DataStax Upgrade Guide

Documentation

${ds.localized.time}

Contents

About the upgrade guide

Upgrade instructions for Cassandra, DataStax Enterprise, and OpsCenter.

This guide includes detailed instructions on upgrading DataStax Enterprise, Cassandra, OpsCenter, and DataStax Agents. Be sure to follow the recommended order for upgrading nodes, best practices, considerations, and impacts of changes. This guide also provides information on upgrading the DataStax AMI and reverting to earlier versions.

Upgrade instructions

- Upgrading DataStax Enterprise
- Upgrading Cassandra
- Upgrading OpsCenter

Upgrading from DataStax Community to DataStax Enterprise

Upgrade instructions from DataStax Community to DataStax Enterprise.

dse.yaml

The location of the `dse.yaml` file depends on the type of installation:

Installer-Services	`/etc/dse/dse.yaml`
Package installations	`/etc/dse/dse.yaml`
Installer-No Services	*`install_location`*`/resources/dse/conf/dse.yaml`
Tarball installations	*`install_location`*`/resources/dse/conf/dse.yaml`

cassandra.yaml

The location of the `cassandra.yaml` file depends on the type of installation:

Installer-Services	`/etc/cassandra/cassandra.yaml`
Package installations	`/etc/cassandra/cassandra.yaml`
Installer-No Services	*`install_location`*`/conf/cassandra.yaml`
Tarball installations	*`install_location`*`/conf/cassandra.yaml`

About this task

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

Procedure

1. To ensure that your version of DataStax Community can be upgraded directly to the version of Cassandra that is used by DataStax Enterprise, review the CHANGES.txt details on upgrading Cassandra.
2. 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.
3. Stop the node.
4. Back up your configuration files.

   Depending on how you install the product, the configuration files might be overwritten with default values during the installation.
5. 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
  ```

6. Install the new product using one of the following:

   - Binary tarball installs
   - Debian-based installs
   - RHEL-based installs
   - DataStax GUI/Text installer
   - Unattended installer

7. To configure the product, use your backup configuration files to merge any necessary modifications into the configuration files for the new version.
8. 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 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.
9. Start the node.

   - Installer-Services and Package installations: start DataStax Enterprise as a service.
   - Installer-No Services and Tarball installations: start DataStax Enterprise as a stand-alone process.

10. If your DataStax Enterprise upgrade includes a major upgrade of Cassandra, upgrade the SSTables on each node.

    ```
    $ nodetool upgradesstables
    ```

    Cassandra requires upgrading SSTables for major releases.

    - DataStax Enterprise 4.7 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

11. Verify that the upgraded data center names still match the data center names that are used in the keyspace schema definition:

    ```
    $ nodetool status
    ```

12. 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.

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

Upgrading DataStax Enterprise

Upgrade to the most current version of DataStax Enterprise.

This section describes how to upgrade to the latest version of DataStax Enterprise.

Guidelines and general upgrade steps

Guidelines, best practices, and general upgrade steps.

The upgrade process for DataStax Enterprise provides minimal downtime (ideally zero). 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.

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.

Upgrade paths

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

Current version	Required interim version	Target version
• DataStax Enterprise 3.2.10 and earlier • DataStax Community or Open Source Cassandra 1.2 and earlier	• DataStax Enterprise 4.0 and later • DataStax Community or Open Source Cassandra 2.0.x	• DataStax Enterprise 4.7
• DataStax Enterprise 3.2.4 and earlier • DataStax Community or Open Source Cassandra 1.2.15 and earlier	• DataStax Enterprise 3.2.5 and later • DataStax Community or Open Source Cassandra 1.2.16	• DataStax Enterprise 4.6 • DataStax Enterprise 4.5 • DataStax Enterprise 4.0
• DataStax Enterprise 2.2.1 and earlier • DataStax Community or Open Source Cassandra 1.1.8 and earlier • DataStax Community or Open Source Cassandra 1.2.8 and earlier	• DataStax Enterprise 2.2.2 and later • DataStax Community or Open Source Cassandra 1.1.9 • DataStax Community or Open Source Cassandra 1.2.9 to 1.2.15	• DataStax Enterprise 3.2

Upgrade order

Upgrade order matters. Using the following guidelines, upgrade nodes in the recommended order:

- In multiple data center clusters, upgrade all the nodes within one data center before moving on to another data center.
- Upgrade DSE Analytics nodes or data centers first, then Cassandra nodes or data centers, and finally DSE Search nodes or data centers.
- For DSE Analytics nodes, upgrade the Job Tracker node first. Then upgrade Hadoop nodes, followed by Spark nodes.
- Upgrade the seed nodes within a data center first.

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

General upgrade steps

To perform an upgrade with zero downtime, DataStax recommends following these steps to perform the upgrade as a rolling restart:

1. Make a backup of the data by taking a snapshot of the node to be upgraded.
2. Stop the node.
3. Install the new product.
4. Configure the new product.
5. Start the node.
6. Check the logs for warnings, errors, and exceptions.
7. Repeat these steps for each node in the cluster.

Upgrade limitations

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

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

General upgrade limitations

- Do not run `nodetool repair`.
- Do not enable new features.
- Do not issue these types of queries during a rolling restart: `DDL`, `TRUNCATE`
- During upgrades, the nodes on different versions show a schema disagreement.

DSE Analytics (Hadoop and Spark) specific upgrade limitations

- Do not run analytics jobs until all nodes are upgraded.
- You must kill all Spark worker processes before you start the upgrade.

DSE Search and Solr-specific upgrade limitations

- Do not update schemas.
- Do not re-index DSE Search or Solr nodes unless you are following instructions in the upgrade guide procedures to re-index.
- Do not issue these types of queries during a rolling restart: `DDL`, `TRUNCATE`, and Solr queries.
- Disable driver pagination while the cluster is in a partially upgraded state to disable Solr pagination for queries on the 4.7 nodes. During the upgrade process on a cluster with mixed versions where DataStax Enterprise 4.7 supports pagination and earlier versions do not, issuing queries from the upgraded nodes will return only FetchSize results. FetchSize controls how many resulting rows are retrieved simultaneously.

Security upgrade limitations

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

- Do not attempt to set up Kerberos authentication before the upgrade. First upgrade the cluster, and then set up Kerberos.

Upgrade instructions for installing the DataStax Enterprise tarball on any Linux distribution

Install tarball instructions for DataStax Enterprise.

About this task

This procedure shows how to install a new version of the DataStax Enterprise binary tarball to replace an existing installation.

Before performing the installation, be sure to back up your configuration files for future reference.

Upgrading a node and migrating the data

Procedure

1. Download the DataStax Enterprise tarball using your username and password:

   ```
   $ 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. Alternatively, download the tarball from DataStax downloads.

