## Table of Contents

About upgrading .......................................................................................................................... 5
Supported & compatible versions ................................................................................................. 6
Upgrading DSE .......................................................................................................................... 9

- Planning your DSE upgrade ................................................................................................. 9
- Upgrades for patch releases ............................................................................................... 10
  - 6.7.x patch releases ........................................................................................................ 11
  - 6.0.x patch releases ........................................................................................................ 14
  - 5.1.x patch releases ........................................................................................................ 18
  - 5.0.x patch releases ........................................................................................................ 25
  - 4.8.x patch releases ........................................................................................................ 31
- Upgrading from DSE 6.0 to 6.7 ............................................................................................ 35
- Upgrading from DSE 5.1 to 6.7 ............................................................................................ 43
- Upgrading from DSE 5.0 to 6.7 ............................................................................................ 55
- Upgrading from DSE 5.1 to 6.0 ............................................................................................ 78
- Upgrading from DSE 5.0 to 6.0 ............................................................................................ 89
- Upgrading from DataStax Installer ........................................................................................ 110
  - Converting to Yum installation on RHEL-based systems .................................................. 112
  - Converting to APT installation on Debian-based systems ............................................. 113
  - Converting to binary tarball installation ......................................................................... 114
- Upgrading from DSE 5.0 to 5.1 ............................................................................................ 115
- Upgrading to earlier versions ............................................................................................... 133
  - Upgrading to DSE 5.0 ....................................................................................................... 133
  - Upgrading from DSE 4.7 to 4.8 ....................................................................................... 143
  - Upgrading to DSE 4.7 or 4.8 ........................................................................................... 149
  - Upgrading to DSE 4.6 ....................................................................................................... 155
  - Upgrading to DSE 4.0 or 4.5 ........................................................................................... 162
  - Upgrading to DSE 3.2 ....................................................................................................... 171
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upgrading 4.5-5.1 using the DataStax installer</td>
<td>184</td>
</tr>
<tr>
<td>Rolling back an upgrade</td>
<td>186</td>
</tr>
<tr>
<td>Reverting a package installation</td>
<td>186</td>
</tr>
<tr>
<td>Reverting a tarball installation</td>
<td>187</td>
</tr>
<tr>
<td>Upgrading from Cassandra to DSE</td>
<td>189</td>
</tr>
<tr>
<td>Upgrading DDAC and Apache Cassandra</td>
<td>194</td>
</tr>
<tr>
<td>Planning your DDAC upgrade</td>
<td>194</td>
</tr>
<tr>
<td>Upgrading DDAC</td>
<td>194</td>
</tr>
<tr>
<td>Upgrading Cassandra</td>
<td>197</td>
</tr>
<tr>
<td>Upgrading OpsCenter</td>
<td>198</td>
</tr>
<tr>
<td>Before upgrading OpsCenter</td>
<td>198</td>
</tr>
<tr>
<td>DSE OpsCenter upgrade instructions</td>
<td>199</td>
</tr>
<tr>
<td>Upgrading package installations</td>
<td>199</td>
</tr>
<tr>
<td>Upgrading tarball installations</td>
<td>201</td>
</tr>
<tr>
<td>Upgrading when failover is enabled</td>
<td>202</td>
</tr>
<tr>
<td>Upgrading DataStax Agents</td>
<td>202</td>
</tr>
<tr>
<td>Upgrading DataStax Studio</td>
<td>204</td>
</tr>
<tr>
<td>Upgrading the DataStax AMI</td>
<td>205</td>
</tr>
</tbody>
</table>
About upgrading

The *DataStax Upgrade Guide* provides detailed instructions on upgrading DataStax Enterprise, OpsCenter, DataStax Agents, DataStax drivers, and reverting to earlier versions.

**Attention: Read and understand these instructions before upgrading.** Carefully reviewing the planning (*page 9*) and upgrading instructions can ensure a smooth upgrade and avoid pitfalls and frustrations.

Upgrade instructions

- Upgrading DataStax Enterprise (*page 9*)
- Upgrading DSE OpsCenter (*page 198*)
- DataStax driver changes
- Upgrading from Apache Cassandra to DataStax Enterprise (*page 189*)
- Upgrading Apache Cassandra™
Supported and compatible product versions

DataStax Enterprise, Apache Cassandra, CQL, and SSTable compatibility

The following tables list each supported version of DataStax Enterprise (DSE) and its corresponding Apache Cassandra™, CQL, and SSTable versions:

End of Life (EOL) — Patches/bug fixes not available.
End of Service Life (EOSL) — Patches/bug fixes and support not available. Documentation updates are not provided.

**Table 1: Current versions**

<table>
<thead>
<tr>
<th>DataStax Enterprise (DSE)</th>
<th>Apache Cassandra</th>
<th>CQL</th>
<th>SSTable format note (page 7)</th>
<th>SSTable version</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.7</td>
<td></td>
<td>3.4.5 + enhancements</td>
<td>bti</td>
<td>a</td>
</tr>
<tr>
<td>6.0</td>
<td></td>
<td>3.4 + enhancements</td>
<td>bti</td>
<td>a</td>
</tr>
<tr>
<td>5.1</td>
<td></td>
<td>3.4 + enhancements</td>
<td>big</td>
<td>m</td>
</tr>
</tbody>
</table>

**Table 2: End of Service Life (EOSL) versions**

<table>
<thead>
<tr>
<th>DataStax Enterprise (version and date)</th>
<th>Apache Cassandra</th>
<th>CQL</th>
<th>SSTable format note (page 7)</th>
<th>SSTable version</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.0 (2018-12-05)</td>
<td></td>
<td>3.3</td>
<td>big</td>
<td>m</td>
</tr>
<tr>
<td>4.8 (2018-10-16)</td>
<td>2.1 + enhancements</td>
<td>3.1</td>
<td>big</td>
<td>k</td>
</tr>
</tbody>
</table>
### Supported and compatible product versions

<table>
<thead>
<tr>
<th>DataStax Enterprise (version and date)</th>
<th>Apache Cassandra</th>
<th>CQL</th>
<th>SSTable format note (page 7)</th>
<th>SSTable version</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.7 (2017-10-18)</td>
<td>2.1</td>
<td>3.1</td>
<td>big</td>
<td>k</td>
</tr>
<tr>
<td>4.6 (2016-12-28)</td>
<td>2.0</td>
<td>3.1</td>
<td>big</td>
<td>j</td>
</tr>
<tr>
<td>4.5 (2016-06-29)</td>
<td>2.0</td>
<td>3.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.0 (2016-02-25)</td>
<td>2.0</td>
<td>3.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.2 (2015-11-13)</td>
<td>1.2</td>
<td>3.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.1 (2015-07-08)</td>
<td>1.2</td>
<td>3.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.0 (2015-02-25)</td>
<td>1.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.2 (2014-10-04)</td>
<td>1.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1 (2014-07-23)</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** Verify compatibility with the current DSE version when using nodetool upgradesstables, sstableloader, and sstableupgrade. See the SSTable compatibility and upgrade version table 6.7 | 6.0.

### DSE OpsCenter compatibility with DSE

This table lists OpsCenter compatibility for both monitored and storage clusters (if one or more are in use).

<table>
<thead>
<tr>
<th>OpsCenter version</th>
<th>DSE version</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.7</td>
<td>6.7, 6.0, 5.1</td>
</tr>
<tr>
<td>6.5</td>
<td>6.0, 5.1, 5.0 (EOL)</td>
</tr>
<tr>
<td>6.1</td>
<td>5.1, 5.0 (EOL), 4.8 (EOSL)</td>
</tr>
<tr>
<td>6.0</td>
<td>5.0 (EOL), 4.8 (EOSL), 4.7 (EOSL)</td>
</tr>
</tbody>
</table>

**Note:** OpsCenter is only compatible with DataStax Enterprise (DSE) clusters.
DataStax Studio compatibility with DSE

The following table lists DataStax Studio compatibility with DSE:

<table>
<thead>
<tr>
<th>Studio version</th>
<th>DSE version</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.7</td>
<td>6.7</td>
</tr>
<tr>
<td>6.0</td>
<td>6.0</td>
</tr>
<tr>
<td>2.0</td>
<td>5.1</td>
</tr>
<tr>
<td>1.0</td>
<td>5.0 (EOL)</td>
</tr>
</tbody>
</table>

DataStax driver compatibility with DSE and Apache Cassandra

See Version compatibility.

DataStax support for Apache Cassandra

- The DataStax Distribution of Apache Cassandra (DDAC) is based on the DataStax enhanced version of Apache Cassandra™ 3.11 included in DataStax Enterprise (DSE) 5.1. Patch releases for DDAC and maintenance longevity correspond to the DSE 5.1 patches and maintenance schedule.
- DDAC supports the DataStax drivers for Apache Cassandra and DataStax Bulk Loader (dsbulk).
- DataStax does not support open-source Cassandra. You can download the latest open-source versions from Apache Cassandra.
- As of June 2019, DataStax produced documentation for Cassandra 2.1, Cassandra 3.0, CQL 3.3, and CQL 3.1 are no longer supported or maintained. Content for the following DataStax produced Cassandra documents were last updated in early January 2017:
  - # Cassandra 3.x Linux and Windows
  - # Cassandra 3.0 Windows
  - # Cassandra 2.2 Linux and Windows
  - # Cassandra 2.0
  - # Cassandra 1.2
- For open-source Cassandra, see the Apache Cassandra documentation.
Upgrading DataStax Enterprise

This section describes how to upgrade DataStax Enterprise.

**Attention: Read and understand these instructions before upgrading.** Carefully reviewing the planning (page 9) and upgrading instructions can ensure a smooth upgrade and avoid pitfalls and frustrations.

To upgrade from any version of DataStax Community or DataStax Distribution of Apache Cassandra™ to DataStax Enterprise, follow the instructions in Upgrading from Apache Cassandra to DataStax Enterprise (page 189).

Planning your DataStax Enterprise upgrade

The upgrade process for DataStax Enterprise provides minimal downtime (ideally zero). During this process, 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 to provide access to product improvements and new features and reduce version impacts. Ensure that you repair your nodes regularly. Node repair ensures that data on a replica is consistent with data on other 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. Depending on the driver version, you might need to recompile your client application code. See DataStax driver changes.

Upgrades from DSE 4.8 require an interim upgrade to DSE 5.0 (page 133).

**DataStax Installer End-of-life (EOL)**
The DataStax Installer is not supported for DSE 6.0 and later. When DSE 5.1 or DSE 5.0 was installed with the DataStax Installer, you must first change from a standalone installer installation to tarball or package installation for the same DSE version before you can update to DSE 6.0 or later. See Upgrading to DSE 6.0 or DSE 6.7 from DataStax Installer installations (page 110).

**Upgrade to the latest patch release on your current version**
When upgrading to a major version, first upgrade to the latest (page 10) patch release on your current version. Fixes included in the latest patch release can help or smooth the upgrade process.
You can use OpsCenter 6.5 Lifecycle Manager (LCM) to clone a configuration profile and run an upgrade job on a datacenter or node. Upgrade jobs are supported for upgrades within a DSE release series for DSE 5.0.x and later.

The latest DataStax Enterprise versions are:

- DSE 6.7.3
- DSE 6.0.8
- DSE 5.1.15
- DSE 5.0.15

**Backup data**

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

**Tip:** OpsCenter provides a Backup Service that manages enterprise-wide backup and restore operations for DataStax Enterprise clusters. OpsCenter 6.5 and later is recommended.

**Upgrade order**

Upgrade order matters. Upgrade nodes in this order:

1. In multiple datacenter clusters, upgrade every node in one datacenter before moving on to other datacenters.

2. Upgrade the seed nodes within a datacenter first.

3. Upgrade node types in this order:
   
   a. DSE Analytics nodes or datacenters.
      
      For DSE Analytics nodes using DSE Hadoop, upgrade the Job Tracker node first. Then upgrade Hadoop nodes, followed by Spark nodes.

   b. Transactional/DSE Graph nodes or datacenters.

   c. DSE Search nodes or datacenters.

**Upgrades for DataStax Enterprise patch releases**

These topics provide information on upgrading DataStax Enterprise between patch (point) releases, such as upgrading from DataStax Enterprise from 5.1.2 to 5.1.9 or 6.0.0 to 6.0.4.
Attention: Read and understand these instructions before upgrading. Carefully reviewing the planning (page 9) and upgrading instructions can ensure a smooth upgrade and avoid pitfalls and frustrations.

Upgrades for DataStax Enterprise 6.7.x patch releases

Review this information on upgrading DataStax Enterprise (DSE) between patch (point) releases, such as upgrading from DSE 6.7.0 to 6.7.1.

Attention: Read and understand these instructions before upgrading. Carefully reviewing the planning (page 9) and upgrading instructions can ensure a smooth upgrade and avoid pitfalls and frustrations.

General recommendations

Be sure to read the DataStax Enterprise 6.7 release notes.

DataStax recommends upgrading to the latest DSE 6.7.3.

General restrictions and limitations during the upgrade process

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

Restrictions for all nodes during an upgrade

- Do not enable new features.
- Do not run nodetool repair.
- Do not bootstrap or decommission nodes.
- 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.

Note: Nodes on different versions might show a schema disagreement during an upgrade.

DSE Search (Solr) upgrade restrictions and limitations

- Do not update schemas.
- Do not reindex 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.
- If you are not already using Kerberos, do not set up Kerberos authentication before upgrading. First upgrade the cluster, then set up Kerberos.

DSE Graph
Upgrading DataStax Enterprise

- Because DSE Graph is generally run with other workloads, follow the same upgrade limitations for those workloads.
- Do not create or update any graph or other schemas during the upgrade.
- You can safely ignore gremlin and graph related errors during the upgrade.

Preparing to upgrade

1. **Back up** your data. DataStax recommends backing up your data prior to any version upgrade, including logs and custom configurations. A backup provides the ability to revert and restore all the data used in the previous version if necessary.

   **Tip:** OpsCenter provides a Backup Service that manages enterprise-wide backup and restore operations for DataStax Enterprise clusters. OpsCenter 6.7 and later is recommended.

2. **Verify your current product version:**

   ```
   $ dse -v
   ```

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

   The required space depends on the compaction strategy. See [Disk space](#).

4. Familiarize yourself with the changes and features in DataStax Enterprise and Apache Cassandra™. See:

   - DataStax Enterprise release notes for the upgrade version and complete all required actions.
     
     **DataStax Enterprise 6.7 release notes** include required planning, components, changes and enhancements, known issues, and resolved issues.

   - 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 driver changes.

5. Back up the configuration files to a folder that is not in the directory where you normally run commands.

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

6. Run `nodetool repair` to ensure that data on each replica is consistent with data on other nodes.

7. **DSE Search nodes:**
• Ensure that the `catalina.properties` and `context.xml` files are present in the Tomcat `conf` dir. DSE will not start after upgrade if these files are missing.

The default location of the Tomcat `conf` directory depends on the type of installation:

# Package installations: `/etc/dse/tomcat/conf`
# Tarball installations: `installation_location/resources/tomcat/conf`

Upgrade steps

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

1. If you have the OpsCenter Repair Service configured, turn off the Repair Service.

2. Upgrade order matters. Upgrade nodes in this order:
   - In multiple datacenter clusters, upgrade every node in one datacenter before upgrading another datacenter.
   - Upgrade the seed nodes within a datacenter first.
   - Upgrade nodes in this order:
     a. DSE Analytics datacenters
     b. Transactional/DSE Graph datacenters
     c. DSE Search datacenters

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

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

This step saves time when nodes start up after the upgrade, and prevents DSE Search nodes from reindexing data.

4. Stop the node.

5. Use the appropriate method to install the new product version.

6. 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.
• Check for any other configuration files that you might have changed. See Default file locations.

b. Merge the applicable modifications into the new version.

7. Start the node.

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

   $ nodetool status

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

   Warnings, errors, and exceptions are frequently found in the logs when starting 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.

10. Repeat the upgrade and restart on each node (rolling restart) in the cluster following the recommended upgrade order as described in 2 (page 13).

11. After the new version is installed on each node, DataStax recommends upgrading the SSTables on each node.

    Upgrading SSTables is recommended for optimal performance, but is not required for patch releases.

    $ nodetool upgradesstables

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

    Use the --jobs option to set the number of SSTables that upgrade simultaneously. The default setting is 2, which minimizes impact on the cluster. Set to 0 to use all available compaction threads. For information about nodetool upgradesstables, including how to speed it up, see the DataStax Support KB article Nodetool upgradesstables FAQ.

12. If you use the OpsCenter Repair Service, turn on the Repair Service.

**Upgrades for DataStax Enterprise 6.0.x patch releases**

Review this information on upgrading DataStax Enterprise (DSE) between patch (point) releases, such as upgrading from DSE 6.0.0 to 6.0.2.
Attention: Read and understand these instructions before upgrading. Carefully reviewing the planning (page 9) and upgrading instructions can ensure a smooth upgrade and avoid pitfalls and frustrations.

General recommendations

Be sure to read the DataStax Enterprise 6.0 release notes.

DataStax recommends upgrading to the latest DSE 6.0.8.

General restrictions and limitations during the upgrade process

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

Restrictions for all nodes during an upgrade

- Do not enable new features.
- Do not run nodetool repair.
- Do not bootstrap or decommission nodes.
- 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.

Note: Nodes on different versions might show a schema disagreement during an upgrade.

DSE Analytics (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 reindex 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.
- If you are not already using Kerberos, do not set up Kerberos authentication before upgrading. First upgrade the cluster, then set up Kerberos.

DSE Graph

- Because DSE Graph is generally run with other workloads, follow the same upgrade limitations for those workloads.
- Do not create or update any graph or other schemas during the upgrade.
• You can safely ignore gremlin and graph related errors during the upgrade.

Preparing to upgrade

1. **Back up** your data. DataStax recommends backing up your data prior to any version upgrade, including logs and custom configurations. A backup provides the ability to revert and restore all the data used in the previous version if necessary.

   **Tip:** OpsCenter provides a **Backup Service** that manages enterprise-wide backup and restore operations for DataStax Enterprise clusters. OpsCenter 6.5 and later is recommended.

2. **Verify your current product version:**

   ```
   $ dse -v
   ```

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

   The required space depends on the compaction strategy. See [Disk space](#).

4. Familiarize yourself with the changes and features in DataStax Enterprise and Apache Cassandra™. See:
   - DataStax Enterprise release notes for the upgrade version and complete all required actions.
     [DataStax Enterprise 6.0 release notes](#) include required planning, components, changes and enhancements, known issues, and resolved issues.
   - 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 driver changes.

5. Back up the configuration files to a folder that is not in the directory where you normally run commands.

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

6. Run nodetool repair to ensure that data on each replica is consistent with data on other nodes.

7. **DSE Search nodes:**
   - Ensure that the catalina.properties and context.xml files are present in the Tomcat conf dir. DSE will not start after upgrade if these files are missing.
The default location of the Tomcat conf directory depends on the type of installation:

# Package installations: /etc/dse/tomcat/conf
# Tarball installations: installation_location/resources/tomcat/conf

Upgrade steps

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

1. If you have the OpsCenter Repair Service configured, turn off the Repair Service.

2. Upgrade order matters. Upgrade nodes in this order:
   - In multiple datacenter clusters, upgrade every node in one datacenter before upgrading another datacenter.
   - Upgrade the seed nodes within a datacenter first.
   - Upgrade nodes in this order:
     a. DSE Analytics datacenters
     b. Transactional/DSE Graph datacenters
     c. DSE Search datacenters

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

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

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

4. Stop the node.

5. Use the appropriate method to install the new product version.

6. 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.
• Check for any other configuration files that you might have changed. See Default file locations.

b. Merge the applicable modifications into the new version.

7. Start the node.

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

   $ nodetool status

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

   Warnings, errors, and exceptions are frequently found in the logs when starting 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.

10. Repeat the upgrade and restart on each node (rolling restart) in the cluster following the recommended upgrade order as described in 2 (page 17).

11. After the new version is installed on each node, DataStax recommends upgrading the SSTables on each node.

   Upgrading SSTables is recommended for optimal performance, but is not required for patch releases.

   $ nodetool upgradesstables

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

   Use the --jobs option to set the number of SSTables that upgrade simultaneously. The default setting is 2, which minimizes impact on the cluster. Set to 0 to use all available compaction threads. For information about nodetool upgradesstables, including how to speed it up, see the DataStax Support KB article Nodetool upgradesstables FAQ.

12. If you use the OpsCenter Repair Service, turn on the Repair Service.

**Upgrades for DataStax Enterprise 5.1.x patch releases**

Review this information on upgrading DataStax Enterprise (DSE) between patch (point) releases, such as upgrading from DataStax Enterprise 5.1.3 to 5.1.5.
Attention: Read and understand these instructions before upgrading. Carefully reviewing the planning (page 9) and upgrading instructions can ensure a smooth upgrade and avoid pitfalls and frustrations.

General recommendations

Warning: TTL expiration timestamps are susceptible to the year 2038 problem. If the TTL value is long and an expiration date that is greater than the maximum threshold of 2038-01-19T03:14:06+00:00, the data is immediately expired and purged on the next compaction. DataStax strongly recommends upgrading to DSE 5.1.7 or later and taking required action to protect against silent data loss. (DSP-15412).

Attention: DataStax recommends upgrading to the latest DSE 5.1.15.

OpsCenter provides a Backup Service that manages enterprise-wide backup and restore operations for DataStax Enterprise clusters. OpsCenter 6.5 and later is recommended.

General restrictions and limitations during the upgrade process

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

Restrictions for all nodes during an upgrade

- Do not enable new features.
- Do not run nodetool repair.
- Do not bootstrap or decommission nodes.
- 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.

Note: Nodes on different versions might show a schema disagreement during an upgrade.

DSE Analytics (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 reindex 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.
• If you are not already using Kerberos, do not set up Kerberos authentication before upgrading. First upgrade the cluster, then set up Kerberos.

**DSE Graph**

• Because DSE Graph is generally run with other workloads, follow the same upgrade limitations for those workloads.
• Do not create or update any graph or other schemas during the upgrade.
• You can safely ignore gremlin and graph related errors during the upgrade.

**Special upgrade restrictions for 5.1.x patch releases**

**Restrictions when upgrading to DSE 5.1.0-5.1.5**
DSE Search might miss token filtering on mixed versions clusters. Upgrade all nodes to DSE 5.1.6 or later for correct token filtering.

**Restrictions when upgrading to DSE 5.1.4**
DSE Analytics Spark Jobserver uses DSE custom version 0.7.0.125. Applications must use the compatible Spark Jobserver API in the DataStax repository.

**Restrictions when upgrading to DSE 5.1.6 and later**
DSE Analytics DSEFS authorization is enabled only if DSE authorization is enabled. Clusters with DSE authentication enabled and DSE authorization disabled will not have DSEFS authorization after upgrading to 5.1.6 and later. In versions prior to DSE 5.1.6 DSEFS, authorization was enabled when DSE authentication was enabled, and enabling or disabling DSE authorization had no effect on DSEFS.

**Restrictions when upgrading to DSE 5.1.3**

• Creating Materialized View with filtering on non-primary-key base column (CASSANDRA-10368) is disabled, because the liveness of view row is depending on multiple filtered base non-key columns and base non-key column used in view primary-key. This semantic cannot be supported without storage format change (CASSANDRA-13826). For append-only use case, you can still use this feature with a system property switch:

```
-Dcassandra.mv.allow_filtering_nonkey_columns_unsafe=true
```

• The table system_auth.resource_role_permissions_index is no longer used and must be dropped after all nodes are on 5.1.3.
• Non-incremental full repairs are now the default if no option is specified on nodetool repair, unless incremental repair was already run on the table/keyspace being repaired, to maintain backward compatibility. To run incremental repair on new tables, use the #inc option.
• Incremental repairs are no longer supported on tables with materialized views or CDC until its limitations are addressed. An incremental repair triggered on a base table or materialized view run a full repair instead (CASSANDRA-12888).

**Restrictions when upgrading from DSE 5.1.1 or 5.1.2 to DSE 5.1.3**

• Apache Cassandra no longer allows dropping columns on tables with Materialized Views.
• A change was made in the way the Materialized View timestamp is computed. This change can cause an old deletion to a base column, which is a view primary key (PK) column, to not be reflected in the view if you
repair the base table after the upgrade. This condition only occurs when a column deletion to an MV primary key (PK) column not present in the base table PK (via `UPDATE base SET view_pk_col = null` or `DELETE view_pk_col FROM base`) is missed before the upgrade and repaired after the upgrade. If such column deletions are done on a view PK column that is not a base PK, run repair on the base table of all nodes prior to the upgrade. Alternatively, you can fix potential inconsistencies by running repair on the views after upgrade or drop and re-create the views. For more details, see CASSANDRA-11500.

- **Removal of columns not selected in the Materialized View** (via `UPDATE base SET unselected_column = null` or `DELETE unselected_column FROM base`) may not be properly reflected in the view in some situations. DataStax advises against deletions on base columns not selected in views until fixed in CASSANDRA-13826.

Advanced preparation for upgrading from DSE 5.1.1 to 5.1.2 and later 5.1.x releases

This section applies only when upgrading from **DSE 5.1.1 to 5.1.2 and later DSE 5.1.x** releases.

The messaging protocol version in DSE 5.0.9 and DSE 5.1.2 has been changed to VERSION_3014. Schema migrations rely on exact messaging protocol versions. To accommodate schema changes that might occur during the upgrade, force a backward compatible messaging protocol.

1. **Before** you upgrade, restart the node with this start-up parameter:

   ```
   -Dcassandra.force_3_0_protocol_version=true
   ```

   For example:

   ```
   $ installation_location/bin/dse cassandra -Dcassandra.force_3_0_protocol_version=true
   ```

   **Note:** While mixed versions exist during the upgrade, do not add or remove columns from existing tables.

2. After the upgrade is complete on **all** nodes, follow the upgrade steps to restart nodes without this start-up parameter.

Preparing to upgrade

1. **Back up** your data. DataStax recommends backing up your data prior to any version upgrade, including logs and custom configurations. A backup provides the ability to revert and restore all the data used in the previous version if necessary.
Tip: OpsCenter provides a Backup Service that manages enterprise-wide backup and restore operations for DataStax Enterprise clusters. OpsCenter 6.5 and later is recommended.

2. Verify your current product version:

   $ dse -v

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

   The required space depends on the compaction strategy. See Disk space.

4. Familiarize yourself with the changes and features in DataStax Enterprise and Apache Cassandra™:

   • DataStax Enterprise release notes for the upgrade version and complete all required actions.

      DSE release notes include required planning, components, changes and enhancements, known issues, and resolved issues. See 5.1, 5.0, and 4.8.

   • 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 DSE release notes.

   • DataStax driver changes.

5. DSE Search nodes:

   • Ensure that the catalina.properties and context.xml files are present in the Tomcat conf dir. DSE will not start after upgrade if these files are missing.

      The default location of the Tomcat conf directory depends on the type of installation:

      # Package installations: /etc/dse/tomcat/conf
      # Tarball installations: installation_location/resources/tomcat/conf

   • Tune the schema before you upgrade. For DSE 5.1.4 and later, all field definitions in the schema are validated and must be DSE Search compatible, even if the fields are not indexed, have docValues applied, or used for copy-field source.

   • The default behavior of automatic resource generation includes all columns. To improve performance, take action to prevent the fields from being loaded from the database to the indexing path. In the schema, remove or comment out unused fields.
Upgrading DataStax Enterprise

- 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 you changed the schema, do a full reindex.

6. Back up the configuration files to a folder that is not in the directory where you normally run commands.

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

   You can use OpsCenter 6.5 Lifecycle Manager (LCM) to clone a configuration profile and run an upgrade job on a datacenter or node. Upgrade jobs are supported for upgrades within a DSE release series for DSE 5.0.x and later.

7. Run `nodetool repair` to ensure that data on each replica is consistent with data on other nodes.

Upgrade steps

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

Tip: The DataStax installer (page 184) automatically performs many upgrade tasks.

1. If you have the OpsCenter Repair Service configured, turn off the Repair Service.

2. Upgrade order matters. Upgrade nodes in this order:
   - In multiple datacenter clusters, upgrade every node in one datacenter before upgrading another datacenter.
   - Upgrade the seed nodes within a datacenter first.
   - Upgrade nodes in this order:
     a. DSE Analytics datacenters
     b. Transactional/DSE Graph datacenters
     c. DSE Search datacenters

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

   ```
   $ nodetool -h hostname drain
   ```
This step saves time when nodes start up after the upgrade, and prevents DSE Search nodes from having to reindex data.

4. **Stop the node.**

5. Use the appropriate method to install the new product version.

6. 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.
      - Check for any other configuration files that you might have changed. See Default file locations for Installer-Services and package installations or Installer-No Services and tarball installations.

   b. Merge the applicable modifications into the new version.

7. **Start the node.**

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

   ```bash
   $ nodetool status
   ```

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

   Warnings, errors, and exceptions are frequently found in the logs when starting 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.

10. Repeat the upgrade and restart on each node in the cluster following the recommended upgrade order as described in 2 (page 23).

    Upgrading and restarting each node is called a rolling restart.

11. After the new version is installed on each node, DataStax recommends upgrading the SSTables on each node.

    Upgrading SSTables is recommended for optimal performance, but is not required for patch releases.

    ```bash
    $ nodetool upgradesstables
    ```
If the SSTables are already on the current version, the command returns immediately and no action is taken.

Use the `--jobs` option to set the number of SSTables that upgrade simultaneously. The default setting is 2, which minimizes impact on the cluster. Set to 0 to use all available compaction threads. For information about nodetool upgradesstables, including how to speed it up, see the DataStax Support KB article [Nodetool upgradesstables FAQ](#).

12. **DSE Search on DSE 5.1.6 and later** Slow startup on nodes with large encrypted indexes is resolved. However, action is required to realize the performance gains. You must do a full reindex of all encrypted search indexes on each node in your cluster. Plan sufficient time after the upgrade is complete to reindex with `deleteAll=true` in a rolling fashion. For example:

```
$ dsetool reload_core keyspace_name.table_name distributed=false reindex=true deleteAll=true
```

13. **DSE Search** Index time boost support is removed in DSE 5.1.1 and later. Use query time boosting instead. Delete any `_docBoost` columns in backing CQL tables.

Thrift tables where the `_docBoost` column existed will be allowed, but the `_docBoost` will be ignored. Thrift tables are not able to drop the column.

14. This step applies only to upgrades from DSE 5.1.0-5.1.2 to 5.1.3 and later DSE 5.1.x patch releases:

   To continue running incremental repairs, use `nodetool repair -inc`.

   **Important:** To migrate away from incremental repairs to enable non-incremental full (`-full`) or partition (`-pr`) repair, you must set all SSTables on all nodes as unrepaired using `nodetool mark_unrepaired` before running your first repair after upgrading.

15. If you use the OpsCenter Repair Service, turn on the Repair Service.

**Upgrades for DataStax Enterprise 5.0.x patch releases**

Review this information on upgrading DataStax Enterprise (DSE) between patch (point) releases, such as upgrading from DataStax Enterprise upgrading from 5.0.3 to 5.0.11.

DataStax recommends backing up your data prior to any version upgrade, including logs and custom configurations. 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. OpsCenter 6.5 and later is recommended.
**Attention: Read and understand these instructions before upgrading.** Carefully reviewing the planning (page 9) and upgrading instructions can ensure a smooth upgrade and avoid pitfalls and frustrations.

General recommendations

Be sure to read the DataStax Enterprise 5.0 release notes.

**Attention:** TTL expiration timestamps are susceptible to the year 2038 problem. If the TTL value is long and an expiration date that is greater than the maximum threshold of 2038-01-19T03:14:06+00:00, the data is immediately expired and purged on the next compaction. DataStax strongly recommends upgrading to DSE 5.0.15 or later and taking required action to protect against silent data loss. (DSP-15412).

**Attention:** DataStax recommends upgrading to DSE 5.0.15.

General restrictions and limitations during the upgrade process

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

**Restrictions for all nodes during an upgrade**

- Do not enable new features.
- Do not run nodetool repair.
- Do not bootstrap or decommission nodes.
- 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.

**Note:** Nodes on different versions might show a schema disagreement during an upgrade.

**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 reindex 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.
• If you are not already using Kerberos, do not set up Kerberos authentication before upgrading. First upgrade the cluster, then set up Kerberos.

**DSE Graph**

• Because DSE Graph is generally run with other workloads, follow the same upgrade limitations for those workloads.
• Do not create or update any graph or other schemas during the upgrade.
• You can safely ignore gremlin and graph related errors during the upgrade.

Advanced preparation for upgrading DSE 5.0.0-5.0.8 to 5.0.9 and later 5.0.x releases

This section applies **only** when upgrading from **DSE 5.0.0-5.0.8 to 5.0.9 and later 5.0.x releases**.

The messaging protocol version in DSE 5.0.9 and DSE 5.1.2 has been changed to VERSION_3014. Schema migrations rely on exact messaging protocol versions. To accommodate schema changes that might occur during the upgrade, force a backward compatible messaging protocol.

1. **Before** you upgrade, restart the node with this start-up parameter:

   ```
   -Dcassandra.force_3_0_protocol_version=true
   ```

   For example:

   ```
   $ installation_location/bin/dse cassandra "-Dcassandra.force_3_0_protocol_version=true"
   ```

   **Note:** While mixed versions exist during the upgrade, do not add or remove columns from existing tables.

2. After the upgrade is complete on **all** nodes, follow the upgrade steps to restart nodes without this start-up parameter.

Preparing to upgrade

1. **Back up** your data.

   OpsCenter provides a **Backup Service** that manages enterprise-wide backup and restore operations for DataStax Enterprise clusters. OpsCenter 6.5 and later is recommended.

2. **Verify your current product version:**

   ```
   $ dse -v
   ```

3. **Upgrade to the latest patch release on your current version.**

   The latest version of DSE 5.0 is 5.0.15.
You can use OpsCenter 6.5 Lifecycle Manager (LCM) to clone a configuration profile and run an upgrade job on a datacenter or node. Upgrade jobs are supported for upgrades within a DSE release series for DSE 5.0.x and later.

4. Before upgrading, be sure that each node has ample free disk space.
   The required space depends on the compaction strategy. See Disk space.

