# Table of Contents

About OpsCenter ................................................................. 11

New features .................................................................. 11
Key features .................................................................. 16
OpsCenter Labs features ......................................................... 18
  Exporting and importing dashboard presets ....................... 18
  Configuring named route linking ................................. 20
  Sending OpsCenter metrics to a Graphite server .......... 21
  Viewing logs from node details .................................... 22
  Bulk uploading S3 backups using the AWS CLI ............... 24

OpsCenter architecture overview ........................................ 25
DSE OpsCenter Policy Changes ........................................... 26
Send DataStax feedback about OpsCenter ........................... 26

Release notes .................................................................. 28
Release impacts ................................................................. 28
OpsCenter 6.1.11 release notes ........................................... 28
OpsCenter 6.1.10 release notes ........................................... 29
OpsCenter 6.1.9 release notes ............................................. 32
OpsCenter 6.1.8 release notes ............................................. 32
OpsCenter 6.1.7 release notes ............................................. 33
OpsCenter 6.1.6 release notes ............................................. 35
OpsCenter 6.1.5 release notes ............................................. 37
OpsCenter 6.1.4 release notes ............................................. 39
OpsCenter 6.1.3 release notes ............................................. 39
OpsCenter 6.1.2 release notes ............................................. 41
OpsCenter 6.1.1 release notes ............................................. 43
OpsCenter 6.1.0 release notes ............................................. 47

Known and resolved issues for OpsCenter 6.1 and later ........ 51
Installing DSE OpsCenter 6.1

Installing OpsCenter from the RPM package
Installing OpsCenter from the Debian package
Installing OpsCenter with a tarball on any Linux distribution
Installing OpsCenter on Docker
Uninstalling OpsCenter
Installing DataStax Agents
Installing DataStax agents automatically
Installing DataStax agents manually
Setting permissions to run the agent as the DSE user
Configuring JAVA_HOME for DataStax agents

OpsCenter Reference

OpsCenter ports reference
Installation and configuration locations
Starting, stopping, and restarting OpsCenter
Stopping, starting, and restarting DataStax agents

Upgrading OpsCenter

Configuring OpsCenter

OpsCenter Security

OpsCenter SSL overview
Configuring OpsCenter role-based security
Encrypting sensitive configuration values
Authenticating with LDAP
Kerberos authentication
Configuring security logging

Configuring alerts for events
SNMP alerts overview
Enabling SMTP email alerts
Enabling alerts posted to a URL..........................................................145
Configuring data collection and expiration.........................................151
Controlling data collection..................................................................151
Storing collection data on a separate cluster........................................154
Sending OpsCenter metrics to a Graphite server.................................162
OpsCenter DSE definitions files updates..............................................163
  Updating and configuring definitions files properties..........................164
Automatic failover overview...............................................................166
  Enabling automatic failover...............................................................169
Configuring named route linking.........................................................173
Configuring the OpsCenter JVM.........................................................174
Configuring the DataStax Agent JVM................................................176
  Setting and securing the tmp directory for the agent.........................176
  Encrypting JMX communications....................................................177
Changing the replication strategy for the OpsCenter keyspace...............179
Configuration files for OpsCenter.......................................................180
  OpsCenter configuration properties..................................................182
  Cluster configuration properties.....................................................196
  DataStax Agent configuration.........................................................205
  OpsCenter logback.xml configuration..............................................215
Customize scripts for starting and stopping DataStax Enterprise..........217
Example configuration scenarios......................................................218
  Configuring for multiple regions....................................................218
  Configuring for very large clusters................................................220
Using OpsCenter..............................................................................222
  OpsCenter workspace overview......................................................222
    Ring View..................................................................................224
    List View..................................................................................228
Cluster administration................................................................................................................................. 230
Starting and stopping DSE............................................................................................................................ 231
Monitoring node operations......................................................................................................................... 238
Configuring alerts........................................................................................................................................... 239
Adding an alert for agent issues................................................................................................................... 238
Adding an alert for down nodes..................................................................................................................... 242
Configuring an alert for KMIP errors........................................................................................................... 242
Configuring an alert for percentage of in-memory usage............................................................................ 243
Monitoring node operations......................................................................................................................... 245
Viewing the Spark Console............................................................................................................................ 245
Monitoring in-memory usage....................................................................................................................... 247
Viewing logs from node details...................................................................................................................... 249
Managing and maintaining nodes.................................................................................................................. 251
Running cleanup............................................................................................................................................ 251
Performing garbage collection....................................................................................................................... 252
Running compaction..................................................................................................................................... 253
Flushing tables............................................................................................................................................... 254
Decommission a node..................................................................................................................................... 256
Draining a node.............................................................................................................................................. 257
Moving a node................................................................................................................................................. 259
Running a manual repair............................................................................................................................... 260
Configure an alias for a node......................................................................................................................... 262
Starting and stopping DSE.............................................................................................................................. 263
Starting DSE on a node.................................................................................................................................... 263
Stopping DSE on a node................................................................................................................................. 263
Restarting DSE on a node................................................................................................................................. 264
Managing keyspaces and tables..................................................................................................................... 265
Keyspaces....................................................................................................................................................... 265
Browsing data deprecated.............................................................................................................................. 274
Cluster administration................................................................................................................................. 274
Adding an existing cluster......................................................... 274
Disconnecting a cluster from OpsCenter and Lifecycle Manager......... 280
Rebalancing a cluster overview.................................................. 281
Restarting a cluster..................................................................... 283
Changing the display name of a cluster........................................ 285
Downloading diagnostic data......................................................... 286
Generating a cluster report.......................................................... 323
Dashboard performance metrics.................................................. 324
  Performance metrics overview................................................... 324
  Cluster performance metrics..................................................... 331
  Pending task metrics............................................................... 332
  Table performance metrics....................................................... 335
  Tiered storage performance metrics......................................... 337
Message latency metrics............................................................. 341
Search performance metrics....................................................... 345
Graph metrics............................................................................ 346
Operating system performance metrics....................................... 349
Alert metrics.............................................................................. 351
OpsCenter Metrics Tooltips Reference........................................ 358
DSE Management Services.......................................................... 380
  Backup Service......................................................................... 380
  Backup Service overview.......................................................... 380
  Why use the Backup Service?..................................................... 384
  Quick Video Tour: Backup Service.............................................. 384
  Configuring the Backup Service............................................... 384
  Backing up a cluster................................................................. 391
  Restoring a cluster.................................................................. 414
  Cloning cluster data............................................................... 429
Disabling the OpsCenter Performance Service................................. 527
Setting permissions for the OpsCenter Performance Service.............. 527
Tuning a database cluster with the Performance Service.................. 528
Troubleshooting OpsCenter................................................................ 548
Lifecycle Manager.............................................................................. 549
Lifecycle Manager overview................................................................. 549
Accessing OpsCenter Lifecycle Manager.............................................. 550
Getting started workflow.................................................................. 552
Lifecycle Manager: Clusters................................................................. 554
Cluster topology overview................................................................. 554
Importing an unmanaged cluster......................................................... 569
Cloning a managed cluster, datacenter, or node................................. 571
Run an Install and Configure Jobs Overview...................................... 572
Creating custom data directories......................................................... 578
Downloading the generated CA cert..................................................... 579
Lifecycle Manager: Jobs............................................................ 580
Monitoring Job Status Overview........................................................ 580
Viewing job details............................................................................ 582
Configuring SSH connection and idle timeout thresholds for LCM jobs. 585
Lifecycle Manager: Repositories......................................................... 586
Repositories Overview....................................................................... 586
Adding a repository........................................................................... 587
Editing a repository........................................................................... 590
Deleting a repository.......................................................................... 590
Lifecycle Manager: SSH Credentials................................................... 591
SSH Credentials Overview................................................................. 591
Adding SSH credentials..................................................................... 591
Editing SSH credentials.................................................................... 593
OpsCenter API reference for developers

Lifecycle Manager: Configuration Profiles

Configuration profiles overview

Configuration Known Issues and Limitations

Managing DSE Security using LCM

Adding a configuration profile

Editing a configuration profile

Deleting a configuration profile

Cloning a configuration profile

Checking the usage of a configuration profile

Lifecycle Manager configuration options

OpsCenter API reference for developers

Enable and access the DataStax Agent API

DataStax Agent API example curl commands
About OpsCenter

DSE OpsCenter is a visual management and monitoring solution for DataStax Enterprise. OpsCenter provides architects, database administrators, and operations staff with the capabilities to intelligently and proactively ensure their database clusters are running optimally. OpsCenter also simplifies administration tasks such as:

- Adding and expanding clusters
- Configuring nodes
- Viewing performance metrics
- Rectifying issues
- Monitoring cluster health on the dashboard

OpsCenter Lifecycle Manager (page 549) (LCM) provisions DataStax Enterprise (DSE) clusters and centrally manages cluster configurations.

The OpsCenter Compatibility chart indicates which versions of OpsCenter are compatible with the various DataStax Enterprise versions.

New features in DSE OpsCenter 6.1

New features

The following new and improved features are highlighted for the current DataStax Enterprise (DSE) OpsCenter version 6.1 release.

<table>
<thead>
<tr>
<th>DataStax Enterprise 5.1</th>
<th>Support for provisioning and monitoring DataStax Enterprise 5.1 clusters:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Added support for DSE Unified Authentication for JMX for DSE 5.1 and later. See Configuring a JMX Connection using LCM (page 615).</td>
</tr>
<tr>
<td></td>
<td>Added LCM support for row-level access control (page 613) (RLAC).</td>
</tr>
<tr>
<td></td>
<td>Added support in LCM Config Profiles to configure the jvm.options file for DSE 5.1 and later. The options were moved from cassandra-env.sh to jvm.options. See Configuring JVM options in LCM (page 616).</td>
</tr>
<tr>
<td></td>
<td>The schema viewer in the Data area of the OpsCenter Monitoring UI now displays materialized views, secondary indexes, and user-defined types used on a table. Users with permissions can view query statements on the new User-Defined Types (page 267), User-Defined Functions (page 268), and User-Defined Aggregates (page 269) tabs.</td>
</tr>
<tr>
<td></td>
<td>Removed support for Hadoop; deprecated in DSE 5.0 and 6.1 of OpsCenter.</td>
</tr>
</tbody>
</table>
## About OpsCenter

<table>
<thead>
<tr>
<th>Repair Service (page 462)</th>
<th>Overhauled the OpsCenter Repair Service, including refactoring for reliability and performance:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• New Repair Service Status (page 474) dashboard for complete transparency into progress of subrange and incremental repair processes, including remaining time and data to complete a repair cycle.</td>
</tr>
<tr>
<td></td>
<td>• Improved handling around repairs when nodes are down.</td>
</tr>
<tr>
<td></td>
<td>• Restrict incremental repairs by datacenter and racks (page 479) for multi-DC clusters with replicated keyspaces.</td>
</tr>
<tr>
<td></td>
<td>• Ignore specific keyspaces and tables (page 483) from the subrange repair process.</td>
</tr>
<tr>
<td></td>
<td>• Automatically throttles subrange repairs (page 488) when appropriate.</td>
</tr>
<tr>
<td></td>
<td>• Exposed and improved error messages.</td>
</tr>
<tr>
<td></td>
<td>• Fixed critical bugs as noted in the 6.1.10 Release Notes.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Backup Service (page 380)</th>
<th>Added features for flexible backup and restore workflows:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Ability to enable backup failure alerts as an event to the Event log, which can be configured to alert as an email or posted URL to a chat room.</td>
</tr>
<tr>
<td></td>
<td>• Back up and restore a datacenter.</td>
</tr>
<tr>
<td></td>
<td>• Added a rule to the Best Practice Service when Commit Log Archiving is inconsistent.</td>
</tr>
<tr>
<td></td>
<td>• Added an unset throttle warning (page 418) when restoring a backup.</td>
</tr>
<tr>
<td></td>
<td>• Cloning a backup from a Local FS location.</td>
</tr>
<tr>
<td></td>
<td>• Point-in-time restores from a Local FS location.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DataStax agents improvements</th>
<th>Improved flexibility for managing DataStax agents:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Added backward-compatibility with agent versions that allow similar but different agent versions to co-exist with the installed OpsCenter version without loss of feature service. This feature provides flexibility with regard to maintaining a DSE cluster within OpsCenter.</td>
</tr>
<tr>
<td></td>
<td>• Exposed the DataStax Agent API in a Swagger UI console. See [Enable and access the DataStax Agent API](page 627).</td>
</tr>
<tr>
<td></td>
<td>• Enhanced UI design for conveying Agents Status (page 230):</td>
</tr>
<tr>
<td></td>
<td># Vastly improved visibility of agent version and agent compatibility, including better built-in guidance on when it is imperative that all agents get upgraded.</td>
</tr>
<tr>
<td></td>
<td># Tooltips explain each column purpose in the Agent Status.</td>
</tr>
<tr>
<td></td>
<td># A red icon ![ ] appears on the Agents tab for increased visibility when agents are unhealthy.</td>
</tr>
<tr>
<td></td>
<td># Replaced text in each status column and row with intuitive icons with tooltips for faster scanning of agent status.</td>
</tr>
</tbody>
</table>
New metrics

- Added OpsCenter support for DSE Graph metrics (page 346) for dashboard graphs (page 347) and alerts (page 348). Available for DSE versions 5.0 and later.
- Messaging latency metrics (page 341) for both dashboard graphs (page 341) and alerts (page 343) at the datacenter and node levels.

Best Practice Service (page 502)

- Added a rule for Commit Log Archiving Setting Enabled Consistency Best Practice Rule. The rule ensures new nodes added to a cluster with Commit Log Archiving (CLA) enabled also have CLA enabled, which helps prevent data loss particularly during a Point-in-Time restore using the Backup Service. The rule checks to make sure all nodes within a cluster have Commit Log Archiving enabled.

Access Logs from the OpsCenter UI

- Access critical logs from within the Node Details dialog in OpsCenter monitoring. These logs are a labs feature. See Viewing logs from node details (page 22).

Lifecycle Manager (LCM) (page 549)

- LCM now runs jobs concurrently across different clusters for faster job completion.

Updates from OpsCenter 6.0

The following changes are updates from the OpsCenter 6.0 major releases.

Support for using EC2 IAM roles for OpsCenter S3 backups

- Added support to use system default credentials for Amazon S3 backups as described in Working with AWS Credentials. See Adding an Amazon S3 backup location for instructions on adding an Amazon S3 backup location.

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, the user will be granted each of the listed roles, and their resulting OpsCenter permissions will be the merging of permissions 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 if you want to use 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.

Hadoop deprecated
About OpsCenter

DataStax has deprecated support for Hadoop as of DSE 5.1 and OpsCenter 6.1. DataStax Support can discuss alternative options for your organization.

Attempts to import a DSE cluster that has Hadoop enabled will fail. Lifecycle Manager can only import clusters that do not have Hadoop datacenter workloads. LCM cannot run jobs on clusters, datacenters, or nodes that have Hadoop enabled. Please contact DataStax Support for assistance with disabling Hadoop before retrying an import or attempting to run the job again.

The hive-site.xml configuration file has been relocated to the Spark section of the Configuration Profiles in LCM. The Hadoop section has been removed from LCM Config Profiles.

The Vnodes enabled on Hadoop nodes Best Practice Rule has been removed from OpsCenter monitoring.

Deprecated Hadoop-related configuration

The following port configuration option has been removed from opscneterd.conf: [hadoop]
base_job_tracker_proxy_port.

The following configuration options have been removed from cluster_name.conf:

- [hadoop] job_tracker_http_port
- [hadoop] job_tracker_port
- [hadoop] job_tracker_proxy_port
- [kerberos] job_tracker_client_principal

The following agent configuration options have been removed from address.yaml:

- hadoop_conf_location
- hadoop_log_location

Repair Service configuration changes

The Repair Service features have been significantly simplified and improved from both a usability and reliability standpoint. For complete details, see the Repair Service documentation in OpsCenter.

The following configuration options have been removed:

- incremental_range_repair
- ks_update_period
- max_err_threshold
- alert_on_repair_failure
- repair_estimation_factor

The following configuration options have been added:

- incremental_sleep
- incremental_threshold
• prioritization_page_size
• offline_splits
• ignore_keyspaces
• ignore_table
• time_to_completion_target_percentage
• incremental_repair_datacenters
• tokenranges_http_timeout
• max_down_node_retry

Backup Service configuration changes

OpsCenter uses the Amazon SDK for S3 functionality. The [cloud] accepted_certs configuration option is no longer necessary and has been removed from opscenterd.conf.

The backup file queue has been modified into a backup job queue. The backup_file_queue_size configuration option has been removed from address.yaml.

The following configuration options were added to opscenterd.conf under [backup_service] for the Amazon S3 proxy:

• s3_proxy_host
• s3_proxy_port

Improvements to the S3 throttle feature available in the Add Location dialog in the Backup Service have deprecated the following options:

• max-seconds-to-sleep
• seconds-to-read-kill-channel
• multipart-chunk-size
• read-buffer-size
• write-buffer-size

The options have been removed from address.yaml.

Best Practice Service Rule changes

The following changes were made to the Best Practice Service:

• The Commit Log Archiving Setting Enabled Consistency rule has been added. The rule ensures new nodes added to a cluster with Commit Log Archiving (CLA) enabled also have CLA enabled to prevent any potential data loss, particularly during a Point-in-Time restore using the Backup Service. If the Commit Log Archiving setting is not consistent across all nodes in a cluster, the rule fails. For more information, see Configuring commit log backups.
• The Vnodes enabled on Hadoop nodes Best Practice Rule has been removed.

Deprecated agent install script and configuration option
About OpsCenter

The following files related to legacy agent install features are no longer needed in OpsCenter 6.x and have been removed:

- ./agent/pkg/bin/install_agent.sh
- ./opscenterd/agent/bin/install_agent.sh

The agent install process is now internal to Opscenter, and no external script is necessary.

The path_to_installscript configuration option has been removed from the [agents] section of opscenterd.conf. The option if present in your configuration is ignored for OpsCenter 6.1 and later. Updating your configuration to remove the path_to_installscript is not necessary.

**DataStax Agent backward compatibility**

DataStax Agent backward-compatibility starts with OpsCenter 6.1 going forward. The DataStax Agent has backward compatibility with the installed OpsCenter version, which provides flexibility with the timing of upgrading agents. Before DataStax Agent backward compatibility, the OpsCenter version and the DataStax Agent version were required to match exactly. The DataStax Agent Status View lists any potential features that will not operate for a specific agent version and provides guidance on when it is imperative to upgrade agents.

Note: The OpsCenter 6.1 server is not compatible with DataStax Agent versions earlier than 6.1.0. All DataStax Agents must be upgraded to version 6.1.0.

Check the OpsCenter release notes and known issues for any changes that break DataStax Agent backward-compatibility for features.

**Definitions files directory relocation**

OpsCenter definitions are no longer considered configuration files, and have been moved from /etc/opscenter/definitions to /var/lib/opscenter/definitions. Existing definitions files will be left in place in /etc/opscenter/definitions, but will be ignored by OpsCenter. The existing definitions files in /etc/opscenter/definitions may optionally be deleted manually.

**Key features**

The key features of OpsCenter include:
| **Dashboard monitoring** | An Overview that shows any alerts and condenses the dashboards of multiple clusters.  
A Dashboard that displays an overview of commonly monitored performance metrics (page 324).  
Ability to add and edit graphs (page 325) in the dashboard.  
Monitoring capabilities of DSE In-Memory tables (page 247).  
View the Spark console (page 245). |
|---|---|
| **Configuration, security, and administration** | Basic cluster configuration (page 79).  
Security (page 79) options, with the ability to define user roles, authenticate with LDAP, configure SSL, Kerberos, and encrypt sensitive configuration values.  
Administration tasks, such as adding a cluster (page 274), using simple point-and-click actions.  
Multiple cluster management from a single OpsCenter instance using agents (page 59).  
Automatic failover (page 166) from the primary OpsCenter to a backup OpsCenter instance.  
Rebalance (page 281) data across a cluster after adding or removing non-virtual nodes.  
Manage multiple nodes simultaneously (page 235) for certain bulk operations.  
Downloadable Cluster report (page 323).  
Generate a diagnostics tarball (page 286) to send to support for further troubleshooting. |
| **Alerts (page 138)** | Alert warnings of events and impending issues.  
Built-in external notification capabilities and customizable templates. |
| **Metrics (page 324)** | Metrics are collected (page 151) every minute from Cassandra, Analytics, and Search nodes, and stored in a keyspace created by OpsCenter.  
View historical metrics more than one week in the past. |
About OpsCenter

<table>
<thead>
<tr>
<th>DataStax Enterprise Management Services</th>
<th>DSE Management Services (page 380):</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backup Service (page 380) - allows automatic or manual backup and restore of data in DSE clusters.</td>
<td></td>
</tr>
<tr>
<td>Repair Service (page 462) - continuously runs and performs repair operations across a DataStax Enterprise cluster.</td>
<td></td>
</tr>
<tr>
<td>Capacity Service (page 498) - understand cluster performance trends at a glance and plan for future capacity with forecasting.</td>
<td></td>
</tr>
<tr>
<td>Best Practice Service (page 502) - schedule pre-defined best practice rules that check various properties of clusters and environments.</td>
<td></td>
</tr>
<tr>
<td>Performance Service (page 521) - monitor performance metrics and quickly troubleshoot issues with suggested recommendations.</td>
<td></td>
</tr>
</tbody>
</table>

Lifecycle Manager (page 549) Simplify deploying DataStax Enterprise clusters. Centrally manage cluster, datacenter, and node configuration.

OpsCenter Labs features

Labs features in OpsCenter are experimental features in development available for use. Labs features must be explicitly configured before they are available for use. Be sure to send DataStax feedback (page 26) about the labs features so they can be improved.

**Note:** DataStax does not recommend use of or guarantee performance of Labs features in production.

Exporting and importing dashboard presets

Export dashboard configurations to conveniently import into other clusters or other OpsCenter instances. The dashboard configuration exports and imports as a JSON file.

**Note:** The export and import of presets is currently a labs feature. To enable the feature, add the following to opscenterd.conf and restart opscenterd (page 75):

```
[labs]
enable_dashboard_preset_import_export = True
```

opscenterd.conf

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

- Package installations: /etc/opscenter/opscenterd.conf
- Tarball installations: install_location/conf/opscenterd.conf

1. Go to the Dashboard tab that you want to export. Hover on the preset tab and click the drop-down arrow to open the menu. Click Export from the preset menu.
The dashboard preset .json file downloads to your computer.

2. Go to the cluster or instance into which you want to import the presets.

3. Click the **Import** link on the Dashboard.

   ![Dashboard Import Dialog](image)

   The Import Dashboard Preset dialog appears.

4. Enter a name for the preset.
5. Click **Choose File** and select the exported .json file.

6. Click **Import**.

  The dashboard preset is imported into its new location.

---

**Configuring named route linking**

**opscenterd.conf**

The location of the `opscenterd.conf` file depends on the type of installation:

- **Package installations:** /etc/opscenter/opscenterd.conf
- **Tarball installations:** install_location/conf/opscenterd.conf

Enable route URL deep linking to allow navigating to locations within the OpsCenter UI web application and displaying full path URLs in the browser address bar. Enabling the routes feature allows sending a link to users that directs them to a particular page or location within a page, with the URL reflecting the current location within the UI. Routing URLs enhance the user experience with navigating and sharing links, and facilitate learning the structure of the OpsCenter application.

When routes are enabled, clicking on certain areas within the OpsCenter application such as a dashboard preset updates the URL to reflect the current location in the OpsCenter UI. For example, clicking the Latencies tab displays the URL as `localhost:8888/opscenter/index.html#Test_Cluster/dashboard/presets/latencies`. Alternately, manually entering a known URL also navigates to the UI location.

![Link Example](image)

The slash (root) is the overview route. The route is everything in the URL that comes after the hash #. The hash allows for linking and navigation. The routes are not case-sensitive.

If the routes feature is not enabled, the URL does not reflect its current location within the OpsCenter application. The URL always displays as `localhost:8888/opscenter/index.html` regardless of location.

**Note:** Live routes is a Labs feature. Not all locations in the OpsCenter and Lifecycle Manager UIs currently have a URL available for linking.

1. Open `opscenterd.conf` for editing and add the following:
2. Restart opscenterd (page 75).

Sending OpsCenter metrics to a Graphite server

address.yaml

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

- Package installations: /var/lib/datastax-agent/conf/address.yaml
- Tarball installations: install_location/conf/address.yaml

cluster_name.conf

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

- Package installations: /etc/opscenter/clusters/cluster_name.conf
- Tarball installations: install_location/conf/clusters/cluster_name.conf

Configure forwarding metrics from OpsCenter to a Graphite server. This is an OpsCenter Labs feature (that is, under ongoing development but available for use). OpsCenter pushes metrics at a hard-coded interval of 60 seconds.

Graphite stores time-series data and renders static graphs of the data. If your organization uses Graphite in its reporting infrastructure, you can easily integrate the data collecting power of OpsCenter with the open source Graphite monitoring tool.

OpsCenter versions 6.1.2 and later provide the ability to bypass storing metrics in the rollup tables of the DSE cluster. Set the bypass_dse_metrics_storage option to True to bypass redundant storage of metrics when the Graphite labs feature is enabled in OpsCenter.

**Warning:** If the DSE metrics storage has been disabled but the Graphite reporter is not configured for receiving the metrics, a warning is logged that user should enable an option for metrics storage.

**Prerequisites:** Install and configure Graphite.

1. Open cluster_name.conf for editing. Substitute cluster_name with the name of your cluster. Setting agent options through the cluster configuration file sets the corresponding property in address.yaml on every node.

   If necessitated by your environment, open address.yaml for editing and configure at the node level. Do so for every node that requires a specific configuration override.

2. Add the following configuration options:

   [labs] graphite_host
Setting graphite_host enables the forwarding of metrics to a graphite server at the given address. Leaving the graphite_host blank disables forwarding metrics to the graphite server.

**[labs] graphite_port**
Port for graphite's plaintext protocol.

**[labs] graphite_prefix**
A prefix to insert metrics under.

**[labs] bypass_dse_metrics_storage**
Enable or disable storing metrics in a monitored or a separate storage DSE cluster. Metrics are stored in a DSE monitored or storage cluster by default. Default: False.

```bash
[labs]
graphite_host = 127.0.0.1
graphite_port = 2003
graphite_prefix = opscenter
bypass_dse_metrics_storage = True
```

3. Save the configuration file or files.

4. Restart (page 75) the OpsCenter daemon.

5. If you made changes to address.yaml (page 205), restart (page 76) the DataStax agents.

**Viewing logs from node details**

View logs in the Recent Log Information pane within the Node details dialog. The Cassandra System Log, Cassandra Debug Log, and OpsCenter Agent Log are available for viewing within OpsCenter monitoring. The most recent 1000 lines of a log are displayed in the log window.

**Prerequisites:**

*Note:* Viewing logs from within the OpsCenter UI node details is currently a labs feature (that is, under ongoing development but available for use). To enable the feature, add the following to opscenterd.conf and restart opscenterd (page 75):

```bash
[labs]
log_enable = True
```

Enable logs configuration option:

**[labs] log_enable**
Enables Special Log Management Program (SLMP) that allows various forms of log management functionalities.
opscenterd.conf

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

- Package installations: /etc/opscenter/opscenterd.conf
- Tarball installations: install_location/conf/opscenterd.conf

1. In the left navigation pane, click **Cluster#Nodes#List View or Ring View**.

2. Click the node to view its details.

   The Node Details dialog appears.

3. Scroll down to the **Recent Log Information** pane.

4. Select the log to view:
   - Cassandra System Log
   - Cassandra Debug Log
   - OpsCenter Agent Log

5. Click **Refresh**.

   The last 1000 lines of the selected log are fetched and displayed in the **Recent Log Information** pane.

---

OpsCenter 6.1 User Guide Earlier version, latest patch 6.1.11
**Note:** If an agent is down for a node, the log pane is empty. Review the Agent Status *(page 230)* to troubleshoot *(page 239)* agent issues.

### Bulk uploading S3 backups using the AWS CLI

**address.yaml**

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

- **Package installations:** `/var/lib/datastax-agent/conf/address.yaml`
- **Tarball installations:** `install_location/conf/address.yaml`

**cluster_name.conf**

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

- **Package installations:** `/etc/opscenter/clusters/cluster_name.conf`
- **Tarball installations:** `install_location/conf/clusters/cluster_name.conf`

Use the AWS CLI instead of the AWS SDK when bulk loading backups to Amazon S3 locations. Using the AWS CLI rather than the AWS SDK can result in a performance increase, with a noticeable decrease in the time it takes to complete a backup. This is an OpsCenter Labs feature (that is, under ongoing development but available for use). The feature is available in OpsCenter versions 6.1.3 and later.

For more information, see [AWS CLI](https://aws.amazon.com/documentation/cli/) in the Amazon documentation.

**Note:** When the AWS S3 CLI is enabled, the S3 throttling setting is ignored by OpsCenter during backups. See Tuning throttling *(page 25)* for AWS CLI.

**Prerequisites:**

- Install the AWS CLI package on every node. DataStax recommends using the Amazon bundled installer method and upgrading to the latest version of AWS CLI if it is already installed. See Install the AWS CLI using the bundled installer in the Amazon documentation for installation procedures.

**Tip:** As a recommended best practice for OpsCenter, install the AWS CLI bundle using APT as follows:

```
sudo apt-get install -y unzip
unzip awscli-bundle.zip
sudo ./awscli-bundle/install -i /usr/local/aws -b /usr/local/bin/aws
```

**Important:** Regardless of the install procedure used, make sure that the AWS CLI package is installed in the PATH of the *cassandra* user, or whichever user the DataStax agent runs as *(page 67)*.
• **Add an S3 location** *(page 409)* for backups.

1. Open `cluster_name.conf` for editing. Substitute `cluster_name` with the name of your cluster. Setting agent options through the cluster configuration file sets the corresponding property in `address.yaml` on every node.

   If necessitated by your environment, open `address.yaml` for editing and configure at the node level. Do so for every node that requires a specific configuration override.

2. Add the following configuration option:

   ```
   [labs]
   use_s3_cli = True
   ```

3. Save the configuration file or files.

4. **Restart (page 75)** the OpsCenter daemon.

5. If you made changes to `address.yaml` *(page 205)*, **restart (page 76)** the DataStax agents.

**What's next:** **Tune throttling** *(page 25)* for AWS CLI

**Tuning throttling when using AWS CLI**

Use alternative throttle options when using the AWS CLI for bulk uploads because the OpsCenter S3 throttle is ignored at this time when the OpsCenter AWS CLI for S3 **Labs feature** *(page 24)* is enabled.

1. Adjust the `max_concurrent_requests` available in the AWS SDK. Refer to the AWS CLI S3 configuration documentation for details.

2. If necessary, use a tool such as **Trickle** to limit bandwidth.

**OpsCenter architecture overview**

OpsCenter utilizes an agent-based architecture.

DataStax agents must be installed on every managed node in a cluster and are necessary to perform most of the functionality within OpsCenter. When creating a new cluster with Lifecycle Manager, the DataStax agent is automatically installed. When adding a new cluster to manage with OpsCenter, you are given the option to automatically or manually install agents.

The agents use Java Management Extensions (JMX) to monitor and manage each node. DataStax Enterprise exposes a number of statistics and management operations through JMX. Using JMX, OpsCenter obtains metrics from a cluster and issues various node administration commands, such as flushing SSTables or doing a repair.
DSE OpsCenter Policy Changes

Starting with OpsCenter version 6.0 and later, OpsCenter is only compatible with DataStax Enterprise (DSE) clusters. DataStax discontinued OpsCenter compatibility with:

- Open Source Software (OSS) Cassandra clusters
- DataStax Distributions of Cassandra (DDC) clusters, formerly known as DataStax Community (DSC)

Customers currently using an OpsCenter version earlier than 6.0 to provision, manage, and maintain their OSS Cassandra, DSC, and DDC clusters must look for other tools to continue their management and maintenance activities.

Send DataStax feedback about OpsCenter

Thanks for using OpsCenter. Please take a moment to let us know what you think about the OpsCenter customer experience with its current features. DataStax welcomes any ideas for feature requests and improvements or reports of any bugs encountered.

To provide feedback about OpsCenter documentation by email or Twitter, or to access support resources, see guidelines for giving documentation feedback.

1. Click Help#Feedback at the top of the OpsCenter Monitoring console.
   The feedback form appears.
2. Complete the feedback form:
   
   a. Enter your **Name**.
   
   b. Enter a valid **Email** address.
   
   c. Enter your feedback about DSE OpsCenter in the **Feedback** box.

3. Click **Send Feedback**.
   
   DataStax appreciates your comments!
OpsCenter Release Notes

OpsCenter release notes provide information about new and improved features, known and resolved issues, and bug fixes.

Release impacts

Before reading release notes, review the following information to understand upgrade impacts, compatibility with DataStax Enterprise (DSE) versions, and known issues for the OpsCenter version.

Upgrade Information

Important: Review the New features in OpsCenter 6.1 pertinent to the release. Additionally, review the DataStax OpsCenter Upgrade Guide. Configuration and other notable changes are provided in detail.

Known Issues

Important: Review the list of known issues (page 51) before running a new OpsCenter version on a production DSE cluster.

Compatibility

To see which versions of DataStax Enterprise (DSE) are supported with OpsCenter 6.x, see the OpsCenter Compatibility chart.

OpsCenter 6.1.11 release notes

8 October 2018

Highlights

- Added two parameters to the [ldap] section of opscenterd.conf. These changes include support of LDAP searches for users without specifying an Organizational Unit (OU), plus adding the ability to follow LDAP referrals.

See New features (page 11) for more details.

Changes in 6.1.11

The following changes are included in this release.

Core

- Allows OpsCenter to support LDAP searches for users without specifying an Organizational Unit (OU). Also adds the ability to follow LDAP referrals. (OPSC-13384)

The following flags were added to the [ldap] section of opscenterd.conf to manage the mentioned changes:
enforce_single_user_search_result

Returns an error when multiple entries are returned from a user search after all referrals (if applicable) are followed. Set to False if the user_search_base is not confined to one particular OU. Default: True.

follow_referrals

Sets whether the OpsCenter LDAP client should follow referrals. Active Directory typically does not follow referrals. Default: False.

OpsCenter 6.1.10 release notes

26 September 2018

Highlights

• Implemented numerous fixes and enhancements for backing up to Amazon S3.
• Added multi-role support for LDAP authentication.
• Statistics of all Distributed Subrange (DSR) tasks are now reported by the OpsCenter API, including completed, in-progress, and failed statistics.

See New features (page 11) for more details.

Changes in 6.1.10

The following changes are included in this release.

Core

• The DataStax agent now supports Transport Layer Security (TLS) with remote JMX. (OPSC-8375)
• Added multi-role support for LDAP authentication. Added additional 'roles' field to '/users' and '/users/{username}' GET responses for getting all roles that a user belongs to. (OPSC-12740)
• Corrected an issue that prevented the failover OpsCenter instance from connecting to the DataStax Agents during failover. (OPSC-11742)
• Fixed an issue with the POST URL event plugin that was logging error messages during successful POST operations. (OPSC-13643)
• Improved favicon display in several web browsers. (OPSC-13788)
• Fixed an issue where OpsCenter indicated that a change to the OpsCenter keyspace replication strategy failed, when selecting the link from the notification about the OpsCenter keyspace using SimpleStrategy for replication in a multi-datacenter environment. (OPSC-14406)
• OpsCenter now drops compact storage option from all tables inside the configured OpsCenter keyspace. (OPSC-14442)
• Enhanced OpsCenter to properly log exceptions from LDAP if group names contain Unicode characters. (OPSC-14452)
• Reduced memory usage in opscenterd when requests are made to the DataStax Agent. (OPSC-15037)
• Moved destination validation from OpsCenter to the DataStax Agent. (OPSC-14611)

Monitoring
• Support added for new Read Coordination and Hint metrics added in DSE 5.0.12 (OPSC-12230):
  # Read Requests - Local Node Non Replica
  # Read Requests - Preferred Other Replicas
  # Hints on Disk
  # Hint Replay Success Rate
  # Hint Replay Error Rate
  # Hint Replay Timeout Rate
  # Hint Replay Received Rate

• Implemented a change so that data for average time and average request for Solr cores comes from QueryMetrics MBean rather than older Solr MBeans. (OPSC-14845)
• Improved the color scheme for node status in OpsCenter. (OPSC-12618)
• Fixed an issue where the graph zoom button does not work when the legend is expanded. (OPSC-13413)
• Fixed an issue where nodes were sometimes incorrectly indicated as DOWN in OpsCenter when they were actually UP. (OPSC-14299)

Backup Service
• Added support to use system default credentials for Amazon S3 backups as described in Working with AWS credentials. (OPSC-5161)
• Added proxy server support for Amazon S3 backup and restore. (OPSC-6978)
• Backups to Amazon S3 locations now support Amazon S3 Transfer Acceleration. (OPSC-10271)
• Upgraded Java AWS SDK to version 1.11.328. (OPSC-14454)
• Fixed an issue that caused backup history pagination to fail when many events shared the same time. (OPSC-12836)
• Added support for backing up encryption keys other than system_key, plus support for backing up multiple keys. (OPSC-12914)
• Fixed an issue where a restore would fail if the backup was taken shortly after dropping a column from a table. (OPSC-13029)
• Fixed an issue where snapshots containing one or more SASI indexes could not be restored due to an invalid SSTable name exception during the validation checks. (OPSC-13314)
• Fixed a small rendering issue in Restore from Backup: Other Location form. (OPSC-14226)
• Fixed a bug requiring the user to double click on the plus (+) button when selecting a keyspace from the Create Backup dialog. (OPSC-14228)
• Added support to configure the backup storage directory (backup_storage_dir) using the commit log backup settings. (OPSC-14496)
• Fixed an issue when using multi-level prefix paths as locations for the Backup Service. (OPSC-14687)
• Fixed a bug that could cause problems when restoring materialized views. (OPSC-14727)
• Fixed an issue where AWS regions specified as `remote_backup_region` values in the cluster configuration file were not used as bucket defaults. (OPSC-14775)
• Fixed an issue with where the AWS Credentials Provider Chain was not respected in relations to IAM Roles. (OPSC-14939)
• Fixed an issue in OpsCenter where editing an Amazon S3 destination after a restart shows `Enable S3 server-side encryption` and `Enable S3 transfer acceleration` enabled when they are not. (OPSC-14982)
• Fixed an issue that caused schema files to be sent repeatedly to a destination during a backup. (OPSC-15009)
• Fixed a memory leak in the Backup Service job execution cache. (OPSC-15015)
• Reduced memory required when the Backup Service is taking a snapshot. (OPSC-15046)
• Amazon S3 destinations now support selecting a region from all currently available regions in the UI. (OPSC-14692)

Repair Service
• Omitted verbose C3P0 logging from the DataStax Agent log file. (OPSC-14176)
• Added safeguards to prevent orphaned repair tasks from affecting the currently running repair jobs, which could have caused Repair Service jobs to deadlock. (OPSC-14218)
• Statistics of all Distributed Subrange (DSR) tasks are now reported by the OpsCenter API, including completed, in-progress, and failed statistics. (OPSC-14873)
• Fixed a bug for Distributed Subrange Repair (DSR) to honor the `max_parallel_repairs` property, which was remaining at a value of 1 regardless the specified value. (OPSC-14947)

Provisioning
• Added tooltips for several custom URLs to provide examples of what Lifecycle Manager (LCM) needs them to point to. (OPSC-14060)
• LCM UI form dialogs now display a loading animation while loading form values on slow connections. Form inputs will be grayed out while in the loading state. (OPSC-14123)
• Fixed an issue where column names overlapped in the `key_provider` edit dialog in LCM. (OPSC-10990)
• Enlarged the `ssh-key` field in LCM to improve readability when entering SSH keys. (OPSC-13509)
• Fixed a bug where LCM UI form dialogs would reset values to the original state while being edited. (OPSC-14025)
• Fixed an issue where the `native_transport_port_ssl` setting was ignored when changing the default CQL password. (OPSC-14030)
• Modified the locations that LCM uses to traverse the cluster model when determining the SSH management port. (OPSC-14258)
• Improved error message returned when the `$JAVA_HOME` environment variable is invalid. (OPSC-14390)
• Fixed an issue where LCM jobs would fail to terminate when aborted. (OPSC-14410)
• Disallow LCM to import a cluster when opscenterd is not fully communicating with a cluster. (OPSC-13367)

Dashboard
• Fixed an issue where sparklines in the cluster Overview pane would initially load, but not update. (OPSC-13913)

Nodes
• Fixed an issue where decommissioning a node would sometimes result in an error indicating that OpsCenter tried to cancel an already-cancelled event. (OPSC-14016)

OpsCenter 6.1.9 release notes
8 August 2018

Highlights

Implemented a fix for a critical bug that caused all active, compressed SSTable backups to be cleaned up unnecessarily, resulting in incomplete backups. Active, uncompressed SSTable backup files were unaffected.

See New features (page 11) for more details.

Changes in 6.1.9

The following changes are included in this release.

Backup Service
• Fixed a critical bug that caused all active, compressed SSTable backup files (.gz) to be cleaned up unnecessarily, resulting in incomplete backups. (OPSC-14880)

OpsCenter 6.1.8 release notes
25 July 2018

Highlights

Implemented a fix for rolling repairs so that a new repair job starts automatically if the current job fails. This fix applies to all repair job types (incremental, subrange, and distributed subrange).

Note: This issue was a regression that is not present in previous versions of OpsCenter 6.1.x.

See New features (page 11) for more details.

Changes in 6.1.8
The following changes are included in this release.

Repair Service

• Restore "last repaired" timestamps to the Repair Service UI, which were previously missing. (OPSC-14399).
• Protect distributed subrange temporary files from being deleted when DataStax agents start. (OPSC-14719).
• Fixed a bug that prevented a repair cycle if the Repair Service was unable to find a task to run of over max_down_node attempts (OPSC-14733).
• Prevent duplicate distributed subrange tasks (OPSC-14769).

**OpsCenter 6.1.7 release notes**

18 June 2018

**Highlights**

Implemented DSR (Distributed Subrange Repair) as an alternative implementation of subrange repairs within the OpsCenter Repair Service, intended to better scale for large clusters. See [Enabling distributed subrange repairs](page 484).

See [New features](page 11) for more details.

**Changes in 6.1.7**

The following changes are included in this release.

**Core**

• Sensitive password fields in the cluster configuration are no longer returned by the API. (OPSC-4361)
• Added a dialog to OpsCenter UI prompting the user to refresh the browser when the underlying OpsCenter server version has changed. (OPSC-11254)
• Fixed an issue where nodes would be incorrectly marked as DOWN when they are in fact UP after being added again to the cluster. (OPSC-13408)
• Fixed an issue where opscenterd would not always properly update a node’s UP/DOWN state based on the number of peers that report the node to be down. (OPSC-13909)
• Fixed a bug where dialogs in OpsCenter would reposition incorrectly when the dialog contents changed size. (OPSC-13914)

**Monitoring**

• Fixed an issue causing invalid data for the Total Compactions Completed metric. The fix will be published as a live definition file update for all existing OpsCenter 6.1.x installations. Metric data collected prior to deploying the fix will remain invalid. (OPSC-13657)
• Fixed an issue with the node name and IP background color in the Ring view. (OPSC-10635)
Backup Service

- Removed an outdated 3rd-party URL used in an SSTableloader max heap size error message. (OPSC-13794)
- Fixed an issue that could cause backup SSTable cleanup on a destination to miss some files. (OPSC-13223)
- Fixed an issue with backup S3 and local destination file sync that would abnormally halt when errors occurred during the sync. (OPSC-13957)
- Fixed a bug that in some cases prevented restore of backups taken from prior versions of OpsCenter. (OPSC-14034)
- Fixed a bug that prevented restore of a point-in-time (PIT) backup if the On Server destination was missing the snapshot files. (OPSC-14370)
- Fixed a bug that caused errors in the agent log when syncing commit logs to destinations. (OPSC-13645)
- Corrected an issue with truststore argument to SSTableloader during restore. (OPSC-13686)
- Optimized backup file comparison synchronization. (OPSC-14559)

Best Practice Service

- Fixed an issue where the Best Practice Service would flag multiple networks in the network check when there is only one network that is not a loopback device. (OPSC-13190)

Performance Service

- Fixed a bug in the Table view of Performance Service where the Node grid would not render. (OPSC-13959)

Repair Service

- Implemented the DSR (Distributed Subrange Repair) feature, which is an alternative implementation of subrange repairs for the OpsCenter Repair Service. DSR is designed to scale for larger clusters by distributing more work to the agents. (OPSC-13531)
- Added the use_distributed_subrange_repair config option to the Repair Service for running distributed subrange repairs (page 484) (DSR). (OPSC-13538)
- The Repair Service no longer creates or uses a clustername.json persistence file. (OPSC-13306)
- Replaced in-memory repair tasks management with a SQLite database to avoid OOM errors when generating a large number of subrange tasks. (OPSC-13543)
- Switched to streaming json parsing/generation of large HTTP responses when gathering token ranges from agents as well as generating keyspace metadata for the UI to avoid generating large data structures in-memory. (OPSC-13543)
- Fixed a bug where repair service would resume prematurely if a cluster topology change was longer than the configured resume timeout. (OPSC-12165)

Lifecycle Manager (LCM) Provisioning

- Improved the usability of the Rack field in the LCM Add/Edit Node dialog by adding an autocomplete combo box to retain previously entered rack names for selection. (OPSC-13205)
• Tabbing behavior in the LCM UI has been improved. (OPSC-13204)
• Fixed a bug in LCM cluster import when there are datacenter- or node-specific config options. (OPSC-13546)

OpsCenter 6.1.6 release notes

30 January 2018

Highlights

• Lifecycle Manager can now display up to 300 nodes per datacenter, where previously it was only 50.
• The LCM link in the OpsCenter monitoring navigation sidebar is unavailable for users who do not have the Admin role for OpsCenter authentication.

See New features (page 11) for more details.

Changes in 6.1.6

The following changes are included in this release.

Core

• Updated project dependencies to no longer be susceptible to CVE-2016-1000031. (OPSC-13404)
• The Cluster Health pane now properly sorts datacenter names. (OPSC-11438)
• The LCM link in the OpsCenter monitoring navigation sidebar is unavailable for users who do not have the Admin role for OpsCenter authentication. (OPSC-12303)
• The repair_service_subrange.json and repair_service_incremental.json job persistence files have been added to the diagnostic tarball. (OPSC-12887)
• Added an error message to specifically identify JMX timeouts and inform the user about the agent config parameter than can be adjusted to deal with it. (OPSC-12679)
• An updated MINA version now ships that alleviates a CPU issue associated with LDAP. (OPSC-12946)
• Fixed an issue with authentication and LDAP (configured with anonymous bind), where invalid logins prevented valid LDAP logins from working unless opscenterd was restarted. (OPSC-13045)
• Fixed a bug in the Agent Status View that displayed blank rows when there were large numbers of agents. (OPSC-13351)
• Updated secure connections from opscenterd to the agent to use TLS 1.2 instead of TLS 1.1 due to stricter crypto policies in the latest JDK versions. (OPSC-13710)

Monitoring

• Changed the behavior in the OpsCenter UI so that if an alert is tied to a metric that no longer exists (or does not exist due to a failed definitions update), the UI will ignore the corresponding alerts and give an option to manually delete the alert rules instead of outright deleting them. (OPSC-13382)

Backup Service
• Improved logging around some restore backup error cases. (OPSC-12862)
• Fixed an issue that prevented synced snapshot events from appearing in the restore list. (OPSC-13360)
• Improved the speed at which local backup destinations compute files that need to be synced. (OPSC-13410)

Best Practice Service
• Fixed a rendering bug in the Best Practice rule details dialog. (OPSC-13334)

Repair Service
• Improved memory usage when loading saved subrange repair state. (OPSC-13317)
• Fixed an issue with Repair Service subrange task size creation that caused extremely high data sizes to be repaired. (OPSC-13310)
• Fixed logic for comparing $\text{\text{\textbackslash repair\_service}}.\text{\text{\textbackslash max\_pending\_repairs}$ to running repairs when using DSE 5.0+. (OPSC-13392)
• Fixed repair service subrange repairs ignoring mixed case keyspaces when DSE version is earlier than 5.0.7. (OPSC-13637)

Lifecycle Manager (LCM) Provisioning
• Trailing slash is now optional for LCM API endpoint URLs. (OPSC-5868)
• LCM does not allow running a configure job before a successful install job. LCM now enforces running an install job as a prerequisite to running a configure job. (OPSC-11228)
• LCM gives better error messages when the downloaded JRE/JCE archives cannot be extracted. (OPSC-13327)
• The LCM Config Profile editor now displays default values for dictionary grid views when no value has been entered by a user. (OPSC-10216)
• Fixed a race condition in LCM’s usage of SQLite that lead to rare SQLITE_BUSY and foreign key errors. (OPSC-12349)
• LCM automatically populates the `ssl\text{\_truststore}` and `ssl\text{\_truststore\_password}` fields in the Edit Cluster Connections dialog using the corresponding values from `ssl\text{\_keystore}` and `ssl\text{\_keystore\_password}`. The My Keystore and Truststore are the same check boxes have been removed from the Connect to Existing Cluster and Edit Cluster Connections dialogs. (OPSC-12910)
• Fixed a bug that prevented proper alignment of ellipses in the Jobs page of LCM. (OPSC-13192)
• Fixed a bug in the LCM UI where the DSE Version list on the Add Config Profile page did not sort DSE versions correctly (descending order). (OPSC-13212)
• Fixed a bug where importing a non-GPFS snitch cluster into LCM would create an error event in the node events but would not fail the job. The job now properly fails with the unsupported snitch event. (OPSC-13233)
• Fixed an LCM UI bug where adding and subsequently removing custom dictionary entries on edit config profile would disallow closing a dialog. (OPSC-13342)
• Fixed an agent install bug where only approximately 50 nodes were getting installation status updates. (OPSC-13350)
• Disabling client-to-node encryption from LCM now properly updates the related `cluster.conf` file. (OPSC-13460)
• Fix a bug where CQL password changes fail when authentication is enabled and rpc address is 0.0.0.0. (OPSC-13475)
• Fixed a bug in LCM URL redirects so that they retain query string parameters from the original URL. Optimized related-resources links by avoiding redirects. (OPSC-13508)
• Lifecycle Manager can now display up to 300 nodes per datacenter on its UI; previously it was 50. (OPSC-13508)
• Fixed an issue when setting `SPARK_MASTER_LOG_DIR` from LCM. (OPSC-13516)

OpsCenter 6.1.5 release notes

14 November 2017

Highlights

• Added support for Percent Data Repaired (previously SSTables Repaired) metrics at the node and table levels for DSE versions 5.0.6 and later.
• LCM now supports Amazon Linux AMI 2016.09 and 2017.03. (OPSC-6582)

See New features (page 11) for more details.

Changes in 6.1.5

The following changes are included in this release.

