DataStax Upgrade Guide

Documentation

September 8, 2016

Contents

About upgrading

The Upgrade guide provides detailed instructions on upgrading DataStax Enterprise, Cassandra, OpsCenter, DataStax Agents, DataStax drivers, the DataStax AMI, and reverting to earlier versions.

Attention: Read and understand these instructions before upgrading.

You have probably seen the recommendation to read all the instructions. This is a time where it doing so will make a difference. By understanding what to do beforehand, you can ensure your upgrade will be smooth and avoid many pitfalls and frustrations.

Upgrade instructions

- Upgrading DataStax Enterprise on page 4
- Upgrading from DataStax Community to DataStax Enterprise on page 35
- Upgrading Apache Cassandra™ on page 37
- Upgrading DataStax OpsCenter on page 43
- Upgrading DataStax drivers on page 52

Supported and compatible product versions

DataStax Enterprise

The Product compatibility page page provides a list of the DataStax Enterprise supported versions and EOL (End of Life) and EOSL (End of Service Life) information.

OpsCenter

The Product compatibility page provides information on the compatibility of OpsCenter with DataStax Enterprise or Cassandra.

DataStax drivers

The DataStax drivers page provide information on the available drivers and their compatibility with DataStax Enterprise and Cassandra.

Upgrading DataStax Enterprise

This section describes how to upgrade DataStax Enterprise.

Attention: Read and understand these instructions before upgrading.

You have probably seen the recommendation to read all the instructions. This is a time where it doing so will make a difference. By understanding what to do beforehand, you can ensure your upgrade will be smooth and avoid many pitfalls and frustrations.

To upgrade from any version of DataStax Community, follow the instructions in Upgrading from DataStax Community to DataStax Enterprise.

Planning your DataStax Enterprise upgrade

The upgrade process for DataStax Enterprise provides minimal downtime (ideally zero). During this process, you upgrade and restart one node at a time while other nodes continue to operate online. With a few exceptions, the cluster continues to work as though it were on the earlier version of DataStax Enterprise until all of the nodes in the cluster are upgraded.

Factors to consider when planning an upgrade:

Reduce risks

You can reduce risks and effort by employing a continual upgrade strategy. Ensure that you repair your nodes regularly. Node repair ensures that data on a replica is consistent with data on other nodes.

Backup data

DataStax recommends backing up your data prior to any version upgrade. A backup provides the ability to revert and restore all the data used in the previous version if necessary. OpsCenter provides a Backup service that manages enterprise-wide backup and restore operations for DataStax Enterprise clusters.

Upgrade order

Upgrade order matters. Upgrade nodes in this order:

- In multiple datacenter clusters, upgrade every node in one datacenter before moving on to another datacenter.
- Upgrade the seed nodes within a datacenter first.
- Upgrade types in this order:
  1. DSE Analytics nodes or datacenters
  2. Cassandra nodes or datacenters
  3. DSE Search nodes or datacenters
- For DSE Analytics nodes using DSE Hadoop, upgrade the Job Tracker node first. Then upgrade Hadoop nodes, followed by Spark nodes.

Version impacts

Upgrades are impacted by the version you are upgrading from and the version you are upgrading to. The greater the gap between the current version and the target version, the more complex the upgrade.

Note: Be sure to check driver compatibility. Your driver may not be compatible with the upgrade version or require re-compiling.

Upgrades from earlier product versions might require an interim upgrade to a required version:

Current version	Required interim version	Target version
• DataStax Enterprise 4.0, 4.5, or 4.6 • DataStax Community or open source Apache Cassandra™ 2.0.x	• DataStax Enterprise 4.8 • DataStax Community or open source Apache Cassandra 2.1.x	DataStax Enterprise 5.0
• DataStax Enterprise 3.2.10 and earlier • DataStax Community or open source Apache Cassandra 1.2 and earlier	• DataStax Enterprise 4.0, 4.5, or 4.6 • DataStax Community or open source Apache Cassandra 2.0.x	DataStax Enterprise 4.7 or 4.8
• DataStax Enterprise 3.2.4 and earlier • DataStax Community or open source Apache Cassandra 1.2.15 and earlier	• DataStax Enterprise 3.2.5 and later • DataStax Community or open source Apache Cassandra 1.2.16, then 2.0	• DataStax Enterprise 4.6 • DataStax Enterprise 4.5 • DataStax Enterprise 4.0

Current version	Required interim version	Target version
• DataStax Enterprise 2.2.1 and earlier • DataStax Community or open source Apache Cassandra 1.1.8 and earlier • DataStax Community or open source Apache Cassandra 1.2.8 and earlier	• DataStax Enterprise 2.2.2 and later • DataStax Community or open source Apache Cassandra 1.1.9 • DataStax Community or open source Apache Cassandra 1.2.9 to 1.2.15	DataStax Enterprise 3.2

Minor DataStax Enterprise upgrades

The topic provides information on upgrading DataStax Enterprise between point (patch) releases. For example, upgrading from DataStax Enterprise 4.8.4 to 4.8.5.

Attention: Read and understand these instructions before upgrading.

You have probably seen the recommendation to read all the instructions. This is a time where it doing so will make a difference. By understanding what to do beforehand, you can ensure your upgrade will be smooth and avoid many pitfalls and frustrations.

Sections in this topic:

- Upgrade restrictions and limitations
- Upgrade steps

Recommendations

DataStax recommends backing up your data prior to any version upgrade. A backup provides the ability to revert and restore all the data used in the previous version if necessary. OpsCenter provides a Backup service that manages enterprise-wide backup and restore operations for DataStax Enterprise clusters.

Upgrade restrictions and limitations

Restrictions and limitations apply while a cluster is in a **partially upgraded** state.

With these exceptions, the cluster continues to work as though it were on the earlier version of DataStax Enterprise until all of the nodes in the cluster are upgraded.

General upgrade restrictions

- **Do not** enable new features.
- Do not run `nodetool repair`.

DSE Analytics (Hadoop and Spark) upgrade restrictions

- Do not run analytics jobs until all nodes are upgraded.
- Kill all Spark worker processes before you stop the node and install the new version.

DSE Search (Solr) upgrade restrictions and limitations

- Do not update schemas.
- Do not re-index DSE Search nodes during upgrade.
- Do not issue these types of queries during a rolling restart: `DDL` or `TRUNCATE`.

Security upgrade restrictions

- Do not change security credentials or permissions until after the upgrade is complete.
- Do not set up Kerberos authentication before upgrading. First upgrade the cluster, and then set up Kerberos.

Upgrade steps

Tip: The DataStax installer upgrades DataStax Enterprise 4.5 and later. It automatically performs many upgrade tasks.

1. Before upgrading, be sure that each node has ample free disk space.

   The required space depends on the compaction strategy. See Disk space in *Selecting hardware for enterprise implementations*.
2. Familiarize yourself with the changes and features in DataStax Enterprise and Apache Cassandra. See:

   - DataStax Enterprise release notes for the upgrade version.
   - *General upgrading advice for any version* and *New features* for Apache Cassandra in NEWS.txt. Be sure to read the `NEWS.txt` all the way back to your current version.
   - Apache Cassandra™ changes in CHANGES.txt.
   - DataStax Enterprise production-certified changes to Apache Cassandra in the Release notes.
   - DataStax driver changes.
3. Verify your current product version.
4. Run `nodetool repair` to ensure that data on each replica is consistent with data on other nodes.
5. **DSE Search nodes:** All unique key elements must be indexed in the Solr schema.

   To verify unique key elements, review `schema.xml` to ensure that all unique key fields must have `indexed=true`. If required, make changes to `schema.xml` and reindex.
6. Back up the configuration files you use:

   The configuration files are overwritten with default values during installation of the new version.
7. Upgrade order matters. Upgrade nodes in this order:

   With a few exceptions, the cluster continues to work as though it were on the earlier version of DataStax Enterprise until all of the nodes in the cluster are upgraded. Upgrade and restart the nodes one at a time. Other nodes in the cluster continue to operate at the earlier version until all nodes are upgraded.
8. Run `nodetool drain` to flush the commit log of the old installation:

   ```
   $ nodetool drain -h hostname
   ```

   This step saves time when nodes start up after the upgrade, and prevents DSE Search nodes from having to re-index data.
9. Stop the node (5.0, 4.8, 4.7, 4.6).
10. Use the appropriate method to install the new product version (5.0, 4.8, 4.7, 4.6).
11. To configure the new product version:

    a. Compare your backup configuration files to the new configuration files:

       - Look for any deprecated, removed, or changed settings.
       - Be sure you are familiar with the Apache Cassandra and DataStax Enterprise changes and features in the new release.

    b. Merge the applicable modifications into the new version.
12. Start the node (5.0, 4.8, 4.7, 4.6).
13. Upgrade the SSTables now that the new version is installed.

    ```
    $ nodetool upgradesstables
    ```

    Warning: Failure to upgrade SSTables when required results in a significant performance impact and increased disk usage. Upgrading is not complete until the SSTables are upgraded.

14. Verify that the upgraded datacenter names match the datacenter names in the keyspace schema definition:

```
$ nodetool status
```

15. Review the logs for warnings, errors, and exceptions.

    Warnings, errors, and exceptions are frequently found in the logs when starting up an upgraded node. Some of these log entries are informational to help you execute specific upgrade-related steps. If you find unexpected warnings, errors, or exceptions, contact DataStax Support.

16. Repeat the upgrade on each node in the cluster following the recommended upgrade order.

    A rolling restart while upgrading nodes in a cluster provides zero downtime. Upgrade and restart the nodes one at a time. Other nodes in the cluster continue to operate at the earlier version until all nodes are upgraded.

Upgrading to DataStax Enterprise 5.0

Follow these instructions to upgrade from DataStax Enterprise 4.7 or 4.8 to DataStax Enterprise 5.0. If you have an earlier version, upgrade to 4.8 before continuing.

Attention: Read and understand these instructions before upgrading.

You have probably seen the recommendation to read all the instructions. This is a time where it doing so will make a difference. By understanding what to do beforehand, you can ensure your upgrade will be smooth and avoid many pitfalls and frustrations.

Sections in this topic:

Recommendations

DataStax recommends backing up your data prior to any version upgrade. A backup provides the ability to revert and restore all the data used in the previous version if necessary. OpsCenter provides a Backup service that manages enterprise-wide backup and restore operations for DataStax Enterprise clusters.

Upgrade restrictions and limitations

Restrictions and limitations apply while a cluster is in a **partially upgraded** state.

With these exceptions, the cluster continues to work as though it were on the earlier version of DataStax Enterprise until all of the nodes in the cluster are upgraded.

General upgrade restrictions

- **Do not** enable new features.
- Do not run `nodetool repair`.
- Do not issue these types of CQL queries during a rolling restart: `DDL` and `TRUNCATE`.
- During upgrades, the nodes on different versions might show a schema disagreement.
- Failure to upgrade SSTables when required results in a significant performance impact and increased disk usage. Upgrading is not complete until the SSTables are upgraded.

Restrictions for DSE Analytic (Hadoop and Spark) nodes

- Do not run analytics jobs until all nodes are upgraded.

- Kill all Spark worker processes before you stop the node and install the new version.

DSE Search (Solr) upgrade restrictions and limitations

- Do not update schemas.
- Do not re-index DSE Search nodes during upgrade.
- Do not issue these types of queries during a rolling restart: `DDL` or `TRUNCATE`.
- While mixed versions of nodes exist during an upgrade, DataStax Enterprise runs two different servers for backward compatibility. One based on shard_transport_options, the other based on internode_messaging_options. (These options are located in `dse.yaml`.) After all nodes are upgraded to 5.0, shard_transport_options are ignored and internode_messaging_options are used. When all nodes are updated to 5.0, comment out all shard_transport_options instances.

Restrictions for nodes using any kind of security

- Do not change security credentials or permissions until after the upgrade is complete.
- Do not set up Kerberos authentication before upgrading. First upgrade the cluster, and then set up Kerberos.

Upgrading drivers

Be sure to check driver compatibility. Your driver may not be compatible with the upgrade version or require re-compiling.

Preparing to upgrade

Follow these steps to prepare each node for upgrading from DataStax Enterprise 4.7 or 4.8 to DataStax Enterprise 5.0:

1. Before upgrading, be sure that each node has ample free disk space.

   The required space depends on the compaction strategy. See Disk space in *Selecting hardware for enterprise implementations*.
2. Familiarize yourself with the changes and features in this release:
   - DataStax Enterprise 5.0 release notes.
   - *General upgrading advice for any version* and *New features* for Apache Cassandra™ 3.0 in NEWS.txt. Be sure to read the `NEWS.txt` all the way back to your current version.
   - Apache Cassandra™ changes in CHANGES.txt.
   - DataStax Enterprise 5.0 production-certified changes to Apache Cassandra.
   - DataStax driver changes.
3. Verify your current product version. If necessary, upgrade to an interim version:

Current version	Upgrade version
DataStax Enterprise 4.7 or 4.8	DataStax Enterprise 5.0
DataStax Enterprise 4.0, 4.5, or 4.6	DataStax Enterprise 4.8
DataStax Community or open source Apache Cassandra™ 2.0.x	DataStax Enterprise 4.8
DataStax Community 3.0.x	No interim version required.
DataStax Distribution of Apache Cassandra™ 3.x	Upgrade not available.

4. Verify the Java runtime version and upgrade to the recommended version.

   ```
   $ java -version
   ```

   The latest version of Oracle Java SE Runtime Environment 8 (JDK) (1.8.0_40 minimum) or OpenJDK 8 is recommended. The JDK is recommended for development and production systems. The JDK provides useful troubleshooting tools that are not in the JRE, such as jstack, jmap, jps, and jstat.

5. Run nodetool repair to ensure that data on each replica is consistent with data on other nodes.
6. **DSE Search nodes:**
   - The Lucene field cache (solr_field_cache_enabled) file is deprecated. This field is located in the `dse.yaml` file. Instead, for fields that are sorted, faceted, or grouped by, set `docValues="true"` on the field in the `schema.xml` file. Then RELOAD the core and reindex. The default value is false. To override false, set useFieldCache=true in the Solr request.

     During mixed versions upgrades, you can re-enable the field cache (`solr_field_cache_enabled: true`) to allow running queries but not reindexing.
   - All unique key elements must be indexed in the Solr schema.

     To verify unique key elements, review `schema.xml` to ensure that all unique key fields must have `indexed=true`. If required, make changes to `schema.xml` and reindex.
7. Back up the configuration files you use.

   The configuration files are overwritten with default values during installation of the new version.

Upgrade steps

Tip: The DataStax installer upgrades DataStax Enterprise and automatically performs many upgrade tasks.

Follow these steps on each node to upgrade from DataStax Enterprise 4.7 or 4.8 to DataStax Enterprise 5.0.

1. Upgrade order matters. Upgrade nodes in this order:

   With a few exceptions, the cluster continues to work as though it were on the earlier version of DataStax Enterprise until all of the nodes in the cluster are upgraded. Upgrade and restart the nodes one at a time. Other nodes in the cluster continue to operate at the earlier version until all nodes are upgraded.
2. **DSE Analytics nodes:** Kill all Spark worker processes.
3. **DSE Search nodes:** Review these considerations and take appropriate actions:
4. Run nodetool drain to flush the commit log of the old installation:

   ```
   $ nodetool drain -h hostname
   ```

   This step saves time when nodes start up after the upgrade, and prevents DSE Search nodes from having to re-index data.
5. Stop the node:
6. Use the appropriate method to install the new product version:
   - Upgrading RHEL installations
   - Installation instructions for upgrading DataStax Enterprise on Debian-based distributions on page 54
   - Installation instructions for upgrading DataStax Enterprise using the DataStax Enterprise tarball on page 55
7. If the cluster will run Hadoop in a Kerberos secure environment, change the `task-controller` file ownership to root and access permissions to 4750. For example:

   ```
   sudo chown root /usr/share/dse/resources/hadoop/native/Linux-amd64-64/bin/
   task-controller
   sudo chmod 4750 /usr/share/dse/resources/hadoop/native/Linux-amd64-64/bin/
   task-controller
   ```

   Package installations only: The default location of the `task-controller` file should be `/usr/share/dse/resources/hadoop/native/Linux-amd64-64/bin/task-controller`.
8. To configure the new product version:

   a. Compare your backup configuration files to the new configuration files:

- Look for any deprecated, removed, or changed settings.
- Be sure you are familiar with the Apache Cassandra and DataStax Enterprise changes and features in the new release.

b. Merge the applicable modifications into the new version.