5. Familiarize yourself with the changes and features in DataStax Enterprise and Apache Cassandra™:
   - DataStax Enterprise release notes for the upgrade version and complete all required actions.
     DSE release notes include required planning, components, changes and enhancements, known issues, and resolved issues. See 5.1, 5.0, and 4.8.
   - 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 DSE release notes.
   - DataStax driver changes.

6. DSE Search nodes:
   - Tune the schema before you upgrade. For DSE 5.0.10 and later, all field definitions in the schema are validated and must be DSE Search compatible, even if the fields are not indexed, have docValues applied, or used for copy-field source.
   - The default behavior of automatic resource generation includes all columns. To improve performance, take action to prevent the fields from being loaded from the database to the indexing path. In the schema, remove or comment out unused fields.
   - 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 you changed the schema, do a full reindex.

7. Back up the configuration files you use to a folder that is not in the directory where you normally run commands.
   The configuration files are overwritten with default values during installation of the new version.
Upgrade steps

**Tip:** The DataStax installer *(page 184)* automatically performs many upgrade tasks.

1. Run `nodetool repair` to ensure that data on each replica is consistent with data on other nodes.

2. Upgrade order matters. Upgrade nodes in this order:
   
   - In multiple datacenter clusters, upgrade every node in one datacenter before upgrading another datacenter.
   - Upgrade the seed nodes within a datacenter first.
     
     For DSE Analytics nodes using DSE Hadoop, upgrade the Job Tracker node first. Then upgrade Hadoop nodes, followed by Spark nodes.
   
   - Upgrade nodes in this order:
     
     a. DSE Analytics datacenters
     b. Transactional/DSE Graph datacenters
     c. DSE Search datacenters

   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.

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 reindex data.

4. Stop the node.

5. Use the appropriate method to install the new product version.

6. 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.
• Check for any other configuration files that you might have changed. See Default file locations for Installer-Services and package installations or Installer-No Services and tarball installations.

  b. Merge the applicable modifications into the new version.

7. **Start the node.**

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

   ```
   $ nodetool status
   ```

9. **Review the logs for warnings, errors, and exceptions.**

   Warnings, errors, and exceptions are frequently found in the logs when starting 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](https://www.datastax.com/support).

10. **Repeat the upgrade and restart on each node in the cluster following the recommended upgrade order.**

    Upgrading and restarting each node is called a **rolling restart.**

11. After the new version is installed on each node, DataStax recommends upgrading the SSTables on each node.

    Upgrading SSTables is recommended for optimal performance, but is not required for patch releases.

    ```
    $ nodetool upgradesstables
    ```

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

    Use the **--jobs** option to set the number of SSTables that upgrade simultaneously. The default setting is 2, which minimizes impact on the cluster. Set to 0 to use all available compaction threads. For information about nodetool upgradesstables, including how to speed it up, see the DataStax Support KB article [Nodetool upgradesstables FAQ](https://www.datastax.com/support/kb/article/).

12. **This step applies only from DSE 5.0.0-5.0.9 to 5.0.10 and later 5.0.x patch releases:**

    To continue running incremental repairs, use `nodetool repair -inc`.

    **Important:** To migrate away from incremental repairs to enable non-incremental full (`-full`) or partition (`-pr`) repair, you must set all SSTables on all nodes as
Upgrading DataStax Enterprise

unrepaired using `nodetool mark_unrepaired` before running your first repair after upgrading.

13. If you use the OpsCenter Repair Service, turn on the Repair Service.

Upgrades for DataStax Enterprise 4.8.x patch releases

Review this information on upgrading DataStax Enterprise (DSE) between patch (point) releases, such as upgrading from DataStax Enterprise 4.8.4 to 4.8.12.

**Attention: Read and understand these instructions before upgrading.** Carefully reviewing the planning (page 9) and upgrading instructions can ensure a smooth upgrade and avoid pitfalls and frustrations.

**Attention:** TTL expiration timestamps are susceptible to the year 2038 problem. If the TTL value is long and an expiration date that is greater than the maximum threshold of `2038-01-19T03:14:06+00:00`, the data is immediately expired and purged on the next compaction. DataStax strongly recommends upgrading to DSE 4.8.16 and taking required action to protect against silent data loss. ([DSP-15412](#)).

General recommendations

Be sure to read the DataStax Enterprise 4.8 release notes.

**Attention:** DataStax recommends upgrading to DSE 4.8.16.

**Note:** DataStax recommends backing up your data prior to any version upgrade, including logs and custom configurations. 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. OpsCenter 6.5 and later is recommended.

General restrictions and limitations during the upgrade process

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

**Restrictions for all nodes during an upgrade**

- Do not enable new features.
- Do not run `nodetool repair`.
- Do not bootstrap or decommission nodes.
• 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.

Note: Nodes on different versions might show a schema disagreement during an upgrade.

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 reindex 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.
• If you are not already using Kerberos, do not set up Kerberos authentication before upgrading. First upgrade the cluster, then set up Kerberos.

Preparing to upgrade
1. **Back up** your data.
   
   OpsCenter provides a [Backup Service](#) that manages enterprise-wide backup and restore operations for DataStax Enterprise clusters. OpsCenter 6.5 and later is recommended.

2. Verify your current product version:

   ```
   $ dse -v
   ```

3. Upgrade to the latest patch release on your current version.
   
   The latest version of DSE 4.8 is 4.8.16.

4. Before upgrading, be sure that each node has ample free disk space.
   
   The required space depends on the compaction strategy. See [Disk space](#).

5. Familiarize yourself with the changes and features in DataStax Enterprise and Apache Cassandra™:
   
   • DataStax Enterprise release notes for the upgrade version and complete all required actions.
DSE release notes include required planning, components, changes and enhancements, known issues, and resolved issues. See 5.1, 5.0, and 4.8.

- **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 DSE release notes.
- DataStax driver changes.

### 6. DSE Search nodes:

- Tune the schema before you upgrade.

- 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 you changed the schema, do a full reindex.

### 7. Back up the configuration files you use to a folder that is not in the directory where you normally run commands.

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

### Upgrade steps

**Tip:** The DataStax installer *(page 184)* automatically performs many upgrade tasks.

1. Run `nodetool repair` to ensure that data on each replica is consistent with data on other nodes.

2. Upgrade order matters. Upgrade nodes in this order:
   - In multiple datacenter clusters, upgrade every node in one datacenter before upgrading another datacenter.
   - Upgrade the seed nodes within a datacenter first.

   For DSE Analytics nodes using DSE Hadoop, upgrade the Job Tracker node first. Then upgrade Hadoop nodes, followed by Spark nodes.

   - Upgrade nodes in this order:
     a. DSE Analytics datacenters
     b. Transactional/DSE Graph datacenters
c. DSE Search datacenters

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.

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 reindex data.

4. Stop the node.

5. Use the appropriate method to install the new product version.

6. 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.
      - Check for any other configuration files that you might have changed. See Default file locations for Installer-Services and package installation or Installer-No Services and tarball installations).

   b. Merge the applicable modifications into the new version.

7. Start the node.

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

   $ nodetool status

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

   Warnings, errors, and exceptions are frequently found in the logs when starting 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.

10. Repeat the upgrade and restart on each node in the cluster following the recommended upgrade order.
Upgrading and restarting each node is called a rolling restart.

11. After the new version is installed on each node, DataStax recommends upgrading the SSTables on each node.

Upgrading SSTables is recommended for optimal performance, but is not required for patch releases.

```
$ nodetool upgradesstables
```

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

Use the `--jobs` option to set the number of SSTables that upgrade simultaneously. The default setting is 2, which minimizes impact on the cluster. Set to 0 to use all available compaction threads. For information about `nodetool upgradesstables`, including how to speed it up, see the DataStax Support KB article Nodetool upgradesstables FAQ.

12. If you use the OpsCenter Repair Service, turn on the Repair Service.

Upgrading from DataStax Enterprise 6.0 to 6.7

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

Follow these instructions to upgrade from DataStax Enterprise (DSE) 6.0 to DSE 6.7.

Always upgrade to latest patch release (page 10) on your current version before you upgrade to a higher version. Fixes included in the latest patch release might help or smooth the upgrade process.

The latest 6.0.x version of DSE is 6.0.8.

**Attention: Read and understand these instructions before upgrading.** Carefully reviewing the planning (page 9) and upgrading instructions can ensure a smooth upgrade and avoid pitfalls and frustrations.

Apache Cassandra™ version change

SSTables must be upgraded for DataStax Enterprise upgrades between versions that include major Cassandra version changes.

- DataStax Enterprise 6.7 is compatible with Cassandra 3.11.
- DataStax Enterprise 6.0 is compatible with Cassandra 3.11.
- DataStax Enterprise 5.1 uses Cassandra 3.11.
- DataStax Enterprise 5.0 uses Cassandra 3.0.
Upgrading DataStax Enterprise

- DataStax Enterprise 4.7 to 4.8 use Cassandra 2.1.
- DataStax Enterprise 4.0 to 4.6 use Cassandra 2.0.

General recommendations

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

Tip: OpsCenter provides a Backup Service that manages enterprise-wide backup and restore operations for DataStax Enterprise clusters. OpsCenter 6.5 and later is recommended.

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 restrictions and limitations during the upgrade process

- Do not enable new features.
- Do not run nodetool repair. If you have the OpsCenter Repair Service configured, turn off the Repair Service.
- Ensure OpsCenter compatibility. OpsCenter 6.7 is required for managing DSE 6.7 clusters. See the compatibility table.
- During the upgrade, do not bootstrap or decommission nodes.
- Do not issue these types of CQL queries during a rolling restart: DDL and TRUNCATE.
- 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.
- NodeSync waits to start until all nodes are upgraded.

Note: Nodes on different versions might show a schema disagreement during an upgrade.

Restrictions for DSE Analytic (Spark) nodes

- Do not run analytics jobs until all nodes are upgraded.
- All nodes in the cluster must be upgraded to the new version before Spark Worker and Spark Master will start.

DSE Graph nodes restrictions

Graph nodes have the same restrictions as the workload they run on. Do not alter graph schema during upgrades. Workload-specific restrictions apply for analytics and search nodes, such as no OLAP queries during upgrades.

DSE Search upgrade restrictions and limitations

- Do not update schemas.
- Do not reindex DSE Search nodes during upgrade.

**Restrictions for nodes using any kind of security**
- Do not change security credentials or permissions until the upgrade is complete on all nodes.
- If you are not already using Kerberos, do not set up Kerberos authentication before upgrading. First upgrade the cluster, and then set up Kerberos.

**Upgrading drivers and possible impact when driver versions are incompatible**
Be sure to check [driver compatibility](#). Depending on the driver version, you might need to recompile your client application code. See DataStax driver changes.

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:

- **Protocol version**: Because some drivers can use different protocol versions, force the protocol version at start up. For example, keep the Java driver at its current protocol version while the driver upgrade is happening. Switch to the Java driver to the new protocol version only after the upgrade is complete on all nodes in the cluster.

- **Initial contact points**: 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 details on protocol version negotiation, see protocol versions with mixed clusters in the Java driver version you’re using, for example, [Java driver](#).

**Preparing to upgrade**
Follow these steps to prepare each node for upgrading from DSE 6.0 to DSE 6.7.

*Note*: These steps are performed in your current version and use DSE 6.0 documentation.

1. Carefully review [Planning your DataStax Enterprise upgrade](#) (page 9).

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

   The required space depends on the compaction strategy. See [Disk space](#).

3. Familiarize yourself with the changes and features in this release:
   - [DataStax Enterprise 6.7 release notes](#).
   - General upgrading advice for any version. Be sure to read [NEWS.txt](#) all the way back to your current version.
   - [DataStax Enterprise changes in CHANGES.txt](#).
   - DataStax driver changes.

4. Verify that your current product version is DSE 6.0.0 or later.
$ dse -v

These instructions are valid only for upgrades from DSE 6.0 to DSE 6.7.

5. Upgrade to the latest patch release on your current version. The latest 6.0.x version of DSE is 6.0.8.

   Always upgrade to latest patch release on your current version before you upgrade to a higher version. Fixes included in the latest patch release might help or smooth the upgrade process.

6. To prevent potential problems, 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.

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

   $ java -version

   • Recommended. OpenJDK 8 (1.8.0_151 minimum)

     Note: Recommendation changed due to the end of public updates for Oracle JRE/JDK 8. See Oracle Java SE Support Roadmap.

   • Supported. Oracle Java SE 8 (JRE or JDK) (1.8.0_151 minimum)

     Important: Although Oracle JRE/JDK 8 is supported, DataStax does more extensive testing on OpenJDK 8.

8. Run nodetool repair to ensure that data on each replica is consistent with data on other nodes.


   RHEL platforms:

   $ sudo yum install libaio

   Debian:

   $ sudo apt-get install libaio1

10. Back up the configuration files you use to a folder that is not in the directory where you normally run commands.
The configuration files are overwritten with default values during installation of the new version.

Upgrade steps

To upgrade from DSE 6.0 to DSE 6.7, follow these steps on each node in the recommended order. The upgrade process requires upgrading and restarting one node at a time.

**Note:** These steps are performed in your upgraded version and use DSE 6.7 documentation.

1. To flush the commit log of the old installation:

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

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

   **Important:** This step is mandatory when upgrading between major Cassandra versions that change SSTable formats, rendering commit logs from the previous version incompatible with the new version.

2. Stop the node. See [Stopping a DataStax Enterprise node](#).

   - To stop DataStax Enterprise running as a service:
     ```shell
     $ sudo service dse stop
     ```

   - To stop DataStax Enterprise running as a stand-alone process:
     ```shell
     $ bin/dse cassandra-stop
     ```

3. Use the appropriate method to install the new product version on a supported platform:

   - Package installer using YUM
   - Package installer using APT
   - Binary tarball installer

   **Note:** Install the new product version using the same installation type that is on the system, otherwise problems might result.

4. To configure the new version:

   a. Compare your backup configuration files to the new configuration files. Look for any deprecated, removed, or changed settings.

      - Review changes in cassandra.yaml and dse.yaml.
After the upgrade and before restarting with 6.7.0, remove deprecated settings and use new settings.

cassandra.yaml changes

<table>
<thead>
<tr>
<th>Memtable settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deprecated cassandra.yaml settings</td>
</tr>
<tr>
<td>memtable_heap_space_in_mb</td>
</tr>
<tr>
<td>memtable_offheap_space_in_mb</td>
</tr>
<tr>
<td>Replace with this setting</td>
</tr>
<tr>
<td>memtable_space_in_mb</td>
</tr>
</tbody>
</table>

Governs heap and offheap space allocation to set a threshold for automatic memtable flush. The calculated default is 1/4 of the heap size.

<table>
<thead>
<tr>
<th>Changed setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>memtable_allocation_type: offheap_objects</td>
</tr>
</tbody>
</table>

The default method the database uses to allocate and manage memtable memory is offheap_objects.

<table>
<thead>
<tr>
<th>User-defined functions (UDF) settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deprecated cassandra.yaml settings</td>
</tr>
<tr>
<td>user_defined_function_warn_timeout</td>
</tr>
<tr>
<td>user_defined_function_fail_timeout</td>
</tr>
<tr>
<td>Replace with these settings</td>
</tr>
<tr>
<td>user_defined_function_warn_micros: 500</td>
</tr>
<tr>
<td>user_defined_function_fail_micros: 10000</td>
</tr>
<tr>
<td>user_defined_function_warn_heap_mb: 200</td>
</tr>
<tr>
<td>user_defined_function_fail_heap_mb: 500</td>
</tr>
<tr>
<td>user_function_timeout_policy: die</td>
</tr>
</tbody>
</table>

Settings are in microseconds since Java UDFs run faster. The new timeouts are not equivalent to the deprecated settings.

<table>
<thead>
<tr>
<th>Internode encryption settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deprecated cassandra.yaml setting</td>
</tr>
<tr>
<td>server_encryption_options:</td>
</tr>
<tr>
<td>store_type: JKS</td>
</tr>
</tbody>
</table>
Replace with these settings

```yaml
server_encryption_options:
  keystore_type: JKS
  truststore_type: JKS
```

Valid type options are JKS, JCEKS, PKCS12, or PKCS11.

### Internode encryption settings

Deprecated cassandra.yaml setting

```yaml
server_encryption_options:
  store_type: JKS
```

Replace with these settings

```yaml
server_encryption_options:
  keystore_type: JKS
  truststore_type: JKS
```

Valid type options are JKS, JCEKS, PKCS12, or PKCS11.

### Client-to-node encryption settings

Deprecated cassandra.yaml setting

```yaml
client_encryption_options:
  store_type: JKS
```

Replace with these settings

```yaml
client_encryption_options:
  keystore_type: JKS
  truststore_type: JKS
```

Valid type options are JKS, JCEKS, PKCS12, or PKCS11.

dse.yaml changes

### Spark resource and encryption options

Deprecated dse.yaml setting

```yaml
spark_ui_options:
  server_encryption_options:
    store_type: JKS
```

Replace with these settings
Upgrading DataStax Enterprise

**spark_ui_options:**

```
server_encryption_options:
  keystore_type: JKS
  truststore_type: JKS
```

Valid options are JKS, JCEKS, PKCS12, or PKCS11.

b. Merge the applicable configuration file modifications into the new version.

5. Ensure that keyspace replication factors are correct for your environment:
   - Check the keyspace replication factor for **analytics** keyspaces.
   - Check the keyspace replication factor for **system_auth** and **dse_security** keyspaces.

6. Start the node.
   - **Package installations**
   - **Tarball installations**

7. Verify that the upgraded datacenter names match the datacenter names in the keyspace schema definition:
   ```
   $ nodetool status
   ```

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

   Warnings, errors, and exceptions are frequently found in the logs when starting 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.

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

After the upgrade

After all nodes are upgraded and running on DSE 6.7:

1. If you use the OpsCenter Repair Service, turn on the Repair Service.

2. Remove any previously installed JTS JAR files from the classpaths in your DSE installation. JTS (Java Topology Suite) is distributed with DSE 6.7.

3. Spark Jobserver uses DSE custom version 8.0.4.45. Ensure that applications use the compatible Spark Jobserver API from the DataStax repository.
4. DSE 6.7 introduces, and enables by default, the DSE Metrics Collector, a diagnostics information aggregator used to help facilitate DSE problem resolution. For more information on the DSE Metrics Collector, see DataStax Enterprise Metrics Collector.

Upgrading from DataStax Enterprise 5.1 to 6.7

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

Follow these instructions to upgrade from DataStax Enterprise (DSE) 5.1 to DSE 6.7.

Review the DSE 6.0 and DSE 6.7 release notes for all changes.

Note: The DataStax Installer is not supported for DSE 6.0 and later. To upgrade from DSE 5.1 that was installed with the DataStax Installer, you must first change from a standalone installer installation to a tarball or package installation for the same DSE version. See Upgrading to DSE 6.0 or DSE 6.7 from DataStax Installer installations (page 110).

Always upgrade to latest patch release (page 10) on your current version before you upgrade to a higher version. Fixes included in the latest patch release might help or smooth the upgrade process.

The latest 5.1.x version of DSE is 5.1.15.

Attention: Read and understand these instructions before upgrading. Carefully reviewing the planning (page 9) and upgrading instructions can ensure a smooth upgrade and avoid pitfalls and frustrations.

Important: Support for Thrift-compatible tables (COMPACT STORAGE) is dropped. Before upgrading to DSE 6.7, all tables that have COMPACT STORAGE to CQL table format must be migrated. Use the ALTER TABLE DROP COMPACT STORAGE command to migrate Thrift-compatible tables to CQL table format. This command is available in DSE 5.1.6 or later.

General recommendations

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

Tip: OpsCenter provides a Backup Service that manages enterprise-wide backup and restore operations for DataStax Enterprise clusters. OpsCenter 6.5 and later is recommended.
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 restrictions and limitations during the upgrade process**

- **Do not** enable new features.
- Do not run nodetool repair. If you have the OpsCenter Repair Service configured, **turn off** the Repair Service.
- Ensure OpsCenter compatibility. OpsCenter 6.7 is required for managing DSE 6.7 clusters. See the compatibility table.
- During the upgrade, do not bootstrap or decommission nodes.
- Do not issue these types of CQL queries during a rolling restart: **DDL** and **TRUNCATE**.
- NodeSync waits to start until all nodes are upgraded.
- In DSE 5.1.0-5.1.6, the default number of threads used by performance objects is 1. For DSE 5.1.7 and later, the default number of threads used by performance objects is 4. During upgrade, compatible performance objects continue to work during the upgrade process. Incompatible performance objects that require schema changes will work in legacy mode or will start working after the upgrade is complete. Do not change the configuration of performance objects during upgrade. If performance objects were disabled before the upgrade, do not enable them during upgrade. See DSE Performance Service 6.7 | 5.1 | OpsCenter.
- 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.

**Note:** Nodes on different versions might show a schema disagreement during an upgrade.

Restrictions for DSE Advanced Replication nodes

Upgrades are supported only for DSE Advanced Replication V2.

Restrictions for DSE Analytic (Spark) nodes

- Do not run analytics jobs until all nodes are upgraded.
- All nodes in the cluster must be upgraded to the new version before Spark Worker and Spark Master will start.

DSE Graph nodes restrictions

Graph nodes have the same restrictions as the workload they run on. Do not alter graph schema during upgrades. Workload-specific restrictions apply for analytics and search nodes, such as no OLAP queries during upgrades.

DSE Search upgrade restrictions and limitations

- Do not update schemas.
- Do not reindex DSE Search nodes during upgrade.
Upgrading DataStax Enterprise

- DSE 6.7 uses a different Lucene codec than DSE 5.0. Segments written with this new codec cannot be read by earlier versions of DSE. To downgrade to earlier versions, the entire data directory for the search index in question must be cleared.

Restrictions for nodes using any kind of security

- Do not change security credentials or permissions until the upgrade is complete on all nodes.
- If you are not already using Kerberos, do not set up Kerberos authentication before upgrading. First upgrade the cluster, and then set up Kerberos.

Upgrading drivers and possible impact when driver versions are incompatible

Be sure to check driver compatibility. Depending on the driver version, you might need to recompile your client application code. See DataStax driver changes. 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:

- **Protocol version**: Because some drivers can use different protocol versions, force the protocol version at start up. For example, keep the Java driver at its current protocol version while the driver upgrade is happening. Switch to the Java driver to the new protocol version only after the upgrade is complete on all nodes in the cluster.
- **Initial contact points**: 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 details on protocol version negotiation, see protocol versions with mixed clusters in the Java driver version you’re using, for example, Java driver.

Preparing to upgrade

Follow these steps to prepare each node for upgrading from DSE 5.1 to DSE 6.7.

**Note**: These steps are performed in your current version and use DSE 5.1 documentation.

1. Carefully review Planning your DataStax Enterprise upgrade (page 9).

2. Replace ITriggers and custom interfaces.

Several internal and beta extension points were modified to necessitate core storage engine refactoring. All custom implementations, including the following interfaces, must be replaced with supported implementations when upgrading to DSE 6.7. Because a rewrite of the following interfaces is required for DSE 6.7: (For help contact the DataStax Services team.)

- The org.apache.cassandra.triggers.ITrigger interface was modified from augment to augmentNonBlocking for non-blocking internal architecture. Updated trigger implementations must be provided on upgraded nodes. If unsure, drop all existing triggers before upgrading. To check for existing triggers:
SELECT * FROM system_schema.triggers

- The `org.apache.cassandra.index.Index` interface was modified to comply with the core storage engine changes. Updated implementations are required. If unsure, drop all existing custom secondary indexes before upgrading, except DSE Search indexes, which do not need to be replaced. To check for existing indexes:

SELECT * FROM system_schema.indexes

- The `org.apache.cassandra.cql3.QueryHandler`, `org.apache.cassandra.db.commitlog.CommitLogReadHandler`, and other extension points have been changed. See `QueryHandlers`.

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

   The required space depends on the compaction strategy. See `Disk space`.

4. Familiarize yourself with the changes and features in this release:
   - DataStax Enterprise 6.7 release notes.
   - General upgrading advice for any version. Be sure to read `NEWS.txt` all the way back to your current version.
   - DataStax Enterprise changes in `CHANGES.txt`.
   - DataStax driver changes.

5. Verify that your current product version is DSE 5.1.

   $ dse -v

   These instructions are valid only for upgrades from DSE 5.1 to DSE 6.7.

6. Upgrade to the latest patch release on your current version. The latest 5.1.x version of DSE is 5.1.15.

   Always upgrade to latest patch release on your current version before you upgrade to a higher version. Fixes included in the latest patch release might help or smooth the upgrade process.

7. To prevent potential problems, 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.
8. Back up the configuration files you use to a folder that is not in the directory where you normally run commands.

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

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

```
$ java -version
```

- **Recommended.** OpenJDK 8 (1.8.0_151 minimum)
  
  **Note:** Recommendation changed due to the end of public updates for Oracle JRE/JDK 8. See Oracle Java SE Support Roadmap.

- **Supported.** Oracle Java SE 8 (JRE or JDK) (1.8.0_151 minimum)

  **Important:** Although Oracle JRE/JDK 8 is supported, DataStax does more extensive testing on OpenJDK 8.

10. Run `nodetool repair` to ensure that data on each replica is consistent with data on other nodes.

11. Install the libaio package for optimal performance.

   **RHEL platforms:**

   `$ sudo yum install libaio`

   **Debian:**

   `$ sudo apt-get install libaio1`

12. **DSE Analytics nodes:**

   a. Support for Thrift-compatible tables (COMPACT STORAGE) is dropped. Before upgrading, follow the steps to migrate all tables that have COMPACT STORAGE to CQL table format while DSE 5.x.x is running.

   **Note:** Do not migrate system.* tables, COMPACT STORAGE is removed by DSE internally. Modifying the system keyspace is not supported; modification attempts generate an error.

   For DSE Analytics, drop compact storage from all the tables in the "HiveMetaStore" and PortfolioDemo keyspaces.

   After COMPACT STORAGE is dropped, columns to support migration to CQL-compatible table format are added as described in migrating from compact storage.

   **Attention:** DSE 6.0 will not start if COMPACT STORAGE tables are present. Creating a COMPACT STORAGE table in a mixed-version cluster is not
supported. Driver connections to the latest DSE 5.0.x and DSE 5.1.x run in a "NO_COMPACT" mode that causes compact tables to appear as if the compact flags were already dropped, but only for the current session.

b. If you programmatically set the shuffle parameter, you must change the code for applications that use `conf.set("spark.shuffle.service.port", port)`. Instead, use `dse spark-submit` which automatically sets the correct service port based on the authentication state. See Configuring Spark for more information.

c. If DSEFS is enabled, copy CFS hivemetastore directory to dse:

```bash
$ DSE_HOME/bin/dse hadoop fs -cp cfs://127.0.0.1/user/spark/warehouse/ dsefs://127.0.0.1/user/spark/warehouse/
```

After upgrade is complete migrate Spark SQL tables (if used) to the new Hive metastore format:

```bash
$ dse client-tool spark metastore migrate --from 5.0.0 --to 6.0.0
```

d. Cassandra File System (CFS) is removed. Remove the `cfs` and `cfs_archive` keyspaces before upgrading. See the From CFS to DSEFS blog post and the Copying data from CFS to DSEFS documentation for more information.

e. Make sure any use of the `SPARK_LOCAL_DIRS` and `SPARK_EXECUTOR_DIRS` environment variables match their use as described in Setting environment variables.

f. For applications to use the compatible Spark Jobserver API in the DataStax repository, migrate jobs that extend from SparkHiveJob and SparkSqlJob to SparkSessionJob. See example in the DemoSparkSessionJob in the demos directory.

   **Note:** Spark Jobserver is the DSE custom version 8.0.4.45.

The default location of the `demos` directory depends on the type of installation:

- **Package installations:** `/usr/share/dse/demos`
- **Tarball installations:** `installation_location/demos`

13. DSE Search nodes:

- Ensure all use of HTTP writes are changed to use CQL commands for updates and inserts.
- Edit the search index config and make these changes, as needed. See Search index config for valid options to change query behavior for search indexes.
  
  # Remove the unsupported dataDir option. You can still set the location of search indexes.
  
  # Remove mergePolicy, maxMergeDocs, and mergeFactor. For example:
<mergeFactor>25</mergeFactor>
<maxMergeDocs>...</nmergeFactor>
<mergePolicy>...</nmergeFactor>

Use mergePolicyFactory instead, and add mergeScheduler:

<mergeScheduler
  class="org.apache.lucene.index.ConcurrentMergeScheduler"
  <int name="maxThreadCount">16</int>
  <int name="maxMergeCount">32</int>
</mergeScheduler>

<mergePolicyFactory
  class="org.apache.solr.index.TieredMergePolicyFactory"
  <int name="maxMergeAtOnce">10</int>
  <int name="segmentsPerTier">10</int>
</mergePolicyFactory>

# Remove any instance of ExtractingRequestHandler.
# Remove DSENRTCachingDirectoryFactory. Change:

<directoryFactory name="DirectoryFactory"
class="com.datastax.bdp.search.solr.DSENRTCachingDirectoryFactory"/>

to:

<directoryFactory name="DirectoryFactory"
class="solr.StandardDirectoryFactory"/>

• Ensure that the catalina.properties and context.xml files are present in the Tomcat conf dir. DSE will not start after upgrade if these files are missing.

The default location of the Tomcat conf directory depends on the type of installation:

# Package installations: /etc/dse/tomcat/conf
# Tarball installations: installation_location/resources/tomcat/conf

• If earlier DSE versions use a custom configuration for the Solr UI web.xml, change:

<filter-
class>com.datastax.bdp.search.solr.auth.DseAuthenticationFilter</filter-class>

to

<filter-
class>com.datastax.bdp.cassandra.auth.http.DseAuthenticationFilter</filter-class>

• StallMetrics MBean is removed. Change operators that use the MBean.
14. DSE Graph nodes:

- Ensure that edge label names and property key names use only the supported characters. Edge label names and property key names allow only [a-zA-Z0-9], underscore, hyphen, and period. In earlier versions, edge label names and property key names allowed nearly unrestricted Unicode.
  
  # schema.describe() displays the entire schema, even if it contains illegal names.
  # In-place upgrades allow existing schemas with invalid edge label names and property key names.
  # Schema elements with illegal names cannot be updated or added.

Upgrade steps

To upgrade from DSE 5.1 to DSE 6.7, follow these steps on each node in the recommended order. The upgrade process requires upgrading and restarting one node at a time.

**Note:** These steps are performed in your upgraded version and use DSE 6.7 documentation.

1. To flush the commit log of the old installation:

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

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

   **Important:** This step is mandatory when upgrading between major Cassandra versions that change SSTable formats, rendering commit logs from the previous version incompatible with the new version.

2. Stop the node. See Stopping a DataStax Enterprise node.

   - To stop DataStax Enterprise running as a service:
     ```
     $ sudo service dse stop
     ```

   - To stop DataStax Enterprise running as a stand-alone process:
     ```
     $ bin/dse cassandra-stop
     ```

3. Use the appropriate method to install the new product version on a supported platform:

   - Package installer using YUM
   - Package installer using APT
   - Binary tarball installer
Note: Install the new product version using the same installation type that is on the system, otherwise problems might result.

4. To configure the new version:

a. Compare your backup configuration files to the new configuration files:
   - Look for any deprecated, removed, or changed settings in cassandra.yaml and dse.yaml.

After the upgrade and before restarting with DSE 6.7.0, remove deprecated settings and use new settings.

**cassandra.yaml changes**

<table>
<thead>
<tr>
<th>Memtable settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deprecated cassandra.yaml settings</td>
</tr>
<tr>
<td>memtable_heap_space_in_mb</td>
</tr>
<tr>
<td>memtable_offheap_space_in_mb</td>
</tr>
<tr>
<td>Replace with this setting</td>
</tr>
<tr>
<td>memtable_space_in_mb</td>
</tr>
</tbody>
</table>

Governs heap and offheap space allocation to set a threshold for automatic memtable flush. The calculated default is 1/4 of the heap size.

<table>
<thead>
<tr>
<th>Changed setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>memtable_allocation_type: offheap_objects</td>
</tr>
</tbody>
</table>

The default method the database uses to allocate and manage memtable memory is offheap_objects.

<table>
<thead>
<tr>
<th>User-defined functions (UDF) settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deprecated cassandra.yaml settings</td>
</tr>
<tr>
<td>user_defined_function_warn_timeout</td>
</tr>
<tr>
<td>user_defined_function_fail_timeout</td>
</tr>
<tr>
<td>Replace with these settings</td>
</tr>
</tbody>
</table>
user_defined_function_warn_micros: 500
user_defined_function_fail_micros: 10000
user_defined_function_warn_heap_mb: 200
user_defined_function_fail_heap_mb: 500
user_function_timeout_policy: die

Settings are in microseconds since Java UDFs run faster. The new timeouts are not equivalent to the deprecated settings.

**Internode encryption settings**

Deprecated cassandra.yaml setting

```
server_encryption_options:
  store_type: JKS
```

Replace with these settings

```
server_encryption_options:
  keystore_type: JKS
  truststore_type: JKS
```

Valid type options are JKS, JCEKS, PKCS12, or PKCS11.

**Client-to-node encryption settings**

Deprecated cassandra.yaml setting

```
client_encryption_options:
  store_type: JKS
```

Replace with these settings

```
client_encryption_options:
  keystore_type: JKS
  truststore_type: JKS
```

Valid type options are JKS, JCEKS, PKCS12, or PKCS11.

**Credentials cache**

Deprecated cassandra.yaml settings

```
credentials_validity_in_ms
credentials_update_interval_in_ms
```

Comment out or remove these credentials cache settings, if they exist. Caches are optimized without these settings.

dse.yaml changes
### Spark resource and encryption options

**Deprecated dse.yaml setting**

```
spark_ui_options:
  server_encryption_options:
    store_type: JKS
```

Replace with these settings

```
spark_ui_options:
  server_encryption_options:
    keystore_type: JKS
    truststore_type: JKS
```

Valid options are JKS, JCEKS, PKCS12, or PKCS11.

### DSE Search nodes

**Deprecated dse.yaml settings**

Remove these options:

- `cql_solr_query_executor_threads`
- `enable_back_pressure_adaptive_nrt_commit`
- `max_solr_concurrency_per_core`
- `solr_indexing_error_log_options`

DSE 6.7 will not start with these options present.