Core

• XHRStream communications failover automatically to long polling after 1 minute. (OPSC-9454)
• Removed tooltip question marks from the tab order sequence in the Edit Cluster Connections dialog in OpsCenter Monitoring. (OPSC-11022)
• Fixed related resources links for jobs and nodes associated with a cluster. (OPSC-12972)
• Improved layout on a few OpsCenter Monitoring pages and dialogs to avoid excessive scrollbars. (OPSC-12995)
• Fixed a layout issue with cluster seed text area (expanded the Enter host or IP box in the Connect to Existing Cluster dialog). (OPSC-4350)
• Fixed an issue where opscnterds was unnecessarily migrating table metadata on every restart. (OPSC-12001)
• Fixed a rendering bug causing harmless browser console messages when opening the Node Details dialog. (OPSC-12997)
• Fixed a bug with rendering of sparklines on the global OpsCenter Monitoring dashboard. (OPSC-13038)
• Fixed timeout that caused the OpsCenter UI to fail to load. (OPSC-13053)
• Fixed a bug with the Cluster Connections Settings dialog rendering properly when resized smaller. (OPSC-13116)
• Fixed a bug where a secondary OpsCenter instance would generate an invalid link to the primary OpsCenter when https is enabled. (OPSC-13140)
• Fixed rendering edge case in agent grid affecting horizontal scrolling in narrow windows. (OPSC-13262)

Monitoring
• Added support for Percent Data Repaired (previously SSTables Repaired) metrics at the node and table levels for DSE versions 5.0.6 and later. (OPSC-12982)
• Moved the metrics dropdown tooltip to the left to avoid occluding the scrollbar. (OPSC-13077)

Backup Service
• Improved error message when destination pre-test fails. (OPSC-13159)
• Add messages to detail the agent activity during a backup. (OPSC-13242)
• Fixed a UI bug when reporting backup progress on large clusters. (OPSC-4430)
• Fixed a bug with header spacing in the restore from backup list. (OPSC-5778)
• Updated the backup report dialog to include empty tables because their schema is backed up. (OPSC-12703)
• Fixed a bug where only a subset of keyspaces were shown in the Restore from Backup dialog keyspace selector. (OPSC-13010)
• Improved resize behavior of Restore from Backup dialog. (OPSC-13049)
• Corrected an issue that prevented restores to DSE 4.8 clusters with optional client-to-server ssl. (OPSC-13155)
• Corrected an issue where scheduled backups with a cleanup policy would always fail if a retry sync was triggered. (OPSC-13234)

Repair Service
• Increased Repair Service tokenranges_partitions default value to 2^20, matching what DSE uses for its Merkle tree depth in CASSANDRA-5263. (OPSC-12901)
• Fixed an issue where repair service pausing event log messages were generated when the repair service wasn't running. (OPSC-13002)

Lifecycle Manager (LCM) Provisioning
• LCM now supports Amazon Linux AMI 2016.09 and 2017.03. (OPSC-6582)
• Improved the error message when an LCM password change prevents connecting to CQL. (OPSC-12535)
• LCM now allows debug log to be disabled through the logback.xml configuration. (OPSC-13109)
• Added a tooltip for the advanced-jvm-options field in the jvm-options UI of LCM Config Profiles. (OPSC-13122)
• Added a tooltip for the endpoint snitch in the cassandra.yaml UI of LCM Config Profiles. (OPSC-13123)
• Added a 30 second timeout for HTTP requests in Meld. Among other potential HTTP endpoint issues, this addresses LCM jobs hanging while trying to download the JDK. (OPSC-13134)
• The LCM UI now explicitly defaults nodes' rack field to rack1. (OPSC-10814)
• LCM now sets secure permissions on the directories and files it creates for signing and storing SSL certificates. (OPSC-11793)
• LCM now gives a better error message when SSL certificate creation fails due to clock drift on the target node. (OPSC-12321)
• Fixed charset issue resulting in superfluous non-ASCII characters displaying in the LCM UI. (OPSC-12932)
• Fixed a bug on LCM UI where a datacenter name would erroneously be displayed on the Run Job dialogs for a cluster. (OPSC-12965)
• Fixed a bug in LCM where cluster imports were always failing during the supported platform check. (OPSC-13218)
• LCM will now retry the Oracle JRE download twice if it has a connection or timeout error. (OPSC-13293)

OpsCenter 6.1.4 release notes

24 October 2017

Highlights

See New features (page 11) for more details.

Changes in 6.1.4

The following changes are included in this release.

Core
• Fixed an issue with agent stomp connections on nodes with two network adapters. (OPSC-13016)

Lifecycle Manager (LCM) Provisioning
• Fixed issue where Oracle no longer hosts the version of Java that LCM attempts to download by default. (OPSC-13332)

OpsCenter 6.1.3 release notes

21 September 2017

Highlights

• Added an OS supported platform (page 573) check for DSE installs in LCM.
• Added a Labs feature (page 24) to use the AWS CLI instead of the AWS SDK when bulk loading backups to Amazon S3. Significant performance impact!
• Added the ability to sync a snapshot (page 404) to a destination for On Server backups.

See New features (page 11) for more details.
Changes in 6.1.3

The following changes are included in this release.

Core

- Timeouts for all API calls are proportional to default_api_timeout setting. (OPSC-12206)
- Added a Hide and Show toggle for the status panels on the Nodes page. (OPSC-12424)
- Fixed memory leak in node list tooltips. (OPSC-12614)
- Fixed an issue where LDAP sockets remained open after failed login attempts. (OPSC-12656)
- Now show loading indicator while service panels initialize. (OPSC-12714)
- Fixed bug where service panels would stay in error state after server restart. (OPSC-12749)
- Fixed broken View in List link in cluster ring popup menus. (OPSC-12786)
- Fixed stack trace when details requested for a node with agent down. (OPSC-12788)
- Fixed an issue where the DataStax Agent would stop responding due to a bug during configuration changes. (OPSC-12921)

Backup Service

- Added the ability to sync a snapshot (page 404) to a destination for On Server backups. (OPSC-12574)
- Added the ability to auto retry syncing backups to destinations. (OPSC-12575)
- Improved the resiliency of the DataStax Agent during backups. (OPSC-12576)
- Added a configuration option to use the AWS CLI (Command Line Interface) scripts for bulk uploading rather than the S3 API when backing up to Amazon S3. Using the Amazon S3 cli is a labs feature that must be enabled. (OPSC-12688)
- Made the destination verification timeout configurable. (OPSC-10409)
- Local FS destinations for Backup Service now respects the throttling parameter. (OPSC-12174)
- Fixed scrolling on dialog content when browser window is small. (OPSC-12562)
- Corrected an issue where a failed backup could result in excessive error messages in the opscenterd log. (OPSC-11122)
- Backup process is more resilient to lost progress messages. (OPSC-12758)
- Contains fix for leaking file descriptors when the backup service takes a snapshot. (OPSC-12900)
- Fixed an issue where the restore rate was not displaying in the ui. (OPSC-12974)
- Fixed an issue where having more than one scheduled backup would cause the ui list to display nothing. (OPSC-12873)

Repair Service

- Repair Service will now switch away from the Status tab when it is detected to be inactive. (OPSC-11223)
- Fixed an issue where leaving the repair service status open for longer periods of time could slow down the UI. (OPSC-12833)
- Fixed an issue with Out of Memory errors in opscenterd on the Repair Service Status page by removing the bottom Repair Tasks panel and optimizing the repair-status API endpoint. (OPSC-12857)
- Fixed a bug that indicated the incorrect agent minimum version. (OPSC-12896)

**Performance Service**

- Fixed column sorting in the Slow Queries page of the Performance Service. (OPSC-11704)

**Best Practice Service**

- Best Practice Rule for Secondary Index Cardinality no longer fails for system keyspaces. (OPSC-6913)

**Lifecycle Manager (LCM) Provisioning**

- Improved a meld error concerning PID file permissions to be more informative. (OPSC-12537)
- LCM now performs a check during install and import jobs to ensure the OS platform is supported for the version of DSE being installed. This behavior can be overridden with the new `disable_platform_check` config option in the `lifecycle_manager` section of `opscenterd.conf`. (OPSC-11592)
- Corrected a bug that could cause comments to fail to display in LCM cluster, datacenter, and node edit pages. (OPSC-12558)
- Fixed a rendering issue with `user_defined` config profile editor in LCM. (OPSC-12559)

### OpsCenter 6.1.2 release notes

**26 July 2017**

**Highlights**

- Developed new optimizations for making subrange repair metadata processing more performant for large and/or dense clusters. (OPSC-11976)
- LCM ensures that the correct version of DSE is present before modifying the target node in configure jobs. (OPSC-12503)

See [New features](page 11) for more details.

**Changes in 6.1.2**

The following changes are included in this release.

**Core**

- Changed the UI labeling from deleting a cluster to disconnecting a cluster in the Edit Connection Settings dialog to make it clear that the clusters themselves are not actually deleted. (OPSC-10654)
• Removed the redaction of usernames from agent logs. Passwords continue to be redacted. (OPSC-10285)
• Combined the Services and Operations panels in the Nodes page. (OPSC-12270)
• Changed the UI labeling from deleting a cluster to disconnecting a cluster in the Edit Connection Settings dialog to make it clear that the clusters themselves are not actually deleted. (OPSC-10654)
• Added the ability to select or deselect all permissions for OpsCenter roles. (OPSC-11757)
• Fixed an issue that prevented CQL Solr queries from being reflected in OpsCenter metrics. (OPSC-12071)
• Fixed issue with agent connections in some failover situations. (OPSC-11292)
• When the listen_address field in cassandra.yaml file is left blank, OpsCenter agents now default to the same listen address as DSE. (OPSC-12246)
• Fixed a bug where OpsCenter failed to display a correct representation of the cluster state after an LCM job restart. (OPSC-12376)
• Corrected an issue where the agent was using a legacy method to authenticate with DSE. (OPSC-12560)
• Changed the icon for the Last Install Status Pending state in the Agent Status view. The icon is now differentiated from the Installing icon. (OPSC-11229)

Monitoring
• Added ClientRequest metrics for various types of unsuccessful reads and writes. (OPSC-12186)
• OpsCenter metrics storage can now be disabled by using the bypass_dse_metrics_storage configuration option in cluster_name.conf or address.yaml. (OPSC-12235)
• Fixed a bug that could cause the ring for a vnode cluster to display with a large gap. (OPSC-12426)
• Fixed an issue that prevented the selection of day for the duration of an alert. (OPSC-12351)
• Fixed an issue where a change to the percentile of an alert would not take effect without restarting opscenterd. (OPSC-12352)

Backup Service
• Fixed issue where backup report dialog would display date and time of backup in the local time zone of the browser rather than UTC. (OPSC-7261)
• Zipped version of backup files are now automatically deleted after being uploaded to destinations. (OPSC-9131)
• Corrected an issue that required the user to provide a value for local_interface when configuring agents for use on dense nodes. (OPSC-11777)
• Improved commit log status display when no commit logs have been archived. (OPSC-12062)
• There were occasional problems when restoring a backup related to schema changes. This was improved by better logic around waiting for schema agreement. (OPSC-12231)
- Fixed issue where S3 upload failed with message "resetting to invalid mark". (OPSC-12278)
- Fixed issue where Backup Service would incorrectly report a restore as failed due to a misreading of the bulk load status. (OPSC-12317)
- A bug in scheduling was found and fixed where a scheduled job (notably backups) would be rerun almost immediately after the second time it was executed. (OPSC-12205)

**Repair Service**

- Suppress the Repair Won't Finish In Time alert when elapsed throughput is below the configured min_throughput property value. (OPSC-12207)
- Developed new optimizations for making subrange repair metadata processing more performant for large and/or dense clusters. (OPSC-11976)

**Lifecycle Manager (LCM) Provisioning**

- Updated SSH errors to be more helpful. (OPSC-10849)
- Updated unexpected password prompt errors to have a more helpful error message. (OPSC-12484)
- Added support to detect and log when openssl and keytool binaries are missing on the LCM node. (OPSC-12210)
- Improved error message around bad LCM decryption keys. (OPSC-12306)
- Added human readable message under the message field for API errors. (OPSC-12498)
- If the openssl or keytool executables are missing, this will be reported in the LCM UI as a job event. (OPSC-12519)
- Eliminated SQLITE_BUSY errors when using LCM. (OPSC-11885)
- LCM ensures that the correct version of DSE is present before modifying the target node in configure jobs. (OPSC-12503)
- Corrected a bug that could cause comments to fail to display in LCM cluster, datacenter, and node edit pages. (OPSC-12558)

**OpsCenter 6.1.1 release notes**

12 June 2017

**Highlights**

- Many helpful content additions to the diagnostic tarball for better troubleshooting, along with an improved customer experience.
- Support for HTTP Strict Transport Security (HSTS).
- Added SSL truststore configuration options for the agent.
- Summary panels for Services and cluster health have been added to the Nodes page for a quick view of status and fast access to each Services section.

See [New features](page 11) for more details.

**Changes in 6.1.1**
The following changes are included in this release.

Core

- Added some additional files (logback.xml files to /opscenterd and /dse, log4j.properties to /agent, jvm.options and commitlog_archiving.properties to /cassandra, and /etc/hosts) to diagnostic tarball contents. Also fixed an issue with a missing trailing slash that caused agent diagnostic files to be filed directly under the /conf folder instead of the /agent subfolder. (OPSC-11611)
- Agent status information is now included in the diagnostic tarball. (OPSC-7277)
- When reporting a failure to generate a diagnostic tarball due to a lack of available disk space, the amount of space required, the amount of space available, and the working directory for diagnostic tarball generation is now clearly logged. (OPSC-5959)
- The diagnostic tarball no longer creates a file containing an error message rather than a directory if a node’s diagnostic tarball could not be downloaded. A file name agent_requests.json is now included that indicates the success or failure of retrieving diagnostic tarballs from each node. (OPSC-6258)
- Whenever possible, any current information from the Best Practice Service is now included in the diagnostic tarball. (OPSC-4956)
- The agent now re-uses an existing connection to DSE when gathering some diagnostic tarball output rather than forking additional cqlsh processes. (OPSC-7081)
- The agent will retry hung connection attempts to DSE nodes. (OPSC-11935)
- Summary panels with status and links to the Services have been added to the Nodes page. (OPSC-12138)
- Selecting a keyspace is now optional when performing a cleanup, compact, or flush node operation. This feature is not backwards-compatible with agent versions earlier than 6.1.1. (OPSC-510)
- Added truststore settings to agent configuration for both monitored and storage clusters. (OPSC-8003)
- Cleaned up logging on opscenterd shutdown. (OPSC-8278)
- Selecting NetworkTopologyStrategy while editing a keyspace in the OpsCenter Data section now populates the datacenter replication factor parameters with the datacenters in the cluster topology. Any previously specified replication factor values for a given datacenter parameter will be reused. Assigning a replication factor of 0 to a datacenter parameter excludes replication on the datacenter. (OPSC-10680)
- Included change details in audit log messages for updates to role permissions. (OPSC-10929)
- Added summary information about running Services to the cluster landing page. (OPSC-11247)
- Removed repetitive log message 'INFO: Node 172.31.11.1 has multiple tokens (vnodes). Only one picked for display. (MainThread)'. Only affects vnode clusters. (OPSC-11557)
- Added the X-XSS-Protection to content responses to enable the XSS auditor in various supported browsers. (OPSC-11845)
- Fixed a bug where opscenterd would automatically encrypt values in its configs if a system key was present, whether config encryption was enabled or not. (OPSC-7152)
• Allow access to the left navigation, including LCM, when there's only one cluster configured and OpsCenter cannot connect to it. (OPSC-11101)
• Provide clear messages in the diagnostic tarball when using DSE 5.0 or greater that the removal of Thrift has also removed the use of cassandra-cli. (OPSC-11239)
• Improve messaging around timeouts when logging in via LDAP. (OPSC-11396)
• New OpsCenter roles now default to having no permissions for any clusters. (OPSC-11685)
• The DataStax Agent may now use different passwords for the keystore and truststore. (OPSC-11767)
• Fixed a bug when resuming monitoring after a rolling restart. (OPSC-12095)
• Optimized requests for Service status while viewing a service detail page. (OPSC-12136)
• Improved error handling logging around HTTP calls to the agent. (OPSC-12143)
• Improved handling of separate storage cluster. (OPSC-8235)
• Attempting to log in without specifying a username now gives a proper HTTP Unauthorized response instead of a 500 error response. (OPSC-9469)
• If OpsCenter authentication is enabled, users who do not have View Schema permission in their assigned role are not allowed to view schema data in the Data area of OpsCenter monitoring. OpsCenter presents an informational message that instructs users to contact their OpsCenter administrator for access. (OPSC-11319)
• Redacted passwords from debug log statements in WrappedDriver. (OPSC-11448)
• Fixed a bug during agent install that is triggered when no events have been reported. (OPSC-11640)
• Updated solr-cores and range-list routes in agent to address intermittent errors. (OPSC-11674)
• Fixed agent processing of KMIP errors for alerts. This was causing log spam in the agent and was preventing other real-time information from being reported to OpsCenter. (OPSC-11972)
• Fixed a bug where the dashboard would not display when non-ASCII characters are present in a username. (OPSC-12064)
• Fixed an issue where requests would timeout when using AD 2012 with SSL. (OPSC-11995)

Monitoring

• Added Speculative Retries metrics to OpsCenter monitoring. (OPSC-11503)
• Added Coordinator Read Latency table metric. (OPSC-11509)
• Clarify the units in several 'TP: Dropped X' metrics. (OPSC-11714)

Backup Service

• Improved performance when displaying updates in the restore status dialog. (OPSC-1498)
• Added support for folders and subfolders to Amazon S3 buckets used for backups and restores. (OPSC-6845)
• Agent configuration now validates that backup_staging_dir is set to an absolute path. (OPSC-9393)
• Fixed sorting of scheduled backups. (OPSC-11007)
• Horizontal scroll has been added to the OpsCenter Services section when the browser window is too small. (OPSC-11144)
• Improved the error message during certain backup failure cases. (OPSC-11733)
• Fixed the display of the list of backups presented after adding a backup location. (OPSC-11933)
• Fixed an issue where backups to S3 would stop without logging an error. (OPSC-12188)
• Fixed an issue that caused the destination dialog to appear multiple times when restoring from an ad hoc S3 backup. (OPSC-11569)
• Removed checks preventing users from restoring backups across major versions. This restriction was in place because earlier but no longer supported versions of Cassandra did not allow this. It is now possible to restore backups from any version of Cassandra that is compatible for upgrading. (OPSC-11766)

Repair Service
• Increased accuracy when computing repair service job throughput. (OPSC-11171)
• Fixed a NullPointerException in agents when repair service was running and the cluster was rebalanced. (OPSC-11668)
• Repair Service settings shows calculated time to completion separately from the setting’s input field. (OPSC-11985)
• Clarified display of total repair attempts in the repair status page. (OPSC-12004)
• Fixed repair to not use keystones in the ignored_keyspace property when computing the maximum number of repairs that can run in parallel. (OPSC-12045)
• Fixed an issue in the repair service where restarting OpsCenter would disable the repair service. (OPSC-12237)

Performance Service
• Fixed column sorting in the Slow Queries page of the Performance Service. (OPSC-11704)
• Performance Service in OpsCenter can only be configured when at least one agent is connected. The OpsCenter api now returns an empty array instead of an empty object for /perf/config when no agents are connected. (OPSC-11267)

Best Practice Service
• Best Practice Rule for Secondary Index Cardinality no longer fails for system keystones. (OPSC-6913)
• Fixed column sorting in the Slow Queries page of the Performance Service. (OPSC-11704)

Lifecycle Manager (LCM) Provisioning
• LCM now warns users that config-encryption is not supported. (OPSC-7616)
• LCM now tries to assign one seed per rack, two seeds per datacenter, and three seeds for each cluster. (OPSC-11160)
• LCM proceeds with jobs when a become-password is set in the machine-credential even if the target does not prompt for a sudo password, rather than generating a misleading meld-not-started error. (OPSC-11216)
• Updated the Java download UI in LCM to reflect new Oracle URL structure. Also added tooltips to assist users with completing Java Setup fields for Config Profiles. (OPSC-11418)

OpsCenter 6.1.0 release notes

18 April 2017

Highlights

• Support for DSE 5.1.
• Revamped Repair Service! Smarter and more performant repair with real visibility into the progress and results of the repair process.
• DataStax agent improvements: Backward compatibility with the installed OpsCenter version, which provides flexibility with the timing of upgrading agents; improved Agent Status view; and access to the DataStax Agent API in a Swagger UI console.

See New features (page 11) for more details.

Changes in 6.1.0

The following changes are included in this release.

Core

• The Hadoop workload type has been removed from the Datacenter dialog in LCM. Support for Hadoop has been deprecated. (OPSC-10466)
• Changed criteria when the Analytics filter is used to constrain the nodes that provide data for a given graph. Previously only Hadoop nodes were considered when using this filter, but now only Spark nodes are considered. (OPSC-9562)
• Added the `swagger_enabled` config option to `address.yaml` that exposes the DataStax Agent API (page 627) in a Swagger UI console. (OPSC-10337)
• Added an appropriate error when an incompatible route is called. (OPSC-9905)
• Updated agent status to appropriately indicate incompatibility and limitations. (OPSC-9908)
• If an agent's version will not permit a Best Practice Rule to run, a warning is shown when configuring the rule's schedule. (OPSC-9911)
• OpsCenter added support for messaging latency metrics (page 341) when monitoring DSE clusters from versions 5.1 and later. (OPSC-10093)
• Added the ability to configure static JVM options in a `jvm.options` file within Lifecycle Manager config profiles for DSE clusters version 5.1.0 and later. The options were moved from `cassandra-env.sh` to `jvm.options`. See Configuring JVM options in LCM (page 618). (OPSC-10101)
• The set of available cipher suites for TLS encryption of communication between the agent and DSE has been expanded to support the set of ciphers supported by the Java platform. This matches a similar change in Apache Cassandra™, which allows for negotiation of the strongest common cipher suite. (OPSC-10187)
- Added support for DSE Integrated Authentication for JMX in DSE 5.1. See Configuring a JMX Connection using LCM (page 615). (OPSC-8752)
- Added metrics (page 346) for monitoring DSE Graph in OpsCenter. (OPSC-10180)
- The banner that notifies users of agent issues no longer includes agent version mismatches that do not impact features. Incompatible version mismatches still appear in the banner. (OPSC-10327)
- Agent Upgrade information is now presented in a text block above the agents status table. Agents can be upgraded by clicking the Upgrade Agents button. (OPSC-10615)
- The Agent Version is now listed in the Agent Status tab. (OPSC-10616)
- Agents with incompatible versions are now listed as problems in the agent status notification banner. (OPSC-10617)
- Definitions files have been moved out of configuration directories (/etc/opscenter for package installs; install_location/conf for tarball installs.) The new locations are /var/lib/opscenter/definitions and install_location/definitions, respectively. This will be transparent to the majority of users. (OPSC-6704)
- Added version prefix to agent API URIs. (OPSC-7531)
- Removed path_to_installscript option from the [agents] section of opscenterd.conf. That option will now be ignored. The agent install process is now internal to Opscenter, and no external script is necessary. (OPSC-9119)
- Moved agent-to-opscenter SSL settings from JVM_OPTS in the agent startup script to address.yaml. SSL settings in JVM_OPTS will continue to work as before if these settings are not specified in address.yaml. (OPSC-9398)
- Added validation to os-metric/disk-space route. (OPSC-9928)
- Enabled JVM heap dump by default for both opscenterd and the agent when there is an OutOfMemoryError. (OPSC-10861)
- All memberOf attribute values are now evaluated during LDAP memberOf authentication. (OPSC-10893)
- Included change details in audit log messages for updates to role permissions. (OPSC-10929)
- Removed Clojure compiler warnings from agent startup. (OPSC-11068), (OPSC-11452)
- Improved log messages when agents experience NFS read issues. (OPSC-11665)
- Added validation of Content-Type header to agent API. (OPSC-10001)
- No longer automatically install openjdk when directly installing OpsCenter from RPM or Deb packages. (OPSC-10260)
- If OpsCenter authentication is enabled, users who do not have View Schema permission in their assigned role are not allowed to view schema data in the Data area of OpsCenter monitoring. OpsCenter presents an informational message that instructs users to contact their OpsCenter administrator for access. (OPSC-11319)
- Fixed an issue which prevented some Solr operations from working correctly on multi-homed hosts. (OPSC-11437)
- Fixed the /failover endpoint on the secondary OpsCenter instance. (OPSC-10720)
- Fixed an issue which would cause opscenterd to mark an agent's http interface as down. (OPSC-10879)
• Fixed a bug in agent lookup of datacenter name via JMX. This caused move token requests to fail during repairs. (OPSC-11788)
• Fixed a potential SQL injection exploit in SQL authentication of OpsCenter (DatastaxEnterpriseAuth) in which an attacker could potentially gain access to the OpsCenter UI and its APIs using the password of any user that exists in the database. (OPSC-11991)

Monitoring
• The schema viewer in the Data area of the OpsCenter Monitoring UI now displays materialized views, secondary indexes, and user-defined types used on a table. Users with permissions can view query statements on the new User-Defined Types, User-Defined Functions, and User-Defined Aggregates tabs. (OPSC-4760)
• Hover tooltips that describe each column in the Agent status view have been added. (OPSC-10769)
• Added route /v1/monitor to agent to retrieve self-monitoring data on cassandra, monitored and storage clusters, jmx, rollups, messaging, component state, etc. (OPSC-3492)
• Added convenient UI access to critical logs from within the Node details dialog. The logs are a labs feature that must be enabled. See Viewing logs from node details (page 22). (OPSC-9442)
• Added LiveScannedHistogram metric to track the number of cells scanned during a read. (OPSC-10786)
• Allow overriding content of http callback alerts. (OPSC-11054)
• Alert badges in the global dashboard show an alerts list when clicked. (OPSC-4625)
• Added support for the SSTables Repaired (page 481) metrics at the node and table levels for DSE 5.1. (OPSC-10495)
• Added Start Time column to the slow query log table in the Performance Service. (OPSC-10537)
• Fixed a bug where best practice rule for Compaction Strategy would not log a failure. (OPSC-10987)

Lifecycle Manager (LCM)
• RLAC (page 613) can now be enabled through LCM. (OPSC-10099)
• LCM now allows jobs across clusters to run concurrently, but multiple jobs for the same cluster continue to run serially. (OPSC-8457)
• LCM can no longer import clusters that have Hadoop enabled. Migrate to DSE Analytics prior to importing to LCM. (OPSC-10506)
• LCM can no longer run jobs on clusters that have Hadoop enabled. (OPSC-10507)
• The hive-site.xml file is now managed in the Spark section of the Configuration Profiles in LCM. (OPSC-10890)
• Lifecycle Manager now validates cluster and datacenter names as ASCII strings instead of failing on non-ASCII characters during DSE installation. (OPSC-9276)
• The DSE password now properly escapes the percentage (%) character. Users who escaped that character with the double %% workaround need to change their configuration back to a single %. (OPSC-9500)
Backup Service

- OpsCenter now warns users that attempt to restore backups without setting an explicit throttle value for stream throughput. Applies to DSE clusters from versions 4.8.7 and later. (OPSC-10185)
- The [cloud] group and its only property, accepted_certs, are no longer part of OpsCenter config. (OPSC-11114)
- OpsCenter switched to the Amazon SDK for S3 functionality. (OPSC-10033), (OPSC-10036)
- Added support for backup and restore by datacenter. (OPSC-6155)
- Re-implemented bandwidth throttling for S3 operations to make them more efficient and accurate. (OPSC-10073)
- Added support for cloning (page 429) from a backup using Local FS. (OPSC-6135)
- Improved bucket name validation to more closely match what is allowed by AWS. (OPSC-6183)
- Added the ability to alert on backup failure (page 452) to the Event log for failed backups. (OPSC-7635)
- The config parameter backup_file_queue_max has been removed from address.yaml. (OPSC-8045)
- The Commit Log Archiving Consistency Best Practice Rule has been added to ensure all nodes within a cluster have Commit Log Archiving enabled to prevent any data loss during a PIT restore. (OPSC-10299)
- Added support for a Local FS backup location and additional S3 or local locations in the Point in Time restore dialog in the Backup Service. (OPSC-10665), (OPSC-10728)
- Tables with no data in them are now captured in backup data and can have their schemas restored. (OPSC-6558)
- Performance improvements with local and S3 destinations. (OPSC-11727)
- Updated the commit log archiving template to prevent an bug where commit logs would be transferred incompletely to S3. Users with commit log archiving currently active need to disable then enable commit log archiving for this change to take effect. (OPSC-11903)
- Fixed the Unrecognized config key warning in the agent logs. (OPSC-9609)
- Fixed an issue where materialized views were offered as tables that could be restored directly. (OPSC-11917)
- Fixed a bug where a backup to a destination would be marked as successful even though the schema.json failed to upload and OpsCenter would be unable to restore from it. (OPSC-10815)

Repair Service

- Removed configuration settings from Repair Service that are no longer in use: ks_update_period, max_err_threshold, alert_on_repair_failure, repair_estimation_factor, incremental_range_repair. (OPSC-5895)
- A new Status (page 474) tab for the Repair Service conveys detailed information about subrange and incremental repairs. (OPSC-10020)
- Removed the /<cluster ID>/services/repair/invalid_keyspaces API route since it is no longer used in the OpsCenter 6.1 Repair Service. (OPSC-11139)
• The new Repair Service configuration option `time_to_completion_target_percentage` provides control over the speed with which repair jobs complete within the specified `Time to completion` parameter. See Adjusting or disabling the throttle for subrange repairs (page 488). (OPSC-2011)

• Reworded log warning to use a less alarming phrase to describe an activation delay in the Repair Service. (OPSC-6453), (OPSC-8681)

• Enhanced the Repair Service to ignore specific keyspaces and tables in addition to system keyspaces during a subrange repair. (OPSC-10458)

• Excluded materialized views from incremental and subrange repairs. (OPSC-10955)

• Clarified instructions for `Time to completion` and its relationship to `gc_grace_seconds` on the Repair Service settings page, plus provided links to relevant documentation. (OPSC-11454)

• Removed the Repair Service progress bar from the Services page in lieu of a separate and much more detailed Repair Service status page. (OPSC-11560)

• OpsCenter Services displays a Details link rather than a Configure link when services are already enabled. (OPSC-11661)

• The target number of partitions per subrange can now be configured with the `tokenranges_partitions` configuration option. (OPSC-11801)

• Added new Repair Service property `tokenranges_http_timeout` for setting a higher http timeout if necessary when retrieving tokenranges. (OPSC-11187)

• Changed the system keyspaces excluded by the Repair Service to: `dse_perf`, `system`, `system_distributed`, `system_schema`, `system_traces`. (OPSC-11817)

• Always now honor an API request to activate repair service. If a cluster is not yet in a stabilized state in which repairs can run, the Repair Service is activated in a paused state rather than raising an exception. (OPSC-11838)

**Known and resolved issues for OpsCenter 6.1 and later**

The following are known issues that exist in OpsCenter 6.1 and later versions. Each item has a link to more details including workarounds when available. These issues will be addressed in future releases where possible. If you have any questions, contact DataStax Support for assistance.

**OpsCenter 6.5.0**

• When running an LCM job and attempting to abort or terminate the job while it is in progress, termination fails to stop the job unless the abort request is issued prior to the first node completing. Issuing a terminate or abort command after the first node has finished running has no effect. The job will continue to run to completion as if the terminate command had not been issued. (OPSC-14410)

**OpsCenter 6.1.x and 6.5.0**

• OpsCenter does not automatically remove compact storage from its keyspaces when upgrading to OpsCenter 6.5.0. For important details, see Compact storage no longer supported. (OPSC-14442)
OpsCenter Release Notes

OpsCenter 6.1 and later

- For DSE versions 5.1 and later, slow query data is only available since the last time the DataStax agent was restarted. (OPSC-11702)
- If there are approximately 75 or more keyspaces, the DataStax Agent /tokenranges API call runs out of memory with the default heap size. As a temporary workaround, adjust the agent heap size (page 176). (OPSC-11975)
- When using OpsCenter to restore a backup that contains multiple SASI indexes, some or all of these indexes might not restore correctly. The indexes appear in the table schema but might not function correctly. Indexes should be validated at restore time and rebuilt if errors are detected. For more information, see CREATE CUSTOM INDEX (SASI). (OPSC-11746)
  
  **Note:** SASI indexes are experimental for DSE. DataStax does not support SASI indexes for production.

OpsCenter 6.1

- A large number of log messages might display regarding requests to /pit-cleanup if there are a large number of existing commit logs in the staging directory. (OPSC-8349)
- Insufficient permissions on the staging directory can cause the agent to exhaust inotify watches on the system over time. (OPSC-10732)
- Users will see an ungraceful error+stack trace in opscnterd.log if accessing a cluster through the UI/API that no longer exists. The error message contains “ERROR: Unhandled error in Deferred: There are no clusters with name or ID...”. This error message is harmless. (OPSC-8819)
- Enabling SNMP alerts may cause opscnterd to hang on startup in some slower environments. (OPSC-9314; see More Details)
- For DSE versions earlier than 5.0.7, the DataStax Agent can only estimate partition sizes and counts per node or keyspace for repairs by using JMX stats. For DSE versions 5.0.7 and later, the DataStax Agent queries the system size_estimates table for a more precise estimate of partition sizes and counts per range. (OPSC-11417, OPSC-11590)
- For DSE versions 5.0 and later, object permissions currently are not persisted with an OpsCenter backup and thus are not re-applied when that backup is restored. As a result, users must manually manage object permissions externally from OpsCenter. For more details (no workaround available at this time), see the KB support article. (OPSC-11015)
- The solr-index-size (displayed as Search: Core Size) metric in the OpsCenter Monitoring UI is unavailable for DSE versions 5.1.0 through 5.1.3. (OPSC-12267)

- **Lifecycle Manager (LCM)**

  ```# Lifecycle Manager is not currently compatible with DSE Transparent data encryption. See Encrypted DSE configuration values (page 597) for more details. (OPSC-7529)
  ```

  ```# DSE Graph properties: DSE Graph configuration in dse.yaml, which is configurable through LCM Config Profiles. All Graph properties in dse.yaml can be managed through the LCM UI with the exception of gremlin_server.serializers and gremlin_server.scriptEngines. If you are using LCM and need to customize these properties, be sure to leverage the LCM API to make the changes. Future```
changes to the Config Profile using the LCM UI will retain properties set through the API.

# When configuring credentials in a Repository, special characters such as #, $, and so forth are supported, but non-ascii unicode characters are not. (OPSC-8921)
Installing DSE OpsCenter 6.1

Installing OpsCenter from the RPM package

Install the DSE OpsCenter using Yum repositories on RedHat Enterprise Linux (RHEL), CentOS, and Oracle Linux (OL) distributions.

For a complete list of supported platforms, see OpsCenter Supported Platforms.

The CentOS, RHEL, and OL OpsCenter packaged releases create an opscenter user. OpsCenter runs as a service and runs as the opscenter user. The service initialization script is located in /etc/init.d.

If the OpsCenter machine reboots, OpsCenter restarts automatically. To disable restart upon reboot:

```
$ sudo update-rc.d opscenterd disable
```

Prerequisites:

Minimum hardware requirements for the machine on which OpsCenter runs:

- 2 CPU cores
- 2 GB of RAM available to OpsCenter

Permission and software requirements:

- Yum package management utility.
- Latest build of a Technology Compatibility Kit (TCK) Certified OpenJDK version 8 or Oracle Java SE Runtime Environment 8 (JRE or JDK). Earlier or later versions are not supported. See installing the Oracle JDK or OpenJDK.
  
  **Tip:** Use OpsCenter Lifecycle Manager to automatically manage Java and JCE installs (page 618) for DSE clusters.

- DataStax recommends using a recent version of one of the major web browsers. OpsCenter does not support Internet Explorer or Microsoft Edge.

**Important:** End User License Agreement (EULA). By downloading DataStax products, you confirm that you agree to the processing of information as described in the DataStax website privacy policy and agree to the website terms of use.

1. Open the Yum repository specification /etc/yum.repos.d/datastax.repo for editing. For example:

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

2. In this file, add the repository for OpsCenter.
Installing DSE OpsCenter 6.1

The OpsCenter Debian and Ubuntu packaged releases run as a service from root. The service initialization script is located in `/etc/init.d`.

If the OpsCenter machine reboots, OpsCenter restarts automatically. To disable restart upon reboot:

```
$ sudo update-rc.d opscenterd disable
```

Prerequisites:
Minimum hardware requirements for the machine on which OpsCenter runs:

```
[opscenter]
name = DataStax Repository
baseurl = https://rpm.datastax.com/enterprise
enabled = 1
gpgcheck = 0
```

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

```
$ sudo rpm --import https://rpm.datastax.com/rpm/repo_key
```

4. Install the OpsCenter package.

```
$ sudo yum install opscenter
```

For most users, the out-of-box configuration should work just fine. If necessary, you can configure (page 182) OpsCenter for your environment.

5. Start OpsCenter:

```
$ sudo service opscenterd start
```

6. Connect to OpsCenter in a web browser using the following URL:

```
http://opscenter-host:8888/
```

What’s next:

Add (page 274) an existing cluster or provision a new cluster in Lifecycle Manager (page 554).

Installing OpsCenter from the Debian package

Install DSE OpsCenter using APT repositories on Debian or Ubuntu distributions.

For a complete list of supported platforms, see OpsCenter Supported Platforms.

If the OpsCenter machine reboots, OpsCenter restarts automatically. To disable restart upon reboot:

```
$ sudo update-rc.d opscenterd disable
```
Installing DSE OpsCenter 6.1

- 2 CPU cores
- 2 GB of RAM available to OpsCenter

Permission and software requirements:

- APT Package Manager is installed.
- Latest build of a Technology Compatibility Kit (TCK) Certified OpenJDK version 8 or Oracle Java SE Runtime Environment 8 (JRE or JDK). Earlier or later versions are not supported. See installing the Oracle JDK or OpenJDK.

  **Tip:** Use OpsCenter Lifecycle Manager to automatically manage Java and JCE installs *(page 618)* for DSE clusters.

- DataStax recommends using a recent version of one of the major web browsers. OpsCenter does not support Internet Explorer or Microsoft Edge.

  **Important:** End User License Agreement (EULA). By downloading DataStax products, you confirm that you agree to the processing of information as described in the DataStax website privacy policy and agree to the website terms of use.

In a terminal window:

1. Modify the aptitude repository source list file (`/etc/apt/sources.list.d/datastax.sources.list`).

   ```
   $ echo "deb https://debian.datastax.com/enterprise/ stable main" | sudo tee -a /etc/apt/sources.list.d/datastax.sources.list
   ```

2. Add the DataStax repository key to your aptitude trusted keys:

   ```
   $ curl -L https://debian.datastax.com/debian/repo_key | sudo apt-key add -
   ```

3. Install the OpsCenter package using the APT Package Manager:

   ```
   $ sudo apt-get update
   $ sudo apt-get install opscenter
   ```

For most users, the out-of-box configuration should work just fine. If necessary, you can configure *(page 182)* OpsCenter for your environment.

4. Start OpsCenter:

   ```
   $ sudo service opscenterd start
   ```

5. Connect to OpsCenter in a web browser using the following URL:
Installing DSE OpsCenter 6.1

http://opscenter-host:8888/

What's next:
Add (page 274) an existing cluster or provision a new cluster in Lifecycle Manager (page 554).

Installing OpsCenter with a tarball on any Linux distribution

Install DSE OpsCenter on any Linux Distribution or Mac OS X using the OpsCenter binary tarball.

Note: Mac OS X is supported for development and testing purposes only.

For a complete list of supported platforms, see OpsCenter Supported Platforms.

Prerequisites:
Minimum hardware requirements for the machine on which OpsCenter runs:

- 2 CPU cores
- 2 GB of RAM available to OpsCenter

Permission and software requirements:

- Latest build of a Technology Compatibility Kit (TCK) Certified OpenJDK version 8 or Oracle Java SE Runtime Environment 8 (JRE or JDK). Earlier or later versions are not supported. See installing the Oracle JDK or OpenJDK.
  
  Tip: Use OpsCenter Lifecycle Manager to automatically manage Java and JCE installs (page 618) for DSE clusters.

- DataStax recommends using a recent version of one of the major web browsers. OpsCenter does not support Internet Explorer or Microsoft Edge.

Important: End User License Agreement (EULA). By downloading DataStax products, you confirm that you agree to the processing of information as described in the DataStax website privacy policy and agree to the website terms of use.

1. Download the tarball distribution of OpsCenter.

   To install the latest version:

   $ curl -L https://downloads.datastax.com/enterprise/opscenter.tar.gz | tar xz

   To install earlier versions, add the version number in the above command. For example:
Installing DSE OpsCenter 6.1

$ curl -L https://downloads.datastax.com/enterprise/opscenter-6.1.5.tar.gz | tar xz


Files for OpsCenter and a single DataStax agent are now in place.

2. Change to the `opscenter-version_number` directory.

   $ cd opscenter-version_number

3. Start OpsCenter from the install location:

   $ bin/opscenter

   **Note:** Use `bin/opscenter -f` to start OpsCenter in the foreground.

4. Connect to OpsCenter in a web browser using the following URL:

   http://opscenter-host:8888/

**What's next:**

Add (page 274) an existing cluster or provision a new cluster in Lifecycle Manager (page 554).

### Installing OpsCenter on Docker

Create an OpsCenter Docker container using a DataStax maintained Docker image. See the [DataStax Docker docs](https://docs.datastax.com/en/dse/6.1/index.html) for information on downloading and using the DataStax images for Docker.

### Uninstalling OpsCenter

Select the uninstall method to follow for your type of OpsCenter installation.

**Uninstalling an OpsCenter RPM package**

Use this method if OpsCenter was installed using an RPM (page 54) package.

1. **Stop (page 75) OpsCenter.**

2. Open a terminal and enter the following command:

   $ sudo yum remove opscenter
Uninstalling an OpsCenter Debian package

Use this method if OpsCenter was installed using a Debian (page 55) package.

1. **Stop (page 75) OpsCenter.**

2. Open a terminal and enter the following command:

```bash
$ sudo apt-get purge opscenter
```

Uninstalling an OpsCenter binary tarball

Use this method if OpsCenter was installed using a tarball (page 57).

1. **Stop (page 75) OpsCenter.**

2. Open a terminal and enter the following command:

```bash
$ rm -rf /path/to/opscenter
```

## Installing DataStax Agents

DataStax agents must be installed on every managed node in a cluster and are necessary to perform most of the functionality within OpsCenter. When creating a new cluster with Lifecycle Manager, the DataStax agent is automatically installed. When adding a new cluster to manage with OpsCenter, you are given the option to automatically or manually install agents.

### Related information:

**Agents Status View** [View the current installation, configuration, and connection status of agents. Agent status automatically updates as the information becomes available within OpsCenter. Set up, upgrade, and configure agents. Troubleshoot agent installation, configuration, and connections.] (page 230)

### Installing DataStax agents automatically

DataStax agents must be installed on every managed node in a cluster and are necessary to perform most of the functionality within OpsCenter.

After adding a cluster (page 274), OpsCenter will determine whether agents are already properly installed and configured. If they are not, you will be prompted to check the status of the agents and potentially install them from the Agents (page 230) tab.

### Prerequisites:

- Root or sudo access to the machines where the agents will be installed.
- JMX connectivity is enabled on each node in the cluster.
- Either you configured the SSH port (page 182), or accepted the default SSH port (22) for node-agent communication.
Installing DSE OpsCenter 6.1

- **DataStax Enterprise** 4.7 or greater.

1. **Install OpsCenter** *(page 54)*.

2. Start your DataStax Enterprise cluster and the OpsCenter daemon.

3. Open a browser window and go to the OpsCenter URL at
   \[http://opscenter_host:8888\], where \textit{opscenter_host} is the IP or hostname of the OpsCenter machine.

   A dialog prompts you to select a cluster option.

4. Click **Manage Existing Cluster** and **Get Started**.

   The **Connect to Existing Cluster** dialog appears.
5. Add one or more hostnames or IP addresses of the nodes in the cluster. For best results, use private IP addresses.

6. Click **Next**.

   The **Set Up Agents** dialog appears.

   ![Set Up Agents dialog](image)

7. **Install or start agents automatically** is selected by default. Click **Next**.

   If your environment requires manual installation of agents, click **Install agents manually**. Click the link to access the instructions for manually installing agents (page 63).

   The **Agents Credentials** dialog appears.

   ![Agent Credentials dialog](image)

8. Enter SSH credentials to connect to the nodes:

   **Important**: You must specify either a sudo password for the indicated user, or a private SSH key. The installation will fail without one of these values.
a. Enter a **Username**. The user must have root or sudo privileges.

b. Enter a sudo **Password** for the specified user.

c. Enter a **Private Key**.

Entered credentials information is not saved or stored.

9. Click **Submit**.

The **Agents tab (page 230)** opens, starts installing the agents, and displays the progress of the agent installation. The agent services might go up and down during the installation process.

When the installation process successfully completes, the DataStax agents are deployed and configured for each managed node in the DataStax Enterprise cluster.

If there are any issues installing DataStax agents, a banner at the top of the OpsCenter workspace is displayed. The banner cannot be dismissed until the DataStax agent issues have been resolved, but the banner does not prevent using or navigating the OpsCenter UI. Clicking the **Show Details** link in the banner displays the clusters having agent issues and the number of problems currently detected by OpsCenter. Clicking the **# problems** link opens the **Agent Status (page 230)** tab where you can view more detailed information about agent status, view troubleshooting suggestions, and access the **Set Up Agents** button to retry installing the agents.
If you were unable to install the agents through the OpsCenter UI, follow the instructions to manually install the agents (page 63).

Related information:
Agents Status View [View the current installation, configuration, and connection status of agents. Agent status automatically updates as the information becomes available within OpsCenter. Set up, upgrade, and configure agents. Troubleshoot agent installation, configuration, and connections.] (page 230)

Installing DataStax agents manually

If automatic agent installation was unsuccessful, manually install the agents using the procedure appropriate for your installation.

Manually deploying agents - RPM

address.yaml

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

- Package installations: /var/lib/datastax-agent/conf/address.yaml
- Tarball installations: install_location/conf/address.yaml

opscenterd.conf

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

- Package installations: /etc/opscenter/opscenterd.conf
- Tarball installations: install_location/conf/opscenterd.conf

Prerequisites:

- Root or sudo access to the machines where the agents will be installed.
- The DataStax Enterprise cluster is up and running.
- OpsCenter is installed and configured.
- JMX connectivity is enabled on each node in the cluster.

Important: End User License Agreement (EULA). By downloading DataStax products, you confirm that you agree to the processing of information as described in the DataStax website privacy policy and agree to the website terms of use.

In a terminal for both 32- and 64-bit systems:

1. Add the DataStax Yum repository in the /etc/yum.repos.d/datastax.repo file.

```bash
[opscenter]
name = DataStax Repository
baseurl = https://rpm.datastax.com/enterprise
enabled = 1
```
2. If you have enabled signature verification (gpgcheck=1), import the repository key:

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

3. Install the DataStax agent.

```
$ sudo yum install datastax-agent
```

4. In address.yaml (page 205), set stomp_interface to the IP address that OpsCenter is using. You might have to create the address.yaml file.

```
$ echo "stomp_interface: reachable_opscenterd_ip" | sudo tee -a /var/lib/datastax-agent/conf/address.yaml
```

5. If SSL communication is enabled in opscenterd.conf, use SSL in address.yaml.

```
$ echo "use_ssl: 1" | sudo tee -a /var/lib/datastax-agent/conf/address.yaml
```

6. Start the DataStax agent.

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

### Manually deploying agents - tarball

address.yaml

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

- **Package installations:** /var/lib/datastax-agent/conf/address.yaml
- **Tarball installations:** install_location/conf/address.yaml

opscenterd.conf

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

- **Package installations:** /etc/opscenter/opscenterd.conf
- **Tarball installations:** install_location/conf/opscenterd.conf

**Prerequisites:**

- The DataStax Enterprise cluster is up and running.
- OpsCenter is installed and configured.
- JMX connectivity is enabled on each node in the cluster.
- SYSSTAT Utilities (needed for the collection of I/O metrics).
Important: End User License Agreement (EULA). By downloading DataStax products, you confirm that you agree to the processing of information as described in the DataStax website privacy policy and agree to the website terms of use.

1. Download the DataStax agent tarball, expand and unarchive it.
   
   ```
   $ curl -L http://downloads.datastax.com/enterprise/datastax-agent-version_number.tar.gz | tar xz
   ```
   
2. Change into the agent directory.
   
   ```
   $ cd datastax-agent-version_number
   ```
   
3. In `address.yaml` (page 205) set `stomp_interface` to the IP address that OpsCenter is using. You might have to create the `address.yaml` file.
   
   ```
   $ echo "stomp_interface: reachable_opscenterd_ip" >> ./conf/address.yaml
   ```
   
4. If SSL communication is enabled in `opscenterd.conf`, use SSL in `address.yaml` (page 205).

   ```
   $ echo "use_ssl: 1" >> ./conf/address.yaml
   ```
   
5. Start the agent.

   ```
   $ bin/datastax-agent
   ```
   Use the `-f` flag to run in the foreground.

**Manually deploying agents - Debian**

`address.yaml`

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

- **Package installations:** `/var/lib/datastax-agent/conf/address.yaml`
- **Tarball installations:** `install_location/conf/address.yaml`

`opscenterd.conf`

The location of the `opscenterd.conf` file depends on the type of installation:

- **Package installations:** `/etc/opscenter/opscenterd.conf`
- **Tarball installations:** `install_location/conf/opscenterd.conf`

**Prerequisites:**

- Root or sudo access to the machines where the agents will be installed.
• The DataStax Enterprise cluster is up and running.
• OpsCenter is installed and configured.
• JMX connectivity is enabled on each node in the cluster.

Important: End User License Agreement (EULA). By downloading DataStax products, you confirm that you agree to the processing of information as described in the DataStax website privacy policy and agree to the website terms of use.

1. Add the DataStax repository to the `/etc/apt/sources.list.d/datastax.sources.list` file (if you have already not done so).

   ```
   $ echo "deb https://debian.datastax.com/enterprise/stable main" | \
   sudo tee -a /etc/apt/sources.list.d/datastax.sources.list
   ```

2. Add the DataStax repository key to your Aptitude trusted keys.

   ```
   $ curl -L https://debian.datastax.com/debian/repo_key | sudo apt-key add 
   ```

3. Install the DataStax agent.

   ```
   $ sudo apt-get update
   sudo apt-get install datastax-agent
   ```

4. In `address.yaml (page 205)`, set `stomp_interface` to the IP address that OpsCenter is using. You might have to create the `address.yaml` file.

   ```
   $ echo "stomp_interface: reachable_opscenterd_ip" | sudo tee -a /var/lib/datastax-agent/conf/address.yaml
   ```

5. If SSL communication is enabled in `opscenterd.conf`, use SSL in `address.yaml (page 205)`.

   ```
   $ echo "use_ssl: 1" | sudo tee -a /var/lib/datastax-agent/conf/address.yaml
   ```

6. Start the DataStax agent.
$ sudo service datastax-agent start

Setting permissions to run the agent as the DSE user

If you install the DataStax Agent from a tarball, you must manually configure the agent to run as the same DataStax Enterprise (DSE) user and set permissions for this user.

**Note:** When DSE is installed, it creates a cassandra user in the database and runs as this user. It also creates a cassandra user in the operating system. Do not use the cassandra user in production for either the database or operating system. Failing to do so is a security risk.

**Prerequisites:** Ensure the necessary read and write permissions are set for the user or group running the agent:

**Table 1: Directory and File Permissions**

<table>
<thead>
<tr>
<th>Feature functionality</th>
<th>Permissions required</th>
</tr>
</thead>
<tbody>
<tr>
<td>General agent functionality</td>
<td>Read permission to cassandra.yaml</td>
</tr>
<tr>
<td>Configuring a cluster</td>
<td>Read/write permissions to configuration directories and files.</td>
</tr>
<tr>
<td>Backup and restore</td>
<td>• Read/write permissions to configuration directories and files.</td>
</tr>
<tr>
<td></td>
<td>• Read/write permissions to Cassandra data directories.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Note:</strong> A <em>umask</em> (page 68) must also be set to accommodate group permissions for new tables and data.</td>
</tr>
<tr>
<td></td>
<td>• If commit log archiving (page 384)  is enabled, the DataStax Enterprise process must also have permissions to run the agent's archive script and write permissions to the configured backup directory.</td>
</tr>
</tbody>
</table>

**Table 2: Directory and File Locations**

<table>
<thead>
<tr>
<th>Directory/File</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>cassandra.yaml</td>
<td>See Configuration directories and files below.</td>
</tr>
<tr>
<td>Configuration directories and files</td>
<td>• DataStax Enterprise Package and Installer-Services installations: /etc/dse</td>
</tr>
<tr>
<td></td>
<td>• DataStax Enterprise Tarball and Installer-No Services installations: install_location/conf</td>
</tr>
<tr>
<td>Data directories</td>
<td>Default: /var/lib/cassandra</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Location is user-configurable; set in cassandra.yaml.</td>
</tr>
</tbody>
</table>
To set up the umask:

1. Open a terminal.

2. To give read/write permissions for new tables and data, edit the appropriate shell file for the DataStax Enterprise environment:

3. Add the command `umask 002` to the top of the file.

Setting the umask to 002 is required because Cassandra creates new directories or files as 0700 by default, which does not grant read or write permissions.