9. Start the node.
   - Installer-Services and Package installations: See Starting DataStax Enterprise as a service.
   - Installer-No Services and Tarball installations: See Starting DataStax Enterprise as a stand-alone process.
10. Upgrade the SSTables now that the new version is installed.

```
$ nodetool upgradesstables
```

Warning: Failure to upgrade SSTables when required results in a significant performance impact and increased disk usage. Upgrading is not complete until the SSTables are upgraded.

11. Verify that the upgraded datacenter names match the datacenter names in the keyspace schema definition:

```
$ nodetool status
```

12. Review the logs for warnings, errors, and exceptions. Because DataStax Enterprise 5.0 uses Cassandra 3.0, the `output.log` may include warnings about the following:
    - sstable_compression
    - chunk_length_kb
    - memory_allocator
    - memtable_allocation_type
    - offheap_objects
    - netty_server_port - used only during the upgrade to 5.0. After all nodes are running 5.0, requests that are coordinated by this node are longer contact other nodes on this port. Instead requests use Inter-node messaging options.

    Warnings, errors, and exceptions are frequently found in the logs when starting up an upgraded node. Some of these log entries are informational to help you execute specific upgrade-related steps. If you find unexpected warnings, errors, or exceptions, contact DataStax Support.

    During upgrade of DSE Analytics nodes, exceptions about the Task Tracker are logged in the nodes that are not yet upgraded to 5.0. The jobs succeed after the entire cluster is upgraded.
13. Repeat the upgrade on each node in the cluster following the recommended order.
14. If your upgrade includes DSE Search nodes:

a. After all nodes are updated to 5.0, comment out all shard_transport_options instances in each `dse.yaml` file.

b. Restart the nodes to fully shut down the old shard transport. You can do this at your convenience, because although the old shard transport is still running, it is not used.

Warning messages during and after upgrade

You can ignore some log messages that occur during and after an upgrade.

If you made schema changes shortly before upgrading to DataStax Enterprise 5.0, log messages similar to the following might appear after upgrading:

```
WARN  [main] 2016-06-23 12:01:59,693  CommitLogReplayer.java:154
 - Skipped 31 mutations from unknown (probably removed) CF with id
 b0f22357-4458-3cdb-9631-c43e59ce3676
WARN  [main] 2016-06-23 12:01:59,693  CommitLogReplayer.java:154
 - Skipped 1 mutations from unknown (probably removed) CF with id
 3aa75225-4f82-350b-8d5c-430fa221fa0a
```

```
WARN  [main] 2016-06-23 12:01:59,696  CommitLogReplayer.java:154 - Skipped
 1 mutations from unknown (probably removed) CF with id 45f5b360-24bc-3f83-
a363-1034ea4fa697
WARN  [main] 2016-06-23 12:01:59,696  CommitLogReplayer.java:154
 - Skipped 1 mutations from unknown (probably removed) CF with id
 0359bc71-7123-3ee1-9a4a-b9dfb11fc125
WARN  [main] 2016-06-23 12:01:59,697  CommitLogReplayer.java:154
 - Skipped 1 mutations from unknown (probably removed) CF with id
 296e9c04-9bec-3085-827d-c17d3df2122a
```

You can safely ignore these log messages.

Upgrading to DataStax Enterprise 4.7 or 4.8

Follow these instructions to upgrade from DataStax Enterprise 4.0, 4.5, and 4.6 to DataStax Enterprise 4.7 or 4.8.

Attention: Read and understand these instructions before upgrading.

You have probably seen the recommendation to read all the instructions. This is a time where it doing so will make a difference. By understanding what to do beforehand, you can ensure your upgrade will be smooth and avoid many pitfalls and frustrations.

Sections in this topic:

Recommendations

DataStax recommends backing up your data prior to any version upgrade. A backup provides the ability to revert and restore all the data used in the previous version if necessary. OpsCenter provides a Backup service that manages enterprise-wide backup and restore operations for DataStax Enterprise clusters.

Upgrade restrictions and limitations

Restrictions and limitations apply while a cluster is in a **partially upgraded** state.

With these exceptions, the cluster continues to work as though it were on the earlier version of DataStax Enterprise until all of the nodes in the cluster are upgraded.

General upgrade restrictions

- **Do not** enable new features.
- Do not run `nodetool repair`.
- Do not issue these types of CQL queries during a rolling restart: `DDL` and `TRUNCATE`.
- During upgrades, the nodes on different versions might show a schema disagreement.
- Failure to upgrade SSTables when required results in a significant performance impact and increased disk usage. Upgrading is not complete until the SSTables are upgraded.

Restrictions for DSE Analytic (Hadoop and Spark) nodes

- Do not run analytics jobs until all nodes are upgraded.
- Kill all Spark worker processes before you stop the node and install the new version.

Restrictions for DSE Search (Solr) nodes

- Do not update schemas.

- Do not re-index DSE Search nodes during upgrade.
- Do not issue these types of queries during a rolling restart: `DDL` or `TRUNCATE`.
- During the upgrade process on a cluster with mixed versions where DataStax Enterprise 4.7 or 4.8 supports pagination and earlier versions do not, issuing queries from the upgraded nodes will return only FetchSize results.

Restrictions for nodes using any kind of security

- Do not change security credentials or permissions until after the upgrade is complete.
- Do not set up Kerberos authentication before upgrading. First upgrade the cluster, and then set up Kerberos.

Upgrade impact when driver versions are incompatible

Be sure to check driver compatibility. Your driver may not be compatible with the upgrade version or require re-compiling.

During upgrades, you might experience driver-specific impact when clusters have mixed versions of drivers. If your cluster has mixed versions, the protocol version is negotiated with the first host that the driver connects to. To avoid driver version incompatibility during upgrades, use one of these workarounds:

- Force a protocol version at startup. For example, keep the Java driver at v2 while the upgrade is happening. Switch to the Java driver v3 only after the entire cluster is upgraded.
- Ensure that the list of initial contact points contains only hosts with the oldest driver version. For example, the initial contact points contain only Java driver v2.

For driver compatibility, see the driver matrix. For details on protocol version negotiation, see *Protocol version with mixed clusters* in the Java driver version you're using.

Preparing to upgrade from 4.0, 4.5, and 4.6 to 4.7 or 4.8

Follow these steps to prepare to upgrade from DataStax Enterprise 4.0, 4.5, and 4.6 to DataStax Enterprise 4.7 or 4.8.

1. Before upgrading, be sure that each node has ample free disk space.

   The required space depends on the compaction strategy. See Disk space in *Selecting hardware for enterprise implementations*.
2. Familiarize yourself with the changes and features in this release:
   - DataStax Enterprise release notes for 4.7 and 4.8.
   - *General upgrading advice for any version* and *New features* for Apache Cassandra™ 2.1 in NEWS.txt. Be sure to read the `NEWS.txt` all the way back to your current version.
   - Apache Cassandra™ changes in CHANGES.txt.
   - DataStax Enterprise 4.7 production-certified changes to Apache Cassandra.
   - DataStax Enterprise 4.8 production-certified changes to Apache Cassandra.
3. Verify your current product version. If necessary, upgrade to one these required interim versions before upgrading to 4.7 or 4.8:
   - DataStax Enterprise 4.0 and later
   - DataStax Community or open source Apache Cassandra™ 2.0.x
4. Upgrade the SSTables on each node to ensure that all SSTables are on the current version.

   ```
   $ nodetool upgradesstables
   ```

   Warning: Failure to upgrade SSTables when required results in a significant performance impact and increased disk usage. Upgrading is not complete until the SSTables are upgraded.

   If the SSTables are already on the current version, the command returns immediately and no action is taken.
5. Verify the Java runtime version and upgrade to the recommended version.

```
$ java -version
```

The latest version of Oracle Java SE Runtime Environment 7 or 8 or OpenJDK 7 is recommended. The JDK is recommended for development and production systems. The JDK provides useful troubleshooting tools that are not in the JRE, such as jstack, jmap, jps, and jstat.

Note: If using Oracle Java 7, you must use at least 1.7.0_25. If using Oracle Java 8, you must use at least 1.8.0_40.

6. Run nodetool repair to ensure that data on each replica is consistent with data on other nodes.
7. **DSE Search nodes:** All unique key elements must be indexed in the Solr schema.

   To verify unique key elements, review `schema.xml` to ensure that all unique key fields must have `indexed=true`. If required, make changes to `schema.xml` and reindex.
8. Back up the configuration files you use.

   The configuration files are overwritten with default values during installation of the new version.
9. Upgrade order matters. Upgrade nodes in this order:

   With a few exceptions, the cluster continues to work as though it were on the earlier version of DataStax Enterprise until all of the nodes in the cluster are upgraded. Upgrade and restart the nodes one at a time. Other nodes in the cluster continue to operate at the earlier version until all nodes are upgraded.

Upgrading from 4.0, 4.5, and 4.6 to 4.7 or 4.8

Tip: The DataStax installer upgrades DataStax Enterprise and automatically performs many upgrade tasks.

Follow these steps on each node to upgrade from DataStax Enterprise 4.0, 4.5, and 4.6 to DataStax Enterprise 4.7 or 4.8.

1. **DSE Analytics nodes:** Kill all Spark worker processes.
2. **DSE Search nodes:** Review these considerations and take appropriate actions:
   - **For upgrades from 4.6 or earlier:** If your `schema.xml` contains `fieldTypes` using `docValuesFormat="Disk"`, you **must** modify the file to remove the `docValuesFormat` attribute, reload, and optimize your index to rewrite to the default codec. This a requirement for Solr 4.10 and above.
   - **To maintain 4.6 query behavior:**

     Disable driver pagination by editing the `dse.yaml` file and setting `cql_solr_query_paging: off`. DataStax Enterprise 4.7 or 4.8 integrates native driver paging with Solr cursor-based paging (4.7 or 4.8). You can turn on paging after you verify the upgrade.
   - **For upgrades from 4.0.0:** See Special steps for ugrades from 4.0.0 for special instructions.
3. Run nodetool drain to flush the commit log of the old installation:

   ```
   $ nodetool drain -h hostname
   ```

   This step saves time when nodes start up after the upgrade, and prevents DSE Search nodes from having to re-index data.
4. Stop the node (4.7 or 4.8).
5. Use the appropriate method to install the new product version:
   - Upgrading RHEL installations
   - Installation instructions for upgrading DataStax Enterprise on Debian-based distributions on page 54
   - Installation instructions for upgrading DataStax Enterprise using the DataStax Enterprise tarball on page 55

6. To configure the new product version, use your backup configuration files to merge modifications into the configuration files for the new version.
7. Start the node.
   - Installer-Services and Package installations: See *Starting DataStax Enterprise as a service* (4.7 or 4.8).
   - Installer-No Services and Tarball installations: See *Starting DataStax Enterprise as a stand-alone process* 4.7 or 4.8).
8. When the upgrade includes a major upgrade of Apache Cassandra, upgrade the SSTables now that the new version is installed.

   ```
   $ nodetool upgradesstables
   ```

   Warning: Failure to upgrade SSTables when required results in a significant performance impact and increased disk usage. Upgrading is not complete until the SSTables are upgraded.

   Apache Cassandra™ requires upgrading SSTables for major releases.

   - DataStax Enterprise 4.7 to 4.8 uses Cassandra 2.1
   - DataStax Enterprise 4.0 to 4.6 uses Cassandra 2.0
   - DataStax Enterprise 3.1 to 3.2 uses Cassandra 1.2
   - DataStax Enterprise 2.2 to 3.0 uses Cassandra 1.1
   - DataStax Enterprise 1.0 to 2.1 uses Cassandra 1.0
9. **Upgrades from 4.6 or earlier:** If existing tables use the DSE In-Memory option:
   a. Turn off SSTable compression.

      ```
      ALTER TABLE <tablename> WITH compression = {'sstable_compression' :
       ''} ;
      ```

   b. Rewrite existing SSTables without compression:

      ```
      $ nodetool upgradesstables -a <keyspacename> <tablename>
      ```
10. Verify that the upgraded datacenter names match the datacenter names in the keyspace schema definition:

    ```
    $ nodetool status
    ```
11. Review the logs for warnings, errors, and exceptions.

    Warnings, errors, and exceptions are frequently found in the logs when starting up an upgraded node. Some of these log entries are informational to help you execute specific upgrade-related steps. If you find unexpected warnings, errors, or exceptions, contact DataStax Support.

    During upgrade of DSE Analytics nodes, exceptions about the Task Tracker are logged in the nodes that are not yet upgraded to 4.7 or 4.8. The jobs succeed after the entire cluster is upgraded.

    Because DataStax Enterprise 4.7 and 4.8 use Cassandra 2.1, the `output.log` includes the following warnings:

    - Deprecated `cassandra.yaml` options are removed
      - multithreaded_compaction
      - memtable_flush_queue_size
      - compaction_preheat_key_cache
      - in_memory_compaction_limit_in_mb
      - preheat_kernel_page_cache
    - `cassandra-env.sh` change

      ```
      JVM_OPTS="$JVM_OPTS -javaagent:$CASSANDRA_HOME/lib/jamm-0.2.5.jar"
      ```

      to

```
JVM_OPTS="$JVM_OPTS -javaagent:$CASSANDRA_HOME/lib/jamm-0.3.0.jar"
```

12. In DataStax Enterprise 4.8, audit log tables use DateTieredCompactionStrategy (DTCS). DataStax recommends changing tables that were created in earlier releases to use:

```
DTCS: ALTER TABLE dse_audit.audit_log WITH
 COMPACTION={'class':'DateTieredCompactionStrategy'};
```

13. Repeat the upgrade on each node in the cluster following the recommended order.

Upgrading to DataStax Enterprise 4.6

Follow these instructions to upgrade from DataStax Enterprise versions 3.2.5 to 4.5 to DataStax Enterprise 4.6.

Attention: Read and understand these instructions before upgrading.

You have probably seen the recommendation to read all the instructions. This is a time where it doing so will make a difference. By understanding what to do beforehand, you can ensure your upgrade will be smooth and avoid many pitfalls and frustrations.

Sections in this topic:

Recommendations

DataStax recommends backing up your data prior to any version upgrade. A backup provides the ability to revert and restore all the data used in the previous version if necessary. OpsCenter provides a Backup service that manages enterprise-wide backup and restore operations for DataStax Enterprise clusters.

Upgrade restrictions and limitations

Restrictions and limitations apply while a cluster is in a **partially upgraded** state.

With these exceptions, the cluster continues to work as though it were on the earlier version of DataStax Enterprise until all of the nodes in the cluster are upgraded.

General upgrade restrictions

- **Do not** enable new features.
- Do not run `nodetool repair`.
- Do not issue these types of CQL queries during a rolling restart: `DDL` and `TRUNCATE`.
- During upgrades, the nodes on different versions might show a schema disagreement.
- Failure to upgrade SSTables when required results in a significant performance impact and increased disk usage. Upgrading is not complete until the SSTables are upgraded.

Restrictions for DSE Analytic (Hadoop and Spark) nodes

- Do not run analytics jobs until all nodes are upgraded.
- Kill all Spark worker processes before you stop the node and install the new version.