---

**b.** Merge the applicable modifications into the new version.

**c.** Ensure that keyspace replication factors are correct for your environment:

- Check the keyspace replication factor for **analytics keyspaces**.
- Check the keyspace replication factor for **system_auth and dse_security keyspaces**.

5. Start the node.

- **Package installations**
- **Tarball installations**

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

   ```
   $ nodetool status
   ```

7. Review the logs for warnings, errors, and exceptions.
Warnings, errors, and exceptions are frequently found in the logs when starting 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.

8. Repeat the upgrade on each node in the cluster following the recommended order.
   
   Upgrading and restarting each node is called a rolling restart.

Recovery after upgrading to DSE 6.7 without dropping compact storage

Support for Thrift-compatible tables (Compact Storage) is dropped. All tables using Compact Storage must be dropped or migrated to CQL table format before upgrading to DSE 6.7. If a cluster has been upgraded to DSE 6.7 and any Compact Storage tables still exist, follow this procedure to recover and proceed with the upgrade:

1. Downgrade any nodes which were already upgraded to DSE 6.7 to the latest version in the DSE 5.0 or 5.1 series:
   - DSE 5.0.x, downgrade to 5.0.15 or later
   - DSE 5.1.x, downgrade to 5.1.11 or later

2. On each node that was attempted to be started on DSE 6.7, start DSE with the -Dcassandra.commitlog.ignorereplayerrors=true option.

3. On one node (any node) in the cluster, DROP COMPACT STORAGE from tables which use it.

4. Restart DSE to continue the upgrade to DSE 6.7.

After the upgrade

After all nodes are upgraded and running on DSE 6.7:

1. If you use the OpsCenter Repair Service, turn on the Repair Service.

2. Remove any previously installed JTS JAR files from the classpaths in your DSE installation. JTS (Java Topology Suite) is distributed with DSE 6.7.

3. After all nodes are on DSE 6.7 and the required schema change occurs, the new authorization with CassandraAuthorizer enables the use of new columns.

4. DSE 6.7 introduces, and enables by default, the DSE Metrics Collector, a diagnostics information aggregator used to help facilitate DSE problem resolution. For more information on the DSE Metrics Collector, see DataStax Enterprise Metrics Collector.

5. DSE Search only:
• The appender SolrValidationAppender and the logger SolrValidationErrorLogger are no longer used and may safely be removed from logback.xml.

• In contrast to earlier versions, DataStax recommends accepting the new default value of 1024 for `back_pressure_threshold_per_core` in dse.yaml. See Configuring and tuning indexing performance.

• Slow startup on nodes with large encrypted indexes is resolved. However, action is required to realize the performance gains. You must do a full reindex of all encrypted search indexes on each node in your cluster. Plan sufficient time after the upgrade is complete to reindex with deleteAll=true in a rolling fashion. For example:

```
$ dsetool reload_core keyspace_name.table_name distributed=false reindex=true deleteAll=true
```

6. DSE Analytics only:

• Check the replication factor for the dse_analytics keyspace, a new keyspace stores all DSE Analytics internal system data. DataStax recommends setting the replication strategy to NetworkTopologyStrategy (NTS) with a replication factor of at least 3 in each of DSE Analytics datacenters. If a datacenter has more nodes, a larger replication factor should be considered.

• Spark Jobserver uses DSE custom version 0.8.0.45. Applications must use the compatible Spark Jobserver API from the DataStax repository.

7. DSE 6.7 introduces, and enables by default, the DSE Metrics Collector, a diagnostics information aggregator used to help facilitate DSE problem resolution. For more information on the DSE Metrics Collector, see DataStax Enterprise Metrics Collector.

Warning messages during and after upgrade

Error messages provide information to help identify problems. You can ignore some log messages that occur during and after an upgrade.

• Some gremlin_server properties in earlier versions of DSE are no longer required in DSE 6.7. If properties exist in the dse.yaml file after upgrading to DSE 6.7, logs display warnings similar to:

```
WARN  [main] 2017-08-31 12:25:30,523 GREMLIN
DseWebSocketChannelizer.java:149 - Configuration for the
org.apache.tinkerpop.gremlin.driver.ser.GraphSONMessageSerializerGremlinV1d0
serializer in dse.yaml overrides the DSE default - typically it is
best to allow DSE to configure these.
```

You can ignore these warnings or modify dse.yaml so that only the required gremlin server properties are present.

Upgrading from DataStax Enterprise 5.0 to 6.7

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

Follow these instructions to upgrade from DSE 5.0 to 6.7. If you have an earlier version of DSE, upgrade to the latest version of 5.0 (page 133) before continuing.

Review the DSE 5.1, DSE 6.0, and DSE 6.7 release notes for all changes.

**Note:** The DataStax Installer is not supported for DSE 6.0 and later. To upgrade from DSE 5.0 that was installed with the DataStax Installer, you must first change from a standalone installer installation to a tarball or package installation for the same DSE version. See Upgrading to DSE 6.0 or DSE 6.7 from DataStax Installer installations (page 110).

Always upgrade to latest patch release (page 10) on your current version before you upgrade to a higher version. Fixes included in the latest patch release might help or smooth the upgrade process.

The latest version of DSE 5.0 is 5.0.15.

**Attention:** Read and understand these instructions before upgrading. Carefully reviewing the planning (page 9) and upgrading instructions can ensure a smooth upgrade and avoid pitfalls and frustrations.

**Important:** Support for Thrift-compatible tables (COMPACT STORAGE) is dropped. Before upgrading to DSE 6.7, all tables that have COMPACT STORAGE to CQL table format must be migrated. Use the `ALTER TABLE DROP COMPACT STORAGE` command to migrate Thrift-compatible tables to CQL table format. This command is available in DSE 5.0.12 or later.

Apache Cassandra™ version change

Upgrading from DataStax Enterprise 5.0 to 6.7 includes a major Cassandra version change.

- DataStax Enterprise 6.7 is compatible with Cassandra 3.11.
- DataStax Enterprise 6.0 is compatible with Cassandra 3.11.
- DataStax Enterprise 5.1 uses Cassandra 3.11.
- DataStax Enterprise 5.0 uses Cassandra 3.0.
- DataStax Enterprise 4.7 to 4.8 use Cassandra 2.1.
- DataStax Enterprise 4.0 to 4.6 use Cassandra 2.0.

Be sure to follow the recommendations for upgrading the SSTables.

General recommendations

DataStax recommends backing up your data prior to any version upgrade, including logs and custom configurations. 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. OpsCenter 6.5 and later is recommended.

**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 restrictions and limitations during the upgrade process**

- Do not enable new features.
- Do not run nodetool repair. If you have the OpsCenter Repair Service configured, **turn off** the Repair Service.
- Ensure OpsCenter compatibility. OpsCenter 6.7 is required for managing DSE 6.7 clusters. See the compatibility table.
- During the upgrade, do not bootstrap or decommission nodes.
- Do not issue these types of CQL queries during a rolling restart: DDL and TRUNCATE.
- NodeSync waits to start until all nodes are upgraded.
- In DSE 5.0, the default number of threads used by performance objects is 1. For DSE 6.7, the default number of threads used by performance objects is 4. During upgrade, compatible performance objects continue to work during the upgrade process. Incompatible performance objects that require schema changes will work in legacy mode or will start working after the upgrade is complete. Do not change the configuration of performance objects during upgrade. If performance objects were disabled before the upgrade, do not enable them during upgrade. See DSE Performance Service 6.7 | 5.0 | OpsCenter.
- 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.

**Note:** Nodes on different versions might show a schema disagreement during an upgrade.

**Restrictions for DSE Advanced Replication nodes**

Upgrades are supported only for DSE Advanced Replication V2.

**DSE Search upgrade restrictions and limitations**

- Do not update schemas.
- Do not reindex DSE Search nodes during upgrade.
- DSE 6.7 uses a different Lucene codec than DSE 5.0. Segments written with this new codec cannot be read by earlier versions of DSE. To downgrade to earlier versions, the entire data directory for the search index in question must be cleared.
Upgrading DataStax Enterprise

- DSE Search in DataStax Enterprise 6.7 uses Apache Solr 6.0. This significant change requires advanced planning and specific actions before and after the upgrade.

**Important:** Before you upgrade DSE Search or SearchAnalytics workloads, you must follow the specific tasks in the advanced preparation for upgrading DSE Search and SearchAnalytics nodes section.

**Restrictions for nodes using any kind of security**
- Do not change security credentials or permissions until the upgrade is complete on all nodes.
- If you are not already using Kerberos, do not set up Kerberos authentication before upgrading. First upgrade the cluster, and then set up Kerberos.

**Upgrading drivers and possible impact when driver versions are incompatible**
Be sure to check [driver compatibility](#). Depending on the driver version, you might need to recompile your client application code. See DataStax driver changes. 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:

- **Protocol version:** Because some drivers can use different protocol versions, force the protocol version at start up. For example, keep the Java driver at its current protocol version while the driver upgrade is happening. Switch to the Java driver to the new protocol version only after the upgrade is complete on all nodes in the cluster.
- **Initial contact points:** 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 details on protocol version negotiation, see [protocol versions with mixed clusters](#) in the Java driver version you’re using, for example, [Java driver](#).

**Advanced preparation for upgrading DSE Search and SearchAnalytics nodes**

Before starting the Preparing to upgrade (page 65) steps, complete all the advanced preparation steps on DSE Search and SearchAnalytics nodes while DSE 5.0 is still running.

Plan sufficient time to implement and test the required changes before the upgrade:

- Schema changes require a full reindex.
- Configuration changes require reloading the core.

1. Change HTTP queries to CQL:
   - Delete-by-id is removed, use CQL DELETE by primary key instead.
   - Delete-by-query no longer supports wildcards, use CQL TRUNCATE instead.
2. If any Solr core was created on DSE 4.6 or earlier and never reindexed after being upgraded to DSE 4.7 or later, you must reindex on DSE 5.0 before upgrading to DSE 6.7.

3. Ensure that the `shard_transport_options` in `dse.yaml` are set only for `netty_client_request_timeout`:

   ```yaml
   shard_transport_options:
     netty_client_request_timeout: 60000
   ```

   In DSE 6.7, the shard transport option supports only `netty_client_request_timeout`. Remove any other `shard_transport_options`.

4. If you are using Apache Solr SolrJ, the minimum required version is 6.0.0.

5. For `SpatialRecursivePrefixTreeFieldType` (RPT) in search schemas, you must adjust your queries for these changes:
   - `IsDisjointTo` is no longer supported in queries on `SpatialRecursivePrefixTreeFieldType`. Replace `IsDisjointTo` with a NOT Intersects query. For example:
     ```sql
     foo:0,0 TO 1000,1000 AND ~"Intersects(POLYGON((338 211, 338 305, 404 305, 404 211, 338 211)))"
     ```
   - The `ENVELOPE` syntax is now required for WKT-style queries against `SpatialRecursivePrefixTreeFieldType` fields. You must specify `ENVELOPE(10, 15, 15, 10)`, where queries on earlier releases could specify `10 10 15 15`.

   See Spatial Search for details on using distanceUnits in spatial queries.

6. Edit the `solrconfig.xml` file and make these changes, as needed:
   - Remove these unsupported Solr requestHandlers:
     ```xml
     <requestHandler name="/dataimport" class="solr.DataImportHandler"/>
     <requestHandler name="/update" class="solr.XmlUpdateRequestHandler"/>
     <requestHandler name="/update" class="solr.BinaryUpdateRequestHandler"/>
     <requestHandler name="/update" class="solr.CSVRequestHandler"/>
     <requestHandler name="/update" class="solr.JsonUpdateRequestHandler"/>
     <requestHandler name="/update" class="solr.DataImportHandler"/>
     ```
   - Change the directoryFactory from:
Upgrading DataStax Enterprise

```xml
<directoryFactory name="DirectoryFactory"
    class="$solr.directoryFactory:solr.StandardDirectoryFactory"/>
```
to

```xml
<directoryFactory name="DirectoryFactory"
    class="solr.StandardDirectoryFactory"/>
```

- `<unlockOnStartup>` is unsupported as a result of LUCENE-6508 and SOLR-7942.
- Change the updateLog from:

```xml
<updateLog class="solr.FSUpdateLog" force="false">
```
to

```xml
<updateLog force="false">
```

7. Upgrading DSE search nodes requires replacing unsupported Solr types with supported types.

   **Note:** Special handling is also required for BCDStrField, addressed in step 9 (page 62).

Sorting limitations apply to mixed version clusters. Some of the removed Solr types, due to the way they marshal sort values during distributed queries (combined with the way the suggested new types unmarshal sort values), cannot be sorted on during rolling upgrades when some nodes use an unsupported type and other nodes use the suggested new type. The following type transitions are problematic:

<table>
<thead>
<tr>
<th>Removed Solr field types</th>
<th>Supported Solr field types</th>
</tr>
</thead>
<tbody>
<tr>
<td>ByteField</td>
<td>TrieIntField</td>
</tr>
<tr>
<td>DateField</td>
<td>TrieDateField</td>
</tr>
<tr>
<td>BCDIntField</td>
<td>TrieIntField</td>
</tr>
<tr>
<td>BCDLongField</td>
<td>TrieLongField</td>
</tr>
</tbody>
</table>

Two options are available:

a. Avoid sorting on removed Solr field types until the upgrade is complete for all nodes in the datacenter being queried.

   **Tip:** When using two search datacenters, isolate queries to a single datacenter and then change the schema and reindex the other datacenter. Then isolate queries to the newly reindexed datacenter while you change the schema and upgrade the first datacenter.

b. If you are using BCDIntField or BCDLongField, update the schema to replace BCDIntField and BCDLongField with types that are sort-compatible with the supported Solr types TrieIntField and TrieLongField:
### Removed Solr field types

<table>
<thead>
<tr>
<th>Removed Solr field types</th>
<th>Interim sort-compatible supported Solr field types</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCDIntField</td>
<td>SortableIntField</td>
</tr>
<tr>
<td>BCDLongField</td>
<td>SortableLongField</td>
</tr>
</tbody>
</table>

Change the schema in a distributed fashion, and do not reindex. After the schema is updated on all nodes, then go on to *9 ([page 95](#)).

#### 8. Update the schema and configuration for the Solr field types that are removed from Solr 5.5 and later.

- Update the schema to replace unsupported Solr field types with supported Solr field types:

<table>
<thead>
<tr>
<th>Removed Solr field types</th>
<th>Supported Solr field types</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCDIntField</td>
<td>TrieIntField</td>
</tr>
<tr>
<td>BCDLongField</td>
<td>TrieLongField</td>
</tr>
<tr>
<td>ByteField</td>
<td>TrieIntField</td>
</tr>
<tr>
<td>DateField</td>
<td>TrieDateField</td>
</tr>
<tr>
<td>DoubleField</td>
<td>TrieDoubleField</td>
</tr>
<tr>
<td>FloatField</td>
<td>TrieFloatField</td>
</tr>
<tr>
<td>IntField</td>
<td>TrieIntField</td>
</tr>
<tr>
<td>LongField</td>
<td>TrieLongField</td>
</tr>
<tr>
<td>ShortField</td>
<td>TrieIntField</td>
</tr>
<tr>
<td>SortableDoubleField</td>
<td>TrieDoubleField</td>
</tr>
<tr>
<td>SortableFloatField</td>
<td>TrieFloatField</td>
</tr>
<tr>
<td>SortableIntField</td>
<td>TrieIntField</td>
</tr>
<tr>
<td>SortableLongField</td>
<td>TrieLongField</td>
</tr>
<tr>
<td>BCDIntField</td>
<td>TrieIntField</td>
</tr>
<tr>
<td>BCDLongField</td>
<td>TrieLongField</td>
</tr>
<tr>
<td>BCDStrField (see 10 (<a href="#">page 96</a>) if used)</td>
<td>TrieIntField</td>
</tr>
</tbody>
</table>

- **If you are using type mapping version 0, or you do not specify a type mapper,** verify or update the `solrconfig.xml` to use `dseTypeMappingVersion 1`:

  ```xml
  <dseTypeMappingVersion>1</dseTypeMappingVersion>
  ```

  If the Solr core is backed by a CQL table and the type mapping is unspecified, use type mapping version 2.

- Reload the core:
If you were using the unsupported data types, do a full reindex node-by-node:

```
$ dsetool reload_core keyspace_name.table_name schema=filepath
    solrconfig=filepath

reindex=true deleteAll=true distributed=false
```

**Note:** In DSE 5.1 and later, auto generated schemas use data type mapper 2.

9. **If using BCDStrField:** In DSE 5.0 and earlier, DSE mapped Cassandra text columns to BCDStrField. The deprecated BCDStrField is removed.

The recommended strategy is to upgrade the data type to TrieIntField. However, DSE cannot map text directly to TrieIntField. If you are using BCDStrField, you must complete one of these options before the upgrade:

- If BCDStrField is no longer used, remove the BCDStrField field from the Solr schema. Reindexing is not required.
- If you want to index the field as a TrieIntField, and a full reindex is acceptable, change the underlying database column to use the type int.
- If you want to keep the database column as text and you still want to do simple matching queries on the indexed field, switch from BCDStrField to StrField in the schema. Indexing should not be required, but the field will no longer be appropriate for numeric range queries or sorting because StrField uses a lexicographic order, not a numeric one.
- **Not recommended:** If you want to keep the database column as text and still want to perform numeric range queries and sorts on the former BCDStrField, but would rather change their application than perform a full reindex:
  
  ```
  # Change the field to StrField in the Solr schema with indexed=false.
  # Add a new copy field with the type TrieIntField that has its values supplied by the original BCDStrField.
  ```

  This solution still requires reindex to work, because the copy field target must be populated. This non-recommended option is supplied only to support a sub-optimal data model; for example, a text column with values that would fit only into an int.

After you make these schema changes, do a rolling, node-by-node `reload_core` with reindex=true, distributed=false, and deleteAll=true.

**Note:** If you have two datacenters and upgrade them one at a time, reload the core with distributed=true and deleteAll=true.

10. Tune the schema before you upgrade. After the upgrade, all field definitions in the schema are validated and must be DSE Search compatible, even if the fields are not indexed, have docValues applied, or used for copy-field source. The default behavior of automatic resource generation includes all columns. To improve performance, take action
to prevent the fields from being loaded from the database. Include only the required fields in the schema by removing or commenting out unused fields in the schema.

Advanced preparation for upgrading DSE Graph nodes with search indexes

These steps apply to graph nodes that have search indexes.

Before starting the Preparing to upgrade (page 65) steps, complete these advanced preparation steps while DSE 5.0 is still running.

Note: Upgrading DSE Graph nodes with search indexes requires these edits to the solrconfig file. Configuration changes require reloading the core. Plan sufficient time to implement and test changes that are required before the upgrade.

Edit the solrconfig.xml file and make these changes, as needed:

- Remove these requestHandlers: XmlUpdateRequestHandler, BinaryUpdateRequestHandler, CSVRequestHandler, JsonUpdateRequestHandler, and DataImportHandler. Solr deprecated and then removed these requestHandlers.
  
  For example:

  ```xml
  <requestHandler name="/dataimport" class="solr.DataImportHandler"/>
  
  or
  
  <requestHandler name="/update" class="solr.XmlUpdateRequestHandler"/>
  
  <unlockOnStartup> is now unsupported as a result of LUCENE-6508 and SOLR-7942.

  - Reload the core so that this configuration change is respected:
    
    ```bash
    $ dsetool reload_core keyspace_name.table_name reindex=false
    ```

Advanced preparation for upgrading DSE Analytics nodes

Upgrades from DSE 5.0 to DSE 6.7 include a major upgrade to Spark 2.2 with a tighter integration between DSE and Spark. For information on Spark 2.2, see the Spark documentation. Spark 2.2 uses Scala 2.11.

A new Spark resource manager uses the DataStax Java Driver and native CQL protocol for managing communication between DSE Spark nodes. DSE 5.0 used Spark RPC. This resource manage change impacts Spark applications that ran on DSE 5.0.

1. Spark 2.2 uses Scala 2.11. You must recompile all DSE 5.0 Scala Spark applications against Scala 2.11 and use only Scala 2.11 third-party libraries.

   Changing the dse-spark-dependencies in your build files is not sufficient to change the compilation target. See the example projects for how to set up your build files.
2. Spark applications should use `dse://` URLs instead of `spark://spark_master_IP:Spark_RPC_port_number` URLs, as described in Specifying Spark URLs.

You no longer need to specify the Spark master IP address or hostname when using `dse://` URLs. Connecting to any Spark node will redirect the request to the master node.

3. If you have existing Spark application code that uses `spark://Spark master IP:Spark_RPC_port` to connect, it will no longer work.

For example, the following code worked in DSE 5.0 but will not work in DSE 5.1 or later.

```scala
val conf = new SparkConf(true)
    .setMaster("spark://192.168.123.10:7077")
    .setAppName("cassandra-demo")
    .set("cassandra.connection.host", "192.168.123.10") // initial contact
    .set("cassandra.username", "cassandra")
    .set("cassandra.password", "cassandra")
val sc = new SparkContext(conf)
```

To connect to DSE 6.7, you no longer need to call `setMaster`:

```scala
val conf = new SparkConf(true)
    .setAppName("cassandra-demo")
    .set("cassandra.connection.host", "192.168.123.10") // initial contact
    .set("cassandra.username", "cassandra")
    .set("cassandra.password", "cassandra")
val sc = new SparkContext(conf)
```

To specify the master using `setMaster`, use the `dse://` URL format.

4. You can restrict Spark jobs to specific database roles. See Managing Spark application permissions.

5. You can set the Spark executor process owners, as described in Running Spark processes as separate users.

6. The user submitting the Spark application no longer has to be the same database role. See Specifying Spark URLs to change the master connection submission to use a different user or cluster than the database connection.

Back up DSEFS data

These steps only apply to nodes that use DSEFS. Before starting the Preparing to upgrade (page 65) steps, complete these advanced preparation steps.
The DSEFS schema used by the database is improved, but the old schema is still supported and will not be modified during the upgrade. To use the new DSEFS schema with existing DSEFS data, the current DSEFS data to local storage using `dse hadoop fs -cp` command:

```
$ dse hadoop fs -cp /* /local_backup_location
```

Preparing to upgrade

Follow these steps to prepare each node for upgrading from DSE 5.0 to DSE 6.7.

**Note:** These steps are performed in your current version and use DSE 5.0 documentation.

1. Upgrade to the latest patch release on your current version. Fixes included in the latest patch release can simplify the upgrade process.

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

   The required space depends on the compaction strategy. See [Disk space](#).

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

   - [DataStax Enterprise 6.7 release notes](#).
   - General upgrading advice for any version. Be sure to read [NEWS.txt](#) all the way back to your current version.
   - DataStax Enterprise changes in [CHANGES.txt](#).
   - DataStax driver changes.

4. Replace ITriggers and custom interfaces.

   Several internal and beta extension points were modified to necessitate core storage engine refactoring. All custom implementations, including the following interfaces, must be replaced with supported implementations when upgrading to DSE 6.7. Because a rewrite of the following interfaces is required for DSE 6.7: (For help contact the [DataStax Services team](#).)

   - The [org.apache.cassandra.triggers.ITrigger](#) interface was modified from `augment` to `augmentNonBlocking` for non-blocking internal architecture. Updated trigger implementations must be provided on upgraded nodes. If unsure, drop all existing triggers before upgrading. To check for existing triggers:

     ```
     SELECT * FROM system_schema.triggers
     ```

   - The [org.apache.cassandra.index.Index](#) interface was modified to comply with the core storage engine changes. Updated implementations are required. If unsure, drop all existing custom secondary indexes before upgrading, except DSE Search indexes, which do not need to be replaced. To check for existing indexes:
SELECT * FROM system_schema.indexes

- The org.apache.cassandra.cql3.QueryHandler,
  org.apache.cassandra.db.commitlog.CommitLogReadHandler, and other
  extension points have been changed. See QueryHandlers.

5. Support for Thrift-compatible tables (COMPACT STORAGE) is dropped. Before
upgrading, follow the steps to migrate all tables that have COMPACT STORAGE to CQL
table format while DSE 5.0 is running.

  Note: Do not migrate system.* tables, COMPACT STORAGE is removed by DSE
  internally.

For DSE Analytics, drop compact storage from all the tables in the "HiveMetaStore" and
PortfolioDemo keyspaces.

After COMPACT STORAGE is dropped, columns to support migration to CQL-compatible
table format are added as described in migrating from compact storage.

  Attention: DSE 6.7 will not start if COMPACT STORAGE tables are present.
  Creating a COMPACT STORAGE table in a mixed-version cluster is not supported.

6. If audit logging is configured to use CassandraAuditWriter, run these commands as
superuser on DSE 5.0 nodes, and then ensure that the entire cluster has schema
agreement:

```bash
$ ALTER TABLE dse_audit.audit_log ADD authenticated text;
ALTER TABLE dse_audit.audit_log ADD consistency text;
```

7. Upgrade the SSTables on each node to ensure that all SSTables are on the current
version.

```bash
$ nodetool upgradesstables
```

This step is required for DataStax Enterprise upgrades that include a major Cassandra
version changes.

  Warning: Failure to upgrade SSTables when required results in a significant
  performance impact and increased disk usage.

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

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

```bash
$ java -version
```

  - Recommended. OpenJDK 8 (1.8.0_151 minimum)
Note: Recommendation changed due to the end of public updates for Oracle JRE/JDK 8. See Oracle Java SE Support Roadmap.

- Supported. Oracle Java SE 8 (JRE or JDK) (1.8.0_151 minimum)

Important: Although Oracle JRE/JDK 8 is supported, DataStax does more extensive testing on OpenJDK 8.

9. Run `nodetool repair` to ensure that data on each replica is consistent with data on other nodes.

10. Install the libaio package for optimal performance.

   RHEL platforms:
   
   ```bash
   $ sudo yum install libaio
   ```

   Debian:
   
   ```bash
   $ sudo apt-get install libaio1
   ```

11. DSE Analytics nodes:

   - If you programmatically set the shuffle parameter, you must change the code for applications that use `conf.set("spark.shuffle.service.port", port)`. Instead, use `dse spark-submit` which automatically sets the correct service port based on the authentication state. See Configuring Spark for more information.

   - If DSEFS is enabled, copy CFS hivemetastore directory to dse:

     ```bash
     $ DSE_HOME/bin/dse hadoop fs -cp cfs://127.0.0.1/user/spark/warehouse/ dsefs://127.0.0.1/user/spark/warehouse/
     ```

     After upgrade is complete migrate Spark SQL tables (if used) to the new Hive metastore format:

     ```bash
     $ dse client-tool spark metastore migrate --from 5.0.0 --to 6.0.0
     ```

   - Cassandra File System (CFS) is removed. Remove the `cfs` and `cfs_archive` keyspaces before upgrading. See the From CFS to DSEFS blog post and the Copying data from CFS to DSEFS documentation for more information.

12. DSE Search nodes:

   - DSE Search in DSE 6.7 uses Apache Solr™ 6.0. Complete all of the steps in Advanced preparation for upgrading DSE Search and SearchAnalytics nodes (page 58).

   - Ensure all use of HTTP writes are changed to use CQL commands for updates and inserts.
- Edit the search index config and make these changes, as needed. See Search index config for valid options to change query behavior for search indexes.

  # Remove the unsupported dataDir option. You can still set the location of search indexes.

  # Remove mergePolicy, maxMergeDocs, and mergeFactor. For example:

  ```
  <mergeFactor>25</mergeFactor>
  <maxMergeDocs>...</
  <mergePolicy>...</
  ```

  Use mergePolicyFactory instead, and add mergeScheduler:

  ```
  <mergeScheduler
    class="org.apache.lucene.index.ConcurrentMergeScheduler">
    <int name="maxThreadCount">16</int>
    <int name="maxMergeCount">32</int>
  </mergeScheduler>
  ...
  <mergePolicyFactory
    class="org.apache.solr.index.TieredMergePolicyFactory">
    <int name="maxMergeAtOnce">10</int>
    <int name="segmentsPerTier">10</int>
  </mergePolicyFactory>
  ```

  # Remove any instance of ExtractingRequestHandler.

  # Remove DSENRTCachingDirectoryFactory. Change:

  ```
  <directoryFactory name="DirectoryFactory"
    class="com.datastax.bdp.search.solr.DSENRTCachingDirectoryFactory"/>
  ```

  to:

  ```
  <directoryFactory name="DirectoryFactory"
    class="solr.StandardDirectoryFactory"/>
  ```

- Ensure that the catalina.properties and context.xml files are present in the Tomcat conf dir. DSE will not start after upgrade if these files are missing.

  The default location of the Tomcat conf directory depends on the type of installation:

  # Package installations: /etc/dse/tomcat/conf

  # Tarball installations: installation_location/resources/tomcat/conf

- If earlier DSE versions use a custom configuration for the Solr UI web.xml, change:

  ```
  <filter-class>com.datastax.bdp.search.solr.auth.DseAuthenticationFilter</filter-class>
  ```

  to
• StallMetrics MBean is removed. Change operators that use the MBean.

13. DSE Graph nodes:

• If your graph nodes have search indexes that you added with gremlin, complete the steps in Advanced preparation for upgrading DSE Graph nodes with search indexes (page 63).

• Ensure that edge label names and property key names use only the supported characters. Edge label names and property key names allow only [a-zA-Z0-9], underscore, hyphen, and period. In earlier versions, edge label names and property key names allowed nearly unrestricted Unicode.

  # schema.describe() displays the entire schema, even if it contains illegal names.

  # In-place upgrades allow existing schemas with invalid edge label names and property key names.

  # Schema elements with illegal names cannot be updated or added.

14. Back up the configuration files you use to a folder that is not in the directory where you normally run commands.

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

Upgrade steps

To upgrade from DSE 5.0 to DSE 6.7, follow these steps on each node in the recommended order. The upgrade process requires upgrading and restarting one node at a time.

**Note:** These steps are performed in your upgraded version and use DSE 6.7 documentation.

1. **DSE Analytics nodes:** Kill all Spark worker processes.

2. To flush the commit log of the old installation:

   $ nodetool -h hostname drain

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

   **Important:** This step is mandatory when upgrading between major Cassandra versions that change SSTable formats, rendering commit logs from the previous version incompatible with the new version.
3. Stop the node.

4. Use the appropriate method to install the new product version on a supported platform:
   - Package installer using YUM
   - Package installer using APT
   - Binary tarball installer

   **Note:** Install the new product version using the same installation type that is on the system, otherwise problems might result.

5. To configure the new product version:
   a. Compare your backup configuration files to the new configuration files:
      - Review changes in cassandra.yaml and dse.yaml.

      After the upgrade and before restarting with 6.7.0, remove deprecated settings and use the new settings.

      **cassandra.yaml changes**

<table>
<thead>
<tr>
<th>Memtable settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deprecated cassandra.yaml settings</td>
</tr>
<tr>
<td>memtable_heap_space_in_mb</td>
</tr>
<tr>
<td>memtable_offheap_space_in_mb</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Replace with this setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>memtable_space_in_mb</td>
</tr>
</tbody>
</table>

Governs heap and offheap space allocation to set a threshold for automatic memtable flush. The calculated default is 1/4 of the heap size.

<table>
<thead>
<tr>
<th>Changed setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>memtable_allocation_type: offheap_objects</td>
</tr>
</tbody>
</table>

The default method the database uses to allocate and manage memtable memory is offheap_objects.

<table>
<thead>
<tr>
<th>User-defined functions (UDF) settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deprecated cassandra.yaml settings</td>
</tr>
<tr>
<td>user_defined_function_warn_timeout</td>
</tr>
<tr>
<td>user_defined_function_fail_timeout</td>
</tr>
</tbody>
</table>
Replace with these settings

- user_defined_function_warn_micros: 500
- user_defined_function_fail_micros: 10000
- user_defined_function_warn_heap_mb: 200
- user_defined_function_fail_heap_mb: 500
- user_function_timeout_policy: die

Settings are in microseconds since Java UDFs run faster. The new timeouts are not equivalent to the deprecated settings.

**Internode encryption settings**

Deprecated cassandra.yaml setting

```
server_encryption_options:
  store_type: JKS
```

Replace with these settings

```
server_encryption_options:
  keystore_type: JKS
  truststore_type: JKS
```

Valid type options are JKS, JCEKS, PKCS12, or PKCS11.

**Client-to-node encryption settings**

Deprecated cassandra.yaml setting

```
client_encryption_options:
  store_type: JKS
```

Replace with these settings