   Get the *username* and *password* from your DataStax registration confirmation email. If you don’t have the email, register on the DataStax web site.
2. Create a directory for the new installation and move it to that directory.
3. Unpack the DataStax Enterprise tarball:

   ```
   $ tar xzvf dse-4.7 tarball name
   ```
4. **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 available here:

   ```
   $ ./bin/switch_snappy 1.0.4
   ```
5. 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
   ```

Upgrade installation instructions for Debian-based distributions

Install instructions for DataStax Enterprise on Debian distributions.

About this task

This procedure shows how to install a new version of DataStax Enterprise using the Advanced Package Tool, `apt-get`.

Ensure that you back up your configuration files before starting this upgrade process because apt-get overwrites the modifications you have made to the configuration files.

Procedure

1. If you were previously using a version of DataStax Community, add the DataStax repository to `/etc/apt/sources.list` using your username and password.

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

   Get the *username* and *password* from your DataStax registration confirmation email. If you don't have the email, register on the DataStax web site.
2. Upgrade the node.

   ```
   $ sudo apt-get update
   $ sudo apt-get install dse-full
   ```
3. If the prompt appears informing you of the disk space to be used, type Y to continue.

Upgrade installation instructions for RHEL-based distributions

Install instructions for DataStax Enterprise on RHEL distributions.

About this task

This procedure shows how to install a new version of DataStax Enterprise using the Yum Package Manager.

Ensure that you back up your configuration files before starting this upgrade process. However, Yum might also back them up in place using a `.rpmsave` extension. For example, `cassandra.yaml.rpmsave`.

Procedure

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

   ```
   $ sudo vi /etc/yum.repos.d/datastax.repo
   ```
2. Replace the contents of the file with the following lines using your username and password.

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

   Get the *username* and *password* from your DataStax registration confirmation email. If you do not have the email, register on the DataStax web site.
3. Upgrade the node.

   ```
   $ sudo yum remove dse-full
   $ sudo yum install dse-full
   ```
4. If a prompt informs you of the download size and asks for confirmation to continue, type Y to continue.

Upgrading Linux installations using the DataStax installer

DataStax Enterprise upgrade instructions for the DataStax GUI/Text Installer.

cassandra.yaml

The location of the `cassandra.yaml` file depends on the type of installation:

Installer-Services	`/etc/cassandra/cassandra.yaml`

Package installations	`/etc/cassandra/cassandra.yaml`
Installer-No Services	*install_location*`/conf/cassandra.yaml`
Tarball installations	*install_location*`/conf/cassandra.yaml`

dse.yaml

The location of the `dse.yaml` file depends on the type of installation:

Installer-Services	`/etc/dse/dse.yaml`
Package installations	`/etc/dse/dse.yaml`
Installer-No Services	*install_location*`/resources/dse/conf/dse.yaml`
Tarball installations	*install_location*`/resources/dse/conf/dse.yaml`

About this task

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 cassandra.yaml and dse.yaml with your new entries.

Before you begin

You must meet the following requirements to upgrade to the latest version of DataStax Enterprise:

- You have an installation of DataStax Enterprise 4.0.x or later.
- Your current installation was installed using either:
  - The `yum` (RHEL or CentOS) or `apt` (Debian or Ubuntu) package managers.
  - The GUI installer as a services install.

Procedure

1. Download the installer for your computer from the DataStax download page.
2. From the directory where you downloaded the install file, make it executable and run it using the `sudo` command.

   ```
   $ chmod +x DataStaxEnterprise-4.7.x-linux-x64-installer.run ## Changes
    permission to executable
   $ sudo ./DataStaxEnterprise-4.7.x-linux-x64-installer.run
   ```
3. Follow the instructions in the setup wizard. See Installing DataStax Enterprise using GUI or Text mode for a detailed description of the settings in the wizard.
4. Start the services.

   ```
   $ sudo service dse start ## Starts the DataStax Enterprise server
   $ sudo service datastax-agent start ## Starts the DataStax Agent
   $ sudo service opscenterd start ## Starts the OpsCenter
   ```
5. Verify that DataStax Enterprise is running using the `nodetool` command.

   ```
   $ nodetool status
   ```

Version-specific upgrade instructions

Instructions for upgrading to specific versions of DataStax Enterprise.

This section contains instructions for upgrading to specific versions of DataStax Enterprise.

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

To upgrade from a Brisk release, contact DataStax Support.

Upgrading to DataStax Enterprise 4.7

Follow these instructions before upgrading to DataStax Enterprise 4.7.

Considerations and prerequisites

- If upgrading from product versions earlier than these interim versions, you must perform an interim upgrade to one of these required versions before you can upgrade to DataStax Enterprise 4.7:
  - DataStax Enterprise 4.0 and later
  - DataStax Community or Open Source Cassandra 2.0.x
- Latest version of Oracle Java SE Runtime Environment 7 or 8 or OpenJDK 7 is recommended.

  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.
- Log exceptions occur during upgrade of analytics nodes.

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