Restrictions for DSE Search (Solr) nodes

- Do not update schemas.
- Do not re-index DSE Search nodes during upgrade.
- Do not issue these types of queries during a rolling restart: `DDL` or `TRUNCATE`.

- During the upgrade process on a cluster with mixed versions where DataStax Enterprise 4.7 or 4.8 supports pagination and earlier versions do not, issuing queries from the upgraded nodes will return only FetchSize results.

Restrictions for nodes using any kind of security

- Do not change security credentials or permissions until after the upgrade is complete.
- Do not set up Kerberos authentication before upgrading. First upgrade the cluster, and then set up Kerberos.

Upgrade impact when driver versions are incompatible

Be sure to check driver compatibility. Your driver may not be compatible with the upgrade version or require re-compiling.

During upgrades, you might experience driver-specific impact when clusters have mixed versions of drivers. If your cluster has mixed versions, the protocol version is negotiated with the first host that the driver connects to. To avoid driver version incompatibility during upgrades, use one of these workarounds:

- Force a protocol version at startup. For example, keep the Java driver at v2 while the upgrade is happening. Switch to the Java driver v3 only after the entire cluster is upgraded.
- Ensure that the list of initial contact points contains only hosts with the oldest driver version. For example, the initial contact points contain only Java driver v2.

For driver compatibility, see the driver matrix. For details on protocol version negotiation, see *Protocol version with mixed clusters* in the Java driver version you're using.

Preparing to upgrade from 3.2.5 and later to 4.6.

Tip: The DataStax installer upgrades DataStax Enterprise and automatically performs many upgrade tasks.

If you do not use the DataStax installer, follow these steps on each node to prepare to upgrade from DataStax Enterprise 3.2.5 and later to DataStax Enterprise 4.6.

1. Before upgrading, be sure that each node has ample free disk space.

   The required space depends on the compaction strategy. See Disk space in *Selecting hardware for enterprise implementations*.
2. Verify your current product version. If necessary, upgrade to one these required interim versions before upgrading to 4.6:
   - DataStax Enterprise 3.2.5 and later
   - DataStax Community or open source Apache Cassandra™ 1.2.16
3. Upgrade the SSTables on each node to ensure that all SSTables are on the current version.

   ```
   $ nodetool upgradesstables
   ```

   Warning: Failure to upgrade SSTables when required results in a significant performance impact and increased disk usage. Upgrading is not complete until the SSTables are upgraded.

   If the SSTables are already on the current version, the command returns immediately and no action is taken.
4. **Only for upgrades from 3.2.x:** Edit the cassandra.yaml file and remove or comment out the following options:

   ```
   # auth_replication_options:
   # replication_factor: 1
   ```
5. **Only for upgrades from 4.0.0 with search nodes to 4.5:** See Upgrading from DataStax Enterprise 4.0.0 with search nodes.
6. Verify the Java runtime version and upgrade to the recommended version.

```
$ java -version
```

The latest version of Oracle Java SE Runtime Environment 7 or 8 or OpenJDK 7 is recommended. The JDK is recommended for development and production systems. The JDK provides useful troubleshooting tools that are not in the JRE, such as jstack, jmap, jps, and jstat.

Note: If using Oracle Java 7, you must use at least 1.7.0_25. If using Oracle Java 8, you must use at least 1.8.0_40.

7. Familiarize yourself with the changes and features in this release:
   - DataStax Enterprise 4.6 release notes.

     Endpoint snitch: Starting in DataStax Enterprise 4.6, the endpoint snitch is set in cassandra.yaml, not dse.yaml. The `com.datastax.bdp.snitch.DseDelegateSnitch` is replaced by `com.datastax.bdp.snitch.DseSimpleSnitch` in `cassandra.yaml` and the `endpoint_snitch` option has been removed from `dse.yaml`.

     Note: The DataStax Installer automatically sets the default `endpoint_snitch` to `DseSimpleSnitch` and removes the option from the `dse.yaml` file.
   - General upgrade advice and Apache Cassandra features in NEWS.txt. If you are upgrading from an earlier release, read `NEWS.txt` all the way back to your current version.
   - Apache Cassandra changes in CHANGES.txt.
8. Back up the configuration files you use.

   The configuration files are overwritten with default values during installation of the new version.
9. Run nodetool repair to ensure that data on each replica is consistent with data on other nodes.
10. Upgrade order matters. Upgrade nodes in this order:

    With a few exceptions, the cluster continues to work as though it were on the earlier version of DataStax Enterprise until all of the nodes in the cluster are upgraded. Upgrade and restart the nodes one at a time. Other nodes in the cluster continue to operate at the earlier version until all nodes are upgraded.

Upgrading from 3.2.5 and later to 4.6

Follow these steps on each node to upgrade from DataStax Enterprise 3.2.5 and later to DataStax Enterprise 4.6.

1. Run nodetool drain to flush the commit log of the old installation:

   ```
   $ nodetool drain -h hostname
   ```

   This step saves time when nodes start up after the upgrade, and prevents DSE Search nodes from having to re-index data.
2. **DSE Analytics nodes:** Kill all Spark worker processes.
3. Stop the node (4.7 or 4.8).
4. Use the appropriate method to install the new product version:
   - Upgrading RHEL installations
   - Installation instructions for upgrading DataStax Enterprise on Debian-based distributions on page 54
   - Installation instructions for upgrading DataStax Enterprise using the DataStax Enterprise tarball on page 55
5. Open `cassandra.yaml` to set the `endpoint_snitch` option to the same snitch that is set in `delegated_snitch` in `dse.yaml`.

   ```
   endpoint_snitch: com.datastax.bdp.snitch.DseSimpleSnitch
   ```
6. Remove the `delegated_snitch` option from the old `dse.yaml` file.

7. To configure the new product version, use your backup configuration files to merge modifications into the configuration files for the new version.
8. Start the node.
   - Installer-Services and Package installations: See Starting DataStax Enterprise as a service.
   - Installer-No Services and Tarball installations: See Starting DataStax Enterprise as a stand-alone process).
9. When the upgrade includes a major upgrade of Apache Cassandra, upgrade the SSTables now that the new version is installed.

   ```
   $ nodetool upgradesstables
   ```

   Warning: Failure to upgrade SSTables when required results in a significant performance impact and increased disk usage. Upgrading is not complete until the SSTables are upgraded.

   Apache Cassandra™ requires upgrading SSTables for major releases.

   - DataStax Enterprise 4.7 to 4.8 uses Cassandra 2.1
   - DataStax Enterprise 4.0 to 4.6 uses Cassandra 2.0
   - DataStax Enterprise 3.1 to 3.2 uses Cassandra 1.2
   - DataStax Enterprise 2.2 to 3.0 uses Cassandra 1.1
   - DataStax Enterprise 1.0 to 2.1 uses Cassandra 1.0
10. Verify that the upgraded datacenter names match the datacenter names in the keyspace schema definition:

    ```
    $ nodetool status
    ```
11. Review the logs for warnings, errors, and exceptions.

    Warnings, errors, and exceptions are frequently found in the logs when starting up an upgraded node. Some of these log entries are informational to help you execute specific upgrade-related steps. If you find unexpected warnings, errors, or exceptions, contact DataStax Support.
12. Repeat the upgrade on each node in the cluster following the recommended order.

Upgrading to DataStax Enterprise 4.0 or 4.5

Sections in this topic:

Attention: Read and understand these instructions before upgrading.

You have probably seen the recommendation to read all the instructions. This is a time where it doing so will make a difference. By understanding what to do beforehand, you can ensure your upgrade will be smooth and avoid many pitfalls and frustrations.

Recommendations

DataStax recommends backing up your data prior to any version upgrade. A backup provides the ability to revert and restore all the data used in the previous version if necessary. OpsCenter provides a Backup service that manages enterprise-wide backup and restore operations for DataStax Enterprise clusters.

Upgrade restrictions and limitations

Restrictions and limitations apply while a cluster is in a **partially upgraded** state.

With these exceptions, the cluster continues to work as though it were on the earlier version of DataStax Enterprise until all of the nodes in the cluster are upgraded.

General upgrade restrictions

- **Do not** enable new features.
- Do not run `nodetool repair`.
- Do not issue these types of CQL queries during a rolling restart: `DDL` and `TRUNCATE`.
- During upgrades, the nodes on different versions might show a schema disagreement.
- Failure to upgrade SSTables when required results in a significant performance impact and increased disk usage. Upgrading is not complete until the SSTables are upgraded.

Restrictions for DSE Analytic (Hadoop and Spark) nodes

- Do not run analytics jobs until all nodes are upgraded.
- Kill all Spark worker processes before you stop the node and install the new version.

Restrictions for DSE Search (Solr) nodes

- Do not update schemas.
- Do not re-index DSE Search nodes during upgrade.
- Do not issue these types of queries during a rolling restart: `DDL` or `TRUNCATE`.
- During the upgrade process on a cluster with mixed versions where DataStax Enterprise 4.7 or 4.8 supports pagination and earlier versions do not, issuing queries from the upgraded nodes will return only FetchSize results.

Restrictions for nodes using any kind of security

- Do not change security credentials or permissions until after the upgrade is complete.
- Do not set up Kerberos authentication before upgrading. First upgrade the cluster, and then set up Kerberos.

Upgrade impact when driver versions are incompatible

Be sure to check driver compatibility. Your driver may not be compatible with the upgrade version or require re-compiling.

During upgrades, you might experience driver-specific impact when clusters have mixed versions of drivers. If your cluster has mixed versions, the protocol version is negotiated with the first host that the driver connects to. To avoid driver version incompatibility during upgrades, use one of these workarounds:

- Force a protocol version at startup. For example, keep the Java driver at v2 while the upgrade is happening. Switch to the Java driver v3 only after the entire cluster is upgraded.
- Ensure that the list of initial contact points contains only hosts with the oldest driver version. For example, the initial contact points contain only Java driver v2.

For driver compatibility, see the driver matrix. For details on protocol version negotiation, see *Protocol version with mixed clusters* in the Java driver version you're using.

Preparing to upgrade from 3.2.5 or later to 4.0 or 4.5

Tip: The DataStax installer upgrades DataStax Enterprise and automatically performs many upgrade tasks.

If you do not use the DataStax installer, follow these steps to prepare to upgrade from DataStax Enterprise 3.2.5 or later to DataStax Enterprise 4.0 to 4.5.

1. Before upgrading, be sure that each node has ample free disk space.

   The required space depends on the compaction strategy. See Disk space in *Selecting hardware for enterprise implementations*.
2. Verify your current product version. If necessary, upgrade to one these required interim versions before upgrading to 4.0 or 4.5:

- DataStax Enterprise 3.2.5 and later
- DataStax Community or open source Apache Cassandra™ 1.2.16

3. **Only for upgrades from 3.2.x:** Edit the cassandra.yaml file and remove or comment out the following options:

```
# auth_replication_options:
# replication_factor: 1
```

4. **Only for upgrades from 4.0.0 with search nodes to 4.5:** See Upgrading from DataStax Enterprise 4.0.0 with search nodes.
5. Upgrade the SSTables on each node to ensure that all SSTables are on the current version.

```
$ nodetool upgradesstables
```

Warning: Failure to upgrade SSTables when required results in a significant performance impact and increased disk usage. Upgrading is not complete until the SSTables are upgraded.

If the SSTables are already on the current version, the command returns immediately and no action is taken.

6. If you are upgrading from DataStax Enterprise 4.0.0 and have DSE Search nodes, see Special steps for ugrades from 4.0.0.
7. Verify the Java runtime version and upgrade to the recommended version.

```
$ java -version
```

The latest version of Oracle Java SE Runtime Environment 7 or 8 or OpenJDK 7 is recommended. The JDK is recommended for development and production systems. The JDK provides useful troubleshooting tools that are not in the JRE, such as jstack, jmap, jps, and jstat.

Note: If using Oracle Java 7, you must use at least 1.7.0_25. If using Oracle Java 8, you must use at least 1.8.0_40.

8. Familiarize yourself with the changes and features in this release:
   - DataStax Enterprise release notes for 4.0 and 4.5.
   - *General upgrading advice for any version* and *New features* for Apache Cassandra™ 2.0 in NEWS.txt. Be sure to read the `NEWS.txt` for each version all the way back to your current version.
   - Apache Cassandra™ changes in CHANGES.txt.
9. Back up the configuration files you use.

The configuration files are overwritten with default values during installation of the new version.

Upgrading from 3.2.5 or later to 4.0 or 4.5

Follow these steps to upgrade from 3.2.5 or later to DataStax Enterprise 4.0 or 4.5.

1. Verify your current product version. If necessary, upgrade to one these required interim versions before upgrading to 4.0 or 4.5:
   - DataStax Enterprise 3.2.5 and later
   - DataStax Community or open source Cassandra™ 1.2.16
2. **Only for upgrades from 3.2.x:** Edit the cassandra.yaml file and remove or comment out the following options:

```
# auth_replication_options:
# replication_factor: 1
```

3. **Only for upgrades from 4.0.0 with search nodes to 4.5:** See Upgrading from DataStax Enterprise 4.0.0 with search nodes.
4. Upgrade the SSTables on each node to ensure that all SSTables are on the current version.

```
$ nodetool upgradesstables
```

Warning: Failure to upgrade SSTables when required results in a significant performance impact and increased disk usage. Upgrading is not complete until the SSTables are upgraded.

If the SSTables are already on the current version, the command returns immediately and no action is taken.

5. If you are upgrading from DataStax Enterprise 4.0.0 and have DSE Search nodes, see Upgrading from DataStax Enterprise 4.0.0 with search nodes.
6. Verify the Java runtime version and upgrade to the recommended version.

```
$ java -version
```

The latest version of Oracle Java SE Runtime Environment 7 or 8 or OpenJDK 7 is recommended. The JDK is recommended for development and production systems. The JDK provides useful troubleshooting tools that are not in the JRE, such as jstack, jmap, jps, and jstat.

Note: If using Oracle Java 7, you must use at least 1.7.0_25. If using Oracle Java 8, you must use at least 1.8.0_40.

7. Familiarize yourself with the changes and features in this release:
   - DataStax Enterprise release notes for 4.0 and 4.5.
   - *General upgrading advice for any version* and *New features* for Apache Cassandra™ 2.0 in NEWS.txt. Be sure to read the `NEWS.txt` for each version all the way back to your current version.
   - Apache Cassandra™ changes in CHANGES.txt.
8. Back up the configuration files you use.

   The configuration files are overwritten with default values during installation of the new version.
9. Upgrade order matters. Upgrade nodes in this order:

   With a few exceptions, the cluster continues to work as though it were on the earlier version of DataStax Enterprise until all of the nodes in the cluster are upgraded. Upgrade and restart the nodes one at a time. Other nodes in the cluster continue to operate at the earlier version until all nodes are upgraded.
10. Run nodetool drain to flush the commit log of the old installation:

```
$ nodetool drain -h hostname
```

This step saves time when nodes start up after the upgrade, and prevents DSE Search nodes from having to re-index data.

11. Stop the node (4.0).
12. Use the appropriate method to install the new product version:
    - Installation instructions for upgrading DataStax Enterprise using the DataStax Enterprise tarball on page 55
    - Installation instructions for upgrading DataStax Enterprise on Debian-based distributions on page 54
    - Upgrading RHEL installations
13. To configure the new product version, use your backup configuration files to merge modifications into the configuration files for the new version.
14. Start the node.
    - Installer-Services and Package installations: See Starting DataStax Enterprise as a service.
    - Installer-No Services and Tarball installations: See Starting DataStax Enterprise as a stand-alone process).
15. When the upgrade includes a major upgrade of Apache Cassandra, upgrade the SSTables now that the new version is installed.

```
$ nodetool upgradesstables
```

Warning: Failure to upgrade SSTables when required results in a significant performance impact and increased disk usage. Upgrading is not complete until the SSTables are upgraded.