```
client_encryption_options:
  keystore_type: JKS
  truststore_type: JKS
```

Valid type options are JKS, JCEKS, PKCS12, or PKCS11.

**dse.yaml changes**

- Look for any deprecated, removed, or changed settings.

**Shard transport**

Deprecated dse.yaml settings
shard_transport_options:
  type: netty
  netty_server_port: 8984
  netty_server_acceptor_threads:
  netty_server_worker_threads:
  netty_client_worker_threads:
  netty_client_max_connections:
  netty_client_request_timeout:

The http transport type is removed.

shard_transport_options:
  type: http
  http_shard_client_conn_timeout: 0
  http_shard_client_socket_timeout: 0

New dse.yaml settings

shard_transport_options:
  netty_client_request_timeout: 60000

The shard_transport_options supports only netty_client_request_timeout. Remove any other options under shard_transport_options.

DSE Search nodes

Deprecated dse.yaml settings

Remove these options:

cqi_soir_query_executor_threads
enable_back_pressure_adaptive_nrt_commit
max_soir_concurrency_per_core
soir_indexing_error_log_options

DSE 6.7 will not start with these options present.

DSE Analytics nodes: DSEFS settings

Changed dse.yaml settings

Although DSEFS is enabled by default, the dsefs.enabled setting is commented out. To enable DSEFS, uncomment all dsefs_options settings.

- Ensure that keyspace replication factors are correct for your environment:
  
  # Check the keyspace replication factor for analytics keyspaces.
  # Check the keyspace replication factor for system_auth and dse_security keyspaces.

- The upgrade installs a new server.xml for Tomcat 8. If your existing server.xml has custom connectors, migrate those connectors to the new server.xml before starting the upgraded nodes.
• Be sure you are familiar (page 65) with the Apache Cassandra and DataStax Enterprise changes and features in the new release.

6. DSE Analytics nodes: If your DSE 5.0 clusters had any datacenters running in Analytics Hadoop mode and if the DseSimpleSnitch was used, you must use one of these options for starting nodes in your cluster. Select the option that works best for your environment:
   • For nodes in the datacenters running in Analytics Hadoop mode, start those nodes in Spark mode.
   • Add the special start-up parameter -Dcassandra.ignore_dc=true for each node, then start in cassandra mode. This flag is required only once after upgrading. Subsequent restarts do not use this flag. You can leave the flag in the configuration file or remove it after the first restart of each node.

7. Start the node.
   • Package installations
   • Tarball installations

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

   $ nodetool status

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

   Warnings, errors, and exceptions are frequently found in the logs when starting 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.

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

    Upgrading and restarting each node is called a rolling restart.

11. When the upgrade includes a major Cassandra version, you must upgrade the SSTables. DataStax recommends upgrading the SSTables on one node at a time or when using racks, one rack at a time.

    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.

    $ nodetool upgradesstables
If the SSTables are already on the current version, the command returns immediately and no action is taken. See SSTable compatibility and upgrade version.

Use the --jobs option to set the number of SSTables that upgrade simultaneously. The default setting is 2, which minimizes impact on the cluster. Set to 0 to use all available compaction threads. DataStax recommends running the upgradesstables command on one node at a time or when using racks, one rack at a time.

**Note:** You can run the upgradesstables command before all the nodes are upgraded as long as you run this command on only one node at a time or when using racks, one rack at a time. Running upgradesstables on too many nodes will degrade performance.

Recovery after upgrading to DSE 6.7 without dropping compact storage

Support for Thrift-compatible tables (Compact Storage) is dropped. All tables using Compact Storage must be dropped or migrated to CQL table format before upgrading to DSE 6.7. If a cluster has been upgraded to DSE 6.7 and any Compact Storage tables still exist, follow this procedure to recover and proceed with the upgrade:

1. Downgrade any nodes which were already upgraded to DSE 6.7 to the latest version in the DSE 5.0 or 5.1 series:
   - DSE 5.0.x, downgrade to 5.0.15 or later
   - DSE 5.1.x, downgrade to 5.1.15 or later

2. On each node that was attempted to be started on DSE 6.7, start DSE with the following option:

   ```
   -Dcassandra.commitlog.ignorereplayerrors=true
   ```

3. On one node (any node) in the cluster, DROP COMPACT STORAGE from tables which use it.

4. Restart DSE to continue the upgrade to DSE 6.7.

After the upgrade

After all nodes are upgraded and running on DSE 6.7, complete these steps:

1. If you use the OpsCenter Repair Service, turn on the Repair Service.

2. Remove any previously installed JTS JAR files from the classpaths in your DSE installation. JTS (Java Topology Suite) is distributed with DSE 6.7.

3. After all nodes are on DSE 6.7 and the required schema change occurs, the new audit logging feature (CassandraAuditWriter) enables the use of new columns.
4. Drop the following legacy tables, if they exist: system_auth.users, system_auth.credentials, and system_auth.permissions.

As described in General upgrade advice, authentication and authorization subsystems now support role-based access control (RBAC).

5. Review your security configuration. To use security, enable and configure DSE Unified Authentication.

In cassandra.yaml, the default authenticator is DseAuthenticator and the default authorizer is DseAuthorizer. Other authenticators and authorizers are no longer supported. Security is disabled in dse.yaml by default.

6. TimeWindowCompactionStrategy (TWCS) is set only on new dse_perf tables. Manually change dse_perf tables that were created in earlier releases to use TWCS. For example:

```
ALTER TABLE dse_perf.read_latency_histograms WITH
  COMPACTION={'class':'TimeWindowCompactionStrategy'};
```

7. DSE 6.7 introduces, and enables by default, the DSE Metrics Collector, a diagnostics information aggregator used to help facilitate DSE problem resolution. For more information on the DSE Metrics Collector, see DataStax Enterprise Metrics Collector.

8. DSE Search only:

- The appender SolrValidationErrorAppender and the logger SolrValidationErrorLogger are no longer used and may safely be removed from logback.xml.
- In contrast to earlier versions, DataStax recommends accepting the new default value of 1024 for back_pressure_threshold_per_core in dse.yaml. See Configuring and tuning indexing performance.
- If SpatialRecursivePrefixTreeFieldType (RPT) is used in the search schema, replace the units field type with a suitable (degrees, kilometers, or miles) distanceUnits, and then verify that spatial queries behave as expected.
- The appender SolrValidationErrorAppender and the logger SolrValidationErrorLogger are no longer used and may safely be removed from logback.xml.
- Applies only if you are using HTTP writes with JSON documents (deprecated), a known issue causes auto generated solrconfig.xml to have invalid requestHandler for JSON core creations after upgrade to 5.1.0. Change the auto generated solrconfig.xml:

```
<requestHandler name="/update/json"
  class="solr.UpdateUpdateRequestHandler" startup="lazy"/>
```

to

```
<requestHandler name="/update/json"
  class="solr.UpdateRequestHandler" startup="lazy"/>
```
• Slow startup on nodes with large encrypted indexes is resolved. However, action is required to realize the performance gains. You must do a full reindex of all encrypted search indexes on each node in your cluster. Plan sufficient time after the upgrade is complete to reindex with deleteAll=true in a rolling fashion. For example:

$ dsetool reload_core keyspace_name.table_name distributed=false reindex=true deleteAll=true

9. DSEFS only:

A new schema is available for DSEFS.

**Warning:** Dropping a keyspace is not recoverable without a backup. If you have non-temporary data, do not drop the dsefs keyspace. No action is required. DSEFS will continue to work using the DSE 5.0 schema.

If you have no data in DSEFS or if you are using DSEFS only for temporary data, follow these steps to use the new schema:

a. Stop DSEFS on all nodes. In the dsefs_options section of dse.yaml, set **enabled:** false.

b. Restart the DSE node.

c. Drop the dsefs keyspace:

$ DROP KEYSPACE dsefs

d. Clear the dsefs data directories on each node.

For example, if the dsefs_options section of dse.yaml has **data_directories** configured as:

```
dsefs_options:
  ...
  data_directories:
    - dir: /var/lib/dsefs/data
```

this command removes the directories:

$ rm -r /var/lib/dsefs/data/*

e. Start DSEFS with DSE 6.7 to use the new schema.

f. If you backed up existing DSEFS data before the upgrade, copy the data back into DSEFS from local storage.

10. DSE Analytics only:
• Spark Jobserver uses DSE custom version 0.8.0.44. Applications must use the compatible Spark Jobserver API from the DataStax repository.
• If you are using Spark SQL tables, migrate them to the new Hive metastore format:

$ dse client-tool spark metastore migrate --from 5.0.0 --to 6.0.0

11. Ensure that keyspace replication factors are correct for your environment:
   • Check the keyspace replication factor for analytics keyspaces.
   • Check the keyspace replication factor for system_auth and dse_security keyspaces.

12. DSE 6.7 introduces, and enables by default, the DSE Metrics Collector, a diagnostics information aggregator used to help facilitate DSE problem resolution. For more information on the DSE Metrics Collector, see DataStax Enterprise Metrics Collector.

Warning messages during and after upgrade

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

• When upgrading nodes with DSE Advanced Replication, there might be some WriteTimeoutExceptions during a rolling upgrade while mixed versions of nodes exist. Some write consistency limitations apply while mixed versions of nodes exist. The WriteTimeout issue is resolved after all nodes are upgraded.
• Some gremlin_server properties in earlier versions of DSE are no longer required. If properties exist in the dse.yaml file after upgrading, logs display warnings similar to:

  WARN  [main] 2017-08-31 12:25:30,523 GREMLIN
  DseWebSocketChannelizer.java:149 - Configuration for the
  org.apache.tinkerpop.gremlin.driver.ser.GraphSONMessageSerializerGremlinV1d0
  serializer in dse.yaml overrides the DSE default - typically it is
  best to allow DSE to configure these.

  You can ignore these warnings or modify dse.yaml so that only the required gremlin
  server properties are present.

Error messages provide information to help identify problems.

• If you see an error message like:

  - Cannot start node if snitch's data center (Cassandra) differs
  from previous data center (Analytics). Please fix the snitch
  configuration, decommission and rebootstrap this node or use the flag
  -Dcassandra.ignore_dc=true.
Follow upgrade instructions in 6 (page 106). You must start in Spark mode or add the special start-up parameter \texttt{-Dcassandra.ignore_dc=true} for each node.

Upgrading from DataStax Enterprise 5.1 to 6.0

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

Follow these instructions to upgrade from DataStax Enterprise (DSE) 5.1 to DSE 6.0. If you are on DSE 5.0, see Upgrading from DataStax Enterprise 5.0 to 6.0 (page 89).

\textbf{Note:} The DataStax Installer is not supported for DSE 6.0 and later. To upgrade from DSE 5.1 that was installed with the DataStax Installer, you must first change from a standalone installer installation to a tarball or package installation for the same DSE version. See Upgrading to DSE 6.0 or DSE 6.7 from DataStax Installer installations (page 110).

Always upgrade to latest patch release (page 10) on your current version before you upgrade to a higher version. Fixes included in the latest patch release might help or smooth the upgrade process.

The latest 5.1.x version of DSE is 5.1.15.

\textbf{Attention: Read and understand these instructions before upgrading.} Carefully reviewing the planning (page 9) and upgrading instructions can ensure a smooth upgrade and avoid pitfalls and frustrations.

\textbf{Important:} Support for Thrift-compatible tables (COMPACT STORAGE) is dropped in DSE 6.0. Before upgrading to DSE 6.0, all tables that have COMPACT STORAGE to CQL table format must be migrated. Use the \texttt{ALTER TABLE DROP COMPACT STORAGE} command to migrate Thrift-compatible tables to CQL table format. This command is available in DSE 5.1.6 or later.

General recommendations

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

\textbf{Tip:} OpsCenter provides a Backup Service that manages enterprise-wide backup and restore operations for DataStax Enterprise clusters. OpsCenter 6.5 and later is recommended.

\textbf{Attention:} DSE Graph backup snapshots taken with OpsCenter 6.1 will not load to DSE 6.0.
General restrictions and limitations during the upgrade process

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. If you have the OpsCenter Repair Service configured, turn off the Repair Service.
- Ensure OpsCenter compatibility. See the compatibility table.
- During the upgrade, do not bootstrap or decommission nodes.
- Do not issue these types of CQL queries during a rolling restart: DDL and TRUNCATE.
- The default number of threads used by performance objects increased from 1 to 4. During upgrade, compatible performance objects continue to work during the upgrade process. Incompatible performance objects that require schema changes will work in legacy mode or will start working after the upgrade is complete. Do not change the configuration of performance objects during upgrade. If performance objects were disabled before the upgrade, do not enable them during upgrade.
- NodeSync waits to start until all nodes are upgraded.
- 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.

Note: Nodes on different versions might show a schema disagreement during an upgrade.

Restrictions for DSE Advanced Replication nodes

Upgrades are supported only for DSE Advanced Replication V2.

Restrictions for DSE Analytic (Spark) nodes

- Do not run analytics jobs until all nodes are upgraded.
- All nodes in the cluster must be upgraded to the new version before Spark Worker and Spark Master will start.

DSEFS nodes restrictions

During upgrade, DSEFS will not start on upgraded nodes. After all nodes are upgraded to 6.0.0, the DSEFS keyspace is adjusted and then DSEFS starts.

DSE Graph nodes restrictions

Graph nodes have the same restrictions as the workload they run on. Do not alter graph schema during upgrades. Workload-specific restrictions apply for analytics and search nodes, such as no OLAP queries during upgrades.

DSE Search upgrade restrictions and limitations

- Do not update schemas.
- Do not reindex DSE Search nodes during upgrade.
• DSE 6.0 introduces a new Lucene codec. Segments written with this new codec cannot be read by earlier versions of DSE. To downgrade to earlier versions, the entire data directory for the search index in question must be cleared.

**Important:** Before you upgrade DSE Search or SearchAnalytics workloads, you must follow the specific tasks in Preparing to upgrade (page 80) section.

**Restrictions for nodes using any kind of security**

• Do not change security credentials or permissions until the upgrade is complete on all nodes.

• If you are not already using Kerberos, do not set up Kerberos authentication before upgrading. First upgrade the cluster, and then set up Kerberos.

**Upgrading drivers and possible impact when driver versions are incompatible**

Be sure to check driver compatibility. Depending on the driver version, you might need to recompile your client application code. See DataStax driver changes. 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:

• **Protocol version:** Because some drivers can use different protocol versions, force the protocol version at start up. For example, keep the Java driver at its current protocol version while the driver upgrade is happening. Switch to the Java driver to the new protocol version only after the upgrade is complete on all nodes in the cluster.

• **Initial contact points:** 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 details on protocol version negotiation, see protocol versions with mixed clusters in the Java driver version you're using, for example, Java driver.

**Preparing to upgrade**

Follow these steps to prepare each node for upgrading from DSE 5.1 to DSE 6.0.

**Note:** These steps are performed in your current version and use DSE 5.1 documentation.

1. Carefully review Planning your DataStax Enterprise upgrade (page 9).

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

2. Replace ITriggers and custom interfaces.
Several internal and beta extension points were modified to necessitate core storage engine refactoring. All custom implementations, including the following interfaces, must be replaced with supported implementations when upgrading to DSE 6.0. Because a rewrite of the following interfaces is required for DSE 6.0: (For help contact the DataStax Services team.)

- The `org.apache.cassandra.triggers.ITrigger` interface was modified from `augment` to `augmentNonBlocking` for non-blocking internal architecture. Updated trigger implementations must be provided on upgraded nodes. If unsure, drop all existing triggers before upgrading. To check for existing triggers:

  ```
  SELECT * FROM system_schema.triggers
  ```

- The `org.apache.cassandra.index.Index` interface was modified to comply with the core storage engine changes. Updated implementations are required. If unsure, drop all existing custom secondary indexes before upgrading, except DSE Search indexes, which do not need to be replaced. To check for existing indexes:

  ```
  SELECT * FROM system_schema.indexes
  ```

- The `org.apache.cassandra.cql3.QueryHandler`, `org.apache.cassandra.db.commitlog.CommitLogReadHandler`, and other extension points have been changed. See QueryHandlers.

3. Before upgrading, be sure that each node has ample free disk space.
   The required space depends on the compaction strategy. See Disk space.

4. Familiarize yourself with the changes and features in this release:
   - DataStax Enterprise 6.0 release notes.
   - General upgrading advice for any version. Be sure to read NEWS.txt all the way back to your current version.
   - DataStax Enterprise changes in CHANGES.txt.
   - DataStax driver changes.

5. Verify your current product version.

   ```
   $ dse -v
   ```

   These instructions are valid only for upgrades from DSE 5.1 to DSE 6.0.

6. Back up the configuration files you use to a folder that is not in the directory where you normally run commands.
   The configuration files are overwritten with default values during installation of the new version.
7. Upgrade to the latest patch release on your current version. The latest 5.1.x version of DSE is 5.1.15.

Always upgrade to latest patch release on your current version before you upgrade to a higher version. Fixes included in the latest patch release might help or smooth the upgrade process.

8. Upgrade the SSTables on each node to ensure that all SSTables are on the current version.

   $ nodetool upgradesstables

This step is required for DataStax Enterprise upgrades that include a major Cassandra version changes.

   **Warning:** Failure to upgrade SSTables when required results in a significant performance impact and increased disk usage.

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

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

   $ java -version

   - **Recommended. OpenJDK 8 (1.8.0_151 minimum)**
     
     **Note:** Recommendation changed due to the end of public updates for Oracle JRE/JDK 8. See Oracle Java SE Support Roadmap.

   - **Supported. Oracle Java SE 8 (JRE or JDK) (1.8.0_151 minimum)**

     **Important:** Although Oracle JRE/JDK 8 is supported, DataStax does more extensive testing on OpenJDK 8.

10. Run nodetool repair to ensure that data on each replica is consistent with data on other nodes.

11. Install the libaio package for optimal performance.

    RHEL platforms:

    $ sudo yum install libaio

    Debian:

    $ sudo apt-get install libaio1

12. DSE Analytics nodes:
a. Support for Thrift-compatible tables (COMPACT STORAGE) is dropped. Before upgrading, follow the steps to migrate all tables that have COMPACT STORAGE to CQL table format while DSE 5.x.x is running.

   **Note:** Do not migrate system.* tables, COMPACT STORAGE is removed by DSE internally. Modifying the system keyspace is not supported; modification attempts generate an error.

   For DSE Analytics, drop compact storage from all the tables in the "HiveMetaStore" and PortfolioDemo keyspaces.

   After COMPACT STORAGE is dropped, columns to support migration to CQL-compatible table format are added as described in migrating from compact storage.

   **Attention:** DSE 6.0 will not start if COMPACT STORAGE tables are present. Creating a COMPACT STORAGE table in a mixed-version cluster is not supported. Driver connections to the latest DSE 5.0.x and DSE 5.1.x run in a "NO_COMPACT" mode that causes compact tables to appear as if the compact flags were already dropped, but only for the current session.

b. If you programmatically set the shuffle parameter, you must change the code for applications that use `conf.set("spark.shuffle.service.port", port)`. Instead, use `dse spark-submit` which automatically sets the correct service port based on the authentication state. See Configuring Spark for more information.

c. If DSEFS is enabled, copy CFS hivemetastore directory to dse:

   ```
   $ DSE_HOME/bin/dse hadoop fs -cp cfs://127.0.0.1/user/spark/warehouse/ dsefs://127.0.0.1/user/spark/warehouse/
   ```

   After upgrade is complete migrate Spark SQL tables (if used) to the new Hive metastore format:

   ```
   $ dse client-tool spark metastore migrate --from 5.0.0 --to 6.0.0
   ```

d. Cassandra File System (CFS) is removed. Remove the cfs and cfs_archive keyspaces before upgrading. See the From CFS to DSEFS blog post and the Copying data from CFS to DSEFS documentation for more information.

e. Make sure any use of the SPARK_LOCAL_DIRS and SPARK_EXECUTOR_DIRS environment variables match their use as described in Setting environment variables.

f. For applications to use the compatible Spark Jobserver API in DataStax repository, migrate jobs that extend from SparkHiveJob and SparkSqlJob to SparkSessionJob. See example in the DemoSparkSessionJob in the demos directory.

   **Note:** Spark Jobserver is the DSE custom version 8.0.4.45.

   The default location of the demos directory depends on the type of installation:
• **Package installations:** /usr/share/dse/demos
• **Tarball installations:** installation_location/demos

13. **DSE Search nodes:** Review DSE 6.0 release notes for all changes.
   - Ensure all use of HTTP writes are changed to use CQL commands for updates and inserts.
   - Edit the search index config and make these changes, as needed. See Search index config for valid options to change query behavior for search indexes.
     # Remove the unsupported dataDir option. You can still set the location of search indexes.
     # Remove mergePolicy, maxMergeDocs, and mergeFactor. For example:

```
<mergeFactor>25</mergeFactor>
<maxMergeDocs>...
<mergePolicy>...
```

Use mergePolicyFactory instead, and add mergeScheduler:

```
<mergeScheduler
  class="org.apache.lucene.index.ConcurrentMergeScheduler">
  <int name="maxThreadCount">16</int>
  <int name="maxMergeCount">32</int>
</mergeScheduler>
...  
<mergePolicyFactory
  class="org.apache.solr.index.TieredMergePolicyFactory">
  <int name="maxMergeAtOnce">10</int>
  <int name="segmentsPerTier">10</int>
</mergePolicyFactory>
```

# Remove any instance of ExtractingRequestHandler.
# Remove DSENRTCachingDirectoryFactory. Change:

```
<directoryFactory name="DirectoryFactory"
  class="com.datastax.bdp.search.solr.DSENRTCachingDirectoryFactory"/>
```

to:

```
<directoryFactory name="DirectoryFactory"
  class="solr.StandardDirectoryFactory"/>
```

• **Ensure that the catalina.properties and context.xml files are present in the Tomcat conf dir. DSE will not start after upgrade if these files are missing.**

The default location of the Tomcat conf directory depends on the type of installation:
# Package installations: /etc/dse/tomcat/conf
# Tarball installations: installation_location/resources/tomcat/conf

- If earlier DSE versions use a custom configuration for the Solr UI web.xml, change:

  ```xml
  <filter-class>com.datastax.bdp.search.solr.auth.DseAuthenticationFilter</filter-class>
  ```

  to

  ```xml
  <filter-class>com.datastax.bdp.cassandra.auth.http.DseAuthenticationFilter</filter-class>
  ```

- StallMetrics MBean is removed. Change operators that use the MBean.

14. DSE Graph nodes:

- Ensure that edge label names and property key names use only the supported characters. Edge label names and property key names allow only [a-zA-Z0-9], underscore, hyphen, and period. In earlier versions, edge label names and property key names allowed nearly unrestricted Unicode.

  ```
  # schema.describe() displays the entire schema, even if it contains illegal names.
  # In-place upgrades allow existing schemas with invalid edge label names and property key names.
  # Schema elements with illegal names cannot be updated or added.
  ```

Upgrade steps

To upgrade from DSE 5.1 to DSE 6.0, follow these steps on each node in the recommended order. The upgrade process requires upgrading and restarting one node at a time.

**Note:** These steps are performed in your upgraded version and use DSE 6.7 documentation.

**Note:** You can ignore some warning messages that are displayed (page 89) during and after upgrade.

1. To flush the commit log of the old installation:

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

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

   **Important:** This step is mandatory when upgrading between major Cassandra versions that change SSTable formats, rendering commit logs from the previous version incompatible with the new version.
2. Stop the node. See Stopping a DataStax Enterprise node.
   • To stop DataStax Enterprise running as a service:
     
     ```
     $ sudo service dse stop
     ```
   • To stop DataStax Enterprise running as a stand-alone process:
     
     ```
     $ bin/dse cassandra-stop
     ```

3. Use the appropriate method to install the new product version on a supported platform:
   • Package installer using YUM
   • Package installer using APT
   • Binary tarball installer

   **Note:** Install the new product version using the same installation type that is on the system, otherwise problems might result.

4. 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.
      • **DSE Search nodes**
        
        ```
        # While the node is down, edit dse.yaml and remove these options:
        # cql_solr_query_executor_threads
        # enable_back_pressure_adaptive_nrt_commit
        # max_solr_concurrency_per_core
        # solr_indexing_error_log_options
        DSE 6.0 will not start with these options present.
        ```
      • Be sure you are familiar (page 81) with the Apache Cassandra and DataStax Enterprise changes and features in the new release.
      • Ensure that keyspace replication factors are correct for your environment:
        ```
        # Check the keyspace replication factor for analytics keyspaces.
        # Check the keyspace replication factor for system_auth and dse_security keyspaces.
        ```

   b. Edit the cassandra.yaml file and comment out or remove these credentials cache settings, if they exist:
      
      ```
      credentials_validity_in_ms
      credentials_update_interval_in_ms
      ```
Caches are optimized without these settings.

c. Merge the applicable modifications into the new version.

5. Start the node.
   - Package installations: See Starting DataStax Enterprise as a service.
   - Tarball installations: See Starting DataStax Enterprise as a stand-alone process.

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

   $ nodetool status

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

   Warnings, errors, and exceptions are frequently found in the logs when starting 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.

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

9. When the upgrade includes a major Cassandra version, you must upgrade the SSTables. DataStax recommends upgrading the SSTables on one node at a time or when using racks, one rack at a time.

   **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.

   $ nodetool upgradesstables

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

   Use the `--jobs` option to set the number of SSTables that upgrade simultaneously. The default setting is 2, which minimizes impact on the cluster. Set to 0 to use all available compaction threads. DataStax recommends running the `upgradesstables` command on one node at a time or when using racks, one rack at a time.

   **Note:** You can run the `upgradesstables` command before all the nodes are upgraded as long as you run this command on only one node at a time or when using racks, one rack at a time. Running `upgradesstables` on too many nodes will degrade performance.
Recovery after upgrading to DSE 6.0 without dropping compact storage

DSE 6.0 removed support for Thrift-compatible tables (Compact Storage). All tables using Compact Storage must be dropped or migrated to CQL table format before upgrading to DSE 6.0. If a cluster has been upgraded to DSE 6.0 and any Compact Storage tables still exist, follow this procedure to recover and proceed with the upgrade:

1. Downgrade any nodes which were already upgraded to DSE 6.0 to the latest version in the DSE 5.0 or 5.1 series:
   - DSE 5.0.x, downgrade to 5.0.15 or later
   - DSE 5.1.x, downgrade to 5.1.15 or later

2. On each node that was attempted to be started on DSE 6.0, start DSE with the --Dcassandra.commitlog.ignorereplayerrors=true option.

3. On one node (any node) in the cluster, DROP COMPACT STORAGE from tables which use it. Modifying the system keyspace is not supported; modification attempts generate an error.

4. Restart DSE to continue the upgrade to DSE 6.0.

After the upgrade

After all nodes are upgraded and running on DSE 6.0, complete these steps:

1. If you use the OpsCenter Repair Service, turn on the Repair Service.

2. After all nodes are on DSE 6.0 and the required schema change occurs, the new authorization with CassandraAuthorizer enables the use of new columns.

3. **DSE Search only:**
   - The appender SolrValidationErrorAppender and the logger SolrValidationErrorLogger are no longer used and may safely be removed from logback.xml.
   - In contrast to earlier versions, DataStax recommends accepting the new default value of 1024 for back_pressure_threshold_per_core in dse.yaml. See Configuring and tuning indexing performance.
   - Slow startup on nodes with large encrypted indexes is resolved. However, action is required to realize the performance gains. You must do a full reindex of all encrypted search indexes on each node in your cluster. Plan sufficient time after the upgrade is complete to reindex with deleteAll=true in a rolling fashion. For example:

     $ dsetool reload_core keyspace_name.table_name distributed=false reindex=true deleteAll=true

4. **DSE Analytics only:**
• Check the replication factor for the dse-analytics keyspace, a new keyspace stores all DSE Analytics internal system data. DataStax recommends setting the replication strategy to NetworkTopologyStrategy (NTS) with a replication factor of at least 3 in each of DSE Analytics datacenters. If a datacenter has more nodes, a larger replication factor should be considered.

• Spark Jobserver uses DSE custom version 0.8.0.44. Applications must use the compatible Spark Jobserver API from the DataStax repository.

Warning messages during and after upgrade
Error messages provide information to help identify problems. You can ignore some log messages that occur during and after an upgrade.

• Some gremlin_server properties in earlier versions of DSE are no longer required. If properties exist in the dse.yaml file after upgrading, logs display warnings similar to:

```java
WARN  [main] 2017-08-31 12:25:30,523 GREMLIN
DseWebSocketChannelizer.java:149 - Configuration for the
org.apache.tinkerpop.gremlin.driver.ser.GraphSONMessageSerializerGremlinV1d0
serializer in dse.yaml overrides the DSE default - typically it is
best to allow DSE to configure these.
```

You can ignore these warnings or modify dse.yaml so that only the required gremlin server properties are present.

Upgrading from DataStax Enterprise 5.0 to 6.0

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

Follow these instructions to upgrade from DataStax Enterprise 5.0 to DataStax Enterprise 6.0.

If you have DSE 4.8 or earlier, upgrade to the latest version of 5.0 (page 133) before continuing.

Always upgrade to latest patch release (page 10) on your current version before you upgrade to a higher version. Fixes included in the latest patch release might help or smooth the upgrade process.

**Note:** The DataStax Installer is not supported for DSE 6.0 and later. To upgrade from DSE 5.0 that was installed with the DataStax Installer, you must first change from a standalone installer installation to a tarball or package installation for the same DSE version. See Upgrading to DSE 6.0 or DSE 6.7 from DataStax Installer installations (page 110).

The latest version of DSE 5.0 is 5.0.15.
Attention: Read and understand these instructions before upgrading. Carefully reviewing the planning (page 9) and upgrading instructions can ensure a smooth upgrade and avoid pitfalls and frustrations.

Important: Support for Thrift-compatible tables (COMPACT STORAGE) is dropped in DSE 6.0. Before upgrading to DSE 6.0, you must migrate all tables that have COMPACT STORAGE to CQL table format. The command to migrate Thrift-compatible tables to CQL table format is available in DSE 5.0.12 or later.

Apache Cassandra™ version change

Upgrading from DataStax Enterprise 5.0 to 6.0 includes a major Cassandra version change.

- DataStax Enterprise 6.0 is compatible with Cassandra 3.11.
- DataStax Enterprise 5.0 uses Cassandra 3.0.

Be sure to follow the recommendations for upgrading the SSTables.

General recommendations

DataStax recommends backing up your data prior to any version upgrade, including logs and custom configurations. 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. OpsCenter 6.5 and later is recommended.

General restrictions and limitations during the upgrade process

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. If you have the OpsCenter Repair Service configured, turn off the Repair Service.
- Ensure OpsCenter compatibility. See DataStax OpsCenter compatibility with DataStax Enterprise.
- During the upgrade, do not bootstrap or decommission nodes.
- Do not issue these types of CQL queries during a rolling restart: DDL and TRUNCATE.
- NodeSync waits to start until all nodes are upgraded.
- Do not enable Change Data Capture (CDC) on a mixed-version cluster. Upgrade all nodes to DSE 5.1 or later before enabling CDC.
- 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.
Note: Nodes on different versions might show a schema disagreement during an upgrade.

Restrictions for DSE Analytic (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 Advanced Replication

- Support for DSE Advanced Replication V1 in DSE 5.0 is removed. Because DSE Advanced Replication V2 is substantially revised, for V1 installations, you must first upgrade to DSE 5.1.x (page 115) and migrate to DSE Advanced Replication to V2, and then upgrade to DSE 6.0 (page 78).

Restrictions for DSEFS

Mixed versions of DSEFS are not supported during the upgrade process.

- Complete the upgrade on all nodes before running fsck. Starting with DSE 5.1.3, all nodes must be able to report proper block status to the node running fsck. If you run fsck on an upgraded node in a mixed version cluster, nodes with versions earlier than DSE 5.1.3 do not properly report block status and cause the fsck to incorrectly assume that data is corrupt or unavailable. The fsck will incorrectly try to repair them.
- A protocol change in DSE 5.1.3 improves efficiency of passing JSON arrays between DSEFS server and client. Upgrade all nodes in the cluster before using the DSEFS shell.

Restrictions for DSE Graph

Graph nodes have the same restrictions as the workload they run on. General graph restrictions apply for all nodes, such as not altering graph schema during upgrades. Workload-specific restrictions apply for analytics and search nodes, such as no OLAP queries during upgrades.

Restrictions for DSE Search

- Do not update schemas.
- Do not reindex DSE Search nodes during upgrade.
- DSE 6.0 introduces a new Lucene codec. Segments written with this new codec cannot be read by earlier versions of DSE. To downgrade to earlier versions, the entire data directory for the search index in question must be cleared.
- DSE Search in DataStax Enterprise 6.0 uses Apache Solr 6.0. This significant change requires advanced planning and specific actions before and after the upgrade.

   Important: Before you upgrade DSE Search or SearchAnalytics workloads, you must follow the specific steps in Advanced preparation for upgrading DSE Search and SearchAnalytics nodes (page 92).

Restrictions for nodes using any kind of security
• Do not change security credentials or permissions until the upgrade is complete on all nodes.
• If you are not already using Kerberos, do not set up Kerberos authentication before upgrading. First upgrade the cluster, and then set up Kerberos.

Security changes
After upgrading, the default authenticator is DseAuthenticator and default authorizer is DseAuthorizer in cassandra.yaml. Other authorizers and authenticators are no longer supported, follow the steps in After the upgrade (page 107).

Upgrading drivers and possible impact when driver versions are incompatible
Be sure to check driver compatibility. Depending on the driver version, you might need to recompile your client application code. See DataStax driver changes. 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:

• **Protocol version:** Because some drivers can use different protocol versions, force the protocol version at start up. For example, keep the Java driver at its current protocol version while the driver upgrade is happening. Switch to the Java driver to the new protocol version only after the upgrade is complete on all nodes in the cluster.

• **Initial contact points:** 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 details on protocol version negotiation, see protocol versions with mixed clusters in the Java driver version you're using, for example, Java driver.

Advanced preparation for upgrading DSE Search and SearchAnalytics nodes
Before starting the Preparing to upgrade (page 99) steps, complete all the advanced preparation steps on DSE Search and SearchAnalytics nodes while DSE 5.0 is still running.

Plan sufficient time to implement and test the required changes before the upgrade:
• Schema changes require a full reindex.
• Configuration changes require reloading the core.

1. Change HTTP queries to CQL:
   • Delete-by-id is removed, use CQL DELETE by primary key instead.
   • Delete-by-query no longer supports wildcards, use CQL TRUNCATE instead.

2. If any Solr core was created on DSE 4.6 or earlier and never reindexed after being upgraded to DSE 4.7 or later, you must reindex on DSE 5.0 before upgrading to DSE 5.1 or later.

3. Ensure that the shard_transport_options in dse.yaml are set only for netty_client_request_timeout:
In DSE 5.1 and later, the shard transport option supports only `netty_client_request_timeout`. Remove any other `shard_transport_options`.

4. If you are using Apache Solr SolrJ, the minimum required version is **6.0.0**.

5. For `SpatialRecursivePrefixTreeFieldType (RPT)` in search schemas, you must adjust your queries for these changes:
   - `IsDisjointTo` is no longer supported in queries on `SpatialRecursivePrefixTreeFieldType`. Replace `IsDisjointTo` with a NOT Intersects query. For example:
     ```
     foo:0,0 TO 1000,1000 AND -"Intersects(POLYGON((338 211, 338 305, 404 305, 404 211, 338 211)))")"
     ```
   - The ENVELOPE syntax is now required for WKT-style queries against `SpatialRecursivePrefixTreeFieldType` fields. You must specify `ENVELOPE(10, 15, 15, 10)`, where queries on earlier releases could specify `10 10 15 15`.

   See [Spatial Search](#) for details on using distanceUnits in spatial queries.

6. **For upgrades to DSE 6.0.0 and DSE 6.0.1 only** Stored=true copy fields are not supported and cause schema validation to fail. The stored=true copyField directive has not been supported since DSE 4.7, so you probably do not have Stored=true copy fields. If you do:
   - Change the stored attribute value of all copyField directives from true to false in the `schema.xml` file and then use `dsetool reload_core` to reload the modified schema.
   - You must ensure that application designs and implementations recognize this change.

   **Note:** DSE 6.0.2 and later ignores stored=true.

7. Edit the `solrconfig.xml` file and make these changes, as needed:
   - Remove these requestHandlers: `XmlUpdateRequestHandler`, `BinaryUpdateRequestHandler`, `CSVRequestHandler`, `JsonUpdateRequestHandler`, `DataImportHandler`. Solr deprecated and then removed these requestHandlers.

   For example:
   ```
   <requestHandler name="/dataimport" class="solr.DataImportHandler"/>
   ```
   or
   ```
   <requestHandler name="/update"
   class="solr.XmlUpdateRequestHandler"/>
   ```
• Change the directoryFactory from:

```xml
<directoryFactory name="DirectoryFactory"
class="${solr.directoryFactory:solr.StandardDirectoryFactory}"/>
```

to

```xml
<directoryFactory name="DirectoryFactory"
class="solr.StandardDirectoryFactory"/>
```

• `<unlockOnStartup>` is now unsupported as a result of LUCENE-6508 and SOLR-7942.

• Change the updateLog from:

```xml
<updateLog class="solr.FSUpdateLog" force="false">
```

to

```xml
<updateLog force="false">
```

8. Upgrading DSE search nodes to DSE 5.1 and later requires replacing unsupported Solr types with supported types.

**Note:** Special handling is also required for BCDStrField, addressed in 10 (page 96).

Sorting limitations apply to mixed version clusters. Some of the removed Solr types, due to the way they marshal sort values during distributed queries (combined with the way the suggested new types unmarshal sort values), cannot be sorted on during rolling upgrades when some nodes use an unsupported type and other nodes use the suggested new type. The following type transitions are problematic:

<table>
<thead>
<tr>
<th>Removed Solr field types</th>
<th>Supported Solr field types</th>
</tr>
</thead>
<tbody>
<tr>
<td>ByteField</td>
<td>TrieIntField</td>
</tr>
<tr>
<td>DateField</td>
<td>TrieDateField</td>
</tr>
<tr>
<td>BCDIntField</td>
<td>TrieIntField</td>
</tr>
<tr>
<td>BCDLongField</td>
<td>TrieLongField</td>
</tr>
</tbody>
</table>

Two options are available:

a. Avoid sorting on removed Solr field types until the upgrade to DSE 5.1 or later is complete for all nodes in the datacenter being queried.

**Tip:** When using two search datacenters, isolate queries to a single datacenter and then change the schema and reindex the other datacenter. Then isolate queries to the newly reindexed datacenter while you change the schema and upgrade the first datacenter.
b. If you are using BCDIntField or BCDLongField, update the schema to replace BCDIntField and BCDLongField with types that are sort-compatible with the supported Solr types TrieIntField and TrieLongField:

<table>
<thead>
<tr>
<th>Removed Solr field types</th>
<th>Interim sort-compatible supported Solr field types</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCDIntField</td>
<td>SortableIntField</td>
</tr>
<tr>
<td>BCDLongField</td>
<td>SortableLongField</td>
</tr>
</tbody>
</table>

Change the schema in a distributed fashion, and do not reindex. After the schema is updated on all nodes, then go on to 9 (page 95).

9. Update the schema and configuration for the Solr field types that are removed from Solr 5.5 and later.

- Update the schema to replace unsupported Solr field types with supported Solr field types:

<table>
<thead>
<tr>
<th>Removed Solr field types</th>
<th>Supported Solr field types</th>
</tr>
</thead>
<tbody>
<tr>
<td>ByteField</td>
<td>TrieIntField</td>
</tr>
<tr>
<td>DateField</td>
<td>TrieDateField</td>
</tr>
<tr>
<td>DoubleField</td>
<td>TrieDoubleField</td>
</tr>
<tr>
<td>FloatField</td>
<td>TrieFloatField</td>
</tr>
<tr>
<td>IntField</td>
<td>TrieIntField</td>
</tr>
<tr>
<td>LongField</td>
<td>TrieLongField</td>
</tr>
<tr>
<td>ShortField</td>
<td>TrieIntField</td>
</tr>
<tr>
<td>SortableDoubleField</td>
<td>TrieDoubleField</td>
</tr>
<tr>
<td>SortableFloatField</td>
<td>TrieFloatField</td>
</tr>
<tr>
<td>SortableIntField</td>
<td>TrieIntField</td>
</tr>
<tr>
<td>SortableLongField</td>
<td>TrieLongField</td>
</tr>
<tr>
<td>BCDIntField</td>
<td>TrieIntField</td>
</tr>
<tr>
<td>BCDLongField</td>
<td>TrieLongField</td>
</tr>
<tr>
<td>BCDStrField (see 10 (page 96) if used)</td>
<td>TrieIntField</td>
</tr>
</tbody>
</table>

- If you are using type mapping version 0, or you do not specify a type mapper, verify or update the solrconfig.xml to use dseTypeMappingVersion 1:

```xml
<dseTypeMappingVersion>1</dseTypeMappingVersion>
```

If the Solr core is backed by a CQL table and the type mapping is unspecified, use type mapping version 2.
• Reload the core:

    $ dsetool reload_core keyspace_name.table_name schema=filepath
    solrconfig=filepath

If you were using the unsupported data types, do a full reindex node-by-node:

    $ dsetool reload_core keyspace_name.table_name schema=filepath
    solrconfig=filepath reindex=true deleteAll=true distributed=false

**Note:** In DSE 5.1 and later, auto generated schemas use data type mapper 2.

10. **If using BCDStrField**: In DSE 5.0 and earlier, DSE mapped Cassandra text columns to BCDStrField. The deprecated BCDStrField was removed in DSE 5.1.0.

   The recommended strategy is to upgrade the data type to TrieIntField. However, DSE cannot map text directly to TrieIntField. If you are using BCDStrField, you must complete one of these options before the upgrade.

   a. If BCDStrField is no longer used, remove the BCDStrField field from the Solr schema. Reindexing is not required.

   b. If you want to index the field as a TrieIntField, and a full reindex is acceptable, change the underlying database column to use the type int.

   c. If you want to keep the database column as text and you still want to do simple matching queries on the indexed field, switch from BCDStrField to StrField in the schema. Indexing should not be required, but the field will no longer be appropriate for numeric range queries or sorting, because StrField uses a lexicographic order, not a numeric one.

   d. **Not recommended**: If you want to keep the database column as text and still want to perform numeric range queries and sorts on the former BCDStrField, but would rather change their application than perform a full reindex:

      • Change the field to StrField in the Solr schema with indexed=false.
      • Add a new copy field with the type TrieIntField that has its values supplied by the original BCDStrField.

   This solution still requires reindex to work, because the copy field target must be populated. This non-recommended option is supplied only to support a sub-optimal data model; for example, a text column with values that would fit only into an int.

After you make these schema changes, do a rolling, node-by-node `reload_core` with reindex=true, distributed=false, and deleteAll=true.

**Note:** If you have two datacenters and upgrade them one at a time, reload the core with distributed=true and deleteAll=true.
11. Tune the schema before you upgrade. For DSE 5.1.4 and later, all field definitions in the schema are validated and must be DSE Search compatible, even if the fields are not indexed, have docValues applied, or used for copy-field source. The default behavior of automatic resource generation includes all columns. To improve performance, take action to prevent the fields from being loaded from the database. Include only the required fields in the schema by removing or commenting out unused fields in the schema.

Advanced preparation for upgrading DSE Graph nodes with search indexes

These steps apply to graph nodes that have search indexes. Before starting the Preparing to upgrade (page 99) steps, complete these advanced preparation steps while DSE 5.0 is still running.

Upgrading DSE Graph nodes with search indexes requires these edits to the solrconfig file. Configuration changes require reloading the core. Plan sufficient time to implement and test changes that are required before the upgrade.

Edit the solrconfig.xml file and make these changes, as needed:

- Remove these requestHandlers: XmlUpdateRequestHandler, BinaryUpdateRequestHandler, CSVRequestHandler, JsonUpdateRequestHandler, and DataImportHandler. Solr deprecated and then removed these requestHandlers. For example:

  ```
  <requestHandler name="/dataimport" class="solr.DataImportHandler"/>
  or
  <requestHandler name="/update" class="solr.XmlUpdateRequestHandler"/>
  ```

- `<unlockOnStartup>` is now unsupported as a result of LUCENE-6508 and SOLR-7942.

- Reload the core so that this configuration change is respected:

  ```bash
  $ dsetool reload_core keyspace_name.table_name reindex=false
  ```

Advanced preparation for upgrading DSE Analytics nodes

Upgrades from DSE 5.0 to DSE 6.0 include a major upgrade to Spark 2.2, as well as a tighter integration between DSE and Spark. For information on Spark 2.2, see the Spark documentation. Spark 2.2 uses Scala 2.11.

A new Spark resource manager uses the DataStax Java Driver and native CQL protocol for managing communication between DSE Spark nodes. DSE 5.0 used Spark RPC. This resource manager change impacts Spark applications that ran on DSE 5.0.

1. Spark 2.2 uses Scala 2.11. You must recompile all DSE 5.0 Scala Spark applications against Scala 2.11 and use only Scala 2.11 third-party libraries.
Changing the `dse-spark-dependencies` in your build files is not sufficient to change the compilation target. See the example projects for how to set up your build files.

2. Spark applications should use `dse://` URLs instead of `spark://spark_master_IP:Spark_RPC_port_number` URLs, as described in Specifying Spark URLs.

You no longer need to specify the Spark master IP address or hostname when using `dse://` URLs. Connecting to any Spark node will redirect the request to the master node.

3. If you have existing Spark application code that uses `spark://Spark master IP:Spark RPC port` to connect, it will no longer work.

For example, the following code worked in DSE 5.0 but will not work in DSE 5.1 or later.

```scala
val conf = new SparkConf(true)
  .setMaster("spark://192.168.123.10:7077")
  .setAppName("cassandra-demo")
  .set("cassandra.connection.host", "192.168.123.10") // initial contact
  .set("cassandra.username", "cassandra")
  .set("cassandra.password", "cassandra")
val sc = new SparkContext(conf)
```

To connect to DSE 5.1 and later, you no longer need to call `setMaster`. This code will work in DSE 5.1 and later:

```scala
val conf = new SparkConf(true)
  .setAppName("cassandra-demo")
  .set("cassandra.connection.host", "192.168.123.10") // initial contact
  .set("cassandra.username", "cassandra")
  .set("cassandra.password", "cassandra")
val sc = new SparkContext(conf)
```

If you need to specify the master using `setMaster`, use the `dse://` URL format.

4. Starting in DSE 5.1, you can restrict Spark jobs to specific database roles. See Managing Spark application permissions.

5. Starting in DSE 5.1, you can set the Spark executor process owners, as described in Running Spark processes as separate users.

6. The user submitting the Spark application no longer has to be the same database role. See Specifying Spark URLs for information on how to change the master connection submission to use a different user or cluster than the database connection.

Backing up DSEFS data
These steps only apply to nodes that use DSEFS. Before starting the Preparing to upgrade (page 99) steps, complete these advanced preparation steps.

The DSEFS schema used by the database was improved in DSE 5.1, but the old schema is still supported and will not be modified during the upgrade. To use the new DSEFS schema with existing DSEFS data, backup the DSEFS data before upgrading:

Backup the current DSEFS data to local storage using `dse hadoop fs -cp` command:

```sh
$ dse hadoop fs -cp /* /local_backup_location
```

Preparing to upgrade

Follow these steps to prepare each node for upgrading from DataStax Enterprise 5.0 to DataStax Enterprise 6.0.

**Note:** These steps are performed in your current version and use DSE 5.0 documentation.

1. Upgrade to the latest patch release on your current version. Fixes included in the latest patch release can simplify the upgrade process.

2. Before upgrading, be sure that each node has ample free disk space.
   
   The required space depends on the compaction strategy. See Disk space.

3. Familiarize yourself with the changes and features in this release:
   - DataStax Enterprise 6.0 release notes.
   - General upgrading advice for any version. Be sure to read NEWS.txt all the way back to your current version.
   - DataStax Enterprise changes in CHANGES.txt.
   - DataStax driver changes.

4. Replace ITriggers and custom interfaces.

   Several internal and beta extension points were modified to necessitate core storage engine refactoring. All custom implementations, including the following interfaces, must be replaced with supported implementations when upgrading to DSE 6.0. Because a rewrite of the following interfaces is required for DSE 6.0: (For help contact the DataStax Services team.)

   - The `org.apache.cassandra.triggers.ITrigger` interface was modified from `augment` to `augmentNonBlocking` for non-blocking internal architecture. Updated trigger implementations must be provided on upgraded nodes. If unsure, drop all existing triggers before upgrading. To check for existing triggers:

     ```sql
     SELECT * FROM system_schema.triggers
     ```
• The `org.apache.cassandra.index.Index` interface was modified to comply with the core storage engine changes. Updated implementations are required. If unsure, drop all existing custom secondary indexes before upgrading, except DSE Search indexes, which do not need to be replaced. To check for existing indexes:

```
SELECT * FROM system_schema.indexes
```

• The `org.apache.cassandra.cql3.QueryHandler`, `org.apache.cassandra.db.commitlog.CommitLogReadHandler`, and other extension points have been changed. See `QueryHandlers`.

5. Support for Thrift-compatible tables (COMPACT STORAGE) is dropped. Before upgrading, follow the steps to migrate all tables that have COMPACT STORAGE to CQL table format while DSE 5.x.x is running.

   **Note:** Do not migrate system.* tables, COMPACT STORAGE is removed by DSE internally.

   For DSE Analytics, drop compact storage from all the tables in the HiveMetaStore and PortfolioDemo keysaces.

After COMPACT STORAGE is dropped, columns to support migration to CQL-compatible table format are added as described in migrating from compact storage. Modifying the system keyspace is not supported; modification attempts generate an error.

   **Attention:** DSE 6.0 will not start if COMPACT STORAGE tables are present. Creating a COMPACT STORAGE table in a mixed-version cluster is not supported. Driver connections to the latest DSE 5.0.x and DSE 5.1.x run in a "NO_COMPACT" mode that causes compact tables to appear as if the compact flags were already dropped, but only for the current session.

6. If audit logging is configured to use `CassandraAuditWriter`, run these commands as super user on DSE 5.0 nodes, and then ensure that the entire cluster has schema agreement:

```
$ ALTER TABLE dse_audit.audit_log ADD authenticated text;
ALTER TABLE dse_audit.audit_log ADD consistency text;
```

7. Upgrade the SSTables on each node to ensure that all SSTables are on the current version.

```
$ nodetool upgradesstables
```

This step is required for DataStax Enterprise upgrades that include a major Cassandra version changes.

   **Warning:** Failure to upgrade SSTables when required results in a significant performance impact and increased disk usage.
If the SSTables are already on the current version, the command returns immediately and no action is taken.

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

$ java -version

- **Recommended.** OpenJDK 8 (1.8.0_151 minimum)
  
  **Note:** Recommendation changed due to the end of public updates for Oracle JRE/JDK 8. See Oracle Java SE Support Roadmap.

- **Supported.** Oracle Java SE 8 (JRE or JDK) (1.8.0_151 minimum)

  **Important:** Although Oracle JRE/JDK 8 is supported, DataStax does more extensive testing on OpenJDK 8.

9. Run `nodetool repair` to ensure that data on each replica is consistent with data on other nodes.

10. Install the libaio package for optimal performance.

    **RHEL platforms:**

    $ sudo yum install libaio

    **Debian:**

    $ sudo apt-get install libaio1

11. **DSE Analytics nodes:**

    - Support for Thrift-compatible tables (COMPACT STORAGE) is dropped. Before upgrading, follow the steps to migrate all tables that have COMPACT STORAGE to CQL table format while DSE 5.x.x is running.

      **Note:** Do not migrate system.* tables, COMPACT STORAGE is removed by DSE internally. Modifying the system keyspace is not supported; modification attempts generate an error.

    For DSE Analytics, drop compact storage from all the tables in the "HiveMetaStore" and PortfolioDemo keyspaces.

    After COMPACT STORAGE is dropped, columns to support migration to CQL-compatible table format are added as described in migrating from compact storage.

      **Attention:** DSE 6.0 will not start if COMPACT STORAGE tables are present. Creating a COMPACT STORAGE table in a mixed-version cluster is not supported. Driver connections to the latest DSE 5.0.x and DSE 5.1.x run in a "NO_COMPACT" mode that causes compact tables to appear as if the compact flags were already dropped, but only for the current session.
• If you programmatically set the shuffle parameter, you must change the code for applications that use `conf.set("spark.shuffle.service.port", port)`. Instead, use `dse spark-submit` which automatically sets the correct service port based on the authentication state. See Configuring Spark for more information.

• If DSEFS is enabled, copy CFS hivemetastore directory to dse:

```bash
$ DSE_HOME/bin/dse hadoop fs -cp cfs://127.0.0.1/user/spark/warehouse/ dsefs://127.0.0.1/user/spark/warehouse/
```

After upgrade is complete migrate Spark SQL tables (if used) to the new Hive metastore format:

```bash
$ dse client-tool spark metastore migrate --from 5.0.0 --to 6.0.0
```

• Cassandra File System (CFS) is removed. Remove the `cfs` and `cfs_archive` keyspaces before upgrading. See the From CFS to DSEFS blog post and the Copying data from CFS to DSEFS documentation for more information.

12. DSE Search nodes:

• DSE Search in DataStax Enterprise 6.0 uses Apache Solr™ 6.0. Complete all of the steps in Advanced preparation for upgrading DSE Search and SearchAnalytics nodes (page 92).

• Ensure all use of HTTP writes are changed to use CQL commands for updates and inserts.

• Edit the search index config and make these changes, as needed. See Search index config for valid options to change query behavior for search indexes.

  # Remove the unsupported dataDir option. You can still set the location of search indexes.

  # Remove mergePolicy, maxMergeDocs, and mergeFactor. For example:

  ```xml
  <mergeFactor>25</mergeFactor>
  <maxMergeDocs>...
  <mergePolicy>...
  ```

  Use mergePolicyFactory instead, and add mergeScheduler:

  ```xml
  <mergeScheduler
    class="org.apache.lucene.index.ConcurrentMergeScheduler">
    <int name="maxThreadCount">16</int>
    <int name="maxMergeCount">32</int>
  </mergeScheduler>

  ...  
  <mergePolicyFactory
    class="org.apache.solr.index.TieredMergePolicyFactory">
    <int name="maxMergeAtOnce">10</int>
    <int name="segmentsPerTier">10</int>
  </mergePolicyFactory>
  ```

  # Remove any instance of ExtractingRequestHandler.
# Remove DSENRTCachingDirectoryFactory. Change:

```xml
<directoryFactory name="DirectoryFactory"
class="com.datastax.bdp.search.solr.DSENRTCachingDirectoryFactory"/>
```

to:

```xml
<directoryFactory name="DirectoryFactory"
class="solr.StandardDirectoryFactory"/>
```

- Ensure that the `catalina.properties` and `context.xml` files are present in the Tomcat `conf` dir. DSE will not start after upgrade if these files are missing.

The default location of the Tomcat `conf` directory depends on the type of installation:

- Package installations: `/etc/dse/tomcat/conf`
- Tarball installations: `installation_location/resources/tomcat/conf`

- If earlier DSE versions use a custom configuration for the Solr UI `web.xml`, change:

```xml
<filter-class>com.datastax.bdp.search.solr.auth.DseAuthenticationFilter</filter-class>
```

to

```xml
<filter-class>com.datastax.bdp.cassandra.auth.http.DseAuthenticationFilter</filter-class>
```

- StallMetrics MBean is removed. Change operators that use the MBean.

**13. DSE Graph nodes:**

- If your graph nodes have search indexes that you added with gremlin, complete the steps in Advanced preparation for upgrading DSE Graph nodes with search indexes *(page 97).*
- Ensure that edge label names and property key names use only the supported characters. Edge label names and property key names allow only `[a-zA-Z0-9]`, underscore, hyphen, and period. In earlier versions, edge label names and property key names allowed nearly unrestricted Unicode.
  
  ```
  # schema.describe() displays the entire schema, even if it contains illegal names.
  # In-place upgrades allow existing schemas with invalid edge label names and property key names.
  # Schema elements with illegal names cannot be updated or added.
  ```

**14.** Back up the configuration files you use to a folder that is not in the directory where you normally run commands.
The configuration files are overwritten with default values during installation of the new version.

15. **Upgrades from 5.0.0 to 5.0.8** and from **DSE 5.1.0 and 5.1.1 to DSE 5.1.2 and later releases**

The messaging protocol version in DSE 5.1.2 has been changed to VERSION_3014. Schema migrations rely on exact messaging protocol versions. To accommodate schema changes that might occur during the upgrade, force a backward compatible messaging protocol.

**Before** you upgrade, restart the node with this start-up parameter:

```
-Dcassandra.force_3_0_protocol_version=true
```

For example:

```
$ installation_location/bin/dse cassandra -Dcassandra.force_3_0_protocol_version=true
```

**Note:** While mixed versions exist during the upgrade, do not add or remove columns from existing tables.

**After** the upgrade is complete on all nodes, restart nodes without this flag.

**Upgrade steps**

To upgrade from DSE 5.0 to DSE 6.0, follow these steps on each node in the recommended order. The upgrade process requires upgrading and restarting one node at a time.

**Note:** These steps are performed in your upgraded version and use DSE 6.7 documentation.

**Note:** You can ignore some warning messages that are displayed (page 110) during and after upgrade.

1. **DSE Analytics nodes:** Kill all Spark worker processes.

2. To flush the commit log of the old installation:

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

This step saves time when nodes start up after the upgrade and prevents DSE Search nodes from having to reindex data.
**Important:** This step is mandatory when upgrading between major Cassandra versions that change SSTable formats, rendering commit logs from the previous version incompatible with the new version.

3. Stop the node. See [Stopping a DataStax Enterprise node](#).
   - To stop DataStax Enterprise running as a service:
     
     ```
     $ sudo service dse stop
     ```
   - To stop DataStax Enterprise running as a stand-alone process:
     
     ```
     $ bin/dse cassandra-stop
     ```

4. Use the appropriate method to install the new product version on a supported platform:
   - Package installer using YUM
   - Package installer using APT
   - Binary tarball installer
   
   **Note:** Install the new product version using the same installation type that is on the system, otherwise problems might result.

5. 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.

      ```
      # DSE Search nodes
      # While the node is down, edit dse.yaml and remove these options:
      - cql_solr_query_executor_threads
      - enable_back_pressure_adaptive_nrt_commit
      - max_solr_concurrency_per_core
      - solr_indexing_error_log_options
      DSE 6.0 will not start with these options present.
      
      # DSE Analytics nodes
      Note: Although DSEFS is enabled by default in DSE 5.1.0 and later, the dsefs.enabled setting is commented out in dse.yaml. To enable DSEFS, uncomment the dsefs_options.enabled setting. (DSP-13310)
      - The upgrade installs a new server.xml for Tomcat 8. If your existing server.xml has custom connectors, migrate those connectors to the new server.xml before starting the upgraded nodes.
      ```
Upgrading DataStax Enterprise

- Be sure you are familiar (page 99) with the Apache Cassandra and DataStax Enterprise changes and features in the new release.
- Ensure that keyspace replication factors are correct for your environment:
  # Check the keyspace replication factor for analytics keyspaces.
  # Check the keyspace replication factor for system_auth and dse_security keyspaces.

6. **DSE Analytics nodes:** If your DSE 5.0 clusters had any datacenters running in Analytics Hadoop mode and if the DseSimpleSnitch was used, you must do one of these:
   - For nodes in the datacenters running in Analytics Hadoop mode, start those nodes in Spark mode.
   - Add the special start-up parameter `-Dcassandra.ignore_dc=true` for each node, then start in cassandra mode. This flag is required only once after upgrading. Subsequent restarts do not use this flag. You can leave the flag in the configuration file or remove it after the first restart of each node.

7. Start the node.
   - Package installations: See Starting DataStax Enterprise as a service.
   - Tarball installations: See Starting DataStax Enterprise as a stand-alone process.

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

```
$ nodetool status
```

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

   Warnings, errors, and exceptions are frequently found in the logs when starting 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.

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

11. When the upgrade includes a major Cassandra version, you must upgrade the SSTables. DataStax recommends upgrading the SSTables on one node at a time or when using racks, one rack at a time.

   **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.

```
$ nodetool upgradesstables
```
If the SSTables are already on the current version, the command returns immediately and no action is taken. See SSTable compatibility and upgrade version.

Use the \texttt{--jobs} option to set the number of SSTables that upgrade simultaneously. The default setting is 2, which minimizes impact on the cluster. Set to 0 to use all available compaction threads. DataStax recommends running the \texttt{upgradesstables} command on one node at a time or when using racks, one rack at a time.

\textbf{Note:} You can run the \texttt{upgradesstables} command before all the nodes are upgraded as long as you run this command on only one node at a time or when using racks, one rack at a time. Running \texttt{upgradesstables} on too many nodes will degrade performance.

Recovery after upgrading to DSE 6.0 without dropping compact storage

DSE 6.0 removed support for Thrift-compatible tables (Compact Storage). All tables using Compact Storage must be dropped or migrated to CQL table format before upgrading to DSE 6.0. If a cluster has been upgraded to DSE 6.0 and any Compact Storage tables still exist, follow this procedure to recover and proceed with the upgrade:

1. Downgrade any nodes which were already upgraded to DSE 6.0 to the latest version in the DSE 5.0 or 5.1 series:
   - DSE 5.0.x, downgrade to 5.0.15 or later
   - DSE 5.1.x, downgrade to 5.1.15 or later

2. On each node that was attempted to be started on DSE 6.0, start DSE with the \texttt{-Dcassandra.commitlog.ignorereplayerrors=true} option.

3. On one node (any node) in the cluster, \texttt{DROP COMPACT STORAGE} from tables which use it. Modifying the system keyspace is not supported; modification attempts generate an error.

4. Restart DSE to continue the upgrade to DSE 6.0.

After the upgrade

After all nodes are upgraded and running on DSE 6.0, complete these steps:

1. If you use the OpsCenter Repair Service, turn on the Repair Service.

2. After all nodes are on DSE 6.0 and the required schema change occurs, the new audit logging feature (CassandraAuditWriter) enables the use of new columns.

3. Drop the following legacy tables, if they exist: system_auth.users, system_auth.credentials, and system_auth.permissions.
As described in General upgrade advice, authentication and authorization subsystems now support role-based access control (RBAC).

4. Review your security configuration. To use security, enable and configure DSE Unified Authentication.

In cassandra.yaml, the default authenticator is DseAuthenticator and the default authorizer is DseAuthorizer. Other authenticators and authorizers are no longer supported. Security is disabled in dse.yaml by default.

5. TimeWindowCompactionStrategy (TWCS) is set only on new dse_perf tables. Manually change dse_perf tables that were created in earlier releases to use TWCS. For example:

```
ALTER TABLE dse_perf.read_latency_histograms WITH
COMPACTION={'class':'TimeWindowCompactionStrategy'};
```

6. DSE Search only:

- The appender SolrValidationErrorAppender and the logger SolrValidationErrorLogger are no longer used and may safely be removed from logback.xml.
- In contrast to earlier versions, DataStax recommends accepting the new default value of 1024 for back_pressure_threshold_per_core in dse.yaml. See Configuring and tuning indexing performance.
- If SpatialRecursivePrefixTreeFieldType (RPT) is used in the search schema, replace the units field type with a suitable (degrees, kilometers, or miles) distanceUnits, and then verify that spatial queries behave as expected.
- The appender SolrValidationErrorAppender and the logger SolrValidationErrorLogger are no longer used and may safely be removed from logback.xml.
- Applies only if you are using HTTP writes with JSON documents (deprecated), a known issue causes auto generated solrconfig.xml to have invalid requestHandler for JSON core creations after upgrade to 5.1.0. Change the auto generated solrconfig.xml:

```
<requestHandler name="/update/json"
    class="solr.UpdateUpdateRequestHandler" startup="lazy"/>
```

to

```
<requestHandler name="/update/json"
    class="solr.UpdateRequestHandler" startup="lazy"/>
```

- Slow startup on nodes with large encrypted indexes is resolved. However, action is required to realize the performance gains. You must do a full reindex of all encrypted search indexes on each node in your cluster. Plan sufficient time after the upgrade is complete to reindex with deleteAll=true in a rolling fashion. For example:
7. DSEFS only:

A new schema is available for DSEFS.

**Warning:** Dropping a keyspace is not recoverable without a backup. If you have non-temporary data, do not drop the dsefs keyspace. No action is required. DSEFS will continue to work using the DSE 5.0 schema.

If you have no data in DSEFS or if you are using DSEFS only for temporary data, follow these steps to use the new schema:

a. Stop DSEFS on all nodes. In the `dsefs_options` section of `dse.yaml`, set `enabled: false`.

b. Restart the DSE node.

c. Drop the dsefs keyspace:

   ```bash
   $ DROP KEYSPACE dsefs
   ```

d. Clear the dsefs data directories on each node.

   For example, if the `dsefs_options` section of `dse.yaml` has `data_directories` configured as:

   ```yaml
   dsefs_options:
     ...
     data_directories:
       - dir: /var/lib/dsefs/data
   ```

   This command removes the directories:

   ```bash
   $ rm -r /var/lib/dsefs/data/*
   ```

e. Start DSEFS with DSE 6.0 to use the new schema.

f. If you backed up existing DSEFS data before the upgrade, copy the data back into DSEFS from local storage.

8. DSE Analytics only:

- Spark Jobserver uses DSE custom version 0.8.0.44. Applications must use the compatible Spark Jobserver API from the DataStax repository.
- If you are using Spark SQL tables, migrate them to the new Hive metastore format:
$ dse client-tool spark metastore migrate --from 5.0.0 --to 6.0.0

9. Ensure that keyspace replication factors are correct for your environment:
   - Check the keyspace replication factor for analytics keyspaces.
   - Check the keyspace replication factor for system_auth and dse_security keyspaces.

Warning messages during and after upgrade

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

- When upgrading nodes with DSE Advanced Replication, there might be some WriteTimeoutExceptions during a rolling upgrade while mixed versions of nodes exist. Some write consistency limitations apply while mixed versions of nodes exist. The WriteTimeout issue is resolved after all nodes are upgraded.
- Some gremlin_server properties in earlier versions of DSE are no longer required. If properties exist in the dse.yaml file after upgrading, logs display warnings similar to:

```
WARN [main] 2017-08-31 12:25:30,523 GREMLIN
DseWebSocketChannelizer.java:149 - Configuration for the org.apache.tinkerpop.gremlin.driver.ser.GraphSONMessageSerializerGremlinV1d0 serializer in dse.yaml overrides the DSE default - typically it is best to allow DSE to configure these.
```

You can ignore these warnings or modify dse.yaml so that only the required gremlin server properties are present.

Error messages provide information to help identify problems.

- If you see an error message like:

```
ERROR [main] 2016-07-21 13:52:46,941 CassandraDaemon.java:737 - Cannot start node if snitch's data center (Cassandra) differs from previous data center (Analytics). Please fix the snitch configuration, decommission and rebootstrat this node or use the flag -Dcassandra.ignore_dc=true.
```

Follow upgrade instructions in 6 (page 106). You must start in Spark mode or add the special start-up parameter -Dcassandra.ignore_dc=true for each node.

Upgrading to DSE 6.0 or DSE 6.7 from DataStax Installer installations

The DataStax Installer is not available for DataStax Enterprise (DSE) 6.0 and later. Use this strategy to upgrade to DSE 6.0 or DSE 6.7 when DSE 5.0 or DSE 5.1 was installed using the DataStax Installer.
This strategy advises only the high-level steps for converting from DataStax Installer installations. Be sure to read all of the upgrade documentation, starting with *Planning your DataStax Enterprise upgrade (page 9)*. Contact DataStax Support for questions.

**Attention: Read and understand these instructions before upgrading.** Carefully reviewing the planning *(page 9)* and upgrading instructions can ensure a smooth upgrade and avoid pitfalls and frustrations.

This upgrade strategy has two required steps:

1. Migrate from the standalone installer installation to tarball or package installation for the same DSE version.

2. Upgrade from that DSE version to your target DSE version following the DSE upgrade documentation.

**Prerequisites:** For DSE 5.0 or DSE 5.1 installations, you must determine if the DataStax Installer was used to create:

- **Services installation**
  
  Typically when the DSE installer was run with root permissions. With a services installation, you start DSE using the `service dse start` command.

  **Tip:** For the smoothest transition from a services installation, DataStax recommends the package install method. When installed from a package (Yum or APT), DataStax Enterprise runs as a service.

- **No-services installation**
  
  When the DSE installer was run without root permissions or with custom directories. With a no-services installation, you start DSE with the `dse` command.

  **Tip:** For the smoothest transition from a no-services installation, DataStax recommends the tarball install method. When installed from a tarball, DataStax Enterprise runs as a stand-alone process.

The installation method used determines the migration steps.

**Reduce risks**

You can reduce risks and effort by employing a continual upgrade strategy to provide access to product improvements and new features and reduce version impacts. Ensure that you repair your nodes regularly. Node repair ensures that data on a replica is consistent with data on other nodes.

**Perform this migration on each node in the cluster**

In production clusters where service cannot be interrupted, perform this migration in a rolling fashion. Ensuring that each node is healthy and able to service queries for existing data before proceeding to the next node.

1. Verify the current DSE version.
$ dse -v

**Important:** You must install the exact patch version to successfully complete the migration.

2. Back up the configuration files.

3. Back up your data. DataStax recommends backing up your data prior to any version upgrade, including logs and custom configurations. A backup provides the ability to revert and restore all the data used in the previous version if necessary.

   **Tip:** OpsCenter provides a [Backup Service](https://docs.datastax.com/en/opscenter/6.5/opscenter/opscenter-backups.html) that manages enterprise-wide backup and restore operations for DataStax Enterprise clusters. OpsCenter 6.5 and later is recommended.

4. Use the DataStax Installer to uninstall your current version:
   - DSE 5.1 uninstall
   - DSE 5.0 uninstall

   **Important:** When prompted, be sure to keep the existing data and configuration files in place.

5. Use the appropriate steps for converting your existing DSE installation:
   - Converting to Yum installation on RHEL-based systems *(page 112)*
   - Converting to APT installation on Debian-based systems *(page 113)*
   - Converting to binary tarball installation *(page 114)*

**Converting to Yum installation on RHEL-based systems**

The DataStax Installer is not available for DSE 6.0 and later. Follow these steps to change an existing DSE services installation that was created using the DataStax Installer to a Yum-based package installation.

**Prerequisites:** The packages you install must match the existing DSE patch release version that you verified in 1 *(page 111)* in **Upgrading to DSE 6.0 or DSE 6.7 from DataStax Installer installations *(page 110)*.

1. Move the existing DSE Installer backup directory to another location:

   ```
   $ cd /usr/share/dse
   $ sudo mv backups backups.old
   ```
2. Configure and install DSE at the same patch version that you verified in 1 (page 111) in Upgrading to DSE 6.0 or DSE 6.7 from DataStax Installer installations (page 110). Because you are keeping the existing configuration files in place when you used the DataStax Installer to uninstall your current version, the Yum installation respects those files and they will be used by the package installed version of DSE. Use the installation instructions for your DSE version:
   - Install DSE 5.1 on RHEL-based systems using Yum
   - Install DSE 5.0 on RHEL-based systems using Yum

For example, to install DSE 5.1.11:

```
$ sudo install dse-full=5.1.11-1
dse=5.1.11-1 dse-libso1r=5.1.11-1 dse-libtomcat=5.1.11-1
dse-liblog4j=5.1.11-1 dse-libcassandra=5.1.11-1 dse-
libspark=5.1.11-1
dse-libgraph=5.1.11-1 dse-libhadoop2-client-native=5.1.11-1
dse-libhadoop2-client=5.1.11-1
```

3. Repeat these steps on each remaining node.

**Converting to APT installation on Debian-based systems**

The DataStax Installer is not available for DSE 6.0 and later. Follow these steps to change an existing DSE services installation that was created using the DataStax Installer to a APT-based package installation.

**Prerequisites:** The packages you install must match the existing DSE patch release version that you verified in 1 (page 111) in Upgrading to DSE 6.0 or DSE 6.7 from DataStax Installer installations (page 110).

1. Move the old DSE Installer backup directory to another location:

```
$ cd /usr/share/dse
$ sudo mv backups backups.old
```

2. Configure and install DSE at the same patch version that you verified in 1 (page 111). Use the version-appropriate installation instructions but add `Dpkg` options to the `apt-get install` step.

For example, to install DSE 5.1.11:

```
$ sudo apt-get --force-confdef --force-confold install dse-full=5.1.11-1
dse=5.1.11-1 dse-libso1r=5.1.11-1 dse-libtomcat=5.1.11-1
dse-liblog4j=5.1.11-1 dse-libcassandra=5.1.11-1 dse-
libspark=5.1.11-1
dse-libgraph=5.1.11-1 dse-libhadoop2-client-native=5.1.11-1
dse-libhadoop2-client=5.1.11-1
```
This change keeps the existing configuration files in place and prevents them from being overwritten by the new packages.

- Install DSE 5.1 on Debian-based systems using APT
- Install DSE 5.0 on Debian-based systems using APT

Be sure to list all packages with the same version.

3. Repeat these steps on each remaining node.

**Converting to binary tarball installation**

The DataStax Installer is not available for DSE 6.0 and later. Follow these steps to change an existing DSE no-services installation that was created using the DataStax Installer to a binary tarball installation.

**Prerequisites:** The version you install must match the existing DSE patch release version that you verified in 1 (page 111) in Upgrading to DSE 6.0 or DSE 6.7 from DataStax Installer installations (page 110).

1. **DSE Analytics nodes:** Re-create the Spark RDD directory, adjusting the base direction as appropriate:

   ```bash
   $ mkdir ~/dse/spark/rdd
   ``

2. Download DSE at the same patch version that you verified in 1 (page 111) in Upgrading to DSE 6.0 or DSE 6.7 from DataStax Installer installations (page 110). Use the installation instructions for your DSE version:
   - Install DSE 5.1 using the binary tarball
   - Install DSE 5.0 using the binary tarball

3. Extract the tarball into a different directory:
   For example, to extract the DSE 5.1.11 tarball to /tmp/dse511tar:
   ```bash
   $ tar xvzf dse-5.1.11-bin.tar.gz -C /tmp/dse511tar
   ``

4. To configure the newly installed DSE version (same as your original version):
   - **a.** Compare your backup configuration files to the new tarball configuration files:
     - Look for any deprecated, removed, or changed settings.
     - Transfer customizations from the no-services installation of your current DSE version into the tarball installation.
• Be sure to use the same directory locations as set in the original version.

5. Repeat these steps on each remaining node.

6. Start the DSE cluster and verify proper operation for the tarball DSE installation.

7. After all nodes are confirmed to be operating, the tarball installation is eligible for upgrade (page 9) to the target DSE 6.0 or 6.7 version.

### Upgrading from DataStax Enterprise 5.0 to 5.1

Follow these instructions to upgrade to from DataStax Enterprise 5.0 to 5.1. If you have an earlier version of DSE, upgrade to 5.0 (page 133) before continuing.

Always upgrade to latest patch release (page 10) on your current version before you upgrade to a higher version. Fixes included in the latest patch release might help or smooth the upgrade process.

The latest version of DSE 5.1 is 5.1.15.

**Warning:** TTL expiration timestamps are susceptible to the year 2038 problem. If the TTL value is long and an expiration date that is greater than the maximum threshold of 2038-01-19T03:14:06+00:00, the data is immediately expired and purged on the next compaction. DataStax strongly recommends upgrading to DSE 5.1.7 or later and taking required action to protect against silent data loss. (DSP-15412).

**Attention:** Read and understand these instructions before upgrading. Carefully reviewing the planning (page 9) and upgrading instructions can ensure a smooth upgrade and avoid pitfalls and frustrations.

### Apache Cassandra™ version change

Upgrading from DataStax Enterprise 5.0 to 5.1 includes a major Cassandra version change.

- DataStax Enterprise 5.1 uses Cassandra 3.11.
- DataStax Enterprise 5.0 uses Cassandra 3.0.

Be sure to follow the recommendations for upgrading the SSTables.

### General recommendations

DataStax recommends backing up your data prior to any version upgrade, including logs and custom configurations. 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. OpsCenter 6.5 and later is recommended.
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 restrictions and limitations during the upgrade process**

- **Do not** enable new features.
- Do not run nodetool repair. If you have the OpsCenter Repair Service configured, turn off the Repair Service.
- Ensure OpsCenter compatibility. See DataStax OpsCenter compatibility with DataStax Enterprise.
- During the upgrade, do not bootstrap or decommission nodes.
- Do not issue these types of CQL queries during a rolling restart: DDL and TRUNCATE.
- Do not enable Change Data Capture (CDC) on a mixed-version cluster. Upgrade all nodes to DSE 5.1 or later before enabling CDC.
- 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 when upgrading from DSE 5.0.10 and later**

**Attention:** Upgrades to DSE versions earlier than 5.1.3 are not recommended. See below (page 116).

**Restrictions when upgrading to DSE 5.1.3**

- Creating Materialized View with filtering on non-primary-key base column (CASSANDRA-10368) is disabled, because the liveness of view row is depending on multiple filtered base non-key columns and base non-key column used in view primary-key. This semantic cannot be supported without storage format change (CASSANDRA-13826). For append-only use case, you can still use this feature with a system property switch:
  
  ```
  -Dcassandra.mv.allow_filtering_nonkey_columns_unsafe=true
  ```

- The table system_auth.resource_role_permissions_index is no longer used and must be dropped after all nodes are on 5.1.3.
- Non-incremental full repairs are now the default if no option is specified on nodetool repair, unless incremental repair was already run on the table/keyspace being repaired, to maintain backward compatibility. To run incremental repair on new tables, use the #inc option.

**Restrictions for DSE Analytic (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 DSEFS
Mixed versions are not supported during the upgrade process to DSE 5.1.3 and later.

- Complete the upgrade on all nodes before running fsck. Starting with DSE 5.1.3, all nodes must be able to report proper block status to the node running fsck. If you run fsck on an upgraded node in a mixed version cluster, nodes with versions earlier than DSE 5.1.3 do not properly report block status and cause the fsck to incorrectly assume that data is corrupt or unavailable. The fsck will incorrectly try to repair them.
- A protocol change in DSE 5.1.3 improves efficiency of passing JSON arrays between DSEFS server and client. Upgrade all nodes in the cluster before using the DSEFS shell.

Restrictions for DSE Graph
Graph nodes have the same restrictions as the workload they run on. General graph restrictions apply for all nodes, such as not altering graph schema during upgrades. Workload-specific restrictions apply for analytics and search nodes, such as no OLAP queries during upgrades.

Restrictions for DSE Search
DSE Search in DataStax Enterprise 5.1 uses Apache Solr 6.0. This significant change requires advanced planning and specific actions before and after the upgrade.

**Important:** Before you upgrade DSE Search or SearchAnalytics workloads, you must follow the specific steps in Advanced preparation for upgrading DSE Search and SearchAnalytics nodes *(page 118)*.

Restrictions for nodes using any kind of security

- Do not change security credentials or permissions until the upgrade is complete on all nodes.
- If you are not already using Kerberos, do not set up Kerberos authentication before upgrading. First upgrade the cluster, and then set up Kerberos.

Security changes
After upgrading to DSE 5.1, the default authenticator is DseAuthenticator and the default authorizer is DseAuthorizer in cassandra.yaml. Other authorizers and authenticators are no longer supported, follow steps in After the upgrade *(page 130)*.

**Note:** After the upgrade, DataStax recommends that the authenticator setting in cassandra.yaml is DseAuthenticator.

Although the internal authentication schema is compatible with PasswordAuthenticator, the advanced security features provided with DSE Unified Authentication are not supported.

Upgrading drivers and possible impact when driver versions are incompatible
Be sure to check driver compatibility. Depending on the driver version, you might need to recompile your client application code. See DataStax driver changes.
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:

- **Protocol version:** Because some drivers can use different protocol versions, force the protocol version at start up. For example, keep the Java driver at its current protocol version while the driver upgrade is happening. Switch to the Java driver to the new protocol version only after the upgrade is complete on all nodes in the cluster.

- **Initial contact points:** 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 details on protocol version negotiation, see protocol versions with mixed clusters in the Java driver version you're using, for example, Java driver.

Advanced preparation for upgrading DSE Search and SearchAnalytics nodes

Before starting the Preparing to upgrade (page 125) steps, complete all the advanced preparation steps on DSE Search and SearchAnalytics nodes while DSE 5.0 is still running.

Plan sufficient time to implement and test the required changes before the upgrade:

- Schema changes require a full reindex.
- Configuration changes require reloading the core.

1. If any Solr core was created on DSE 4.6 or earlier and never reindexed after being upgraded to DSE 4.7 or later, you must reindex on DSE 5.0 before upgrading to DSE 5.1.

2. Ensure that the `shard_transport_options` in dse.yaml are set only for `netty_client_request_timeout`:

   ```yaml
   shard_transport_options:
     netty_client_request_timeout: 60000
   ```

   In DSE 5.1, the shard transport option supports only `netty_client_request_timeout`. Remove any other `shard_transport_options`.

3. Ensure that the `catalina.properties` and `context.xml` files are present in the Tomcat `conf` dir. DSE will not start after upgrade if these files are missing.

   The default location of the Tomcat `conf` directory depends on the type of installation:
   - **Package installations:** /etc/dse/tomcat/conf
   - **Tarball installations:** installation_location/resources/tomcat/conf

4. If you are using Apache Solr SolrJ, the minimum required version is 6.0.0.

5. For SpatialRecursivePrefixTreeFieldtype (RPT) in search schemas, you must adjust your queries for these changes:
- **IsDisjointTo** is no longer supported in queries on `SpatialRecursivePrefixTreeFieldtype`. Replace `IsDisjointTo` with a NOT Intersects query. For example:

```plaintext
foo:0,0 TO 1000,1000 AND -"Intersects(POLYGON((338 211, 338 305, 404 305, 404 211, 338 211)))"
```

- The `ENVELOPE` syntax is now required for WKT-style queries against `SpatialRecursivePrefixTreeFieldtype` fields. You must specify `ENVELOPE(10, 15, 15, 10)`, where queries on earlier releases could specify `10 10 15 15`.

See [Spatial Search](#) for details on using distanceUnits in spatial queries.

6. Stored=true copy fields are not supported and cause schema validation to fail. The `stored=true` copyField directive has not been supported since DSE 4.7, so you probably do not have Stored=true copy fields. If you do:

   - Change the stored attribute value of all copyField directives from true to false in the `schema.xml` file and then use `dsetool reload_core` to reload the modified schema.
   - You must ensure that application designs and implementations recognize this change.

7. Edit the `solrconfig.xml` file and make these changes, as needed:

   - Remove these requestHandlers: `XmlUpdateRequestHandler`, `BinaryUpdateRequestHandler`, `CSVRequestHandler`, `JsonUpdateRequestHandler`, `DataImportHandler`. Solr deprecated and then removed these requestHandlers. For example:

     ```xml
     <requestHandler name="/dataimport" class="solr.DataImportHandler"/>
     or
     <requestHandler name="/update"
                   class="solr.XmlUpdateRequestHandler"/>
     ```

   - Change the directoryFactory from:

     ```xml
     <directoryFactory name="DirectoryFactory"
                      class="${solr.directoryFactory:solr.StandardDirectoryFactory}"/>
     ```

     to

     ```xml
     <directoryFactory name="DirectoryFactory"
                      class="solr.StandardDirectoryFactory"/>
     ```

   - `<unlockOnStartup>` is now unsupported as a result of `LUCENE-6508` and `SOLR-7942`.
   - Change the updateLog from:
Upgrading DSE search nodes to DSE 5.1 requires replacing unsupported Solr types with supported types.

**Note:** Special handling is also required for BCDStrField, addressed in 10 (page 122).

Some of the removed Solr types, due to the way they marshal sort values during distributed queries (combined with the way the suggested new types unmarshal sort values), cannot be sorted on during rolling upgrades when some nodes use an unsupported type and other nodes use the suggested new type. The following type transitions are problematic:

<table>
<thead>
<tr>
<th>Removed Solr field types</th>
<th>Supported Solr field types</th>
</tr>
</thead>
<tbody>
<tr>
<td>ByteField</td>
<td>TrieIntField</td>
</tr>
<tr>
<td>DateField</td>
<td>TrieDateField</td>
</tr>
<tr>
<td>BCDIntField</td>
<td>TrieIntField</td>
</tr>
<tr>
<td>BCDLongField</td>
<td>TrieLongField</td>
</tr>
</tbody>
</table>

Two options are available:

**a.** Avoid sorting on removed Solr field types until the upgrade to DSE 5.1 is complete for all nodes in the datacenter being queried.

**Tip:** When using two search datacenters, isolate queries to a single datacenter and then change the schema and reindex the other datacenter. Then isolate queries to the newly reindexed datacenter while you change the schema and upgrade the first datacenter.

**b.** If you are using BCDIntField or BCDLongField, update the schema to replace BCDIntField and BCDLongField with types that are sort-compatible with the supported Solr types TrieIntField and TrieLongField:

<table>
<thead>
<tr>
<th>Removed Solr field types</th>
<th>Interim sort-compatible supported Solr field types</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCDIntField</td>
<td>SortableIntField</td>
</tr>
<tr>
<td>BCDLongField</td>
<td>SortableLongField</td>
</tr>
</tbody>
</table>

Change the schema in a distributed fashion, and do not reindex. After the schema is updated on all nodes, then go to 9 (page 121).
9. Update the schema and configuration for the Solr field types that are removed from Solr 5.5 and later.

   • Update the schema to replace unsupported Solr field types with supported Solr field types:

<table>
<thead>
<tr>
<th>Removed Solr field types</th>
<th>Supported Solr field types</th>
</tr>
</thead>
<tbody>
<tr>
<td>ByteField</td>
<td>TrieIntField</td>
</tr>
<tr>
<td>DateField</td>
<td>TrieDateField</td>
</tr>
<tr>
<td>DoubleField</td>
<td>TrieDoubleField</td>
</tr>
<tr>
<td>FloatField</td>
<td>TrieFloatField</td>
</tr>
<tr>
<td>IntField</td>
<td>TrieIntField</td>
</tr>
<tr>
<td>LongField</td>
<td>TrieLongField</td>
</tr>
<tr>
<td>ShortField</td>
<td>TrieIntField</td>
</tr>
<tr>
<td>SortableDoubleField</td>
<td>TrieDoubleField</td>
</tr>
<tr>
<td>SortableFloatField</td>
<td>TrieFloatField</td>
</tr>
<tr>
<td>SortableIntField</td>
<td>TrieIntField</td>
</tr>
<tr>
<td>SortableLongField</td>
<td>TrieLongField</td>
</tr>
<tr>
<td>BCDIntField</td>
<td>TrieIntField</td>
</tr>
<tr>
<td>BCDLongField</td>
<td>TrieLongField</td>
</tr>
<tr>
<td>BCDStrField (see 10 (page 122) if used)</td>
<td>TrieIntField</td>
</tr>
</tbody>
</table>

   • If you are using type mapping version 0, or you do not specify a type mapper, verify or update the solrconfig.xml to use dseTypeMappingVersion 1:

   `<dseTypeMappingVersion>1</dseTypeMappingVersion>`

   If the Solr core is backed by a CQL table and the type mapping is unspecified, use type mapping version 2.

   • Reload the core:

   `dsetool reload_core keyspace_name.table_name schema=filepath solrconfig=filepath`

   If you were using the unsupported data types, do a full reindex node-by-node:

   `dsetool reload_core keyspace_name.table_name schema=filepath solrconfig=filepath reindex=true deleteAll=true distributed=false`

   **Note:** In DSE 5.1, auto generated schemas use data type mapper 2.
10. If using BCDStrField: In DSE 5.0 and earlier, DSE mapped Cassandra text columns to BCDStrField. The BCDStrField was deprecated and is removed in DSE 5.1.0.

The recommended strategy is to upgrade the data type to TrieIntField. However, DSE cannot map text directly to TrieIntField. If you are using BCDStrField, you must complete one of these options before the upgrade.

a. If BCDStrField is no longer used, remove the BCDStrField field from the Solr schema. Reindexing is not required.

b. If you want to index the field as a TrieIntField, and a full reindex is acceptable, change the underlying database column to use the type int.

c. If you want to keep the database column as text and you still want to do simple matching queries on the indexed field, switch from BCDStrField to StrField in the schema. Indexing should not be required, but the field will no longer be appropriate for numeric range queries or sorting, because StrField uses a lexicographic order, not a numeric one.

d. Not recommended: If you want to keep the database column as text and still want to perform numeric range queries and sorts on the former BCDStrField, but would rather change their application than perform a full reindex:
   - Change the field to StrField in the Solr schema with indexed=false.
   - Add a new copy field with the type TrieIntField that has its values supplied by the original BCDStrField.

This solution still requires reindex to work, because the copy field target must be populated. This non-recommended option is supplied only to support a sub-optimal data model; for example, a text column with values that would fit only into an int.

After you make these schema changes, do a rolling, node-by-node dsetool reload_core with reindex=true, distributed=false, and deleteAll=true.

Note: If you have two datacenters and upgrade them one at a time, reload the core with distributed=true deleteAll=true.

11. Tune the schema before you upgrade. For DSE 5.1.4 and later, all field definitions in the schema are validated and must be DSE Search compatible, even if the fields are not indexed, have docValues applied, or used for copy-field source. The default behavior of automatic resource generation includes all columns. To improve performance, take action to prevent the fields from being loaded from the database. Include only the required fields in the schema by removing or commenting out unused fields in the schema.
Advanced preparation for upgrading DSE Graph nodes with search indexes

These steps apply to graph nodes that have search indexes. Before starting the Preparing to upgrade (page 125) steps, complete these advanced preparation steps while DSE 5.0 is still running.

Upgrading DSE Graph nodes with search indexes requires these edits to the solrconfig file. Configuration changes require reloading the core. Plan sufficient time to implement and test changes that are required before the upgrade.

Edit the solrconfig.xml file and make these changes, as needed:

- Remove these requestHandlers: XmlUpdateRequestHandler, BinaryUpdateRequestHandler, CSVRequestHandler, JsonUpdateRequestHandler, and DataImportHandler. Solr deprecated and then removed these requestHandlers.

  For example:

  ```
  <requestHandler name="/dataimport" class="solr.DataImportHandler"/>
  ```

  or

  ```
  <requestHandler name="/update" class="solr.XmlUpdateRequestHandler"></requestHandler>
  ```

- `<unlockOnStartup>` is now unsupported as a result of LUCENE-6508 and SOLR-7942.

- Reload the core so that this configuration change is respected:

  ```
  $ dsetool reload_core keyspace_name.table_name reindex=false
  ```

Advanced preparation for upgrading DSE Analytics nodes

DSE 5.1 includes a major upgrade to Spark 2.0, as well as a tighter integration between DSE and Spark. For information on Spark 2.0, see the Spark documentation. Spark 2.0 uses Scala 2.11.

DSE 5.1 includes a new Spark resource manager that uses the DataStax Java Driver and native the CQL protocol for managing communication between DSE Spark nodes. DSE 5.0 used Spark RPC. This impacts Spark applications that ran on DSE 5.0.

1. Spark 2.0 uses Scala 2.11. All DSE 5.0 Scala Spark applications will need to be recompiled against Scala 2.11 and only use Scala 2.11 third-party libraries.

   Changing the dse-spark-depencencies in your build files is not sufficient to change the compilation target. See the example projects for how to set up your build files.

2. Spark applications should use dse:// URLs instead of spark://spark master IP:Spark RPC port number URLs, as described in Specifying Spark URLs.
You no longer need to specify the Spark master IP address or hostname when using dse:// URLs. Connecting to any Spark node will redirect the request to the master node.

3. If you have existing Spark application code that uses spark://Spark master IP:Spark RPC port to connect, it will no longer work.

For example the following code worked in DSE 5.0 but will not work in DSE 5.1.

```scala
val conf = new SparkConf(true)
  .setMaster("spark://192.168.123.10:7077")
  .setApp_name("cassandra-demo")
  .set("cassandra.connection.host" , "192.168.123.10") // initial contact
  .set("cassandra.username", "cassandra")
  .set("cassandra.password", "cassandra")
val sc = new SparkContext(conf)
```

To connect to DSE in 5.1, you no longer need to call `setMaster`. This code will work in DSE 5.1:

```scala
val conf = new SparkConf(true)
  .setApp_name("cassandra-demo")
  .set("cassandra.connection.host" , "192.168.123.10") // initial contact
  .set("cassandra.username", "cassandra")
  .set("cassandra.password", "cassandra")
val sc = new SparkContext(conf)
```

If you need to specify the master using `setMaster`, use the dse:// URL format.

4. Starting in DSE 5.1 you can restrict Spark jobs to specific database roles. See Setting up DSE Spark application permissions.

5. Starting in DSE 5.1, you can set the Spark executor process owners, as described in Running Spark processes as separate users.

6. The user submitting the Spark application no longer has to be the same database role. See Managing Spark application permissions for information on how to change the master connection submission to use a different user or cluster than the database connection.

**Backing up DSEFS data**

These steps only apply to nodes that use DSEFS. Before starting the Preparing to upgrade (page 125) steps, complete these advanced preparation steps.

The DSEFS schema used by the database was improved in DSE 5.1, but the old schema is still supported and will not be modified during the upgrade. To use the new DSEFS schema with existing DSEFS data, you must back up the DSEFS data before upgrading:
Back up the current DSEFS data to local storage using `dse hadoop fs -cp` command:

```
$ dse hadoop fs -cp /* /local_backup_location
```

Preparing to upgrade

Follow these steps to prepare each node for upgrading from DataStax Enterprise 5.0 to DataStax Enterprise 5.1.

**Note:** These steps are performed in your current version and use DSE 5.0 documentation.

1. Upgrade to the latest patch release on your current version. Fixes included in the latest patch release can simplify the upgrade process.

2. Before upgrading, be sure that each node has ample free disk space.
   The required space depends on the compaction strategy. See [Disk space](#).

3. Familiarize yourself with the changes and features in this release:
   - DataStax Enterprise 5.1 release notes.
   - General upgrading advice for any version. Be sure to read NEWS.txt all the way back to your current version.
   - DataStax Enterprise changes in CHANGES.txt.
   - DataStax driver changes.

4. Verify your current product version. If necessary, upgrade to an interim version:

<table>
<thead>
<tr>
<th>Current version</th>
<th>Upgrade version</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataStax Enterprise 5.0</td>
<td>DataStax Enterprise 5.1</td>
</tr>
<tr>
<td>DataStax Enterprise 4.7 or 4.8</td>
<td>DataStax Enterprise 5.0</td>
</tr>
<tr>
<td>DataStax Enterprise 4.0, 4.5, or 4.6</td>
<td>DataStax Enterprise 4.8</td>
</tr>
<tr>
<td>DataStax Community or open source Apache Cassandra™ 2.0.x</td>
<td>DataStax Enterprise 4.8</td>
</tr>
<tr>
<td>DataStax Community 3.0.x</td>
<td>DataStax Enterprise 5.1 (page 189)</td>
</tr>
<tr>
<td>DataStax Distribution of Apache Cassandra™ 3.x</td>
<td>DataStax Enterprise 5.0 (page 133)</td>
</tr>
</tbody>
</table>

5. Upgrade the SSTables on each node to ensure that all SSTables are on the current version.

```
$ nodetool upgradesstables
```
This is required for DataStax Enterprise upgrades that include major Cassandra version changes.

**Warning:** Failure to upgrade SSTables when required results in a significant performance impact and increased disk usage. For information about nodetool upgradesstables, including how to speed it up, see the DataStax Support KB article [Nodetool upgradesstables FAQ](#).

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

6. **Verify the Java runtime version and upgrade to the recommended version.**

   $ java -version

   - **Recommended.** **OpenJDK 8** *(1.8.0_151 minimum)*
     
     **Note:** Recommendation changed due to the end of public updates for Oracle JRE/JDK 8. See [Oracle Java SE Support Roadmap](#).

   - **Supported.** **Oracle Java SE 8** *(JRE or JDK)* *(1.8.0_151 minimum)*

   The JDK provides useful troubleshooting tools that are not in the JRE, such as jstack, jmap, jps, and jstat.

   **Important:** Although Oracle JRE/JDK 8 is supported, DataStax does more extensive testing on OpenJDK 8 starting with DSE 5.1.11.

7. Run nodetool repair to ensure that data on each replica is consistent with data on other nodes.

8. **DSE Analytics nodes:** If you programmatically set the shuffle parameter, you must change the code for applications that use `conf.set("spark.shuffle.service.port", port)`. Instead, use `dse spark-submit` which automatically sets the correct service port based on the authentication state. See [Spark configuration](#).

9. **DSE Search nodes:** DSE Search in DataStax Enterprise 5.1 uses Apache Solr 6.0. This significant change requires advanced planning and specific actions before and after the upgrade. Complete all of the steps in [Advanced preparation for upgrading DSE Search and SearchAnalytics nodes](#).

10. **DSE Graph nodes:** If your graph nodes have search indexes that you added with gremlin, complete the steps in [Advanced preparation for upgrading DSE Graph nodes with search indexes](#).

11. Back up the configuration files you use to a folder that is not in the directory where you normally run commands.
The configuration files are overwritten with default values during installation of the new version.

12. Upgrades from 5.0.0 to 5.0.8 and from DSE 5.1.0 and 5.1.1 to DSE 5.1.2 and later DSE 5.1.x releases

The messaging protocol version in DSE 5.1.2 has been changed to VERSION_3014. Schema migrations rely on exact messaging protocol versions. To accommodate schema changes that might occur during the upgrade, force a backward compatible messaging protocol.

**Before** you upgrade, restart the node with this start-up parameter:

```
-Dcassandra.force_3_0_protocol_version=true
```

For example:

```
$ installation_location/bin/dse cassandra -Dcassandra.force_3_0_protocol_version=true
```

**Note:** While mixed versions exist during the upgrade, do not add or remove columns from existing tables.

**After** the upgrade is complete on all nodes, restart nodes without this flag.

Upgrade steps

**Tip:** The DataStax installer *(page 184)* upgrades DataStax Enterprise and automatically performs many upgrade tasks.

Follow these steps on each node to upgrade from DataStax Enterprise 5.0 to DataStax Enterprise 5.1.

**Note:** These steps are performed in your upgraded version and use DSE 6.7 documentation.

**Note:** You can ignore some warning messages that are displayed *(page 143)* during and after upgrade.

1. **DSE Analytics nodes:** Kill all Spark worker processes.

2. To flush the commit log of the old installation:

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

This step saves time when nodes start up after the upgrade and prevents DSE Search nodes from having to reindex data.
Important: This step is mandatory when upgrading between major Cassandra versions that change SSTable formats, rendering commit logs from the previous version incompatible with the new version.

3. Stop the node.

4. Use the appropriate method to install the new product version on a supported platform:
   - Tarball: 5.1
   - Debian: 5.1
   - RHEL: 5.1

   Be sure to read and follow the installation instructions for each installation method.

   Note: Install the new product version using the same installation method that is on the system. The upgrade proceeds with installation regardless of the installation method and might result in issues.

5. 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.

      Note: Although DSEFS is enabled by default in DSE 5.1.0, the dsefs.enabled setting is commented out in the new DSE 5.1.0 dse.yaml file. To enable DSEFS, uncomment the dsefs_options.enabled setting after upgrade to DSE 5.1.0. (DSP-13310)

      - The upgrade installs a new server.xml for Tomcat 8. If your existing server.xml has custom connectors, migrate those connectors to the new DSE 5.1 server.xml before starting the upgraded nodes.

      - Be sure you are familiar (page 125) with the Apache Cassandra and DataStax Enterprise changes and features in the new release.

      - Ensure that keyspace replication factors are correct for your environment:

        # Check the keyspace replication factor for analytics keyspaces.
        # Check the keyspace replication factor for system_auth and dse_security keyspaces.

   b. Merge the applicable modifications into the new version.

6. DSE Analytics nodes:

   DSE 5.1 and later runs analytics nodes in Spark mode. If your DSE 5.0 clusters had any datacenters running in Analytics Hadoop mode and if the DseSimpleSnitch was used, you must do one of these:
• For nodes in the datacenters running in Analytics Hadoop mode, start those nodes in Spark mode.
• Add the special start-up parameter `-Dcassandra.ignore_dc=true` for each node, then start in cassandra mode. This flag is required only once after upgrading to DSE 5.1. Subsequent restarts do not use this flag. You can leave the flag in the configuration file or remove it after the first restart of each node.

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

   ```
   $ nodetool status
   ```

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

   Warnings, errors, and exceptions are frequently found in the logs when starting 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.

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

11. When the upgrade includes a major Cassandra version, you must upgrade the SSTables. DataStax recommends upgrading the SSTables on one node at a time or when using racks, one rack at a time.

   **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.

   ```
   $ nodetool upgradesstables
   ```

   If the SSTables are already on the current version, the command returns immediately and no action is taken. See [SSTable compatibility and upgrade version](#).

   Use the `--jobs` option to set the number of SSTables that upgrade simultaneously. The default setting is 2, which minimizes impact on the cluster. Set to 0 to use all available compaction threads. DataStax recommends running the `upgradesstables` command on one node at a time or when using racks, one rack at a time.
Note: You can run the `upgradesstables` command before all the nodes are upgraded as long as you run this command on only one node at a time or when using racks, one rack at a time. Running `upgradesstables` on too many nodes will degrade performance.

After the upgrade

After all nodes are upgraded and running on DSE 5.1, complete these steps:

1. Review your security configuration. To use security, enable and configure DSE Unified Authentication.
   
   In `cassandra.yaml`, the default authenticator is DseAuthenticator and the default authorizer is DseAuthorizer. Other authenticators and authorizers are no longer supported. Security is disabled in `dse.yaml` by default.

2. **TimeWindowCompactionStrategy** (TWCS) is set only on new `dse_perf` tables. Manually change `dse_perf` tables that were created in earlier releases to use TWCS. For example:
   
   ```
   ALTER TABLE dse_perf.read_latency_histograms WITH COMPACTION={'class':'TimeWindowCompactionStrategy'};
   ```

3. If you use the OpsCenter Repair Service, turn on the Repair Service.

4. **DSE Search only:**
   
   - Index time boost support is removed in DSE 5.1.1 and later. Use query time boosting instead. Delete any `_docBoost` columns in backing CQL tables.
     
     Thrift tables where the `_docBoost` column existed will be allowed, but the `_docBoost` will be ignored. Thrift tables are not able to drop the column.

   - If `SpatialRecursivePrefixTreeFieldType` (RPT) is used in the search schema, replace the units field type with a suitable (degrees, kilometers, or miles) `distanceUnits`, and then verify that spatial queries behave as expected.

   - For optimal indexing of multipolygon shapes, you must set `useJtsMulti="false"`. For example:
     
     ```
     <fieldType autoIndex="true" useJtsMulti="false" class="solr.SpatialRecursivePrefixTreeFieldType">
     <distErrPct="0.0125"
     distanceUnits="kilometers" geo="true" name="WktField"
     spatialContextFactory="org.locationtech.spatial4j.context.jts.JtsSpatialContextFactory"/>
     ```

5. **DSE Search only for DSE 5.1.6 and later**
   
   - Slow startup on nodes with large encrypted indexes is resolved. However, action is required to realize the performance gains. You must do a full reindex of all encrypted
search indexes on each node in your cluster. Plan sufficient time after the upgrade is complete to reindex with deleteAll=true in a rolling fashion. For example:

```
dsetool reload_core keyspace_name.table_name distributed=false reindex=true deleteAll=true
```

- Applies only if you are using HTTP writes with JSON documents (deprecated), a known issue causes auto generated solrconfig.xml to have invalid requestHandler for JSON core creations after upgrade to 5.1.0. Change the auto generated solrconfig.xml:

```
<requestHandler name="/update/json"
    class="solr.UpdateUpdateRequestHandler" startup="lazy"/>
```

6. **DSEFS only**: A new schema is available for DSEFS.

   **Warning**: Dropping a keyspace is not recoverable without a backup. If you have non-temporary data, do not drop the dsefs keyspace. No action is required. DSEFS will continue to work using the DSE 5.0 schema.

   If you have no data in DSEFS or if you are using DSEFS only for temporary data, follow these steps to use the new schema:

   a. Stop DSEFS on all nodes. In the `dsefs_options` section of `dse.yaml`, set `enabled=false`.

   b. Drop the dsefs keyspace:

   ```
   DROP KEYSPACE dsefs
   ```

   c. Clear the dsefs data directories on each node.

   For example, if the dsefs_options section of `dse.yaml` has `data_directories` configured as:

   ```
   dsefs_options:
   ...  
   data_directories:
   - dir: /var/lib/dsefs/data
   ```

   This command removes the directories:

   ```
   rm -r /var/lib/dsefs/data/*
   ```

   d. Start DSEFS with DSE 5.1 to use the new schema.
e. If you backed up existing DSEFS data before the upgrade, copy the data back into DSEFS from local storage.

7. **DSE Analytics only:** If you are using Spark SQL tables, migrate them to the new Hive metastore format used by Spark SQL using the `dse spark-sql-metastore-migrate` command.

   ```bash
   $ dse spark-sql-metastore-migrate
   ```

8. **DSE Advanced Replication only:** Because DSE Advanced Replication is substantially revised, you must migrate to the newer version in DSE 5.1. Both V1 and V2 are run in DSE 5.1 in parallel to accomplish the migration.

   a. Create a V2 destination for the hub configured in V1:

   ```bash
   $ dse advrep destination create --name upgradedest --addresses 10.200.100.250 --transmission-enabled false
   ```

   b. Create a V2 channel for each V1 channel:

   ```bash
   $ dse advrep channel create --source-keyspace demo --source-table sensor_readings --destination upgradedest --source-id edge1 --source-id-column hub_id --priority 1
   ```

   c. Resume collection and transmission:

   ```bash
   $ dse advrep channel resume --source-keyspace demo --source-table sensor_readings --collection-enabled true --transmission-enabled true
   ```

   d. Make sure the V2 is running and replicating, then disable collection on the V1 channels:

   ```bash
   $ dse advrep --v1 edge channel pause --keyspace demo --table sensor_readings
   ```

   e. Wait for V1 to drain the V1 replication log, checking for a zero count:

   ```bash
   $ dse advrep --v1 edge rl-count
   ```

   f. Delete V1 channels and disable V1 hub:

   ```bash
   $ for f in `dse advrep --v1 edge list-conf|cut -f1 -d' '`; do dse advrep --v1 edge remove-conf --$f; done;
   ```
Warning messages during and after upgrade

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

- When upgrading nodes with DSE Advanced Replication, there might be some WriteTimeoutExceptions during a rolling upgrade while mixed versions of nodes exist. Some write consistency limitations apply while mixed versions of nodes exist. The WriteTimeout issue is resolved after all nodes are upgraded.

Error messages provide information to help identify problems.

- If you see an error message like:

```
- Cannot start node if snitch's data center (Cassandra) differs from previous data center (Analytics). Please fix the snitch configuration, decommission and rebootstrap this node or use the flag -Dcassandra.ignore_dc=true.
```

Follow upgrade instructions in step 6 (page 128). You must start in Spark mode or add the special start-up parameter -Dcassandra.ignore_dc=true for each node.

Upgrading to earlier versions

Upgrading to DataStax Enterprise 5.0

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

Always upgrade to latest patch release (page 10) on your current version before you upgrade to a higher version. Fixes included in the latest patch release might help or smooth the upgrade process.

- The latest version of DSE 4.8 is 4.8.16
- The latest version of DSE 4.7 is 4.7.9.

**Attention:** TTL expiration timestamps are susceptible to the year 2038 problem. If the TTL value is long and an expiration date that is greater than the maximum threshold of 2038-01-19T03:14:06+00:00, the data is immediately expired and purged on the next compaction. DataStax strongly recommends upgrading to DSE 5.0.15 or later and taking required action to protect against silent data loss. (DSP-15412).

**Attention:** Read and understand these instructions before upgrading. Carefully reviewing the planning (page 9) and upgrading instructions can ensure a smooth upgrade and avoid pitfalls and frustrations.

On this page:

- Apache Cassandra version change (page 134)
- General recommendations (page 134)
• General restrictions and limitations during the upgrade process (page 134)
• Preparing to upgrade (page 136)
• Upgrade steps (page 139)
• Warning messages during and after upgrade (page 143)

Apache Cassandra™ version change

Upgrading from DataStax Enterprise 4.7 or 4.8 to 5.0 includes a major Cassandra version change.
• Upgrading SSTables is required for upgrades that contain major Apache Cassandra releases:
  # DataStax Enterprise 5.0 uses Cassandra 3.0.
  # DataStax Enterprise 4.7 to 4.8 use Cassandra 2.1.
  # DataStax Enterprise 4.0 to 4.6 use Cassandra 2.0.

Be sure to follow the recommendations for upgrading the SSTables.

General recommendations

DataStax recommends backing up your data prior to any version upgrade, including logs and custom configurations. 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. OpsCenter 6.5 and later is recommended.

General restrictions and limitations during the upgrade process

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 during an upgrade**

• **Do not** enable new features.
• Do not run nodetool repair. If you have the OpsCenter Repair Service configured, **turn off** the Repair Service.
• Ensure OpsCenter compatibility. See [DataStax OpsCenter compatibility](https://docs.datastax.com/en/opscenter/6.5/opscenter começar.html) with DataStax Enterprise.
• During the upgrade, do not bootstrap or decommission nodes.
• Do not issue these types of CQL queries during a rolling restart: DDL and TRUNCATE.
• Do not enable Change Data Capture (CDC) on a mixed-version cluster. Upgrade all nodes to DSE 5.1 or later before enabling CDC.
• 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.
Note: Nodes on different versions might show a schema disagreement during an upgrade.

- NodeSync waits to start until all nodes are upgraded.
- Do not enable Change Data Capture (CDC) on a mixed-version cluster. Upgrade all nodes before enabling CDC.
- Ensure OpsCenter compatibility. OpsCenter 6.7 is required for managing DSE 6.7 clusters. See DSE OpsCenter compatibility with DataStax Enterprise.

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 reindex 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, internode_messaging_options are used. The internode_messaging_options are used by several components of DataStax Enterprise. For 5.0 and later, all internode messaging requests use this service.

Restrictions for nodes using any kind of security

- Do not change security credentials or permissions until the upgrade is complete on all nodes.
- If you are not already using Kerberos, do not set up Kerberos authentication before upgrading. First upgrade the cluster, and then set up Kerberos.

Upgrading drivers and possible impact when driver versions are incompatible

Be sure to check driver compatibility. Depending on the driver version, you might need to recompile your client application code. See DataStax driver changes. 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:

- **Protocol version**: Because some drivers can use different protocol versions, force the protocol version at start up. For example, keep the Java driver at its current protocol version while the driver upgrade is happening. Switch to the Java driver to the new protocol version only after the upgrade is complete on all nodes in the cluster.
- **Initial contact points**: 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 details on protocol version negotiation, see protocol versions with mixed clusters in the Java driver version you’re using, for example, Java driver.

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.

2. Familiarize yourself with the changes and features in this release:
   - Be sure your platform is supported.
   - OpenJDK 8 or Oracle Java SE Runtime Environment 8 (JDK) (1.8.0_40 minimum). Earlier or later versions are not supported.
   - 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:

<table>
<thead>
<tr>
<th>Current version</th>
<th>Upgrade version</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataStax Enterprise 4.7 or 4.8</td>
<td>DataStax Enterprise 5.0</td>
</tr>
<tr>
<td>DataStax Enterprise 4.0, 4.5, or 4.6</td>
<td>DataStax Enterprise 4.8</td>
</tr>
<tr>
<td>DataStax Community or open source Apache Cassandra™ 2.1.x</td>
<td>DataStax Enterprise 4.8</td>
</tr>
<tr>
<td>DataStax Community 3.0.x</td>
<td>No interim version required.</td>
</tr>
<tr>
<td>DataStax Distribution of Apache Cassandra™ 3.x</td>
<td>Upgrade not available.</td>
</tr>
</tbody>
</table>

4. Upgrade the SSTables on each node to ensure that all SSTables are on the current version.
   This is required for DataStax Enterprise upgrades that include a major Cassandra version changes.
   Warning: Failure to upgrade SSTables when required results in a significant performance impact and increased disk usage.
$ nodetool upgradesstables

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

Use the --jobs option to set the number of SSTables that upgrade simultaneously. The default setting is 2, which minimizes impact on the cluster. Set to 0 to use all available compaction threads. For information about nodetool upgradesstables, including how to speed it up, see the DataStax Support KB article Nodetool upgradesstables FAQ.

**Warning:** Failure to upgrade SSTables when required results in a significant performance impact and increased disk usage. For more information about nodetool upgradesstables, including how to speed it up, see the DataStax Support KB article Nodetool upgradesstables FAQ.

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

$ java -version

- **Recommended.** OpenJDK 8 (1.8.0_151 minimum)

  **Note:** Recommendation changed due to the end of public updates for Oracle JRE/JDK 8. See Oracle Java SE Support Roadmap.

- **Supported.** Oracle Java SE 8 (JRE or JDK) (1.8.0_151 minimum)

The JDK is recommended for development and production systems, and provides useful troubleshooting tools that are not in the JRE, such as jstack, jmap, jps, and jstat.

**Important:** Although Oracle JRE/JDK 8 is supported, DataStax does more extensive testing on OpenJDK 8 starting with DSE 5.0.15.

6. Run nodetool repair to ensure that data on each replica is consistent with data on other nodes.

7. 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.

- HTTP-based Solr shard transport option is deprecated. Use Inter-node messaging options instead. For 5.0, all internode messaging requests use this internal messaging service. HTTP will be removed in 5.1.
- Tune the schema **before** you upgrade. For DSE 5.0.10 and later, all field definitions in the schema are validated and must be DSE Search supported, even if the fields are not indexed, have docValues applied, or used for copy-field source. The default behavior of automatic resource generation includes all columns. To improve performance, take action to prevent the fields from being loaded from the database. Include only the required fields in the schema by removing or commenting out unused fields in the schema. After you change the schema, **reload the Solr core**.

8. **DSE Search partition key names**

The partition key names of COMPACT STORAGE tables backed by DSE Search indexes match the `uniqueKey` in `schema.xml`. For example, consider the following table is created with compact storage:

```
CREATE TABLE keyspace_name.table_name (key text PRIMARY KEY, foo text, solr_query text)
WITH COMPACT STORAGE
```

and the Solr `schema.xml` is:

```
...<uniqueKey>id</uniqueKey>
...
```

then rename the key in the table to match the schema:

```
ALTER TABLE ks.table RENAME key TO id;
```

9. **DSE Analytics nodes**

When performing a rolling upgrade in a datacenter from DSE 4.8 to DSE 5.0 manually update the name of the metastore table used by Spark in `hive-site.xml`.

**Note:** Only perform this step if you want a rolling upgrade with no interruption before the entire datacenter is upgraded. DSE 5.0 will elect the Spark Master after the entire datacenter is upgraded if you don't manually update `hive-site.xml`.

For tarball installations:

```
$ sudo perl -i -pe 's|cfs:///user/spark/warehouse|cfs:///user/hive/warehouse|g' /etc/dse/spark/hive-site.xml
```
$ sudo perl -i -pe 's|sparkmetastore|MetaStore|g' /etc/dse/spark/hive-site.xml

For package installations:

$ sudo perl -i -pe 's|cfs:///user/spark/warehouse|cfs:///user/hive/warehouse|g' /usr/local/lib/dse/resources/spark/conf/hive-site.xml

$ sudo perl -i -pe 's|sparkmetastore|MetaStore|g' /usr/local/lib/dse/resources/spark/conf/hive-site.xml

Before DSE 5.0, Spark used the Hive metastore table `HiveMetaStore.MetaStore`. Starting in DSE 5.0, the Hive and Spark metastore tables have been separated, and Spark uses the `HiveMetaStore.sparkmetastore` table. If DSE 5.0 starts and the metastore table is missing, the node will wait for entire cluster to be upgraded before starting Spark because it will have to create the metastore table first. Manually updating the configuration allows Spark nodes to create the metastore table and elect a Master in a mixed datacenter.

Be prepared for some inconveniences during the rolling upgrade. If the Spark contact point is set to a DSE 5.0 node, it will be able to use only DSE 5.0 replicas to access data. However if the contact point is set to a DSE 4.8 node, it will be able to access data on all replicas in the cluster.

10. Back up the configuration files you use to a folder that is not in the directory where you normally run commands.

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

Upgrade steps

**Tip:** The DataStax installer (page 184) 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. Some warning messages are displayed during and after upgrade (page 143).

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

1. Upgrade order matters. Upgrade nodes in this order:
   - In multiple datacenter clusters, upgrade every node in one datacenter before upgrading another datacenter.
   - Upgrade the seed nodes within a datacenter first.
For DSE Analytics nodes using DSE Hadoop, upgrade the Job Tracker node first. Then upgrade Hadoop nodes, followed by Spark nodes.

- Upgrade nodes in this order:
  
  a. DSE Analytics datacenters
  
  b. Transactional/DSE Graph datacenters
  
  c. DSE Search datacenters

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. To flush the commit log of the old installation:

   $ nodetool -h hostname drain

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

   **Important:** This step is mandatory when upgrading between major Cassandra versions that change SSTable formats, rendering commit logs from the previous version incompatible with the new version.

4. **Stop the node.**

5. Use the appropriate method to install DSE 5.0 on a supported platform:

   - Binary tarball
   - Debian-based systems using APT
   - RHEL-based systems using Yum

   **Attention:** For upgrades on RHEL-based systems that have demos installed, you must specify the package installation in a single line, and specify the version for dse-full and dse-demos. For example:

   $ sudo yum install dse-full-5.0.15-1 dse-demos-5.0.15-1

   **Note:** Install the new product version using the same installation method that is on the system. The upgrade proceeds with installation regardless of the installation method and might result in issues.
6. 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`.

7. 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 (page 136) with the Apache Cassandra and DataStax Enterprise changes and features in the new release.
      - Ensure that keyspace replication factors are correct for your environment:
        ```
        # Set the keyspace replication factor for analytics keyspaces.
        # Set the keyspace replication factor for system_auth and dse_security keyspaces.
        ```
   b. Merge the applicable modifications into 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. 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. Because DataStax Enterprise 5.0 uses Cassandra 3.0, the `output.log` might 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 no longer contact other nodes on this port. Instead requests use inter-node messaging options. The internode_messaging_options are used by several components of DataStax Enterprise. For 5.0 and later, all internode messaging requests use this service.

Warnings, errors, and exceptions are frequently found in the logs when starting 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.

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

12. After all nodes are upgraded, you must drop the following legacy tables:
   system_auth.users, system_auth.credentials and system_auth.permissions. This step is required for all workloads when legacy tables exist.

   As described in Cassandra NEWS.txt, the authentication and authorization subsystems have been redesigned to support role-based access control (RBAC), which results in a change to the schema of the system_auth keyspace.

13. DSE Search only for DSE 5.0.12 and later After the upgrade, you must do a full reindex of all encrypted search indexes on each node in your cluster. Slow startup on nodes with large encrypted indexes is resolved. However, action is required to realize the performance gains. Plan sufficient time after the upgrade is complete to reindex with deleteAll=true in a rolling fashion. For example:

   dsetool reload_core keyspace_name.table_name distributed=false reindex=true deleteAll=true

14. After the new version is installed on all nodes, upgrade the SSTables:

   $ 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.

   Use the --jobs option to set the number of SSTables that upgrade simultaneously. The default setting is 2, which minimizes impact on the cluster. Set to 0 to use all available
compaction threads. For information about nodetool upgradesstables, including how to speed it up, see the DataStax Support KB article Nodetool upgradesstables FAQ.

15. For multiple datacenter deployments, change the replication factor of the system_distributed keyspace to NetworkTopologyStrategy.

16. If you use the OpsCenter Repair Service, turn on the Repair Service.

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 from DataStax Enterprise 4.7 to 4.8**

Follow these instructions to upgrade from DataStax Enterprise 4.7 to 4.8.

Always upgrade to latest patch release on your current version before you upgrade to a higher version. Fixes included in the latest patch release might help or smooth the upgrade process.

The latest version of DSE 4.7 is 4.7.9.

**Attention: Read and understand these instructions before upgrading.** Carefully reviewing the planning (page 9) and upgrading instructions can ensure a smooth upgrade and avoid pitfalls and frustrations.

**Attention: TTL expiration timestamps are susceptible to the year 2038 problem.** If the TTL value is long and an expiration date that is greater than the maximum threshold of 2038-01-19T03:14:06+00:00, the data is immediately expired and purged on the
next compaction. DataStax strongly recommends upgrading to DSE 4.8.16 and taking required action to protect against silent data loss. (DSP-15412).

General recommendations

DataStax recommends backing up your data prior to any version upgrade, including logs and custom configurations. 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. OpsCenter 6.5 and later is recommended.

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.
- During the upgrade, do not bootstrap or decommission nodes.
- Do not issue these types of CQL queries during a rolling restart: DDL and TRUNCATE.
- During the upgrade, 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 (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 reindex DSE Search nodes during upgrade.
- Do not issue these types of queries during a rolling restart: BATCH 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.
Preparing to upgrade from 4.7 to 4.8

Follow these steps to prepare to upgrade from DataStax Enterprise 4.7 to 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.

2. Familiarize yourself with the changes and features in this release:
   - Release notes for DataStax Enterprise 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 4.8 production-certified changes to Apache Cassandra.

3. Verify your current product version is the latest patch for DataStax Enterprise 4.7.
   The latest version is 4.7.9.

4. Upgrade the SSTables on each node to ensure that all SSTables are on the current version.

   This is required for DataStax Enterprise upgrades that include a major Cassandra version changes.

   Warning: Failure to upgrade SSTables when required results in a significant performance impact and increased disk usage.

   $ nodetool upgradesstables

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

   Use the --jobs option to set the number of SSTables that upgrade simultaneously. The default setting is 2, which minimizes impact on the cluster. Set to 0 to use all available compaction threads. For information about nodetool upgradesstables, including how to speed it up, see the DataStax Support KB article Nodetool upgradesstables FAQ.

   Warning: Failure to upgrade SSTables when required results in a significant performance impact and increased disk usage. For information about nodetool upgradesstables, including how to speed it up, see the DataStax Support KB article Nodetool upgradesstables FAQ.

5. Verify the Java runtime version and upgrade to the recommended 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 to a folder that is not in the directory where you normally run commands.

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

**Steps for upgrading from 4.7 to 4.8**

**Tip:** The *DataStax installer* (page 184) upgrades DataStax Enterprise and automatically performs many upgrade tasks.

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

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

1. **Upgrade order matters.** Upgrade nodes in this order:
   - In multiple datacenter clusters, upgrade every node in one datacenter before upgrading another datacenter.
   - Upgrade the seed nodes within a datacenter first.

      For DSE Analytics nodes using DSE Hadoop, upgrade the Job Tracker node first. Then upgrade Hadoop nodes, followed by Spark nodes.

   - Upgrade nodes in this order:
     - **a.** DSE Analytics datacenters
b. Transactional/DSE Graph datacenters

c. DSE Search datacenters

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:
   - 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.

4. To flush the commit log of the old installation:

   $ nodetool -h hostname drain

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

   **Important:** This step is mandatory when upgrading between major Cassandra versions that change SSTable formats, rendering commit logs from the previous version incompatible with the new version.

5. Stop the node.

6. Use the appropriate installation type to install the new product version on a supported platform:
   - RHEL: 4.6, 4.7, 4.8, 5.0
   - Debian: 4.6, 4.7, 4.8, 5.0
   - Tarball: 4.6, 4.7, 4.8, 5.0

   **Note:** Install the new product version using the same installation type that is on the system. The upgrade proceeds with installation regardless of the installation type. If you use a different installation type, the upgrade might result in issues.

7. 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.
Upgrading DataStax Enterprise

- Be sure you are familiar with the Apache Cassandra and DataStax Enterprise changes and features in the new release.
- Ensure that keyspace replication factors are correct for your environment:
  
  # Set the keyspace replication factor for analytics keyspaces.
  # Set the keyspace replication factor for system_auth and dse_security keyspace.

b. Merge the applicable modifications into 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. 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 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.

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

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

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

13. After the new version is installed, upgrade the SSTables on the upgraded nodes.

    This is recommended for optimal performance, but is not required.

    `$ nodetool upgradesstables`

    If the SSTables are already on the current version, the command returns immediately and no action is taken.
Use the `--jobs` option to set the number of SSTables that upgrade simultaneously. The default setting is 2, which minimizes impact on the cluster. Set to 0 to use all available compaction threads. For information about nodetool upgradeStables, including how to speed it up, see the DataStax Support KB article Nodetool upgradeStables FAQ.

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. Carefully reviewing the planning (page 9) and upgrading instructions can ensure a smooth upgrade and avoid pitfalls and frustrations.

Cassandra version change

Upgrading 4.0, 4.5, and 4.6 to DataStax Enterprise 4.7 or 4.8 includes a major Cassandra version change. Be sure to follow the recommendations for upgrading the SSTables.

Upgrading to DataStax Enterprise 4.7 includes these changes:

- A Cassandra version change from 2.0 to 2.1, and changes the default value of the `commitlog_total_space_in_mb` value in `cassandra.yaml` from 1024 MB to 8192 MB. Adjust the `commitlog_total_space_in_mb` setting for your environment to ensure that you do not run out of disk space after upgrade.
- Logging is changed from log4j to logback, with changes to logging retention policies. Configure the logger by setting options in `logback.xml`. See Configuring logging.

General recommendations

DataStax recommends backing up your data prior to any version upgrade, including logs and custom configurations. 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. OpsCenter 6.5 and later is recommended.

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.
- During the upgrade, do not bootstrap or decommission nodes.
Upgrading DataStax Enterprise

- Do not issue these types of CQL queries during a rolling restart: **DDL** and **TRUNCATE**.
- During the upgrade, 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 (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 reindex DSE Search nodes during upgrade.
- Do not issue these types of queries during a rolling restart: **BATCH** 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.

Preparing to upgrade from 4.0, 4.5, or 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](#).

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 or 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.

This is required for DataStax Enterprise upgrades that include a major Cassandra version changes.

**Warning:** Failure to upgrade SSTables when required results in a significant performance impact and increased disk usage.

```
$ nodetool upgradesstables
```

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

Use the `--jobs` option to set the number of SSTables that upgrade simultaneously. The default setting is 2, which minimizes impact on the cluster. Set to 0 to use all available compaction threads. For information about nodetool upgradesstables, including how to speed it up, see the DataStax Support KB article Nodetool upgradesstables FAQ.

**Warning:** Failure to upgrade SSTables when required results in a significant performance impact and increased disk usage. For information about nodetool upgradesstables, including how to speed it up, see the DataStax Support KB article Nodetool upgradesstables FAQ.

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 reload the Solr core.

8. Back up the configuration files you use to a folder that is not in the directory where you normally run commands.

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

Steps for upgrading from 4.0, 4.5, or 4.6 to 4.7 or 4.8

**Tip:** The DataStax installer ([page 184](#)) upgrades DataStax Enterprise and automatically performs many upgrade tasks.

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

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. Upgrade order matters. Upgrade nodes in this order:
   - In multiple datacenter clusters, upgrade every node in one datacenter before upgrading another datacenter.
   - Upgrade the seed nodes within a datacenter first.

   For DSE Analytics nodes using DSE Hadoop, upgrade the Job Tracker node first. Then upgrade Hadoop nodes, followed by Spark nodes.

   - Upgrade nodes in this order:
     a. DSE Analytics datacenters
     b. Transactional/DSE Graph datacenters
     c. DSE Search datacenters

   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:
   - 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, 4.8). You can turn on paging after
you verify the upgrade.

- **For upgrades from 4.0.0:** See Special steps for upgrades from DataStax Enterprise
  4.0.0 (page 170) for special instructions.

4. To flush the commit log of the old installation:
   
   $ nodetool -h hostname drain
   
   This step saves time when nodes start up after the upgrade, and prevents DSE Search
   nodes from having to reindex data.

   **Important:** This step is mandatory when upgrading between major Cassandra
   versions that change SSTable formats, rendering commit logs from the previous
   version incompatible with the new version.

5. Stop the node (4.7, 4.8).

6. Use the appropriate installation type to install the new product version on a supported
   platform:

   - RHEL: 4.6, 4.7, 4.8, 5.0
   - Debian: 4.6, 4.7, 4.8, 5.0
   - Tarball: 4.6, 4.7, 4.8, 5.0

   **Note:** Install the new product version using the same installation type that is on
   the system. The upgrade proceeds with installation regardless of the installation
   type. If you use a different installation type, the upgrade might result in issues.

7. 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.

8. Start the node.
• Installer-Services and Package installations: See Starting DataStax Enterprise as a service (4.7, 4.8).
• Installer-No Services and Tarball installations: See Starting DataStax Enterprise as a stand-alone process 4.7, 4.8).

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 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"
   ```

11. 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'};
   ```

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

13. If existing tables use the DSE In-Memory option:
a. Turn off SSTable compression:

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

b. Rewrite existing SSTables without compression:

```bash
$ nodetool upradesstables -a <keyspacename> <tablename>
```

Use the `--jobs` option to set the number of SSTables that upgrade simultaneously. The default setting is 2, which minimizes impact on the cluster. Set to 0 to use all available compaction threads. For information about nodetool upradesstables, including how to speed it up, see the DataStax Support KB article Nodetool upradesstables FAQ.

14. Upgrade the SSTables on the remaining nodes:

```bash
$ nodetool upradesstables
```

**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.

Use the `--jobs` option to set the number of SSTables that upgrade simultaneously. The default setting is 2, which minimizes impact on the cluster. Set to 0 to use all available compaction threads. For information about nodetool upradesstables, including how to speed it up, see the DataStax Support KB article Nodetool upradesstables FAQ.

# 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. Carefully reviewing the planning (page 9) and upgrading instructions can ensure a smooth upgrade and avoid pitfalls and frustrations.

Cassandra version change

Upgrading from DataStax Enterprise 3.2.5 to 4.6 includes a major Cassandra version change. Be sure to follow the recommendations for upgrading the SSTables.
General recommendations

DataStax recommends backing up your data prior to any version upgrade, including logs and custom configurations. 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. OpsCenter 6.5 and later is recommended.

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 during an upgrade

- **Do not** enable new features.
- Do not run nodetool repair. If you have the OpsCenter Repair Service configured, turn off the Repair Service.
- Ensure OpsCenter compatibility. See DataStax OpsCenter compatibility with DataStax Enterprise.
- During the upgrade, do not bootstrap or decommission nodes.
- Do not issue these types of CQL queries during a rolling restart: DDL and TRUNCATE.
- Do not enable Change Data Capture (CDC) on a mixed-version cluster. Upgrade all nodes to DSE 5.1 or later before enabling CDC.
- 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.

**Note:** Nodes on different versions might show a schema disagreement during an upgrade.

- NodeSync waits to start until all nodes are upgraded.
- Do not enable Change Data Capture (CDC) on a mixed-version cluster. Upgrade all nodes before enabling CDC.
- Ensure OpsCenter compatibility. OpsCenter 6.7 is required for managing DSE 6.7 clusters. See DSE OpsCenter compatibility with DataStax Enterprise.

DSE Graph nodes restrictions

Graph nodes have the same restrictions as the workload they run on. Do not alter graph schema during upgrades. Workload-specific restrictions apply for analytics and search nodes, such as no OLAP queries during upgrades.

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 Analytic (Spark) nodes
• Do not run analytics jobs until all nodes are upgraded.

Restrictions for DSE Analytic (Spark) nodes
• Do not run analytics jobs until all nodes are upgraded.
• All nodes in the cluster must be upgraded to the new version before Spark Worker and Spark Master will start.

Restrictions for DSE Analytic (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 upgrade restrictions and limitations
• Do not update schemas.
• Do not reindex DSE Search nodes during upgrade.
• DSE 6.0 and later versions use a new Lucene codec. Segments written with this new codec cannot be read by earlier versions of DSE. To downgrade to earlier versions, the entire data directory for the search index in question must be cleared.
• # DSE Search in DataStax Enterprise 6.7 uses Apache Solr 6.0. This significant change requires advanced planning and specific actions before and after the upgrade.

DSE Search upgrade restrictions and limitations
• Do not update schemas.
• Do not reindex DSE Search nodes during upgrade.
• DSE 6.0 and later versions use a new Lucene codec. Segments written with this new codec cannot be read by earlier versions of DSE. To downgrade to earlier versions, the entire data directory for the search index in question must be cleared.
• DSE Search in DataStax Enterprise 5.1 and later uses Apache Solr 6.0. This significant change requires advanced planning and specific actions before and after the upgrade.

Important: Before you upgrade DSE Search or SearchAnalytics workloads, you must follow the specific tasks in the Advanced preparation for upgrading DSE Search and SearchAnalytics nodes (page 58) section.

DSE Search upgrade restrictions and limitations
• Do not update schemas.
• Do not reindex DSE Search nodes during upgrade.

Restrictions for DSE Search (Solr) nodes
• Do not update schemas.
• Do not reindex DSE Search nodes during upgrade.
• Do not issue these types of queries during a rolling restart: BATCH 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 the upgrade is complete on all nodes.
• If you are not already using Kerberos, do not set up Kerberos authentication before upgrading. First upgrade the cluster, and then set up Kerberos.

Restrictions for nodes using any kind of security
• Do not change security credentials or permissions until after the upgrade is complete.

Upgrading drivers and possible impact when driver versions are incompatible
Be sure to check driver compatibility. Depending on the driver version, you might need to recompile your client application code. See DataStax driver changes. 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:

• **Protocol version:** Because some drivers can use different protocol versions, force the protocol version at start up. For example, keep the Java driver at its current protocol version while the driver upgrade is happening. Switch to the Java driver to the new protocol version only after the upgrade is complete on all nodes in the cluster.

• **Initial contact points:** 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 details on protocol version negotiation, see protocol versions with mixed clusters in the Java driver version you’re using, for example, Java driver.

Preparing to upgrade from 3.2.5 and later to 4.6.

**Tip:** The DataStax installer (page 184) 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.
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
   ```

   This step is required for DataStax Enterprise upgrades that include a major Cassandra version changes.

   **Warning:** Failure to upgrade SSTables when required results in a significant performance impact and increased disk usage.

   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 (page 170).

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 to a folder that is not in the directory where you
   normally run commands.

   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.

Upgrading from 3.2.5 and later to 4.6

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

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

1. Upgrade order matters. Upgrade nodes in this order:
   - In multiple datacenter clusters, upgrade every node in one datacenter before
     upgrading another datacenter.
   - Upgrade the seed nodes within a datacenter first.

     For DSE Analytics nodes using DSE Hadoop, upgrade the Job Tracker node first.
     Then upgrade Hadoop nodes, followed by Spark nodes.

   - Upgrade nodes in this order:
     a. DSE Analytics datacenters
     b. Transactional/DSE Graph datacenters
     c. DSE Search datacenters

   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. To flush the commit log of the old installation:

   $ nodetool -h hostname drain

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

   **Important:** This step is mandatory when upgrading between major Cassandra versions that change SSTable formats, rendering commit logs from the previous version incompatible with the new version.

3. **DSE Analytics nodes**: Kill all Spark worker processes.

4. Stop the node (**Stop the node**).

5. Use the appropriate installation type to install the new product version on a **supported platform**:

   - RHEL: 4.6, 4.7, 4.8, 5.0
   - Debian: 4.6, 4.7, 4.8, 5.0
   - Tarball: 4.6, 4.7, 4.8, 5.0

   **Note:** Install the new product version using the same installation type that is on the system. The upgrade proceeds with installation regardless of the installation type. If you use a different installation type, the upgrade might result in issues.