  Disable driver pagination while the cluster is in a partially upgraded state to disable Solr pagination for queries on the 4.7 nodes. During the upgrade process on a cluster with mixed versions where DataStax Enterprise 4.7 supports pagination and earlier versions do not, issuing queries from the upgraded nodes will return only FetchSize results. FetchSize controls how many resulting rows are retrieved simultaneously.

  To retain the same behavior for your queries in 4.7 as in 4.6, disable protocol-level paging for CQL-based Solr queries. DataStax Enterprise 4.7 integrates native driver paging with Solr cursor-based paging.
- If you are upgrading from Cassandra 1.x, see Cassandra 2.1.x restrictions.

Note: If you are upgrading from DataStax Enterprise 4.0.0 and have DSE Search nodes, see Upgrading from DataStax Enterprise 4.0.0 for special instructions.

Procedure

1. Familiarize yourself with the changes and features in this release:
   - DataStax Enterprise release notes for 4.5, 4.6, and 4.7.
   - General upgrade advice and Cassandra features in NEWS.txt.
   - Cassandra changes in CHANGES.txt.
2. Review the limitations that apply while the cluster is in a partially upgraded state.
3. If you are upgrading directly to DataStax Enterprise 4.7 from versions 4.0.x, or 4.5 and skipping an upgrade to 4.6, you **must** update the endpoint snitch in the Upgrading to DataStax Enterprise 4.6 procedure.
4. Review the limitations that apply while the cluster is in a partially upgraded state.

5. 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.
6. Stop the node.
7. Back up your configuration files.

   Depending on how you install the product, the configuration files might be overwritten with default values during the installation.
8. Install the new product using one of the following:

   - Binary tarball installs
   - Debian-based installs
   - RHEL-based installs
   - DataStax GUI/Text installer
   - Unattended installer
9. To configure the product, use your backup configuration files to merge any necessary modifications into the configuration files for the new version.
10. Start the node.

    - Installer-Services and Package installations: start DataStax Enterprise as a service.
    - Installer-No Services and Tarball installations: start DataStax Enterprise as a stand-alone process.
11. If your DataStax Enterprise upgrade includes a major upgrade of Cassandra, upgrade the SSTables on each node.

    ```
    $ nodetool upgradesstables
    ```

    Cassandra requires upgrading SSTables for major releases.

    - DataStax Enterprise 4.7 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. If you have existing tables that use the DSE In-Memory option under DataStax Enterprise 4.6 or earlier:

    a) Turn off SSTable compression.

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

    b) Rewrite existing SSTables without compression:

    ```
    $ nodetool upgradesstables -a <keyspacename> <tablename>
    ```
13. Verify that the upgraded data center names still match the data center names that are used in the keyspace schema definition:

    ```
    $ nodetool status
    ```
14. 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.

    Because DataStax Enterprise 4.7 uses 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"
    ```

  If you find unexpected warnings, errors, or exceptions, contact DataStax Support.

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

Upgrading to DataStax Enterprise 4.6

Follow these instructions to upgrade to DataStax Enterprise 4.6 and set the endpoint snitch.

dse.yaml

The location of the `dse.yaml` file depends on the type of installation:

Installer-Services	`/etc/dse/dse.yaml`
Package installations	`/etc/dse/dse.yaml`
Installer-No Services	*`install_location`*`/resources/dse/conf/dse.yaml`
Tarball installations	*`install_location`*`/resources/dse/conf/dse.yaml`

cassandra.yaml

The location of the `cassandra.yaml` file depends on the type of installation:

Installer-Services	`/etc/cassandra/cassandra.yaml`
Package installations	`/etc/cassandra/cassandra.yaml`
Installer-No Services	*`install_location`*`/conf/cassandra.yaml`
Tarball installations	*`install_location`*`/conf/cassandra.yaml`

Considerations and prerequisites

- If upgrading from product versions earlier than these interim versions, you must perform an interim upgrade to one of these required versions before you can upgrade to DataStax Enterprise 4.6:
  - DataStax Enterprise 3.2.5 and later
  - DataStax Community or Open Source Cassandra 1.2.16

  After you upgrade to a required interim version, upgrade your sstables: `nodetool upgradesstables`.

  Interim version upgrades might also require special steps. Follow the steps for the version that you are upgrading from.

  - Upgrading from DataStax Enterprise 3.2.x
  - Upgrading from DataStax Enterprise 4.0.0
- Latest version of Oracle Java SE Runtime Environment 7 or 8 or OpenJDK 7 is recommended..

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.

- 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 Standalone Installer automatically sets the default `endpoint_snitch` to `DseSimpleSnitch` and removes the option from the dse.yaml.

About this task

Procedure

1. Familiarize yourself with the changes and features in this release:
   - DataStax Enterprise release notes for 4.5, 4.6, and 4.7.
   - General upgrade advice and Cassandra features in NEWS.txt.
   - Cassandra changes in CHANGES.txt.
2. Review the limitations that apply while the cluster is in a partially upgraded state.
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.
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. Install the new product using one of the following:
   - Binary tarball installs
   - Debian-based installs
   - RHEL-based installs
   - DataStax GUI/Text installer
   - Unattended installer
7. 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
   ```
8. Remove the `delegated_snitch` option from the old `dse.yaml`.
9. Make other required configurations for DataStax Enterprise 4.6.