```
umask 002
```

## Configuring JAVA_HOME for DataStax agents

DataStax agents do not pick up the environment variables of the currently logged in user by default. If Java is not in the machine’s `PATH`, there are errors in the agent log on startup:

```
nohup: cannot run command 'java': No such file or directory
```

- On the Cassandra nodes where the agents are installed, create the file `/etc/default/datastax-agent` and set the environment variables for `JAVA_HOME` and any other custom environment variables that the agent might need. For example:

```
JAVA_HOME = /usr/lib/jvm/java-8-oracle
```

## OpsCenter Reference

### OpsCenter ports reference

OpsCenter ports reference for setting firewall rules

Use the OpsCenter ports reference for guidance when configuring firewall rules. The ports reference provides details for the default port numbers used by OpsCenter, Lifecycle Manager, and DataStax Agents. The ports are set to defaults. The configuration options and file or UI locations for overriding port defaults are indicated in the description column of the following table.
Table 3: OpsCenter Ports

<table>
<thead>
<tr>
<th>Port/Protocol</th>
<th>Source</th>
<th>Destination</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8888/TCP</td>
<td>Management workstation</td>
<td>OpsCenter server</td>
<td>OpsCenter web-based user interface. The <code>opscenterd</code> daemon listens on this port for HTTP requests coming directly from the browser. Configure with the [webserver] port (page 182) option in <code>opscenterd.conf</code>.</td>
</tr>
<tr>
<td>DSE nodes</td>
<td>OpsCenter server</td>
<td>OpsCenter server</td>
<td>LCM Meld configs and job-events. The <code>opscenterd</code> daemon listens on this port for configuration requests and job-events coming from DSE nodes during the provisioning and installing agent processes. Configure with the [webserver] port (page 182) option in <code>opscenterd.conf</code>.</td>
</tr>
</tbody>
</table>
| 7199/TCP      | DataStax agent          | DSE               | JMX monitoring port. Each agent opens a JMX connection to its local node (the DataStax Enterprise process listening on this port). The JMX protocol requires that the client then reconnect on a randomly chosen port (1024+) after the initial handshake.

**Note:** It is not necessary to set this local port for firewall purposes.

Configure with the [jmx] port (page 196) option in `cluster_name.conf`, or if necessary, in `address.yaml`. |
### Port/Protocol

<table>
<thead>
<tr>
<th>Port/Protocol</th>
<th>Source</th>
<th>Destination</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8443/TCP</td>
<td>Management workstation</td>
<td>OpsCenter server</td>
<td>SSL. The port on which to serve SSL traffic. The <code>opscenterd</code> daemon listens on this port for HTTPS requests being redirected from port 8888. The SSL port is the alternative UI/API port when SSL is enabled (<a href="#">page 82</a>). When SSL is on, 8888 redirects to 8443 and serves no content. Configure with the <code>[webserver] ssl_port (page 182)</code> in <code>opscenterd.conf</code>.</td>
</tr>
<tr>
<td>8443/TCP</td>
<td>DSE nodes</td>
<td>OpsCenter server</td>
<td>LCM Meld configs and job-events. The <code>opscenterd</code> daemon listens on this port for configuration requests and job-events coming from DSE nodes during the provisioning and installing agent processes. Configure with the <code>[webserver] ssl_port (page 182)</code> in <code>opscenterd.conf</code>.</td>
</tr>
<tr>
<td>25/TCP (non-SSL) 465 (SSL)</td>
<td>OpsCenter server</td>
<td>SMTP server</td>
<td>SMTP for email alerting. See Enabling SMTP email alerts (<a href="#">page 143</a>). Configure in one or more uniquely named <code>config_location/event-plugins/email.conf</code> files.</td>
</tr>
<tr>
<td>9042/TCP</td>
<td>OpsCenter server</td>
<td>DSE</td>
<td>Native transport: The native transport port for the cluster configured in <code>native_transport_port</code> in <code>cassandra.yaml</code>. Port 9042 must be open from the OpsCenter server to all DSE nodes. Configure with the <code>cql_port (page 197)</code> option in <code>cluster_name.conf</code>. <strong>Note</strong>: The port must also be open on a storage cluster (<a href="#">page 154</a>). Configure with the <code>[storage_cassandra] cql_port (page 199)</code>.</td>
</tr>
<tr>
<td>Port/Protocol</td>
<td>Source</td>
<td>Destination</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| DataStax agent | DSE | | The agent and `opscenterd` must be able to connect to the `native_transport_port`.  
**Note:** It is not necessary to set this local port for firewall purposes. |
| 61619/TCP | OpsCenter server | OpsCenter server | OpsCenter stomp port. A port used by the Stomp service for internal communications on the OpsCenter server.  
**Note:** It is not necessary to set this local port for firewall purposes.  
Configure with the `[agents] incoming_port (page 186)` in `opscenterd.conf`. |
| 61620/TCP | DataStax agents | OpsCenter server | Agent communications over Stomp. The `opscenterd` daemon listens on this port for TCP traffic coming from the agents.  
Configure with the `stomp_port (page 206)` option in `address.yaml`. |
| 61621/TCP | OpsCenter Server | DataStax agents | DataStax agent port for DataStax agents http/https service. The agents listen on this port for TCP traffic initiated by OpsCenter. Depending on the OpsCenter and Agent configuration, traffic is either http or https protocol.  
Configure with the `[agents] api_port (page 200)` in `cluster_name.conf` or in `address.yaml`. |
<table>
<thead>
<tr>
<th>Port/Protocol</th>
<th>Source</th>
<th>Destination</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>22/TCP</td>
<td>OpsCenter server</td>
<td>LCM deployment targets and agent install deployment targets</td>
<td>SSH. Standard SSH (Secure Shell) port. Lifecycle Manager and the agent installation process log into target nodes over SSH. Configured with the [agents] ssh_port (page 186) in opscenterd.conf. <strong>Note:</strong> The SSH Management Port can be overridden at the cluster, datacenter, or node level in the Clusters (page 554) workspace of Lifecycle Manager (page 549).</td>
</tr>
<tr>
<td>443/TCP</td>
<td>OpsCenter server</td>
<td>OpsCenter download host</td>
<td>OpsCenter Definitions. OpsCenter definition files allow updating the support of DSE versions without requiring upgrading to an updated version of OpsCenter itself. See Updating and configuring definitions files properties (page 164). Configure with the [definitions] download_port (page 183) in opscenterd.conf.</td>
</tr>
<tr>
<td>389/TCP and UDP</td>
<td>OpsCenter server</td>
<td>LDAP or AD server</td>
<td>LDAP or AD. Default port for non-SSL LDAP and AD (page 118). Configure with the [ldap] server_port (page 184) option in opscenterd.conf.</td>
</tr>
<tr>
<td>636/TCP and UDP</td>
<td>OpsCenter server</td>
<td>LDAP or AD server</td>
<td>LDAP or AD SSL. Default port for SSL traffic to LDAP and AD. Configure with the [ldap] server_port (page 184) option in opscenterd.conf.</td>
</tr>
<tr>
<td>Port/Protocol</td>
<td>Source</td>
<td>Destination</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>--------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>7080/TCP</td>
<td>OpsCenter server</td>
<td>DSE</td>
<td>Spark Jobs UI. Port at which the Spark jobs UI is exposed. Configure the port to view Spark Master nodes ([page 245](page 245)) in the Spark console access from the Node Details dialog within OpsCenter. Configure with the [spark master_http_port](page 202) option in <code>cluster_name.conf</code>. For information about defining a datacenter in OpsCenter LCM for Spark workloads, see [Adding a datacenter](page 560). For information about using Spark with the DataStax Enterprise platform, see [analyzing data using Spark](page 560).</td>
</tr>
<tr>
<td>7081/TCP</td>
<td>Management workstations</td>
<td>OpsCenter server</td>
<td>Spark Jobs UI. OpsCenter proxies the Spark Jobs UI. Configure with the [spark base_master_proxy_port](page 188) option in <code>opscenterd.conf</code>.</td>
</tr>
<tr>
<td>162/UDP</td>
<td>OpsCenter server</td>
<td>SNMP Manager</td>
<td>SNMP. Listening port on the SNMP manager to receive [SNMP](page 139) traps. Configure with one or more uniquely named <code>config_location</code> ([page 74](page 74))/<code>event-plugins/snmp.conf</code> files.</td>
</tr>
<tr>
<td>2003/TCP</td>
<td>DataStax agent</td>
<td>Graphite server</td>
<td>Graphite (labs feature in OpsCenter). Port to which OpsCenter sends Graphite ([page 21](page 21)) metrics. Configure with the [labs graphite_port](page 204) option in <code>cluster_name.conf</code> or <code>address.yaml</code> if necessary.</td>
</tr>
</tbody>
</table>
### Installing DSE OpsCenter 6.1

<table>
<thead>
<tr>
<th>Port/Protocol</th>
<th>Source</th>
<th>Destination</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3128/TCP</td>
<td>LCM targets</td>
<td>Third-party proxy server</td>
<td>Package proxy port in LCM for DSE installs. Default port for the Lifecycle Manager package proxy for downloading software installation packages for DataStax Enterprise and its dependencies such as Java. Configure in the Package Proxy (page 621) section of the Config Profile in the Lifecycle Manager UI.</td>
</tr>
</tbody>
</table>

**cassandra.yaml**

The location of the `cassandra.yaml` file depends on the type of installation:
- **Package installations**: `/etc/cassandra/cassandra.yaml`
- **Tarball installations**: `install_location/conf/cassandra.yaml`

**opscenterd.conf**

The location of the `opscenterd.conf` file depends on the type of installation:
- **Package installations**: `/etc/opscenter/opscenterd.conf`
- **Tarball installations**: `install_location/conf/opscenterd.conf`

**address.yaml**

The location of the `address.yaml` file depends on the type of installation:
- **Package installations**: `/var/lib/datastax-agent/conf/address.yaml`
- **Tarball installations**: `install_location/conf/address.yaml`

**cluster_name.conf**

The location of the `cluster_name.conf` file depends on the type of installation:
- **Package installations**: `/etc/opscenter/clusters/cluster_name.conf`
- **Tarball installations**: `install_location/conf/clusters/cluster_name.conf`

### Installation and configuration locations

#### Default file locations for package installations

File locations when installed from a Debian or RPM package.

<table>
<thead>
<tr>
<th>Directory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>/var/lib/opscenter</code></td>
<td>SSL certificates for encrypted agent/dashboard communications</td>
</tr>
<tr>
<td><code>/var/log/opscenter</code></td>
<td>Log directory</td>
</tr>
</tbody>
</table>
Installing DSE OpsCenter 6.1

<table>
<thead>
<tr>
<th>Directory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/var/run/opscenter</td>
<td>Runtime files</td>
</tr>
<tr>
<td>/usr/share/opscenter</td>
<td>JAR, agent, web application, and binary files</td>
</tr>
<tr>
<td>/usr/share/doc/opscenter</td>
<td>Licenses and other documents</td>
</tr>
<tr>
<td>/etc/opscenter</td>
<td>Configuration files</td>
</tr>
<tr>
<td>/etc/init.d</td>
<td>Service start-up script</td>
</tr>
</tbody>
</table>

Default file locations tarball installations

The installation files listed in the following table are located under the OpsCenter installation location.

<table>
<thead>
<tr>
<th>Directory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/agent</td>
<td>Agent installation files</td>
</tr>
<tr>
<td>/bin</td>
<td>Startup and configuration binaries</td>
</tr>
<tr>
<td>/content</td>
<td>Web application files</td>
</tr>
<tr>
<td>/conf</td>
<td>Configuration files</td>
</tr>
<tr>
<td>/doc</td>
<td>Licenses and other documents</td>
</tr>
<tr>
<td>/lib and src</td>
<td>Library files</td>
</tr>
<tr>
<td>/log</td>
<td>OpsCenter log files</td>
</tr>
<tr>
<td>/ssl</td>
<td>SSL files for OpsCenter-to-agent communications</td>
</tr>
</tbody>
</table>

Starting, stopping, and restarting OpsCenter

Commands for starting, stopping, and restarting OpsCenter for each type of installation.

- **Package installs**
  
  # The available service opscenterd options are:

  ```
  $ sudo service opscenterd start|stop|status|restart|force-reload
  ```

  # Packaged installations include startup scripts in the /etc/init.d directory for running OpsCenter as a service.

  For example, the following script starts OpsCenter on the node where the command runs.
Installing DSE OpsCenter 6.1

$ sudo /etc/init.d/opscenterd start

- **Tarball installs**

Starting opscenterd:

Use the `-f` option to start the agent in the foreground.

$ install_location/bin/opscenter

Stopping and restarting opscenterd:

1. To stop OpsCenter, find the opscenterd process ID (`pid`) and kill the process using its PID number:

   $ ps -ef | grep opscenter

   $ sudo kill pid

2. To start OpsCenter:

   Use the `-f` option to start the agent in the foreground.

   $ install_location/bin/opscenter

**Startup log for OpsCenter**

A preliminary startup.log file is created upon startup of the opscenterd process. The startup log file can contain debug information or stack traces if opscenterd fails to start up before a normal opscenterd.log can be created.

startup.log

The location of the `startup.log` file depends on the type of installation:

- **Package installations**: `/var/log/opscenter/startup.log`
- **Tarball installations**: `install_location/opscenterd/log/startup.log`

opscenterd.log

The location of the `opscenterd.log` file depends on the type of installation:

- **Package installations**: `/var/log/opscenter/opscenterd.log`
- **Tarball installations**: `install_location/log/opscenterd.log`

**Stopping, starting, and restarting DataStax agents**

- **Stop the DataStax agent**:

  # Package installations:
$ sudo service datastax-agent stop

#  Tarball installations:
To stop the DataStax agent, find the DataStax agent Java process ID (PID) and kill the process using its PID number.

$ ps -ef | grep datastax-agent
$ sudo kill pid

• **Start the DataStax agent:**
  #  Package installations:
  $ sudo service datastax-agent start

  **Note:** The DataStax agent starts automatically.

  #  Tarball installations:
  Use the `-f` option to start the agent in the foreground.

  $ install_location/bin/datastax-agent

• **Restart the DataStax agent:**
  #  Package installations:
  $ sudo service datastax-agent restart

  #  Tarball installations:
  1. To stop the DataStax agent, find the DataStax agent Java process ID (PID) and kill the process using its PID number:

     $ ps -ef | grep datastax-agent

     $ sudo kill pid

  2. Start the agent:

     Use the `-f` option to start the agent in the foreground.

     $ install_location/bin/datastax-agent
Upgrading OpsCenter

See the DSE OpsCenter Upgrade Guide for detailed instructions on upgrading OpsCenter.
Configuring OpsCenter

OpsCenter Security

OpsCenter SSL overview

For increased security, configuring the following SSL for OpsCenter is a recommended best practice:

- Configure SSL support for the OpsCenter web application (page 79)
- Configure SSL between OpsCenter and DataStax agents (page 82)

Enabling HTTPS for the OpsCenter server

opscenterd.conf

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

- **Package installations:** /etc/opscenter/opscenterd.conf
- **Tarball installations:** install_location/conf/opscenterd.conf

Enable Hypertext Transfer Protocol Secure (HTTPS) support in OpsCenter and specify SSL information for better security. You can enable or disable HTTPS support for OpsCenter. To enable HTTPS, follow the steps below. For additional security, enable HTTP Strict Transport Security to enforce OpsCenter to return an HSTS header for added protection against protocol downgrade attacks or cookie hijacks.

**Note:** Disabling HTTPS in an HSTS environment can be time-consuming. See the prerequisites in Disabling HTTPS with HSTS for the OpsCenter Server (page 80).

1. Open the OpsCenter configuration file, opscenterd.conf.

2. Scroll to the [webserver] section.

   This snippet from opscenterd.conf shows the default [webserver] section to change:

   ```
   [webserver]
   port = 8888
   interface = 127.0.0.1
   # The following settings can be used to enable ssl support for the
   # opscenter web application. Change these values to point to the
   # ssl certificate and key that you wish to use for your OpsCenter
   # install, as well as the port you would like
   # to serve ssl traffic from.
   #ssl_keyfile = /var/lib/opscenter/ssl/opscenter.key
   #ssl_certfile = /var/lib/opscenter/ssl/opscenter.pem
   ```
3. Remove the comment markers (#) in front of `ssl_keyfile`, `ssl_certfile`, and `ssl_port`.

   Use the default values for `ssl_keyfile` and `ssl_certfile`, or replace them with the path to your own private and public certificates.

   **Tip:** See the [OpsCenter ports reference](#page 68) for ports information.

   If your organization is using certificates signed by a commercial certificate authority like Verisign or Thawte, you must provide the complete certificate chain. In addition to the certificate that you were issued, this includes a root certificate and typically one or more intermediate (or chained) certificates. Your certificate provider can help you determine the necessary list of certificates. The PEM format allows concatenating multiple certificates together. For certificates with a trust chain, add the whole chain into a single PEM file and specify the location in `ssl_certfile`. Digicert has more information detailing certificate concatenation in PEM files: [https://www.digicert.com/ssl-support/pem-ssl-creation.htm](https://www.digicert.com/ssl-support/pem-ssl-creation.htm).

4. Enable the HTTP Strict Transport Security option to force OpsCenter to return an HSTS header in HTML responses that go over HTTPS. The HSTS maximum age represents the length of time in seconds that supported browsers should consider an HSTS header fresh, which is 1 year by default. If the max age has been exceeded, browsers refuse to connect to OpsCenter with unencrypted HTTP.

   ```
   [webserver]
   port = 8888
   interface = 127.0.0.1
   ssl_keyfile = /var/lib/opscenter/ssl/opscenter.key
   ssl_certfile = /var/lib/opscenter/ssl/opscenter.pem
   ssl_port = 8443
   hsts_enabled = True
   hsts_max_age = 31536000
   ```


5. Save `opscenterd.conf` and **restart OpsCenter** *(page 75).*

**Disabling HTTPS with HSTS for the OpsCenter Server**

Disable HTTPS with HSTS for the OpsCenter server. To disable HTTPS without HSTS, simply comment out the SSL entries. If HTTPS is used in an HSTS environment, follow the steps in this procedure to phase out the HSTS header before altogether disabling HTTPS with HSTS.

**Prerequisites:**
Configuring OpsCenter

Removing HTTPS from an OpsCenter server is an uncommon action in production systems, but may be necessary during development and testing. HSTS complicates the process of disabling HTTPS because browsers that have recently received an HSTS header will refuse to connect to OpsCenter over unencrypted HTTP even if the HTTPS service has been disabled. When disabling HTTPS with HSTS, first lower \texttt{hsts\_max\_age} to a very small value such as 10 seconds (the default is 31536000 seconds, which equates to 1 year). Drastically lowering the value effectively disables HSTS on each browser that subsequently connects to OpsCenter and receives the updated header, ensuring that the header expires almost immediately. Continue to run OpsCenter with HTTPS enabled and a very short \texttt{hsts\_max\_age} for a sufficient period of time so that all browsers that visit OpsCenter receive the updated header. At that point, HTTPS and HSTS can be disabled entirely without disrupting the ability of supported browsers to connect to Opscenter.

If HTTPS has already been disabled and browsers are currently unable to connect to OpsCenter, the following options are available:

- HTTPS can be re-enabled (page 79) and \texttt{hsts\_max\_age} lowered as described above.
- Manually clear the HSTS header cache in browsers. This requires visiting every browser that accesses OpsCenter, and referring to browser manufacturer documentation on clearing the HSTS header cache.

\texttt{opscenterd.conf}

The location of the \texttt{opscenterd.conf} file depends on the type of installation:

- Package installations: /\texttt{etc/opscenter/opscenterd.conf}
- Tarball installations: \texttt{install\_location/conf/opscenterd.conf}

1. Open the OpsCenter configuration file, \texttt{opscenterd.conf}.

2. In the \texttt{[webserver]} section, make the following changes as appropriate:
   
   a. If HSTS is enabled with a long max age, lower the \texttt{hsts\_max\_age} value.
   
   b. Save \texttt{opscenterd.conf} and restart OpsCenter (page 75).
   
   c. Run OpsCenter in this state for a while until you’re confident that all browser clients have connected and received updated hsts-headers with a short max-age.

3. Open \texttt{opscenterd.conf} again. Comment out the ssl and max age options. Set \texttt{hsts\_enabled} to False (or comment out) as shown in the following example:

```plaintext
[webserver]
port = 8888
interface = 127.0.0.1
#ssl\_keyfile = /\texttt{var/lib/opscenter/ssl/opscenter.key}
```
Configuring OpsCenter

```bash
#ssl_certfile = /var/lib/opscenter/ssl/opscenter.pem
#ssl_port = 8443
hsts_enabled = False
#hsts_max_age = 31536000
```

4. Save `opscenterd.conf` and **restart OpsCenter (page 75)**.

   If any clients with HSTS headers are unable to connect, either re-enable https with a short max age or consult your browser vendor docs about clearing the HSTS header cache.

**Configuring SSL/TLS between OpsCenter and the DataStax Agents**

OpsCenter uses Transport Layer Security (TLS), referred to as its predecessor Secure Socket Layer (SSL), to encrypt the communication protocol and authenticate traffic between DataStax agents and the main OpsCenter daemon. By default, SSL is disabled.

**Warning:** Running OpsCenter without SSL should only be done when running OpsCenter and DataStax Enterprise under the following conditions:

- On a secure internal network.
- In a development environment where agents and OpsCenter run on the same computer protected from network threats.
- In a situation where there is no concern about someone listening to OpsCenter traffic.

Otherwise, configuring SSL between OpsCenter and agents is strongly recommended as a security best practice.

**Enabling SSL/TLS for OpsCenter and Agent communication - Package Installs**

Enable SSL between OpsCenter and the DataStax Agent by editing the `opscenterd.conf` configuration file.

**Note:** The SSL key generation process is done automatically at package install time. Unless you need to regenerate SSL key files, skip to 6 (page 84) to edit the proper configuration files to enable SSL.

**Prerequisites:** OpsCenter requires the `.der` file format for SSL. If the existing [agents] `ssl_certfile (page 186)` in `opscenterd.conf` is in a `.pem` format, run the following command to convert the format:

```bash
$ openssl x509 -outform der -in /var/lib/opscenter/ssl/opscenter.pem -out /var/lib/opscenter/ssl/opscenter.der
```

For more information about SSL cert file formats, see converting SSL certificates.
1. If the SSL files already exist in the /usr/share/opscenter/ssl directory, they are not automatically recreated. Before running setup.py in 3 (page 83), remove the old SSL files from that directory.

2. Make sure that the /tmp directory does not have the noexec flag set.

   The setup.py script requires the /tmp directory to have exec permissions. See Setting and securing the tmp directory for the agent (page 176).

3. Run the OpsCenter setup.py script:

   $ sudo /usr/share/opscenter/bin/jython /usr/share/opscenter/bin/setup.py

   The script generates the SSL keys and certificates used by the OpsCenter daemon and the agents to communicate with one another in the following directory: /usr/share/opscenter/ssl.

4. Copy the following files to /var/lib/opscenter/ssl:

   - opscenter.key: OpsCenter SSL private key
   - opscenter.der: OpsCenter SSL certificate
   - agentKeyStore: DataStax Agent keystore
   - agentKeyStore.der: DataStax Agent SSL certificate
   - agentKeyStore.key: DataStax Agent keyfile

   Note: The agentKeyStore.key keyfile is necessary only for non-Java clients.

   $ sudo cp /usr/share/opscenter/ssl/opscenter.key /var/lib/opscenter/ssl

   $ sudo cp /usr/share/opscenter/ssl/opscenter.der /var/lib/opscenter/ssl

   $ sudo cp /usr/share/opscenter/ssl/agentKeyStore /var/lib/opscenter/ssl

   $ sudo cp /usr/share/opscenter/ssl/agentKeyStore.der /var/lib/opscenter/ssl

   $ sudo cp /usr/share/opscenter/ssl/agentKeyStore.key /var/lib/opscenter/ssl

   For non-Java clients, copy the agentKeyStore.key keyfile as well.

   $ sudo cp /usr/share/opscenter/ssl/agentKeyStore.key /var/lib/opscenter/ssl

5. Change ownership of the files copied in the previous step to the opscenter user and the opscenter group:
$ sudo chown opscenter:opscenter /var/lib/opscenter/ssl/opscenter.key
$ sudo chown opscenter:opscenter /var/lib/opscenter/ssl/opscenter.der
$ sudo chown opscenter:opscenter /var/lib/opscenter/ssl/agentKeyStore
$ sudo chown opscenter:opscenter /var/lib/opscenter/ssl/agentKeyStore.der

For non-Java clients, change the ownership for the agentKeyStore.key keyfile as well.

$ sudo chown opscenter:opscenter /var/lib/opscenter/ssl/agentKeyStore.key

6. Open `opscenterd.conf` in an editor and add an `[agents]` section with the `use_ssl` option set to enable SSL. See `configuring the agent for ssl (page 205)` for more details on the `use_ssl` option.

```
[agents]
use_ssl = true
ssl_keyfile = /var/lib/opscenter/ssl/opscenter.key
ssl_certfile = /var/lib/opscenter/ssl/opscenter.der
agent_keyfile = /var/lib/opscenter/ssl/agentKeyStore
agent_keyfile_raw = /var/lib/opscenter/ssl/agentKeyStore.key
agent_certfile = /var/lib/opscenter/ssl/agentKeyStore.der
```

7. Restart (page 75) the OpsCenter daemon.

8. If you need to connect to a cluster in which agents have already been deployed, log in to each of the nodes and reconfigure the `address.yaml` file. Reconfigure the agents on all nodes.

   **Tip:** If you do not want to manually edit all of the node configuration files, follow the procedure to install DataStax Agents automatically.

   a. Copy `/var/lib/opscenter/ssl/agentKeyStore` from the OpsCenter machine to `/var/lib/datastax-agent/ssl/agentKeyStore` on each node in the cluster.

   ```bash
   $ scp /var/lib/opscenter/ssl/agentKeyStore user@node:/var/lib/datastax-agent/ssl/
   ```
Where `node` is either the host name of the node or its IP address and `user` is the user ID on the node.

b. Log into each node in the cluster using `ssh`.

```
$ ssh user@node
```

Where `node` is either the host name of the node or its IP address and `user` is the user ID on the node.

c. Edit the `address.yaml` file, changing the value of `use_ssl` to 1.

```
$ sudo vi /var/lib/datastax-agent/conf/address.yaml
```

use_ssl: 1

d. Restart the agent.

```
$ sudo service datastax-agent restart
```

9. After `opscenterd` and all agents have been configured and restarted, verify proper agent connection through the Agent Status (page 230) tab.

**Enabling SSL/TLS for OpsCenter and Agent communication - Tarball Installations**

`opscenterd.conf`

The location of the `opscenterd.conf` file depends on the type of installation:

- Package installations: `/etc/opscenter/opscenterd.conf`
- Tarball installations: `install_location/conf/opscenterd.conf`

`address.yaml`

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

- Package installations: `/var/lib/datastax-agent/conf/address.yaml`
- Tarball installations: `install_location/conf/address.yaml`

To enable SSL for tarball installations, edit the configuration file and run a script to generate the keys used by OpsCenter and the agents.

**Prerequisites**: OpsCenter requires the `.der` file format for SSL. If your existing `[agents] ssl_certfile` in `opscenterd.conf` is in a `.pem` format, run the following command to convert the format:

```
$ openssl x509 -outform der -in /install_location/ssl/opscenter.pem -out /install_location/ssl/opscenter.der
```
Configuring OpsCenter

[agents] ssl_certfile
The location of the SSL certificate used for SSL traffic between OpsCenter and the DataStax Agents. The default location is /var/lib/opscenter/ssl/opscenter.der for package installations and install_location/ssl/opscenter.der for tarball installations.

For more information about SSL cert file formats, see converting SSL certificates.

1. **Warning:** If the SSL files already exist in the install_location/ssl directory, they are not automatically recreated. Before running setup.py, remove the old SSL files from that directory.

Run the OpsCenter setup.py script:

```
$ sudo install_location/bin/jython install_location/bin/setup.py
```

The script generates the SSL keys and certificates used by the OpsCenter daemon and the agents to communicate with one another in the following directory:

```
install_location/ssl
```

2. Open opscenterd.conf in an editor and add an [agents] section with the use_ssl option set to enable SSL. See configuring the agent for ssl (page 206) for more details on the use_ssl option.

```
$ sudo vi install_location/conf/opscenterd.conf
```

```
[agents]
use_ssl = true
ssl_keyfile = install_location/ssl/opscenter.key
ssl_certfile = install_location/ssl/opscenter.der
agent_keyfile = install_location/ssl/agentKeyStore
agent_keyfile_raw = install_location/ssl/agentKeyStore.key # for non-Java clients
agent_certfile = install_location/ssl/agentKeyStore.der
```

3. **Restart** (page 75) the OpsCenter daemon.

4. If you need to connect to a cluster in which agents have already been deployed, log in to each of the nodes and reconfigure the address.yaml file. Reconfigure the agents on all nodes.

   **Tip:** If you do not want to manually edit all of the node configuration files, follow the installing DataStax agents automatically (page 59) procedure.

   a. On each node in the cluster, copy install_location/ssl/agentKeyStore from the OpsCenter machine to /var/lib/datastax-agent/ssl/agentKeyStore for agent_install_location/ssl/
Configuring OpsCenter

agentKeyStore for package installs, or agent_install_location/ssl/agentKeyStore for tarball installs.

```bash
$ scp /opt/opscenter/ssl/agentKeyStore user@node:/var/lib/datastax-agent/ssl/
```

Where `node` is either the host name of the node or its IP address and `user` is the user ID on the node.

**b.** Log into each node in the cluster using `ssh`.

```bash
$ ssh user@node
```

Where `node` is either the host name of the node or its IP address and `user` is the user ID on the node.

c. **Edit the address.yaml file, changing the value of `use_ssl` to 1.**

```bash
$ sudo vi install_location/conf/address.yaml
```

```
use_ssl: 1
```

d. **Restart the agent.**

```bash
$ sudo install_location/bin/datastax-agent
```

5. After opscenterd and all agents have been configured and restarted, verify proper agent connection through the **Agent Status (page 230)** tab.

**Disabling SSL/TLS for OpsCenter and Agent communication - Package Installations**

opscenterd.conf

The location of the `opscenterd.conf` file depends on the type of installation:

- Package installations: `/etc/opscenter/opscenterd.conf`
- Tarball installations: `install_location/conf/opscenterd.conf`

address.yaml

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

- Package installations: `/var/lib/datastax-agent/conf/address.yaml`
- Tarball installations: `install_location/conf/address.yaml`

By default SSL is turned off in OpsCenter. Perform this task if you have configured the agents on a cluster to use SSL previously and now need to turn SSL off.
1. Open `opscenterd.conf` in an editor and add the following to disable SSL.

   ```
   $ sudo vi /etc/opscenter/opscenterd.conf
   [agents]
   use_ssl = false
   ```

2. **Restart (page 75)** the OpsCenter daemon.

3. Reconfigure the agents.

   **Tip:** If you do not want to manually edit all of the node configuration files, follow the installing DataStax agents automatically (page 59) procedure.

   a. Log into each node in the cluster using `ssh`.

   ```
   $ ssh user@node
   ```

   Where `node` is either the host name of the node or its IP address and `user` is the user ID on the node.

   b. Edit the `address.yaml` file, changing the value of `use_ssl` to 0.

   ```
   $ sudo vi /var/lib/opscenter/address.yaml
   use_ssl: 0
   ```

   c. Restart the agent.

   ```
   $ sudo service datastax-agent restart
   ```

4. After `opscenterd` and all agents have been configured and restarted, verify proper agent connection through the Agent Status (page 230) tab.

**Disabling SSL/TLS for OpsCenter and Agent communication - Tarball Installations**

`opscenterd.conf`

The location of the `opscenterd.conf` file depends on the type of installation:

- Package installations: `/etc/opscenter/opscenterd.conf`
- Tarball installations: `install_location/conf/opscenterd.conf`
Configuring OpsCenter

address.yaml

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

- **Package installations:** /var/lib/datastax-agent/conf/address.yaml
- **Tarball installations:** install_location/conf/address.yaml

By default, SSL is turned off in OpsCenter. You only need to perform this task if you have previously configured the agents on a cluster to use SSL and now want to turn SSL off.

1. Open opscenterd.conf in an editor and add the following lines to disable SSL.

   ```
   $ vi install_location/conf/opscenterd.conf
   
   [agents]
   use_ssl = false
   ```

2. **Restart (page 75)** the OpsCenter daemon.

3. Reconfigure the agents.

   **Tip:** If you do not want to manually edit all of the node configuration files, follow the installing DataStax agents automatically (page 59) procedure.

   a. Log into each node in the cluster using ssh.

      ```
      $ ssh user@node
      
      Where node is either the host name of the node or its IP address and user is the user ID on the node.
      ```

   b. Edit the address.yaml file, changing the value of use_ssl to 0.

      ```
      $ sudo vi install_location/conf/address.yaml
      
      use_ssl: 0
      ```

   c. Restart the agent.

      ```
      $ sudo install_location/bin/datastax-agent
      ```
4. After `opscenterd` and all agents have been configured and restarted, verify proper agent connection through the **Agent Status** *(page 230)* tab.

**Connect to DSE with client-to-node encryption in OpsCenter and the DataStax Agents**

The `opscenterd` machine and the DataStax Agents act as clients to each DSE machine. Therefore, `opscenterd` and the DataStax Agents need to use their own keystores (to present its certificate) and truststores (to verify the DSE server certificate) during the SSL handshake process.

**Note:** If two-way auth is enabled, DSE needs to verify traffic from `opscenterd` and the DataStax Agents using a truststore.

Follow these instructions to set up OpsCenter and the DataStax Agents to use client-to-node encryption using one-way or two-way auth.

**Prerequisites:**
- **Client-to-node encryption** must be enabled on the DSE cluster.
- The keystores and truststores (optional) must be created for all DSE nodes. For more information, see **Setting up SSL certificates** in the DSE documentation.

**Figure 3:** Client-to-node encryption

![Diagram showing SSL connection between OpsCenter Client, DataStax Agent Client, and DSE Nodes]

**cluster_name.conf**

The location of the `cluster_name.conf` file depends on the type of installation:
- **Package installations:** `/etc/opscenter/clusters/cluster_name.conf`
- **Tarball installations:** `install_location/conf/clusters/cluster_name.conf`

1. Create a keystore on the `opscenterd` machine.
$ keytool -genkey -alias opscenter -keyalg RSA -keypass password -storepass password -keystore opscenter.jks

This command creates a keystore named opscenter.jks in the current directory.

2. Export the opscenterd certificate.

$ keytool -export -alias opscenter -storepass password -file opscenter.crt -keystore opscenter.jks

This command exports the certificate named opscenter.crt that was stored in opscenter.jks in the current directory.

3. Create a truststore on the opscenterd machine and import each node’s public certificate.

$ keytool -import -v -trustcacerts -alias node1 -file node1.crt -keystore truststore.jks -keypass password

The command creates a truststore by importing node1’s certificate. Repeat this command using the certificate from each node.

4. **Note:** This step is optional and should be done only if two-way-auth is enabled when using DSE client-to-node encryption. This should be done on every node in the cluster.

Import the opscenterd certificate into the truststore on every DSE node.

$ keytool -import -v -trustcacerts -alias opscenter -file opscenter.crt -keystore dse_truststore.jks -keypass password

This command imports (and trusts) the opscenter.crt certificate into a truststore named dse_truststore.jks.

5. **Note:** The DataStax Agent uses one file as both a keystore and truststore for OpsCenter versions earlier than 6.1.1. Therefore, each agent can reuse the same truststore that was created when enabling client-to-node encryption in DSE. The only additional step is to add the certificate to the truststore for the node in which the DataStax Agent is monitoring.

Import the DSE certificate into its truststore so that the DataStax Agent can use the combined truststore and keystore for client-to-node encryption. Repeat for each node.

$ keytool -import -alias node1 -file node1.crt -keystore dse_truststore.jks -storepass password
This command imports `node1.crt` into the truststore file `dse_truststore.jks`, thus making `dse_truststore.jks` both a truststore and keystore. Repeat this process on every DSE machine.

6. Configure OpsCenter and the DataStax agents to use client-to-node encryption.

When client-to-node encryption (page 604) is enabled for a cluster using Lifecycle Manager, the `ssl_truststore` and `ssl_truststore_password` fields are automatically propagated in `cluster_name.conf` with the corresponding values from `ssl_keystore` and `ssl_keystore_password` for both opscnter and the agent:

LCM propagates the `ssl_keystore` value into `ssl_keystore` and `ssl_truststore`; and the `ssl_keystore_password` value into `ssl_keystore_password` and `ssl_truststore_password`.

- **a.** Configure the client-to-node settings using the **Edit Cluster Connection Settings** dialog in the OpsCenter Monitoring UI. Entering the values in the UI populates the corresponding configuration options in `cluster_name.conf`. See **Configuring cluster settings** (page 93).

- **b.** Or, configure the options directly in the cluster configuration file `cluster_name.conf`.

  Edit the cluster configuration file and enter the information for the SSL keystore and SSL truststore created in previous steps.

  **Note:** If using a separate storage cluster (page 154) (recommended), one additional keystore/truststore needs to be created that holds the certs of the nodes in the storage cluster as well as the cert/key for the agent machine. The path and password to said keystore/truststore should go in the `[agents]` section of the cluster configuration file.

```plaintext
[cassandra]
# Note: If the truststore and keystore are the same file, enter
# the same path/password
# for both the keystore and truststore
ssl_keystore = /path/to/keystore/file/on/opsc/machine
ssl_keystore_password = password_of_keystore
ssl_truststore = /path/to/truststore/file/on/opsc/machine
ssl_truststore_password = password_of_truststore