Apache Cassandra™ requires upgrading SSTables for major releases.

- DataStax Enterprise 4.7 to 4.8 uses Cassandra 2.1
- DataStax Enterprise 4.0 to 4.6 uses Cassandra 2.0
- DataStax Enterprise 3.1 to 3.2 uses Cassandra 1.2
- DataStax Enterprise 2.2 to 3.0 uses Cassandra 1.1
- DataStax Enterprise 1.0 to 2.1 uses Cassandra 1.0

16. Verify that the upgraded datacenter names match the datacenter names in the keyspace schema definition:

```
$ nodetool status
```

17. Review the logs for warnings, errors, and exceptions.

    Warnings, errors, and exceptions are frequently found in the logs when starting up an upgraded node. Some of these log entries are informational to help you execute specific upgrade-related steps. If you find unexpected warnings, errors, or exceptions, contact DataStax Support.

18. Repeat the upgrade on each node in the cluster following the recommended order.

Special steps for upgrades from DataStax Enterprise 4.0.0

Due to a bug in DataStax Enterprise 4.0.0, upgrading clusters with search nodes from DataStax Enterprise 4.0.0 to 4.0.x requires special steps to prevent data loss.

Note: This bug impacts upgrades only from DataStax Enterprise 4.0.0.

Procedure

1. Drain each node in the cluster, but do not stop the node.
2. Reload the Solr core.

   In the following example, the Solr core is `wiki.solr` running on the local host on port 8983.

   ```
   $ curl -X POST "http://127.0.0.1:8983/solr/admin/cores?
   action=RELOAD&name=wiki.solr&reindex=false&deleteAll=false"
   ```

3. Upgrade the cluster.
4. Re-index the Solr core.

   In the following example, the Solr core is `wiki.solr` running on the local host on port 8983.

   ```
   $ curl -X POST "http://127.0.0.1:8983/solr/admin/cores?
   action=RELOAD&name=wiki.solr&reindex=true"
   ```

Upgrading to DataStax Enterprise 3.2

Sections in this topic:

- Upgrade steps

Attention: Read and understand these instructions before upgrading.

You have probably seen the recommendation to read all the instructions. This is a time where it doing so will make a difference. By understanding what to do beforehand, you can ensure your upgrade will be smooth and avoid many pitfalls and frustrations.

Recommendations

DataStax recommends backing up your data prior to any version upgrade. A backup provides the ability to revert and restore all the data used in the previous version if necessary.

Upgrade limitations

Limitations apply while a cluster is in a partially upgraded state.

With these exceptions, the cluster continues to work as though it were on the earlier version of DataStax Enterprise until all of the nodes in the cluster are upgraded.

General upgrade restrictions

- **Do not** enable new features.
- Do not run `nodetool repair`.
- Do not issue these types of CQL queries during a rolling restart: `DDL` and `TRUNCATE`.
- During upgrades, the nodes on different versions might show a schema disagreement.
- Failure to upgrade SSTables when required results in a significant performance impact and increased disk usage. Upgrading is not complete until the SSTables are upgraded.

Security upgrade limitations

- Do not change security credentials or permissions until after the upgrade is complete.

Upgrade impact when driver versions are incompatible

Be sure to check driver compatibility. Your driver may not be compatible with the upgrade version or require re-compiling.

During upgrades, you might experience driver-specific impact when clusters have mixed versions of drivers. If your cluster has mixed versions, the protocol version is negotiated with the first host that the driver connects to. To avoid driver version incompatibility during upgrades, use one of these workarounds:

- Force a protocol version at startup. For example, keep the Java driver at v2 while the upgrade is happening. Switch to the Java driver v3 only after the entire cluster is upgraded.
- Ensure that the list of initial contact points contains only hosts with the oldest driver version. For example, the initial contact points contain only Java driver v2.

For driver compatibility, see the driver matrix. For details on protocol version negotiation, see *Protocol version with mixed clusters* in the Java driver version you're using.

Preparing to upgrade from DataStax Enterprise 2.2.2 and later to DataStax Enterprise 3.2

Tip: The DataStax installer upgrades DataStax Enterprise and automatically performs many upgrade tasks.

If you do not use the DataStax installer, follow these steps to prepare to upgrade from DataStax Enterprise 2.2.2 and later to DataStax Enterprise 3.2.

1. Before upgrading, be sure that each node has ample free disk space.

   The required space depends on the compaction strategy. See Disk space in *Selecting hardware for enterprise implementations*.

2. Verify your current product version. If necessary, upgrade to one these required interim versions before upgrading to 3.2:
   - DataStax Enterprise 2.2.2 and later
   - DataStax Community or open source Apache Cassandra™ 1.1.9
   - DataStax Community or open source Apache Cassandra 1.2.9 to 1.2.15
3. For upgrades from DataStax Enterprise 3.0.x and 2.2.x, review and observe the specific actions in:
   - Upgrading from 3.0
   - Upgrading from 2.2
4. Verify the Java runtime version and upgrade to the recommended version.

   ```
   $ java -version
   ```

   The latest version of Oracle Java SE Runtime Environment 7 or 8 or OpenJDK 7 is recommended. The JDK is recommended for development and production systems. The JDK provides useful troubleshooting tools that are not in the JRE, such as jstack, jmap, jps, and jstat.

   Note: If using Oracle Java 7, you must use at least 1.7.0_25. If using Oracle Java 8, you must use at least 1.8.0_40.
5. Familiarize yourself with the changes and features in this release:
   - DataStax Enterprise release notes for 3.2.
   - General upgrade advice and Apache Cassandra features in NEWS.txt. If you are upgrading from an earlier release, read `NEWS.txt` all the way back to your current version.
   - Apache Cassandra changes in CHANGES.txt.
6. For upgrades from DataStax Enterprise 2.1.x with search nodes, see Upgrading from 2.2
7. Upgrade the SSTables on each node to ensure that all SSTables are on the current version.

   ```
   $ nodetool upgradesstables
   ```

   Warning: Failure to upgrade SSTables when required results in a significant performance impact and increased disk usage. Upgrading is not complete until the SSTables are upgraded.

   If the SSTables are already on the current version, the command returns immediately and no action is taken.
8. Back up the configuration files you use.

   The configuration files are overwritten with default values during installation of the new version.
9. Upgrade order matters. Using the following guidelines, upgrade nodes in the recommended order:
   - In multiple datacenter clusters, upgrade all the nodes within one datacenter before moving on to another datacenter.
   - Upgrade the seed nodes within a datacenter first.
   - Upgrade analytics nodes or datacenters first, then Cassandra nodes or datacenters, and finally search nodes or datacenters.
   - For analytics nodes, upgrade the Job Tracker node first. Then upgrade Hadoop nodes.

Upgrading to DataStax Enterprise 3.2

Follow these steps to upgrade to DataStax Enterprise 3.2.

1. Run nodetool drain to flush the commit log of the old installation:

   ```
   $ nodetool drain -h hostname
   ```

   This step saves time when nodes start up after the upgrade.
2. Stop the node.
3. Use the appropriate method to install the new product version:

- Upgrading RHEL installations
- Installation instructions for upgrading DataStax Enterprise on Debian-based distributions on page 54
- Installation instructions for upgrading DataStax Enterprise using the DataStax Enterprise tarball on page 55

4. To configure the new product version, use your backup configuration files to merge modifications into the configuration files for the new version.
5. **Only for upgrades from 2.2.x and 3.0.x to 3.2.x**, edit the `cassandra.yaml` file to change the partitioner setting to match the previous partitioner. The RandomPartitioner (`org.apache.cassandra.dht.RandomPartitioner`) was the default partitioner in DataStax Enterprise 2.2.x and 3.0.x which used Apache Cassandra 1.2.
6. **Only for upgrades from 3.1.x to 3.2.0**, temporarily enable the old Gossip protocol in a cluster.

   After installing the new version, but before the first restart of each node, enable the old protocol so that each upgraded node can connect to the nodes awaiting the upgrade. Add the following line to `/etc/cassandra/cassandra-env.sh` for packaged installs or *`install_location`*`/conf/cassandra-env.sh` for tarball installs:

   ```
   VM_OPTS="$JVM_OPTS -Denable-old-dse-state=true
   ```

   After upgrading the entire cluster, remove this line from `cassandra-env.sh` on each node so it uses the new protocol, and then perform a second rolling restart.
7. Start the node.
   - Installer-Services and Package installations: See Starting DataStax Enterprise as a service.
   - Installer-No Services and Tarball installations: See Starting DataStax Enterprise as a stand-alone process).
8. When the upgrade includes a major upgrade of Apache Cassandra, upgrade the SSTables now that the new version is installed.

   ```
   $ nodetool upgradesstables
   ```

   Warning: Failure to upgrade SSTables when required results in a significant performance impact and increased disk usage. Upgrading is not complete until the SSTables are upgraded.

   Apache Cassandra™ requires upgrading SSTables for major releases.

   - DataStax Enterprise 4.7 to 4.8 uses Cassandra 2.1
   - DataStax Enterprise 4.0 to 4.6 uses Cassandra 2.0
   - DataStax Enterprise 3.1 to 3.2 uses Cassandra 1.2
   - DataStax Enterprise 2.2 to 3.0 uses Cassandra 1.1
   - DataStax Enterprise 1.0 to 2.1 uses Cassandra 1.0
9. Verify that the upgraded datacenter names match the datacenter names in the keyspace schema definition:

   ```
   $ nodetool status
   ```
10. Review the logs for warnings, errors, and exceptions.

    Warnings, errors, and exceptions are frequently found in the logs when starting up an upgraded node. Some of these log entries are informational to help you execute specific upgrade-related steps. If you find unexpected warnings, errors, or exceptions, contact DataStax Support.

    For **upgrades from DataStax Enterprise 3.0.x**, ignore these expected error messages:

    - An exception that looks something like this might appear in logs during a rolling upgrade.

      ```
      ERROR 15:36:54,908 Exception in thread Thread[GossipStage:1,5,main ]
       java.lang.NumberFormatException: For input string:
       "1276058875953519237987654777869130792 96"
      . . .
      ```

- When upgrading Cassandra 1.2 nodes, messages that are related to a node that is attempting to push mutations to the new system_auth keyspace:

```
ERROR [WRITE-/192.168.123.11] 2013-06-22 14:13:42,336
 OutboundTcpConnection.java (line 222)
 error writing to /192.168.123.11
java.lang.RuntimeException: Can't serialize ColumnFamily ID
 2d324e48-3275-3517-8dd5-9a2c5b0856c5
to be used by version 5, because int <-> uuid mapping could not be
 established
(CF was created in mixed version cluster).
at
 org.apache.cassandra.db.ColumnFamilySerializer.cfIdSerializedSize(ColumnFamilySeria
```

- For upgrades on Solr nodes:

```
ERROR 00:57:17,785 Cannot activate core: ks.cf_10000_keys_50_cols
ERROR 00:57:17,786 <indexDefaults> and <mainIndex> configuration sections
 are discontinued.
 Use <indexConfig> instead.
ERROR 01:29:55,145 checksum mismatch in segments file (resource:
  ChecksumIndexInput (MMapIndexInput ( path = "/var/lib/cassandra/data/
solr.data/ks.   cf_10000_keys_50_cols/index/segments_6" )))
ERROR 01:29:55,145 Solr index ks.cf_10000_keys_50_cols seems to be
 corrupted:
  please CREATE the core again with  recovery = true to start reindexing
 data.
ERROR 01:29:55,145 Cannot activate core: ks.cf_10000_keys_50_cols
ERROR 01:29:55,146 checksum mismatch in segments file  (resource:
 ChecksumIndexInput
   (MMapIndexInput ( path = "/var/lib/cassandra/data/solr.data/ks.
 cf_10000_keys_50_cols/index/segments_6" )))
org.apache.lucene.index.CorruptIndexException: checksum mismatch in
 segments file
   (resource: ChecksumIndexInput (MMapIndexInput
   ( path = "/var/lib/cassandra/data/solr.data/ks.cf_10000_keys_50_cols/
index/segments_6" )))
```

11. Repeat the upgrade on each node in the cluster following the recommended order:
12. **Only for upgrades from 3.1.x to 3.2.0**, after the upgrade and before the first restart of each node, enable the old protocol so that each upgraded node can connect to the nodes awaiting the upgrade.

    a. Remove the following line from `/etc/cassandra/cassandra-env.sh` for packaged installs or *`install_location`*`/conf/cassandra-env.sh` for tarball installs:

    ```
    VM_OPTS="$JVM_OPTS -Denable-old-dse-state=true
    ```

    b. After removing the line from `cassandra-env.sh`, perform a second rolling restart.
13. **Only for upgrades from 3.0 and 3.1.x** When upgrading from earlier versions, the first upgraded node will automatically alter `dse_system` to use the `EverywhereStrategy` and attempt to run `nodetool repair dse_system`. This operation might fail if other nodes are down during the upgrade. Review `/var/log/cassandra/system.log` for errors or warnings. If automatic switching fails, after all the nodes are up, manually update the `dse_system keyspace` to use `EverywhereStrategy`. In `cqlsh`, enter:

    ```
    ALTER KEYSPACE dse_system WITH replication = {'class':
     'EverywhereStrategy'};
    ```

    Then enter the following command on all nodes:

    ```
    $ nodetool repair dse_system
    ```

Upgrading from 3.0 to 3.2

Review this information and follow these instructions to upgrade from DataStax Enterprise 3.0.x to DataStax Enterprise 3.2.x.

Analytics nodes

While upgrading a cluster, some column families created through Hadoop interfaces might not appear to contain data. After the upgrade process has completed, the data is visible again.

Partitioner

Edit the `cassandra.yaml` file to change the partitioner setting to match the previous partitioner. The default RandomPartitioner (org.apache.cassandra.dht.RandomPartitioner) was the default partitioner prior to Apache Cassandra™ 1.2.

CQL 3

Do not issue any CQL 3 queries until all nodes are upgraded and schema disagreements are resolved.

Security recommendations

The `client_encryption_options` for enabling client-to-node SSL have been removed from dse.yaml starting in 3.1.2. To enable client-to-node SSL, set the option in the cassandra.yaml file.

Before upgrading from 3.0.x to 3.2.x, if you use these DataStax Enterprise security features, adjust the replication strategy and options in the `cassandra.yaml` file to configure a replication factor for the `dse_auth` keyspace greater than 1:

- Kerberos
- Object permission management (internal authorization)
- Internal authentication

Adjust the replication factor for `dse_auth` on each node in the cluster. After updating the `cassandra.yaml` file and restarting the node, run `nodetool repair` to repair the first range returned by the partitioner for the keyspace:

```
$ nodetool repair dse_auth -pr
```

This should only take a few seconds to complete.

The new version of Apache Cassandra™ updates the security options. First simply merge the following settings into the new configuration files:

- authenticator
- authorizer
- auth_replication_strategy
- auth_replication_options
- any other diffs

Use the old settings while you are upgrading the cluster so that backward compatibility is maintained. For example, the new file contains the old, Cassandra 1.1 authenticator and authorizer options at this point:

- authenticator: `com.datastax.bdp.cassandra.auth.PasswordAuthenticator`
- authorizer: `org.apache.cassandra.auth.CassandraAuthorizer`

If you are upgrading a secure cluster, there may be a significant delay to each node's first startup as the security migration takes place (up to 1 minute). The delay is due to ensuring that the ring is fully connected before the migration starts. During the upgrade of a secure cluster, you may see a security related error message (documented below). You will see the following message in the log when the node has completed the migration:

```
INFO [NonPeriodicTasks:1 ] 2013-06-22 15:01:08,173
```

```
Auth.java  (line 208 ) Migration of legacy auth data is complete.
You should now switch to org.apache.cassandra.auth implementations in
 cassandra.yaml.
```

After all nodes have been upgraded, change these options to the new Cassandra 1.2 values and perform a rolling restart as explained below.