   Using the backups that you made of your configuration files, merge any modifications that you have previously made into the configuration files for the new version.
10. Start the node.
    - Installer-Services and Package installations: start DataStax Enterprise as a service.
    - Installer-No Services and Tarball installations: start DataStax Enterprise as a stand-alone process.
11. If your DataStax Enterprise upgrade includes a major upgrade of Cassandra, upgrade the SSTables on each node.

```
$ nodetool upgradesstables
```

Cassandra requires upgrading SSTables for major releases.

- DataStax Enterprise 4.7 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 data center names still match the data center 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 to DataStax Enterprise 4.0 or 4.5

Follow these instructions to upgrade to DataStax Enterprise 4.0 or 4.5.

dse.yaml

The location of the `dse.yaml` file depends on the type of installation:

Installer-Services	`/etc/dse/dse.yaml`
Package installations	`/etc/dse/dse.yaml`
Installer-No Services	*install_location*/resources/dse/conf/dse.yaml
Tarball installations	*install_location*/resources/dse/conf/dse.yaml

cassandra.yaml

The location of the `cassandra.yaml` file depends on the type of installation:

Installer-Services	`/etc/cassandra/cassandra.yaml`
Package installations	`/etc/cassandra/cassandra.yaml`
Installer-No Services	*install_location*/conf/cassandra.yaml
Tarball installations	*install_location*/conf/cassandra.yaml

Considerations and prerequisites

- If upgrading from product versions earlier than these interim versions, you must perform an interim upgrade to one of these required versions before you can upgrade to DataStax Enterprise 4.0 or 4.5:

    - DataStax Enterprise 3.2.5 and later
    - DataStax Community or Open Source Cassandra 1.2.16

    After you upgrade to a required interim version, upgrade your sstables: `nodetool upgradesstables`.

    Interim version upgrades might also require special steps. Follow the steps for the version that you are upgrading from.

  - Upgrading from DataStax Enterprise 3.2.x
  - Upgrading from DataStax Enterprise 4.0.0
- If you are upgrading from DataStax Enterprise 4.0.0 and have DSE Search nodes, see Upgrading from DataStax Enterprise 4.0.0 for special instructions.
- Oracle Java 7.

Procedure

1. Familiarize yourself with the changes and features in this release:

   - DataStax Enterprise release notes for 4.5, 4.6, and 4.7.
   - General upgrade advice and Cassandra features in NEWS.txt.
   - Cassandra changes in CHANGES.txt.
2. Review the limitations that apply while the cluster is in a partially upgraded state.
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.
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. Install the new product using one of the following:

   - Binary tarball installs
   - Debian-based installs
   - RHEL-based installs
   - DataStax GUI/Text installer
   - Unattended installer
7. To configure the product, use your backup configuration files to merge any necessary modifications into the configuration files for the new version.
8. Start the node.

   - Installer-Services and Package installations: start DataStax Enterprise as a service.
   - Installer-No Services and Tarball installations: start DataStax Enterprise as a stand-alone process.
9. If your DataStax Enterprise upgrade includes a major upgrade of Cassandra, upgrade the SSTables on each node.

   ```
   $ nodetool upgradesstables
   ```

   Cassandra requires upgrading SSTables for major releases.

   - DataStax Enterprise 4.7 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 data center names still match the data center names that are used in the keyspace schema definition:

    ```
    $ nodetool status
    ```
11. 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.

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

Upgrading from DataStax Enterprise 4.0.0

Instructions on upgrading from DataStax Enterprise 4.0.0.

About this task

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 does not impact upgrades from versions earlier than 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 from version 3.2.x

Instructions on upgrading from DataStax Enterprise 3.2.x

cassandra.yaml

The location of the `cassandra.yaml` file depends on the type of installation:

Installer-Services	`/etc/cassandra/cassandra.yaml`
Package installations	`/etc/cassandra/cassandra.yaml`
Installer-No Services	*`install_location`*`/conf/cassandra.yaml`
Tarball installations	*`install_location`*`/conf/cassandra.yaml`

Remove unsupported options in `cassandra.yaml`

Remove or comment out the following options in cassandra.yaml on each node:

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

These options are no longer supported by DataStax Enterprise.

Cassandra 2.0 upgrade changes

DataStax Enterprise 4.0.x to 4.6.x includes Cassandra 2.0.x, see Cassandra 2.0.x restrictions.