[agents]
# The agent uses one keystore/truststore file that operates as
# both a keystore and a truststore
# Applicable to OpsCenter versions earlier than 6.1.1
ssl_keystore = /path/to/dse/truststore/file/on/agent/machine
ssl_keystore_password = password_of_keystore
```

---

Page 92

OpsCenter 6.1 User Guide Earlier version, latest patch 6.1.11
# The agent has separate keystore and truststore
# Applicable to OpsCenter versions 6.1.1 and later.
ssl_truststore = /path/to/trusted/certs
ssl_truststore_password = pw_for_agents_to_access_trusted_certs

# Optional if using a separate storage cluster
storage_ssl_keystore = /path/to/storage_cluster/keystore/file/on/agent/machine
storage_ssl_keystore_password = password_of_keystore
# Separate truststore options applicable to OpsCenter versions 6.1.1 and later
storage_ssl_truststore = /path/to/trusted/certs
storage_ssl_truststore_password = pw_for_agents_to_access_trusted_certs

7. Restart OpsCenter (page 75).

**Editing OpsCenter cluster connections for authentication or encryption**

The connection settings for a cluster define how OpsCenter connects to a DSE cluster. For example, if Kerberos authentication or client-to-node encryption was enabled on a cluster, you need to specify that information in the cluster connection settings.

cluster_name.conf

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

- **Package installations:** `/etc/opscenter/clusters/cluster_name.conf`
- **Tarball installations:** `install_location/conf/clusters/cluster_name.conf`

1. Select the cluster to edit from the **Cluster** menu in OpsCenter Monitoring.

2. Click **Settings#Cluster Connections**.
   
   The **Edit Cluster Connection Settings** dialog appears.
Tip: Select other clusters to edit connection settings for from the Cluster list.

3. If applicable, change the IP addresses of cluster nodes.

4. If applicable, change the JMX Port and Native Transport Port listen port (page 68) numbers if you are not using the defaults.

5. If applicable, click JMX is enabled on my cluster to add or edit user credentials (username and passwords) if the JMX port requires authentication.

6. If applicable, click Native transport security is enabled on my cluster to add or edit user credentials (username and password) if the Native Transport port requires authentication.

7. If applicable to your environment, select DSE security (kerberos) is enabled on my cluster and complete the fields.
a. Enter the **Service Name**. For example, if the server principal on your nodes is `dse/nodeX.example.com@EXAMPLE.COM`, this field should be `dse`.

b. Enter the **Opscenterd Client Principal** for the OpsCenter process/machine to use. Example: `opscenterd/opscenterd.EXAMPLE.COM`.

c. Enter the location of the keytab OpsCenter machine in **Opscenterd Keytab Location**, which contains credentials for the `opscenter_client_principal`. Example: `/etc/opscenter/security/krb5_opsc.keytab`.

d. Enter the client principal for the DataStax Agent process/machine to use in **DataStax Agent Client Principal**. Example: `dxagent/_HOST`.

   **Important:** Because each `datastax-agent` has a different principal name, the DataStax Agent Client Principal entered in this field is a placeholder. The `kerberos_client_principal` (*page 136*) property must be set in the `address.yaml` file for each `datastax-agent`. For example:

   ```yaml
   kerberos_client_principal: datastax-agent@dsenode1/dsenode2/dsenode3
   ```

e. Enter the location of the keytab on the DataStax Agent machines in **DataStax Agent Keytab Location**, which contains credentials for the `agent_client_principal`. Example: `/usr/agent/conf/krb5_agent.keytab`.

For more information, see Setting up Kerberos and the Kerberos tutorial.

8. If configuring client-to-node settings, select **Client-to-node encryption is enabled on my cluster**. Indicate the following paths for OpsCenter and each agent to use for connecting directly to the monitored DSE cluster.
When client-to-node encryption (page 604) is enabled for a cluster using Lifecycle Manager, the ssl_truststore and ssl_truststore_password fields are automatically propagated in cluster_name.conf with the corresponding values from ssl_keystore and ssl_keystore_password for both opscenterd and the agent: LCM propagates the ssl_keystore value into ssl_keystore and ssl_truststore; and the ssl_keystore_password value into ssl_keystore_password and ssl_truststore_password.

![Image](image_url)

**Note:** For information about creating keystores and truststores, see Enabling client-to-node encryption in OpsCenter (page 90).

a. Enter the **OpsCenter Keystore Path**, which is the SSL keystore location for OpsCenter (opscenterd) to use for connecting to the monitored DSE cluster. The value entered in the UI populates the [cassandra] ssl_keystore property in the OpsCenter cluster configuration file (cluster_name.conf).

b. Enter the **Password** for the **OpsCenter Keystore Path**. The value entered in the UI populates the [cassandra] ssl_keystore_password property in cluster_name.conf.

c. Enter the **OpsCenter Truststore Path**, which is the SSL truststore location for OpsCenter (opscenterd) to use for connecting to the monitored DSE cluster. This value should be the same as **OpsCenter Keystore Path** if the same file is used as both the keystore and the truststore (that is, there is not a separate truststore). The value entered in the UI populates the [cassandra] ssl_truststore property in cluster_name.conf.
d. Enter the **Password** for the **OpsCenter Truststore Path**. This value should be the same as password for the **OpsCenter Keystore Path** if the same file is used as both the keystore and the truststore (that is, there is not a separate truststore). The value entered in the UI populates the `[cassandra] ssl_truststore_password` property in `cluster_name.conf`.

e. Enter the **Agent Keystore Path**, which is the SSL keystore location for each agent to use for connecting to the monitored DSE cluster. The value entered in the UI populates the `[agents] ssl_keystore` property in `cluster_name.conf`.

f. Enter the **Password** for the **Agent Keystore Path**. The value entered in the UI populates the `[agents] ssl_keystore_password` property in `cluster_name.conf`.

g. Enter the **Agent Truststore Path**, which is the SSL truststore location for each agent to use for connecting to the monitored DSE cluster. This value should be the same as password for the **Agent Keystore Path** if the same file is used as both the keystore and the truststore (that is, there is not a separate truststore). The value entered in the UI populates the `[agents] ssl_truststore` property in `cluster_name.conf`.

h. Enter the **Password** for the **Agent Truststore Path**. This value should be the same as password for the **Agent Keystore Path** if the same file is used as both the keystore and the truststore (that is, there is not a separate truststore). The value entered in the UI populates the `[agents] ssl_truststore_password` property in `cluster_name.conf`.

9. Click **Save Cluster**.

**Related information:**

Disconnecting a cluster from OpsCenter and Lifecycle Manager

[Disconnect a cluster from monitoring within OpsCenter and managing within Lifecycle Manager. Disconnecting a cluster from OpsCenter does not delete the cluster itself.](page 280)

**SSL configuration options for OpsCenter**

Reference of available SSL configuration options in each level of OpsCenter configuration file (page 180) (daemon, cluster, agent), sorted by alphabetical order within each section.

OpsCenter daemon (opscenterd) SSL configuration options

SSL configuration options available in opscenterd.conf:

[agents] **agent_certfile**

The location of the certfile sent to the DataStax Agents when using SSL communication between OpsCenter and the DataStax Agents. The default location is `/var/lib/opscenter/ssl/agentKeyStore.der` for package
installations and install_location/ssl/agentKeyStore.der for tarball installations.

**[agents] agent_keyfile**

The location of the keyfile sent to the DataStax Agents when using SSL communication between OpsCenter and the DataStax Agents. The default location is /var/lib/opscenter/ssl/agentKeyStore for package installations and install_location/ssl/agentKeyStore for tarball installations.

*Note:* Do not use the agent_keyfile when manually generating and deploying keys.

**[agents] agent_keyfile_raw**

The raw keystore file stored in the Java keystore from agent_keyfile. This parameter is required only when configuring high availability, so that the secondary OpsCenter instance can communicate with the primary OpsCenter instance. The failover opscenterd processes on the secondary OpsCenter instance use this key to establish a STOMP connection to the primary opscenterd instance.

**[agents] ssl_certfile**

The location of the SSL certificate used for SSL traffic between OpsCenter and the DataStax Agents. The default location is /var/lib/opscenter/ssl/opscenter.der for package installations and install_location/ssl/opscenter.der for tarball installations.

**[agents] ssl_keyfile**

The location of the SSL key file used for SSL traffic between OpsCenter and the DataStax Agents. The default location is /var/lib/opscenter/ssl/opscenter.key for package installations and install_location/ssl/opscenter.key for tarball installations.

**[agents] use_ssl**

Specifies whether traffic between OpsCenter and the DataStax Agents should use SSL. The default value is False.

**[webserver] ssl_certfile**

The location where the SSL certificate resides. This option requires ssl_keyfile and optionally ssl_port to also be set.

**[webserver] ssl_keyfile**

The location where the SSL keyfile resides. This option requires ssl_certfile and optionally ssl_port to also be set.

**[webserver] ssl_port**

The port on which to serve SSL traffic. The default port is 8443.

Cluster SSL configuration options

Cluster-specific SSL configuration options available in cluster_name.conf:

**[agents] ssl_keystore**

The SSL keystore location for DataStax Agents to use to connect to CQL on the monitored cluster.

**[agents] ssl_keystore_password**
Configuring OpsCenter

The SSL keystore password for DataStax Agents to use to connect to CQL on the monitored cluster.

[agents] ssl_truststore
The SSL truststore location for DataStax Agents to use for trusted certs.

[agents] ssl_truststore_password
The SSL truststore password for DataStax Agents to use for trusted certs.

[agents] storage_ssl_keystore
The SSL keystore location for DataStax Agents to use to connect to CQL on the storage cluster.

[agents] storage_ssl_keystore_password
The SSL keystore password for DataStax Agents to use to connect to CQL on the storage cluster.

[agents] storage_ssl_truststore
The SSL truststore location for DataStax Agents to use for trusted certs on the storage cluster.

[agents] storage_ssl_truststore_password
The SSL truststore password for DataStax Agents to use for trusted certs on the storage cluster.

[cassandra] ssl_keystore
The SSL keystore location for OpsCenter to use to connect to Cassandra directly.

[cassandra] ssl_keystore_password
The SSL keystore password for OpsCenter to use to connect to Cassandra directly.

[cassandra] ssl_truststore
The SSL truststore location for OpsCenter to use to connect to Cassandra directly.

[cassandra] ssl_truststore_password
The SSL truststore password for OpsCenter to use to connect to Cassandra directly.

[storage_cassandra] ssl_keystore
The SSL keystore location for OpsCenter to use to connect to Cassandra directly.

[storage_cassandra] ssl_keystore_password
The SSL keystore password for OpsCenter to use to connect to Cassandra directly.

[storage_cassandra] ssl_truststore
The SSL truststore location for OpsCenter to use to connect to Cassandra directly.

[storage_cassandra] ssl_truststore_password
The SSL truststore password for OpsCenter to use to connect to Cassandra directly.

Agent configuration options

SSL configuration options available for agents in address.yaml:

monitored_ssl_keystore
The SSL keystore location for the monitored cluster that agents use to connect to CQL. Example: monitored_ssl_keystore: /etc/dse/conf/.keystore

monitored_ssl_keystore_password
Configuring OpsCenter

The SSL keystore password for the monitored cluster that agents use to connect to CQL. Example: `{monitored_ssl_keystore_password: keystore-pass}` [This field may be encrypted for additional security.]

**monitored_ssl_truststore**
The SSL truststore location for the monitored cluster that agents use to connect to CQL. Example: `{monitored_ssl_truststore: /etc/dse/conf/.truststore`.

**monitored_ssl_truststore_password**
The SSL truststore password for the monitored cluster that agents use to connect to CQL. Example: `{monitored_ssl_truststore_password: truststore-pass}` [This field may be encrypted for additional security.]

**opscenter_ssl_keystore**
On target nodes where DataStax Agents are running, the path to the SSL keystore file that the Agents use to connect to opscenterd. Example: `{opscenter_ssl_keystore: /usr/share/opscenter/ssl/agentKeystore`.

**opscenter_ssl_key_password**
The SSL keystore password that the agents use to connect to opscenterd. Example: `{opscenter_ssl_key_password: keystore-pass}` [This field may be encrypted for additional security.]

**opscenter_ssl_truststore**
The path to the truststore file that the agents use to connect to opscenterd. Example: `{opscenter_ssl_truststore: /usr/share/opscenter/ssl/trustStore`.

**opscenter_ssl_truststore_password**
The SSL truststore password that the agents use to connect to opscenterd. Default: Uses the keystore password if an SSL truststore password is not specified. Example: `{opscenter_ssl_truststore_password: trust-pass}` [This field may be encrypted for additional security.]

**ssl_truststore**
The SSL truststore location for the storage cluster that agents use to connect to CQL. Example: `{ssl_truststore: /etc/dse/conf/.truststore`.

**ssl_truststore_password**
The SSL truststore password for the storage cluster that agents use to connect to CQL. Example: `{ssl_truststore_password: truststore-pass}` [This field may be encrypted for additional security.]

**use_ssl**
Whether or not to use SSL communication between the agent and opscenterd. Affects both the STOMP connection and agent HTTP server. Corresponds to `[agents].use_ssl in opscenterd.conf`. Setting this option to true turns on SSL connections. Example: `{use_ssl: true`.

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

- Package installations: `/etc/opscenter/clusters/cluster_name.conf`
- Tarball installations: `install_location/conf/clusters/cluster_name.conf`
Configuring OpsCenter

opscenterd.conf

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

- Package installations: /etc/opscenter/opscenterd.conf
- Tarball installations: install_location/conf/opscenterd.conf

address.yaml

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

- Package installations: /var/lib/datastax-agent/conf/address.yaml
- Tarball installations: install_location/conf/address.yaml

Configuring OpsCenter role-based security

By default, access control is disabled. Any user that knows the OpsCenter URL can view all objects and perform all tasks.

OpsCenter access roles overview

DataStax Enterprise (DSE) customers have the ability to define custom, fine-grained access roles for their users. OpsCenter can be configured to require users to log in using OpsCenter authentication. Permissions to perform certain operations can be granted to each role, and a role can be assigned to users. A user can only be assigned one role, and each role applies to all clusters.

Note: Authenticating with LDAP in OpsCenter requires defining roles (page 125) for LDAP users. If using LDAP authentication, users can have multiple roles. Upon logging in, all permissions for each role a user is assigned to are merged.

Admin role privileges

The admin role is built-in to OpsCenter and cannot be edited or removed. By default, the admin role is the only role created automatically when authentication is enabled. Only users with the admin role can manage users and roles, add new clusters, or manually update definition files.

Important: Changing the default admin password is strongly recommended the first time you log in.

Custom user role privileges

Only those assigned an admin role can manage roles. Each role represents permissions for all clusters managed by OpsCenter. Any functionality in OpsCenter that a user does not have permission for appears as gray and unavailable to that logged in user.

If using the OpsCenter API, users without sufficient permissions will receive an HTTP 401, Unauthorized response from the API.
Configuring OpsCenter

**Note:** Adding a cluster does not automatically add permissions for any existing roles. After adding a cluster, apply the permissions to the cluster for each role as appropriate for your organization.

**Important:** In OpsCenter 6.5.3 and later, you must update custom scripts and applications that use the OpsCenter API if you want to use 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.

Role permissions

When defining custom roles, each role can have specific permissions enabled for that role. Each user can only be assigned a single role, which contains permissions for all clusters managed by OpsCenter. If using LDAP authentication, users can have multiple roles. Use the **Cluster** menu to view permissions for each cluster for a selected role. To hide a cluster for users within a selected role, uncheck all permissions.
# Table 4: Role permissions

<table>
<thead>
<tr>
<th>Permission</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Core functionality</strong></td>
<td></td>
</tr>
<tr>
<td>View Cluster</td>
<td>Allows users to view a cluster in the Clusters area of the OpsCenter Monitoring UI.</td>
</tr>
<tr>
<td>Install Agents <em>(page 59)</em></td>
<td>Allows users to install or upgrade agents automatically or manually.</td>
</tr>
<tr>
<td>Edit Connection Settings</td>
<td>Allows users to edit the cluster connection settings <em>(page 93)</em> for a DSE cluster monitored in OpsCenter.</td>
</tr>
<tr>
<td>Manage Alerts</td>
<td>Allows users to add alerts for monitoring conditions in DSE clusters.</td>
</tr>
<tr>
<td>Cluster Configuration</td>
<td>Allows users to configure the Performance Service.</td>
</tr>
<tr>
<td><strong>Services</strong></td>
<td></td>
</tr>
<tr>
<td>Backup Service <em>(page 380)</em></td>
<td>Allows users to perform backups and restores.</td>
</tr>
<tr>
<td>Best Practice Service <em>(page 502)</em></td>
<td>Allows users to configure and schedule Best Practice Service rules for managing DSE clusters.</td>
</tr>
<tr>
<td>Repair Service <em>(page 462)</em></td>
<td>Allows users to start, stop, and configure the Repair Service for running repairs on DSE clusters.</td>
</tr>
<tr>
<td>Performance Service <em>(page 521)</em> Configuration</td>
<td>Allows users to configure the Performance Service.</td>
</tr>
<tr>
<td>Performance Service CQL Tracing</td>
<td>Allows users to trace slow CQL queries <em>(page 531)</em> when troubleshooting query issues.</td>
</tr>
<tr>
<td><strong>Node Operations</strong></td>
<td></td>
</tr>
<tr>
<td>Start and Stop</td>
<td>Allows users to start <em>(page 263)</em> and stop <em>(page 263)</em> DSE nodes. Start and stop nodes from the Other Actions menu options available in the List view <em>(page 228)</em>, or from the Actions menu in the Node Details view.</td>
</tr>
<tr>
<td>Cleanup</td>
<td>Allows users to run a cleanup <em>(page 251)</em> on one or more keyspaces.</td>
</tr>
<tr>
<td>Compact</td>
<td>Allows users to run compaction <em>(page 253)</em> on a keyspaces and their tables. Major compactions are not recommended unless there is a compelling reason to do so.</td>
</tr>
</tbody>
</table>
### Configuring OpsCenter

<table>
<thead>
<tr>
<th>Permission</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Drain</strong></td>
<td>Allows users to drain (<a href="#">page 257</a>) a node. The <strong>Drain</strong> option is available from the <strong>Actions</strong> menu in the Node Details dialog view, and also available when restarting DSE on a node (<a href="#">page 264</a>).</td>
</tr>
<tr>
<td><strong>Flush</strong></td>
<td>Allows users to flush (<a href="#">page 254</a>) a keyspace and its tables. Flushing a keyspace might affect system performance when there are many live, large memtables.</td>
</tr>
<tr>
<td><strong>Garbage Collection</strong></td>
<td>Allows users to perform garbage collection (<a href="#">page 252</a>) on nodes. Running GC causes a spike in latency.</td>
</tr>
<tr>
<td><strong>Repair</strong></td>
<td>Allows users to manually run an ad hoc repair (<a href="#">page 260</a>) operation on selected nodes in the List view.</td>
</tr>
<tr>
<td><strong>Data</strong></td>
<td></td>
</tr>
<tr>
<td><strong>View Schema</strong></td>
<td>Allows users to view the CQL statements for the schema in the Data workspace of OpsCenter Monitoring. Users must have the View Schema permission to view Tables, View UDT (<a href="#">page 267</a>), View UDF (<a href="#">page 268</a>), and View UDA (<a href="#">page 269</a>). Those users without view schema permission are shown a message explaining they must have the role permission for viewing anything in the Data workspace, and to contact their OpsCenter administrator to obtain access privileges.</td>
</tr>
<tr>
<td><strong>Modify Schema</strong></td>
<td>Allows users to edit keyspace settings, delete keyspaces, or delete tables in the Data workspace of OpsCenter.</td>
</tr>
<tr>
<td><strong>Truncate Data</strong></td>
<td>Allows users to truncate data (<a href="#">page 272</a>) from a table. The <strong>Truncate</strong> link appears as gray and unavailable for users who do not have this permission granted for their role.</td>
</tr>
<tr>
<td><strong>Cluster Topology</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Add Nodes</strong></td>
<td>Deprecated. Now users add nodes to an existing DSE cluster using Lifecycle Manager. Anyone assigned an admin role can use any feature of LCM.</td>
</tr>
<tr>
<td><strong>Rebalance Cluster (non-vnode)</strong></td>
<td>Allows users to rebalance a non-vnode cluster (<a href="#">page 281</a>). Not applicable to vnodes.</td>
</tr>
<tr>
<td><strong>Move</strong></td>
<td>Allows users to move a node, enter a new token, and assign the new token to the node. During a move node operation, the node is unavailable and cluster performance might be affected. Not applicable to vnodes. Access the <strong>Move</strong> option from the <strong>Other Actions</strong> menu available in the <strong>List view</strong> (<a href="#">page 228</a>), or from the <strong>Actions</strong> menu in the Node Details dialog view.</td>
</tr>
<tr>
<td><strong>Decommission</strong></td>
<td>Allows users to decommission a node (<a href="#">page 256</a>) from the <strong>Actions</strong> menu in the Node Details dialog view.</td>
</tr>
<tr>
<td>Permission</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>Remove Tokens</td>
<td>Allows removing tokens using the APIs.</td>
</tr>
</tbody>
</table>

### Enabling authentication in OpsCenter

`opscenterd.conf`  
The location of the `opscenterd.conf` file depends on the type of installation:
- **Package installations**: `/etc/opscenter/opscenterd.conf`
- **Tarball installations**: `install_location/conf/opscenterd.conf`

OpsCenter offers granular, role-based permission control for user and role management. By default, authentication is disabled. The first time authentication is enabled, a default admin account is created with username `admin` and password `admin`.

If you enable OpsCenter authentication, DataStax strongly recommends enabling SSL communication (page 82) between OpsCenter and the agents.

**Important:** Changing the default admin password is strongly recommended the first time you log in.

1. Edit the `opscenterd.conf` file and enable authentication.

   Set `enabled=True` in the `[authentication]` section.

   ```
   [authentication]
   enabled=True
   ```

2. Restart OpsCenter (page 75).

3. Open the OpsCenter UI in a browser.

   `http://localhost:8888`

4. Enter the default username of `admin` and the password `admin`.

### Logging in and out using OpsCenter authentication

If OpsCenter authentication (page 105) is enabled, follow these instructions to log in and out of OpsCenter and change the default admin password.

**Note:** After changing the hash algorithm (page 112) for the `password_hash_type`, instruct users to log in again so that OpsCenter can rehash and restore the user passwords. Because password hash algorithms are one-way functions that cannot be reversed, logging in again is necessary to update previously hashed user passwords.

1. Go to the main OpsCenter URL in a web browser.
Configuring OpsCenter

http://localhost:8888

2. A login dialog appears. Enter your username and password. The default admin username is admin and the default admin password is admin.

3. To change the default admin password:
   a. Click the admin username on the upper right and select Change Password.

   ![Change Password dialog](image)

   b. Enter the current password, enter the new password, confirm the new password, and click Save.

   c. The Password Updated dialog indicates the password has been updated. Click Close.

4. Log out by clicking your username in the top navigation bar and clicking Log Out.

Managing users and roles

Manage users and role permissions visually through the OpsCenter UI. Follow these instructions to manage users and roles in OpsCenter.

Note: When LDAP authentication is enabled, adding and editing users is disabled. Only role editing is available when LDAP is enabled.

OpsCenter supports all UTF-8 encoded characters for roles, usernames, and passwords.

1. Log in to OpsCenter as an admin. Click Settings#Users & Roles.
   The Users and Roles dialog appears.
2. Add a user:
   
   a. Click **Add User**.
      
      The **Add User** dialog appears.
      
      ![Add User dialog]
      
      b. Enter the username, password, and select a role for the user.
      
      c. Click **Save**.

3. Edit a user:
   
   a. Click the **Edit** icon for the user you want to edit.
   
   b. To change the user's password, enter and confirm the new password, and click **Save**.
      
   c. To change the user's role, select the new role from the **Roles** list and click **Save**.

4. Delete a user:
a. Click the **Delete** icon for the user you want to delete and click **Delete** to confirm.

5. Edit a role:

   a. Click **Manage Roles**.
      
      The **Manage Roles** dialog appears.

   b. To edit an existing role, click the **Edit** icon.
      
      The **Edit Role** dialog appears.
c. Select the cluster to apply role permissions to.

d. Select the options the user role has permissions for. To hide a cluster for users within a selected role, uncheck all permissions.

   **Tip:** Click **Select All** or **Unselect All** to quickly enable all or no permissions.

   e. Click **Save**.

   **Note:** All changes to roles and permissions are logged (page 137) for security auditing purposes.

   f. To apply role permissions for each cluster, repeat 5.a (page 108) through 5.e (page 109).

6. Add a role and assign it to users:
a. Click **Add Role**.

The **Add Role** dialog appears.

b. Enter the name of the role in **Role Name**, select the permissions from the appropriate feature check boxes, and click **Save**.

By default, new roles do not have any permissions.

**Tip:** Click **Select All** or **Unselect All** to quickly enable all or no permissions.

a. In the Users dialog, click the **Edit** icon for the user you want to add to the role.

b. In the **Role** list, select the role, and click **Save**.

c. Edit the role to apply its permissions to each cluster as appropriate.

7. Delete a role:
a. Select the role you want to delete in the Manage Roles dialog.

b. Click the Delete icon.

Changing the location of the password database

Change the default location of the password database passwd.db used for OpsCenter authentication if you prefer another location. The password database is created when authentication is enabled.

Change the location of the password database in the opscenterd.conf file.

passwd.db

The default location of the password database passwd.db for OpsCenter authentication depends on the type of installation:

- **Package installations**: /etc/opscenter/passwd.db
- **Tarball installations**: install_location/passwd.db

opscenterd.conf

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

- **Package installations**: /etc/opscenter/opscenterd.conf
- **Tarball installations**: install_location/conf/opscenterd.conf

1. Edit the opscenterd.conf file and change the location of the password database.

   Set passwd_db to the new location in the [authentication] section.

   ```
   [authentication]
   passwd_db = path to new password database
   ```

   **Warning:**

   - If you have already enabled authentication, copy the existing passwd.db file to the new location. If you do not copy the password database to the new location, OpsCenter will create a new password database in the specified location when it is started. Existing users and roles will be lost.
   
   - Your organization is responsible for backing up the passwd_db database. You must also configure failover (page 166) to mirror the passwd_db if your organization has failover enabled.
2. **Restart OpsCenter (page 75).**

### Configuring the user password hash algorithm

Configure the algorithm to hash user passwords for OpsCenter authentication. The default as of OpsCenter 6.0 is `bcrypt+blake2b-512`. Earlier versions of OpsCenter used `sha256`. OpsCenter versions prior to 6.0 are automatically migrated to the new default `bcrypt+blake2b-512` for increased password protection.

**Available password_hash_type options include:**

- `bcrypt+blake2b-512`
- `pbkdf2+blake2b-512`
- `pbkdf2+sha512`
- `pbkdf2+sha3-256`
- `bcrypt+sha512`

**opscenterd.conf**

The location of the `opscenterd.conf` file depends on the type of installation:

- **Package installations:** `/etc/opscenter/opscenterd.conf`
- **Tarball installations:** `install_location/conf/opscenterd.conf`

1. **Open the `opscenterd.conf` file for editing.**

   Set `password_hash_type` to the desired hashing option in the `[authentication]` section.

   ```ini
   [authentication]
   password_hash_type = pbkdf2+sha3-256
   ```

2. **Restart OpsCenter (page 75).**

3. **Instruct users to log in again (page 105)** so that OpsCenter can rehash and restore the user passwords. Because password hash algorithms are one-way functions that cannot be reversed, logging in again is necessary to update previously hashed user passwords.

### Encrypting sensitive configuration values

Activate configuration encryption for privacy and increased security for sensitive configuration values such as passwords. Sensitive configuration values within the OpsCenter UI are encrypted dynamically, then transmitted and written in an encrypted state to the relevant configuration files.
Manually editing configuration files requires manually encrypting the value and copying it to the appropriate location. Use the OpsCenter system key tool to manually encrypt (page 117) configuration values.

**Note:** Credentials used to access existing destinations for scheduled backups must be encrypted manually. For example, you must manually encrypt the `access_secret` field for any scheduled backups to Amazon S3.

System encryption key

The OpsCenter system key tool allows creating a key used for encryption on the `opscenterd` machine and all the nodes in a cluster. The system key tool resides in the `/bin` directory of `opscenterd`, such as `/usr/share/opscenter/bin`. Decrypting values is not supported.

<table>
<thead>
<tr>
<th>AES encryption modes (cipher algorithm)</th>
<th>Key strengths</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECB</td>
<td>128- or 256-bit</td>
</tr>
<tr>
<td>CBC</td>
<td>128- or 256-bit</td>
</tr>
<tr>
<td>CFB</td>
<td>128- or 256-bit</td>
</tr>
<tr>
<td>OFB</td>
<td>128- or 256-bit</td>
</tr>
</tbody>
</table>

**Note:** Using 256-bit key strength requires upgrading the JRE with enhanced security jar files. Download and install the Java Cryptography Extension (JCE), unzip the jar files, and place them under `$JAVA_HOME/jre/lib/security`. JCE-based products are restricted for export to certain countries by the U.S. Export Administration Regulations.

Encrypted fields

When configuration encryption is active in OpsCenter, any sensitive configuration values in the OpsCenter UI that are required to be encrypted are encrypted automatically by OpsCenter. The majority of sensitive configuration values can only be changed by directly editing the appropriate configuration file with the manually-encrypted configuration value.

**Cluster configuration fields**

The cluster configuration `cluster_name.conf` fields that require encryption include:

- `[jmx]:password`
- `[cassandra]:password, ssl_keystore_password, ssl_truststore_password`
- `[storage_cassandra]:password, ssl_keystore_password, ssl_truststore_password`
Configuring OpsCenter

- [agents]: ssl_keystore_password and ssl_truststore_password (monitored cluster), storage_ssl_keystore_password, storage_ssl_truststore_password (separate storage cluster)
- [agent_kerberos]: keytab, ticket_cache
- [ldap]: search_password
- [backup_service]: s3_proxy_host, s3_proxy_port

The following fields in `email.conf` require encryption:

- smtp_pass

This file is located in `install_location/event-plugins/email.conf`. Encryption for the smtp_pass field must be manually enabled.

DataStax Agent configuration fields

You are not required to configure the following DataStax Agent configuration fields in `address.yaml`. OpsCenter provides the values from `opscenterd.conf` to the DataStax Agents when it connects.

**Important:** If you set the DataStax Agent configuration fields values in `address.yaml`, and set `config_encryption_active` to true in `address.yaml` in addition to `opscenterd.conf`, you must supply the encrypted values for those fields.

The DataStax Agent configuration fields that require encryption include:

- access_secret
- storage_key
- jmx_pass
- cassandra_pass
- monitored_cassandra_pass
- ssl_keystore_password (storage cluster)
- ssl_truststore_password (storage cluster)
- monitored_ssl_keystore_password (monitored cluster)
- monitored_ssl_truststore_password (monitored cluster)

cluster_name.conf

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

- Package installations: `/etc/opscenter/clusters/cluster_name.conf`
- Tarball installations: `install_location/conf/clusters/cluster_name.conf`

opscenterd.conf

The location of the `opscenterd.conf` file depends on the type of installation:

- Package installations: `/etc/opscenter/opscenterd.conf`
- Tarball installations: `install_location/conf/opscenterd.conf`
address.yaml

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

- **Package installations:** /var/lib/datastax-agent/conf/address.yaml
- **Tarball installations:** `install_location/conf/address.yaml`

### Activating configuration encryption

Activate configuration encryption to automatically or manually encrypt sensitive configuration values. Reactivate encryption configuration if it was temporarily deactivated.

opscenterd.conf

The location of the `opscenterd.conf` file depends on the type of installation:

- **Package installations:** /etc/opscenter/opscenterd.conf
- **Tarball installations:** `install_location/conf/opscenterd.conf`

1. Open `opscenterd.conf` for editing.

2. In the `[security]` section, set the `config_encryption_active` option to True:

   ```
   [security]
   config_encryption_active = True
   ```

3. Restart opscenterd *(page 75)*.

### Creating a system key to encrypt sensitive configuration values

Follow these instructions to create the system key that enables automatically or manually encrypting sensitive configuration values such as passwords. Copy the system key to the agent for each node. Optionally, configure a custom name for the `opsc_system_key` or a path to the key when placed in a non-default location. When adjusting configuration files for an existing cluster, manually encrypt *(page 117)* the configuration values.

**Prerequisites:**

- If using 256-bit encryption key strength, upgrade the JRE with enhanced security jar files. Download and install the Java Cryptography Extension (JCE), unzip the jar files, and place them under `${JAVA_HOME}/jre/lib/security`.

address.yaml

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

- **Package installations:** /var/lib/datastax-agent/conf/address.yaml
- **Tarball installations:** `install_location/conf/address.yaml`
Configuring OpsCenter

**opscenterd.conf**

The location of the `opscenterd.conf` file depends on the type of installation:

- **Package installations:** `/etc/opscenter/opscenterd.conf`
- **Tarball installations:** `install_location/conf/opscenterd.conf`

1. In your `opscenterd` directory, run the system tool to create the key with the desired mode and key strength:
   
   - For package installations, run the system tool with `sudo`.
     
     ```sh
     $ bin/opscenter_system_key_tool sudo create ECB 128
     ```
   
   - For tarball installations, run the system tool without `sudo`.
     
     ```sh
     $ bin/opscenter_system_key_tool create ECB 128
     ```

   By default, the system key is named `opsc_system_key` and is located in the same directory as your `opscenterd.conf`.

   The name of the system key and the path to the encryption key are configurable in both `opscenterd.conf` and `address.yaml`. The options in `address.yaml` take precedence.

   ```yaml
   [security]
   config_encryption_active = True
   config_encryption_key_name = opsc_system_key
   config_encryption_key_path =
   ```

   The only fields in `address.yaml` you might need to enter are `config_encryption_key_name` and `config_encryption_key_path` if you are placing the keys in a location that is not standard on the agents.

   **[security] config_encryption_active**
   
   Specifies whether OpsCenter should attempt to decrypt sensitive config values.

   **[security] config_encryption_key_name**
   
   Name of the system key used to encrypt/decrypt stored passwords.

   **[security] config_encryption_key_path**
   
   Path to the encryption key.
   
   If left blank, the directory of `opscenterd.conf` will be used.

2. For any package installation, change the permissions for the new key to `opscenter`.
   
   ```sh
   $ sudo chown opscenter:opscenter /etc/opscenter/opsc_system_key
   ```

3. Manually copy the system key file to the agent for each node. The key file must reside in the same directory as the `address.yaml` for the agent.
   
   - For package installations, install the agent in `../etc/datastax-agent/`. 
Configuring OpsCenter

$ cp local/opsc_system_key ../etc/datastax-agent/

- For tarball installations, install the agent in ../agent/local.

$ cp local/opsc_system_key ../agent/local

4. Stop OpsCenter (page 75) and stop the agents (page 76).

5. Restart OpsCenter and the agents.

Manually encrypting a configuration value

Use the system key tool to manually encrypt sensitive configuration values. Manually editing configuration files requires manually encrypting the value and copying it to the appropriate location.

cluster_name.conf

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

- Package installations: /etc/opscenter/clusters/cluster_name.conf
- Tarball installations: install_location/conf/clusters/cluster_name.conf

1. Change to the directory for the OpsCenter daemon (opscenterd). For example, your opscenterd directory might be /usr/share/opscenter/bin in a package installation.

   $ cd path_to/opscenterd_directory

2. Run the system key tool with the value parameter.

   $ opscenter_system_key_tool value

3. When prompted, enter and confirm the value to encrypt.

   Enter value to encrypt:
   Confirm value to encrypt:

   The system key tool displays the encrypted value.

4. Copy and paste the encrypted value into the appropriate location in the configuration file. For an existing cluster, manually update the encryption-required fields in the cluster_name.conf file.

   Important: For a new cluster or node, do not paste the encrypted value into the password or other encryption-required fields of the OpsCenter interface.
OpsCenter automatically encrypts the sensitive fields such as passwords and writes the encrypted values to the configuration files.

5. Repeat the previous steps for each configuration value that requires encryption.

6. Restart OpsCenter (page 75).

**Deactivating configuration encryption**

Follow these steps to temporarily deactivate configuration encryption for any needed troubleshooting. After completing troubleshooting, activate configuration encryption (page 115) again.

**Important:** Configuration encryption is a recommended best practice for increased security.

**opscenterd.conf**

The location of the `opscenterd.conf` file depends on the type of installation:

- **Package installations:** `/etc/opscenter/opscenterd.conf`
- **Tarball installations:** `install_location/conf/opscenterd.conf`

1. Open `opscenterd.conf` for editing and deactivate configuration encryption:

   ```
   [security]
   config_encryption_active = False
   ```

2. Restart `opscenterd` (page 75).

**Authenticating OpsCenter users with LDAP**

The Lightweight Directory Access Protocol (LDAP) is a standard way of authenticating users across applications. OpsCenter supports LDAP authentication for external LDAP services.

When you enable LDAP authentication in OpsCenter, users that are managed by external LDAP servers can be authenticated by OpsCenter. DataStax Enterprise supports LDAP as well. See Defining an LDAP scheme.

LDAP must be configured for both DataStax Enterprise and OpsCenter if you plan to use LDAP with both applications. DataStax Enterprise and OpsCenter can share the same underlying user base accessed through LDAP or Active Directory, but those applications do not share their LDAP configuration settings. LDAP configuration settings for DataStax Enterprise and OpsCenter are nearly identical.

**Configuring LDAP**

Configure LDAP (Lightweight Directory Access Protocol) for users accessing OpsCenter.
Configuring OpsCenter

LDAP configuration is extremely flexible with many configuration options possible within OpsCenter. To peruse all of the available [ldap] configuration options, see OpsCenter configuration properties (page 182). This procedure provides a basic configuration example based on searching for a user in both user and group categories to authenticate a user.

Prerequisites:

There must be a properly configured LDAP v3 server running. The supported LDAP servers are:

- Microsoft Active Directory:
  - # Windows 2008
  - # Windows 2012
- OpenLDAP 2.4.x
- Oracle Directory Server Enterprise Edition 11.1.1.7.0

Additional requirements:

- If your organization started with standard OpsCenter authentication (page 105) and subsequently switched to implementing LDAP, delete the old passwd.db file.
- Roles: If using LDAP groups, create and mirror in OpsCenter the user role names (page 125) and permissions that are in LDAP. Role permissions are stored in OpsCenter, not LDAP. Users must have at least one role to be able to log in to OpsCenter when LDAP is enabled.

opscenterd.conf

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

- Package installations: /etc/opscenter/opscenterd.conf
- Tarball installations: install_location/conf/opscenterd.conf

passwd.db

The default location of the password database passwd.db for OpsCenter authentication depends on the type of installation:

- Package installations: /etc/opscenter/passwd.db
- Tarball installations: install_location/passwd.db

1. Open the opscenterd.conf file for editing.

2. Add an [authentication] section with the following options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>passwd_db</td>
<td>Contains the required OpsCenter user role information.</td>
</tr>
<tr>
<td>enabled</td>
<td>Set to True to enable LDAP authentication.</td>
</tr>
</tbody>
</table>
### Configuring OpsCenter

**authentication_method**

Set to LDAP, regardless if configuring Active Directory.

```yaml
[authentication]
passwd_db = ./passwd.db
enabled = True
authentication_method = LDAP
```

3. **Set the configuration for your LDAP server.** Add an `[ldap]` section to `opscenterd.conf` with the following LDAP server options as appropriate for your LDAP implementation:

<table>
<thead>
<tr>
<th><strong>server_host</strong></th>
<th>The host name of the LDAP server.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>server_port</strong></td>
<td>The port on which the LDAP server listens. For example, 389 or 636.</td>
</tr>
<tr>
<td></td>
<td>• <strong>389</strong> <em>(page 68)</em> is the default port for non-SSL LDAP and AD.</td>
</tr>
<tr>
<td></td>
<td>• <strong>636</strong>(page 68) is the default port for SSL LDAP and AD.</td>
</tr>
<tr>
<td></td>
<td>For more information about ports, see <em>OpsCenter ports</em> <em>(page 68).</em></td>
</tr>
<tr>
<td><strong>uri_scheme</strong></td>
<td>In LDAPv2 environments, TLS is normally started using the LDAP Secure URI scheme instead of the normal LDAP URI scheme. OpenLDAP command line tools allow either scheme to be used with the <code>-H</code> flag and with the URI <code>ldap.conf(5)</code> option. Defaults to <code>ldap</code> for <code>ldap_security = None</code>; defaults to <code>ldaps</code> for <code>ldap_security = SSL</code> or <code>TLS</code>.</td>
</tr>
<tr>
<td><strong>search_dn</strong></td>
<td>The username of the user that is used to search for other users on the LDAP server. When a user attempts to authenticate with LDAP, OpsCenter searches for the user in LDAP to discover whether the user exists and which roles the user is associated with. The only permission that the search user needs to have in the LDAP system is the ability to perform LDAP searches.</td>
</tr>
</tbody>
</table>
|                 | **Note:** If the `search_dn` and `search_password` *(that constitute the search user entry point for locating users in LDAP)* are omitted from the configuration, LDAP attempts to make an *anonymous bind* to perform the user search.
<table>
<thead>
<tr>
<th><strong>search_password</strong></th>
<th>The password of the search_dn user.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>user_search_base</strong></td>
<td>The search base for your domain, used to look up users. Set the ou and dc elements for your LDAP domain. For example, this can be set to ou=users,dc=domain,dc=top level domain. More specifically: ou=users,dc=example,dc=com. Active Directory uses a different user search base. For example: CN=search,CN=Users,DC=Active Directory domain name,DC=internal. More specifically: CN=search,CN=Users,DC=example-sales,DC=internal.</td>
</tr>
<tr>
<td><strong>user_search_filter</strong></td>
<td>The LDAP search filter used to uniquely identify a user. The default setting is (uid={0}), which looks for a user by unique user identifier. The value of the {0} variable is the username provided when logging in to OpsCenter. When using Active Directory, set the filter to (sAMAccountName={0}). Note: There is a known limitation in OpsCenter when using search filters for Active Directory. See troubleshooting LDAP (page 126).</td>
</tr>
<tr>
<td><strong>group_search_base</strong></td>
<td>The LDAP search base used to find a group. Example: ou=groups,dc=qaldap,dc=datastax,dc=lan</td>
</tr>
<tr>
<td><strong>group_search_filter</strong></td>
<td>Deprecated. The LDAP search filter used to find a user's group. Example: (member=cn={0},ou=users,dc=nodomain). Within the group_search_base, filter for members based on cn. For existing Active Directory implementations that have this configuration option already set, the group_search_filter_with_dn overwrites the returned value with the user's DN.</td>
</tr>
<tr>
<td><strong>group_search_filter_with_dn</strong></td>
<td>The LDAP search filter used to find a user's group. Uses the full user's DN from a user search. Overrides the deprecated group_search_filter. Example: (member={0}).</td>
</tr>
<tr>
<td><strong>group_name_attribute</strong></td>
<td>The LDAP field name used to identify a group's name. For example: cn.</td>
</tr>
</tbody>
</table>
### admin_group_name

The name of the admin group or a comma-separated list of admin group names; for example: admin, superusers. OpsCenter automatically creates the roles with admin permissions for the roles provided in the `admin_group_name` list. **Escape any restricted LDAP characters.** If your group name contains restricted LDAP characters such as ",", a comma, you must escape them. For example, two admin groups "foo, bar" and "baz" should be entered as: `foo \, bar, baz`

### user_memberof_attribute

Set to the attribute on the user entry containing group membership information. Set this option when using a `memberof_search` for the `group_search_type`.

OpsCenter allows for an alternate method of determining a user's role. When using `memberof_search`, rather than doing a directory search in LDAP for any roles that match the user, only the user is inspected. You can specify which attribute for a user is inspected. For example, you can define a user with a new attribute such as `opscenter_role` and populate it with the user's role in OpsCenter. Specify the value of the new attribute so that OpsCenter can inspect the user attribute.

### group_search_type

Defines how group membership is determined for a user. Available options:

- **directory_search:** *(default)* Performs a subtree search of `group_search_base` using `group_search_filter` to filter the results.
- **memberof_search:** gets groups from the `user_memberof_attribute` of a user. Using this option requires the directory server to have memberof support. When using the `memberof_search` rather than `directory_search` for group searches, you do not need to specify the `group_search_base` or `group_search_filter` options.
### user_memberof_stores_dn

Set to `True` if the `memberof` attribute's value is distinguished names of groups. This option must be set to `True` when configuring Active Directory, OpenLDAP, or when any other LDAP implementation returns a DN for the memberOf attribute value.

**Note:** If using an Oracle LDAP implementation, this option should be set to `True` if `user_memberof_attribute` is set to `isMemberOf`.

Default: `False`.

Set `user_memberof_stores_dn` to `False` if the attribute specified by `user_memberof_attribute` denotes 0 or more group names that correspond to the roles in OpsCenter. For example, if the `user_memberof_attribute` is set to `employeeType`, set the `user_memberof_stores_dn` option to `False` because the `employeeType` attribute value is not a distinguished name.

**Tip:** If the `user_memberof_attribute_stores_dn` is `False` and log in fails, and OpsCenter suspects the group name might be a DN, a warning is logged:

```
[opscenterd] WARN: It looks like you might be using Active Directory for authentication. You may need to set the 'user_memberof_attribute_stores_dn' config value to True and set the group_name_attribute config value appropriately in opscenterd.conf.
```

### ldap_security

The type of security to use with LDAP: None, TLS, or SSL. When set to TLS, uses TLS start. Setting this option to TLS or SSL sets the `uri_scheme` to LDAPS. Setting this option to None sets the `uri_scheme` to LDAP.

### truststore

Path to the truststore for SSL certificates.

### truststore_type

Type of the truststore. Default: JKS (Java Keystore).
## Configuring OpsCenter

<table>
<thead>
<tr>
<th><strong>truststore_pass</strong></th>
<th>The password to access the truststore.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>connection_timeout</strong></td>
<td>The number of seconds to wait before concluding that the LDAP server is down. Default: 20 seconds.</td>
</tr>
</tbody>
</table>

The following example configuration reflects a typical SSL LDAP (OpenLDAP or Oracle) implementation. The server_port value of 636 is for an SSL configuration.

If the search_dn and search_password options shown in lines 10 and 11 are omitted, LDAP attempts to make an anonymous bind to perform the user search.

This configuration example searches for a user in both user (user_search_base and user_search_filter) and group (group_search_base and group_search_filter) categories to authenticate a user. The group_search_type (line 19) is directory_search.

**Note:** The #user_search_base and #user_search_filter options are commented out in lines 14 and 15 because they are only applicable to Active Directory (AD) configuration.

```plaintext
01  [authentication]
02  passwd_db = ./passwd.db
03  enabled = True
04  authentication_method = LDAP
05
06  [ldap]
07  server_host = ldap.myCompany.lan
08  server_port = 636
09  uri_scheme = ldaps
10  search_dn = cn=admin,dc=devldap,dc=datastax,dc=lan
11  search_password = ****
12  user_search_base = ou=users,dc=devldap,dc=datastax,dc=lan
13  user_search_filter = (uid={})
14  #user_search_base = CN=search,CN=Users,DC=datastax,DC=internal
15  #user_search_filter = (sAMAccountName={}) # AD filter
16  group_search_base = ou=users,dc=devldap,dc=datastax,dc=lan
17  group_search_filter_with_dn = (member={})
18  group_name_attribute = cn
19  group_search_type = directory_search
20  admin_group_name = superusers,superusers2
21  ldap_security = SSL_TLS
22  truststore_type = JKS
23  truststore = ./truststore.jks
24  truststore_pass = secret
```

The following example reflects an Active Directory (AD) for Windows 2008 configuration. Unlike the previous LDAP example for OpenLDAP or Oracle, this AD
configuration makes use of `user_search_base` (line 12) and `user_search_filter` (line 13) for Active Directory configuration options. Also, the user search base for AD shown in line 12 differs in format from the LDAP example.

The `user_memberof_stores_dn` option in line 18 is explicitly set to `True` so that OpsCenter correctly handles the value of the `memberof` attribute shown in line 17 as a distinguished name (DN). The `user_memberof_stores_dn` option is also applicable to an OpenLDAP configuration.

```ini
01 [authentication]
02 passwd_db = ./passwd.db
03 enabled = True
04 authentication_method = LDAP
05
06 [ldap]
07 server_host = mywin2008.myCompany.lan
08 server_port = 636
09 uri_scheme = ldap
10 search_dn = CN=Administrator,CN=Users,DC=prodwin2008,DC=datastax,DC=lan
11 search_password = ****
12 user_search_base = CN=Users,DC=prodwin2008,DC=datastax,DC=lan # AD base
13 user_search_filter = (sAMAccountName={0}) # AD filter
14 admin_group_name = superusers
15 group_search_type = memberof_search
16 group_name_attribute = cn
17 user_memberof_attribute = memberof
18 user_memberof_stores_dn = True
19 ldap_security = SSL_TLS
20 truststore_type = JKS
21 truststore = /tmp/path_to_truststore_win2008
22 truststore_pass = secret
```

4. **Restart OpsCenter (page 75)** for the changes to take effect.

**Adding a role for an LDAP user**

When an LDAP user has been assigned LDAP groups, at least one of those groups must map to a role in OpsCenter. Otherwise, the user cannot log in to OpsCenter.

Add a parallel role in OpsCenter that mirrors the name of one of the LDAP groups assigned to a user. OpsCenter grants the matching role to the user.

If the list of a user's LDAP groups map to more than one role in OpsCenter, the user will be granted each of the listed roles, and their resulting OpsCenter permissions will be the merging of permissions for all of their OpsCenter roles.

The `group_search_type` (page 185) property indicates which method is used to determine LDAP group membership:
Configuring OpsCenter

- If using `directory_search`, the `group_search_filter_with_dn` must return a list of LDAP roles that matches at least one of the OpsCenter roles.
- If using `memberof_search`, the list of LDAP roles from the user’s `memberof` attribute must match at least one of the OpsCenter roles.

When LDAP is enabled, only role editing is supported in OpsCenter role-based security (page 101). Creating or editing users is disabled when LDAP is enabled because the users originate from LDAP and are managed therein. When creating or editing user roles, OpsCenter LDAP supports non-ASCII character sets for the role name. Because LDAP supports non-ASCII character sets for users, OpsCenter also supports non-ASCII character sets for users logging in to OpsCenter.

**Note:** Only an OpsCenter admin can add roles.

**Prerequisites:** Configure the admin role in the `opscenterd.conf` by setting the `admin_group_name` (page 185) configuration option. Then, log in to OpsCenter with a user mapped to that role so you can add any needed roles.

1. Click **Settings#Roles**.
   
   The Manage Roles dialog appears.

2. Click **Add Role**.

3. Select the cluster.

4. Enter a role name.

5. Select the appropriate permissions and click **Save**.

**Troubleshooting OpsCenter LDAP**

**Debugging LDAP using logback.xml**

To debug LDAP with deep visibility, change the level from `INFO` to `DEBUG` in the `<logger/>` line of `logback.xml`:

```
<logger name="org.apache.directory" level="INFO" additivity="false"/>
```

to

```
<logger name="org.apache.directory" level="DEBUG"/>
```

The `DEBUG` logging level gives you extreme visibility into the exact queries that Opscenter is sending to the LDAP server along with the responses. Revert the log level when you are done debugging.

**Common error messages**

**Error:** Failed to log in: {'desc': 'Protocol error'}
Error thrown when a `group_search_filter_with_dn` is not specified in `opscenterd.conf`.

**Error: Failed to log in: Invalid username or password.**
This error might occur even with a valid username and password. If so, OpsCenter might not have found a matching role for the user in OpsCenter. Either manually create the role [page 125] in OpsCenter for the user; or, if the user is an OpsCenter Admin user, set the `admin_group_name` in `opscenterd.conf` to the name of the role or group that is returned from the LDAP group query after the `group_search_filter` has been applied. The filter must filter out all but one of the groups that the user belongs to; otherwise, the User has more than one defined role error is the result.

**Error: Failed to log in: User myuser has no roles defined in LDAP**
This happens when the LDAP search returns zero roles for the authenticated user.

- If you are using a `group_search_type` of `directory_search`, modify your `group_search_filter_with_dn` so that one or more role names are returned from LDAP. Note that exactly one of those LDAP role names must match an OpsCenter role name for authentication to be successful.
- If you are using a `group_search_type` of `memberof_search`, ensure the `user_memberof_attribute` for the user contains a list of LDAP groups which names an OpsCenter role.

**Error: Failed to log in: User myuser has no matching OpsCenter role in LDAP group(s): ...**
This happens when a user belongs to one or more LDAP groups, but none of those groups match any roles defined in Op.ocenter. The list of User’s groups returned from LDAP must contain at least one of the roles configured in OpsCenter. See role prerequisites [page 118] and creating a role [page 125] for an LDAP user.

- If you are using a `group_search_type` of `directory_search`, modify the `group_search_filter_with_dn` so that the returned list of LDAP groups contains exactly one of the OpsCenter role names.
- If you are using a `group_search_type` of `memberof_search`, ensure the `user_memberof_attribute` for the user contains a list of LDAP groups that names an OpsCenter role.

If using Active Directory (AD), the user should belong to at least one non-special group that is not a built-in group. A special group is one of the built-in Active Directory groups such as Domain Users.

**Note:** All OpsCenter AD users must be part of the Domain Users group for the OpsCenter LDAP integration to function correctly. If Active Directory users are not a member of Domain Users, those users are not found in LDAP directory searches.

**Error: Failed to log in: Specified search user username unable to bind.**
Result when searching with an incorrect username for the `search_dn` or the incorrect password for the `search_password` criteria.

**Error: In order to perform this operation a successful bind must be completed on the connection.**
Configuring OpsCenter

The initial search is failing because the user doesn't have enough permissions for the way the query is structured. There are a few things to try:

- Set `user_search_filter` to the default value:
  ```
  user_search_filter = (sAMAccountName={0})
  ```

- Try changing `user_search_base` so that search doesn't start from top-level domain:
  ```
  user_search_base = OU=Users,OU=Corp,DC=[value],DC=[value],DC=com
  ```

- If you have a custom schema, change the `group_search_filter` and `group_search_filter_dn` to match it.

**opscenterd.conf**

The location of the `opscenterd.conf` file depends on the type of installation:

- Package installations: `/etc/opscenter/opscenterd.conf`
- Tarball installations: `install_location/conf/opscenterd.conf`

**logback.xml**

The location of the `logback.xml` file depends on the type of installation:

- Package installations: `/etc/opscenter/logback.xml`
- Tarball installations: `install_location/conf/logback.xml`

**Kerberos authentication with OpsCenter**

OpsCenter can use Kerberos to authenticate to DataStax Enterprise clusters. Understanding Kerberos principal formatting is crucial for successfully configuring OpsCenter to use Kerberos authentication.

**Important:** OpsCenter supports only one Kerberos configuration per cluster.

The Kerberos principal includes the host and IP address for the cluster. For example, the IP address 192.168.1.102 might be mapped to the principal `cassandra@EXAMPLE.COM`. This information is stored in a configuration file unique to the cluster. For example, `cluster_name.conf`.

Each monitored cluster can have an associated cluster for storing metrics and other data. However, because OpsCenter supports only one Kerberos configuration per cluster, a separate Kerberos configuration cannot be specified for the storage cluster. Therefore, a single set of credentials cannot be used to authenticate to both the monitored cluster and the storage cluster.

**Kerberos principal formatting**

A user in Kerberos is known as a `principal`, which is composed of three parts: primary, instance, and realm. Realm is similar to a domain, and each principal is fully qualified with the name of the realm. In the following examples, the realm is `EXAMPLE.COM`.
Configuring OpsCenter

The first part of the principal (primary) represents a specific identity within the realm, which is typically a user. For example, user123@EXAMPLE.COM represents a user named user123 that belongs to a realm named EXAMPLE.COM.

The instance is an optional component of the realm that users can specify to define a host where the service runs. For example, service456/server.example.com@EXAMPLE.COM indicates a principal for service456, which runs on the server.example.com host, in the EXAMPLE.COM realm.

**Configuring OpsCenter for Kerberos authentication**

When configuring OpsCenter for Kerberos authentication, create and configure the OpsCenter principals first, and then add the cluster to OpsCenter.

**Prerequisites:** Before configuring OpsCenter to use Kerberos authentication, configure DSE for Kerberos. The DSE documentation provides guidelines, samples, and a tutorial for configuring Kerberos with DSE.

1. Prepare DSE nodes for Kerberos.
2. Review the Kerberos guidelines.
3. Complete the Kerberos tutorial.

**dse.yaml**

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

<table>
<thead>
<tr>
<th>Type of Installation</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Package installations, Installer-Services installations</td>
<td>/etc/dse/dse.yaml</td>
</tr>
<tr>
<td>Tarball installations, Installer-No Services installations</td>
<td>installation_location/</td>
</tr>
<tr>
<td></td>
<td>resources/dse/conf/dse.yaml</td>
</tr>
</tbody>
</table>

**address.yaml**

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

- Package installations: /var/lib/datastax-agent/conf/address.yaml
- Tarball installations: install_location/conf/address.yaml

**cluster_name.conf**

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

- Package installations: /etc/opscenter/clusters/cluster_name.conf
- Tarball installations: install_location/conf/clusters/cluster_name.conf

**Kerberos principal formatting**

The method of creating Kerberos principals differs depending on the type of Kerberos Key Distribution Center (KDC) server used. The following procedures are based on MIT Kerberos.
Kerberos principals must be fully qualified with a realm, which in the following examples is EXAMPLE.COM. The realm follows the @ sign in the principal, and is typically all upper case. The first part of the principal represents a specific identity in the realm, such as a user or service.

Kerberos user principals typically have a single component, plus the realm. For example, user@EXAMPLE.COM. Kerberos service principals represent programs of different types, and follow the format service/HOST@EXAMPLE.COM. The HOST is the name of the server, such as node1.example.com.

**Important:** In the following procedure, the format of the service principal must match the format of the information in the keytab that was created with the kadmin command. For example, if the service principal is service/HOST@EXAMPLE.COM, then the principal in the keytab must use the same format.

The Kerberos commands in this procedure use the following format. For more details on using the kadmin CLI, see the kadmin help.

```
kadmin -p user_name/admin
addprinc -randkey service_name/FQDN
addprinc -randkey HTTP/FQDN
quit
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>kadmin</td>
<td>Launch Kerberos admin shell with an administrator account that has add privileges.</td>
</tr>
<tr>
<td>addprinc</td>
<td>Create a new service principal for the node.</td>
</tr>
<tr>
<td>-randkey</td>
<td>Set the key of the principal to a random value.</td>
</tr>
<tr>
<td>service_name</td>
<td>User-defined name of the service. For example, opscenterd.</td>
</tr>
<tr>
<td>FQDN</td>
<td>The fully qualified domain name of the node. For example, node1.example.com</td>
</tr>
</tbody>
</table>

1. Use the kinit command to obtain and cache an initial ticket-granting ticket for the admin user. Enter the password for the admin user when prompted.

    $ kinit cassandra/admin@dse
    Password for cassandra/admin@dse:

    where dse is the cluster to log in to.

2. Run the kadmin command to access the kadmin CLI. Enter the password for the admin user when prompted to access the kadmin: prompt.

    $ kadmin
    Password for cassandra/admin@dse:
3. Create a service principal for the OpsCenter daemon (opscenterd), plus an additional principal for HTTP communication. The service principal format for OpsCenter is `opscenterd/FQDN`, where `FQDN` is the Fully Qualified Domain Name of the OpsCenter node. The following example uses `opscenterd` as the principal name, but can be any valid name.

   **Note:** If configuring high availability (page 166) for OpsCenter, a principal must be created for the OpsCenter primary and backup instances.

   ```
   $ kadmin: addprinc -randkey opscenterd/node1.example.com
   $ kadmin: addprinc -randkey HTTP/node1.example.com
   ```

4. Create a shared principal for the OpsCenter DataStax agents, or create one principal per agent. The service principal format for the DataStax agent node is `dxagent/FQDN`, where `FQDN` is the Fully Qualified Domain Name of the DataStax agent node. The following example uses `dx-agent` as the principal name, but can be any valid name.

   **Important:** If creating one principal for each DataStax agent, manually update the `kerberos_client_principal` (page 136) property for each agent in `address.yaml`.

   ```
   $ kadmin: addprinc -randkey dx-agent/node2.example.com
   $ kadmin: addprinc -randkey HTTP/node2.example.com
   ```

5. Verify that the principals have been added by running the `listprincs` command with `kadmin`.

   ```
   $ kadmin: listprincs
   HTTP/node1.example.com@EXAMPLE.COM
   HTTP/node2.example.com@EXAMPLE.COM
   opscenterd/node1.example.com@EXAMPLE.COM
   dx-agent/node2.example.com@EXAMPLE.COM
   kadmin/admin@EXAMPLE.COM
   ```

   where `node*.example.com` is the FQDN and `EXAMPLE.COM` is the Kerberos realm, which must be all uppercase.

6. Create a Cassandra user for the OpsCenter daemon principal. The CQL role must be the same as the principal name created in 3 (page 131). In this example, the principal name and the CQL role name is `opscenterd`.

   ```
   ```
7. Create corresponding Cassandra users for all DataStax agent principals created in 4 (page 131). The CQL role name must be the same as the principal name. In this example, dx-agent is both the principal name and the CQL role name.

CREATE ROLE 'dx-agent/node1@EXAMPLE.COM' WITH LOGIN = true;

CREATE ROLE 'dx-agent/node2@EXAMPLE.COM' WITH LOGIN = true;

CREATE ROLE 'dx-agent/node3@EXAMPLE.COM' WITH LOGIN = true;

EXIT

To view the roles that were created on the node, run the `LIST ROLES` command in `cqlsh`.

LIST ROLES

8. Use the kadmin CLI to create two keytab files that map to the previously created principals: one for the OpsCenter node, and one for DataStax agents. If using one principal for each DataStax agent, create one keytab for each.

The keytab file is used to store the Kerberos principals created in the previous steps.

$ kadmin: ktadd -k /tmp/krb5_opsc.keytab opscenterd/node1.EXAMPLE.COM

$ kadmin: ktadd -k /tmp/krb5_agent.keytab dx-agent/node1.EXAMPLE.COM

$ kadmin: ktadd -k /tmp/krb5_agent.keytab dx-agent/node2.EXAMPLE.COM

$ kadmin: ktadd -k /tmp/krb5_agent.keytab dx-agent/node3.EXAMPLE.COM

$ kadmin: quit

To obtain principal information (outside of kadmin), use the `klist` command. In the following example, the command requests the ticket from the `krb5_opsc.keytab` file.

$ klist -kt krb5_opsc.keytab

Keytab name: FILE:krb5_opsc.keytab
KVNO Timestamp Principal
--- ------------------- ----------------------------
2 01/26/2018 18:16:18 opscenterd/node1.EXAMPLE.COM
9. On the OpsCenter node and on each node running a DataStax agent, create a directory to store the keytab files. The recommended directory on each node is `/etc/opscenter/security`.

```
$ mkdir /etc/opscenter/security
```

10. Copy the generated keytab files to location created in 9 (page 133). For example, `/etc/opscenter/security/krb5_opsc.keytab`.

```
$ scp /tmp/krb5_opsc.keytab opscenterd@node1.EXAMPLE.COM:/etc/opscenter/security

$ scp /tmp/krb5_agent.keytab dx-agent@node1.EXAMPLE.COM:/etc/opscenter/security

$ scp /tmp/krb5_agent.keytab dx-agent@node2.EXAMPLE.COM:/etc/opscenter/security

$ scp /tmp/krb5_agent.keytab dx-agent@node3.EXAMPLE.COM:/etc/opscenter/security
```

The keytab locations are set in the following properties in the `cluster_name.conf` file after adding the cluster to OpsCenter in 12 (page 133).

<table>
<thead>
<tr>
<th>Node</th>
<th>Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpsCenter</td>
<td>[kerberos] opscenterd_keytab_location (page 203)</td>
</tr>
<tr>
<td>DataStax agent</td>
<td>[kerberos] agent_keytab_location (page 203)</td>
</tr>
</tbody>
</table>

11. Change the owner of the keytabs and the keytabs directory for OpsCenter and the DataStax agent.

Replace the directory paths with the keytab locations created in the previous step.

```
$ sudo chown cassandra /etc/opscenter/security \
    /etc/opscenter/security/krb5_opsc.keytab

$ sudo chown cassandra /usr/agent/conf \
    /usr/agent/conf/krb5_agent.keytab
```

12. Add the cluster (page 274) to OpsCenter. The user adding the cluster to OpsCenter must have privileges on the DSE node to add the cluster.

*Note:* When adding the cluster, select **DSE security (kerberos) is enabled on my cluster** and input the required information.
a. Enter the **Service Name**. This name must match the user name of the Kerberos service used by DSE, which is defined in dse.yaml. For example, if the server principal is `dse/HOST@EXAMPLE.COM`, the Service Name should be `dse`.

```
kberos_options:
  ...
  service_principal: dse/HOST@EXAMPLE.COM
```

b. Enter the **Opscenterd Client Principal** created in 3 (page 131).
For example: `opscenterd/opscenterd.EXAMPLE.COM`

c. Enter the **Opscenterd Keytab Location** location created in 8 (page 132). This keytab contains credentials for the `opscenter_client_principal`.
For example: `/etc/opscenter/security/krb5_opsc.keytab`

d. Enter the **DataStax Agent Client Principal** created in 4 (page 131).
For example: `kerberos_client_principal: dx-agent/node1@EXAMPLE.COM`

*Important*: If using a different principal for each DataStax agent, the value entered in this field is a placeholder only. Because each DataStax agent has a different principal name, set the `kerberos_client_principal` (page 136) property in address.yaml for each DataStax agent.

e. Enter the **DataStax Agent Keytab Location** location created in 8 (page 132). This keytab contains credentials for the `agent_client_principal`.
For example: `/etc/opscenter/security/krb5_agent.keytab`
13. After specifying the Kerberos information, click **Next** and complete the remaining prompts to add the node to OpsCenter.

14. If different principals were created for each DataStax agent in 4 (page 131), restart each agent.

After the cluster is added to OpsCenter, the values entered for each field are saved in a configuration file specific to the cluster. The file is created as `/etc/opscenter/clusters/cluster_name.conf`.

**OpsCenter Kerberos configuration options**

This reference lists the available OpsCenter configuration options for Kerberos.

**Note:** The OpsCenter console is the most convenient way to configure basic OpsCenter connection settings (page 93) for authentication and encryption.

Cluster configuration for Kerberos

The following configuration options are available in `cluster_name.conf`:

- **[kerberos]** `default_service`  
  The default Kerberos service name (Example: cassandra).

- **[kerberos]** `default_hostname`  
  The default Kerberos hostname.

- **[kerberos]** `default_client_principal`  
  The default Kerberos client principal (Example: cassandra@realm).

- **[kerberos]** `default_client_user`  
  The default Kerberos client user.

- **[kerberos]** `opscenterd_client_principal`  
  The OpsCenter client principal in Kerberos (Example: user@realm).

- **[kerberos]** `opscenterd_keytab_location`  
  Full path to the keytab containing keys for `opscenterd_client_principal` on the OpsCenter machine.

- **[kerberos]** `agent_client_principal`  
  The DataStax agent client principal in Kerberos (Example: user@realm).

- **[kerberos]** `agent_keytab_location`  
  Full path to the keytab containing keys for `agent_client_principal` on the DataStax agent machine.

- **[kerberos]** `debug`  
  Whether to output debug messages during Kerberos connection attempts from OpsCenter.

Agent configuration for Kerberos

The following configuration options are available in `address.yaml`:

- **kerberos_service**  
  The Kerberos service name to use when using Kerberos authentication within DSE. Example: `kerberos_service: cassandra-kerberos`
**Configuring OpsCenter**

**kerberos_keytab_location**
The Kerberos keytab location when using Kerberos authentication within DSE.
Example: `kerberos_keytab_location: /path/to/keytab.keytab`

**kerberos_client_principal**
The Kerberos client principal to use when using Kerberos authentication within DSE. Example: `kerberos_client_principal: cassandra@hostname`

**address.yaml**
The location of the `address.yaml` file depends on the type of installation:
- Package installations: `/var/lib/datastax-agent/conf/address.yaml`
- Tarball installations: `install_location/conf/address.yaml`

**cluster_name.conf**
The location of the `cluster_name.conf` file depends on the type of installation:
- Package installations: `/etc/opscenter/clusters/cluster_name.conf`
- Tarball installations: `install_location/conf/clusters/cluster_name.conf`

**Related information:**
*Configuration files for OpsCenter* [Configure capabilities by manually modifying the `opscenterd.conf`, `cluster_name.conf`, and `address.yaml` configuration files.](#) (page 180)

**Troubleshooting Kerberos in OpsCenter**

Troubleshoot OpsCenter Kerberos connections with debug options.