Note: If using Kerberos authentication, there are no credentials data to migrate, but user records must still be updated. Merge the related diffs from the old to the new file.

1. Edit the `cassandra.yaml` to switch to the official Apache versions of `PasswordAuthenticator` and `CassandraAuthorizer`:

   ```
   authenticator: org.apache.cassandra.auth.PasswordAuthenticator
   authorizer: org.apache.cassandra.auth.CassandraAuthorizer
   ```

2. Remove or comment out these options from the `cassandra.yaml` file:

   - auth_replication_strategy
   - auth_replication_options
   - replication_factor

   Note:

   If you have not disabled both `auth_replication_strategy` and `replication_factor`, you will see an error. For information about correcting this error, see Issues in the DataStax Enterprise 3.2.5 release notes.
3. Optionally, adjust the replication factor of the `system_auth` keyspace. The amount of data in this keyspace is typically very small, so leaving it replicated across the cluster is relatively cheap.

Virtual nodes (vnodes)

DataStax recommends using vnodes only on datacenters running Cassandra workloads. To disable vnodes on datacenters that run Hadoop or Solr workloads, set num_tokens to 1 in `cassandra.yaml`.

Solr

If you make changes to the configuration of a Solr node after upgrading, you must set the type mapping correctly as explained in Configuring the Solr type mapping version.

Recommissioning a node

If you decommissioned a node in the last 72 hours:

- Do not recommission the node until another 72 hours has passed.
- If you wish to recommission the node after 72 hours, run `nodetool gossipinfo`. Check the STATUS line for the token of the decommissioned node and verify that it does not exist. If it does not exist, then the node has been deleted and it is safe to recommission the node.
- If you need to bring the node into the cluster, contact Support on how to kill the node.

Upgrading from 2.2 to 3.2

Review this information for upgrades from DataStax Enterprise 2.2.x to 3.2.x.

Security recommendations

Upgrade the entire cluster before setting up security and then do another rolling restart.

Hadoop

The ownership of the Hadoop `mapred` staging directory in the CassandraFS has changed. After upgrading, you need to set the owner of `/tmp/hadoop-`*`dseuser`*`/mapred/staging` to the DataStax Enterprise user. For example, if you run DataStax Enterprise 3.1 as root, use the following command on Linux:

```
$ dse hadoop fs -chown root /tmp/hadoop-root/mapred/staging
```

Solr

Do not issue Solr queries after upgrading from DataStax Enterprise 2.1.x or earlier until all nodes are upgraded and schema disagreements are resolved.

Solr configuration files from previous versions of DataStax Enterprise will be invalidated by the new version of Solr included in this release. Follow these steps to update your Solr configuration file on the first Solr node you upgrade, before upgrading any other nodes:

1. Open the `system.log` file and look for the message about the Solr error.

   The error message briefly describes the changes you need to make.
2. Correct these errors in your `solrconfig.xml` files, then post the corrected files.

   Existing cores cannot be loaded until the `solrconfig.xml` errors are resolved.
3. Issue the following command to recover indexes on each upgraded Solr node. On the first node upgraded, this process should happen after the Solr configuration file has been uploaded. Note that in the command below you will need to substitute the name of your Solr core.

   ```
   $ curl -v "http://localhost:8983/solr/admin/cores?action=CREATE&solr
    core.solr&recovery=true"
   ```

The following is an example of how to perform these steps using our Solr-based demos. If you wish to do this on a test cluster, first run the `solr`, `wiki` and `logging` demos on a test cluster running the earlier version of DataStax Enterprise.

Go to the directory containing your Solr application. For example, go to the `demos` directory:

- Binary installation

  ```
  $ cd install_location/demos
  ```
- Package installation

  ```
  $ cd /usr/share/dse-demos
  ```

Run the following commands to HTTP-POST your modified custom `solrconfig.xml` to DSE Search. For example, from the `demos` or `dse-demos` directory, run the following commands:

- From the `solr_stress` directory:

  ```
  $ curl -v --data-binary @solrconfig.xml -H 'Content-type:text/xml;
   charset=utf-8'
  http://localhost:8983/solr/resource/demo.solr/solrconfig.xml
  ```
- From the `wikipedia` directory:

  ```
  $ curl -v --data-binary @solrconfig.xml -H 'Content-type:text/xml;
   charset=utf-8'
  http://localhost:8983/solr/resource/wiki.solr/solrconfig.xml
  ```
- From the `log_search` directory:

  ```
  $ curl -v --data-binary @solrconfig.xml -H 'Content-type:text/xml;
   charset=utf-8'
  http://localhost:8983/solr/resource/Logging.log_entries/solrconfig.xml
  ```

After running each `curl` command, a `SUCCESS` message appears.

This step is only required once, when the first node is upgraded.

After each node is upgraded, run the `CREATE` command with the recovery option set to true, and the distributed option set to false:

```
$ curl -v "http://localhost:8983/solr/admin/cores?
action=CREATE&name=demo.solr&recovery=true"
$ curl -v  "http://localhost:8983/solr/admin/cores?
action=CREATE&name=wiki.solr&recovery=true"
$ curl -v  "http://localhost:8983/solr/admin/cores?
action=CREATE&name=Logging.log_entries&recovery=true"
```

Partitioner

Edit the `cassandra.yaml` file to change the partitioner setting to match the previous partitioner. The default RandomPartitioner (org.apache.cassandra.dht.RandomPartitioner) was the default partitioner prior to Apache Cassandra™ 1.2.

CQL 3

Do not issue any CQL 3 queries until all nodes are upgraded and schema disagreements are resolved.

Security recommendations

The `client_encryption_options` for enabling client-to-node SSL have been removed from dse.yaml starting in 3.1.2. To enable client-to-node SSL, set the option in the cassandra.yaml file.

Before upgrading from 3.0.x to 3.2.x, if you use these DataStax Enterprise security features, adjust the replication strategy and options in the `cassandra.yaml` file to configure a replication factor for the `dse_auth` keyspace greater than 1:

- Kerberos
- Object permission management (internal authorization)
- Internal authentication

Adjust the replication factor for `dse_auth` on each node in the cluster. After updating the `cassandra.yaml` file and restarting the node, run `nodetool repair` to repair the first range returned by the partitioner for the keyspace:

```
$ nodetool repair dse_auth -pr
```

This should only take a few seconds to complete.

The new version of Apache Cassandra™ updates the security options. First simply merge the following settings into the new configuration files:

- authenticator
- authorizer
- auth_replication_strategy
- auth_replication_options
- any other diffs

Use the old settings while you are upgrading the cluster so that backward compatibility is maintained. For example, the new file contains the old, Cassandra 1.1 authenticator and authorizer options at this point:

- authenticator: `com.datastax.bdp.cassandra.auth.PasswordAuthenticator`
- authorizer: `org.apache.cassandra.auth.CassandraAuthorizer`

If you are upgrading a secure cluster, there may be a significant delay to each node's first startup as the security migration takes place (up to 1 minute). The delay is due to ensuring that the ring is fully connected before the migration starts. During the upgrade of a secure cluster, you may see a security related error message (documented below). You will see the following message in the log when the node has completed the migration:

```
INFO [NonPeriodicTasks:1 ] 2013-06-22 15:01:08,173
Auth.java  (line 208 ) Migration of legacy auth data is complete.
You should now switch to org.apache.cassandra.auth implementations in
 cassandra.yaml.
```

After all nodes have been upgraded, change these options to the new Cassandra 1.2 values and perform a rolling restart as explained below.

Note: If using Kerberos authentication, there are no credentials data to migrate, but user records must still be updated. Merge the related diffs from the old to the new file.

1. Edit the `cassandra.yaml` to switch to the official Apache versions of `PasswordAuthenticator` and `CassandraAuthorizer`:

   ```
   authenticator: org.apache.cassandra.auth.PasswordAuthenticator
   authorizer: org.apache.cassandra.auth.CassandraAuthorizer
   ```

2. Remove or comment out these options from the `cassandra.yaml` file:

   - auth_replication_strategy
   - auth_replication_options
   - replication_factor

   Note:

   If you have not disabled both `auth_replication_strategy` and `replication_factor`, you will see an error. For information about correcting this error, see Issues in the DataStax Enterprise 3.2.5 release notes.
3. Optionally, adjust the replication factor of the `system_auth` keyspace. The amount of data in this keyspace is typically very small, so leaving it replicated across the cluster is relatively cheap.

Virtual nodes (vnodes)

DataStax recommends using vnodes only on datacenters running Cassandra workloads. To disable vnodes on datacenters that run Hadoop or Solr workloads, set num_tokens to 1 in `cassandra.yaml`.

Solr

If you make changes to the configuration of a Solr node after upgrading, you must set the type mapping correctly as explained in Configuring the Solr type mapping version.

Upgrading DataStax Enterprise using the DataStax installer

This procedure shows how to upgrade to the latest version of DataStax Enterprise using the GUI installer.

The DataStax installer upgrades DataStax Enterprise and automatically performs many upgrade tasks, including:

- Draining the currently running node.
- Preserving the configuration files.
- Removing previously installed packages.
- Updating the cassandra.yaml and dse.yaml configuration files with new entries.

Prerequisites

To upgrade to the latest version of DataStax Enterprise using the DataStax Installer, ensure the following requirements are met:

- DataStax Enterprise 4.5 or 4.6 for upgrading to 4.8.
- DataStax Enterprise 4.7 or 4.8 for upgrading to 5.0.

Attention: Read and understand these instructions before upgrading.

You have probably seen the recommendation to read all the instructions. This is a time where it doing so will make a difference. By understanding what to do beforehand, you can ensure your upgrade will be smooth and avoid many pitfalls and frustrations.

Upgrading Linux installations using the DataStax installer

1. Familiarize yourself with the changes and features the release:
   - DataStax Enterprise release notes for 4.7, 4.8, or 5.0.
   - General upgrade advice and Cassandra features in NEWS.txt. If you are upgrading from an earlier version, read `NEWS.txt` all the way back to your current version.
   - Cassandra changes in CHANGES.txt.
2. Verify your current product version. If necessary, upgrade to an interim version.

Current version	Upgrade version
DataStax Enterprise 4.7 or 4.8	DataStax Enterprise 5.0
DataStax Enterprise 4.0, 4.5, or 4.6	DataStax Enterprise 4.8
DataStax Community or open source Apache Cassandra™ 2.0.x	DataStax Enterprise 4.8
DataStax Community 3.0.x	No interim version required.
DataStax Distribution of Apache Cassandra™ 3.x	Upgrade not available.

3. Verify the Java runtime version and upgrade to the recommended version.

```
$ java -version
```

DataStax Enterprise 4.7 or 4.8

The latest version of Oracle Java SE Runtime Environment 7 or 8 or OpenJDK 7 is recommended. The JDK is recommended for development and production systems. The JDK provides useful troubleshooting tools that are not in the JRE, such as jstack, jmap, jps, and jstat.

Note: If using Oracle Java 7, you must use at least 1.7.0_25. If using Oracle Java 8, you must use at least 1.8.0_40.

DataStax Enterprise 5.0

The latest version of Oracle Java SE Runtime Environment 8 (JDK) (1.8.0_40 minimum) or OpenJDK 8 is recommended. The JDK is recommended for development and production systems. The JDK provides useful troubleshooting tools that are not in the JRE, such as jstack, jmap, jps, and jstat.

4. Download the installer for your computer from the DataStax downloads page.
5. From the directory where you downloaded the install file, make it executable and run it using the sudo command.

```
$ chmod +x DataStaxEnterprise-version_number-linux-x64-installer.run ##
 Changes permission to executable
$ sudo ./DataStaxEnterprise-version_number-linux-x64-installer.run
```

6. Follow the instructions in the setup wizard. For a detailed description of the settings in the wizard, see the installation instructions in 4.7, 4.8, or 5.0.
7. Start DataStax Enterprise:

```
$ sudo service dse start ## Starts the DataStax Enterprise server
```

8. (DataStax Enterprise 4.7 and 4.8 only) Start the Datastax Agent:

```
$ sudo service datastax-agent start
```

   Note: For DataStax Enterprise 5.0, see the OpsCenter 6.0 documentation.
9. Verify that DataStax Enterprise is running:

```
$ nodetool status
```

Rolling back an upgrade

This section describes how to revert DataStax Enterprise to an earlier version.

Revert to a previous version from a package installation

Procedure

1. Uninstall all DataStax Enterprise packages.
   - **Debian and Ubuntu**

     ```
     # apt-get remove dse-full
     ```
   - **RHEL and CentOS**

     ```
     # yum remove dse-full
     ```
2. Restore the snapshot taken before the upgrade by copying the SSTable files from the snapshot directory to the data directory of each column family. If you have multiple snapshots, look at the timestamp to find the most recent one. Data that was inserted after the snapshot was taken is not restored.

   In the following example, the snapshot directory is *data_directory_location*/*keyspace_name*/*table_name*/snapshots/*snapshot_name* and the data directory is /data.

   ```
   # cd data_directory_location/keyspace_name/table_name/
   snapshots/snapshot_name
   # cp  -R * data_directory_location/keyspace_name/table_name
   ```
3. Reinstall the old version as described in the documentation for that release of DataStax Enterprise.
4. If you are using Solr, rebuild the index as described in Re-indexing in full.

Revert to a previous version from a tarball installation

Procedure

1. Rename the current installation directory.

   ```
   # mv dse4.0 dse4.0.bak
   ```
2. Restore the snapshot taken before the upgrade by copying the SSTable files from the snapshot directory to the data directory of each column family. If you have multiple snapshots, look at the

timestamp to find the most recent one. Data that was inserted after the snapshot was taken is not restored.

In the following example, the snapshot directory is *data_directory_location*/*keyspace_name*/*table_name*/snapshots/*snapshot_name* and the data directory is /data.

```
# cd data_directory_location/keyspace_name/table_name/
snapshots/snapshot_name
# cp  -R * data_directory_location/keyspace_name/table_name
```

3. Copy the old cassandra.yaml file from the old install directory to the new one.

```
# cp dse4.0.bak/resources/cassandra/config/conf/cassandra.yaml
 <new_install_dir>/resources/cassandra/config/conf/
```

4. Reinstall the old version as described in the documentation for that release of DataStax Enterprise.
5. If you are using Solr, rebuild the index as described in Re-indexing in full.

Upgrading from DataStax Community to DataStax Enterprise

Attention: Read and understand these instructions before upgrading.

You have probably seen the recommendation to read all the instructions. This is a time where it doing so will make a difference. By understanding what to do beforehand, you can ensure your upgrade will be smooth and avoid many pitfalls and frustrations.

Before upgrading to DataStax Enterprise from any DataStax Community version, complete these steps on each node in your cluster.

Prerequisites

DataStax recommends backing up your data prior to any version upgrade. A backup provides the ability to revert and restore all the data used in the previous version if necessary. See Backing up and restoring data.

Procedure

1. Familiarize yourself with the changes and features in this release:
   - DataStax Enterprise release notes for 4.5, 4.6, 4.7, 4.8, and 5.0.
   - General upgrade advice and Cassandra features in NEWS.txt. If you are upgrading from an earlier release, read NEWS.txt all the way back to your current version.
   - Ensure that your version of DataStax Community can be upgraded directly to the version of Cassandra that is used by DataStax Enterprise. See the Cassandra changes in CHANGES.txt.
2. Before you perform the DataStax Enterprise upgrade that includes a major upgrade of Apache Cassandra, upgrade the SSTables on each node to ensure that all SSTables are on the current version.

```
$ nodetool upgradesstables
```

If the SSTables are already on the current version, the command returns immediately and no action is taken.