Procedure

1. Familiarize yourself with the changes and features in this release:
   - DataStax Enterprise release notes for 4.5, 4.6, and 4.7.
   - General upgrade advice and Cassandra features in NEWS.txt.
   - Cassandra changes in CHANGES.txt.
2. Review the limitations that apply while the cluster is in a partially upgraded state.
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.
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. Install the new product using one of the following:
   - Binary tarball installs
   - Debian-based installs
   - RHEL-based installs
   - DataStax GUI/Text installer
   - Unattended installer
7. To configure the product, use your backup configuration files to merge any necessary modifications into the configuration files for the new version.
8. Start the node.
   - Installer-Services and Package installations: start DataStax Enterprise as a service.
   - Installer-No Services and Tarball installations: start DataStax Enterprise as a stand-alone process.
9. If you are upgrading over a major release of Cassandra (for example, from DataStax Enterprise 3.2 to 4.0, or Cassandra 1.1 to 1.2, upgrade the SSTables on each node.

   ```
   $ nodetool upgradesstables
   ```
10. 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.

Upgrading to DataStax Enterprise 3.2

Follow these instructions to upgrade to DataStax Enterprise 3.2.x.

Considerations and prerequisites

- If upgrading from product versions earlier than these interim versions, you must perform an interim upgrade to one of these required versions before you can upgrade to DataStax Enterprise 3.2:
  - DataStax Enterprise 2.2.2 and later
  - DataStax Community or Open Source Cassandra 1.1.9
  - DataStax Community or Open Source Cassandra 1.2.9 to 1.2.15

  After you upgrade to a required interim version, upgrade your sstables: `nodetool upgradesstables`.

- Java 7 or later

Interim version upgrades might also have require special steps. Follow the steps for the version that you are upgrading from.

- Upgrading from 3.1.x
- Upgrading from version 3.0.x
- Upgrading from version 2.2.x

Procedure

1. Familiarize yourself with the changes and features in this release:
   - DataStax Enterprise release notes for 4.5, 4.6, and 4.7.
   - General upgrade advice and Cassandra features in NEWS.txt.
   - Cassandra changes in CHANGES.txt.
2. Review the limitations that apply while the cluster is in a partially upgraded state.
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.
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. Install the new product using one of the following:
   - Binary tarball installs
   - Debian-based installs
   - RHEL-based installs
   - DataStax GUI/Text installer
   - Unattended installer
7. To configure the product, use your backup configuration files to merge any necessary modifications into the configuration files for the new version.
8. Start the node.
   - Installer-Services and Package installations: start DataStax Enterprise as a service.
   - Installer-No Services and Tarball installations: start DataStax Enterprise as a stand-alone process.
9. If you are upgrading over a major release of Cassandra (for example, from DataStax Enterprise 3.2 to 4.0, or Cassandra 1.1 to 1.2, upgrade the SSTables on each node.

   ```
   $ nodetool upgradesstables
   ```
10. Verify that the upgraded data center names still match the data center names that are used in the keyspace schema definition:

    ```
    $ nodetool status
    ```
11. 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.
12. Repeat the upgrade on each node in the cluster following the recommended upgrade order.

Upgrading from version 3.1.x

Upgrade from DataStax Enterprise 3.1.x to the latest version.

About this task

Follow these instructions to upgrade to the latest version from DataStax Enterprise 3.1.x.

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
```

Note: This is unnecessary when upgrading to DataStax Enterprise 3.2.1 or later.

After upgrading the entire cluster, remove this line from `cassandra-env.sh` on each node so it uses the new protocol, and perform a second rolling restart.

Manually updating the dse_system keyspace to use the EverywhereStrategy

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 the `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 version 3.0.x

Upgrade from DataStax Enterprise 3.0.x to the latest version of DataStax Enterprise.

cassandra.yaml

The location of the `cassandra.yaml` file depends on the type of installation:

Installer-Services	`/etc/cassandra/cassandra.yaml`
Package installations	`/etc/cassandra/cassandra.yaml`
Installer-No Services	`*install_location*/conf/cassandra.yaml`
Tarball installations	`*install_location*/conf/cassandra.yaml`

dse.yaml

The location of the `dse.yaml` file depends on the type of installation:

Installer-Services	`/etc/dse/dse.yaml`
Package installations	`/etc/dse/dse.yaml`
Installer-No Services	`*install_location*/resources/dse/conf/dse.yaml`
Tarball installations	`*install_location*/resources/dse/conf/dse.yaml`

About this task

Follow these instructions to upgrade to the latest version from DataStax Enterprise 3.0.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

Merge your partitioner setting from the old to the new file. Do not attempt to use the Cassandra 1.2 default partitioner option, Murmur3Partitioner, in the new file unless you were already using it.

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, 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 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.

SSTable upgrades

After restarting each node, consider upgrading SSTables. Upgrading SSTables is highly recommended under these conditions:

- If you use counter columns
- If you are upgrading from Cassandra 1.0.x or earlier
- If you are upgrading from a DataStax Enterprise version having Cassandra 1.0.x or earlier

Upgrade SSTables before doing these operations:

- move
- repair
- bootstrap

Because these operations copy SSTables within the cluster and the on-disk format sometimes changes between major versions, DataStax recommends upgrading SSTables now to prevent possible future SSTable incompatibilities:

- Tarball: `*install_location*/bin/nodetool -h upgradesstables`
- Package or AMI: `nodetool -h upgradesstables`

Virtual nodes

DataStax recommends using virtual nodes only on data centers running purely Cassandra workloads. You should disable virtual nodes on data centers running either Hadoop or Solr workloads by setting num_tokens to 1 in the `cassandra.yaml`.

Solr

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

Expected error messages

If you are upgrading from DataStax Enterprise 3.0.x, an exception that looks something like this might appear in logs during a rolling upgrade. Ignore these error messages:

```
ERROR 15:36:54,908 Exception in thread Thread[GossipStage:1,5,main ]
 java.lang.NumberFormatException: For input string:
 "127605887595351923798765477786913079296"
. . .
```

When upgrading Cassandra 1.2 nodes, you can ignore the following error messages related to when a node 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(ColumnFamilySerializer
```

When upgrading a Solr node, you can ignore the following error:

```
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" )))
```

Recommissioning a node

If you decommissioned a node in the last 72 hours:

- Do not recommission the node until 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 for detailed information on how to kill the node.