**cluster_name.conf**
The location of the `cluster_name.conf` file depends on the type of installation:
- Package installations: `/etc/opscenter/clusters/cluster_name.conf`
- Tarball installations: `install_location/conf/clusters/cluster_name.conf`

1. Open the cluster-specific configuration file, `cluster_name.conf` for editing. Replace `cluster_name` with the name of your cluster.
2. Add the following to the `[kerberos]` section to output debug messages during Kerberos connections attempts from OpsCenter:

```
[kerberos]
debug = True
```

The debug option outputs the contents of the server section from the `jaas-krb5.conf` file, informing you of the settings in use that you can verify against your configuration settings.

3. Restart OpsCenter (page 75).
4. If deeper debugging is necessary, add `-Dsun.security.krb5.debug=true` to `$OPSC_JVM_OPTS`. The JVM parameter outputs verbose information about reasons why Kerberos connections attempts are failing, such as not authenticating due to key expiration, or no keys present in keytab, or cannot find keytab, for example.

For more information about JVM, see Configuring the OpsCenter JVM (page 174).

Configuring security logging

Configure comprehensive security logging in `logback.xml` that records user activity within OpsCenter, such as:

- Creating or deleting users and roles
- Adding or deleting users from roles
- Changing permissions for a role, including the specific details about all permission changes
- Resetting user passwords
- Logging in attempts both successful and unsuccessful

In addition, all login attempts that are anonymous or admin binds with regard to LDAP authentication are also logged. Therefore, comparisons between the OpsCenter authentication logging and LDAP logs should match exactly.

For more information, see OpsCenter logback.xml configuration (page 215).

logback.xml

The location of the `logback.xml` file depends on the type of installation:

- **Package installations**: `/etc/opscenter/logback.xml`
- **Tarball installations**: `install_location/conf/logback.xml`

1. Open `logback.xml` for editing.

2. Add the following section:

```xml
<appender name="security" class="ch.qos.logback.core.rolling.RollingFileAppender">
  <file>./log/security.log</file>
  <encoder>
    <pattern>%date{ISO8601, UTC} [%X{cluster_id}] %msg == %X{structured}== (%thread)\n%exception{20}</pattern>
  </encoder>
  <rollingPolicy class="ch.qos.logback.core.rolling.FixedWindowRollingPolicy">
    <fileNamePattern>./log/security.%i.log</fileNamePattern>
    <minIndex>1</minIndex>
    <maxIndex>10</maxIndex>
  </rollingPolicy>
</appender>
```
<triggeringPolicy
class="ch.qos.logback.core.rolling.SizeBasedTriggeringPolicy">
<maxFileSize>10MB</maxFileSize>
</triggeringPolicy>
<filter class="ch.qos.logback.classic.filter.ThresholdFilter">
<level>INFO</level>
</filter>
</appender>

3. To set structured data output, set $X{structured}$ as shown in the following example:

```
<pattern>%date{ISO8601, UTC} [%X{cluster_id}] %msg ==%X{structured}==\(%thread\)%n%exception{20}</pattern>
```

4. Restart OpsCenter (page 75).

The following example shows structured versus unstructured log output:

```
2016-04-07 21:42:16,664 [] Login failure user: foo (MainThread)

2016-04-07 21:42:16,664 [] Login failure user: foo =={"event":
"login-failure", "user": "foo", "opsc-initiated": false, "ip":
"0:0:0:0:0:0:0:1"}== (MainThread)
```

### Configuring alerts for events

Configure email alerts, post alerts to a URL (including a HipChat POST URL integration), or integrate with an enterprise reporting system by enabling SNMP trap alerts.

The OpsCenter Event Log page in the **Activities** section displays a continuously updated list of events and alerts. SNMP trap alerts display within the infrastructure monitoring solution for an organization.

**Log levels**

The following logging levels are available for Cassandra, DataStax Enterprise, and OpsCenter events from most (DEBUG) to least (ALERT) verbose, and from lowest (DEBUG) to highest (ALERT) severity:

- **DEBUG** (0)
- **INFO** (1)
- **WARN** (2)
- **ERROR** (3)
- **CRITICAL** (4)
- **ALERT** (5)
Alerts

Alerts are disabled by default and must be explicitly configured. Alerts can be sent remotely by email, through HTTP POST to a selected URL, or through SNMP (page 139) traps sent to an enterprise monitoring solution. All alerts contain information about each captured event. Optionally, you can configure OpsCenter to send alerts for selected levels of events or specific clusters. Alerts are triggered by events from the OpsCenter API or UI only; not by events triggered from the command line. For instance, a nodetool move operation submitted from the command line does not trigger an alert. However, a move operation launched using the Nodes#List View#Other Actions#Move option in the OpsCenter UI does trigger an alert.

Table 6: Alerts fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>api_source_ip</td>
<td>IP that originally sent the request through an API call</td>
<td>67.169.50.240</td>
</tr>
<tr>
<td>target_node</td>
<td>Destination of a STREAMING action.</td>
<td>10.1.1.11</td>
</tr>
<tr>
<td>event_source</td>
<td>Component that caused the event.</td>
<td>OpsCenter (i.e., restart, start)</td>
</tr>
<tr>
<td>user</td>
<td>OpsCenter user that caused the event.</td>
<td>opscenter_user</td>
</tr>
<tr>
<td>time</td>
<td>Normal timestamp for the event.</td>
<td>1311025650414527</td>
</tr>
<tr>
<td>action</td>
<td>Type of event.</td>
<td>20</td>
</tr>
<tr>
<td>subject</td>
<td>Customizable subject line of the email alert.</td>
<td>[WARN] OpsCenter Event - Node reported as being down: 127.0.0.1</td>
</tr>
<tr>
<td>message</td>
<td>Description of the event.</td>
<td>Garbage Collecting node 10.1.1.13</td>
</tr>
<tr>
<td>level</td>
<td>Numerical code for the log level.</td>
<td>1</td>
</tr>
<tr>
<td>source_node</td>
<td>Node where the event originated.</td>
<td>10.1.1.13</td>
</tr>
<tr>
<td>cluster</td>
<td>Name of the cluster where the event originated.</td>
<td>DSEProdCluster</td>
</tr>
<tr>
<td>level_str</td>
<td>Logging level of the event.</td>
<td>INFO</td>
</tr>
</tbody>
</table>

SNMP alerts overview

Send Simple Network Management Protocol (SNMP) traps to an enterprise monitoring system such as those provided by Nagios. SNMP is a standardized framework for infrastructure monitoring. Organizations use SNMP to monitor the status of their network infrastructure.

OpsCenter supports SNMP traps, not full SNMP agent, for integration with an agentless monitoring solution. The OpsCenter SNMP implementation supports sending traps over SNMPv1, SNMPv2c, and SNMPv3 protocols. The device being monitored is opscenterd, the OpsCenter daemon. Traps send an alert from a monitored device (opscenterd) to the SNMP
master (such as a Nagios application). OpsCenter supports SNMP as an event plugin, similar to the email and post URL alert plugins. SNMP requires import of an MIB.

### OpsCenter MIB fields

The OpsCenter MIB is SMIv2-compliant. The fields identified in the trap when using the OpsCenter MIB (Management Information Base) include those standard to all alerts (page 139), with the exception of an additional time field.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>opscAlertLevel</td>
<td>The ID of the severity level for an alert.</td>
</tr>
<tr>
<td>opscAlertLevelStr</td>
<td>The string of the severity level for an alert.</td>
</tr>
<tr>
<td>opscAlertMessage</td>
<td>The description of the event that triggered the alert.</td>
</tr>
<tr>
<td>opscAlertAction</td>
<td>The numerical type of event that triggered the alert.</td>
</tr>
<tr>
<td>opscAlertCluster</td>
<td>The name of the cluster on which the event was triggered.</td>
</tr>
<tr>
<td>opscAlertEventSrc</td>
<td>Alert Event Cluster. The name of the node on which the event was triggered.</td>
</tr>
<tr>
<td>opscAlertSrcNode</td>
<td>Alert Source Node. The IP of the node on which the event was triggered.</td>
</tr>
<tr>
<td>opscAlertTargNode</td>
<td>Alert target node. The destination of a STREAMING action.</td>
</tr>
<tr>
<td>opscAlertEpochTime</td>
<td>The time of event occurrence as a Unix Epoch timestamp.</td>
</tr>
<tr>
<td>opscAlertStrTime</td>
<td>Alert String Time. The date and time that the event occurred. Timestamp in a human-readable format.</td>
</tr>
<tr>
<td>opscAlertAPISrcIP</td>
<td>Alert API Source IP. The IP that originally sent the request.</td>
</tr>
<tr>
<td>opscAlertUser</td>
<td>The OpsCenter user that caused the event.</td>
</tr>
</tbody>
</table>

### Enabling SNMP alerts

Enable and configure SNMP alerts in the `config_location` (page 74)/event-plugins/snmp.conf file.

Multiple instances of the `snmp.conf` file results in multiple instances of an event plugin. The SNMP configuration file can have any name provided the file has the .conf file extension, contains the valid event plugin identifier [snmp], and resides in the event-plugins directory location. All configuration files are loaded, so differentiate the file names such as snmp1.conf, snmp2.conf, and so on. Multiple plugin instances facilitate flexibility when setting up alerts to SNMP targets for development and production clusters. All OpsCenter event plugins support multiple configurations files and subsequent plugin instances.

**Prerequisites:** Import the OpsCenter MIB (page 140) (DATASTAX-OPSCALERT-MIB.txt) located in the event-plugins directory into the infrastructure monitoring application of choice, such as a Nagios application. The MIB is a formal specification of...
the fields within OpsCenter traps. Installing the MIB is mandatory for using the SNMP traps effectively; without doing so, the field titles will not render properly. Refer to the installation and configuration instructions provided by the application vendor.

1. In a terminal, open `snmp.conf` for editing.

2. Enable the SNMP plugin for sending traps.

   ```
   [snmp]
   enabled=1
   ```

3. The engine ID that uniquely identifies the SNMP engine is commented out by default. By default, if the `engine_id` is unset, OpsCenter creates a local engine ID based on the local IP address. If you prefer to manually set the `engine_id`, uncomment the `engine_id`. Change the last four octets (01:02:03:04) of the `engine_id` string to a unique octet string for each OpsCenter installation.

   ```
   [snmp]
   ... 
   engine_id=80:00:00:00:05:50:60:70:80
   ```

4. For the `target_ip`, enter the IPv4 address of the SNMP receiver.

5. If you do not want to accept the remaining default configuration values, consult the following table for assistance with completing the fields relevant to each of your OpsCenter installations:

   **Table 8: SNMP configuration fields**

<table>
<thead>
<tr>
<th>SNMP configuration options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>enabled</td>
<td>Enables (1) or disables (0) the SNMP plugin instance. Default: 0 (disabled).</td>
</tr>
<tr>
<td>levels</td>
<td>Comma-delimited list of event severities that the SNMP plugin should fire for. If not specified, all levels are listened for by default. Available options:</td>
</tr>
<tr>
<td></td>
<td>• DEBUG</td>
</tr>
<tr>
<td></td>
<td>• INFO</td>
</tr>
<tr>
<td></td>
<td>• WARN</td>
</tr>
<tr>
<td></td>
<td>• ERROR</td>
</tr>
<tr>
<td></td>
<td>• CRITICAL</td>
</tr>
<tr>
<td></td>
<td>• ALERT</td>
</tr>
<tr>
<td>clusters</td>
<td>Comma-delimited list of cluster names that the plugin should fire for. If no cluster is specified, the alert is called for events on all clusters by default.</td>
</tr>
<tr>
<td><strong>SNMP configuration options</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>------------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>engine_id</td>
<td>SNMP engine ID that uniquely identifies the SNMP engine. The last four octets of the ID (01:02:03:04 by default) must be unique between multiple OpsCenter installations. Default: 80:00:00:00:05:01:02:03:04 for the Octets default format scheme. SNMP version is not relevant to this setting. For more details on SNMP management, see RFC3411.</td>
</tr>
<tr>
<td>Format Schemes</td>
<td>The fifth octet in the engine_id determines the format scheme that specifies the nature of the remaining trailing octets. Available format schemes:</td>
</tr>
<tr>
<td></td>
<td>• 01: IPv4 Address scheme</td>
</tr>
<tr>
<td></td>
<td>• 02: IPv6 Address scheme</td>
</tr>
<tr>
<td></td>
<td>• 03: MAC Address scheme</td>
</tr>
<tr>
<td></td>
<td>• 04: Text Address scheme</td>
</tr>
<tr>
<td></td>
<td>• 05: Octets scheme (default)</td>
</tr>
<tr>
<td>target_ip</td>
<td>IPv4 address of the SNMP trap target. Default: 127.0.0.1</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Only IPv4 is supported at this time.</td>
</tr>
<tr>
<td>target_port</td>
<td>Listening port on the SNMP manager to receive SNMP traps. Default: 162.</td>
</tr>
<tr>
<td>use_snmpv3</td>
<td>When set to 1, this SNMP plugin instance will use the SNMPv3 configuration options (username, auth_protocol, auth_key, privacy_protocol, privacy_key) with SNMP authentication. When set to 0, this SNMP plugin instance will use the community_name with SNMPv1/2c authentication. Default: 0 (uses SNMPv1/2c).</td>
</tr>
<tr>
<td>community_name</td>
<td>SNMPv1/2c-specific configuration. Community name for authentication with the SNMP trap target. Default: public.</td>
</tr>
<tr>
<td>user</td>
<td>Username for SNMPv3 authentication. Default: opscusername.</td>
</tr>
<tr>
<td>auth_protocol</td>
<td>SNMPv3 authentication hashing algorithm. Specifies the authentication protocol to use with SNMPv3. Available options:</td>
</tr>
<tr>
<td></td>
<td>• SHA (default)</td>
</tr>
<tr>
<td></td>
<td>• MD5</td>
</tr>
<tr>
<td></td>
<td>• NoAuth</td>
</tr>
<tr>
<td>auth_key</td>
<td>SNMPv3 authentication key. Authentication key as configured for your OpsCenter installation in your monitoring software. Default: auth1key.</td>
</tr>
</tbody>
</table>
### Configuring OpsCenter

#### SNMP configuration options

<table>
<thead>
<tr>
<th><strong>privacy_protocol</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>SNMPv3 privacy protocol. Available standards-compliant options:</td>
<td></td>
</tr>
<tr>
<td>· AES (default)</td>
<td></td>
</tr>
<tr>
<td>· DES</td>
<td></td>
</tr>
<tr>
<td>· NoPriv</td>
<td></td>
</tr>
<tr>
<td>Available Extended Security Options (ESO) from the SNMP Research Group:</td>
<td></td>
</tr>
<tr>
<td>· 3DES</td>
<td></td>
</tr>
<tr>
<td>· AES192</td>
<td></td>
</tr>
<tr>
<td>· AES256</td>
<td></td>
</tr>
</tbody>
</table>

| **privacy_key** | Privacy key as configured for your OpsCenter install in your monitoring software. Default: privkey1. |

### Enabling SMTP email alerts

OpsCenter can send alerts to multiple email addresses. To enable email alerts, edit the config_location (page 74)/event-plugins/email.conf file and provide valid SMTP server host and port information.

Multiple instances of the email.conf file results in multiple instances of an event plugin. The email configuration file can have any name provided the file has the .conf file extension, contains the valid event plugin identifier [email], and resides in the event-plugins directory location. All configuration files are loaded, so differentiate the file names such as email1.conf, email2.conf, and so on. Multiple plugin instances facilitate flexibility when setting up alerts to different email recipients for development and production clusters. All OpsCenter event plugins support multiple configurations files and subsequent plugin instances.

**Prerequisites:** Make sure that you have valid SMTP mail accounts to send and receive alerts.

1. On the OpsCenter daemon host, open the email.conf file for editing.

2. Set enabled to 1.

3. Provide valid values for your SMTP host, port (25, or 465 if using SSL), user, and password.

4. For secure communications, enable Secure Sockets Layer (SSL) or Transport Layer Security (TLS) protocol on your system. Typically, SSL is required.

5. Provide valid values for the to_addr and from_addr email addresses. The to_addr value is the account that will receive alerts.
Configuring OpsCenter

To send alerts to multiple email addresses, enter the email addresses as a comma-delimited list.

6. Set the specific levels of alerts to send. The default is to listen for all levels.

7. Specify the clusters for which the email alert config will run. The default is to call the alert for events on all clusters.

8. Customize the template for the email subject and message body as desired. Available options are described in the config file comments.

9. Save the configuration file and **restart (page 75)** the OpsCenter daemon.

For a configuration with email alerts to multiple recipients enabled for warning and error levels to all clusters, with SSL enabled, the `email.conf` looks like:

```plaintext
[email]
# set to 1 to enable email
enabled=1

# levels can be comma-delimited list of any of the following:
# DEBUG, INFO, WARN, ERROR, CRITICAL, ALERT
# If left empty, will listen for all levels
levels=WARN,ERROR

# clusters is a comma-delimited list of cluster names for which
# this alert config will be eligible to run.
# If left empty, this alert will be called for events on all
# clusters
clusters=
smtplib_host=smtplib.gmail.com
smtp_port=465
smtp_user=mercury@gmail.com
smtp_pass=*********
smtplib_use_ssl=1
smtp_use_tls=0
smtp_retries=1
smtp_timeout=5
to_addr=dse_admin@acme.com, devOps@acme.com, opsc_admin@acme.com
from_addr=mercury@gmail.com

# Customizable templates for subject and body. The key specified in
# {}'s must map to the items provided in json map at the end of
# the emails. For example, some available keys are:
# node, cluster, datetime, level_str, message, target_node,
# event_source, success, api_source_ip, user, source_node
# more advanced formatting options explained here: https://
docs.python.org/2/library/string.html#formatspec
```
A custom email template example:

```
subject=\{level_str\} OpsCenter Event on \{cluster\} - \{message\}

template=Message: \{message\}
    Time:    \{datetime\}
    Level:   \{level_str\}
    Cluster: \{cluster\}
    Node:    \{node\}
    User:    \{user\}
```

Enabling alerts posted to a URL

Configure OpsCenter to send alert data to a specified URL that has a page capable of receiving and processing POST data. For example, a simple PHP script containing `print_r($_POST);` will echo the received POST request. An example POST request with its payload:

```
POST / HTTP/1.0
Host: localhost
User-Agent: Twisted PageGetter
Content-Length: 184
Content-type: application/x-www-form-urlencoded
connection: close
```
Configuring OpsCenter

The request body contains fields described in Alerts (page 139) along with the cluster name where applicable. Both JSON and form URL-encoded content types are supported. To specify the field format and which fields are sent, reference the configuration file for examples of the key-value pair formatting.

Multiple instances of the config_location (page 74) posturl.conf file results in multiple instances of the event plugin. The configuration file can have any name provided the file has the .conf file extension, contains the valid event plugin identifier [posturl], and resides in the event-plugins directory location. All configuration files are loaded, so differentiate the file names such as posturl1.conf, posturl2.conf, and so on. Multiple plugin instances facilitate flexibility when setting up alerts for development and production clusters. All OpsCenter event plugins support multiple configurations files and subsequent plugin instances.

Tip: OpsCenter can post alerts via http or https. To enable https, update the URL from http:// to https://.

opscenterd.conf

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

- Package installations: /etc/opscenter/opscenterd.conf
- Tarball installations: install_location/conf/opscenterd.conf

Prerequisites:

- Make sure your web server and posting script are configured to receive alerts.
- If your machine is connected behind a proxy, open opscenterd.conf for editing and add the following under [http_proxy_settings]:

```plaintext
http_proxy_url = http://your_proxy:port
http_proxy_username = your_proxy_username
http_proxy_password = your_proxy_password
```

1. On the OpsCenter daemon host, open posturl.conf for editing.
2. Set enabled=1.
3. Specify the levels and clusters for which the alert configuration will run if you do not want alerts called for all levels and clusters.
4. Adjust the connection timeout (connection_timeout) if the connection to the posturl is taking too long.
5. For url, provide a valid path to the posting script.
6. Set a username and password for HTTP Basic authentication.

7. Set the type of posted data for `post_type`. Supported options are `json` or `form` (default). Customize the message using the available key values described in the config file comments.

8. Save `posturl.conf` and restart (page 75) the OpsCenter daemon.

9. Verify (page 148) events are posting correctly.

In a system with posting form data enabled for critical and alert-level events for all clusters, `posturl.conf` looks like:

```plaintext
[posturl]
enabled=1

# levels can be a comma-delimited list of any of the following:
# DEBUG,INFO,WARN,ERROR,CRITICAL,ALERT
# If left empty, will listen for all levels
levels=CRTICAL,ALERT

# clusters is a comma-delimited list of cluster names for which
# this alert config will be eligible to run.
# If left empty, this alert will be called for events on all
# clusters
clusters=

# the URL to send a HTTP POST to
url=http://host/path/to/script

# Set a username for basic HTTP authorization
#username=foo

# Set a password for basic HTTP authorization
#password=bar

# Set the type of posted data. Available options are 'json' or 'form'
post_type=form

# Fields specified here will override the default event data fields.
#
# They must be formatted as key-value pair, with key and value
# separated by
# an equals (=). Each pair after the first must be on its own line,
# indented beyond the first line
```
# You may use tokens found within the default event data for or in values. For example, some available keys are:
# cluster, time, level_str, message, target_node, event_source,
# success, api_source_ip, user, source_node
# Keys must be encapsulated in {brackets}.
#fields=textKey=value
# mixedKey=cluster-{cluster}
# event-msg={message}

Example adding a static custom field to posturl.conf using the fields prefix. Specify all of the fields to include:

default fields message={message}
message_type=CRITICAL

Example form data displaying the message and cluster name:

FORM DATA: array (
   'event-msg' => 'OpsCenter starting up.,'
   'mixedKey' => 'cluster-C1',
   'textKey' => 'value',
)

Example JSON data displaying the message and cluster name:

JSON DATA: {"event-msg": "OpsCenter starting up." , "mixedKey": "cluster:C1", "textKey": "value"}
3. Deploy the script. You might need to restart the web server.

4. Launch a logged event, such as an OpsCenter restart or garbage compaction from Dashboard→Cluster→List View.

Output to /tmp looks something like this:

```php
Array
( [api_source_ip ] => 67.169.50.240
  [target_node ] => None
  [event_source ] => OpsCenter
  [user ] => None
  [ time ] => 1311025598851602
  [action ] => 20
  [message ] => Garbage Collecting node 50.1.1.24
    [level ] => 1
    [source_node ] => 50.1.1.24
    [level_str ] => INFO
)
```

## Posting URL alerts to a Slack channel

Post URL alerts to one or more Slack channels.

Multiple instances of the config_location (page 74) `slack_posturl.conf` file results in multiple instances of the event plugin. The Slack configuration file can have any name provided the file has the .conf file extension, contains the valid event plugin identifier [posturl], and resides in the event-plugins directory location. All configuration files are loaded, so differentiate the file names such as `slack_posturl1.conf`, `slack_posturl2.conf`, and so on. Multiple plugin instances facilitate flexibility when setting up alerts to different Slack channels for development and production clusters. All OpsCenter event plugins support multiple configurations files and subsequent plugin instances.

**Tip:** OpsCenter can post alerts via http or https. To enable https, update the URL from http:// to https://.

**Prerequisites:** Set up an incoming webhook integration using slack. For instructions, refer to the slack API.


2. Copy and paste the contents from the example below into the file.

   **Example** `slack_posturl.conf`:
Configuring OpsCenter

[posturl]
enabled=1

# levels can be comma-delimited list of any of the following:
# DEBUG, INFO, WARN, ERROR, CRITICAL, ALERT
# If left empty, will listen for all levels
levels=

# clusters is a comma-delimited list of cluster names for which
# this alert config will be eligible to run.
# If left empty, this alert will be called for events on all
clusters
clusters=

# the URL to send an HTTP POST to
url=https://hooks.slack.com/services/your/webhook/url

# Set a username for basic HTTP authorization
# username=foo

# Set a password for basic HTTP authorization
# password=bar

# The number of seconds before a connection to url or proxy_url
# will be aborted due to a timeout
connection_timeout=5

# Set the type of posted data. Available options are 'json' or
# 'form'
post_type=json

# Fields specified here will override the default event data fields.
# They must be formatted as key-value pair, with key and value
# separated by
# an equals (=). Each pair after the first must be on its own line,
# indented beyond the first line
#
# You may use tokens found within the default event data for or in
# values. For example, some available keys are:
#     cluster, time, level_str, message, target_node, event_source,
#     success, api_source_ip, user, source_node
# Keys must be encapsulated in {brackets}.
# fields=textKey=value
#     mixedKey=cluster-{cluster}
#     event-msg={message}
fields=text={level_str} from {cluster}: {message}

# Customizable templates for http post data. Some available keys
# are:
#     node, cluster, datetime, level_str, message, target_node,
#     event_source, success, api_source_ip, user, source_node
3. Ensure that the URL in the example config file `url=https://hooks.slack.com/services/your/webhook/url` is replaced with your custom-defined incoming webhook URL.

4. Save `slack_posturl.conf` and restart (page 75) the OpsCenter daemon.
   Verify alerts are getting populated in the slack channel that you defined for your webhook.

Configuring data collection and expiration

OpsCenter creates its own keyspace within a cluster for storing collected metrics. This data can also be stored on a cluster other than the one currently being managed by OpsCenter. Metrics data is collected at regular intervals and stored within a cluster in a keyspace named `OpsCenter`. The tables containing metric data continue to grow. Configure how long to keep historical metrics. Data expires after configurable time periods.

Controlling data collection

To control consumption of disk space, OpsCenter limits the growth of OpsCenter performance data by:

- Excluding specified keyspaces and tables from performance data collection
- Shortening the time period after which performance data automatically expires

Excluding keyspaces and tables from data collection

By default, OpsCenter does not collect performance data for its own keyspace or the Cassandra system keyspace. You can manually add any other keyspaces or tables that you do not want to monitor in the `[cassandra_metrics]` section of the `cluster_name.conf` configuration file. For example, to prevent data collection for the keyspace test as well as the table `Keyspace1.Standard1`, uncomment and edit the following values in the OpsCenter cluster configuration file (`cluster_name.conf`):
Configuring OpsCenter

```plaintext
[cassandra_metrics]
ignored_keyspaces = system, OpsCenter, test
ignored_column_families = Keyspace1.Standard1
```

Tables (formerly column families) are specified in the format:

```
<keyspace_name>.<column_family_name>.
```

**Note:** Configuration properties still refer to tables as column families in OpsCenter.

**cluster_name.conf**

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

- Package installations: `/etc/opscenter/clusters/cluster_name.conf`
- Tarball installations: `install_location/conf/clusters/cluster_name.conf`

**Related information:**

**Metrics Collection Properties** [Use these metrics properties to limit the keyspaces and tables (column families) for metrics collection and change default expiration periods (ttl) for performance data.](page 152)

**Metrics Collection Properties**

- **[cassandra_metrics] ignored_keyspaces**
  - A list of keyspaces to **not** collect metrics for, separated by commas. The default value is `system, system_traces, system_auth, dse_auth, and OpsCenter`.  
- **[cassandra_metrics] ignored_column_families**
  - A list of tables to **not** collect metrics for, separated by commas. Each entry should be of the form "ks.cf". For example: `metrics_ignored_column_families = system.NodeInfo, system.Schema, Keyspace1.Standard1`
- **[cassandra_metrics] ignored_solr_cores**
  - A list of solr cores to **not** collect metrics for, separated by commas. Each entry should be of the form "ks.cf". For example: `metrics_ignored_solr_cores = Keyspace1.Standard1, solr.wiki`.

These properties set the expiration time for data stored in the OpsCenter keyspace. Each time period for rolling up data points into summary views has a separate expiration threshold, or time-to-live (ttl) value expressed in seconds. By default, shorter time periods have lower values that result in more efficient expiration and compaction of the relatively larger volumes of data. Uncomment these properties to change the default expiration periods for performance data. Properties and default values are:

- **[cassandra_metrics] 1min_ttl**
  - Sets the time in seconds to expire 1 minute data points. The default value is 604800 (7 days).
- **[cassandra_metrics] 5min_ttl**
  - Sets the time in seconds to expire 5 minute data points. The default value is 2419200 (28 days).
- **[cassandra_metrics] 2hr_ttl**
Sets the time in seconds to expire 2 hour data points. The default value is 31536000 (365 days).

[cassandra_metrics] 24hr_ttl
Sets the time to expire 24 hour data points. The default value is 0, or never.

Related information:
Excluding keyspaces and tables from data collection [By default, OpsCenter does not collect performance data for its own keyspace or the Cassandra system keyspace. You can manually add any other keyspaces or tables that you do not want to monitor in the [cassandra_metrics] section of the cluster_name.conf configuration file.] (page 151)

Changing performance data expiration times

cluster_name.conf

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

- Package installations: /etc/opscenter/clusters/cluster_name.conf
- Tarball installations: install_location/conf/clusters/cluster_name.conf

Performance data stored in OpsCenter expires after configurable time periods. The default values are designed to provide efficient compaction and eventual deletion of the data, with faster expiration times for the more granular, larger-volume data roll-ups.

- One-minute roll-ups (1min_ttl) expire after one week, or 604800 seconds.
- Five-minute roll-ups (5min_ttl) expire after four weeks, or 2419200 seconds.
- Two-hour roll-ups (2hr_ttl) expire after one year, or 31536000 seconds.

To change expiration time period:

In this example, the one-minute and five-minute roll-ups are set to expire twice as fast as the defaults, two-hour roll-ups are set to be kept indefinitely (expiration is disabled), and the 24 hour roll-ups are not going to be stored.

1. Edit the cluster_name.conf file.

2. Add the following time-to-live (ttl) values in seconds under a [cassandra_metrics] section:

   [cassandra_metrics]

   1min_ttl = 302400
   5min_ttl = 1209600
   2hr_ttl = 0
   24hr_ttl = -1

3. Restart OpsCenter (page 75).
Data collected after restarting OpsCenter expires according to the new settings. The data collected before restarting OpsCenter expires according to the setting in effect when it was collected.

**Storing collection data on a separate cluster**

**cluster_name.conf**

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

- **Package installations:** `/etc/opscenter/clusters/cluster_name.conf`
- **Tarball installations:** `install_location/conf/clusters/cluster_name.conf`

**address.yaml**

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

- **Package installations:** `/var/lib/datastax-agent/conf/address.yaml`
- **Tarball installations:** `install_location/conf/address.yaml`

Configure a separate storage cluster for metrics. Store collection data on a separate DataStax Enterprise cluster as an alternative to the default of OpsCenter storing data in an OpsCenter keyspace on the same DataStax Enterprise cluster being monitored.

OpsCenter metrics can consume excessive disk space. Configuring a separate storage cluster for metrics and controlling data collection (page 151) can prevent issues with performance or collecting metrics.

**Important:** In production environments, DataStax strongly recommends storing data in a separate DataStax Enterprise cluster.

There are two different cluster connections in the agent. When monitored and storage clusters are separate, the agent monitors the health of the monitored cluster on one of those connections, and writes metrics to the storage cluster on the other connection. If there is not a separate storage cluster, both connection pools point to the same DataStax Enterprise cluster instance. OpsCenter supports connecting to a DataStax Enterprise storage cluster when TLS/SSL is enabled. See Configuring SSL/TLS for DSE using LCM (page 604).

**Note:** In versions earlier than OpsCenter 5.2.2, the authentication mechanism and credentials must be identical between the storage cluster and the monitored clusters. As of OpsCenter 5.2.2 and later, OpsCenter supports distinct SSL or authentication connection settings between a monitored cluster and its corresponding storage cluster.

**Prerequisites:**

- The seed nodes must be accessible without Kerberos security.
- A unique keyspace must be used for each DataStax Enterprise cluster monitored by OpsCenter. If you are storing data for multiple clusters, DataStax recommends adding the cluster name as a suffix to the default keyspace name of OpsCenter. For example,
set the keyspace name to `OpsCenter_Cluster1` for the storage cluster to differentiate it from the OpsCenter keyspace for the monitored cluster.

1. Open the storage cluster configuration file `cluster_name.conf` for editing. The `cluster_name.conf` represents the named configuration file for a particular cluster. Replace `cluster_name` with your actual cluster name.

2. Add a `[storage_cassandra]` section with the applicable storage configuration options for your environment.

The following example configuration assumes similar authentication between the monitored and storage cluster:

```
[storage_cassandra]
username = cassandra
password = password
seed_hosts = host1, host2
api_port = 9160
cql_port = 9042
keyspace = OpsCenter_Cluster1
```

**Tip:** You can set most of the agent configuration options in the `[agent_config]` section of the `cluster_name.conf` file. The options in the `[agent_config]` section must match the corresponding configuration option name in `address.yaml`. Setting agent options through the cluster configuration file sets the corresponding property in `address.yaml` on every node. Some properties or some cases might require setting these properties directly in `address.yaml` on applicable agents. Setting agent configuration options that require a list entry (value1,value2,value3 and so forth) is not supported in `cluster_name.conf`.

Available cluster configuration options for monitored and storage clusters:

**[storage_cassandra] seed_hosts**
Configure when using a different cluster for OpsCenter storage. A Cassandra seed node is used to determine the ring topology and obtain gossip information about the nodes in the cluster. This should be the same comma-delimited list of seed nodes as the one configured for your DataStax Enterprise cluster by the seeds property in the `cassandra.yaml` configuration file.

**[storage_cassandra] api_port**
Configure when using a different cluster for OpsCenter storage. The Thrift remote procedure call port configured for your cluster. Same as the `rpc_port` property in the `cassandra.yaml` configuration file. Default is 9160.

**[storage_cassandra] cql_port**
Configure when using a different cluster for OpsCenter storage. The CQL port configured for your cluster, the default port is 9042.

**[storage_cassandra] local_dc_pref**
To reliably determine cluster information, OpsCenter can require a minimum of 2 nodes to connect to. If you specified a single value for `seed_hosts`, OpsCenter selects a second node in the cluster to fulfill this requirement. You
can specify local_dc_pref as a datacenter name to constrain OpsCenter to use that datacenter to pick the second node.

**[storage_cassandra] used_hosts_per_remote_dc**
Configure when using a different cluster for OpsCenter storage. If using local_dc_pref, this option specified how many remote dc connections may be used as a fallback, the default value is 1.

**[storage_cassandra] connect_timeout**
Configure when using a different cluster for OpsCenter storage. Sets the timeout, in seconds, of a thrift connection from OpsCenter to Cassandra. The default value is 6.0.

**[storage_cassandra] bind_interface**
Configure when using a different cluster for OpsCenter storage. The interface used for thrift connections.

**[storage_cassandra] connection_pool_size**
Configure when using a different cluster for OpsCenter storage. The number of connections to thrift to build for the connection pool. The default value is 5.

**[storage_cassandra] username**
Configure when using a different cluster for OpsCenter storage. The username used to connect to Cassandra if authentication is enabled.

**[storage_cassandra] password**
Configure when using a different cluster for OpsCenter storage. The password used to connect to Cassandra if authentication is enabled.

**[storage_cassandra] send_rpc**
Configure when using a different cluster for OpsCenter storage. Specifies whether to send the Cassandra RPC IP to DataStax Agents. The default value is True.

**[storage_cassandra] keyspace**
The name of the keyspace used for OpsCenter data. The keyspace name must be unique for each managed cluster.

**[storage_cassandra] ssl_keystore**
The SSL keystore location for OpsCenter to use to connect to Cassandra directly.

**[storage_cassandra] ssl_keystore_password**
The SSL keystore password for OpsCenter to use to connect to Cassandra directly.

**[storage_cassandra] ssl_truststore**
The SSL truststore location for OpsCenter to use to connect to Cassandra directly.

**[storage_cassandra] ssl_truststore_password**
The SSL truststore password for OpsCenter to use to connect to Cassandra directly.

3. If your storage and monitored clusters require different authentication or ssl settings, open cluster_name.conf for editing and add the settings in an **[agents]**, **[storage_cassandra]**, and **[cassandra] (monitored)** sections as applicable for your environment:
ssl_keystore = /etc/dse/keystore
ssl_keystore_password = example
storage_ssl_keystore = /etc/dse/keystore_storage
storage_ssl_keystore_password = example2

[storage_cassandra]
username = cassandra
password = password
seed_hosts = host1, host2
api_port = 9160
cql_port = 9042
keyspace = OpsCenter_Cluster1

[cassandra] #monitored
username = baz
password = test

Note: If only the cassandra_user settings were configured for a storage cluster prior to 5.2.2, with no differentiation in authentication, opscenterd sets both the cassandra_user for storage and the monitored_cassandra_user settings on your behalf when upgrading to 5.2.2.

Available cluster configuration options for monitored and storage clusters:

[agents] ssl_keystore
The SSL keystore location for DataStax Agents to use to connect to CQL on the monitored cluster.

[agents] ssl_keystore_password
The SSL keystore password for DataStax Agents to use to connect to CQL on the monitored cluster.

[agents] storage_ssl_keystore
The SSL keystore location for DataStax Agents to use to connect to CQL on the storage cluster.

[agents] storage_ssl_keystore_password
The SSL keystore password for DataStax Agents to use to connect to CQL on the storage cluster.

[cassandra] seed_hosts
A Cassandra seed node is used to determine the ring topology and obtain gossip information about the nodes in the cluster. This should be the same comma-delimited list of seed nodes as the one configured for your DataStax Enterprise cluster by the seeds property in the cassandra.yaml configuration file. The default value is localhost.

[cassandra] rolling_restart_retry_delay
The number of seconds to wait between retry attempts when connecting to Cassandra after restarting a node. Default: 5.

[cassandra] rolling_restart_retry_attempts
The maximum number of connection retry attempts after restarting a Cassandra node. Default: 25.

[cassandra] rolling_restart_error_threshold
A rolling restart will be cancelled if the number of errors during the restart reaches this number. This helps prevent having too many nodes down in your cluster if something catastrophic happens during a rolling restart. Default: 1

[cassandra] restart_delay
During a rolling restart, the time in seconds OpsCenter waits after sending the command to stop Cassandra before sending the command to start it again. The default is 30 seconds.

**Note:** For a complete list of cluster_name.conf options, see Cassandra connection properties (page 196). For a complete list of address.yaml options, see DataStax Agent configuration (page 205).

4. If your environment requires setting options in address.yaml for reasons such as integration with third-party CMS applications, set the storage configuration options (sans the monitored_prefix) and corresponding monitored_configuration options as appropriate for your environment. If you set these agent options in address.yaml, you must do so for every node, whereas setting the options in cluster_name.conf automatically propagates the settings to all agents.

Available configuration options for storage and monitored clusters in address.yaml:

**cassandra_port**
Port used to connect to the storage cassandra node. The native transport port. Example: cassandra_port: 9042

**thrift_port**
Port used to connect to storage thrift server. The default setting is 9160. This information will be sent by opscenterd for convenience, but can be configured locally as needed. Example: thrift_port: 9160

**cassandra_user**
The Username used to connect to storage cassandra when authentication is enabled. Example: cassandra_user: cassandra

**cassandra_pass**
The password used to connect to storage cassandra when authentication is enabled. Example: cassandra_pass: cassandra [This field may be encrypted for additional security.]

**ssl_keystore**
The SSL keystore location for the storage cluster that agents use to connect to CQL. Example: ssl_keystore: /etc/dse/conf/.keystore

**ssl_keystore_password**
The SSL keystore password for the storage cluster that agents use to connect to CQL. Example: ssl_keystore_password: keystore-pass [This field may be encrypted for additional security.]

**ssl_truststore**
The SSL truststore location for the storage cluster that agents use to connect to CQL. Example: ssl_truststore: /etc/dse/conf/.truststore

**ssl_truststore_password**
The SSL truststore password for the storage cluster that agents use to connect to CQL. Example: `ssl_truststore_password: truststore-pass` [This field may be encrypted for additional security.]

**monitored_cassandra_port**

Port used to connect to the monitored cassandra node. The native transport port. Example: `monitored_cassandra_port: 9042`

**monitored_thrift_port**

Port used to connect to monitored thrift server. The default setting is 9160. This information will be sent by opscenterd for convenience, but can be configured locally as needed. Example: `monitored_thrift_port: 9160`

**monitored_cassandra_user**

The Username used to connect to monitored cassandra when authentication is enabled. Example: `monitored_cassandra_user: cassandra`

**monitored_cassandra_pass**

The password used to connect to monitored cassandra when authentication is enabled. Example: `monitored_cassandra_pass: cassandra-pass` [This field may be encrypted for additional security.]

**monitored_ssl_keystore**

The SSL keystore location for the monitored cluster that agents use to connect to CQL. Example: `monitored_ssl_keystore: /etc/dse/conf/.keystore`

**monitored_ssl_keystore_password**

The SSL keystore password for the monitored cluster that agents use to connect to CQL. Example: `monitored_ssl_keystore_password: keystore-pass` [This field may be encrypted for additional security.]

**monitored_ssl_truststore**

The SSL truststore location for the monitored cluster that agents use to connect to CQL. Example: `monitored_ssl_truststore: /etc/dse/conf/.truststore`

**monitored_ssl_truststore_password**

The SSL truststore password for the monitored cluster that agents use to connect to CQL. Example: `monitored_ssl_truststore_password: truststore-pass` [This field may be encrypted for additional security.]

---

5. Restart OpsCenter (page 75) for the changes to take effect.

6. After the storage cluster is online, decide whether to keep the metrics from the OpsCenter keyspace on the original cluster.

### Removing duplicate storage cluster with LCM

When you configure a separate storage cluster for metrics, the OpsCenter keyspace is recreated on the storage cluster. Because the OpsCenter UUID changes, you will see duplicate clusters in LCM. To remove the duplicate cluster in Lifecycle Manager (LCM), resync the OpsCenter cluster UUID.

**Prerequisites:**

- Configure a separate OpsCenter storage cluster for metrics.
Configuring OpsCenter

• Create a unique keyspace to use for each DSE cluster monitored by OpsCenter. If you are storing data for multiple clusters, DataStax recommends adding the cluster name as a suffix to the default keyspace name of OpsCenter. For example, set the keyspace name to OpsCenter_Cluster1Storage for the storage cluster to differentiate it from the OpsCenter keyspace for the monitored cluster.

1. Get the unique_cluster_id for the OpsCenter storage cluster.

```bash
$ curl http://127.0.0.1:8888/cluster_name/cluster/unique_cluster_id

// 20190604153731
// http://127.0.0.1:8888/cluster_name/cluster/unique_cluster_id

"unique_cluster_id"
```

2. Get a list of LCM clusters to get the unique opsc-cluster-id and href_URL for the LCM cluster.

```bash
$ curl http://127.0.0.1:8888/api/v2/lcm/clusters/

// 20190604153842
// http://127.0.0.1:8888/api/v2/lcm/clusters

{
   "next": null,
   "previous": null,
   "last": 1,
   "count": 1,
   "per-page": 50,
   "current": 1,
   "results": [
      {
         "opsc-cluster-id": "opsc_cluster_id",
         "name": "cluster_name",
         "id": "id",
         "href": "href_URL",
         ...
      }
   ]
}
```

**Note:** The opsc_cluster_id is not expected to match the unique_cluster_id from the curl result.

3. If opsc-cluster-id and unique_cluster_id do not match, replace the opsc-cluster-id value with the unique_cluster_id using the LCM cluster href_URL.

```bash
$ curl -X PUT href_URL -H "Content-Type: application/json" -d 
'{"opsc-cluster-id": "unique_cluster_id"}'
```
4. Refresh the LCM web-interface.

   The duplicate cluster should no longer be visible.

Example: Removing duplicate from db_monitoring cluster

Get the unique_cluster_id from db_monitoring cluster.

```bash
$ curl http://127.0.0.1:8888/db_monitoring/cluster/unique_cluster_id
```

Based on the results, the unique_cluster_id is 6885d453-a816-4c9e-b06f-7050d28c50b5.

```bash
// 20190604153731
// http://127.0.0.1:8888/db_monitoring/cluster/unique_cluster_id
"6885d453-a816-4c9e-b06f-7050d28c50b5"
```

Get the unique opsc-cluster-id and href_URL for the db_monitoring LCM cluster.

```bash
$ curl http://127.0.0.1:8888/api/v2/lcm/clusters/
```

Based on the results, the opsc_cluster_id is f1da1420-7846-4c84-bf7d-399d48e6a51e and the href_URL is http://127.0.0.1:8888/api/v2/lcm/clusters/cd658793-e763-4e5a-9960-b19fe935f0db.

```bash
// 20190604153842
// http://127.0.0.1:8888/api/v2/lcm/clusters
{
    "next": null,
    "previous": null,
    "last": 1,
    "count": 1,
    "per-page": 50,
    "current": 1,
    "results": [
        {
            "opsc-cluster-id": "f1da1420-7846-4c84-bf7d-399d48e6a51e",
            "name": "db_monitoring",
            "id": "cd658793-e763-4e5a-9960-b19fe935f0db",
            "href": "http://127.0.0.1:8888/api/v2/lcm/clusters/cd658793-e763-4e5a-9960-b19fe935f0db",
```

Note: If the OpsCenter and LCM cluster UUIDs match and you have duplicate clusters, contact DataStax Support.
Replace the `opsc-cluster-id` value with the `unique_cluster_id`.

```
$ curl -X PUT http://127.0.0.1:8888/api/v2/lcm/clusters/cd658793-e763-4e5a-9960-b19fe935f0db -H "Content-Type: application/json" -d '{"opsc-cluster-id": "6885d453-a816-4c9e-b06f-7050d28c50b5"}'
```

The results show that the `opsc-cluster-id` matches the `unique_cluster_id`.

```
{
  "opsc-cluster-id": "6885d453-a816-4c9e-b06f-7050d28c50b5",
  "name": "db_monitoring",
  "id": "cd658793-e763-4e5a-9960-b19fe935f0db",
  "href": "http://127.0.0.1:8888/api/v2/lcm/clusters/cd658793-e763-4e5a-9960-b19fe935f0db",
  ...
}
```

---

**Sending OpsCenter metrics to a Graphite server**

**address.yaml**

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

- **Package installations**: `/var/lib/datastax-agent/conf/address.yaml`
- **Tarball installations**: `install_location/conf/address.yaml`

**cluster_name.conf**

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

- **Package installations**: `/etc/opscenter/clusters/cluster_name.conf`
- **Tarball installations**: `install_location/conf/clusters/cluster_name.conf`

Configure forwarding metrics from OpsCenter to a Graphite server. This is an OpsCenter Labs feature (that is, under on-going development but available for use). OpsCenter pushes metrics at a hard-coded interval of 60 seconds.

Graphite stores time-series data and renders static graphs of the data. If your organization uses Graphite in its reporting infrastructure, you can easily integrate the data collecting power of OpsCenter with the open source Graphite monitoring tool.

OpsCenter versions 6.1.2 and later provide the ability to bypass storing metrics in the rollup tables of the DSE cluster. Set the `bypass_dse_metrics_storage` option to `True` to bypass redundant storage of metrics when the Graphite labs feature is enabled in OpsCenter.
Warning: If the DSE metrics storage has been disabled but the Graphite reporter is not configured for receiving the metrics, a warning is logged that user should enable an option for metrics storage.

Prerequisites: Install and configure Graphite.

1. Open cluster_name.conf for editing. Substitute cluster_name with the name of your cluster. Setting agent options through the cluster configuration file sets the corresponding property in address.yaml on every node.
   If necessitated by your environment, open address.yaml for editing and configure at the node level. Do so for every node that requires a specific configuration override.

2. Add the following configuration options:
   
   **[labs]**
   
   **graphite_host**
   Setting graphite_host enables the forwarding of metrics to a graphite server at the given address. Leaving the graphite_host blank disables forwarding metrics to the graphite server.
   
   **[labs]**
   
   **graphite_port**
   Port for graphite's plaintext protocol.
   
   **[labs]**
   
   **graphite_prefix**
   A prefix to insert metrics under.
   
   **[labs]**
   
   **bypass_dse_metrics_storage**
   Enable or disable storing metrics in a monitored or a separate storage DSE cluster. Metrics are stored in a DSE monitored or storage cluster by default. Default: False.
   
   ```
   [labs]
   graphite_host = 127.0.0.1
   graphite_port = 2003
   graphite_prefix = opscenter
   bypass_dse_metrics_storage = True
   ```

3. Save the configuration file or files.

4. Restart (page 75) the OpsCenter daemon.

5. If you made changes to address.yaml (page 205), restart (page 76) the DataStax agents.

**OpsCenter definition files for supporting updated DSE versions**

OpsCenter ships with a set of definition files that can be updated independently of OpsCenter itself. OpsCenter uses definition files to enable support for newer versions of DataStax Enterprise without the need to upgrade the currently installed version of OpsCenter.
By default, OpsCenter automatically downloads updated definitions when they become available. Definition files are updated independently of OpsCenter by automatically downloading new definitions at regular intervals from a central server (or by manually downloading definitions on an ad hoc basis). The opscenterd process checks a central server located at opscenter.datastax.com, and pulls down updates as needed to the set of definition files specific to the installed version of OpsCenter.

Definitions directories locations on the file system depend on the installation type and are configurable:

```
[definitions] definitions_dir
```

The file system location where definition files are stored. The default location is `/var/lib/opscenter/definitions` for package installations and `install_location/definitions` for tarball installations.

Basic steps in the process of automatically updating definitions are logged in opscenterd.log at INFO, and full details are logged at DEBUG.

You can modify the default interval (page 165) or disable (page 165) the automatic downloads altogether.

**opscenterd.log**

The location of the `opscenterd.log` file depends on the type of installation:

- **Package installations**: `/var/log/opscenter/opscenterd.log`
- **Tarball installations**: `install_location/log/opscenterd.log`

## Updating and configuring definitions files properties

The definitions properties are for configuring the OpsCenter updater, which automatically downloads and updates the definition files that enable support for different releases of DataStax Enterprise. If an update is found, a remote file named `definition_files.tgz` is downloaded and applied to the appropriate definition directory.

The OpsCenter machine must have an internet connection to automatically update the definitions files, or to do so manually with an API call (page 165); otherwise, you must manually download (page 166) and update the definitions file.

**Note:** After manually downloading or changing properties in the definitions file, restart OpsCenter (page 75) for the changes to take effect.

### Definition files configuration properties

Configure the definition file properties in the `opscenterd.conf` file. Enable or disable the automatic update. Set the frequency with which to check for definition file updates. Override the default download filename, directory, or port.

```
[definitions] use_ssl
```
Specifies whether SSL should be used to get definition file updates. This option requires OpenSSL on the OpsCenter host. The default value is True.

**[definitions]** definitions_dir
The file system location where definition files are stored. The default location is `/var/lib/opscenter/definitions` for package installations and `install_location/definitions` for tarball installations.

**[definitions]** auto_update
Specifies whether OpsCenter should automatically attempt to periodically update the definition files. The default value is True.

**[definitions]** download_host
The host that definition file update requests will be made to. The default host is opscenetr.datastax.com.

**[definitions]** download_port
The port used to request definition file updates on `download_host`. The default port is 443.

**[definitions]** download_filename
The name of the tar file on the `download_host` that contains definition file updates. The default name is `definitions_files.tgz`.

**[definitions]** hash_filename
The name of the hash file on the `download_host` used to determine if the definition file requires an update. The default file is `version.md5`.

**[definitions]** sleep
The duration in seconds between checks to update the definition files. The default time is 3600 seconds (1 hour).

Modifying the update interval

The definition files are automatically updated every hour by default. Modify the default interval by setting the `sleep` option in the `[definitions]` section of `opscenterd.conf`. The `sleep` interval should be specified in seconds. For example, set the update interval to 7200 seconds (every 2 hours):

```
[definitions]
sleep = 7200
```

Disabling automatic updates of definitions files

Disable the automatic update process by setting `auto_update` to `False` in the `[definitions]` section of `opscenterd.conf`:

```
[definitions]
auto_update = False
```

Manually updating definition files with an API call

If you prefer to update manually using the API, make an API call directly to OpsCenter to check and download any updates. The call can be made regardless of whether automatic updates are enabled.
Configuring OpsCenter

```bash
curl -X POST http://my.opscenter.host:8888/rc/definition/update
```

The call returns true if an update was applied; and returns false if no update was found.

For more information about OpsCenter API calls, including updating definitions for LCM, see [getting definitions](#).

### Manually updating definition files for offline OpsCenter instances

If the OpsCenter machine does not have an internet connection, you can manually update the definitions.