3. Run nodetool drain to flush the commit log of the old installation:

   ```
   $ nodetool drain -h hostname
   ```

   This step saves time when nodes start up after the upgrade, and prevents DSE Search nodes from having to re-index data.
4. Stop the node (4.7, 4.8, 5.0).
5. Back up your configuration files.

   Depending on how you install the product, the configuration files might be overwritten with default values during the installation.
6. Uninstall DataStax Community.

   If you installed the DataStax Community from packages in APT or RPM repositories, you must remove DataStax Community before setting up and installing from the appropriate repository.

   - For packages installed from APT repositories:

     ```
     $ sudo apt-get remove "dsc*" "cassandra*" "apache-cassandra*"
     ```

     This action shuts down Cassandra if it is still running.
   - For packages installed from Yum repositories:

     ```
     $ sudo yum remove "dsc*" "cassandra*" "apache-cassandra*"
     ```

     The old Cassandra configuration file might be renamed to `cassandra.yaml.rpmsave`, for example:

     ```
     warning: /etc/cassandra/default.conf/cassandra.yaml
     saved as /etc/cassandra/default.conf/cassandra.yaml.rpmsave
     ```
7. Install the new product using one of the following:

   - Installation instructions for upgrading DataStax Enterprise using the DataStax Enterprise tarball on page 55
   - Installation instructions for upgrading DataStax Enterprise on Debian-based distributions on page 54
   - Upgrading RHEL installations
8. To configure the product, use your backup configuration files to merge any necessary modifications into the configuration files for the new version.
9. Depending on the version you are upgrading to, convert the snitches.

   - Starting in DataStax Enterprise 4.6, the endpoint snitch is set in cassandra.yaml, not dse.yaml. The `com.datastax.bdp.snitch.DseDelegateSnitch` is replaced by `com.datastax.bdp.snitch.DseSimpleSnitch` in `cassandra.yaml` and the `endpoint_snitch` option has been removed from `dse.yaml`.
   - For upgrades to versions earlier than 4.6, the snitch in DataStax Enterprise is set in dse.yaml. Convert the snitches from cassandra.yaml to dse.yaml.

endpoint_snitch URL	Upgrade task
org.apache.cassandra.locator.SimpleSnitch	Leave the DseDelegateSnitch as set in the `cassandra.yaml` file and leave the default delegated_snitch in the new `dse.yaml` file unchanged.
org.apache.cassandra.locator.PropertyFileSnitch	Copy/paste the `cassandra-topology.properties` file from the old installation to *`install_location`*`/resources/cassandra/conf`, overwriting the new properties file. Set the

endpoint_snitch URL	Upgrade task
	delegated_snitch setting in the new `dse.yaml` file to:`org.apache.cassandra.locator.PropertyFileSnitch`.
Any other snitch URL	Change the default delegated_snitch in the new `dse.yaml` file to your current snitch setting.

For DataStax Enterprise 4.6 and earlier only, the default delegated_snitch (`com.datastax.bdp.snitch.DseSimpleSnitch`) is specified in the dse.yaml file.

10. Start the node.
    - Installer-Services and Package installations: See *Starting DataStax Enterprise as a service* (4.7, 4.8, 5.0).
    - Installer-No Services and Tarball installations: See *Starting DataStax Enterprise as a stand-alone process* 4.7 or 4.8, 5.0).
11. When your DataStax Enterprise upgrade includes a major upgrade of Apache Cassandra, upgrade the SSTables on each node now that the upgrade is complete.

    ```
    $ nodetool upgradesstables
    ```

    Apache Cassandra™ requires upgrading SSTables for major releases.
    - DataStax Enterprise 5.0 uses Cassandra 3.0 (not 3.x)
    - DataStax Enterprise 4.7 to 4.8 uses Cassandra 2.1
    - DataStax Enterprise 4.0 to 4.6 uses Cassandra 2.0
    - DataStax Enterprise 3.1 to 3.2 uses Cassandra 1.2
    - DataStax Enterprise 2.2 to 3.0 uses Cassandra 1.1
    - DataStax Enterprise 1.0 to 2.1 uses Cassandra 1.0
12. Verify that the upgraded datacenter names still match the datacenter names that are used in the keyspace schema definition:

    ```
    $ nodetool status
    ```
13. Be sure to review the logs for warnings, errors, and exceptions.

    Warnings, errors, and exceptions are frequently found in the logs when starting up an upgraded node. Some of these log entries are informational to help you execute specific upgrade-related steps. If you find unexpected warnings, errors, or exceptions, contact DataStax Support.
14. Repeat the upgrade on each node in the cluster following the recommended upgrade order.

Upgrading Apache Cassandra™

This section describes how to upgrade Apache Cassandra.

NEWS.txt contains the latest information on upgrading specific versions. When upgrading from an earlier release, read `NEWS.txt` all the way back to your current version.

Attention: Read and understand these instructions before upgrading.

You have probably seen the recommendation to read all the instructions. This is a time where it doing so will make a difference. By understanding what to do beforehand, you can ensure your upgrade will be smooth and avoid many pitfalls and frustrations.

Apache Cassandra™ versions requiring intermediate upgrades

Be sure to read the NEWS.txt for each version back to your current version.

Apache Cassandra 3.0.x restrictions

- Upgrade from Cassandra 2.2 versions later or equal to 2.2.2 directly to Cassandra 3.0.x.
- Upgrade from Cassandra 2.1 versions later or equal to 2.1.9 directly to Cassandra 3.0.x.
- Direct upgrade from Cassandra 2.0 and older versions is not supported.

Apache Cassandra 2.2.x restrictions

- Upgrade from Cassandra 2.1 versions later or equal to 2.1.9 directly to Cassandra 2.2.x.
- Direct upgrade from Cassandra 2.0 and older versions is not supported.

Apache Cassandra 2.1.x restrictions

- Upgrade from Cassandra 2.0.7 or later directly to Cassandra 2.1.
- Cassandra 2.1 is not compatible with Cassandra 1.x SSTables. To upgrade:

  1. Upgrade the SSTables.
  2. Upgrade the nodes to Cassandra 2.0.7 or later.
  3. Start the cluster.
  4. Upgrade the SSTables again.
  5. Stop the cluster.
  6. Upgrade to Cassandra 2.1.

Apache Cassandra 2.0.x restrictions

- Upgrade to Cassandra 2.0.x directly from Cassandra 1.2.9 or later.
- Upgrade from Cassandra earlier than 1.2.9 to Cassandra 2.0.x using a rolling restart:

  1. Upgrade the SSTables.
  2. Upgrade the entire cluster to 1.2.9
  3. Upgrade to Cassandra 2.0.

Changes impacting upgrade to Apache Cassandra™ 2.1

Changes that can affect upgrading to Apache Cassandra 2.1.x are:

- The option to omit cold SSTables with size-tiered compaction has been removed. It is almost always preferable to use date-tiered compaction for workloads that have cold data.
- `CAS` and new features in CQL such as `DROP COLUMN` assume that cell timestamps are microseconds-since-epoch.

  Do not use these features if you are using client-specified timestamps with some other source.
- Unknown keyspace replication options are no longer accepted.
- Apache Cassandra 2.0 and 2.1 use a new version of CQL (and `cqlsh`) based on the CQL specification 3.1.0.
- Use lowercase property map keys in `ALTER` and `CREATE` statements.

  In earlier releases, CQL property map keys used in `ALTER` and `CREATE` statements were case-insensitive. For example, `CLASS` or `class` and `REPLICATION_FACTOR` or `replication_factor` were permitted. The case sensitivity of the property map keys was inconsistent with the treatment of other string literals and incompatible with formatting of `NetworkTopologyStrategy` property

maps, which have case-sensitive datacenter names. In 2.1, property map keys such as `class` and `replication_factor` are case-sensitive. Lowercase property map keys are shown in this example:

```
CREATE KEYSPACE test WITH replication =
  { 'class' : 'SimpleStrategy', 'replication_factor' : '1' };
```

- You might need to fix queries that have loose type validation of CQL constants that now have strong validation.

  Using BLOBs as string constants is deprecated in favor of blob constants.
- Virtual nodes (vnodes) are enabled by default in the 2.0 and later cassandra.yaml. Disable vnodes before upgrading clusters that do not use vnodes.
- `auto_bootstrap` of a single-token node with no `initial_token` now picks a random token instead of bisecting an existing token range.

  Using vnodes is recommended after completing the upgrade; otherwise, specify an initial token.
- `reduce_cache_sizes_at`, `reduce_cache_capacity_to`, and `flush_largest_memtables_at` options have been removed from `cassandra.yaml`.
- `CacheServiceMBean.reduceCacheSizes()` has been removed. Use `CacheServiceMBean.set{Key,Row}CacheCapacityInMB()` instead.
- `authority` option in `cassandra.yaml` has been deprecated since 1.2.0, but it has been completely removed starting in 2.0. Use the `authorizer` option.
- index_interval is now a CQL table property. You can change the value of `index_interval` after upgrading using `ALTER TABLE`.

  During the upgrade, Apache Cassandra uses the value defined in old `cassanda.yaml` as the default for upgraded tables.
- The deprecated `native_transport_min_threads` option has been removed in `cassandra.yaml`.
- **Be sure** to upgrade the SSTables (`nodetool upgradesstables`). Failure to upgrade SSTables when required results in a significant performance impact and increased disk usage. Upgrading is not complete until the SSTables are upgraded.

Changes that apply only to upgrading to Apache Cassandra 2.0.x

- The `nodetool upgradesstables` command upgrades/rewrites only the SSTables that are not on the current version, which is usually what you want.

  Use the new `-a` flag to recover the old behavior of rewriting all SSTables.
- Tables using `LeveledCompactionStrategy` do not create a row-level bloom filter by default.

  In Cassandra versions earlier than 1.2.2, the default value differs from the current value. Manually set the false positive rate to 1.0 (to disable) or 0.01. (Enable, if you make many requests for rows that do not exist.)

Guidelines and general Apache Cassandra™ upgrade steps

Best practices

- Regular node maintenance: periodically running nodetool repair ensures that data on each replica is consistent with data on other nodes.
- Employ a continual upgrade strategy for each year. Upgrades are impacted by the version you are upgrading from and the version you are upgrading to. The greater the gap between the current version and the target version, the more complex the upgrade.
- While the cluster is in a partially upgraded state, observe the upgrade limitations.

- DataStax recommends backing up your data prior to any version upgrade. A backup provides the ability to revert and restore all the data used in the previous version if necessary. See Backing up and restoring data.

Attention: Read and understand these instructions before upgrading.

You have probably seen the recommendation to read all the instructions. This is a time where it doing so will make a difference. By understanding what to do beforehand, you can ensure your upgrade will be smooth and avoid many pitfalls and frustrations.

General upgrade procedures

1. Take a snapshot of all keyspaces before the upgrade.

   You can rollback to the previous version if necessary. Cassandra is able to read data files created by the previous version, but the inverse is not always true. Taking a snapshot is fast, especially if you have JNA installed, and takes effectively zero disk space until you start compacting the live data files again.
2. Make sure any client drivers, such as Hector or Pycassa clients, are compatible with the new version.
3. Familiarize yourself with the changes and features in this release:
   - General upgrade advice and Apache Cassandra features in NEWS.txt. When upgrading from an earlier release, read `NEWS.txt` all the way back to your current version.
   - Cassandra changes in CHANGES.txt.
4. Run nodetool drain before shutting down the existing Cassandra service. This prevents overcounts of counter data, and also speeds up restart post-upgrade.
5. Follow the instructions in Upgrade procedures.
6. Monitor the log files for any issues.
7. After upgrading and restarting all Cassandra processes, restart client applications.
8. After upgrading all nodes in the cluster, consider upgrading existing nodes to vnodes.

   Upgrading to vnodes is optional but has a number of important advantages. See Enabling virtual nodes on an existing production cluster.

General upgrade limitations for Apache Cassandra™

Upgrade limitations

Limitations apply while a cluster is in a partially upgraded state.

With these exceptions, the cluster continues to work as though it were on the earlier version of DataStax Enterprise until all of the nodes in the cluster are upgraded.

General upgrade restrictions

- **Do not** enable new features.
- Do not run `nodetool repair`.
- Do not issue these types of CQL queries during a rolling restart: `DDL` and `TRUNCATE`.
- During upgrades, the nodes on different versions might show a schema disagreement.
- Failure to upgrade SSTables when required results in a significant performance impact and increased disk usage. Upgrading is not complete until the SSTables are upgraded.

Upgrading procedures for Apache Cassandra™

General upgrade restrictions

- **Do not** enable new features.

- Do not run `nodetool repair`.
- Do not issue these types of CQL queries during a rolling restart: `DDL` and `TRUNCATE`.
- During upgrades, the nodes on different versions might show a schema disagreement.
- Failure to upgrade SSTables when required results in a significant performance impact and increased disk usage. Upgrading is not complete until the SSTables are upgraded.

Prerequisites

- Apache Cassandra 2.0.x and 2.1.x - The latest version of the Java SE Runtime Environment (JRE) 7 is required. The JDK is recommended.
- Apache Cassandra 2.2.x, 3.0.x, and 3.x - The latest version of the Java SE Runtime Environment (JRE) 8 is required. The JDK is recommended.
- Remove all dead nodes.

  Do not upgrade if nodes in the cluster are down. Use the nodetool removenode command to remove dead nodes.

  Tip: In earlier releases, `nodetool removenode` was `nodetool removetoken`.
- In the old cassandra.yaml, check the value of index_interval and change it if you want a different default value applied to tables during the upgrade. In Cassandra 2.0 and later, `index_interval` has been moved out of the `cassandra.yaml` and is now a table property.

  During the upgrade, the value defined in the old `cassandra.yaml` is applied as the default property to your upgraded tables. You can use CQL to alter this property after the upgrade.
- If your cluster does **not** use virtual nodes (vnodes), disable vnodes in each new `cassandra.yaml` before doing the rolling restart. To disable:

  1. In the cassandra.yaml file, set `num_tokens` to 1.
  2. Uncomment the `initial_token` property and set it to 1 or to the value of a generated token for a multi-node cluster.

  Note: Vnodes are enabled by default starting with Cassandra 2.0.0.). Vnodes are not enabled or did not exist in earlier Cassandra versions. To switch to vnodes in existing cluster, see *Enabling virtual nodes on an existing production cluster* for your Cassandra version.

Procedure

1. Familiarize yourself with the changes and features in this release:
   - General upgrade advice and Cassandra features in NEWS.txt. If you are upgrading from an earlier release, read `NEWS.txt` all the way back to your current version.
   - Cassandra changes in CHANGES.txt.
2. When your Apache Cassandra upgrade includes a major version, such as Cassandra 2.1 to 3.0, or a major point release, such as Cassandra 2.0 to 2.1, upgrade the SSTables on each node to ensure that all SSTables are on the current version. If the SSTables are already on the current version, the command returns immediately and no action is taken.

   ```
   $ nodetool upgradesstables
   ```

   Warning: Failure to upgrade SSTables when required results in a significant performance impact and increased disk usage. Upgrading is not complete until the SSTables are upgraded.
3. Run nodetool drain before shutting down the existing Cassandra service. This prevents overcounts of counter data, and also speeds up restart post-upgrade.
4. Stop the node.
5. Back up your configuration files.

Depending on how you install the product, these files might get overwritten with default values during the installation. After backing up your configuration, follow the appropriate installation instructions depending on your current installation type.

6. Install the new version of Apache Cassandra:
   - Debian-based installations
   - RHEL-based installations
   - Tarball installations
7. Configure the new product.

   Using the backups you made of your configuration files, merge any modifications you have previously made into the new configuration files for the new version. Configuration options change often, so be sure to double check the version restrictions for additional steps and changes regarding configuration.