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
```

Note: This is unnecessary when upgrading to DataStax Enterprise 3.2.1 or later.

After upgrading the entire cluster, remove this line from `cassandra-env.sh` on each node so it uses the new protocol, and perform a second rolling restart.

Manually updating the dse_system keyspace to use the EverywhereStrategy

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 the `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 DataStax Enterprise 2.2.x

Follow these instructions to upgrade from DataStax Enterprise 2.2.x.

About this task

Follow these instructions to upgrade from DataStax Enterprise 2.2.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

Merge your partitioner setting from the old to the new file. Do not attempt to use the Cassandra 1.2 default partitioner option, Murmur3Partitioner, in the new file unless you were already using it.

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, 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 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.

SSTable upgrades

After restarting each node, consider upgrading SSTables. Upgrading SSTables is highly recommended under these conditions:

- If you use counter columns
- If you are upgrading from Cassandra 1.0.x or earlier
- If you are upgrading from a DataStax Enterprise version having Cassandra 1.0.x or earlier

Upgrade SSTables before doing these operations:

- move
- repair
- bootstrap

Because these operations copy SSTables within the cluster and the on-disk format sometimes changes between major versions, DataStax recommends upgrading SSTables now to prevent possible future SSTable incompatibilities:

- Tarball: `install_location/bin/nodetool -h upgradesstables`
- Package or AMI: `nodetool -h upgradesstables`

Virtual nodes

DataStax recommends using virtual nodes only on data centers running purely Cassandra workloads. You should disable virtual nodes on data centers running either Hadoop or Solr workloads by setting num_tokens to 1 in the `cassandra.yaml`.

Solr

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

Expected error messages

If you are upgrading from DataStax Enterprise 3.0.x, an exception that looks something like this might appear in logs during a rolling upgrade. Ignore these error messages:

```
ERROR 15:36:54,908 Exception in thread Thread[GossipStage:1,5,main ]
 java.lang.NumberFormatException: For input string:
 "127605887595351923798765477786913079296"
. . .
```

When upgrading Cassandra 1.2 nodes, you can ignore the following error messages related to when a node 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(ColumnFamilySerializer
```

When upgrading a Solr node, you can ignore the following error:

```
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" )))
```

Recommissioning a node

If you decommissioned a node in the last 72 hours:

- Do not recommission the node until 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 for detailed information on how to kill the node.

