables had been dropped from the cluster, or when the connection was lost for some other reason. Now we use a try lock when attempting to acquire the GSL in such cases, thus causing another setup check attempt to be made at a later time if the global schema lock is not available. (Bug #32550019, Bug #35949017)

- API nodes did not record any information in the log relating to disconnects due to missed heartbeats from the data nodes. (Bug #29623286)

Index

W