1. Download the definitions tarball:

   ```
   https://opscenter.datastax.com/definitions/6.1.0/definition_files.tgz
   
   **Note:** The URL depends on the currently installed version of OpsCenter. Replace `6.1.0` as shown in the above URL with your currently installed version of OpsCenter.
   ```

2. If applicable, remove any existing files in the definitions directory (`definitions_dir`):

   ```bash
   [definitions]
   definitions_dir
   The file system location where definition files are stored. The default location is `/var/lib/opscenter/definitions` for package installations and `install_location/definitions` for tarball installations.
   ```

3. Unpack the definitions files tarball into the definitions directory.

4. Restart `opscenterd` ([page 75](#)).

**opscenterd.conf**

The location of the `opscenterd.conf` file depends on the type of installation:

- **Package installations**: `/etc/opscenter/opscenterd.conf`
- **Tarball installations**: `install_location/conf/opscenterd.conf`

### Automatic failover overview

Automatic failover provides continuous high availability of OpsCenter for managing mission-critical data on DataStax Enterprise clusters without manual intervention or downtime.

Currently, OpsCenter allows one backup instance to a primary instance in an active-passive configuration. The **OpsCenter Failover Enabled Best Practice Rule ([page 502](#))** recommends enabling failover. When no backup is configured, the rule fails and sends an alert. After enabling failover ([page 169](#)), the best practice rule passes the next time it runs if it detected a correctly configured backup OpsCenter. If the newly configured backup
OpsCenter detects any DataStax Community or open source Cassandra clusters, it logs an entry and shuts itself down.

**Note:** If a non-DataStax Enterprise cluster is added after enabling automatic failover, OpsCenter fires an alert that automatic failover will not work and the backup OpsCenter instance shuts down.

### Failover behavior

The primary and backup OpsCenter instances send and listen for heartbeat messages on stomp channels to communicate status with each other. The primary OpsCenter sends a heartbeat message regardless of whether a backup OpsCenter is configured. The primary OpsCenter listens for messages from the heartbeat reply stomp channel to determine if a backup is configured. The `primary_opscenter_location` configuration file you create on the backup OpsCenter instance contains the IP address of the primary OpsCenter instance that the backup OpsCenter monitors. The configured backup OpsCenter listens for heartbeat messages from the primary OpsCenter to determine whether the primary OpsCenter is up. If the backup OpsCenter detects no heartbeat from the primary OpsCenter during the configured window (60 seconds by default), the backup OpsCenter initiates the failover process and automatically assumes the responsibilities of the primary OpsCenter. The backup OpsCenter automatically reconfigures the agents by automatically changing `stomp_interface` (page 206) in `address.yaml` to connect to the backup instance instead of the failing primary instance.

**Warning:** Ensure that `address.yaml` is not being managed by third-party Configuration Management. During failover, OpsCenter automatically changes `stomp_interface` in `address.yaml` to point to the backup opscenterd instance. If a separate Configuration Management system is managing `address.yaml`, that change might be undone when the Configuration Management system pushes its next update.

### Failover recovery

After a failover, the former backup OpsCenter that took over as primary remains the primary OpsCenter. At that point, configure another backup OpsCenter by recreating the `primary_opscenter_location` (page 173) file that points the new backup instance to the IP address of the primary instance to monitor. If you are configuring the former primary OpsCenter as the new backup instance, ensure the server is healthy again before restarting the server.

**Note:** If a failover occurred due to a network split, the formerly primary OpsCenter must be manually shut down, and another backup configured when network connectivity has been restored. Upon startup, each OpsCenter instance generates a unique id (uid), which is stored in the `failover_id` file. In the event of a network split, a `failover_id` uniquely identifies each OpsCenter to agents and prevents both OpsCenter machines from running operations post-failover, which could corrupt data. The location of `failover_id` file depends on the type of install and is configurable (page 173).
Failover aftereffects

After an automatic failover, minimal manual intervention if any is required for recovery, depending on the root cause of the failover and what processes were in progress at that time. Generally, the effects of failing over are similar to restarting OpsCenter, with a few notable exceptions:

- **Alerts** - Trigger as normal. An exception is an alert firing and unfiring within the failover window; in which case the alert is never triggered.
- **Authentication** - Logs out existing user sessions. User sessions do not persist. Users must log in again.
- **Backup** - Skips a scheduled backup if it falls within the failover window. Backup does not occur until the next scheduled time.
- **Restore** - Continues the restore operation if failover occurred mid-restore; however, the result of the restore cannot be communicated because the backup OpsCenter was unaware the restore transpired.
- **Repair Service** - Resumes from the last saved state. Be sure to mirror the repair service directory (page 170). An OpsCenter instance failure does not affect repairs currently running on any nodes. New repairs do not continue until an automatic failover successfully completes or the OpsCenter instance that failed is brought up again.
- **Provisioning** - Provisioning jobs that were in progress when the primary Lifecycle Manager (page 549) fails attempt to complete on the primary, and may fail. Lifecycle Manager does not attempt to automatically resume jobs on the backup OpsCenter, but manually Running the job (page 572) again allows the job to proceed to completion.

Troubleshooting failover

In most cases, the backup OpsCenter instance selects the correct IP address for reconfiguring agents after a failover as described in failover behavior (page 167). If for some reason the incorrect IP address is not automatically being selected to update all agents, explicitly set the report_interface (page 194) property in opscenterd.conf on the backup OpsCenter instance.

**Note:** This workaround assumes the snitch is not Ec2MultiRegionSnitch.

Failover when upgrading OpsCenter

When failover is configured, there is a recommended process to follow when upgrading OpsCenter. For more information, see upgrading OpsCenter when failover is enabled.

address.yaml

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

- **Package installations:** /var/lib/datastax-agent/conf/address.yaml
- **Tarball installations:** install_location/conf/address.yaml

opscenterd.conf

The location of the opscenterd.conf file depends on the type of installation:
• Package installations: /etc/opscenter/opscenterd.conf
• Tarball installations: install_location/conf/opscenterd.conf

### Enabling automatic failover

**opscenterd.conf**

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

- Package installations: /etc/opscenter/opscenterd.conf
- Tarball installations: install_location/conf/opscenterd.conf

**address.yaml**

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

- Package installations: /var/lib/datastax-agent/conf/address.yaml
- Tarball installations: install_location/conf/address.yaml

**lcm.db**

The location of the Lifecycle Manager database *lcm.db* depends on the type of installation:

- Package installations: /var/lib/opscenter/lcm.db
- Tarball installations: install_location/lcm.db

**Note:** The data (cluster topology models, configuration profiles, credentials, repositories, job history, and so forth) for Lifecycle Manager is stored in the *lcm.db* database. Your organization is responsible for backing up the database. You must also configure failover (page 169) to mirror the *lcm.db.*

**lcm.key**

The location of the Lifecycle Manager database encryption key *lcm.key* depends on the type of installation:

- Package installations: /etc/opscenter/lcm.key
- Tarball installations: install_location/keys/lcm.key

**passwd.db**

The default location of the password database *passwd.db* for OpsCenter authentication depends on the type of installation:

- Package installations: /etc/opscenter/passwd.db
- Tarball installations: install_location/passwd.db

Follow these steps to enable automatic failover from the primary OpsCenter instance to the designated backup OpsCenter instance. Enabling failover requires minimal initial set up on the backup OpsCenter.
Important: When configuring automatic failover, both the primary and secondary OpsCenter instances must be running the same OpsCenter version. The OpsCenter daemon (opscenterd) and the DataStax Agents must be running the same version before configuring high availability. DataStax cannot guarantee results if the primary and secondary OpsCenter instances are running different versions.

To enable automatic OpsCenter failover:

Prerequisites:

Warning: Ensure that address.yaml is not being managed by third-party Configuration Management. During failover, OpsCenter automatically changes stomp_interface in address.yaml to point to the backup opscenterd instance. If a separate Configuration Management system is managing address.yaml, that change might be undone when the Configuration Management system pushes its next update.

1. Set up a hostname/IP that can switch between primary and backup OpsCenter instances to avoid changing the browser URL for OpsCenter if a failover occurs.

   If you do not set up a hostname or IP for seamless URL switching post-failover, inform your OpsCenter users of any alternate URL to access OpsCenter.

2. Mirror the configuration directories stored on the OpsCenter primary to the OpsCenter backup using the method you prefer, such as NFS mount or rysnc.

   a. If SSL is enabled, mirror the contents of the SSL configuration directory on the primary OpsCenter machine to the backup OpsCenter machine.

      - /var/lib/opscenter/ssl (package installs)
      - install_location/ssl (tarball installs)

      $ scp /var/lib/opscenter/ssl/* secondary:/var/lib/opscenter/ssl

   b. Mirror the contents of the main configuration directory on the primary OpsCenter machine to the backup OpsCenter machine.

      - /etc/opscenter (package installs)
      - install_location/conf (tarball installs)

      $ scp /etc/opscenter/* secondary:/etc/opscenter

   Note: The failover_configuration_directory should not be mirrored across OpsCenter installs when configuring OpsCenter to support failover.

   c. Mirror the contents of the persist_directory (page 189) location that indicates the current status of the Repair Service. The location of the persist directory for the Repair Service depends on the type of install:
• /var/lib/opscenter/repair_service (package installs)
• install_location/repair_service (tarball installs)

$ scp /var/lib/opscenter/repair_service/* secondary:/var/lib/opscenter/repair_service

Repair Service progress is stored on the filesystem. If using an NFS mount to mirror to, the Repair Service starts up after a failover from approximately the same point where it was interrupted. If manually copying directories or using rsync, the Repair Services resumes from whenever the Repair Service directory was last synced. Otherwise, the Repair Service simply restarts rather than continuing from where it left off.

d. Mirror the Lifecycle Manager database lcm.db:

• /var/lib/opscenter/lcm.db (package installs)
• install_location/lcm.db (tarball installs)

$ scp /var/lib/opscenter/lcm.db secondary:/var/lib/opscenter/lcm.db

e. Mirror the Lifecycle Manager database encryption key lcm.key:

• /etc/opscenter/lcm.key (package installs)
• install_location/keys/lcm.key (tarball installs)

$ scp /etc/opscenter/lcm.key secondary:/etc/opscenter/lcm.key

f. If Lifecycle Manager has generated any certificates for clusters configured to use node-to-node or client-to-node encryption, mirror the Lifecycle Manager certificate authority.

• /var/lib/opscenter/ssl/lcm (package installs)
• install_location/ssl/lcm/cacerts (tarball installs)

$ scp -r /var/lib/opscenter/ssl/lcm secondary:/var/lib/opscenter/ssl/

g. If OpsCenter role-based security (page 101) is enabled, mirror the roles and password database passwd.db:

• # /etc/opscenter/passwd.db (package installs)
  # install_location/passwd.db (tarball installs)

$ scp /etc/opscenter/passwd.db secondary:/etc/opscenter/passwd.db
h. Create and run an automated script to keep the mirrored directories in sync.

The following example cron scripts run rsync to synchronize the configuration directories every 5 minutes for a package installation:

```bash
*/5 * * * * /usr/bin/rsync -az /etc/opscenter
<user>@<backup_host>:/etc/opscenter
*/5 * * * * /usr/bin/rsync -az /var/lib/opscenter/ssl
<user>@<backup_host>:/var/lib/opscenter/ssl
```

The following example cron scripts run rsync to synchronize the configuration directories every 5 minutes for a tarball installation:

```bash
*/5 * * * * /usr/bin/rsync -az install_location/conf
<user>@<backup_host>:install_location/conf
*/5 * * * * /usr/bin/rsync -az install_location/ssl
<user>@<backup_host>:install_location/ssl
```

**Note:**

When a failover occurs, you must manually stop the sync scripts on the former primary and start the sync scripts on the new primary. Failure to do so will result in configuration changes on the new primary being overwritten by stale files from the former primary.

3. If you want to override the default values, edit the [failover] section of the OpsCenter configuration file opscenterd.conf.

**Note:** Making any changes to the opscenterd.conf file requires restarting OpsCenter (page 75).

**Table 9: OpsCenter daemon failover default configuration parameters**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>heartbeat_period</td>
<td>Frequency in seconds with which the primary OpsCenter sends a heartbeat to the backup OpsCenter.</td>
<td>10</td>
</tr>
<tr>
<td>heartbeat_reply_period</td>
<td>Frequency in seconds with which the OpsCenter backup sends a heartbeat to the primary OpsCenter.</td>
<td>300</td>
</tr>
</tbody>
</table>
### Configuring OpsCenter

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>heartbeat_fail_window</td>
<td>Amount of time in seconds that must elapse before the lack of a heartbeat triggers a failover.</td>
<td>60</td>
</tr>
</tbody>
</table>
| failover_configuration_directory | Directory location where failover-specific configuration is stored. The failover_id file is also located in the failover directory.  
**Note:** The failover configuration directory should not be mirrored or replicated across OpsCenter installs when configuring OpsCenter to support failover.  
• /var/lib/opscenter/failover/ (package installs)  
• install_location/failover/ (tarball installs) | |

4. On the backup OpsCenter in the failover directory, create a primary_opscenter_location configuration file that indicates the IP address of the primary OpsCenter daemon to monitor:

   • /var/lib/opscenter/failover/primary_opscenter_location (package installs)  
   • install_location/failover/primary_opscenter_location (tarball installs)

The primary_opscenter_location file should only contain the IP address of the primary OpsCenter instance and nothing more:

```bash
$ cat primary_opscenter_location
55.100.200.300
```

Ensure the user running OpsCenter has at least read permission for the primary_opscenter_location file. Before the backup OpsCenter can take over as the primary OpsCenter, the backup OpsCenter deletes the primary_opscenter_location file in the event of a failover. After a failover, recreate the primary_opscenter_location file on the newly designated backup OpsCenter.

### Configuring named route linking

opscenterd.conf

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

- Package installations: /etc/opscenter/opscenterd.conf
Configuring OpsCenter

- **Tarball installations:** install_location/conf/opscenterd.conf

Enable route URL deep linking to allow navigating to locations within the OpsCenter UI web application and displaying full path URLs in the browser address bar. Enabling the routes feature allows sending a link to users that directs them to a particular page or location within a page, with the URL reflecting the current location within the UI. Routing URLs enhance the user experience with navigating and sharing links, and facilitate learning the structure of the OpsCenter application.

When routes are enabled, clicking on certain areas within the OpsCenter application such as a dashboard preset updates the URL to reflect the current location in the OpsCenter UI. For example, clicking the Latencies tab displays the URL as localhost:8888/opscenter/index.html#Test_Cluster/dashboard/presets/latencies. **Alternately, manually entering a known URL also navigates to the UI location.**

The slash (root) is the overview route. The route is everything in the URL that comes after the hash #. The hash allows for linking and navigation. The routes are not case-sensitive.

If the routes feature is not enabled, the URL does not reflect its current location within the OpsCenter application. The URL always displays as localhost:8888/opscenter/index.html regardless of location.

**Note:** Live routes is a Labs feature. Not all locations in the OpsCenter and Lifecycle Manager UIs currently have a URL available for linking.

1. Open opscenterd.conf for editing and add the following:

   ```
   [labs]
   live_routes = True
   ```

2. **Restart opscenterd (page 75).**

Configuring the OpsCenter JVM

The OpsCenter JVM has been tuned to handle both large and small clusters with initial memory and garbage collection JVM parameters. These default values should be sufficient for most users. However, certain scenarios might arise that might require users to further tune these parameters.
There is an environmental variable `OPSC_JVM_OPTS` that can be set to override the default parameters. Here are some parameters you might consider tuning:

- **-Xmx**: The maximum amount of heap space available for `opscenterd` to use. This default is 1024m (1GB), which should be enough for most installations. If you are managing multiple clusters or otherwise see OutOfMemory errors in `opscenterd.log`, try doubling the heap to 2048m (2GB). If memory-related issues continue, please contact DataStax Support for assistance.

- **-Xms**: The initial amount of heap space that the `opscenterd` process uses on startup. Lower this value if `opscenterd` is having trouble starting.

- **Garbage Collection parameters** (`-XX:+UseConcMarkSweepGC`, `-XX:+CMSParallelRemarkEnabled`, `-XX:+ScavengeBeforeFullGC`, `-XX:+CMSScavengeBeforeRemark`) are not recommended for modification. However, if there is a need to modify and tune other garbage collection parameters, follow this guide [https://docs.oracle.com/javase/8/docs/technotes/guides/vm/gctuning/](https://docs.oracle.com/javase/8/docs/technotes/guides/vm/gctuning/).

- **-Djava.io.tmpdir=/path/to/dir**: The directory that Jython uses to load and execute dynamic classes (default is the system tmp directory). Change this to another directory if you are having trouble loading OpsCenter due to permissions on the default directory.

To modify these parameters, you can simply set the environment variable `OPSC_JVM_OPTS` prior to starting OpsCenter (page 75).

**Note**: If OpsCenter is running while these parameters are set, a restart is required.

Update the maximum heap size JVM argument for a tarball installation using the environment variable:

```
$ export OPSC_JVM_OPTS=-Xmx2048m
$ sudo service opscenterd restart
```

Update the maximum heap size JVM argument for a package installation in `bin/opscenter`:

```
$ sudo service opscenterd restart
```
Configuring OpsCenter

opscenterd.log

The location of the `opscenterd.log` file depends on the type of installation:

- **Package installations:** /var/log/opscenter/opscenterd.log
- **Tarball installations:** `install_location/log/opscenterd.log`

### Configuring the DataStax Agent JVM

The DataStax Agent JVM is designed to have a minimal footprint on your nodes and simply work without extra configuration for most environments.

JVM properties for the agent are configured in `datastax-agent-env.sh`.

#### Agent Heap Size

For most environments, the DataStax agent will require 1GB of heap. If an agent is experiencing OutOfMemory issues, try changing the `-Xmx` property in `datastax-agent-env.sh` to `2048M (-Xmx2048M)`. If issues still persist, please contact DataStax Support so DataStax can determine whether your environment requires even more heap or if there is an issue in the agent.

#### Setting and securing the tmp directory for the agent

Many enterprise security policies require mounting `/tmp` directories with the no execution (`noexec`) flag. OpsCenter runs the JVM using Jython, which requires access to its temporary directory to which it can copy executable files. The JDK requires an executable directory to start and is mapped to `/tmp` by default. If the `/tmp` directory has the `noexec` flag set, the agent cannot start.

By default, the DataStax agent when installed by deb or rpm packages runs as the same user as DSE, which is `cassandra`. If you are running the agent as a different user (tarball install), see Setting permissions to run the agent as the DSE user (page 67).

Remap the default temporary directory to a different (executable) directory and change its permissions to allow full access for the agent. Follow this procedure to securely mount a `/tmp` directory for OpsCenter and the DataStax agent.

**datastax-agent-env.sh**

The default location of the DataStax agent environment shell script `datastax-agent-env.sh` depends on the type of installation:

- **Package installations:** `/etc/datastax-agent/datastax-agent-env.sh`
- **Tarball installations:** `install_location/agent/conf/datastax-agent-env.sh`

1. Create a temporary directory and grant full access to the `cassandra` user.

2. Map the new temporary directory in the `datastax-agent-env.sh` file by specifying its path. Set the `-Djava.io.tmpdir` property to the path for the new `/tmp` directory.
JVM_OPTS="-Xmx512M -Djava.io.tmpdir=/path/to/tmp/dir"

The example also shows doubling the agent max heap size to 512M with the -Xmx property.

3. Save the file and restart the agents (page 76).

**Encrypting JMX communication between the DataStax agent and DSE**

Complete the following steps to encrypt JMX communications between the DataStax agent and DataStax Enterprise (DSE). Enabling this encryption causes the DataStax agent to use an SSL-aware socket factory when connecting to DSE, but does not enable encryption for DSE itself.

By default, JMX remote connections are disabled and JMX security authentication is disabled for both local and remote connections in the cassandra-env.sh file:

```
JVM_OPTS="-Dcom.sun.management.jmxremote.authenticate=false"
```

If you want to configure JMX authentication for DSE, you must Enable DSE Unified Authentication.

**Prerequisites:** Complete all steps to Set up SSL certificates.

1. Stop DSE (page 263) on every node in the cluster.

2. Stop the DataStax agent on each node where it is running.
   - Package installations:
     ```
     $ sudo service datastax-agent stop
     ```
   - Tarball installations:
     To stop the DataStax agent, find the DataStax agent Java process ID (PID) and kill the process using its PID number.
     ```
     $ ps -ef | grep datastax-agent
     $ sudo kill pid
     ```

3. Navigate to the jvm.options file and add the following section for SSL settings.

   ```
   # SSL settings
   -Dcom.sun.management.jmxremote.ssl=true
   -Dcom.sun.management.jmxremote.ssl.need.client.auth=true
   -Djavax.net.ssl.keyStore=/path_to_keyStore/keyStore_name.jks
   -Djavax.net.ssl.keyStorePassword=keyStore_password
   ```
Configuring OpsCenter

-Djavax.net.ssl.trustStore=/path_to_trustStore/trustStore_name.jks
-Djavax.net.ssl.trustStorePassword=trustStore_password

com.sun.management.jmxremote.ssl
Set to true to enable SSL for JMX connections between DSE and the DataStax agent. If enabled, com.sun.management.jmxremote.ssl.need.client.auth must also be set to true to enable full security.

com.sun.management.jmxremote.ssl.need.client.auth
Set to true to enable two-way certificate authentication.

keyStore_name
Name of the SSL keystore.

keyStore_password
Password for the keystore indicated by javax.net.ssl.keyStore.

trustStore_name
Name of the SSL truststore.

trustStore_password
Password for the truststore indicated by javax.net.ssl.trustStore.

4. Start DSE on each node in the cluster.

$ sudo service dse start

Wait for the nodes to become available before continuing.

5. To run the DataStax agent with SSL encryption, add the following settings to the datastax-agent-env.sh file on the nodes where the DataStax agent is running.

Important: The keystore for DSE is the truststore for the DataStax agent (and vice versa), as shown in the following example. The keyStore variable points to the trustStore, and the keyStorePassword variable uses the trustStore password. The opposite is true for the trustStore variable.

JVM_OPTS="$JVM_OPTS -Xmx1024M"
JVM_OPTS="$JVM_OPTS -Djavax.net.ssl.keyStore=/path_to_trustStore/trustStore_name.jks"
JVM_OPTS="$JVM_OPTS -Djavax.net.ssl.keyStorePassword=trustStore_password"
JVM_OPTS="$JVM_OPTS -Djavax.net.ssl.trustStore=/path_to_keyStore/keyStore_name.jks"
JVM_OPTS="$JVM_OPTS -Djavax.net.ssl.trustStorePassword=keyStore_password"
JVM_OPTS="$JVM_OPTS -Ddatastax.agent.jmx.usessl=true"

datastax.agent.jmx.usessl
Add this setting and set to true to enable SSL encryption for the DataStax agent.

6. Restart the DataStax agent for changes to take effect.
Configuring OpsCenter

- **Package installations:**
  
  ```
  $ sudo service datastax-agent restart
  ```

- **Tarball installations:**
  
  a. To stop the DataStax agent, find the DataStax agent Java process ID (PID) and kill the process using its PID number:

  ```
  $ ps -ef | grep datastax-agent
  $ sudo kill pid
  ```

  b. Start the agent:

  Use the `-f` option to start the agent in the foreground.

  ```
  $ install_location/bin/datastax-agent
  ```

7. After restarting the DataStax agent on each node where it is running, check OpsCenter to ensure that all DataStax agents are running and that all nodes are available.

### Changing the replication strategy for the OpsCenter keyspace

Using the NetworkTopologyStrategy replication is recommended for the OpsCenter keyspace when running OpsCenter in a multi-datacenter environment. Prompts guide changing the keyspace to a compatible replication strategy.

**Warning:** The OpsCenter keyspace is using SimpleStrategy replication, which is not recommended in multi-datacenter clusters. 

1. If the OpsCenter keyspace warning is visible when you first launch OpsCenter, expand the warning and click the link to edit the keyspace (page 266).

   The **Edit Keyspace keyspace_name** dialog for the keyspace appears.
2. Change the strategy from SimpleStrategy to NetworkTopologyStrategy.

3. Enter the datacenters and replication factor for each datacenter. Click Add Datacenter to add more datacenters.

4. Set the durable_writes option to True (recommended) or False (default).

    **Warning:** Be careful when setting the durable writes option. When set to False, data written to the keyspace bypasses the commit log, which could cause data loss.

5. Click Save Keyspace.

### Configuration files for OpsCenter

Configure capabilities by manually modifying the opscenterd.conf, cluster_name.conf, and address.yaml configuration files.

**Note:** The OpsCenter console is the most convenient way to configure basic OpsCenter connection settings (*page 93*) for authentication and encryption.

- opscenterd.conf: configures the properties of the OpsCenter daemon.
- cluster_name.conf: configures properties for each cluster monitored by OpsCenter. OpsCenter creates the cluster_name.conf file named the same as the cluster when a cluster is added to OpsCenter.

  **Tip:** You can set most of the agent configuration options in the [agent_config] section of the cluster_name.conf file. The options in the [agent_config] section must match the corresponding configuration option name in address.yaml. Setting agent options through the cluster configuration file sets the corresponding property in address.yaml on every node. Some properties or some cases might require setting these properties directly in address.yaml.
on applicable agents. Setting agent configuration options that require a list entry (value1, value2, value3 and so forth) is not supported in \texttt{cluster\_name.conf}.

- \texttt{address.yaml}: configures the properties for the DataStax agent. Setting options in \texttt{address.yaml} must be done for every node. You can set most of these properties in the \texttt{[agents]} section of \texttt{cluster\_name.conf} on the opscenterd machine and the properties propagate automatically to all agents.
- \texttt{logback.xml}: configures the logging properties for the OpsCenter daemon. This includes HTTP request logging, cluster logging, and security logging.

**Precedence for OpsCenter configuration files**

In versions of OpsCenter prior to 5.2, the settings in the \texttt{cluster\_name.conf} configuration file took precedence over settings in \texttt{address.yaml}. In OpsCenter version 5.2 and going forward, the reverse is true: \texttt{address.yaml} settings take precedence over \texttt{cluster\_name.conf}. To summarize, OpsCenter 5.2 and later configuration files precedence, settings in \texttt{address.yaml} override settings in \texttt{cluster\_name.conf}, which in turn override default configuration settings.

**Changes to configuration options**

To keep current on changes to available configuration options with each release, be sure to review the Upgrade Guide.

\texttt{cluster\_name.conf}

The location of the \texttt{cluster\_name.conf} file depends on the type of installation:

- Package installations: /etc/opscenter/clusters/\texttt{cluster\_name.conf}
- Tarball installations: \texttt{install\_location/conf/clusters/cluster\_name.conf}

\texttt{opscenterd.conf}

The location of the \texttt{opscenterd.conf} file depends on the type of installation:

- Package installations: /etc/opscenter/\texttt{opscenterd.conf}
- Tarball installations: \texttt{install\_location/conf/opscenterd.conf}

\texttt{address.yaml}

The location of the \texttt{address.yaml} file depends on the type of installation:

- Package installations: /var/lib/datastax-agent/conf/address.yaml
- Tarball installations: \texttt{install\_location/conf/address.yaml}

\texttt{logback.xml}

The location of the \texttt{logback.xml} file depends on the type of installation:

- Package installations: /etc/opscenter/logback.xml
• Tarball installations: install_location/conf/logback.xml

OpsCenter configuration properties

These properties are configured in the opscenterd.conf file.

**Note:** After changing properties in this file, restart OpsCenter (page 75) for the changes to take effect.

[webserver] port
The HTTP port used for client connections to the OpsCenter web server. The default port is 8888.
Optional HTTPS support. To enable, remove the comment markers (#) in front of properties prefixed with ssl in the opscenterd.conf file, as described in Configuring HTTPS (page 79).

[webserver] interface
The interface that the web server uses to listen for client connections. The interface must be an externally accessible IP address or host name. The default interface is 127.0.0.1.

[webserver] staticdir
The location in the file system where static content for the OpsCenter web site resides. The default location is /usr/share/opscenter/content for package installations and install_location/content for tarball installations.

[webserver] ssl_keyfile
The location where the SSL keyfile resides. This option requires ssl_certfile and optionally ssl_port to also be set.

[webserver] ssl_certfile
The location where the SSL certificate resides. This option requires ssl_keyfile and optionally ssl_port to also be set.

[webserver] ssl_port
The port on which to serve SSL traffic. The default port is 8443.

[webserver] hsts_enabled
Determines whether to use HTTP Strict Transport Security. Enabling this option causes OpsCenter to return an HSTS header in HTML responses that go over HTTPS. The HTHS header makes HTTPS->HTTP downgrade attacks more difficult by instructing supported browsers to refuse connecting to OpsCenter over unencrypted HTTP. Enabling this option also requires enabling TLS with ssl_certfile and ssl_keyfile. See also hsts_max_age. Default: False.

[webserver] hsts_max_age
The length of time in seconds that supported browsers should consider an HSTS header fresh and refuse to connect to OpsCenter with unencrypted HTTP. See also: hsts_enabled. Default: 31536000 seconds (1 year).

[webserver] sub_process_timeout
The time in seconds OpsCenter waits for subprocesses to complete before a timeout. The default value is 600 seconds. OpsCenter spawns subprocesses for some tasks, such as scp, these tasks have a configurable timeout.

[webserver] jvm_fn_pool_size
Some functions are extremely slow and are evaluated using an underlying java thread pool so that they use multiple cores instead of the single core used by twisted. This setting specifies the size of that underlying thread pool.

### [http_proxy_settings] proxy_url
The URL of a proxy server that OpsCenter will use for external http/https connections. Currently only applies to the posturl plugin. Example: http://127.0.0.1:8080

### [http_proxy_settings] proxy_username
Proxy username for basic proxy HTTP authorization. Currently only applies to the posturl plugin.

### [http_proxy_settings] proxy_password
Proxy password for basic proxy HTTP authorization. Currently only applies to the posturl plugin.

### [security] config_encryption_active
Specifies whether opscenter should attempt to encrypt sensitive config values

### [security] config_encryption_key_name
Name of the system key used to encrypt / decrypt passwords stored.

### [security] config_encryption_key_path
Path to the encryption key. If left blank the directory of opscenterd.conf will be used

### [logging] resource_usage_interval
The interval, in seconds, in which OpsCenter logs the system resource usage. The default value is 60.

### [logging] ignored_dict_keys
These are dictionary keys that should not be logged. We have a static list that we never log but additional keys can be added here.

### [stomp] port
The port the stomp service uses to communicate with the front end. The default port is 61619.

### [stomp] interface
The interface the stomp service uses to communicate with the front end. The default interface is 127.0.0.1.

### [stomp] ui_stomp_timeout
The max timeout (in seconds) for stomp connections with the front end. If you experience difficulties with the OpsCenter UI, try increasing this value.

### [definitions] use_ssl
Specifies whether SSL should be used to get definition file updates. This option requires OpenSSL on the OpsCenter host. The default value is True.

### [definitions] definitions_dir
The file system location where definition files are stored. The default location is /var/lib/opscenter/definitions for package installations and install_location/definitions for tarball installations.

### [definitions] auto_update
Specifies whether OpsCenter should automatically attempt to periodically update the definition files. The default value is True.

### [definitions] download_host
The host that definition file update requests will be made to. The default host is opscenter.datastax.com.

### [definitions] download_port
Configuring OpsCenter

The port used to request definition file updates on download_host. The default port is 443.

[definitions] download_filename
The name of the tar file on the download_host that contains definition file updates. The default name is definitions_files.tgz.

[definitions] hash_filename
The name of the hash file on the download_host used to determine if the definition file requires an update. The default file is version.md5.

[definitions] sleep
The duration in seconds between checks to update the definition files. The default time is 3600 seconds (1 hour).

[authentication] passwd_db
Full path to the file for configuring password authentication (page 101) for OpsCenter. If this file does not exist, OpsCenter does not verify passwords. The default location is /etc/opscenter/passwd.db for package installations and install_location/passwd.db for tarball installations.

[authentication] enabled
Configures whether user authentication is enabled or not. The default setting is False.

[authentication] timeout
This sets the session timeout, in seconds. Defaults to no timeout. If a timeout is desired, the minimum value it can be set to is 60 seconds.

[authentication] authentication_method
Configured Authentication method, options include DatastaxEnterpriseAuth and LDAP, for Active Directory we support AD over LDAP, so please specify LDAP.

[authentication] password_hash_type
The 1-way password hash algorithm to use when storing user passwords. Options include: bcrypt+blake2b-512 (default), pbkdf2+blake2b-512, pbkdf2+sha512, pbkdf2+sha3-256, bcrypt+sha512.

[authentication] sqlite_timeout
The number of seconds to wait before a query to SQLite is considered hung and should be aborted.

[authentication] sqlite_connection_timeout
The number of seconds that opscenterd waits to receive a successful SQLite connection before a timeout. Default: 5

[authentication] sqlite_max_active_connections
The maximum number of simultaneous active connections to the SQLite database. Default: 200

[authentication_provider] Empty_Section
Empty Section

[ldap] server_host
The LDAP server host

[ldap] server_port
The LDAP server port

[ldap] uri_scheme
URI Scheme, defaults to ldap for ldap_security = None, defaults to ldaps for ldap_security = SSL or TLS
[ldap] **search_dn**
DN of the user that be used to search for users on the LDAP server. This user should only have the necessary permissions to do the search. If not present then an anonymous bind will be used for the search. Example: `cn=admin,dc=qaldap,dc=datastax,dc=lan`.

[ldap] **search_password**
Password of the search user.

[ldap] **user_search_base**
The ldap search base used to find a user, example:
`ou=users,dc=qaldap,dc=datastax,dc=lan`.

[ldap] **user_search_filter**
The ldap search filter used to uniquely identify a user, example: `(&uid={0})`.

[ldap] **group_search_base**
The ldap search base used to find a group, example:
`ou=groups,dc=qaldap,dc=datastax,dc=lan`.

[ldap] **group_search_filter**
Deprecated. The ldap search filter used to find a user's group, example:
`(member=cn={0},ou=users,dc=nodomain)`.

[ldap] **group_search_filter_with_dn**
The ldap search filter used to find a user's group. Uses the full user's DN from a user search. Overrides **group_search_filter**. Example: `(member={0})`.

[ldap] **group_name_attribute**
The ldap field name used to identify a group's name, example: `cn`.

[ldap] **admin_group_name**
The name of the admin group or a comma separated list of admin group names, example: `admin, superusers`.
If your group name contains restricted LDAP characters such as ",", you must escape them for example two admin groups "foo, bar" and "baz" should be entered as: `foo \, bar, baz`.

[ldap] **ldap_security**
The type of security to use with LDAP: None, TLS, SSL, SSL_TLS when set to TLS or SSL_TLS uses TLS start.

[ldap] **truststore**
Path to the truststore.

[ldap] **truststore_type**
Type for the truststore.

[ldap] **truststore_pass**
Password for the truststore.

[ldap] **user_memberof_attribute**
Set to the attribute on the user entry containing group membership information.

[ldap] **user_memberof_stores_dn**
Set to True if the memberof attribute's value is distinguished names of groups. Default: False.

[ldap] **group_search_type**
The group_search_type defines how group membership will be determined for a user. It
can be one of:

directory_search - will do a subtree search of group_search_base using
group_search_filter to filter the results
memberof_search - will get groups from the memberof attribute of the user. This
requires the directory server to have memberof support

[ldap] follow_referrals
Sets whether the OpsCenter LDAP client should follow referrals. AD typically does
not follow referrals. Default: False.

[ldap] enforce_single_user_search_result
Returns an error when multiple entries are returned from a user search after all
referrals (if applicable) are followed. Set to False if the user_search_base is not
confined to one particular OU. Defaults to True.

[ldap] connection_timeout
The number of seconds to wait before concluding the ldap server is down, defaults
to 30 seconds.

[agents] config_sleep
The durations in seconds in between Updates to the agent config md5. The default
value is 420 seconds (7 minutes).

[agents] ssh_port
The Secure Shell (SSH) port that listens for agent-OpsCenter communications.
The default port is 22. Add an [agents] section, if one doesn't already exist, to the
opscenterd.conf. In this section, add the ssh_port option and a value for the port
number:

```
ssh_port = 2222
```

[agents] incoming_port
The port used by OpsCenter for incoming stomp communication. The default port is
61620.

[agents] incoming_interface
The interface used by OpsCenter for incoming stomp traffic from the DataStax
Agents. The default interface is 0.0.0.0.

[agents] use_ssl
Specifies whether traffic between OpsCenter and the DataStax Agents should use
SSL. The default value is False.

[agents] agent_install_poll_period
How often OpsCenter will update agent install status during an agent install

[agents] agent_install_mute_period
The number of seconds to wait after agent install before checking DataStax Agents
for known problems

[agents] agent_install_timeout_period
The number of seconds to wait for the install to complete before automatically
terminating the install.

[agents] ssl_keyfile
The location of the SSL key file used for SSL traffic between OpsCenter and
the DataStax Agents. The default location is /var/lib/opscenter/ssl/
opscenter.key for package installations and install_location/ssl/
opscenter.key for tarball installations.

[agents] ssl_certfile
The location of the SSL certificate used for SSL traffic between OpsCenter and the DataStax Agents. The default location is /var/lib/opscenter/ssl/opscenter.der for package installations and install_location/ssl/opscenter.der for tarball installations.

[agents] agent_keyfile
The location of the keyfile sent to the DataStax Agents when using SSL communication between OpsCenter and the DataStax Agents. The default location is /var/lib/opscenter/ssl/agentKeyStore for package installations and install_location/ssl/agentKeyStore for tarball installations.

**Note:** Do not use the agent_keyfile when manually generating and deploying keys.

[agents] agent_keyfile_raw
The raw keystore file stored in the Java keystore from agent_keyfile. This parameter is required only when configuring high availability, so that the secondary OpsCenter instance can communicate with the primary OpsCenter instance. The failover opscentered processes on the secondary OpsCenter instance use this key to establish a STOMP connection to the primary opscentered instance.

[agents] agent_certfile
The location of the certfile sent to the DataStax Agents when using SSL communication between OpsCenter and the DataStax Agents. The default location is /var/lib/opscenter/ssl/agentKeyStore.der for package installations and install_location/ssl/agentKeyStore.der for tarball installations.

[agents] ssh_executable
The location of the ssh executable binary. The default locations is /usr/bin/ssh.

[agents] scp_executable
The location of the scp executable binary. The default location is /usr/bin/scp.

[agents] ssh_keygen_executable
The location of the ssh-keygen executable binary. The default location is /usr/bin/ssh-keygen.

[agents] ssh_keyscan_executable
The location of the ssh-keyscan executable binary. The default location is /usr/bin/ssh-keyscan.

[agents] ssh_user_known_hosts_file
The location of the OpsCenter user's known_hosts file that will be used by OpsCenter during SSH communications. The default location is ~/.ssh/known_hosts.

[agents] ssh_sys_known_hosts_file
The location of the system wide known_hosts file that will be used by OpsCenter during SSH communications. The default location is /etc/ssh/ssh_known_hosts.

[agents] tmp_dir
The path to a tmp directory on the OpsCenter machine used for installing DataStax Agents. The default location is /usr/share/opscenter/tmp/ for package installations and install_location/tmp for tarball installations.

[agents] not_seen_threshold
The maximum time in seconds since the last agent status about a specific connection, such as stomp, was sent before that agent connection is considered
down. This threshold also affects how long OpsCenter waits before marking node health as unknown. Default value: 180 seconds.

[agents] call_agent_retry
Number of times to try different up nodes on agent requests

[agents] agent_aggregation_flush
The number of seconds between log flushes for verbose aggregated and periodically batched agent messages. Default: 600 seconds.

[agents] http_poll_period
The frequency in seconds between attempts to poll agent http health. Default value: 60 seconds.

[stat_reporter] initial_sleep
The delay in seconds before the cluster stats reporter starts to run. The default value is 300 (5 minutes).

[stat_reporter] interval
The interval in seconds between usage metric reports to DataStax Support. By default, OpsCenter sends usage metrics about the cluster to DataStax Support every day.

To disable the phone-home functionality, add the following lines to your opscenterd.conf file:

```
interval = 0
```

Additional configuration metric collection properties are available in Metrics Collection Properties (page 152).

[stat_reporter] url
The URL to which the metric usage report is sent for phone-home. The default URL is phonehome.datastax.com.

[stat_reporter] port
The port for the metric usage report phone-home service. The default port is 8889.

[stat_reporter] ssl_port
If communication using SSL is possible, then use this port for the phone-home service. The default port is 443.

[stat_reporter] ssl_key
The location of the SSL key file to use for SSL communication for the phone-home service. The default location is /var/lib/opscenter/ssl/stats.pem for package installations and install_location/ssl/stats.pem for tarball installations.

[spark] base_master_proxy_port
Base port to use for setting up the HTTP proxy for the Spark master. Spark master UI is exposed at port 7080 and increasing ports from that + 10k (17080) are largely unused.

[feedback] host
The host to which to send OpsCenter user feedback. The default host is phonehome.datastax.com.

[feedback] port
The port usee when sending OpsCenter user feedback. The default port is 8890.

[labs] orbited_longpoll
This option increases the time between polling requests to orbited for data updates.
Configuring OpsCenter

[labs] latest_version_check
   Enables or disables the latest version check in the OpsCenter UI.

[labs] enable_dashboard_preset_import_export
   Enables import/export of dashboard presets.

[labs] live_routes
   Enable the display of client-side routes in the location bar.

[labs] log_enable
   Enables Special Log Management Program (SLMP) that allows various forms of log
   management functionalities.

[repair_service] persist_directory
   The location in which to store a file with the current repair service status. The
   default location is /var/lib/opscenter/repair_service for package
   installations and install_location/repair_service for tarball installations.

[repair_service] restart_period
   The period of time in seconds repair service pauses in response to certain events
   before verifying the cluster stability and restarting repairs. Default: 300 (5 minutes).

[repair_service] cluster_stabilization_period
   The frequency in seconds the repair service checks for cluster stability before
   making repairs. This check begins when repair service is activated (either by a user
   or after an OpsCenter restart) and repeats until the cluster is stable. Default: 30.

[repair_service] single_task_err_threshold
   The maximum number of times to retry a repair task before temporarily skipping the
   task and moving on to the next task. The skipped task is moved to the end of the
   repairs queue to retry later. After the maximum retries is reached, an alert is fired.
   Default: 10.

[repair_service] max_parallel_repairs
   The maximum number of subrange repairs to run in parallel. If unspecified or set to
   0, the Repair Service calculates the correct number of maximum repairs to run in
   parallel. Default: 0.

[repair_service] max_pending_repairs
   The maximum number of pending repairs allowed to be running on a node at one
time. Default: 5.

[repair_service] single_repair_timeout
   The maximum length of time for a repair to complete, in seconds. Default: 3600 (1
hour).

[repair_service] min_repair_time
   The minimum length of time in seconds for a repair to complete. If a repair finishes
   sooner it will be padded with a sleep. Default: 5.

[repair_service] prioritization_page_size
   The maximum number of remaining subrange repair tasks for the repair service
   to evaluate when choosing the next low-impact repair to run. Default: 512. This
   parameter applies to subrange repairs only.

[repair_service] offline_splits
   The minimum number of subrange splits for a node to have per keyspace when
   falling back to offline splits if the node is unable to communicate with its agent to
   get more optimal splits. Default: 256. This parameter applies to subrange repairs
   only.

[repair_service] min_throughput
Configuring OpsCenter

The minimum throughput in bytes needed to calculate parallel repairs. See also num_recent_throughputs. Default: 512. This parameter applies to subrange and incremental repairs only, and is not applicable to distributed subrange repairs.

[repair_service] num_recent_throughputs
The maximum most recent completed repair throughputs used to calculate the average repair throughput, which is then used to determine how many parallel repairs are needed. See also min_throughput. Default: 500. This parameter applies to subrange and incremental repairs only, and is not applicable to distributed subrange repairs.

[repair_service] error_logging_window
The frequency in seconds to log errors and trigger alerts after exceeding time_to_completion. Default: 86400 (1 day).

[repair_service] snapshot_override
Specifies whether to override the default snapshot repair behavior. Specifying this option as True runs validation compaction sequentially rather than in parallel. Default: False.

[repair_service] ignore_keyspaces
The list of keyspaces to ignore during a subrange or distributed subrange repair, in addition to system keyspaces or those with a replication factor of 1.

[repair_service] ignore_tables
The list of keyspace-qualified table names to exclude during a subrange or distributed subrange repair, including reserved tables, those that reside in a system keyspace, and tables belonging to keyspaces with a replication factor of 1. Example: keyspace1.standard1, keyspace1.standard2.

[repair_service] incremental_repair_tables
The list of keyspaces and tables to include in incremental repairs. The OpsCenter.settings and OpsCenter.backup_reports tables are included by default. Example: keyspace1.standard1, keyspace1.standard2.

[repair_service] incremental_repair_datacenters
Restricts incremental repairs by datacenters or racks. Setting this option improves performance by limiting the repair requests to only those replicas within the datacenters and any specified racks. Example: dc1,dc2:rack1,dc2:rack2. The default behavior sends repair requests to all datacenters and racks for all replicas.

[repair_service] incremental_sleep
The number of seconds to pause after completing all incremental repairs for the cluster. Default: 3600 (1 hour).

[repair_service] incremental_threshold
The minimum number of bytes required to consider a table for incremental repairs (DSE 5.1+ only). The default value of 1 byte means that if there is any unrepaired data in a table, the Repair Service will run an incremental repair. Be cautious of setting this value too high. If not enough data is written to exceed the threshold in the gc_grace_seconds period, deletes might be lost. Default: 1.

[repair_service] incremental_err_alert_threshold
The threshold for the number of errors during incremental repair to ignore before alerting that incremental repair seems to be failing more than an acceptable amount. Default: 20.

[repair_service] time_to_completion_target_percentage
A percentage of the time to completion that the repair service should target, including slowing down or reducing parallelism as necessary to avoid overtaxing the cluster. Default: 65. This parameter applies to subrange and incremental repairs only, and is not applicable to distributed subrange repairs.

**[repair_service] tokenranges_http_timeout**
The timeout in seconds for the HTTP call to the agent to retrieve node token range splits. The default value (30) is deliberately set higher than the default value for DataStax Agents http_timeout in clustername.conf (10). Default: 30. This parameter applies to subrange repairs only, and is not applicable to incremental or distributed subrange repairs.

**[repair_service] persist_period**
The minimum number of seconds between Repair Service writing the persist file to disk. Default: 3600 (1 hour). This parameter applies to subrange and incremental repairs only, and is not applicable to distributed subrange repairs.

**[repair_service] tokenranges_partitions**
Target number of partitions for each range in a subrange repair. This value is used by the Repair Service to create repair tasks, and is set to the max DSE merkle tree depth by default. Default: 1048576.

**Note:** Do not set tokenranges_partitions higher than default. See Tuning Repair Service for multi-datacenter environments (page 495).

**[repair_service] max_down_node_retry**
The maximum number of attempts to retry a repair task when a node containing a replica is down. The default is 1080 retry attempts. Retries occur every 10 seconds. The default 1080 retries elapses after 10800 seconds (3 hours), which corresponds to the default cassandra hinted-handoff expiration. Example: To double the time allowed to attempt repairs on a down node or replica to 6 hours, set the number of retries to 2160. Default: 1080.

**[repair_service] enable_distributed_subrange_repair**
Specifies whether to enable the Distributed Subrange Repair job type in the Repair Service, which is designed to better scale for large, dense clusters that require an extremely large number of individual subrange repairs. Default: False.

**[repair_service] parallel_tasks_update_interval**
Interval at which to recalculate parallel tasks to use for the Repair service in seconds. Default: 120 (2 minutes). This parameter applies to subrange and incremental repairs only, and is not applicable to distributed subrange repairs.

**[ui] default_api_timeout**
The default timeout value in seconds for an API call from the OpsCenter UI to the OpsCenter API. The default value is 10. Some API calls require a timeout longer than 10 seconds. In those cases, the API call timeouts are scaled relative to the default_api_timeout (for example, 6 * default_api_timeout). Changing the default_api_timeout affects those timeouts accordingly.

**[ui] max_metrics_requests**
The maximum concurrent metrics requests from the OpsCenter UI to opscenterd. The default value is 16.

**[ui] node_detail_refresh_delay**
The time in seconds between polling calls to update node details. The default value is 5.
Configuring OpsCenter

**[ui] storagemap_ttl**
How often, in seconds, the data in the storage capacity chart is updated in the OpsCenter UI. It is set to 300 seconds (5 minutes) by default so changes to storage capacity on individual nodes may not be reflected in the UI for up to 5 minutes.

**[request_tracker] queue_size**
The maximum number of requests that can be tracked. The default value is 10,000.

**[clusters] add_cluster_timeout**
How long, in seconds, OpsCenter will wait when adding a cluster before reporting an error. The default value is 300 seconds. Adding a cluster includes connecting to DSE, getting topology information, and creating the OpsCenter schema. Increasing this value may be necessary when running a very large cluster with vnodes enabled.

**[clusters] startup_sleep**
How long, in seconds, OpsCenter will wait between connecting to clusters on startup. The default value is 0 (no wait).

**[clusters] max_schema_agreement_wait**
The maximum time in seconds to wait for schema agreement after executing a schema-altering query. Default value: 30 seconds. Increase the value if there are schema creation or schema alteration errors present in the opscenterd log.

**[failover] heartbeat_period**
How often OpsCenter should heartbeat to the backup. Default: 10 seconds.

**[failover] heartbeat_reply_period**
How often a backup OpsCenter should heartbeat to the primary Opscenter. Default: 300 seconds.

**[failover] heartbeat_fail_window**
The amount of time required before a lack of heartbeat triggers failover. Default: 60 seconds.

**[failover] failover_configuration_directory**
The directory where failover specific configuration is stored. This directory should not be mirrored/replicated across OpsCenter installs when configuring OpsCenter to support failover. The default location is /var/lib/opscenter/failover/ for package installations and install_location/failover/ for tarball installations.

**[failover] override_primary_redirect_url**
If set, overrides the primary OpsCenter URL automatically generated during failover. Example: https://opscenter:80

**[lifecycle_manager] db_location**
The location of the lcm.db database used for storing Lifecycle Manager information. Default: /var/lib/opscenter/lcm.db

**[lifecycle_manager] cacerts_directory**
The directory to use when storing SSL-related files that are automatically generated during provisioning tasks. Default: /var/lib/opscenter/ssl/lcm

**[lifecycle_manager] cacerts_truststore_location**
The location of the truststore containing the root CA cert for signing certificates. Default: /var/lib/opscenter/ssl/lcm/lcm-auto-generated.truststore

**[lifecycle_manager] cacerts_truststore_password**
The password to open the cacerts truststore.

**[lifecycle_manager] crypto_key_file**
The key to use to encrypt sensitive data in the database for storing Lifecycle Manager information. The encryption key is automatically generated if it does not already exist. Default: /etc/opscenter/lcm.key

```
[lifecycle_manager] job_manager_polling_interval
Polling frequency in seconds when performing Lifecycle Manager tasks. Default: 5 seconds.

[lifecycle_manager] node_install_idle_timeout
Duration of inactivity in seconds before marking an install job as idle. Default: 900 seconds.

[lifecycle_manager] node_restart_idle_timeout
Duration of inactivity in seconds before marking a node restart job as idle. Default: 120 seconds.

[lifecycle_manager] node_configure_idle_timeout
Duration of inactivity in seconds before marking a node configuration job as idle. Default: 120 seconds.

[lifecycle_manager] node_cluster_import_idle_timeout
Duration of inactivity in seconds before marking a cluster import job as idle. Default: 120 seconds.

[lifecycle_manager] node_install_agent_idle_timeout
Duration of inactivity in seconds before marking an agent install job as idle. Default: 120 seconds.

[lifecycle_manager] cassandra_connection_timeout
Duration in seconds to allow for connecting to DataStax Enterprise when provisioning a DataStax Enterprise cluster. Default: 20 seconds.

[lifecycle_manager] meld_directory
The path to the meld provisioning script

[lifecycle_manager] opscenterd_agent_package_directory
The directory lcm will find agent packages in for installing DataStax Agents.

[lifecycle_manager] ssh_connect_timeout_in_seconds
The number of seconds to wait for the SSH server on a target node to respond to a connection attempt during an LCM job. Default: 30 seconds.

[lifecycle_manager] ssh_max_attempts
The maximum number of times to attempt an SSH connection to a given target during an LCM job. Default: 3.

[lifecycle_manager] ssh_retry_delay_in_seconds
The number of seconds to wait between SSH connection retries during an LCM job. Default: 10 seconds.

[lifecycle_manager] disable_platform_check
Platform check ensures DSE compatibility and support. Disable at your own risk. Default: False.

[backup_service] auto_sync_retry_attempts
The number of times to auto retry sending a backup to a destination if there are failures. Default: 3.

[backup_service] auto_sync_retry_delay
The amount of time in minutes to wait between retry attempts. Default: 60.

[backup_service] destination_validation_timeout
The amount of time in seconds to wait for DataStax Agents to respond when validation a backup destination. Default: 30.
```
[backup_service] s3_proxy_host
The optional proxy host for connection to AWS S3

[backup_service] s3_proxy_port
The optional proxy port for connection to AWS S3

[agents] path_to_sudowrap
The location of the sudo_with_pass.py wrapper for old Red Hat installations. The default location is /usr/share/opscenter/bin/sudo_with_pass.py for package installations and install_location/bin/sudo_with_pass.py for tarball installations.