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
```

Note: This is unnecessary when upgrading to DataStax Enterprise 3.2.1 or later.

After upgrading the entire cluster, remove this line from `cassandra-env.sh` on each node so it uses the new protocol, and perform a second rolling restart.

Manually updating the dse_system keyspace to use the EverywhereStrategy

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 the `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 the DataStax AMI

Instructions for upgrading the DataStax Amazon Machine Image (AMI)

About this task

Follow the upgrade instructions for Debian or Ubuntu:

- Upgrading DataStax Enterprise
- Upgrading Cassandra

Rolling back an upgrade

Revert to an earlier DataStax Enterprise version

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

Revert to a previous version from a package install

How to revert your DataStax Enterprise installation to a previous version from a package install.

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.

   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

How to revert to a previous version of DataStax Enterprise from a tarball installation.

cassandra.yaml

The location of the `cassandra.yaml` file depends on the type of installation:

Installer-Services	`/etc/cassandra/cassandra.yaml`
Package installations	`/etc/cassandra/cassandra.yaml`
Installer-No Services	*install_location*`/conf/cassandra.yaml`
Tarball installations	*install_location*`/conf/cassandra.yaml`

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.

   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 Cassandra

This section describes how to upgrade to the latest version of Cassandra.

Cassandra versions requiring staggered upgrades

Upgrading from some versions of Cassandra require intermediate upgrades.

Cassandra 2.0.x restrictions

After downloading DataStax Community, upgrade to Cassandra directly from Cassandra 1.2.9 or later. Cassandra 2.0 is not network- or SSTable-compatible with versions older than 1.2.9. If your version of Cassandra is earlier than 1.2.9 and you want to perform a rolling restart, first upgrade the entire cluster to 1.2.9, and then to Cassandra 2.0.

Cassandra 2.1.x restrictions

Upgrade to Cassandra 2.1 from Cassandra 2.0.7 or later.

Cassandra 2.1 is not compatible with Cassandra 1.x SSTables. First upgrade the nodes to Cassandra 2.0.7 or later, start the cluster, upgrade the SSTables, stop the cluster, and then upgrade to Cassandra 2.1.

Best practices for upgrading Cassandra

General Cassandra upgrade procedure including best practices.

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. Check the Upgrading section of NEWS.txt for information about upgrading.
4. Familiarize yourself with changes and fixes in this release.

   A complete list is available in CHANGES.txt, and general upgrade advice for different features is available in NEWS.txt.
5. Run nodetool drain before shutting down the existing Cassandra service. This prevents overcounts of counter data, and also speeds up restart post-upgrade.
6. Follow the instructions in the Upgrading procedures for Cassandra.
7. Monitor the log files for any issues.
8. After upgrading and restarting all Cassandra processes, restart client applications.
9. 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.

Upgrading procedures for Cassandra

Detailed upgrade instructions for Cassandra.

cassandra.yaml

The location of the `cassandra.yaml` file depends on the type of installation:

Installer-Services	`/etc/cassandra/cassandra.yaml`
Package installations	`/etc/cassandra/cassandra.yaml`
Installer-No Services	*`install_location`*`/conf/cassandra.yaml`
Tarball installations	*`install_location`*`/conf/cassandra.yaml`

About this task

This section describes the procedure for upgrading nodes in a cluster.

Before you begin

- The latest version of the Java SE Runtime Environment (JRE) 7 is required for Cassandra 2.x.
- All dead nodes must be removed.

  Do not upgrade if nodes in the cluster are down. Use the nodetool removenode command, which was called `nodetool removetoken` in earlier releases, to remove dead nodes.
- 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 alter this property after the upgrade using CQL.
- If your cluster does not use vnodes, in each new `cassandra.yaml`, disable vnodes before doing the rolling restart.

  In Cassandra 2.0.x, virtual nodes (vnodes) are enabled by default. Disable vnodes in the 2.0.x version before upgrading.

  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.

Procedure

1. Stop the node.
2. 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.
3. Install the new version of Cassandra:

   - Upgrading Debian-based installations
   - Upgrading RHEL-based installations
   - Upgrading Tarball installations
4. 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.
5. Start the node

6. If you are upgrading from a major version (for example, from 1.2 to 2.0) or a major point release (for example, from Cassandra 2.0 to 2.1), upgrade the SSTables on each node.

   ```
   $ nodetool upgradesstables
   ```
7. Check the logs for warnings, errors, and exceptions.
8. Repeat on each node in the cluster.

   General limitations while cluster is in a partially upgraded state:

   - Do not run `nodetool repair`.
   - Do not use new features.
   - Do not issue these types of queries during a rolling restart: `DDL`, `TRUNCATE`.

Upgrading Debian-based installations

Upgrade Cassandra on Debian or Ubuntu distributions and preserve cassandra.yaml settings.

About this task

Follow these steps to get the new version, merge your customizations of the old `cassandra.yaml` file to the new one, and then complete the upgrade.

cassandra.yaml

The location of the `cassandra.yaml` file depends on the type of installation:

Installer-Services	`/etc/cassandra/cassandra.yaml`
Package installations	`/etc/cassandra/cassandra.yaml`
Installer-No Services	*install_location*`/conf/cassandra.yaml`
Tarball installations	*install_location*`/conf/cassandra.yaml`

Procedure

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

   ```
   $ sudo apt-get install dsc21
   ```
3. Open the old and new `cassandra.yaml` 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 file as `cassandra.yaml`.

Upgrading RHEL-based installations

Upgrade Cassandra on RHEL or CentOS distributions and preserve cassandra.yaml settings

About this task

Follow these steps to remove the old installation, merge your customizations of the old `cassandra.yaml` file to the new one, and then complete the upgrade.

cassandra.yaml

The location of the `cassandra.yaml` file depends on the type of installation:

Installer-Services	`/etc/cassandra/cassandra.yaml`
Package installations	`/etc/cassandra/cassandra.yaml`
Installer-No Services	*install_location*/conf/cassandra.yaml
Tarball installations	*install_location*/conf/cassandra.yaml

Procedure

1. On each Cassandra node, remove the old installation. For example:

```
$ sudo yum remove dsc20
```

2. Install the new version.

```
$ sudo yum install dsc21
```

   The installer creates the file `cassandra.yaml.rpmnew` in `/etc/cassandra/default.conf/`.
3. Open the old and new `cassandra.yaml` files and diff them.
4. 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.
5. Save the file as cassandra.yaml.

Upgrading Tarball installations

Upgrade Cassandra with a tarball installation and preserve cassandra.yaml settings

About this task

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

cassandra.yaml

The location of the `cassandra.yaml` file depends on the type of installation:

Installer-Services	`/etc/cassandra/cassandra.yaml`
Package installations	`/etc/cassandra/cassandra.yaml`
Installer-No Services	*install_location*/conf/cassandra.yaml
Tarball installations	*install_location*/conf/cassandra.yaml

Procedure

1. Save the cassandra.yaml file from the old installation to a safe place.
2. On each node, download and unpack the binary tarball package from the downloads section of the Cassandra website.
3. In the new installation, open the `cassandra.yaml` for writing.
4. In the old installation, open the `cassandra.yaml`.

5. Diff the new and old `cassandra.yaml` files.
6. 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.
7. 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
   ```

Changes impacting upgrade

Considerations when upgrading to Cassandra.

Changes that can affect upgrading to 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.
- Cassandra 2.0 and above has 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 data center names. In this release,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 having loose type validation of CQL constants that now have strong validation.

  Using BLOBs as string constants is deprecated in favor of blob constants.
- 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, 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`.

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

- The nodetool upgradesstables command only upgrades/rewrites 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 versions of Cassandra 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 (to enable, if you make many requests for rows that do not exist).

cassandra.yaml

The location of the `cassandra.yaml` file depends on the type of installation:

Installer-Services	`/etc/cassandra/cassandra.yaml`
Package installations	`/etc/cassandra/cassandra.yaml`
Installer-No Services	*`install_location`*`/conf/cassandra.yaml`
Tarball installations	*`install_location`*`/conf/cassandra.yaml`

Upgrading OpsCenter

Cassandra and DataStax Enterprise versions compatible with OpsCenter versions.