   Ensure that the latest default values from `cassandra-env.sh` match your local `cassandra-env.sh` file.
8. Start the node.

   An explicit compatibility check is performed on startup. If you skipped any upgrade step, startup fails. Changes from the new release are not modified, and SSTables are not upgraded to the new format. Review the error messages to correct the problem and then start the node.
9. One each node after the upgrade is performed, run `$ nodetool upgradesstables`.

   Upgrading SSTables is required for Cassandra upgrades that include a major version (for example, from Cassandra 1.2 to 2.0) or a major point release (for example, from Cassandra 2.0 to 2.1).
10. Check the logs for warnings, errors, and exceptions.
11. Repeat these upgrade steps on each node in the cluster.

Upgrading Apache Cassandra™ RHEL-based installations

Follow these steps to remove the old Apache Cassandra installation, merge your customizations of the old configuration file to the new one, and then complete the upgrade.

Procedure

1. Save the configuration files from the old installation to a safe place.
2. On each Cassandra node, remove the old installation. For example:

   ```
   $ sudo yum remove dsc20
   ```
3. Install the new version.

   ```
   $ sudo yum install dsc<i>version_number</i>
   ```

   The installer creates the file `cassandra.yaml.rpmnew` in `/etc/cassandra/default.conf/`.
4. Open the old and new configuration files and diff them.
5. Merge the diffs by hand, including the partitioner setting, from the old file into the new one.

   Unless your old release uses the `Murmur3Partitioner`, do not use the default partitioner setting in the new `cassandra.yaml`, `Murmur3Partitioner`. The `Murmur3Partitioner` can only be used for new clusters. After data has been added to the cluster, you cannot change the partitioner without reworking tables, which is not practical. Use your old partitioner setting in the new `cassandra.yaml` file.
6. Save the new configuration files.

Upgrading Apache Cassandra™ Debian-based installations

Follow these steps to get the new version, merge your customizations of the old configuration files and to the new ones, and then complete the upgrade.

Procedure

1. Save the configuration files from the old installation to a safe place.
2. On each of your Cassandra nodes, install the new version.

```
$ sudo apt-get install cassandra-version_number
```

3. Open the old and new configuration files and diff them.
4. Merge the diffs by hand, including the partitioner setting, from the old file into the new one.

   Do not use the default partitioner setting in the new `cassandra.yaml` because it has changed in this release to the `Murmur3Partitioner`. The `Murmur3Partitioner` can only be used for new clusters. After data has been added to the cluster, you cannot change the partitioner without reworking tables, which is not practical. Use your old partitioner setting in the new `cassandra.yaml` file.
5. Save the configuration files.

Upgrading Apache Cassandra™ Tarball installations

Follow these steps to download and unpack the Apache Cassandra binary tarball, merge your customizations of the old `cassandra.yaml` file into the new one, and then complete the upgrade.

Procedure

1. Save the configuration files from the old installation to a safe place.
2. On each node, download and unpack the binary tarball package from Planet Cassandra.
3. Open the configuration files in both the new and old installations.
4. Diff the new and old configuration files.
5. Merge the diffs, including the partitioner setting, by hand from the old file into the new one.

   Do not use the default partitioner setting in the new cassandra.yaml because it has changed in this release to the `Murmur3Partitioner`. The `Murmur3Partitioner` can only be used for new clusters. After data has been added to the cluster, you cannot change the partitioner without reworking tables, which is not practical. Use your old partitioner setting in the new `cassandra.yaml` file.
6. On RHEL or CentOS 5 platforms only, replace the `snappy-java-1.5.0.jar` with version 1.4.1.1 of Snappy available here.

```
$ rm lib/snappy-java-1.0.5.jar
$ cd lib
$ curl -OL https://snappy-java.googlecode.com/files/snappy-java-1.0.4.1.jar
```

Upgrading DataStax OpsCenter

Use the information in this section to upgrade to OpsCenter 6.0 from earlier versions.

Note: Upgrading from OpsCenter versions 4.x to version 5.2 requires upgrading to version 5.1 first. OpsCenter 6.0 does not support Apache Cassandra™.

OpsCenter product compatibility

See OpsCenter compatibility with DataStax Enterprise and Apache Cassandra compatibility.

Upgrading package installations

These steps provide information on upgrading to OpsCenter 6.0 for package installs and restarting the `opscenterd` daemon.

Note:

- You must upgrade to version 5.1 or later before upgrading to OpsCenter 6.0.
- Before upgrading to OpsCenter 5.2, you must upgrade OpsCenter 4.1 or earlier versions to OpsCenter 5.1. Follow these same steps to upgrade to 5.1 and startup opscenterd 5.1 successfully one time before proceeding to upgrade to OpsCenter 5.2.

Prerequisites

- Review 6.0 upgrade considerations for changes in configuration files, metrics, and APIs that affect upgrades to OpsCenter 6.0.
- Review 5.2 upgrade considerations for updates to configuration options and precedence behavior for OpsCenter configuration files.
- Review 5.1 configuration considerations if you need to upgrade to 5.1 from versions 4.1 or earlier before upgrading to 5.2.

Important: Before upgrading to version 6.0, be sure to back up your OpsCenter keyspace if you anticipate, plan, or need to downgrade to your earlier version of OpsCenter. Downgrading OpsCenter is a very manual and case-specific process. If you require a downgrade, please contact DataStax Support for assistance before proceeding.

Procedure

1. Be sure that OpsCenter is compatible with your version of DataStax Enterprise.
2. On the OpsCenter daemon host, run the appropriate command to update the packages:
   - **Debian or Ubuntu**

     ```
     # apt-get update
     ```
   - **RHEL or CentOS**

     ```
     # yum clean all
     ```
3. Install the upgraded OpsCenter package:
   - **Debian or Ubuntu**:

     ```
     # apt-get install opscenter
     ```
   - **RHEL or CentOS**:

     ```
     # yum install opscenter
     ```
4. If the package manager prompts you for options regarding `opscenterd.conf`, choose to keep your currently installed version.
5. Restart the OpsCenter daemon.

```
# service opscenterd restart
```

Upgrading tarball installations

These steps provide information on upgrading to OpsCenter 6.0 for tarball installs and restarting the `opscenterd` daemon.

Note:

- You must upgrade to version 5.1 or later before upgrading to OpsCenter 6.0.
- Before upgrading to OpsCenter 5.2, you must upgrade OpsCenter 4.1 or earlier versions to OpsCenter 5.1. Follow these same steps to upgrade to 5.1 and startup opscenterd 5.1 successfully one time before proceeding to upgrade to OpsCenter 5.2.

Prerequisites

- Review 6.0 upgrade considerations for changes in configuration files, metrics, and APIs that affect upgrades to OpsCenter 6.0.
- Review 5.2 upgrade considerations for updates to configuration options and precedence behavior for OpsCenter configuration files.
- Review 5.1 configuration considerations if you need to upgrade to 5.1 from versions 4.1 or earlier before upgrading to 5.2.

Important: Before upgrading to version 6.0, be sure to back up your OpsCenter keyspace if you anticipate, plan, or need to downgrade to your earlier version of OpsCenter. Downgrading OpsCenter is a very manual and case-specific process. If you require a downgrade, please contact DataStax Support for assistance before proceeding.

Procedure

1. Be sure that OpsCenter is compatible with your version of DataStax Enterprise.
2. Download and extract the new tarball.
3. Copy the following files and directories from the old tarball installation directory to the new one.

```
conf/clusters/*
conf/event-plugins/*
conf/install_id
conf/logback.xml (6.0+)
conf/opscenterd.conf
./passwd.db
./lcm.db (6.0+)
```

4. If SSL is enabled, copy the contents of the SSL configuration directory: *`install_location`*`/ssl/*`.
5. Stop the `opscenterd` instance (if it is running) and start it from the new tarball installation directory.
6. >Upgrade the agents either through the GUI or by manually installing from the new tarballs.

Upgrading OpsCenter when failover is enabled

Follow this process when upgrading OpsCenter and failover is enabled.

Procedure

1. Stop the secondary (backup) OpsCenter instance.
2. Upgrade the primary OpsCenter instance:
   a) Stop OpsCenter.
   b) Upgrade OpsCenter using the package or tarball instructions as appropriate.
   c) Start OpsCenter.
3. Upgrade the secondary Opscenter instance.
4. Start the secondary OpsCenter instance.

Upgrading from the former standalone installer

Standalone installer background

The standalone installer for OpsCenter was removed in the OpsCenter 6.0 release. Follow these instructions to upgrade an OpsCenter instance installed from the former standalone installer in OpsCenter versions 5.1.x through 5.2.x to OpsCenter 6.0 and later. Upgrade to OpsCenter 6.0 and later using either the package or tarball installs instead of the former standalone installer.

Note: DataStax recommends using a package installation if possible for simplification of future upgrades.

When using the standalone installer in versions of OpsCenter earlier than 6.0, all of the OpsCenter contents go into a single directory similar to a tarball install. The default is `/usr/share/opscenter`; however, that directory can be configured during an installation. If the user is running with sudo or root permissions during installation, the installer gives the option to set up services. Setting up as a service creates an `init.d` script so users can run `sudo service opscenterd` to start or stop OpsCenter. The service creates a symlink for paths used by packages to point to the path previously mentioned; for example, `/var/log/opscenter` has a symbolic link to `/usr/share/opscenter/logs`.

Tip: The tarball install does not include the `/etc/init.d` script for running OpsCenter as a service. To continue running OpsCenter as a service with the script, install OpsCenter from a package.

Prerequisites

- Review 6.0 upgrade considerations for changes in configuration files, metrics, and APIs that affect upgrades to OpsCenter 6.0.
- Use the Backup Service applicable to your currently installed version of OpsCenter (5.1.x or 5.2.x) to back up the OpsCenter keyspace.

  Important: Before upgrading to version 6.0, be sure to back up your OpsCenter keyspace if you anticipate, plan, or need to downgrade to your earlier version of OpsCenter. Downgrading OpsCenter is a very manual and case-specific process. If you require a downgrade, please contact DataStax Support for assistance before proceeding.
- The Backup Service does not back up config files. Back up your config files in the `/conf` directories. If applicable, back up the SSL configuration. If OpsCenter authentication is enabled, back up the password database `./passwd.db`. Config and other files to back up:

  ```
  conf/clusters/*
  conf/event-plugins/*
  conf/install_id
  conf/opscenterd.conf
  ./passwd.db
  ```
- Uninstall the OpsCenter instance using the appropriate installer script located in *`install_location`*`/opscenter/bin/`, or use the former standalone uninstaller.

Uninstall OpsCenter

1. Go to the OpsCenter installation directory.
2. Launch the uninstaller:
   - Linux:

     ```
     $ ./uninstall ## Run the uninstaller as root or sudo if needed
     ```
   - Mac OS X: In the *install_location*`/opscenter/`, double-click **uninstaller**.
3. A dialog prompts you to confirm the type of uninstall.

   CAUTION: Be sure you have backed up the files according to the recommended prerequisites before proceeding.

   Click **Yes** to confirm fully uninstalling the application.

Upgrade from the standalone installer to a package

1. Follow the recommended prerequisites.
2. Follow the steps to install OpsCenter using the package method of your choice: Debian or RPM.
3. After installing the package, copy the files below to `/etc/opscenter/`:

   ```
   conf/clusters
   conf/event-plugins
   conf/install_id
   conf/opscenterd.conf
   ./passwd.db
   ```

   For example:

   ```
   $ cp -r conf/clusters /etc/opscenter/
   ```
4. If SSL is enabled, copy the contents of the SSL configuration directory:

   ```
   $ cp -r /path/to/backup/ssl /var/lib/opscenter/
   ```
5. Restart OpsCenter.

Upgrade from the standalone installer to a tarball

1. Follow the recommended prerequisites.
2. Follow the instructions to install OpsCenter with the tarball.
3. After installing the tarball, copy the files below to *install_location*`/conf/`:

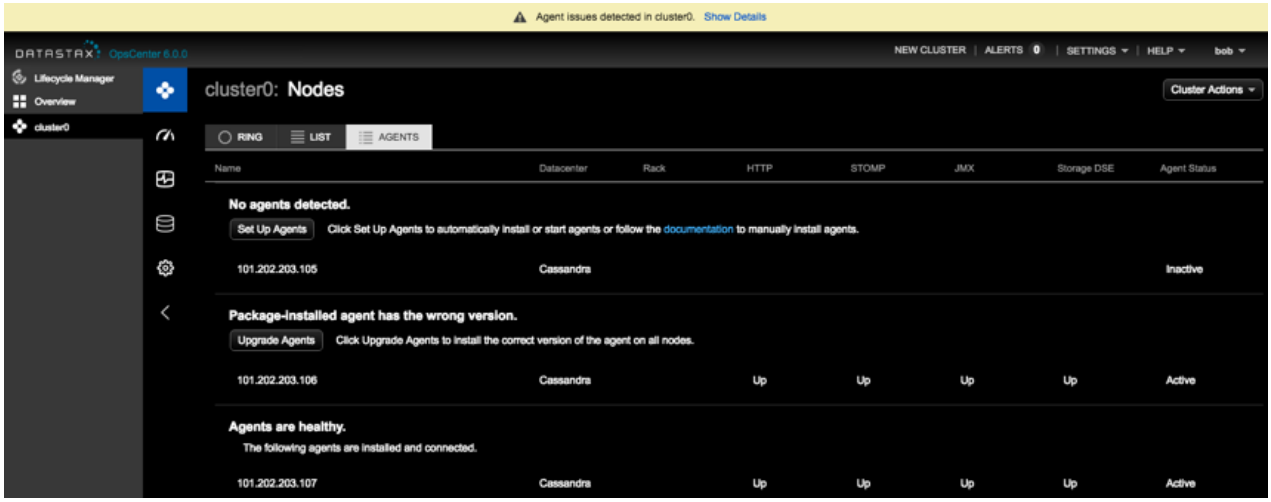
   ```
   conf/clusters
   conf/event-plugins
   conf/install_id
   conf/opscenterd.conf
   ./passwd.db
   ```
4. If SSL is enabled, copy the contents of the SSL configuration directory:

   ```
   $ cp -r /path/to/backup/ssl /path/to/install_location/ssl
   ```
5. Restart OpsCenter.

Upgrading agents

Upgrade the DataStax agents on each node in the managed clusters after restarting the upgraded OpsCenter daemon.

If DataStax agents require upgrading for the new release, you are prompted to do so by an **Upgrade Agents** button in the Agents view:



For more information, see Installing DataStax Agents.

Upgrading agent manually from tarballs

If the upgrading the agents manually with tarballs, copy the new `agent.tar.gz` to all nodes, extract it, and copy the following files from the old agent tarball directories to the new ones:

```
conf/*
ssl/*
```

OpsCenter 6.0 upgrade considerations

Review the changes in configuration files, metrics, and APIs that affect upgrades to OpsCenter 6.0.

SSL configuration changes

In the `[cassandra]` section of *`cluster_name`*`.conf`:

- `ssl_ca_certs` has been replaced by `ssl_keystore` and `ssl_keystore_password`.
- `ssl_client_pem` and `ssl_client_key` have been replaced by `ssl_truststore` and `ssl_truststore_password`.
- The `ssl_validate` option has been removed.

See troubleshooting SSL connections.