6. 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
   ```

7. Remove the `delegated_snitch` option from the old `dse.yaml` file.

8. To configure the new version, use your backup configuration files to merge modifications into the configuration files for the new version.

9. Start the node:

   - Installer-Services and Package
   - Installer-No Services and Tarball

10. Verify that the upgraded datacenter names match the datacenter names in the keyspace schema definition:
11. Review the logs for warnings, errors, and exceptions.

Warnings, errors, and exceptions are frequently found in the logs when starting 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.

13. When the upgrade includes a major Cassandra version, you must upgrade the SSTables. DataStax recommends upgrading the SSTables on one node at a time or when using racks, one rack at a time.

   **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.

   $ nodetool upgradesstables

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

   Use the --jobs option to set the number of SSTables that upgrade simultaneously. The default setting is 2, which minimizes impact on the cluster. Set to 0 to use all available compaction threads. DataStax recommends running the upgradesstables command on one node at a time or when using racks, one rack at a time.

   **Note:** You can run the upgradesstables command before all the nodes are upgraded as long as you run this command on only one node at a time or when using racks, one rack at a time. Running upgradesstables on too many nodes will degrade performance.

### Upgrading to DataStax Enterprise 4.0 or 4.5

**Attention:** Read and understand these instructions before upgrading. Carefully reviewing the planning (page 9) and upgrading instructions can ensure a smooth upgrade and avoid pitfalls and frustrations.

Cassandra version change

Upgrading to DataStax Enterprise 4.0 or 4.5 includes a major Cassandra version change. Be sure to follow the recommendations for upgrading the SSTables.
General recommendations

DataStax recommends backing up your data prior to any version upgrade, including logs and custom configurations. 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. OpsCenter 6.5 and later is recommended.

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 during an upgrade

- **Do not** enable new features.
- Do not run nodetool repair. If you have the OpsCenter Repair Service configured, turn off the Repair Service.
- Ensure OpsCenter compatibility. See DataStax OpsCenter compatibility with DataStax Enterprise.
- During the upgrade, do not bootstrap or decommission nodes.
- Do not issue these types of CQL queries during a rolling restart: DDL and TRUNCATE.
- Do not enable Change Data Capture (CDC) on a mixed-version cluster. Upgrade all nodes to DSE 5.1 or later before enabling CDC.
- 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.

**Note:** Nodes on different versions might show a schema disagreement during an upgrade.

- NodeSync waits to start until all nodes are upgraded.
- Do not enable Change Data Capture (CDC) on a mixed-version cluster. Upgrade all nodes before enabling CDC.
- Ensure OpsCenter compatibility. OpsCenter 6.7 is required for managing DSE 6.7 clusters. See DSE OpsCenter compatibility with DataStax Enterprise.

DSE Graph nodes restrictions

Graph nodes have the same restrictions as the workload they run on. Do not alter graph schema during upgrades. Workload-specific restrictions apply for analytics and search nodes, such as no OLAP queries during upgrades.

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 Analytic (Spark) nodes
• Do not run analytics jobs until all nodes are upgraded.

Restrictions for DSE Analytic (Spark) nodes
• Do not run analytics jobs until all nodes are upgraded.
• All nodes in the cluster must be upgraded to the new version before Spark Worker and Spark Master will start.

Restrictions for DSE Analytic (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 upgrade restrictions and limitations
• Do not update schemas.
• Do not reindex DSE Search nodes during upgrade.
• DSE 6.0 and later versions use a new Lucene codec. Segments written with this new codec cannot be read by earlier versions of DSE. To downgrade to earlier versions, the entire data directory for the search index in question must be cleared.
• # DSE Search in DataStax Enterprise 6.7 uses Apache Solr 6.0. This significant change requires advanced planning and specific actions before and after the upgrade.

DSE Search upgrade restrictions and limitations
• Do not update schemas.
• Do not reindex DSE Search nodes during upgrade.
• DSE 6.0 and later versions use a new Lucene codec. Segments written with this new codec cannot be read by earlier versions of DSE. To downgrade to earlier versions, the entire data directory for the search index in question must be cleared.
• DSE Search in DataStax Enterprise 5.1 and later uses Apache Solr 6.0. This significant change requires advanced planning and specific actions before and after the upgrade.

Important: Before you upgrade DSE Search or SearchAnalytics workloads, you must follow the specific tasks in the Advanced preparation for upgrading DSE Search and SearchAnalytics nodes (page 58) section.

DSE Search upgrade restrictions and limitations
• Do not update schemas.
• Do not reindex DSE Search nodes during upgrade.

Restrictions for DSE Search (Solr) nodes
• Do not update schemas.
• Do not reindex DSE Search nodes during upgrade.
• Do not issue these types of queries during a rolling restart: \textsc{batch} or \textsc{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 the upgrade is complete on all nodes.
• If you are not already using Kerberos, do not set up Kerberos authentication before upgrading. First upgrade the cluster, and then set up Kerberos.

Restrictions for nodes using any kind of security
• Do not change security credentials or permissions until after the upgrade is complete.

Upgrading drivers and possible impact when driver versions are incompatible
Be sure to check driver compatibility. Depending on the driver version, you might need to recompile your client application code. See DataStax driver changes.
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:

• **Protocol version:** Because some drivers can use different protocol versions, force the protocol version at start up. For example, keep the Java driver at its current protocol version while the driver upgrade is happening. Switch to the Java driver to the new protocol version only after the upgrade is complete on all nodes in the cluster.

• **Initial contact points:** 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 details on protocol version negotiation, see protocol versions with mixed clusters in the Java driver version you’re using, for example, Java driver.

Preparing to upgrade from 3.2.5 or later to 4.0 or 4.5

**Tip:** The DataStax installer (page 184) 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.
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 (page 170).

5. Upgrade the SSTables on each node to ensure that all SSTables are on the current version.

   This is required for DataStax Enterprise upgrades that include a major Cassandra version changes.

   **Warning:** Failure to upgrade SSTables when required results in a significant performance impact and increased disk usage.

   `$ nodetool upgradesstables`

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

   Use the `--jobs` option to set the number of SSTables that upgrade simultaneously. The default setting is 2, which minimizes impact on the cluster. Set to 0 to use all available compaction threads. For information about nodetool upgradesstables, including how to speed it up, see the DataStax Support KB article Nodetool upgradesstables FAQ.

   **Warning:** Failure to upgrade SSTables when required results in a significant performance impact and increased disk usage. For for information about nodetool upgradesstables, including how to speed it up, see the DataStax Support KB article Nodetool upgradesstables FAQ.

6. If you are upgrading from DataStax Enterprise 4.0.0 and have DSE Search nodes, see Special steps for upgrades from DataStax Enterprise 4.0.0 (page 170).

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 to a folder that is not in the directory where you normally run commands.
   
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:

   ```yaml
   # 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 (page 170).

4. Upgrade the SSTables on each node to ensure that all SSTables are on the current version.

   ```bash
   $ nodetool upgradesstables
   ```

   This step is required for DataStax Enterprise upgrades that include a major Cassandra version changes.
Warning: Failure to upgrade SSTables when required results in a significant performance impact and increased disk usage.

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 (page 170).

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 to a folder that is not in the directory where you normally run commands.

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

9. Upgrade order matters. Upgrade nodes in this order:
   • In multiple datacenter clusters, upgrade every node in one datacenter before upgrading another datacenter.
   • Upgrade the seed nodes within a datacenter first.

   For DSE Analytics nodes using DSE Hadoop, upgrade the Job Tracker node first. Then upgrade Hadoop nodes, followed by Spark nodes.

   • Upgrade nodes in this order:
   a. DSE Analytics datacenters
b. Transactional/DSE Graph datacenters

c. DSE Search datacenters

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. To flush the commit log of the old installation:

$$\text{nodetool } -h \text{ hostname drain}$$

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

**Important:** This step is mandatory when upgrading between major Cassandra versions that change SSTable formats, rendering commit logs from the previous version incompatible with the new version.

11. Stop the node (4.0).

12. Use the appropriate method to install the new product version:

- RHEL: 4.0, 4.5
- Debian: 4.0, 4.5
- Tarball: 4.0, 4.5

13. To configure the new 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. Verify that the upgraded datacenter names match the datacenter names in the keyspace schema definition:

$$\text{nodetool status}$$

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

Warnings, errors, and exceptions are frequently found in the logs when starting 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.

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

18. When the upgrade includes a major Cassandra version, you must upgrade the SSTables. DataStax recommends upgrading the SSTables on one node at a time or when using racks, one rack at a time.

   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.

   $ nodetool upgradesstables

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

Use the --jobs option to set the number of SSTables that upgrade simultaneously. The default setting is 2, which minimizes impact on the cluster. Set to 0 to use all available compaction threads. DataStax recommends running the upgradesstables command on one node at a time or when using racks, one rack at a time.

   Note: You can run the upgradesstables command before all the nodes are upgraded as long as you run this command on only one node at a time or when using racks, one rack at a time. Running upgradesstables on too many nodes will degrade performance.

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.

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&amp;name=wiki.solr&amp;reindex=false&amp;deleteAll=false"

3. Upgrade the cluster.

4. Reindex 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

**Attention: Read and understand these instructions before upgrading.** Carefully reviewing the planning ([page 9](#)) and upgrading instructions can ensure a smooth upgrade and avoid pitfalls and frustrations.

### Cassandra version change

Upgrading to DataStax Enterprise 3.2 includes a major Cassandra version change. Be sure to follow the recommendations for upgrading the SSTables.

### General recommendations

DataStax recommends backing up your data prior to any version upgrade, including logs and custom configurations. 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 during an upgrade

- **Do not** enable new features.
- Do not run `nodetool repair`. If you have the OpsCenter Repair Service configured, turn off the Repair Service.
- Ensure OpsCenter compatibility. See [DataStax OpsCenter compatibility](#) with DataStax Enterprise.
- During the upgrade, do not bootstrap or decommission nodes.
- Do not issue these types of CQL queries during a rolling restart: **DDL** and **TRUNCATE**.
- Do not enable Change Data Capture (CDC) on a mixed-version cluster. Upgrade all nodes to DSE 5.1 or later before enabling CDC.
- 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.

**Note:** Nodes on different versions might show a schema disagreement during an upgrade.
Upgrading DataStax Enterprise

- NodeSync waits to start until all nodes are upgraded.
- Do not enable Change Data Capture (CDC) on a mixed-version cluster. Upgrade all nodes before enabling CDC.
- Ensure OpsCenter compatibility. OpsCenter 6.7 is required for managing DSE 6.7 clusters. See DSE OpsCenter compatibility with DataStax Enterprise.

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

Upgrading drivers and possible impact when driver versions are incompatible
Be sure to check driver compatibility. Depending on the driver version, you might need to recompile your client application code. See DataStax driver changes. 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:

  - **Protocol version**: Because some drivers can use different protocol versions, force the protocol version at start up. For example, keep the Java driver at its current protocol version while the driver upgrade is happening. Switch to the Java driver to the new protocol version only after the upgrade is complete on all nodes in the cluster.
  - **Initial contact points**: 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 details on protocol version negotiation, see protocol versions with mixed clusters in the Java driver version you’re using, for example, Java driver.

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

**Tip:** The DataStax installer (page 184) 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.

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 DataStax Enterprise 3.0 to 3.2 (page 177)
   • Upgrading from DataStax Enterprise 2.2 to 3.2 (page 180)

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 Solr (page 180) restrictions.

7. Upgrade the SSTables on each node to ensure that all SSTables are on the current version.

   $ nodetool upgradesstables

   This step is required for DataStax Enterprise upgrades that include a major Cassandra version changes.

   **Warning:** Failure to upgrade SSTables when required results in a significant performance impact and increased disk usage.

   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 to a folder that is not in the directory where you normally run commands.
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 transactional 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 -h hostname drain
```

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 on a supported platform:

- RHEL: 3.2
- Debian: 3.2
- Tarball: 3.2

   **Note:** Install the new product version using the same installation method that is on the system. The upgrade proceeds with installation regardless of the installation method and might result in issues.

4. To configure the new 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.

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

   ```
   $ nodetool status
   ```

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

   Warnings, errors, and exceptions are frequently found in the logs when starting 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: "127605887595351923798765477786913079296"
   . . .
   ```

   - 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:

   ```
   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).

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" ))
```

10. Repeat the upgrade on each node in the cluster following the recommended 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 DSE Analytics nodes or datacenters first, then Cassandra nodes or
     datacenters, and finally DSE Search nodes or datacenters.
   - For DSE Analytics nodes, upgrade the Job Tracker node first. Then upgrade
     Hadoop nodes, followed by Spark nodes.

11. 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 /etc/cassandra/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.

12. 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
```

13. When the upgrade includes a major Cassandra version, you must upgrade the SSTables. DataStax recommends upgrading the SSTables on one node at a time or when using racks, one rack at a time.

   **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.

```
$ nodetool upgradestables
```

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

Use the --jobs option to set the number of SSTables that upgrade simultaneously. The default setting is 2, which minimizes impact on the cluster. Set to 0 to use all available compaction threads. DataStax recommends running the upgradestables command on one node at a time or when using racks, one rack at a time.

   **Note:** You can run the upgradestables command before all the nodes are upgraded as long as you run this command on only one node at a time or when using racks, one rack at a time. Running upgradestables on too many nodes will degrade performance.

**Upgrading from DataStax Enterprise 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 could 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 might see a security related error message (documented below). You will see the following message in the log when the node has completed the migration:

```
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 DataStax Enterprise 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.

```bash
$ 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

  ```bash
  $ cd install_location/demos
  ```

- Package installation

  ```bash
  $ 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:

  ```bash
  $ curl -v --data-binary @solrconfig.xml -H 'Content-type:text/xml; charset=utf-8'
  ```

- From the wikipedia directory:

  ```bash
  $ curl -v --data-binary @solrconfig.xml -H 'Content-type:text/xml; charset=utf-8'
  ```

- From the log_search directory:
Upgrading DataStax Enterprise

$ curl -v --data-binary @solrconfig.xml -H 'Content-type:text/xml; charset=utf-8'

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 could 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 might see a security related error message (documented below). You will see the following message in the log when the node has completed the migration:

```
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**
   - `authenticator` 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 4.5-5.1 using the DataStax installer**

These steps show how to upgrade to versions of DataStax Enterprise (DSE) up to 5.1 using the GUI installer.

**Note:** The DataStax installer has been removed in DSE 6.0 and later.

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

- Drains the currently running node.
- Preserves the configuration files and places them in a backup directory.
- Removes previously installed packages.
- Updates 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
- DataStax Enterprise 5.0 for upgrading to 5.1

**Attention:** Read and understand these instructions before upgrading. Carefully reviewing the planning (page 9) and upgrading instructions can ensure a smooth upgrade and avoid pitfalls and frustrations.
Upgrading Linux installations using the DataStax installer

1. Familiarize yourself with the changes and features in the release:
   • DataStax Enterprise release notes for 5.1, 5.0, and 4.8.
   • 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.

<table>
<thead>
<tr>
<th>Current version</th>
<th>Upgrade version</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataStax Enterprise 5.0</td>
<td>DataStax Enterprise 5.1</td>
</tr>
<tr>
<td>DataStax Enterprise 4.7 or 4.8</td>
<td>DataStax Enterprise 5.0</td>
</tr>
<tr>
<td>DataStax Enterprise 4.0, 4.5, or 4.6</td>
<td>DataStax Enterprise 4.8</td>
</tr>
<tr>
<td>DataStax Community or open source Apache Cassandra™ 2.0.x</td>
<td>DataStax Enterprise 4.8</td>
</tr>
<tr>
<td>DataStax Community 3.0.x</td>
<td>DataStax Enterprise 5.1 (page 189)</td>
</tr>
<tr>
<td>DataStax Distribution of Apache Cassandra™ 3.x</td>
<td>DataStax Enterprise 5.0 (page 133)</td>
</tr>
</tbody>
</table>

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 and later**

The JDK is recommended for development and production systems, and 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 or use the following command:
$ curl --user dsa_email_address:password -O https://
downloads.datastax.com/enterprise/DataStaxEnterprise-5.1.x-linux-x64-
installer.run

For example, alice can run this command to download DSE 5.1.0:

$ curl --user alice@email.com:password -O https://
downloads.datastax.com/enterprise/DataStaxEnterprise-5.1.0-linux-x64-
installer.run

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 5.1, 5.0, or 4.8.

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. For
DataStax Enterprise 5.1, see the OpsCenter 6.1 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

1. Uninstall all DataStax Enterprise packages.

   • Debian and Ubuntu

     $ sudo apt-get purge "dse-*" "datastax-*"
• **RHEL and CentOS**

```bash
$ sudo yum remove "dse-*" "datastax-*"
```

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`.

   ```bash
   $ sudo cd data_directory_location/keyspace_name/table_name/
   snapshots/snapshot_name
   $ sudo 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 DSE Search, rebuild the index (4.5, 4.8, 5.0, 5.1, 6.0).

**Related information:**
- Uninstalling DataStax Enterprise 5.1 ([page](#))
- Uninstalling DataStax Enterprise 5.0 ([page](#))
- Uninstalling DataStax Enterprise 4.8 ([page](#))
- Uninstalling DataStax Enterprise 4.5 ([page](#))

### Revert to a previous version from a tarball installation

1. Rename the current installation directory.

   ```bash
   # 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`.

   ```bash
   $ sudo cd data_directory_location/keyspace_name/table_name/
   snapshots/snapshot_name
   ```
3. Copy the old cassandra.yaml file from the old install directory to the new one.

```bash
$ 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 DSE Search, rebuild the index (4.5, 4.8, 5.0, 5.1, 6.0).

Related information:
- Uninstalling DataStax Enterprise 5.1 (page)
- Uninstalling DataStax Enterprise 5.0 (page)
- Uninstalling DataStax Enterprise 4.8 (page)
Upgrading from Apache Cassandra to DataStax Enterprise

Attention: Read and understand these instructions before upgrading. Carefully reviewing the planning (page 9) and upgrading instructions can ensure a smooth upgrade and avoid pitfalls and frustrations.

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 paths

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

<table>
<thead>
<tr>
<th>Upgrade from Apache Cassandra™</th>
<th>Upgrade to DataStax Enterprise</th>
<th>Required interim version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cassandra 3.0 and 3.11</td>
<td>DSE 6.7</td>
<td>Not required</td>
</tr>
<tr>
<td>Cassandra 3.0 and 3.11</td>
<td>DSE 6.0</td>
<td>Not required</td>
</tr>
<tr>
<td>Cassandra 3.0 and 3.11</td>
<td>DSE 5.1</td>
<td>Not required</td>
</tr>
<tr>
<td>Cassandra 3.0</td>
<td>DSE 5.0</td>
<td>Not required</td>
</tr>
<tr>
<td>Cassandra 2.1</td>
<td>DSE 5.0</td>
<td>DSE 4.8</td>
</tr>
<tr>
<td>Cassandra 2.0 and earlier</td>
<td></td>
<td>Cassandra 2.1</td>
</tr>
</tbody>
</table>

Questions? Contact DataStax Support.

General upgrade restrictions during an upgrade

- Do not enable new features.
- Do not run nodetool repair. If you have the OpsCenter Repair Service configured, turn off the Repair Service.
- Ensure OpsCenter compatibility. See DataStax OpsCenter compatibility with DataStax Enterprise.
- During the upgrade, do not bootstrap or decommission nodes.
- Do not issue these types of CQL queries during a rolling restart: DDL and TRUNCATE.
• Do not enable Change Data Capture (CDC) on a mixed-version cluster. Upgrade all nodes to DSE 5.1 or later before enabling CDC.

• 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.

**Note:** Nodes on different versions might show a schema disagreement during an upgrade.

• NodeSync waits to start until all nodes are upgraded.

• Do not enable Change Data Capture (CDC) on a mixed-version cluster. Upgrade all nodes before enabling CDC.

• Ensure OpsCenter compatibility. OpsCenter 6.7 is required for managing DSE 6.7 clusters. See [DSE OpsCenter compatibility](#) with DataStax Enterprise.

**Restrictions for nodes using any kind of security**

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

**Upgrading drivers and possible impact when driver versions are incompatible**

Be sure to check [driver compatibility](#). Depending on the driver version, you might need to recompile your client application code. See DataStax driver changes. 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:

• **Protocol version:** Because some drivers can use different protocol versions, force the protocol version at start up. For example, keep the Java driver at its current protocol version while the driver upgrade is happening. Switch to the Java driver to the new protocol version only after the upgrade is complete on all nodes in the cluster.

• **Initial contact points:** 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 details on protocol version negotiation, see protocol versions with mixed clusters in the Java driver version you’re using, for example, [Java driver](#).

**Upgrade order**

Upgrade order matters. In multiple datacenter clusters, upgrade every node in one datacenter before upgrading another datacenter.

1. Analytics datacenters: seed nodes first, and then the rest of the analytics nodes.

2. Cassandra (transactional) nodes or datacenters

3. Search nodes or datacenters

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.

Follow these steps on each node:

1. Before upgrading to DataStax Enterprise from any Apache Cassandra™ version, DataStax recommends backing up your data prior to any version upgrade, including logs and custom configurations. A backup provides the ability to revert and restore all the data used in the previous version if necessary. See Backing up and restoring Cassandra data.

2. Familiarize yourself with the changes and features in the release:
   • DataStax Enterprise release notes for 4.7, 4.8, 5.0, and 5.1.
   • General upgrade advice and Cassandra features in NEWS.txt/DSE CHANGES.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 Cassandra can be upgraded directly to the version of Cassandra that is used by DataStax Enterprise. See the Cassandra changes in CHANGES.txt/DSE CHANGES.txt.

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.

4. Run nodetool drain to flush the commit log of the old installation:
   $ nodetool -h hostname drain

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

5. Stop the node. (2.1, 2.2, 3.0)

6. Back up your configuration files.

   Back up your configuration to ensure that the configuration files are not overwritten with the default values.

7. Uninstall Cassandra.

   If you installed Cassandra from packages in APT or RPM repositories, you must remove the packages before setting up and installing DataStax Enterprise from the appropriate repository.
   • For packages installed from APT repositories:
Upgrading from Apache Cassandra to DataStax Enterprise

```
$ sudo apt-get autoremove "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
  ```

- When Cassandra was installed with a binary tarball:

  ```
  $ ps auwx | grep cassandra
  $ sudo kill cassandra_pid
  ```

  And then remove the Cassandra installation directory.

8. Install DataStax Enterprise using one of the following:

   - **RHEL:** 4.6, 4.7, 4.8, 5.0
   - **Debian:** 4.6, 4.7, 4.8, 5.0
   - **Tarball:** 4.6, 4.7, 4.8, 5.0

9. To configure the product, use your backup configuration files to merge any necessary modifications into the new configuration files.

10. Start the node:

    - Packages and Installer-Services installations: See *Starting DataStax Enterprise as a service* (4.8, 5.0, 5.1).
    - Installer-No Services and Tarball installations: See *Starting DataStax Enterprise as a stand-alone process* (4.8, 5.0, 5.1).

11. To ensure optimal performance, upgrade the SSTables on each node now that the upgrade is complete.

    ```
    $ nodetool upgradesstables
    ```

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

12. 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. Verify that the upgraded datacenter names still match the datacenter names that are used in the keyspace schema definition:

   $ nodetool status

14. Repeat the upgrade on each node in the cluster following the recommended order.
Upgrading DataStax Distribution of Apache Cassandra™ and Apache Cassandra™

This section describes how to upgrade DDAC and Apache Cassandra™.

Planning your DataStax Distribution of Apache Cassandra™ upgrade

The upgrade process for DataStax Distribution of Apache Cassandra™ (DDAC) provides minimal downtime (ideally zero). During this process, 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 DDAC 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 to provide access to product improvements and new features and reduce version impacts. Ensure that you repair your nodes regularly. Node repair ensures that data on a replica is consistent with data on other 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.

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

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

1. In multiple datacenter clusters, upgrade every node in one datacenter before moving on to other datacenters.

2. Upgrade the seed nodes within a datacenter first.

Upgrading DDAC

Review this information on upgrading DataStax Distribution of Apache Cassandra™ (DDAC) between patch (point) releases, such as upgrading from DDAC 5.1.11 to 5.1.12.
Attention: Read and understand these instructions before upgrading. Carefully reviewing the planning (page 194) and upgrading instructions can ensure a smooth upgrade and avoid pitfalls and frustrations.

General recommendations

Attention: DataStax recommends upgrading to the latest version of DDAC 5.1.15.

General restrictions and limitations during the upgrade process

Restrictions and limitations apply while a cluster is in a partially upgraded state. The cluster continues to work as though it were on the earlier version of DDAC until all of the nodes in the cluster are upgraded.

Restrictions for all nodes during an upgrade

- Do not enable new features.
- Do not run nodetool repair.
- Do not bootstrap or decommission nodes.
- 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.

Note: Nodes on different versions might show a schema disagreement during an upgrade.

Preparing to upgrade

1. Back up your data. DataStax recommends backing up your data prior to any version upgrade, including logs and custom configurations. A backup provides the ability to revert and restore all the data used in the previous version if necessary.

2. Verify your current product version:

   $ cassandra -v

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

   The required space depends on the compaction strategy. See Disk space.

4. Familiarize yourself with the changes and features in DDAC and Apache Cassandra™:

   - DDAC release notes for the upgrade version and complete all required actions.

5. Back up the configuration files to a folder separate from the installation folder.

   The configuration files are overwritten with default values during installation of the new version.
6. Run `nodetool repair` to ensure that data on each replica is consistent with data on other nodes.

Upgrade steps

The upgrade process for DDAC provides minimal downtime (ideally zero). During this process, 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 DDAC until all of the nodes in the cluster are upgraded.

1. Upgrade order matters. Upgrade nodes in this order:
   - In multiple datacenter clusters, upgrade every node in one datacenter before upgrading another datacenter.
   - Upgrade the seed nodes within a datacenter first.

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

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

   This step saves time when nodes start up after the upgrade.

3. Stop the node.

4. Install the new product version, Installing the DataStax Distribution of Apache Cassandra™.

5. To configure the new product version:

   a. Compare your backup configuration files files to the new configuration files:
      - Look for any deprecated, removed, or changed settings.
      - Be sure you are familiar with the Apache Cassandra™ and DDAC changes and features in the new release.
      - Check for any other configuration files that you might have changed. See configuration files.

   b. Merge the applicable modifications into the new version.

6. Start the node.

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

   ```
   $ nodetool status
   ```

8. Review the logs for warnings, errors, and exceptions.
Warnings, errors, and exceptions are frequently found in the logs when starting 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.

9. Repeat the upgrade and restart on each node in the cluster following the recommended upgrade order as described in 1 (page 196).

10. After the new version is installed on each node, DataStax recommends upgrading the SSTables on each node.

Upgrading SSTables is recommended for optimal performance, but is not required for patch releases.

```bash
$ nodetool upgradesstables
```

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

Use the `--jobs` option to set the number of SSTables that upgrade simultaneously. The default setting is 2, which minimizes impact on the cluster. Set to 0 to use all available compaction threads. For information about nodetool upgradesstables, including how to speed it up, see the DataStax Support KB article Nodetool upgradesstables FAQ.

Upgrading Apache Cassandra

DataStax supports upgrading from Apache Cassandra™ to DataStax Enterprise (page 189). For upgrading Apache Cassandra versions, see the Apache Cassandra website.
Upgrading DSE OpsCenter

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

Attention: Read and understand these instructions before upgrading. Carefully reviewing the planning and upgrading instructions can ensure a smooth upgrade and avoid pitfalls and frustrations.

Important: For managing DataStax Enterprise (DSE) version 6.7 clusters, you must upgrade to DSE OpsCenter 6.7. See the following table for compatibility with DSE versions:

<table>
<thead>
<tr>
<th>OpsCenter version</th>
<th>DSE version</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.7</td>
<td>6.7, 6.0, 5.1</td>
</tr>
<tr>
<td>6.5</td>
<td>6.0, 5.1, 5.0 (EOL)</td>
</tr>
<tr>
<td>6.1</td>
<td>5.1, 5.0 (EOL), 4.8 (EOSL)</td>
</tr>
<tr>
<td>6.0</td>
<td>5.0 (EOL), 4.8 (EOSL), 4.7 (EOSL)</td>
</tr>
</tbody>
</table>

OpsCenter uses definition files to enable support for multiple, supported versions of DSE without the need to immediately upgrade the currently installed version of OpsCenter. DataStax recommends keeping current with the latest major version releases to take advantage of new and improved features, enhanced performance, and fixed bugs or known issues.

In OpsCenter versions 6.1 and later, the OpsCenter and agent compatibility awareness feature notifies you of any potential features that might not operate on older agent versions.

Note: OpsCenter 6.0 and later does not support Apache Cassandra™. For more information, see DSE OpsCenter policy changes.

Before upgrading DSE OpsCenter

For any major release, review the changes in features, configuration files, metrics, and APIs impacting upgrades to OpsCenter.

Important: DataStax strongly recommends upgrading to the most recent version of OpsCenter, which is 6.7.3. Upgrading to this version ensures that you take advantage of new and improved features, enhanced performance, and fixed bugs or known issues.

See New features for a description of features added in each OpsCenter release.
Upgrade considerations

Before upgrading to OpsCenter 6.7.3, carefully consider the following changes and follow any prescribed actions.

Support for multiple user roles using LDAP authentication

Users can have multiple roles when using LDAP authentication. If the list of a user’s groups map to more than one role in OpsCenter, each user will be granted each of the listed roles, and their resulting OpsCenter permissions will be merged for all of their roles. See Adding a role for an LDAP user.

**Important:** In OpsCenter 6.1.10 and later and OpsCenter 6.5.3 and later, you must update custom scripts and applications that use the OpsCenter API when using multiple user roles with LDAP authentication. If a custom script or application that uses the OpsCenter API did not account for multiple user roles, and a user has multiple roles, the script or application will fail because the `role` attribute cannot be found. The single `role` attribute will be provided for users that have only one role. If your application or script has users with only one role, then updates are not required for continued use.

LCM now installs OpenJDK as the default option

Starting in January 2019, Oracle is ending public updates for the Oracle Java 8 runtime, requiring commercial users to purchase a paid license to obtain updates.

To simplify licensing going forward, LCM in OpsCenter 6.7.0 is introducing support for installing OpenJDK as the default option. LCM will install OpenJDK as distributed by operating system vendors through `.deb` or `.rpm` packages. Existing clusters will be migrated to OpenJDK unless Oracle is explicitly selected as the Java vendor prior to running the first install or upgrade job from within OpsCenter 6.7.0.

To understand how this change effects your OpsCenter installation, see Choosing a Java vendor in Lifecycle Manager.

DSE OpsCenter upgrade instructions

Complete the following instructions to upgrade OpsCenter. Select the upgrade instructions that apply to your installation type.

Upgrading DSE OpsCenter package installations

These steps provide information on upgrading to OpsCenter 6.7.3 for package installs and restarting the `opscenterd` daemon.
Important: Upgrading from OpsCenter 6.1 to 6.7.3 requires an interim upgrade to OpsCenter 6.5. Upgrade from OpsCenter 6.1 to the most current release of OpsCenter 6.5 before upgrading to OpsCenter 6.7.3.

Prerequisites:

Review Upgrade considerations (page 198) for changes in configuration files, metrics, and APIs that affect upgrades to OpsCenter 6.7.3.

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

Additionally, view the compatibility table to ensure that OpsCenter is compatible with your version of DataStax Enterprise.

1. If you previously configured the api_port under the [cassandra] or [storage_cassandra] headers in cluster_name.conf, delete the entries.

   Caution: If this entry still exists under the [cassandra] or [storage_cassandra] headers, OpsCenter will not start.

2. On the OpsCenter daemon host, run the appropriate command to update the packages:

   • Debian or Ubuntu

     $ sudo apt-get update

   • RHEL or CentOS

     Note: Yum automatically updates package lists when necessary.

3. Install the upgraded OpsCenter package:

   Replace version_number shown in the example below with the version to which you are upgrading.

   • Debian or Ubuntu:

     $ sudo apt-get install opscener=version_number

   • RHEL or CentOS:

     $ sudo yum install opscener-version_number

4. If the package manager prompts you for options regarding opscenerd.conf, keep your currently installed version.

5. Restart the OpsCenter daemon.
$ sudo service opscenterd restart

What's next: Upgrade DataStax Agents (page 202) from the OpsCenter interface.

**Upgrading DSE OpsCenter tarball installations**

These steps provide information on upgrading to OpsCenter 6.7.3 using the OpsCenter tarball and restarting the opscenterd daemon.

**Important:** Upgrading from OpsCenter 6.1 to 6.7.3 requires an interim upgrade to OpsCenter 6.5. Upgrade from OpsCenter 6.1 to the most current release of OpsCenter 6.5 before upgrading to OpsCenter 6.7.3.

**Prerequisites:**

Review Upgrade considerations (page 198) for changes in configuration files, metrics, and APIs that affect upgrades to OpsCenter 6.7.3.

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

Additionally, view the compatibility table to ensure that OpsCenter is compatible with your version of DataStax Enterprise.

1. If you previously configured the api_port under the [cassandra] or [storage_cassandra] headers in cluster_name.conf, delete the entries.

   **Caution:** If this entry still exists under the [cassandra] or [storage_cassandra] headers, OpsCenter will not start.

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
   - conf/opscenterd.conf
   - ./passwd.db
   - ./lcm.db
   - ./keys/lcm.key

   For example, the following command copies the files in the /conf/clusters/* directory from the old tarball location to the new one:
Upgrading DSE OpsCenter

4. If SSL is enabled, copy the contents of the SSL configuration directory:
   `install_location/ssl/*`.

5. If `opscenterd` is running, stop the instance and start it from the new tarball installation directory.

What's next: Upgrade DataStax Agents (page 203) manually from tarballs.

Upgrading DSE OpsCenter when failover is enabled

Follow this process when upgrading OpsCenter and failover is enabled.

1. Stop the secondary (backup) OpsCenter instance.

2. Upgrade the primary OpsCenter instance:
   a. Stop OpsCenter.
   b. Upgrade OpsCenter using the package (page 199) or tarball (page 201) instructions as appropriate.
   c. Start OpsCenter.

3. Repeat 2 (page 202) to upgrade the secondary OpsCenter instance.

4. Start the secondary OpsCenter instance.

What's next: Upgrade DataStax Agents (page 202) from the OpsCenter UI or from tarballs.

Upgrading DataStax Agents

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

For more information, see Installing DataStax Agents.

Upgrading DataStax Agents automatically from the OpsCenter interface

If DataStax Agents require upgrading, use the OpsCenter interface to automatically upgrade them.

Important: All nodes in the cluster must use the same SSH keys when automatically upgrading the DataStax Agents. If nodes use different SSH keys, either normalize the SSH keys across all nodes, or manually install the DataStax Agents.
In OpsCenter versions 6.1 and later, the OpsCenter and agent compatibility awareness feature notifies you of any potential features that might not operate on older agent versions.

1. In the OpsCenter interface, click cluster_name > Nodes, and then click the Agents tab.

   A message displays indicating the number of DataStax Agents running an earlier version that might have incompatibilities with OpsCenter 6.7.3.

2. In the DataStax Agents Status View of the OpsCenter interface, click Upgrade all agents.

   The Set Up Agents window appears.

3. Choose whether to Install or start agents automatically or Install agents manually.

Upgrading DataStax Agents manually from tarballs

If upgrading the DataStax Agents manually with tarballs:

1. Copy the new agent.tar.gz file to all nodes in the cluster:

   $ scp agent.tar.gz node_name

2. Extract the tarball:

   $ tar -xvf agent.tar.gz

3. Copy the following files in the tarball from the old DataStax Agent tarball directories to the new ones:

   $ scp -r old_tarball_directory/conf/* new_tarball_directory/conf/*
   $ scp -r old_tarball_directory/ssl/* new_tarball_directory/ssl/*
Upgrading DataStax Studio

Upgrading DataStax Studio

Notebooks created in earlier versions of DataStax Studio can be used with new Studio versions:

- When the new version of Studio starts, all notebooks created in an earlier version are automatically upgraded.
- Notebooks are not impacted, lost, or corrupted with the upgrade.
- After the notebooks are upgraded, they are no longer compatible with the earlier version.

To upgrade an earlier version of DataStax Studio to DataStax Studio 6.0 or 6.7:

1. Back up the `user_home_directory/.datastax_studio` directory.
   
   For information about this directory, see User data in DataStax Studio 6.0 | 6.7.

2. Install DataStax Studio 6.7 | 6.0.

3. Start DataStax Studio 6.7 | 6.0.
Upgrading the DataStax AMI

DataStax no longer hosts the DataStax ComboAMI. You can install DataStax Enterprise in two ways:

- **Create your instances using an AMI for a supported platform and from a trusted source.** Then use the appropriate install method 6.7 | 6.0 | 5.1 for your platform.
- **Use the Lifecycle Manager in OpsCenter to easily provision a DataStax Enterprise cluster for versions 4.7 and later:**

1. Create your instances using an AMI for a supported platform and from a trusted source.

2. Use the Lifecycle Manager: 6.7 | 6.5 | 6.1 to provision and configure your cluster.