Use the information in this section to upgrade to OpsCenter 5.1 from earlier versions.

OpsCenter Compatibility

Cassandra and DataStax Enterprise versions compatible with OpsCenter versions.

Before upgrading, make sure that your Cassandra or DataStax Enterprise version is compatible with the upgraded OpsCenter version.

OpsCenter version	Cassandra version	DataStax Enterprise version
5.1.3	1.2, 2.0	3.1 3.2, 4.0, 4.5, 4.6, 4.7
5.1.2	1.2, 2.0	3.1 3.2, 4.0, 4.5, 4.6, 4.7
5.1	1.2, 2.0	3.1, 3.2 4.0, 4.5, 4.6
5.0.2	1.2, 2.0	3.1, 3.2, 4.0, 4.5, 4.6
5.0	1.2, 2.0	3.1, 3.2, 4.0, 4.5
4.1	1.1, 1.2, 2.0	3.0, 3.1, 3.2, 4.0
4.0	1.1, 1.2, 2.0	3.0, 3.1. 3.2
3.2	1.1, 1.2	2.0, 3.0, 3.1
3.1	1.0, 1.1, 1.2	1.0, 2.0, 3.0
3.0	1.0, 1.1, 1.2	1.0, 2.0, 3.0
2.1	0.8, 1.0, 1.1	1.0, 2.0
2.0	0.8, 1.0, 1.1	1.0, 2.0
1.4	0.7, 0.8, 1.0, 1.1	1.0, 2.0

Upgrading package installations

Steps to upgrade to OpsCenter 5.1.

About this task

These steps provide information on upgrading to OpsCenter 5.1 package and restarting the `opscenterd` daemon.

Procedure

1. Be sure that OpsCenter is compatible with your version of DataStax Enterprise or Cassandra.
2. If you have made changes to the `address.yaml` or `cluster_name.conf` configuration files, see OpsCenter 5.1 configuration file changes.
3. 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
  ```

4. Install the upgraded OpsCenter package:

   - **Debian or Ubuntu**:

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

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

   ```
   # service opscenterd restart
   ```
7. On RHEL or CentOS, if you encounter an error in pyOpenSSL when starting OpsCenter after upgrading, uninstall OpsCenter, delete the `/usr/share/opscenter/lib` directory, and re-install.

   a) Uninstall OpsCenter.

   Debian or Ubuntu

   ```
   # apt-get remove opscenter
   ```

   RHEL or CentOS

   ```
   # yum uninstall opscenter
   ```

   b) Delete the `/usr/share/opscenter/lib` directory.

   ```
   # rm -rf /usr/share/opscenter/lib
   ```

   c) Reinstall OpsCenter.

   Debian or Ubuntu

   ```
   # apt-get install opscenter
   ```

   RHEL or CentOs

   ```
   # yum install opscenter
   ```

Upgrading tarball installations

Upgrade the OpsCenter 5.1 tarball and restart the `opscenterd` daemon.

About this task

Procedure

1. Be sure that OpsCenter is compatible with your version of DataStax Enterprise or Cassandra.
2. If you have made changes to the `address.yaml` or `cluster_name.conf` configuration files, see OpsCenter 5.1 configuration file changes.
3. Download and extract the new tarball.

4. 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/log4j.properties
   conf/opscenterd.conf
   ./passwd.db
   conf/ssl.conf
   ssl/*
   ```

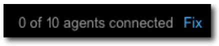
5. Stop the `opscenterd` instance (if it is running) and start it from the new tarball installation directory.
6. Upgrade the agents either through the GUI or by manually installing from the new tarballs.

Upgrading agents

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

About this task

If DataStax agents require upgrading for the new release, you are prompted to do so by a Fix link located near the top of the OpsCenter console.



0 of 10 agents connected Fix

Procedure

For information about installing the upgraded agents, see Installing OpsCenter agents.

From tarballs

If the agents will be upgraded 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/*
```

Changes affecting upgrades to OpsCenter 5.1 configuration files

Changes affecting upgrades to OpsCenter 5.1 configuration files.

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 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.

address.yaml

The location of the address.yaml file depends on the type of installation:

Installer-Services	/var/lib/datastax-agent/conf/address.yaml
Package installations	/var/lib/datastax-agent/conf/address.yaml
Installer-No Services	*install_location*/conf/address.yaml
Tarball installations	*install_location*/conf/address.yaml

cluster_name.conf

The location of the *cluster_name*.conf file depends on the type of installation:

Installer-Services	/etc/opscenter/clusters/*cluster_name*.conf
Package installations	/etc/opscenter/clusters/*cluster_name*.conf
Installer-No Services	*install_location*/conf/clusters/*cluster_name*.conf
Tarball installations	*install_location*/conf/clusters/*cluster_name*.conf

Tips for using DataStax documentation

Navigating the documents

To navigate, use the table of contents or search in the left navigation bar. Additional controls are:

|↔|	Hide or display the left navigation.
« »	Go back or forward through the topics as listed in the table of contents.
✎	Toggle highlighting of search terms.
🖶	Print page.
🐦	See doc tweets and provide feedback.
⋮	Grab to adjust the size of the navigation pane.
¶	Appears on headings for bookmarking. Right-click the ¶ to get the link.
⊕	Toggles the legend for CQL statements and nodetool options.

Other resources

You can find more information and help at:

- Documentation home page
- Datasheets
- Webinars
- Whitepapers
- Developer blogs
- Support