[agents] path_to_deb
The path to the agent Debian package. The default location is /usr/share/opscenter/agent/datastax-agent.deb for package installations and install_location/agent/datastax-agent.deb for tarball installations.

[agents] path_to_rpm
The path to the agent RPM package. The default location is /usr/share/opscenter/agent/datastax-agent.rpm for package installations and install_location/agent/datastax-agent.rpm for tarball installations.

[agents] reported_interface
The interface used when automatically setting up target nodes to communicate with opscenterd. The value is used when installing the DataStax Agent automatically through OpsCenter, during failover, and is also used by the remote script deployed by the Lifecycle Manager (LCM) to communicate status back to the central opscenterd process. The value is written as the stomp_interface property in address.yaml. If not explicitly set, this value is determined by an algorithm that works automatically in most cases. The reported_interface property should be set if that algorithm fails to determine the proper IP that DataStax Agents should use to connect (that is, if stomp_interface is configured incorrectly in address.yaml, or the remote LCM script fails to report back).

[agents] runs_sudo
Sets whether the DataStax Agent will be run using sudo or not. The default value is True. Setting this option to False means the agent will not use sudo, and the agent user will not run using elevated privileges. Setting this option to True means the agent will run using sudo, and elevated privileges.

Statistics reporter properties

OpsCenter communicates data about an installed instance back to DataStax for informational purposes. The data is sent in a key-value JSON format.

If your organization prefers disabling transmission of the collected data to DataStax, add the following in opscenterd.conf and restart OpsCenter (page 75):

[stat_reporter]
interval=0

The following information is recorded about the OpsCenter install:

install_id
This is a random uuid generated when OpsCenter starts for the first time. This is used for associating reports from the same install.
is_paid
   This is a flag indicating whether or not this is the free or enterprise version of
   OpsCenter.
opscenter_version
   The version of OpsCenter in use.
opscenter_ram
   The amount of RAM, in megabytes, on the OpsCenter machine.
opscenter_cores
   The number of cores on the OpsCenter machine.
opscenter_os
   The generic name of the operating system of the OpsCenter machine. For
   example, linux or mac.
opscenter_os_sub
   The specific name of the operating system of the OpsCenter machine. For
   example CentOS, Ubuntu, or Debian.
opscenter_os_version
   The operating system version of the OpsCenter machine.
opscenter_arch
   The architecture of the OpsCenter machine.
opscenter_install_type
   The type of install (package or tarball).
python_version
   The version of python running on the OpsCenter machine.
opscenter_instance_type
   The instance type the OpsCenter machine, if OpsCenter is running in EC2.
separate_storage
   A flag indicating if OpsCenter is storing metrics in the cluster it is monitoring.
config_diff
   A list of the OpsCenter config options that were modified to be different than the
   defaults. This includes the names of the options that were changed but not the
   values of those options.

These statistics are collected about each cluster OpsCenter is monitoring:

cluster_id
   An MD5 hash of the cluster name. Used for identifying unique clusters while
   maintaining anonymity.
conf_id
   An MD5 hash of the file name the config for the cluster is stored in. Used for the
   same purposes as cluster_id.
partitioner
   The partitioner the cluster is using.
snitch
   The snitch the cluster is using.
keyspace_count
   The number of keyspaces in the cluster.
columnfamily_count
Configuring OpsCenter

The number of column families in the cluster.

**strategy_options**
A list of the replication options used for each keyspace in the cluster.

**cql3_cf_count**
The number of column families created with CQL3 in the cluster.

**node_count**
The number of nodes in the cluster.

**avg_token_count**
The average number of tokens per node.

**bdp_version**
A list of the different DataStax Enterprise versions in the cluster.

**rack_map**
A map of each rack in the cluster and how many nodes are in that rack.

**dc_count**
The number of data centers in the cluster.

**free_space**
The amount of free disk space across the cluster.

**used_space**
The amount of used disk space across the cluster.

**cluster_os**
A list of the different operating systems used across the cluster.

**cluster_ram**
The average amount of ram per node in the cluster.

**cluster_cores**
The average number of cores per node in the cluster.

**cluster_instance_types**
A list of the EC2 instance types in the cluster, if EC2 is being used.

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

- **Package installations**: /etc/opscenter/opscenterd.conf
- **Tarball installations**: install_location/conf/opscenterd.conf

### Cluster configuration properties

**Cassandra connection properties**

These properties are configured in the cluster-specific configuration file, cluster_name.conf. Replace *cluster_name* with the name of your cluster.

**Note:** After changing properties in this file, restart OpsCenter (page 75) for the changes to take effect.

**[jmx] port**
The JMX (Java Management Extensions) port of your cluster. In Cassandra versions 0.8 and higher, the JMX port is 7199.

[jmx] username
The JMX (Java Management Extensions) username, if you have authentication enabled.

[jmx] password
The JMX (Java Management Extensions) password, if you have authentication enabled.

[jmx] operations_pool_size
The JMX (Java Management Extensions) operations pool size. Defaults to 4.

[cassandra] seed_hosts
A Cassandra seed node is used to determine the ring topology and obtain gossip information about the nodes in the cluster. This should be the same comma-delimited list of seed nodes as the one configured for your DataStax Enterprise cluster by the seeds property in the cassandra.yaml configuration file. The default value is localhost.

[cassandra] api_port
The Thrift remote procedure call port configured for your cluster. Same as the rpc_port property in the cassandra.yaml configuration file. Default is 9160.

[cassandra] cql_port
The CQL port configured for your cluster, the default port is 9042.

[cassandra] conf_location
The location of the cassandra.yaml configuration file. If install_location is specified, but conf_location is not, then conf_location is assumed to be install_location/conf/cassandra.yaml. If conf_location is specified, it must be the absolute path to the Cassandra configuration file on all nodes. These settings are cluster-wide and require that the specified locations be correct for every node.

[cassandra] install_location
The directory in which Cassandra is installed. If install_location is not specified, OpsCenter looks in the package-specific installation locations. For a tarball installation of DataStax Enterprise, the install_location is dse_install_location/resources/cassandra.

[cassandra] log_location
The directory in which DSE logs reside on a node. The default location is /var/log/cassandra. This value will auto-populate the cassandra_log_location property for each agent.

[cassandra] local_dc_pref
When specified OpsCenter will prefer using connections to the specified data center.

[cassandra] used_hosts_per_remote_dc
If using local_dc_pref, this option specified how many remote dc connections may be used as a fallback, the default value is 1.

[cassandra] connect_timeout
Sets the timeout, in seconds, of a thrift connection from OpsCenter to Cassandra. The default value is 6.0.

[cassandra] host_read_timeout_ms
Configuring OpsCenter

The timeout in milliseconds for a host to be considered unresponsive during CQL read operations, such as querying for data. The timeout value should be set higher than the timeout settings used on the Cassandra side (read_request_timeout_in_ms in cassandra.yaml). Default: DataStax Java Driver default value.

[cassandra] bind_interface
  The interface for thrift connections to use.

[cassandra] connection_pool_size
  The number of connections to thrift to build for the connection pool. The default value is 5.

[cassandra] username
  The username used to connect to Cassandra if authentication is enabled.

[cassandra] password
  The password used to connect to Cassandra if authentication is enabled.

[cassandra] send_rpc
  Specifies whether to send the Cassandra RPC IP to DataStax Agents. The default value is True.

[cassandra] ssl_keystore
  The SSL keystore location for OpsCenter to use to connect to Cassandra directly.

[cassandra] ssl_keystore_password
  The SSL keystore password for OpsCenter to use to connect to Cassandra directly.

[cassandra] ssl_truststore
  The SSL truststore location for OpsCenter to use to connect to Cassandra directly.

[cassandra] ssl_truststore_password
  The SSL truststore password for OpsCenter to use to connect to Cassandra directly.

[cassandra] snapshot_threshold
  The number of nodes in the cluster before OpsCenter will switch from running a backup immediately upon receiving a request to scheduling the backup to run after the next full minute plus any time set in snapshot_wait. The default value is 10. If there are less than 10 nodes in the cluster then OpsCenter will tell all nodes to take a snapshot as soon as it is requested. If there are more than 10 nodes, OpsCenter will tell all nodes to take a snapshot at the current time rounded to the next minute, plus snapshot_wait seconds.

[cassandra] rolling_restart_retry_delay
  The number of seconds to wait between retry attempts when connecting to Cassandra after restarting a node. Default: 5.

[cassandra] rolling_restart_retry_attempts
  The maximum number of connection retry attempts after restarting a Cassandra node. Default: 25.

[cassandra] rolling_restart_error_threshold
  A rolling restart will be cancelled if the number of errors during the restart reaches this number. This helps prevent having too many nodes down in your cluster if something catastrophic happens during a rolling restart. Default: 1

[cassandra] schema_refresh_interval
  The time in milliseconds to wait for schema refreshes after issuing a schema change command. Because this option is intended for use with multiple schema
creators that do not apply to opscenterd, the interval should remain at its default of 0.

[cassandra] restart_delay
During a rolling restart, the time in seconds OpsCenter waits after sending the command to stop Cassandra before sending the command to start it again. The default is 30 seconds.

[storage_cassandra] seed_hosts
Configure when using a different cluster for OpsCenter storage. A Cassandra seed node is used to determine the ring topology and obtain gossip information about the nodes in the cluster. This should be the same comma-delimited list of seed nodes as the one configured for your DataStax Enterprise cluster by the seeds property in the cassandra.yaml configuration file.

[storage_cassandra] api_port
Configure when using a different cluster for OpsCenter storage. The Thrift remote procedure call port configured for your cluster. Same as the rpc_port property in the cassandra.yaml configuration file. Default is 9160.

[storage_cassandra] cql_port
Configure when using a different cluster for OpsCenter storage. The CQL port configured for your cluster, the default port is 9042.

[storage_cassandra] local_dc_pref
To reliably determine cluster information, OpsCenter can require a minimum of 2 nodes to connect to. If you specified a single value for seed_hosts, OpsCenter selects a second node in the cluster to fulfill this requirement. You can specify local_dc_pref as a datacenter name to constrain OpsCenter to use that datacenter to pick the second node.

[storage_cassandra] used_hosts_per_remote_dc
Configure when using a different cluster for OpsCenter storage. If using local_dc_pref, this option specified how many remote dc connections may be used as a fallback, the default value is 1.

[storage_cassandra] connect_timeout
Configure when using a different cluster for OpsCenter storage. Sets the timeout, in seconds, of a thrift connection from OpsCenter to Cassandra. The default value is 6.0.

[storage_cassandra] host_read_timeout_ms
The timeout in milliseconds for a host to be considered unresponsive during CQL read operations, such as querying for data. The timeout value should be set higher than the timeout settings used on the Cassandra side (read_request_timeout_in_ms in cassandra.yaml). Default: DataStax Java Driver default value.

[storage_cassandra] bind_interface
Configure when using a different cluster for OpsCenter storage. The interface used for thrift connections.

[storage_cassandra] connection_pool_size
Configure when using a different cluster for OpsCenter storage. The number of connections to thrift to build for the connection pool. The default value is 5.

[storage_cassandra] username
Configure when using a different cluster for OpsCenter storage. The username used to connect to Cassandra if authentication is enabled.
Configuring OpsCenter

**[storage_cassandra] password**
Configure when using a different cluster for OpsCenter storage. The password used to connect to Cassandra if authentication is enabled.

**[storage_cassandra] send_rpc**
Configure when using a different cluster for OpsCenter storage. Specifies whether to send the Cassandra RPC IP to DataStax Agents. The default value is True.

**[storage_cassandra] keyspace**
The name of the keyspace used for OpsCenter data. The keyspace name must be unique for each managed cluster.

**[storage_cassandra] ssl_keystore**
The SSL keystore location for OpsCenter to use to connect to Cassandra directly.

**[storage_cassandra] ssl_keystore_password**
The SSL keystore password for OpsCenter to use to connect to Cassandra directly.

**[storage_cassandra] ssl_truststore**
The SSL truststore location for OpsCenter to use to connect to Cassandra directly.

**[storage_cassandra] ssl_truststore_password**
The SSL truststore password for OpsCenter to use to connect to Cassandra directly.

**[storage_cassandra] schema_refresh_interval**
The time in milliseconds to wait for schema refreshes after issuing a schema change command. Because this option is intended for use with multiple schema creators that do not apply to opscenterd, the interval should remain at its default of 0.

**[collection] basic_info_period**
The frequency, in seconds, to check Cassandra for a Cassandra API update. The default value is 3600 (1 hour).

**[collection] node_poll_period**
This appears to be read but unused

**[collection] nodelist_poll_period**
The interval in seconds OpsCenter waits to poll the nodes in a cluster. The default value is 30.

**[collection] job_poll_period**
The frequency, in seconds, to poll the job tracker. The default value is 5.

**[collection] cf_poll_period**
The frequency, in seconds, to check for a schema update. The default value is 60.

**[collection] push_throttle_period**
The frequency, in seconds, to push node information to the UI. The default value is 60.

**[metric_storage] plugin**
I think this is unused

**[metric_storage] metric_poll_period**
I think this is unused

**[metric_caching] num_data_points_cached**
The number of data points to cache for cluster metrics. The default value is 50.

**[metric_caching] num_metrics_cached**
The number of metrics to cache for cluster metrics. The default value is 1000.

**[agents] api_port**
Configuring OpsCenter

The port used by DataStax Agents for HTTP traffic. The default port is 61621.

[agents] http_timeout
The timeout, in seconds, for an HTTP call to the agent. The default value is 10.

[agents] ssl_keystore
The SSL keystore location for DataStax Agents to use to connect to CQL on the monitored cluster.

[agents] ssl_keystore_password
The SSL keystore password for DataStax Agents to use to connect to CQL on the monitored cluster.

[agents] ssl_truststore
The SSL truststore location for DataStax Agents to use for trusted certs.

[agents] ssl_truststore_password
The SSL truststore password for DataStax Agents to use for trusted certs.

[agents] storage_ssl_keystore
The SSL keystore location for DataStax Agents to use to connect to CQL on the storage cluster.

[agents] storage_ssl_keystore_password
The SSL keystore password for DataStax Agents to use to connect to CQL on the storage cluster.

[agents] storage_ssl_truststore
The SSL truststore location for DataStax Agents to use for trusted certs on the storage cluster.

[agents] storage_ssl_truststore_password
The SSL truststore password for DataStax Agents to use for trusted certs on the storage cluster.

[agents] ec2_metadata_api_host
The IP address to obtain ec2 metadata such as instance id. The default IP address is 169.254.169.254.

[agents] concurrent_agent_requests
The number of concurrent HTTP requests OpsCenter will make to DataStax Agents for most HTTP operations. The default value is 10.

[agents] concurrent_settings_requests
The number of concurrent DataStax Agents OpsCenter will contact upon start-up or when adding a new cluster. The default value is 10.

[agents] concurrent_snapshot_list_requests
The number of concurrent get snapshot info requests. The default value is 1.

[agents] snapshot_wait
See [cassandra] snapshot_threshold

[agents] remote_backup_region
The S3 region region to connect to for remote backup/restore. The default value is us-west-1.

[agents] backup_staging_dir
This path specifies the directory where commit logs archived to by cassandra. From there the commit logs are sent to the configured destinations and the backup_storage_dir.

[agents] backup_storage_dir
This path specifies the directory where commit logs are stored and retained on each node. This property must be set prior to enabling commit log archiving.
[agents] restore_req_update_period
The frequency (in seconds) that a restore reports progress back to OpsCenter. The default value is automatically calculated based on cluster size. To optimize performance, larger clusters have a longer threshold in which restore progress is sent to the UI. Configure this property accordingly for more or less frequent updates during a restore. Note: Lower numbers might impact performance during a restore.

[agents] diagnostic_tarball_download_timeout
The maximum time in seconds to attempt downloading a diagnostic tarball from an agent. This value might need to be increased on slower machines or for multi-instance clusters. Default value: 120.

[cassandra_metrics] ignored_keyspaces
A list of keyspaces to not collect metrics for, separated by commas. The default value is system, system_traces, system_auth, dse_auth, and OpsCenter.

[cassandra_metrics] ignored_column_families
A list of tables to not collect metrics for, separated by commas. Each entry should be of the form "ks.cf". For example: metrics_ignored_column_families = system.NodeInfo, system.Schema, Keyspace1.Standard1

[cassandra_metrics] ignored_solr_cores
A list of solr cores to not collect metrics for, separated by commas. Each entry should be of the form "ks.cf". For example: metrics_ignored_solr_cores = Keyspace1.Standard1, solr.wiki.

[cassandra_metrics] 1min_ttl
Sets the time in seconds to expire 1 minute data points. The default value is 604800 (7 days).

[cassandra_metrics] 5min_ttl
Sets the time in seconds to expire 5 minute data points. The default value is 2419200 (28 days).

[cassandra_metrics] 2hr_ttl
Sets the time in seconds to expire 2 hour data points. The default value is 31536000 (365 days).

[cassandra_metrics] 24hr_ttl
Sets the time to expire 24 hour data points. The default value is 0, or never.

[cassandra_metrics] metrics_enabled
Specifies whether DataStax Agents should collect Cassandra metrics. The default value is True.

[event_storage] enabled
Specifies whether OpsCenter events should be recorded in the event store. The default value is True.

[destinations] active
Specifies the names of destinations to back up to. The destination names should not have any spaces and should be comma-delimited.

[spark] master_http_port
Port at which the Spark master UI is exposed. Default is 7080.

[spark] master_proxy_port
Override for the computed Spark proxy port.

[kerberos] default_service
The default Kerberos service name (Example: cassandra).
[kerberos] default_hostname
   The default Kerberos hostname.
[kerberos] default_client_principal
   The default Kerberos client principal (Example: cassandra@realm).
[kerberos] default_client_user
   The default Kerberos client user.
[kerberos] opscenterd_client_principal
   The OpsCenter client principal in Kerberos (Example: user@realm).
[kerberos] opscenterd_keytab_location
   Full path to the keytab containing keys for opscenterd_client_principal on the
   OpsCenter machine.
[kerberos] agent_client_principal
   The DataStax agent client principal in Kerberos (Example: user@realm).
[kerberos] agent_keytab_location
   Full path to the keytab containing keys for agent_client_principal on the DataStax
   agent machine.
[kerberos] debug
   Whether to output debug messages during Kerberos connection attempts from
   OpsCenter.
[stomp] batch_size
   The number of request updates OpsCenter will push out at once. The default
   value is 100. This is used to avoid overloading the browser.
[stomp] push_interval
   How often OpsCenter will push out updates to requests. The default value is 3
   seconds. This is used to avoid overloading the browser.
[stomp] alert_push_interval
   How often OpsCenter will push out alert updates. The default value is 1 second.
   This is used to avoid overloading the browser.
[bestpractice] results_ttl
   How long, in seconds, OpsCenter will store the results of Best Practice service
   runs. The default value is 2,419,200 seconds, or 4 weeks.
[forecasting] range_multiplier
   The multiplier for the query range needed to produce forecasts. The default
   multiplier is 3.
[forecasting] function
   The function to use for fitting data. Currently, the only option is polyfit.
[forecasting] polyfit_degree
   The degree of polyfit in forecasting.
[forecasting] required_data_percentage
   Minimum percent of past data required to forecast. The default value is 0.5.
[backups] restore_init_throttle
   The number of DataStax Agents on which OpsCenter will concurrently start the
   restore process. The default value is 20.
[backups] restore_sleep
   How long OpsCenter will sleep between batches of starting the restore process,
   set in restore_init_throttle. The default value is 5 seconds.
[backups] failure_threshold
The percentage of the cluster that can fail to respond before a remote destination restore action fails. Default: 50.

[agent_config] Empty_Section
  Empty Section

[dse] Empty_Section
  Empty Section

[repair_service] Empty_Section
  Empty Section

[repair_service] incremental.sqlite.param.synchronous
  Setting for SQLite synchronous pragma for incremental repairs. Default: NORMAL.

[repair_service] incremental.sqlite.param.journal_mode
  Setting for SQLite journal_mode pragma for incremental repairs. Default: WAL.

[repair_service] subrange.sqlite.param.synchronous
  Setting for SQLite synchronous pragma for subrange repairs. Default: NORMAL.

[repair_service] subrange.sqlite.param.journal_mode
  Setting for SQLite journal_mode pragma for subrange repairs. Default: WAL.

[kerberos_hostnames] Empty_Section
  Empty Section

[kerberos_services] Empty_Section
  Empty Section

[kerberos_client_principals] Empty_Section
  Empty Section

[cluster_display_options] Empty_Section
  Empty Section

[labs] graphite_host
  Setting graphite_host enables the forwarding of metrics to a graphite server at the given address. Leaving the graphite_host blank disables forwarding metrics to the graphite server.

[labs] graphite_port
  Port for graphite’s plaintext protocol.

[labs] graphite_prefix
  A prefix to insert metrics under.

[labs] bypass_dse_metrics_storage
  Enable or disable storing metrics in a monitored or a separate storage DSE cluster. Metrics are stored in a DSE monitored or storage cluster by default. Default: False.

[labs] use_s3_cli
  Enable using the AWS CLI instead of the AWS SDK when bulk loading backups to Amazon S3 locations. Default: False.

[kerberos_hostnames] 192.168.1.101
  Per-node specification for the Kerberos hostname of the service (DSE). A list of IP and hostname pairs. For example 192.168.1.101 = cassandra01.example.com.

[kerberos_services] 192.168.1.101
  Per-node specification of the Kerberos service name. A list of IP, hostname pairs. For example 192.168.1.101 = cassandra.

[kerberos_client_principals] 192.168.1.102
Per-client specification of the Kerberos principal to use. A list of IP, hostname pairs. For example 192.168.1.102 = opscenter-agent01@EXAMPLE.COM.

**[cluster_display_options] display_name**

Display name used by OpsCenter to signify this cluster.

**Metrics Collection Properties**

**[cassandra_metrics] ignored_keyspaces**

A list of keyspaces to not collect metrics for, separated by commas. The default value is system, system_traces, system_auth, dse_auth, and OpsCenter.

**[cassandra_metrics] ignored_column_families**

A list of tables to not collect metrics for, separated by commas. Each entry should be of the form "ks.cf". For example: metrics_ignored_column_families = `system.NodeInfo, system.Schema, Keyspace1.Standard1`

**[cassandra_metrics] ignored_solr_cores**

A list of solr cores to not collect metrics for, separated by commas. Each entry should be of the form "ks.cf". For example: metrics_ignored_solr_cores = Keyspace1.Standard1, solr.wiki.

These properties set the expiration time for data stored in the OpsCenter keyspace. Each time period for rolling up data points into summary views has a separate expiration threshold, or time-to-live (ttl) value expressed in seconds. By default, shorter time periods have lower values that result in more efficient expiration and compaction of the relatively larger volumes of data. Uncomment these properties to change the default expiration periods for performance data. Properties and default values are:

**[cassandra_metrics] 1min_ttl**

Sets the time in seconds to expire 1 minute data points. The default value is 604800 (7 days).

**[cassandra_metrics] 5min_ttl**

Sets the time in seconds to expire 5 minute data points. The default value is 2419200 (28 days).

**[cassandra_metrics] 2hr_ttl**

Sets the time in seconds to expire 2 hour data points. The default value is 31536000 (365 days).

**[cassandra_metrics] 24hr_ttl**

Sets the time to expire 24 hour data points. The default value is 0, or never.

**Related information:**

Excluding keyspaces and tables from data collection [By default, OpsCenter does not collect performance data for its own keyspace or the Cassandra system keyspace. You can manually add any other keyspaces or tables that you do not want to monitor in the [cassandra_metrics] section of the cluster_name.conf configuration file.] (page 151)

**DataStax Agent configuration**

The address.yaml configuration file

The address.yaml file contains configuration options for the DataStax Agent.
Most of these properties can be set in the [agent_config] section of cluster_name.conf on the opscenterd machine, which automatically propagates the properties to all agents. Some properties or some cases might require setting these properties directly in address.yaml on applicable agents. When manually installing agents (page 63), stomp_interface is the only property in most environments that needs to be explicitly configured. When automatically installing agents (page 59), stomp_interface is configured for you.

For more information about viewing agent status and troubleshooting agent issues, see Agents View (page 230).

Configuration options

**use_ssl**
Whether or not to use SSL communication between the agent and opscenterd. Affects both the STOMP connection and agent HTTP server. Corresponds to [agents].use_ssl in opscenterd.conf. Setting this option to true turns on SSL connections. Example: `use_ssl: true`

**stomp_port**
The stomp_port used by opscenterd. Example: `stomp_port: 61620`

**stomp_interface**
Reachable IP address of the opscenterd machine. The connection made will be on stomp_port. Example: `stomp_interface: 127.0.0.1`

**local_interface**
The IP used to identify the node. If broadcast_address is set in cassandra.yaml, this should be the same as that; otherwise, it is typically the same as listen_address in cassandra.yaml. A good check is to confirm that this address is the same as the address that nodetool ring outputs. Example: `local_interface: 172.10.0.2`

**agent_rpc_interface**
The IP that the agent HTTP server listens on. In a multiple region deployment, this is typically a private IP. Default: Matches rpc_interface from cassandra.yaml. Example: `agent_rpc_interface: 172.10.0.2`

**agent_rpc_broadcast_address**
The IP that the central OpsCenter process uses to connect to the DataStax agent. Default: First available resolvable address in this order: broadcast_rpc_address, rpc_address, and listen_address from cassandra.yaml. Example: `agent_rpc_broadcast_address: 172.10.0.2`

**swagger_enabled**
Enables or disables the swagger UI for the agent API. Example: `swagger_enabled: true`

**opscenter_ssl_keystore**
On target nodes where DataStax Agents are running, the path to the SSL keystore file that the Agents use to connect to opscenterd. Example: `opscenter_ssl_keystore: /usr/share/opscenter/ssl/agentKeystore`

**opscenter_ssl_keystore_password**
The SSL keystore password that the agents use to connect to opscenterd. Example: `opscenter_ssl_keystore_password: keystore-pass [This field may be encrypted for additional security.]`

**opscenter_ssl_truststore**
**Configuring OpsCenter**

The path to the truststore file that the agents use to connect to opscenterd.
Example: `opscenter_ssl_truststore: /usr/share/opscenter/ssl/trustStore`

**opscenter_ssl_truststore_password**
The SSL truststore password that the agents use to connect to opscenterd.
Default: Uses the keystore password if an SSL truststore password is not specified.
Example: `opscenter_ssl_truststore_password: trust-pass` [This field may be encrypted for additional security.]

**poll_period**
The length of time, specified in seconds, between attempts to poll metrics.
Example: `poll_period: 60`

**disk_usage_update_period**
The length of time, in seconds, to wait between attempts to poll the disk for usage.
Example: `disk_usage_update_period: 60`

**rollup_rate**
Maximum number of metrics that can be saved to Cassandra over the
[rollup_rate_unit] period of time. This should be at least (⌈[#tables] * 40⌉ + 200) per min
Default: 200 (so 200/sec with default rollup_rate_unit) Example: `rollup_rate: 200`

**rollup_rate_unit**
Unit of time for rollup_rate. Choose from microsecond, millisecond, second, minute,
hour, day, or month. Default: second Example: `rollup_rate_unit: second`

**bypass_dse_metrics_storage**
Enable or disable storing metrics in a separate storage DataStax Enterprise (DSE)
cluster. Metrics are stored in an OpsCenter keyspace on the same DSE cluster
being monitored by default. Note: storing metrics on a separate DSE cluster will
place the entire OpsCenter keyspace on that cluster. Default: false. Example:
`bypass_dse_metrics_storage: false`

**jmx_host**
Host used to connect to local JMX server. The default setting is localhost. This
information will be sent by opscenterd for convenience, but can be configured
locally as needed. Example: `jmx_host: 127.0.0.1`

**jmx_port**
Port used to connect to local JMX server. The default setting is 7199. This
information will be sent by opscenterd for convenience, but can be configured
locally as needed. Example: `jmx_port: 7199`

**jmx_user**
The username used to connect to the local JMX server. Example: `jmx_user: jmx-
username`

**jmx_pass**
The password used to connect to the local JMX server. Example: `jmx_pass: jmx-
password` [This field may be encrypted for additional security.]

**jmx_queue_poll_timeout**
The number of seconds to wait for an available JMX connection before timing out.
Default: 10. Example: `jmx_queue_poll_timeout: 10`

**status_reporting_interval**
The length of time, in seconds, between sending agent health information.
Example: `status_reporting_interval: 20`
Configuring OpsCenter

disk_wait
The amount of time in milliseconds to wait for disk operations when collecting metrics. Default: 5000. Example: disk_wait: 5000

e2_metadata_api_host
The e2 metadata api host, used to determine information about this node, if it is on e2. Example: e2_metadata_api_host: 169.254.169.254

metrics_enabled
Whether or not to collect and store metrics for the local node. Setting this option to false turns off metrics collection. Default: true. Example: metrics_enabled: true

jmx_metrics_threadpool_size
The size of the threadpool used for collecting metrics over JMX. Example:
jmx_metrics_threadpool_size: 6

metrics_ignored_keyspaces
A comma-separated list of keyspaces ignored by metrics collection. Example:
metrics_ignored_keyspaces: ks1, ks2, ks3

metrics_ignored_column_families
A comma-separated list of tables (formerly referred to as column families) ignored by metrics collection. Example: metrics_ignored_column_families: ks1.cf1, ks1.cf2, ks2.cf1

metrics_ignored_solr_cores
A comma separated list of solr cores that will be ignored by metric collection. Example: metrics_ignored_solr_cores: ks1.cf1, ks1.cf2, ks2.cf1

hosts
The DataStax Enterprise node or nodes responsible for storing OpsCenter data. By default, this will be the local node, but may be configured to store data on a separate cluster (page 154). The hosts option accepts an array of strings specifying the IP addresses of the node or nodes. For example, ["1.2.3.4"] or ["1.2.3.4", "1.2.3.5"]. Example: hosts: ["127.0.0.1"]

cassandra_port
Port used to connect to the storage cassandra node. The native transport port. Example: cassandra_port: 9042

thrift_port
Port used to connect to storage thrift server. The default setting is 9160. This information will be sent by opscenterd for convenience, but can be configured locally as needed. Example: thrift_port: 9160

cassandra_user
The Username used to connect to storage cassandra when authentication is enabled. Example: cassandra_user: cassandra

cassandra_pass
The password used to connect to storage cassandra when authentication is enabled. Example: cassandra_pass: cassandra [This field may be encrypted for additional security.]

max_reconnect_time
The maximum time in ms that the agent will wait between cassandra reconnect attempts. Example: max_reconnect_time: 15000

max_pending_repairs
The maximum number of repairs that may be pending, exceeding this number blocks new repairs. Example: max_pending_repairs: 5
ssl_keystore
The SSL keystore location for the storage cluster that agents use to connect to CQL. Example: ssl_keystore: /etc/dse/conf/.keystore

ssl_keystore_password
The SSL keystore password for the storage cluster that agents use to connect to CQL. Example: ssl_keystore_password: keystore-pass [This field may be encrypted for additional security.]

ssl_truststore
The SSL truststore location for the storage cluster that agents use to connect to CQL. Example: ssl_truststore: /etc/dse/conf/.truststore

ssl_truststore_password
The SSL truststore password for the storage cluster that agents use to connect to CQL. Example: ssl_truststore_password: truststore-pass [This field may be encrypted for additional security.]

monitored_cassandra_port
Port used to connect to the monitored cassandra node. The native transport port. Example: monitored_cassandra_port: 9042

monitored_thrift_port
Port used to connect to monitored thrift server. The default setting is 9160. This information will be sent by opscenterd for convenience, but can be configured locally as needed. Example: monitored_thrift_port: 9160

monitored_cassandra_user
The Username used to connect to monitored cassandra when authentication is enabled. Example: monitored_cassandra_user: cassandra

monitored_cassandra_pass
The password used to connect to monitored cassandra when authentication is enabled. Example: monitored_cassandra_pass: cassandra-pass [This field may be encrypted for additional security.]

monitored_ssl_keystore
The SSL keystore location for the monitored cluster that agents use to connect to CQL. Example: monitored_ssl_keystore: /etc/dse/conf/.keystore

monitored_ssl_keystore_password
The SSL keystore password for the monitored cluster that agents use to connect to CQL. Example: monitored_ssl_keystore_password: keystore-pass [This field may be encrypted for additional security.]

monitored_ssl_truststore
The SSL truststore location for the monitored cluster that agents use to connect to CQL. Example: monitored_ssl_truststore: /etc/dse/conf/.truststore

monitored_ssl_truststore_password
The SSL truststore password for the monitored cluster that agents use to connect to CQL. Example: monitored_ssl_truststore_password: truststore-pass [This field may be encrypted for additional security.]

kerberos_service
The Kerberos service name to use when using Kerberos authentication within DSE. Example: kerberos_service: cassandra-kerberos

kerberos_keytab_location
The Kerberos keytab location when using Kerberos authentication within DSE. Example: kerberos_keytab_location: /path/to/keytab.keytab
Configuring OpsCenter

**kerberos_client_principal**
The Kerberos client principal to use when using Kerberos authentication within DSE. Example: \textit{kerberos_client_principal: cassandra@hostname}

**storage_keyspace**
The keyspace that the agent will use to store data. Example: \textit{storage_keyspace: OpsCenter}

**alias**
Provides an alias for the agent to use when sending node details to OpsCenter. The alias is useful when the agent is unable to get the localhost name from InetAddress.getLocalHost(). Example: \textit{alias: MyNodeOne}

**storage_dse_connection_timeout**
The maximum time in seconds that the agent waits while attempting to connect to the DSE cluster. Default: 30. Example: \textit{storage_dse_connection_timeout: 30}

**storage_dse_host_read_timeout**
The maximum time in milliseconds that the agent waits for a storage node to return a response from a read request before considering said node unresponsive. Should be set higher than \textit{read_request_timeout_in_ms} in cassandra.yaml. Example: \textit{storage_dse_host_read_timeout: 10000}

**monitored_dse_connection_timeout**
The maximum time in seconds that the agent waits while attempting to connect to the DSE cluster. Default: 30. Example: \textit{monitored_dse_connection_timeout: 30}

**monitored_dse_host_read_timeout**
The maximum time in milliseconds that the agent waits for a monitored node to return a response from a read request before considering said node unresponsive. Should be set higher than \textit{read_request_timeout_in_ms} in cassandra.yaml. Example: \textit{monitored_dse_host_read_timeout: 10000}

**cassandra_install_location**
The base directory where DataStax Enterprise or Cassandra is installed. When not set, the agent attempts to auto-detect the location but cannot do so in all cases. Example: \textit{cassandra_install_location: /usr/share/dse}

**cassandra_log_location**
The directory in which DSE logs reside. This is only used for the diagnostics tarball, and should only be set if these logs are in a location other than the default. Example: \textit{cassandra_log_location: /var/log/cassandra}

**cassandra_binary_location**
The location of Cassandra's binaries' directory (cqlsh, nodetool, and sstableloader). When not set, the agent attempts to auto-detect the location. Example: \textit{cassandra_binary_location: /usr/bin}

**cassandra_conf_location**
The location of Cassandra's configuration files' directory (cassandra.yaml, cassandra-env.sh). When not set, the agent attempts to auto-detect the location. Example: \textit{cassandra_conf_location: /etc/dse/cassandra}

**dse_env_location**
The location of directory that holds dse-env.sh. When not set, the agent attempts to auto-detect the location. Example: \textit{dse_env_location: /etc/dse}

**dse_binary_location**
The location of directory that holds dsetool. When not set, the agent attempts to auto-detect the location. Example: \textit{dse_binary_location: /usr/bin}
Configuring OpsCenter

**dse_conf_location**
The location of directory that holds dse.yaml. When not set, the agent attempts to auto-detect the location. Example: `dse_conf_location: /etc/dse`

**spark_conf_location**
The location of directory that holds spark-env.sh. When not set, the agent attempts to auto-detect the location. Example: `spark_conf_location: /etc/dse/spark`

**spark_log_location**
The location of directory that holds spark logs. When not set, the agent attempts to auto-detect the location. Example: `spark_log_location: /var/log/spark`

**solr_log_location**
The location of directory that holds solr logs. When not set, the agent attempts to auto-detect the location. Example: `solr_log_location: /var/log/cassandra`

**cassandra_rpc_interface**
When unspecified, the agent will attempt to determine cassandra rpc_address by reading cassandra.yaml for rpc_address. When specified, this agent lookup is skipped and the specified value is used instead. Example: `cassandra_rpc_interface: 172.10.0.2`

**api_port**
The port used for the http api endpoint. Example: `api_port: 61621`

**runs_sudo**
Sets whether the DataStax Agent will be run using sudo or not. Setting this option to false means the agent will not use sudo, and the agent user will not run using elevated privileges. Setting this option to true means the agent will run using sudo, and elevated privileges. Default is true. Example: `runs_sudo: true`

**s3_proxy_host**
The optional proxy host the client will connect through. Example: `s3_proxy_host: localhost`

**s3_proxy_port**
The optional proxy port the client will connect through. Example: `s3_proxy_port: 80`

**restore_req_update_period**
The frequency in seconds with which status updates are sent to opscenterd during Restore operations in the Backup Service. Default: 60. Example: `restore_req_update_period: 60`

**backup_staging_dir**
The directory used for staging commit logs to be backed up. The default location is `/var/lib/datastax-agent/commitlogs/`. Example: `backup_staging_dir: /var/lib/datastax-agent/commitlogs/`

**backup_storage_dir**
The directory used for storing backup files. The default location is `/var/lib/datastax-agent/backups/`. Example: `backup_storage_dir: /var/lib/datastax-agent/backups/`

**tmp_dir**
The location of the temp directory used as the staging directory for Backup Service backups and for Repair Service persistence files. The default location is `/var/lib/datastax-agent/tmp/`. Example: `tmp_dir: /var/lib/datastax-agent/tmp/`

**remote_backup_retries**
Configuring OpsCenter

The number of attempts to make when file download fails during a restore. Default: 3. Example: `remote_backup_retries: 3`

**remote_backup_timeout**
The timeout in milliseconds for the connection used to push backups to remote destinations. Default: 1000. Example: `remote_backup_timeout: 1000`

**use_s3_cli**
Labs feature. Enable using the AWS CLI instead of the AWS SDK when bulk loading backups to Amazon S3 locations. Default: false. Example: `use_s3_cli: true`

**remote_verify_initial_delay**
Initial delay in milliseconds to wait before checking if a file was successfully uploaded during a backup operation. This configuration option works in conjunction with the `remote_verify_max` option to distinguish between broken versus tardy backups when cleaning up SSTables. The `remote_verify_initial_delay` value doubles each time a file transfer validation failure occurs until the value exceeds the `remote_verify_max` value. Default: 1000 (1 second). Example: `remote_verify_initial_delay: 1000`

**remote_verify_max**
The maximum time period to wait after a file upload completed but is still unreadable from the remote destination. When this delay is exceeded, the transfer is considered failed. This configuration option works in conjunction with the `remote_verify_initial_delay` option to distinguish between broken versus tardy backups when cleaning up SSTables. Default: 30000 (30 seconds). Example: `remote_verify_max: 300000`

**restore_on_transfer_failure**
When set to true, a failed file transfer from the remote destination will not halt the restore process. A future restore attempt uses any successfully transferred files. Default: false. Example: `restore_on_transfer_failure: false`

**remote_backup_region**
The AWS region to use for remote backup transfers. Default: us-west-1. Example: `remote_backup_region: us-west-1`

**max_file_transfer_attempts**
The maximum number of attempts to upload a file or create a remote destination. Default: 3. Example: `max_file_transfer_attempts: 30`

**sstableloader_max_heap_size**
The maximum heap size used by the sstableloader during restore operations. Only supported with DSE 4.8.4+. Default: 256M. Example: `sstableloader_max_heap_size: 256M`

**trace_delay**
The time in milliseconds to wait between issuing a query to trace and fetching trace events in the Performance Service Slow Query panel. Default: 300. Example: `trace_delay: 300`

**support_shell_timeout**
The number of seconds to wait for a shell process such as nodetool to run before timing out. This setting is only used for generating a diagnostic tarball. Default: 30. Example: `support_shell_timeout: 30`

**graphite_host**
Setting graphite_host enables the forwarding of metrics to a graphite server at the given address. Leaving the graphite_host blank disables forwarding metrics to the graphite server. Example: `graphite_host: graphite.myhost.com`

**graphite_port**
Port for graphite's plaintext protocol. Example: `graphite_port: 2003`

**graphite_prefix**
A prefix to insert metrics under. Example: `graphite_prefix: opscenter`

**slow_query_past**
How far into the past in milliseconds to look for slow queries. Default: 3600000 (1,000 hours). Example: `slow_query_past: 3600000`

**slow_query_refresh**
Time in seconds between slow query refreshes. Default: 5. Example: `slow_query_refresh: 5`

**slow_query_fetch_size**
The limit to how many slow queries are fetched. Default: 500. Example: `slow_query_fetch_size: 500`

**slow_query_ignore**
A list of keyspaces that the performance service slow query log will ignore. Default: ["OpsCenter" "dse_perf"] Example: `slow_query_ignore: ["OpsCenter" "dse_perf"]`

**config_encryption_active**
Specifies whether opscenter should attempt to decrypt sensitive config values. Default: False

**config_encryption_key_name**
Filename to use for the encryption key. If a custom name is not specified, opsc_system_key is used by default. Example: `config_encryption_key_name: opsc_system_key`

**config_encryption_key_path**
Path where the encryption key should be located. If unspecified, the directory of address.yaml is used by default. Example: `config_encryption_key_path: /var/lib/datastax-agent/conf/`

**running-request-cache-size**
Size of running requests cache Example: `running-request-cache-size: 500`

**finished-request-cache-size**
Size of finished requests cache Example: `finished-request-cache-size: 100`

**tcp_response_timeout**
The tcp response timeout used for JMX specified in milliseconds. This value may need to be set very high in order for some operations to complete on nodes with large amounts of data. 0 for no timeout. Default: 240000 Example: `tcp_response_timeout: 120000`

**pong_timeout_ms**
The number of milliseconds to wait for a pong reply from opscenterd over stomp before timing out the ping. Example: `pong_timeout_ms: 5000`

**destination_pretest_timeout**
Configuring OpsCenter

The maximum amount of time in seconds to verify a destination can be written to and read from. Default: 60. Example: `destination_pretest_timeout: 60`

Configuring DataStax Agents for Multi-Instance Nodes

Configure DataStax Agents for DSE Multi-Instance (dense) Nodes.

`address.yaml`

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

- **Package installations**: `/var/lib/datastax-agent/conf/address.yaml`
- **Tarball installations**: `install_location/conf/address.yaml`

Prerequisites:

Multi-Instance requirements for DataStax Agents:

- DSE Multi-Instance requires one agent process running for each DSE instance on a machine.
- DataStax Agents require a manual tarball agent installation *(page 64)* for multi-instance.
- DataStax Agents require manual configuration in `address.yaml` as outlined in this procedure for Multi-Instance.

1. Open `address.yaml` for editing.

2. Set the following backup options for each agent:

   - **tmp_dir**
     The location of the temp directory used as the staging directory for Backup Service backups and for Repair Service persistence files. The default location is `/var/lib/datastax-agent/tmp`. Example: `tmp_dir: /var/lib/datastax-agent/tmp/

   - **backup_staging_dir**
     The directory used for staging commit logs to be backed up. The default location is `/var/lib/datastax-agent/commitlogs/`. Example: `backup_staging_dir: /var/lib/datastax-agent/commitlogs/`

3. Configure the following option so that the `opscenterd` (OpsCenter daemon) can send requests to the agent:

   - **agent_rpc_interface**
     The IP that the agent HTTP server listens on. In a multiple region deployment, this is typically a private IP. Default: Matches `rpc_interface` from `cassandra.yaml`. Example: `agent_rpc_interface: 172.10.0.2`

     **Note**: The `agent_rpc_interface` must be unique for each agent.
**Configuring OpsCenter**

4. Configure the following option to connect to a Cassandra instance over JMX. The JMX port must always be configured for DSE Multi-Instance.

   **jmx_port**
   
   Port used to connect to local JMX server. The default setting is 7199. This information will be sent by opscenterd for convenience, but can be configured locally as needed. Example: \texttt{jmx\_port: 7199}

   **Note:** The JMX port must be unique for each instance of DSE.

5. Depending on the configuration of your environment, configure the following option for connecting to OpsCenter:

   **stomp\_interface**
   
   Reachable IP address of the opscenterd machine. The connection made will be on stomp\_port. Example: \texttt{stomp\_interface: 127.0.0.1}

6. **Restart the DataStax Agents** *(page 76).*

**OpsCenter logback.xml configuration**

Starting with OpsCenter 6.0, the OpsCenter daemon process uses the Java logback library. All logging configuration is now done in the logback.xml file. This section highlights some of the most common configuration properties. Refer to the logback configuration guides for additional details.

   **Note:** Restart OpsCenter *(page 75)* for logging changes to take effect.

**logback.xml**

The location of the logback.xml file depends on the type of installation:

- Package installations: /etc/opscenter/logback.xml
- Tarball installations: install_location/conf/logback.xml

**opscenterd.log**

The location of the opscenterd.log file depends on the type of installation:

- Package installations: /var/log/opscenter/opscenterd.log
- Tarball installations: install_location/log/opscenterd.log

**opscenterd\_log appender**

The main logback appender used by default for OpsCenter is the opscenterd\_log appender that controls appending log messages to the application log file. The base log <file> name is opscenterd.log. Below is a sample default block.
Configuring OpsCenter

```xml
<appender name="opscenterd_log"
  class="ch.qos.logback.core.rolling.RollingFileAppender">
  <file>./log/opscenterd.log</file>
  <encoder>
    <pattern>%date{ISO8601, UTC} [%X{cluster_id:-opscenterd}] %5level: %msg \(%thread\) %n%exception{20}</pattern>
  </encoder>
  <rollingPolicy
    class="ch.qos.logback.core.rolling.FixedWindowRollingPolicy">
    <fileNamePattern>./log/opscenterd.%i.log</fileNamePattern>
    <minIndex>1</minIndex>
    <maxIndex>10</maxIndex>
  </rollingPolicy>
  <triggeringPolicy
    class="ch.qos.logback.core.rolling.SizeBasedTriggeringPolicy">
    <maxFileSize>10MB</maxFileSize>
  </triggeringPolicy>
</appender>
```

Some common configuration parameters include:

- `<file>`
  This is the base log file name that the current log messages are logged to.

- `<fileNamePattern>`
  This is the pattern logback follows when rolling a log file over. By default, the log file count is inserted into the filename (with the `%i` parameter).

- `<minIndex>`
  This is the integer that logback uses to start counting log files with. The default value is 1.

- `<maxIndex>`
  The integer that logback uses as the max number of log files to keep. If a new log is needed and there are already maxIndex files, logback deletes the minIndex logfile and rolls the remaining log files. The default value is 10.

- `<maxFileSize>`
  The log file size that causes logback to rotate the log files. By default, OpsCenter uses 10MB as a limit. Valid values can be found at [https://logback.qos.ch/manual/appenders.html#SizeBasedTriggeringPolicy](https://logback.qos.ch/manual/appenders.html#SizeBasedTriggeringPolicy).

  **Note:** The same basic properties also apply to the `http_log` appender, which is the log destination for HTTP requests, and the `repair_log` appender, which is the log destination for repair service logs.

### Changing OpsCenter Daemon Logging Level

Changing the `level` parameter in the `<root>` XML element only changes the logging level for the OpsCenter daemon and the cluster logging. Below is an example `<root>` block.

```xml
<root level="INFO">
  <appender-ref ref="opscenterd_log"/>
  <appender-ref ref="STDOUT"/>
</root>
```
Valid values for level include DEBUG, INFO, WARN, ERROR or OFF. By default, OpsCenter is configured to log at the INFO level. Setting the logging level to DEBUG or TRACE increases the verbosity of the log messages for troubleshooting.

In addition to the `<root>` logging level, there are also a set of granular logging-level configurations defined using the `<logger>` directive.

```
<logger name="com.datastax.driver" level="WARN" additivity="false"/>
<logger name="com.datastax.driver.core.FrameCompressor" level="ERROR"/>
<logger name="org.python" level="ERROR"/>
<logger name="org.apache.http" level="ERROR"/>
<logger name="com.mchange" level="ERROR"/>
<logger name="lcm" level="INFO"/>
<logger name="lcm.database.migration" level="WARN"/>
```

DataStax recommends leaving these levels set at their default points. Setting the log levels to a more verbose level might impact the performance of OpsCenter while generating unnecessary output.

Changing Console Log Level, HTTP Request Log Level, Repair Service Log Level, and Security Log Level

Based on some limitations with logback configurations, changing the log level of the HTTP request logs or the repair service logs requires changing the filter inside of the `http_log` and `repair_log` respectively.

```
<filter class="ch.qos.logback.classic.filter.ThresholdFilter">
  <level>INFO</level>
</filter>
```

The valid values for this level are the same as the valid values listed in the section above.

**Customize scripts for starting and stopping DataStax Enterprise**

OpsCenter allows starting and stopping the DataStax Enterprise process on each node in a visual manner. The agent attempts to automatically determine the best way to do this but cannot do so in all cases. You can customize the startup or shutdown of a node using the `start-cassandra` and `stop-cassandra` scripts located in `/usr/share/datastax-agent/bin` (package installs) or `install_location/bin` (tarball installs).

1. Rename the example script in `/usr/share/datastax-agent/bin` (package installs) or `install_location/bin` (tarball installs) to remove the `.example` extension.
   - `start-cassandra.example: example startup script`
Configuring OpsCenter

- stop-cassandra.example: example shutdown script

```bash
$ cd /usr/share/datastax-agent/bin
$ mv start-cassandra.example start-cassandra
```

2. Edit the script to customize the behavior. The script should return an exit code of 0 when successful, and a non-zero value if it fails.

3. Make the script executable.

```bash
$ chmod 755 start-cassandra
```

Related information:
Node management operations [Use OpsCenter to run operations on nodes in an easy to use, visual way that takes the guesswork out of properly managing nodes in a cluster.] (page 235)

Example configuration scenarios

**Configuring for multiple regions**

**address.yaml**

The location of the `address.yaml` file depends on the type of installation:
- Package installations: `/var/lib/datastax-agent/conf/address.yaml`
- Tarball installations: `install_location/conf/address.yaml`

OpsCenter can operate in multiple regions or IP forwarding deployments. Use the following approach for deployments where a public IP forwards to a private IP on the agent, but that machine is not aware of (that is, can't bind to) the public IP.

To configure DataStax agents for multiple regions or IP forwarding:

1. Open the `address.yaml` file for editing.

2. Add the following options to the `address.yaml` file.
   - Setting `local_interface` is optional.
     - `local_interface`
       The IP used to identify the node. If `broadcast_address` is set in `cassandra.yaml`, this should be the same as that; otherwise, it is typically the same as `listen_address` in `cassandra.yaml`. A good check is to confirm that this address is the same as the address that nodetool ring outputs.
       **Example:** `local_interface: 172.10.0.2`

   - `agent_rpc_interface`
Configuring OpsCenter

The IP that the agent HTTP server listens on. In a multiple region deployment, this is typically a private IP. Default: Matches rpc_interface from cassandra.yaml. Example: `agent_rpc_interface: 172.10.0.2`

- **agent_rpc_broadcast_address**
  The IP that the central OpsCenter process uses to connect to the DataStax agent. Default: First available resolvable address in this order: broadcast_rpc_address, rpc_address, and listen_address from cassandra.yaml. Example: `agent_rpc_broadcast_address: 172.10.0.2`

3. Repeat the above steps for each node.

Here is the configuration for a three node cluster that spans two regions:

<table>
<thead>
<tr>
<th>Region: us-west</th>
<th>Availability Zone: us-west-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpsCenter host</td>
<td></td>
</tr>
<tr>
<td>public IP: 198.51.100.5</td>
<td></td>
</tr>
<tr>
<td>private IP: 10.11.12.10</td>
<td></td>
</tr>
<tr>
<td>Node1</td>
<td></td>
</tr>
<tr>
<td>public IP: 198.51.100.1</td>
<td></td>
</tr>
<tr>
<td>private IP: 10.11.12.1</td>
<td></td>
</tr>
<tr>
<td>Cassandra (cassandra.yaml)</td>
<td></td>
</tr>
<tr>
<td>broadcast_address: 198.51.100.1</td>
<td></td>
</tr>
<tr>
<td>listen_address: 10.11.12.1</td>
<td></td>
</tr>
<tr>
<td>Agent (address.yaml)</td>
<td></td>
</tr>
<tr>
<td>local_interface: 198.51.100.1</td>
<td></td>
</tr>
<tr>
<td>agent_rpc_interface: 10.11.12.1</td>
<td></td>
</tr>
<tr>
<td>agent_rpc_broadcast_address: 198.51.100.1</td>
<td></td>
</tr>
<tr>
<td>stomp_interface: 198.51.100.5</td>
<td></td>
</tr>
</tbody>
</table>

| Node2            |                               |
| public IP: 198.51.100.23 |  
| private IP: 10.11.12.15 |  
| Cassandra (cassandra.yaml) |  
| broadcast_address: 198.51.100.23 |  
| listen_address: 10.11.12.15 |  
| Agent (address.yaml) |  
| local_interface: 198.51.100.23 |  
| agent_rpc_interface: 10.11.12.15 |  
| agent_rpc_broadcast_address: 198.51.100.23 |  
| stomp_interface: 198.51.100.5 |  

<table>
<thead>
<tr>
<th>Region: us-east</th>
<th>Availability Zone: us-east-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node1</td>
<td></td>
</tr>
</tbody>
</table>
Configuring for very large clusters

cluster_name.conf

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

- Package installations: /etc/opscenter/clusters/cluster_name.conf
- Tarball installations: install_location/conf/clusters/cluster_name.conf

opscenterd.conf

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

- Package installations: /etc/opscenter/opscenterd.conf
- Tarball installations: install_location/conf/opscenterd.conf

OpsCenter can manage very large clusters up to 1000 nodes.