LDAP configuration changes

Because OpsCenter uses the Java driver, the Python LDAP library has been replaced with the Java LDAP library. As of OpsCenter 6.0, OpsCenter uses a keystore/truststore to manage any SSL/TLS requirements. For LDAP to work properly, migrate to the new configuration parameters.The following cluster configuration parameters have been removed:

- `ssl_cacert`
- `ssl_cert`

- ssl_key
- tls_reqcert
- tls_demand
- debug_ssl
- opt_referrals

The removed configuration options have been replaced with:

- truststore
- truststore_type
- truststore_pass

User password hash for OpsCenter authentication

The default user password hash (sha256) for OpsCenter versions earlier than 6.0 has been deprecated. The default as of OpsCenter 6.0 is bcrypt+blake2b-512. If you want to use an option other than the default, see changing the hash algorithm. Upgrading to OpsCenter 6.0 automatically migrates the user password hash to the new default. When users log in to OpsCenter for the first time after upgrading, their passwords are converted to the new hash.

Password database ownership

When installed with Debian packages, opscenterd now properly runs as the opscenter user instead of root. Because this can cause ownership issues with passwd.db, the 6.0.0 package install attempts to automatically chown it. Those using Debian packages and a custom path for passwd.db need to check and possibly change the ownership of that file to ensure it has read and write permissions by the opscenter user. This is caused by the aforementioned bug fix that allows opscenterd to run as the opscenter user as expected.

Logging configuration

All logging configuration is now done within logback.xml. The following options have been removed from opscenterd.conf:

- [logging] level
- [logging] log_path
- [logging] log_length
- [logging] max_rotate
- [authentication] audit_auth
- [authentication] audit_pattern
- [repair_service] log_directory
- [repair_service] log_length
- [repair_service] max_rotate
- [webserver] log_path

In addition to the configuration file options, the OPSCENTERD_LOG_STDOUT environment variable has also been removed. Enabling console logging is also configured in logback.xml. For more information, see configuring logback.xml in OpsCenter.

Kerberos

Kerberos JCE prerequisite: If using Kerberos with 256-bit encryption, ensure the JCE is installed on the opscenterd machine. For information on installing the JCE, see AES-256 support.

Kerberos configuration options: New configuration options were added to opscenterd.conf to support Kerberos connections in OpsCenter using the DataStax Java Driver for Apache Cassandra™:

- `opscenterd_keytab_location`: Full path to the keytab containing keys for the `opscenterd_client_principal` on the OpsCenter machine.
- `debug`: Whether to output debug messages during Kerberos connection attempts from OpsCenter.

New configuration options were added to `address.yaml`:

- `kerberos_client_principal`: The Kerberos client principal to use when using Kerberos authentication within DSE. Example: cassandra@hostname.
- `kerberos_keytab_location`: The Kerberos keytab location when using Kerberos authentication within DSE. Example: /path/to/keytab.keytab.

Diagnostic tarball configurable timeout

The `diagnostic_tarball_download_timeout` configuration option has been added to allow configuring a timeout when generating a diagnostics tarball. Increasing the default value might be necessary on slower machines or for multi-instance clusters.

The `tarball_process_timeout` option has been removed. The option was actually an agent installation option that is no longer used due to improvements in the agent installation workflow.

Deprecated OpsCenter APIs

The following methods have been removed from Managing Cluster Configurations:

- `POST /{cluster_id}/nodeconf/{node_ip}/`
- `GET /{cluster_id}/dseconf/{node_ip}/nodetype`
- `POST /{cluster_id}/clusterconf/{dc}/`
- `POST /{cluster_id}/dseconf/{node_ip}/nodetype`

Warnings on deprecated Datastax Enterprise metrics

After upgrading a DataStax Enterprise cluster, OpsCenter detects and deletes any obsolete metrics in use within dashboard graph presets or alert rules. When first starting OpsCenter after an upgrade, warnings indicate which metrics are no longer supported and have been deleted.

Metrics inserted asynchronously

Metrics are now inserted asynchronously using native driver capabilities. The following configuration options are obsolete and have been removed from agent configuration:

- `async_queue_size`
- `async_pool_size`

OpsCenter 5.2 upgrade considerations

Review changes that affect upgrades to OpsCenter 5.2, as well as changes in precedence behavior amongst OpsCenter configuration files. A major change includes the connection protocol from thrift to native transport. OpsCenter 5.2 also now requires DataStax Enterprise 4.5 or later, and Apache Cassandra™ 2.0 or later.

Connection protocol changed from thrift to native transport

The central opscenterd process now connects to the nodes in your cluster using native transport instead of thrift. The `native_transport_server` must be enabled on your nodes, and port 9042 must be open on each node to the server running opscenterd. For more information, see ports.

Configuration option changes

OpsCenter 5.2 introduced a change to an option in the *cluster_name*.conf configuration file. The auto_node_discovery setting in the [cassandra] and [storage_cassandra] sections has been replaced with local_dc_pref. If you still have the auto_node_discovery option set, opscenterd fails to start. Update your cluster configurations with the local_dc_pref option.

Precedence for OpsCenter configuration files

In versions of Opscenter earlier than 5.2, the settings in the cluster_name.conf configuration file took precedence over settings in address.yaml. In OpsCenter version 5.2 and going forward, the reverse is true: addresss.yaml settings take precedence over cluster_name.conf. To summarize OpsCenter 5.2 configuration files precedence, settings in address.yaml override settings in cluster_name.conf, which in turn override default configuration settings.

Note: In most environments, stomp_interface is the only property that will need to be explicitly configured in address.yaml, and this might happen automatically. You can set most of these properties in the [agent_config] section of cluster_name.conf on the opscenterd machine and the properties propagate automatically to all agents. Some cases will require setting certain properties directly in address.yaml on applicable agents.

OpsCenter 5.1 upgrade considerations

OpsCenter 5.1 introduced several changes to the options in the address.yaml and *cluster_name*.conf configuration files.

Agent configuration changes

If the address.yaml configuration file used by the agents was manually modified, you must edit it based on the following changes in OpsCenter 5.1:

- The thrift_rpc_interface and storage_thrift_hosts options were replaced with hosts. The hosts option accepts an array of strings specifying the IP addresses of the Apache Cassandra™ or DataStax Enterprise node or nodes where OpsCenter data is stored.

  ```
  hosts: ["123.234.111.11", "10.1.1.1"]
  ```

  Note: If the rpc_address property in cassandra.yaml on these nodes is configured to anything other than 127.0.0.1 (localhost) or 0.0.0.0, you must configure the hosts property in address.yaml for the agent on each node.
- The storage_thrift_port option was removed.
- The thrift_port option was superseded with cassandra_port.
- The storage_thrift_port, autodiscovery_enabled, autodiscovery_interval, storage_dc, thrift_socket_timeout, thrift_conn_timeout, and thrift_max_conns options were removed.
- The thrift_user, storage_thrift_user, thrift_pass, and storage_thrift_pass options were replaced by cassandra_user and cassandra_pass.
- The thrift_ssl_truststore and thrift_ssl_truststore_password options were replaced by ssl_keystore and ssl_keystore_password. The ssl_keystore option is the path to the keystore, not the truststore. The thrift_ssl_truststore_type and thrift_max_frame_size options were removed.
- All the Kerberos options were replaced with a single kerberos_service options specifying the Kerberos service name. Setting the service name enables Kerberos authentication. Kerberos is configured in the kerberos.config file.

Cluster-specific configuration changes

The cluster_name.conf configuration file had the following changes in OpsCenter 5.1:

- In the `cassandra` section, the `send_thrift_rpc` option was renamed `thrift_rpc`.
- In the `agents` section, the `thrift_ssl_truststore` and `thrift_ssl_truststore_password` options were renamed `ssl_keystore` and `ssl_keystore_password`. The `ssl_keystore` option is the path to the keystore, not the truststore. The `thrift_ssl_truststore_type` option was removed.

Upgrading DataStax drivers

Be sure to check driver compatibility. Your driver may not be compatible with the upgrade version or require re-compiling.

Starting with DataStax Enterprise 5.0, DataStax drivers come in two types: DataStax drivers for Apache Cassandra™ and DataStax drivers for DataStax Enterprise 5.0 and later. The DataStax 5.0 drivers are built on top of open-source DataStax drivers for Apache Cassandra and enhanced to ease the development of applications powered by DataStax Enterprise. These drivers support the functionality introduced in DataStax Enterprise 5.0, including DSE Graph, unified authentication, and geospatial types. They are also full compatibility with Apache Cassandra 3.0.

Important: You may need to recompile your client application code, depending on the driver version. If the driver is Cassandra 3.0 compatible, you do not need to recompile. However, if you want to use the DataStax Enterprise 5.0 specific functionality, you must use the DataStax drivers for DataStax Enterprise 5.0 and recompile your application.

All DataStax drivers compatible with Cassandra 3.0 are backwards compatible with Cassandra 2.1. This means you can upgrade your application to use the new driver and then upgrade DataStax Enterprise to 5.0.

1. Use any dependency manager to pull the new dependency. These drivers are available through Maven, NuGet, npm, pip, and so on (Java Maven dependency example):

```
<dependency>
    <groupId>com.datastax.cassandra</groupId>
    <artifactId>cassandra-driver-core</artifactId>
    <version>3.0.0</version>
</dependency>
```

2. Update the corresponding imports and the initialization of cluster and session to use `DseCluster` and `DseSession` (Java code example):

```
// Java driver example which uses geospatial types

import com.datastax.driver.dse.DSECluster;
import com.datastax.driver.dse.DSESession;
import com.datastax.driver.dse.geometry.Point;

public class GeoSpatialExample {
    public static main(String[] args) {
        DseCluster dseCluster = DseCluster.builder()
                .addContactPoint("127.0.0.1")
                .build();
        DseSession dseSession = dseCluster.connect();
    }
}
```

3. Use the APIs specific to the DataStax Enterprise 5.0 (Java code example):

```
// Java driver example which uses geospatial types

import com.datastax.driver.dse.geometry.Point;

public class GeoSpatialExample {
    public static main(String[] args) {
        // Cluster and session initialization code elided
        Row row = dseSession.execute("SELECT coords FROM
 points_of_interest " +
                "WHERE name = 'Eiffel Tower'").one();
        Point coords = row.get("coords", Point.class);

        dseSession.execute("INSERT INTO points_of_interest (name, coords)
 " +
                "VALUES (?, ?)", "Washington Monument", new Point(38.8895,
 77.0352));
        // etc.
    }
}
```

For other driver APIs specific to the DataStax Enterprise 5.0, see the *API documentation* section in the following docs:

- C/C++
- C#
- Java
- Node.js
- PHP
- Python
- Ruby

Upgrading the DataStax AMI

Follow the upgrade instructions for Debian or Ubuntu:

- Installation instructions for upgrading DataStax Enterprise on Debian-based distributions on page 54
- Upgrading Cassandra on Debian or Ubuntu distributions

Note: The DataStax AMI can be used to install DataStax Enterprise 4.8 and earlier or Cassandra 2.1 and earlier on Amazon EC2. The DataStax AMI does not support later versions of Cassandra or DataStax Enterprise.

Installation instructions for upgrading DataStax Enterprise on RHEL-based distributions

This procedure shows how to install a new version of the DataStax Enterprise that replaces an existing installation on RHEL distributions using the Yum Package Manager.

1. Be sure to backup your cassandra.yaml and dse.yaml configuration files.

Yum might also back them up in place using a `.rpmsave` extension. For example, `cassandra.yaml.rpmsave`.

2. Open the Yum repository file for DataStax Enterprise in `/etc/yum.repos.d` for editing:

```
$ sudo vi /etc/yum.repos.d/datastax.repo
```

3. Replace the contents of the file with the following lines using your DataStax Academy credentials:

```
[datastax]
name = DataStax Repo for Apache Cassandra
baseurl = https://username:password@rpm.datastax.com/enterprise
enabled = 1
gpgcheck = 0
```

Attention: Depending on your environment, you might need to replace @ in your email address with %40 and escape any character in your password that is used in your operating system's command line. Examples: \! and \|.

4. If you have enabled signature verification (`gpgcheck=1`), import the DataStax Enterprise repository key:

```
$ rpm --import http://rpm.datastax.com/rpm/repo_key
```

5. Upgrade the node:

- Upgrade to the latest version:

```
$ sudo yum remove dse-full
$ sudo yum install dse-full
```

- Upgrade to a specific version:

```
$ sudo yum remove dse-full-version-1
$ sudo yum install dse-full-version-1
```

For example:

```
$ sudo yum remove dse-full-4.8.3-1
$ sudo yum install dse-full-4.8.4-1
```

Installation instructions for upgrading DataStax Enterprise on Debian-based distributions

This procedure shows how to install a new version of the DataStax Enterprise that replaces an existing installation on RHEL distributions using the Advanced Package Tool (apt-get).

1. Be sure to backup your cassandra.yaml and dse.yaml configuration files.

`apt-get` overwrites the modifications you have made to the configuration files.

2. If you were previously using a version of DataStax Community, add the DataStax repository to `/etc/apt/sources.list` using your DataStax Academy credentials:

```
$ deb https://username:password@debian.datastax.com/enterprise stable main
```

3. Upgrade the node:

- Upgrade to the latest version:

```
$ sudo apt-get remove dse "dse-.*" datastax-agent
$ sudo apt-get update
$ sudo apt-get install dse-full
```

- Upgrade to a specific version:

```
$ sudo apt-get remove dse "dse-.*" datastax-agent
$ sudo apt-get update
$ sudo apt-get install dse-full=version-1 dse=version-1 dse-hive=version-1
 dse-pig=version-1 dse-demos=version-1 dse-libsolr=version-1 dse-
libtomcat=version-1 dse-libsqoop=version-1 dse-liblog4j=version-1
 dse-libmahout=version-1 dse-libhadoop-native=version-1 dse-
libcassandra=version-1 dse-libhive=version-1 dse-libpig=version-1 dse-
libhadoop=version-1 dse-libspark=version
```

Example of `apt-get install`:

```
$ sudo apt-get install dse-full=4.8.4-1 dse=4.8.4-1 dse-hive=4.8.4-1 dse-
pig=4.8.4-1 dse-demos=4.8.4-1 dse-libsolr=4.8.4-1 dse-libtomcat=4.8.4-1
 dse-libsqoop=4.8.4-1 dse-liblog4j=4.8.4-1 dse-libmahout=4.8.4-1 dse-
libhadoop-native=4.8.4-1 dse-libcassandra=4.8.4-1 dse-libhive=4.8.4-1 dse-
libpig=4.8.4-1 dse-libhadoop=4.8.4-1 dse-libspark=4.8.4-1
```

Installation instructions for upgrading DataStax Enterprise using the DataStax Enterprise tarball

This procedure shows how to install a new version of the DataStax Enterprise binary tarball that replaces an existing installation on any Linux distribution.

Upgrading a node and migrating the data

1. Be sure to backup your cassandra.yaml and dse.yaml configuration files.
2. Get the binary tarball:
   - Download and extract the latest version using your DataStax Academy credentials:

     ```
     $ curl -username:password -L http://downloads.datastax.com/enterprise/
     dse.tar.gz | tar xz
     ```

     Note: Because passwords are retained in shell history, DataStax recommends using the --netrc or --netrc-file option.

     The files are downloaded and extracted into the dse-*version* directory.
   - Get a specific version:
     1. Download the tarball from DataStax downloads using your DataStax Academy credentials.
     2. Unpack the DataStax Enterprise tarball:

        ```
        $ tar -xzvf dse-version.tar.gz
        ```

        For example:

        ```
        $ tar -xzvf dse-4.8.6.tar.gz
        ```

        The files are extracted into the dse-*version* directory.
     3. Move the installation directory as required by your environment.
3. **On RHEL 5 or CentOS 5 platforms only**, run this script to replace all instances of `snappy-java-1.0.5.jar` with the 1.0.4.1 JAR:

   ```
   $ ./bin/switch_snappy 1.0.4
   ```

Note: DataStax Enterprise 5.0 does not support the RHEL 5 and CentOS 5 platforms.

4. If you customized the location of the data in the old installation, create a symbolic link to the old data directory:

```
$ cd new_install_location
$ ln -s old_data_directory new_install_location/new_data_directory
```