**Note:** Lifecycle Manager can provision and manage up to 300 nodes per cluster within its UI. See How many nodes can Lifecycle Manager support when creating DataStax Enterprise clusters? (page 549) for more details.

When working with very large clusters, the performance of OpsCenter decreases with the default settings. To improve performance, adjust the cluster settings to increase the time period between polls of a cluster's nodes and token lists.

After adding a very large cluster to OpsCenter, change the following default settings:

1. Open cluster_name.conf for editing.

   a. Increase the node list poll period to 30 minutes by setting the nodelist_poll_period option to 1800 under [collection]:

   ```
   [collection]
   nodelist_poll_period = 1800
   ```
b. If an agent is overloaded, increase the default `http_timeout` if necessary:

```
[agents] http_timeout
    The timeout, in seconds, for an HTTP call to the agent. The default value is 10.
```

```
[agents]
    http_timeout = 20
```

2. Open `opscenterd.conf` for editing and adjust the following settings:

```
[agents] not_seen_threshold
    The maximum time in seconds since the last agent status about a specific connection, such as stamp, was sent before that agent connection is considered down. This threshold also affects how long OpsCenter waits before marking node health as unknown. Default value: 180 seconds.

[agents] http_poll_period
    The frequency in seconds between attempts to poll agent http health. Default value: 60 seconds.

[ui] default_api_timeout
    The default timeout value in seconds for an API call from the OpsCenter UI to the OpsCenter API. The default value is 10. Some API calls require a timeout longer than 10 seconds. In those cases, the API call timeouts are scaled relative to the default_api_timeout (for example, 6 * default_api_timeout). Changing the default_api_timeout affects those timeouts accordingly.
```

```
[agents]
    not_seen_threshold = 620
    http_poll_period = 500

[ui]
    default_api_timeout = 60
```

3. **Restart OpsCenter (page 75).**
Using OpsCenter

OpsCenter workspace overview

The OpsCenter workspace overview introduces the major areas of functionality available from the menus in the OpsCenter UI.

OpsCenter Main menu

The main menu and Cluster Actions menu are available at the top of the UI in every functional area within OpsCenter:

Access the following OpsCenter features from the main menu:

- **New Cluster** - Create a new cluster (page 556) in Lifecycle Manager or manage an existing (page 274) DataStax Enterprise cluster within OpsCenter.
- **Alerts** - Configure alert thresholds for a number of cluster-wide, table, and operating system metrics.
- **Settings** - Access to editing Cluster Connections and User Roles:
  - **Cluster Connections** - Modify cluster settings (page 93) or remove the cluster (page 280) from OpsCenter.
  - **Users & Roles** - Manage users (page 106) with role-based authentication.
- **Help** - Information resources for OpsCenter:
  - **Help Center** - opens the current DSE OpsCenter documentation.
  - **Feedback** opens a form for sending feedback to the OpsCenter team.
  - **Report** generates an HTML report (page 323) with information on the managed clusters.
  - **Diagnostics** downloads a tarball that contains diagnostic information (page 286) about the nodes in a cluster.
- **Username** - If authentication (page 105) is enabled, the username for the currently logged in user is displayed. Click your username to change your password or to log out of OpsCenter.

Navigation menus

The left navigation pane provides a link to Lifecycle Manager, the OpsCenter Overview, and lists the clusters monitored in the OpsCenter instance. The cluster navigation menu provides
access to the functional areas within the OpsCenter UI; such as Nodes, Activities, Data, and Services.

The OpsCenter UI is divided into the following functional areas:

- **Lifecycle Manager** - Launches Lifecycle Manager (page 549) for managing installation and configuration for DataStax Enterprise clusters.
  
  **Note:** Accessing Lifecycle Manager requires the Admin role if OpsCenter authentication (page 101) is enabled. If the Lifecycle Manager menu is gray and unavailable, contact your OpsCenter admin.

- **Overview** - Provides a high-level overview of the clusters and activity within an OpsCenter instance. Displays any active alerts and a summary box for each cluster managed by OpsCenter. The Cluster Actions menu is not available from the Overview because it is a summary of all clusters in an OpsCenter instance.

- **Nodes** - View your cluster from different perspectives (Ring (page 224) or List (page 228) View) and perform certain maintenance operations (page 235) on cluster nodes. View the status of agents, troubleshoot, and install agents from the Agents (page 230) tab.

- **Dashboard** - View information about the clusters managed by OpsCenter and monitor a number of performance metrics (page 324). Real-time and historical performance metrics are available at different granularities: cluster-wide, per node, or per table. Organize your dashboards using presets (page 328), which you can also export and import (page 18) into other OpsCenter instances on other machines. Update notifications for upgrades to DataStax Enterprise are also displayed in the Dashboard.
• **Activities** - Displays all running tasks in the cluster. When tasks are running, the Activities icon displays a badge with the number of currently running tasks. View the most recent OpsCenter log events, such as OpsCenter startup and shutdown, in the Event Log.

• **Data** - Manage keyspaces and tables *(page 265)* (column families) within those keyspaces.

• **Services** - Enable DataStax Enterprise Management Services *(page 380)*; including the Backup *(page 380)*, Repair *(page 462)*, Capacity *(page 498)*, Best Practice *(page 502)*, and Performance *(page 521)* services.

• **Hide/Show Clusters** - Toggle to hide or show the list of clusters in the left navigation pane.

**Ring View**

**Cluster Ring View**

The Ring View displays a cluster as a ring of nodes from which you can determine at a glance node health, data distribution, and datacenter balance within a single visualization. To access the Ring view, click `cluster#Nodes#Ring` tab.

Figure 4: Vnode and non-vnode Ring View

Interpreting the Ring view:
A health summary of each datacenter is centrally located within each ring. Each health icon corresponds to the overall health summary statistics from left to right: Normal, Medium, or High load; nodes that are down or have an unknown status.

The color of each node or node status represents its health, which is determined by system load average (the number shown by the `uptime` command). Per core: 0–0.999 is Normal (green); 1–5 is Medium (yellow); 5+ is High (red). Red on a health summary number also indicates a node is down. Gray on a node slice or dot itself indicates the status of the node is either down (smaller node) or currently unknown (dashed border on non-vnode). Hover over any of the icons to view its description:

The size of each node represents its data size relative to all other nodes in the cluster.

Nodes are positioned around the ring according to their assigned token. In the case of vnodes or `ByteOrderedPartitioner`, nodes are displayed as slices of the ring and sized (page 227) based on the percentage of data they own.

If the datacenter has more nodes than can be displayed on the screen, the datacenter is represented as a condensed ring view, which typically occurs when the datacenter has hundreds of nodes.

Health, Data Size, and Alerts Summary

The Health summary pane located above the rings contains a cluster-wide summary of the data represented within the rings. You can quickly get an idea of the health of your cluster...
without having to manually scan each ring, which is especially useful for larger clusters. Hovering over a number in the health summary highlights the nodes included in that total. Use the Health summary to easily identify potential problem nodes, as well as whether any multiple nodes within a single replica set are experiencing issues.

The Data Size summary pane indicates the total size of the cluster data, the average size per node, and the standard deviation for data size between nodes. Alerts indicates the total number of any alerts that should be investigated.

Clicking a total (normal, medium, and high load, or nodes that are down or status unknown) in the health summary, or on the totals within each ring presents a list of nodes included in the total.

Node details

Hovering over a node or node slice displays some basic details within a ring about that node. The details are updated in real time.

Clicking on a node in the nodes list reveals the node details dialog that displays more information. Use the Actions menu to run various operations (page 235) on the node.
Streams

Whenever any nodes in a datacenter are streaming data to or from another node within the cluster, a streaming icon (indicated by the arrows) is displayed inside of the ring. The appearance of the streaming icon distinguishes between streams contained within that datacenter ("intra-dc streams") and streams between datacenters ("inter-dc streams").

Clicking on the streams icon number in any datacenter opens the Active Streams window, which gives details on all of the active streams in the cluster.

Node positioning

The goal of positioning non-virtual nodes in a ring is to visually represent whether a datacenter is balanced or not (that is, data is more likely to be evenly distributed across nodes). In a healthy ring, nodes are spread out evenly around the ring.
When a cluster uses RandomPartitioner or Murmur3Partitioner for its snitch, its nodes are positioned around the ring according to their assigned token, but there are some cases where positioning by token does not make sense:

- If vnodes are enabled (the default), each node is made up of multiple virtual nodes (256 by default), so positioning by token would mean having hundreds of times as many nodes around the ring.
- If a partitioner that does not use consistent hashing is used, such as ByteOrderedPartitioner, data is not guaranteed to be distributed evenly, so positioning by token also has no guaranteed value.

In those above cases, nodes are positioned based on the percentage of data they own in the ring, so a healthy ring is still represented by nodes being evenly spaced out.

Unknown datacenter list

Rings are displayed by datacenter. This information is gathered from the agent that runs on each node. If a new datacenter has just been added, or if new nodes have been added to a cluster, the unknown datacenter list is temporarily displayed above all rings until OpsCenter fully processes the new cluster information.

List View

List View is an alternative to the graphical Ring View. List View provides faster access to data and more flexibility when viewing data. All data is updated in real time.

To access the List view, click `cluster#Nodes#List` tab.
Select a check box next to the node or nodes to perform an operation on. Select the box next to the Name label to quickly select all nodes. Frequently performed actions such as repair and startup are available from the List View menu. The Other Actions menu provides options for less frequently performed actions, such as moving nodes. For more information, see Node management operations (page 235).

Icon legend

The icons displayed indicate the following:

- Spark Masters are indicated with a solid star icon. The Node Details dialog also indicates the Spark Master in the Type area.
- Spark Workers are indicated with a star outline icon. The Node Details dialog also indicates the Spark Workers in the Type area.
- Solr search nodes are indicated with the magnify glass icon. The Node Details dialog also indicates Search with Graph enabled nodes in the Type area.

Filtering nodes

Filter the list of nodes by Name, Datacenter, Token (applicable for non-vnodes only), Status, Load (CPU), or Data Size column criteria. Filtering nodes reduces the number of nodes displayed in the list, which is useful when working with large clusters that contain hundreds of nodes. Filter by:

- **Name**: Enter the full or partial machine name or IP address in the text box. The list is filtered by the criteria, and the filter criteria appears next to the column label.

- **Datacenter**: Filter by datacenter node type category of All, Analytics, Cassandra, or Solr.
Using OpsCenter

- **Status**: Filter by the node status of All, Active, or Unresponsive. The Status column displays whether a node is up or down, restarting, in a special mode (such as joining, draining, or moving), or running any tasks such as compactions.
- **Load (CPU)**: Filter by All; Low, High, Medium CPU load; or Offline.

If the filter criteria results in no matching nodes, click the **Clear Filters** button.

**Sorting columns**

Click a column label to sort columns in ascending or descending order. View which nodes have the most data, the highest CPU load, and so forth.

**Viewing node details**

Clicking the row for a node in the nodes list reveals the node details dialog that displays more information. Use the **Actions** menu to run various operations (page 235) on the node.

View node details such as Status, Capacity, Uptime, and Memory Usage, including In Memory if in use.

---

**Agents Status View**

View the current installation, configuration, and connection status of agents. Agent status automatically updates in the list view as the information becomes available within OpsCenter. Use the features in the Agents view to set up, upgrade, and configure agents when the status prompts indicate issues and provide buttons to launch the agent installation processes. Interpret the information provided to **troubleshoot (page 239)** agent installation, configuration, and connections.

**Access the Agents view**

To access the Agents view:

- Click `cluster#Nodes#Agents` tab.
Agent issue indicators: If displayed in the **Agent issues detected banner** at the top of the OpsCenter workspace, click the **Show Details#problems** link. The banner displays for agent issues that require immediate attention, such as incompatible agent versions or connection issues. A red exclamation mark ⚠️ on the Agents tab also indicates there are issues that require your attention.

The Agents status page is divided into **Healthy**, **Pending**, and **Unhealthy** sections depending on the current status of your environment. Any unhealthy DataStax agents that need attention appear at the top of the list.
Review agent status information summary

Review the available agent information:

- An **Upgrade all agents** button is available to upgrade nodes that have an incompatible agent version installed. To view which services might be adversely impacted, click the **What is impacted** link.

  **Note:** When the upgrade all agents prompt is displayed, DataStax recommends upgrading all agents immediately to avoid any loss of functionality.

- An **Upgrade agents** button is available to upgrade nodes that do not have the most current version of the agent installed.

  **Note:** DataStax recommends keeping the installed agent versions up-to-date and homogeneous on all nodes.

  For more details, see [DataStax agent version compatibility with OpsCenter version](page 233) below.

- A **Set Up Agents** button is available to queue installation for any nodes that currently do not have the agent installed.

- A **Configure Agents** button is available when agents cannot connect to OpsCenter because the agents are misconfigured. Click **Configure Agents** to update the STOMP configuration for those agents.

- If there is a problem with an agent connection to OpsCenter, the issue and status is clearly indicated in the appropriate column. Hover over each column to view its description:

- When available, additional details display by hovering the mouse pointer over the **status icons** (page 234) in each column:

- To view more details about the current state of an agent, click anywhere in the row for the agent. A JSON representation of the agent status details is displayed. The condition of an agent and the status of its individual services is visible. The type of agent install (package or tarball) is also indicated.
DataStax agent version compatibility with OpsCenter version

The following table summarizes the visual indicators when there are differences between the installed DataStax agent version and the installed OpsCenter version:

<table>
<thead>
<tr>
<th>Installed DataStax agent vs. OpsCenter version</th>
<th>Result and recommendation</th>
<th>Visual indicators</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent version is earlier than OpsCenter 6.1.0. For example, 6.0.3 as shown in the agent issues image above.</td>
<td>Incompatible versions require upgrading agents.</td>
<td>Red exclamation point ! in for installed status in the Last Known Version column; the Upgrade all agents button and What is impacted link are present.</td>
</tr>
</tbody>
</table>
Using OpsCenter

## Installed DataStax agent vs. OpsCenter version

<table>
<thead>
<tr>
<th>Installed DataStax agent vs. OpsCenter version</th>
<th>Result and recommendation</th>
<th>Visual indicators</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent version is not up-to-date with the OpsCenter version but is 6.1.0 or later. For example, OpsCenter is version 6.1.2 (future hypothetical release version) but the agent is version 6.1.1. Going forward from OpsCenter version 6.1, OpsCenter has improved backward-compatibility with agent versions for future releases.</td>
<td>Partially compatible versions. The currently installed version of OpsCenter is backward-compatible enough with the currently installed agent version so as not to cause any loss of service. Upgrading the agent version is recommended.</td>
<td>Yellow warning bell ⚠️ indicates that there could be some incompatibility but agents degrade gracefully.</td>
</tr>
<tr>
<td>Agent version is equal to OpsCenter version.</td>
<td>Completely compatible versions. Ideal scenario.</td>
<td>Green checkmark ✅ for installed status in the Last Known Version column.</td>
</tr>
</tbody>
</table>

### Icon Status Legend

The following table summarizes all of the icons that can appear in the Agents status columns:

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>🔄 (rotating green circles)</td>
<td>Currently installing, configuring, or upgrading an agent.</td>
</tr>
<tr>
<td>🕒 (greenish yellow clock)</td>
<td>Agent install job pending.</td>
</tr>
<tr>
<td>🎨 (yellow question mark)</td>
<td>Unknown status.</td>
</tr>
<tr>
<td>⚠️ (red exclamation point)</td>
<td>Down, incompatible version, agent install failed, or some other issue status as indicated in context within each column in the Agents status view.</td>
</tr>
<tr>
<td>⚠️ (greenish yellow warning bell)</td>
<td>Agent version limitations; partially compatible agent versions.</td>
</tr>
<tr>
<td>✅ (green check mark)</td>
<td>Up, satisfactory, or install complete status.</td>
</tr>
</tbody>
</table>
Related information:
Installing DataStax Agents [DataStax agents must be installed on every managed node in a cluster and are necessary to perform most of the functionality within OpsCenter.](page 59)

Nodes Detail View

The Nodes page for each cluster contains summary panels for Health, Data Size, and available services for DSE clusters. Click the cog icon for quick access to the Repair, Backup, Performance, or Best Practice Service. Click Hide or Show to toggle the view of the status panels.

Alerts are now shown for the cluster and instance levels: at the cluster level next to the Nodes label, and also for the entire OpsCenter instance in the main menu bar.

![Nodes Detail View](image)

Node management operations

Managing multiple nodes

Use OpsCenter to run operations (or actions) on nodes in an easy to use, visual way that takes the guesswork out of properly managing nodes in a cluster. Most node management operations can be run on multiple nodes of your choosing (for example, all the nodes in a cluster, all the nodes in a single datacenter, or a handful of problem nodes). The operations run in a rolling fashion and do not continue on to the next node until the previous one has completed successfully. If the operation fails on a node, the entire process stops.

To run an operation on multiple nodes (bulk operations), select those nodes in List View and choose an appropriate action.
Notifications appear when an operation starts and completes. Clicking **Show Details** takes you to the Activities page.

Managing single nodes

To run an operation on a single node, click that node from **Ring View (page 224)** or **List View (page 228)** and choose an action from the **Actions** menu from the node details dialog:

**Operations details**

**View Metrics (single node only)**

Redirects you to the Dashboard area of OpsCenter where you can select metrics graphs and configure performance views for the selected node.
View Replication (ring view, single datacenter only)
Shows the replication relationships between the selected node and other nodes in the cluster, based on the selected keyspace.

Start/Stop
Starts (page 263) or stops (page 263) the DataStax Enterprise process on a node.

Restart
Restarts (page 264) the DataStax Enterprise process on a node. If running restart on multiple nodes, each node is started as soon as the start command for the previous node returns.

Tip: To run a rolling restart of all nodes in a cluster, select Restart from the Cluster Actions menu.

Cleanup
Removes rows for which the node is no longer responsible. Cleanup is usually performed after changing the partitioner tokens or the replication options for a cluster. See Running cleanup (page 251).

Compact
Performs a major compaction, which is not a recommended procedure for most DataStax Enterprise clusters. See Running compaction (page 253).

Flush
Using OpsCenter

Flushes to disk as persistent SSTables the recent writes currently stored in memory (memtables). See Flushing tables (page 254).

**Repair**

Makes a node consistent with its replicas by doing an in-memory comparison of all the rows of a table and resolving any discrepancies between replicas by updating outdated rows with the current data. See Running a manual repair (page 260).

**Warning:** Do not run a manual repair operation from Node administration or using the command line while the Repair Service is On.

**Perform GC**

Forces the Java Virtual Machine (JVM) on the selected node to perform a garbage collection (GC). See Performing garbage collection (page 252).

**Decommission (single node only)**

Removes a node from the cluster and streams its data to neighboring replicas. See Decommission a node (page 256).

**Drain (single node only)**

Causes the recent writes currently stored in memory (memtables) to be flushed to disk as persistent SSTables and then makes the node read-only. The node stops accepting new writes until DSE is restarted. Draining a node (page 257) is usually done when upgrading a node.

**Move (single node only)**

Changes the partitioner token assignment for the node, thus changing the range of data that the node is responsible for. Not available if vnodes are enabled. See Moving a node (page 259).

**Configuring alerts**

Configure alerts to be notified when issues arise with various aspects of OpsCenter.

**Adding an alert for agent issues**

Add an alert to monitor DataStax agent issues. Any active alerts are indicated next to the Alerts menu in OpsCenter.

1. Click the Alerts menu.

2. In the Active Alerts dialog, click Manage Alerts.
   The Add Alert dialog appears.

3. In the Notify me when menu, choose Agent Issue.
4. Indicate the duration of the condition before alerting.

5. Select the notification frequency of the alert from the Notify every list.

6. Click Save Alert.

What’s next: If an agent issue alert is fired, investigate the agent issues in the Agents View (page 230).

Troubleshooting DataStax Agent Issues

Use this troubleshooting reference for resolving issues with DataStax agents. The issues appear in the Agents View along with troubleshooting tips, links to documentation, and links to the Install Agents dialog.

Table 10: Resolving agent issues

<table>
<thead>
<tr>
<th>Issues</th>
<th>Recommendations</th>
</tr>
</thead>
<tbody>
<tr>
<td>No agents detected.</td>
<td>The agents are not installed. Click <strong>Set Up Agents</strong> to automatically install (page 59) or start agents, or follow the documentation to manually install agents (page 63).</td>
</tr>
<tr>
<td>Agents are not connected to OpsCenter because they are misconfigured.</td>
<td>Click <strong>Configure Agents</strong> to update the stomp configuration for the misconfigured agents.</td>
</tr>
<tr>
<td>Package-installed agent has the wrong version.</td>
<td>The agent version installed for a package installation type does not match the currently installed agent version in OpsCenter. Click <strong>Upgrade Agents</strong> to install the correct version of the agent on all nodes.</td>
</tr>
<tr>
<td>Tar-installed agent has the wrong version.</td>
<td>The agent version being attempted to install for a tarball installation type does not match the currently installed agent version in OpsCenter. Tarball agent installs cannot be automatically upgraded. See the <strong>Upgrade Guide</strong> to upgrade the current version of OpsCenter.</td>
</tr>
<tr>
<td>Issues</td>
<td>Recommendations</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>The DataStax Enterprise cluster being monitored is misconfigured.</td>
<td>The DataStax Enterprise configuration does not match for all nodes in your cluster.</td>
</tr>
<tr>
<td>No HTTP communication to the agent.</td>
<td>OpsCenter cannot talk to these agents over HTTP. Check that the OpsCenter machine can reach these agents via HTTP and check that SSL is correctly enabled or disabled in both OpsCenter and agent configurations. See Configuring SSL (page 82) for more information.</td>
</tr>
<tr>
<td>The DataStax Enterprise storage cluster is misconfigured.</td>
<td>The configuration for the DataStax Enterprise storage cluster does not match for all nodes in your cluster. See Storing collection data on a separate cluster (page 154) and ensure the storage cluster is properly configured.</td>
</tr>
<tr>
<td>Node down.</td>
<td>The agent has confirmed that a node is down and requires investigation into the root cause. In addition to showing a node down in the Agents and Ring (page 224) views, you can add an alert (page 242) for nodes marked as down and configure alert notification (page 138) using email or POSTing to a URL.</td>
</tr>
<tr>
<td>Issues</td>
<td>Recommendations</td>
</tr>
<tr>
<td>--------</td>
<td>------------------</td>
</tr>
</tbody>
</table>
| JMX is misconfigured. | Check the JMX settings configured in address.yaml (page 205). The agent is unable to connect to JMX with the settings configured in OpsCenter:  
  - Either update the configuration in the OpsCenter Connection Settings > Edit Cluster dialog if all nodes have this error, or check the node-specific JMX config if this is an issue on a subset of nodes.  
  - The agent is unable to connect to JMX and is using advanced JMX settings configured in address.yaml. Check and update the JMX settings on the node:  
    - **jmx_host**  
      Host used to connect to local JMX server. The default setting is localhost. This information will be sent by opscenterd for convenience, but can be configured locally as needed. Example: jmx_host: 127.0.0.1  
    - **jmx_port**  
      Port used to connect to local JMX server. The default setting is 7199. This information will be sent by opscenterd for convenience, but can be configured locally as needed. Example: jmx_port: 7199  
    - **jmx_user**  
      The username used to connect to the local JMX server. Example: jmx_user: jmx-username  
    - **jmx_pass**  
      The password used to connect to the local JMX server. Example: jmx_pass: jmx-password  
      [This field may be encrypted for additional security.]  
    - **jmx_metrics_threadpool_size**  
      The size of the threadpool used for collecting metrics over JMX. Example: jmx_metrics_threadpool_size: 6 |
address.yaml

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

- **Package installations:** /var/lib/datastax-agent/conf/address.yaml
- **Tarball installations:** install_location/conf/address.yaml

**Related information:**
Stopping, starting, and restarting DataStax agents [Commands for stopping, starting, and restarting DataStax agents for each type of installation.](page 76)

### Adding an alert for down nodes

Add an alert to notify when a node has been marked as down (page 351) by OpsCenter. Any active alerts are indicated next to the Alerts menu in OpsCenter.

1. Click the **Alerts** menu.

2. In the **Active Alerts** dialog, click **Manage Alerts**.

   The **Add Alert** dialog appears.

3. From the **Notify me when** menu, choose **Node Down** and select the duration of the condition before alerting.

4. From the **Notify every** menu, select the duration of the condition before alerting.

5. Click **Save Alert**.

### Configuring an alert for KMIP errors

Configure an alert to monitor KMIP server status. For more information, see Configuring KMIP encryption. If the DataStax nodes are unable to contact the KMIP server or if the node is not authorized by the KMIP server, OpsCenter displays messages indicating the cause and resolution of the error.
1. Click the **Alerts** menu.

2. In the **Active Alerts** dialog, click **Manage Alerts**. The **Add Alert** dialog appears.

3. In the **Notify me when** menu, choose **KMIP Error**.

4. Indicate the duration of the condition before alerting.

5. Select the notification frequency of the alert and click **Save Alert**. Any KMIP errors are displayed in the **Event Log**.

---

**Configuring an alert for percentage of in-memory usage**

Configure an alert to monitor in-memory usage.

1. Click the **Alerts** menu.

2. In the **Active Alerts** dialog, click **Manage Alerts**. The **Add Alert** dialog appears.

3. In the **Notify me when** menu, choose **Advanced##DSE##In-Memory Percent Used**.
4. Select either **above** or **below** a percentage threshold and indicate the duration of the condition before alerting.

5. Select the notification frequency of the alert from the **Notify every** list.

6. Click **Save Alert**.
The configured In-Memory Percent Used alert appears in the Manage Alert Rules dialog.

Monitoring node operations

View and monitor memory usage, node details, and the Spark Console.

Viewing the Spark Console

Access the Spark Console for a Spark Master node from within OpsCenter. After accessing the Spark web UI, drill into Spark Worker Details.

1. In the left navigation pane, click Cluster#Nodes#List View.

Spark nodes are indicated by a star icon.

2. Click the node in the list to view its details.

   The View Spark Console link appears for nodes that are a Spark Master. Designate the Master Spark nodes by giving the node an alias (page 262) for faster identification.

3. Under Spark Master, click the View Spark Console link.
The Spark console launches in another browser window.
Monitoring in-memory usage

Monitor in-memory usage from within OpsCenter. More information about Creating or altering tables to use DSE In-Memory is available in DSE In-Memory.

A metric and an alert are available for monitoring in-memory usage:

- The In-Memory Percent Used alert (page 243) is available to configure for DataStax Enterprise nodes. If the in-memory usage exceeds the configured threshold, an alert is fired. Investigate the alert and adjust the memory threshold configuration as appropriate.
- The In-Memory Percent Used metric is available to add as a separate graph (page 325) in the dashboard of OpsCenter versions 5.1.2 and later.
A visual cue (an In-Memory label next to the table name) in the Keyspaces area of OpsCenter indicates whether a table uses the In-Memory option. Click **Data#Keyspace#Tables**:

1. In the left navigation pane, click **Cluster#Nodes#List View**.
2. Click the node to view its details.
Using OpsCenter

The details for the node are displayed. The Memory Usage bar graphs indicate System, Heap, and In-Memory Usage. The In-Memory Usage bar graph only appears if the In-Memory option is configured. In version 5.1.2 of OpsCenter, the In-Memory Usage interpretation depends on the DataStax Enterprise version (4.0 to 4.7 and later):

- For DSE versions 4.7 and later, the In-Memory Usage currently shown reflects all tables. Each in-memory table takes up a portion of the usage and displays as a different slice within the in-memory bar graph, up to the maximum threshold. The remainder of the graph represents free space.
- For DSE versions earlier than 4.7, the In-Memory Usage shown reflects per table limits in the bar graph. Since there is no maximum value applicable to all tables, the entire bar graph represents the total in-memory used by a table, split into as many sections as there are in-memory tables. Free space is not represented in the bar graph.

Viewing logs from node details

View logs in the Recent Log Information pane within the Node details dialog. The Cassandra System Log, Cassandra Debug Log, and OpsCenter Agent Log are available for viewing within OpsCenter monitoring. The most recent 1000 lines of a log are displayed in the log window.

Prerequisites:

**Note:** Viewing logs from within the OpsCenter UI node details is currently a labs feature (that is, under ongoing development but available for use). To enable the feature, add the following to opscenterd.conf and restart opscenterd (page 75):
Enable logs configuration option:

```
[labs]
log_enable = True
```

Enables Special Log Management Program (SLMP) that allows various forms of log management functionalities.

`opscenterd.conf`

The location of the `opscenterd.conf` file depends on the type of installation:

- **Package installations**: `/etc/opscenter/opscenterd.conf`
- **Tarball installations**: `install_location/conf/opscenterd.conf`

1. In the left navigation pane, click **Cluster#Nodes#List View or Ring View**.

2. Click the node to view its details.
   
The Node Details dialog appears.

3. Scroll down to the **Recent Log Information** pane.

4. Select the log to view:
   
   - Cassandra System Log
   - Cassandra Debug Log
   - OpsCenter Agent Log

5. Click **Refresh**.
   
The last 1000 lines of the selected log are fetched and displayed in the **Recent Log Information** pane.
Managing and maintaining nodes

Use OpsCenter to perform maintenance on your nodes, clean up extraneous data, and complete repairs.

### Running cleanup

Run the Cleanup option to remove unwanted data after adding a new node to the cluster.

For more information about manually running a cleanup, see the corresponding `nodetool cleanup` command.

**Prerequisites:** If OpsCenter role-based security (page 101) is enabled, be sure that the permission for the `Cleanup` option in Node Operations is enabled for the appropriate user roles.

1. Click *cluster name* → Nodes.

2. In the List view, select one or more nodes.

3. Click *Cleanup* from the main actions bar.

**Tip:** The *Cleanup* option is also available from the *Actions* menu in the Node Details dialog.

The Choose a Keyspace dialog appears.
4. Choose a keyspace and click **Choose**.
   The **Choose a Table** dialog appears if not cleaning up all keyspaces.

5. If applicable, select one or more tables and click **Choose**.
   The **Run Cleanup** dialog prompts you to confirm the operation.

6. Click **Cleanup**.
   A message in the top banner indicates the cleanup is in progress. Click the **Show Details** link to view the progress in the **Activities** page **General** tab. The banner message indicates when the cleanup is done.

### Performing garbage collection

Perform garbage collection (GC) using the Perform GC option in Nodes administration of OpsCenter monitoring. Performing GC forces the Java Virtual Machine (JVM) on the selected node to perform garbage collection.
Note: The Perform GC option in OpsCenter is not the same as the nodetool garbagecollect command.

Prerequisites: If OpsCenter role-based security (page 101) is enabled, be sure that the permission for the Garbage Collection (page ) option in Node Operations is enabled for the appropriate user roles.

1. Click <em>cluster name</em>#Nodes.
2. In the List view, select one or more nodes.
3. From the Other Actions menu, click Perform GC.
   
   Tip: The Perform GC option is also available from the Actions menu in the Node Details dialog.

   The Garbage Collect dialog appears and warns of a spike in latency.

4. Click Run GC.
   
   A message in the top banner indicates garbage collection is in progress. Click the Show Details link to view the progress in the Activities page General tab. The banner message indicates when the garbage collection is complete.

Running compaction

Run compaction on keyspaces and tables from node operations in OpsCenter. Compaction frees up disk space occupied by old SSTables and improves read performance by incrementally replacing old SSTables with compacted SSTables.

For information about manually running a compaction, see the corresponding nodetool compact command. For conceptual information, see Compaction strategies.

Prerequisites: If OpsCenter role-based security (page 101) is enabled, be sure that the permission for the Compact (page ) option in Node Operations is enabled for the appropriate user roles.
1. Click `cluster name#Nodes`.

2. In the List view, select one or more nodes.

3. From the Other Actions menu, click Compact.

   **Tip:** The Compact option is also available from the Actions menu in the Node Details dialog.

   The Choose a Keyspace dialog appears.

4. Select one or more or all keyspaces to compact and click Choose.

5. Select one or more or all tables to compact and click Choose. If you selected all keyspaces, all tables are also selected for compaction.

   The Run Compaction dialog appears and warns of a temporary decrease in performance.

   ![Run Compaction dialog](image)

6. Click Run Compaction.

   A message in the top banner indicates the operation is in progress. Click the Show Details link to view the progress in the Activities page General tab. The banner message indicates when the operation is complete.

### Flushing tables

Flush the memtables to SSTables on disk manually with the Flush menu option in OpsCenter Node Administration. Performing a flush might be necessary before taking a backup, or when performance issues arise due to inadequate flush frequency.

For more information, see the corresponding `nodetool flush` command.

**Prerequisites:** If OpsCenter role-based security (page 101) is enabled, be sure that the permission for the Flush (page ) option in Node Operations is enabled for the appropriate user roles.

1. Click `cluster name#Nodes`. 
2. In the List view, select a node.

3. From the Other Actions menu, click Flush.  

   **Tip:** The Flush option is also available from the Actions menu in the Node Details dialog.

   The Choose a Keyspace dialog appears.

   ![Choose a Keyspace dialog](image)

4. Choose a keyspace to flush and click Choose.

   The Choose a Table dialog appears.

   ![Choose a Table dialog](image)

5. Choose one or more tables to flush and click Choose.

   The Flush Node dialog prompts you to confirm the operation.
6. Click **Flush**.

A message in the top banner indicates the flush is in progress. Click the **Show Details** link to view the progress in the **Activities** page **General** tab. The banner message indicates when the flush is complete.

![Flush message]

**Decommission a node**

Decommission a node using OpsCenter Nodes administration.

1. Click **cluster name** > **Nodes**.

2. In the **Ring** (page 224) or **List** (page 228) view, select the node to decommission.

   The **Node Details** dialog appears.

![Node Details dialog]

3. From the **Actions** menu, click **Decommission**.

   The **Decommission Node** dialog prompts you to confirm the operation.
4. Click **Decommission Node**.

   **Important:** If you decommission and replace a node using the same hardware or virtual machine, the agent service must be restarted after the node is decommissioned. Restart the agents *(page 76)*.

5. **Restart opscenterd (page 75).*

### Draining a node

Drain a node using the Drain option in OpsCenter Nodes administration. Draining a node flushes all memtables from the node to SSTables on disk. Restarting DSE after draining a node is required for the node to start accepting writes again. Typically, a node is drained before upgrading to a new version of DSE.

For information about manually draining a node, see the corresponding `nodetool drain` command.

To simply flush memtables to disk, use the **Flush option (page 254)** in OpsCenter or the corresponding `nodetool flush` command.

**Prerequisites:** If **OpsCenter role-based security (page 101)** is enabled, be sure that the permission for the **Drain** *(page ***) option in **Node Operations** is enabled for the appropriate user roles.

1. Click **cluster name**#Nodes.

2. In the **Ring (page 224)** or **List (page 228)** view, select the node.
   
   The **Node Details** dialog appears.

3. From the **Actions** menu, click **Drain**.
The **Drain Node** dialog prompts you to confirm the operation.

4. Click **Drain Node**.

5. When the drain operations completes, perform the operation that required draining the node, such as upgrading a node manually.

6. Click **Restart** from the **Node Actions** menu.

   **Note**: The node cannot accept writes until DSE is restarted.

The **Restart DSE** dialog appears.
7. Click **Restart DSE**. It is not necessary to select the **Drain before stopping** option again.

## Moving a node

Move a node using OpsCenter Node Administration. Enter a new token to assign to a node. This procedure is not applicable to virtual nodes.

To move a node manually, see the corresponding `nodetool move` command.

**Prerequisites:**
- If OpsCenter role-based security *(page 101)* is enabled, be sure that the permission for the Move *(page 101)* option in **Cluster Topology** is enabled for the appropriate user roles.
- See **Calculating tokens for single-token architecture nodes** to determine the correct token value. The Move menu option is generally used to shift tokens slightly.

1. Click **cluster name**#Nodes.

2. In the **List** view, select the node.

3. From the **Other Actions** menu, click **Move**.

   **Tip:** The Move option is also available from the **Actions** menu in the **Node Details** dialog.

   The Move Node dialog appears.

4. Enter the token in the **New Token** box.

   **Note:** It is not necessary to escape a negative token value; doing so results in an error: Invalid token. Tokens must be numeric.

   If the token is out of range, an error message indicates the valid range. Enter a valid token within the given range.
5. Click **Move Node**.

   The **Move Node** dialog appears. During the move operation, the node is not available and cluster performance could be impacted.

6. Click **Move Node** to confirm. The Moving node... message displays in the **Move Node** dialog.

   A message in the top banner indicates the operation is in progress. Click the **Show Details** link to view the progress in the **Activities** page **General** tab. The banner message indicates when the operation is complete.

**Running a manual repair**

Run a manual ad hoc repair operation on tables in a keyspace. Use the Repair option in the Nodes Administration area of OpsCenter Monitoring.

For more information about manual repairs, see the corresponding *nodetool repair* command.

**Prerequisites:**

- If OpsCenter role-based security *(page 101)* is enabled, be sure that the permission for the **Repair** option in **Node Operations** is enabled for the appropriate user roles.

1. Click **cluster name**\#**Nodes**.

2. In the **List** view, select a node.

3. Click **Repair** from the main actions bar.
**Tip:** The **Repair** option is also available from the **Actions** menu in the **Node Details** dialog.

The **Choose Repair Options** dialog appears.

4. Select one or more or all tables within a keyspace. Click the keyspace name to select all its tables.

   Manual repairs can only be performed on a single keyspace per operation.

5. If applicable to your environment, select **Only use nodes in the same datacenter for the repair**. This option is analogous to using the `-local`, `--in-local-dc` option for nodetool
repair. Otherwise, nodes in other datacenters within a multi-DC environment are used in the repair as appropriate.

6. If applicable, select **Only repair this node's primary range**. This option is analogous to using the `-pr, --partitioner-range` option of nodetool repair.

7. To run the repair simultaneously on all replicas for the node's range, select **Fully concurrent repair**. This option is analogous to using the `-full, --full` option of nodetool repair.

   **Note**: Only select the fully concurrent option if using SSDs (solid-state drives) or do not have concerns with heavy disk I/O.

8. Click **Repair**.

   The **Run Repair?** dialog appears and warns that the repair operation might have a significant impact on system performance.

   ![Run Repair dialog]

9. Click **Run Repair**.

   A message in the top banner indicates the operation is in progress. Click the **Show Details** link to view the progress in the **Activities** page **General** tab. The banner message indicates when the operation is done.

### Configure an alias for a node

Configure an alias to display for a node throughout OpsCenter and Lifecycle Manager. An alias replaces the IP address or hostname displayed for a node. Give each node a meaningful and memorable name for your environment.

**address.yaml**

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

- **Package installations**: `/var/lib/datastax-agent/conf/address.yaml`
- **Tarball installations**: `install_location/conf/address.yaml`

1. Open the **address.yaml** file for editing.

2. Add the following option to the file:
3. Restart the agent. (*page 76*)

4. Repeat the above steps for each node.

## Starting and stopping DSE

Start, stop, and restart DSE on nodes in your cluster using OpsCenter.

### Starting DSE on a node

Start DSE on a node using the Start action in Nodes administration of OpsCenter monitoring.

For information about using commands instead of the OpsCenter UI to manually start DSE, see [Starting DataStax Enterprise as a service](#) and [Starting DataStax Enterprise as a stand-alone process](#).

**Prerequisites:** If OpsCenter role-based security (*page 101*) is enabled, be sure that the permission for the Start and Stop (*page *) option in Node Operations is enabled for the appropriate user roles.

1. Click `cluster#Nodes#List` tab.

2. In the List view, select one or more nodes.

3. From the Other Actions menu, click Start.

   **Tip:** The Start option is also available from the Actions menu in the Node Details dialog.

   A message in the top banner indicates the operation is in progress. Click the Show Details link to view the progress in the Activities page General tab. The banner message indicates when the operation is complete.

### Stopping DSE on a node

Stop DSE on a node using the Stop action in Nodes administration of OpsCenter monitoring.

For information about using commands instead of the OpsCenter UI to stop DSE on a node, see [Stopping a node](#).

**Prerequisites:** If OpsCenter role-based security (*page 101*) is enabled, be sure that the permission for the Start and Stop (*page *) option in Node Operations is enabled for the appropriate user roles.

1. Click `cluster#Nodes#List` tab.
Using OpsCenter

2. In the **List** view, select one or more nodes.

3. From the **Other Actions** menu, click **Stop**.

   **Tip:** The **Stop** option is also available from the **Actions** menu in the **Node Details** dialog.

   The **Stop DSE** dialog appears.

4. Indicate whether to drain the node before stopping.

5. Click **Stop DSE**.

**Restarting DSE on a node**

Restart the DataStax Enterprise service on any node. Restart a node from the nodes list menu in the **List View** (page 228) or from the Node Details dialog using the Actions menu.

1. Click **cluster#Nodes#List** tab.

2. Select the check box for the node or nodes to restart.

3. Click **Restart**.

   The Restart DSE dialog prompts you to confirm the restart.

4. Indicate whether to drain the node before stopping and restarting.
5. Click **Restart DSE**.

**Related information:**

**Restarting a cluster** [Restart an entire cluster in OpsCenter monitoring. Each node in the cluster restarts in a sequential rolling fashion after a sleep time elapses. Adjust the default rolling restart configuration options if necessary.] *(page 283)*

**Managing keyspaces and tables**

Manage keyspaces and the tables within keyspaces using the OpsCenter UI.

**Keyspaces**

Click **Data** in the left pane to list the keyspaces in a monitored cluster. View, edit, and delete keyspaces.

**Note:** Adding a keyspace in OpsCenter is no longer supported as of OpsCenter version 5.2 and later. The Data Explorer feature in OpsCenter has been removed. Add new keyspaces using CQL in DataStax Studio instead. For more information, see DataStax Studio.

**Managing a keyspace**

Edit keyspace settings or delete a keyspace. Adding a keyspace in OpsCenter is no longer supported. Add new keyspaces using CQL instead. See Configuring keyspaces.

1. Click **Data** in the left navigation pane.
   The list of Keyspaces appears.

![Keyspaces in OpsCenter](image-url)
2. Select a keyspace from the list of keyspaces.

   In **Keyspace Settings**, the replica placement strategy options for the keyspace appear.

3. To edit keyspace settings, click **Edit**.

   The Edit Keyspace *keyspace name* dialog appears.
Important: When defining NetworkTopologyStrategy parameters, CQL allows datacenters that are not currently present in a cluster topology. If any datacenter is not part of the monitored cluster, a warning icon and tooltip are displayed. Remove any extraneous datacenters from the keyspace by clicking its delete (trash) icon. To prevent replication of a keyspace to a datacenter, enter 0 (zero) for its Replication Factor.

For more information, see Changing the replication strategy for the OpsCenter keyspace (page 179).

4. To delete a keyspace, select the keyspace and click Delete Keyspace. A warning dialog prompts you to confirm deleting the keyspace.

5. To view table properties, select the keyspace, and select a table from the pane in the Tables tab. For more information, see Managing tables (page 270).

Viewing the CQL for User-Defined Types

View the CQL for User-Defined Types (UDT) within a keyspace in OpsCenter Monitoring. The CQL statements for user-defined types are view-only in OpsCenter.

Note: If OpsCenter authentication (page 101) is enabled, users must have permissions granted in their roles to view the schema (page ). See OpsCenter access roles overview (page 101).

For information about creating a UDT using CQL, see Creating a User-Defined Type in the DataStax CQL documentation.

1. Click Cluster#Data in the left navigation pane.

2. From the list of keyspaces, select a keyspace.
3. Click the **User-Defined Types** tab.

4. Select the defined type to view in the left navigation pane for the keyspace.

   If there are not any user types defined for a keyspace, *No user types in keyspace* displays in the left navigation pane.

![User-Defined Types tab in OpsCenter](Image)

---

**Viewing the CQL for User-Defined Functions**

View the CQL for User-Defined Functions (UDF) within a keyspace in OpsCenter Monitoring. The CQL statements for user-defined functions are view-only in OpsCenter.

**Note:** If **OpsCenter authentication** *(page 101)* is enabled, users must have permissions granted in their roles to **view the schema** *(page 101)*. See OpsCenter access roles overview *(page 101)*.

For information about creating a UDF using CQL, see **Creating a user-defined function** in the DataStax CQL documentation.

1. Click **Cluster#Data** in the left navigation pane.

2. From the list of keyspaces, select a keyspace.

3. Click the **User-Defined Functions** tab.

4. Select the defined function to view in the left navigation pane for the keyspace.

   If there are not any user functions defined for a keyspace, *No user functions in keyspace* displays in the left navigation pane.
Viewing the CQL for User-Defined Aggregates

View the CQL for User-Defined Aggregates (UDA) within a keyspace in OpsCenter Monitoring. The CQL statements for user-defined aggregates are view-only in OpsCenter.

**Note:** If OpsCenter authentication (page 101) is enabled, users must have permissions granted in their roles to view the schema (page ). See OpsCenter access roles overview (page 101).

For information about creating a UDA using CQL, see Creating a user-defined aggregate function in the DataStax CQL documentation.

1. Click **Cluster#Data** in the left navigation pane.
2. From the list of keyspaces, select a keyspace.
3. Click the **User-Defined Aggregates** tab.
4. Select the defined aggregate to view in the left navigation pane for the keyspace.

   If there are not any user types defined for a keyspace, *No user aggregates in keyspace* displays in the left navigation pane.
Managing tables

When you create a table in Cassandra using an application, the CLI, or CQL 2 or earlier, the table appears in OpsCenter. CQL 3, the default query language in Cassandra, does not support dynamic tables. Earlier versions of CQL and the CLI support dynamic tables.

In-memory tables are indicated next to the table name in the details section.

Managing a table

View or delete a table (formerly column family) in a keyspace, view metrics for a table, or truncate (delete) data from a table without deleting the table itself.

**Note:** If OpsCenter authentication (page 101) is enabled, users must have permissions granted in their roles to view the schema or truncate tables. See OpsCenter access roles overview (page 101).

1. Click **Data** in the left navigation pane.

2. From the list of keyspaces, select a keyspace.
   
   The (#TBL) shows how many tables each keyspace contains.

3. From the list of the tables, select a table. The CQL statement for the table appears.
   
   The CQL statement for the table appears.
4. Perform any of the following actions:

- **Delete**: Completely deletes the table from the keyspace. Select one or more tables in a keyspace to delete.
- **View Metrics**: Presents metrics for a table. In the Metric Options dialog, select a table metric to view. To aggregate measurements across the entire cluster, all nodes in the datacenter, or in a particular node, select Cluster Wide, All Nodes, or the IP address of a node. At this point, you can add a graph of the measurements to the Performance Metrics area, or choose a different table to measure.
• **Truncate**: Deletes all data from the table but does not delete the table itself. See *Truncating table data* *(page 272).*

**Truncating data from a table**

Truncate data from a table from within the Data workspace in OpsCenter Monitoring. A message warns you that truncated data is lost from OpsCenter. If a snapshot of the table does not already exist, a snapshot is created on your behalf. That snapshot resides in a snapshot directory within its table directory. If you need to restore the table data at some point in the future, you must do so manually by running sstableloader on the SSTables in the snapshot. The snapshot cannot be restored from the Backup Service. See *Restoring from a snapshot* in the DataStax Cassandra documentation.

To find out more about the ramifications of truncating data, and an alternative method of doing so manually, see TRUNCATE in the DataStax CQL documentation.

**Note:** If *OpsCenter authentication* *(page 105)* is enabled, the user role must have the *Truncate Data* *(page 105)* permission to perform the truncate operation.

1. Click **Cluster#Data** in the left navigation pane.
2. Select the keyspace that contains the table to truncate.
3. In the **Tables** tab, select the table for which you want to truncate data.

   You can only truncate data from one table at a time.
4. Click the **Truncate** link displayed above the CQL pane.

   When OpsCenter authentication is enabled, the Truncate link is only available if the Truncate Data permission is enabled for your assigned role.

   The Truncate Table dialog appears.

   **Warning:** Removal of the data is irreversible in OpsCenter.

5. Click **Truncate name_of_table**.

   The Table Truncated dialog appears and informs you the table was truncated.
6. Click **Close**.

**Browsing data deprecated**

*Note:* The Data Explorer feature in OpsCenter has been removed. Add new keyspaces using CQL in DataStax Studio instead. For more information, see [DataStax Studio](#).

**Cluster administration**

OpsCenter manages multiple DataStax Enterprise clusters with a single install of the central opscenterd server. The [OpsCenter Compatibility chart](#) indicates which versions of OpsCenter are compatible with the various DataStax Enterprise versions. Administer your clusters using the options available from the **Cluster Actions** menu. Generate reports about your clusters and download diagnostic data from the **Help** menu.

**Adding an existing cluster**

Add an existing DataStax Enterprise cluster to manage within OpsCenter. After you add the cluster to OpsCenter, you can import the cluster ([page 569](#)) into Lifecycle Manager to centrally manage the cluster topology and configuration.

*Note:* If configuration encryption ([page 112](#)) is active, OpsCenter automatically encrypts the sensitive fields such as passwords and writes the encrypted values to the configuration files. Do not enter manually encrypted values in the password fields.

1. Click **New Cluster** from the OpsCenter Monitoring main menu.

   The **New Cluster** prompt appears.

2. Click **Manage Existing Cluster** and **Get Started**.

   The **Connect to Existing Cluster** dialog appears.
3. Enter at least one hostname or IP address for a node in the cluster.

- ec2-123-45-6-789.us-west-1.compute.amazonaws.com
- ec2-234-56-7-890.us-west-1.compute.amazonaws.com

4. If applicable, change the **JMX Port** and **Native Transport Port** listen port (page 68) numbers if you are not using the defaults.

5. If applicable, click **JMX is enabled on my cluster** to add or edit user credentials (username and passwords) if the JMX port requires authentication.

6. If applicable, click **Native transport security is enabled on my cluster** to add or edit user credentials (username and password) if the Native Transport port requires authentication.

7. If applicable to your environment, select **DSE security (kerberos) is enabled on my cluster** and complete the fields.
a. Enter the **Service Name**. For example, if the server principal on your nodes is `dse/nodeX.example.com@EXAMPLE.COM`, this field should be `dse`.

b. Enter the **Opscenterd Client Principal** for the OpsCenter process/machine to use. Example: `opscenterd/opscenterd.EXAMPLE.COM`.

c. Enter the location of the keytab OpsCenter machine in **Opscenterd Keytab Location**, which contains credentials for the `opscenter_client_principal`. Example: `/etc/opscenter/security/krb5_opsc.keytab`.

d. Enter the client principal for the DataStax Agent process/machine to use in **DataStax Agent Client Principal**. Example: `dxagent/_HOST`.

   **Important**: Because each datastax-agent has a different principal name, the DataStax Agent Client Principal entered in this field is a placeholder. The `kerberos_client_principal` (page 136) property must be set in the address.yaml file for each datastax-agent. For example:

   ```
   kerberos_client_principal: datastax-agent@dsenode1/dsenode2/dsenode3
   ```

e. Enter the location of the keytab on the DataStax Agent machines in **DataStax Agent Keytab Location**, which contains credentials for the `agent_client_principal`. Example: `/usr/agent/conf/krb5_agent.keytab`.

   For more information, see Setting up Kerberos and the Kerberos tutorial.

8. If configuring client-to-node settings, select **Client-to-node encryption is enabled on my cluster**. Indicate the following paths for OpsCenter and each agent to use for connecting directly to the monitored DSE cluster.
When client-to-node encryption (page 604) is enabled for a cluster using Lifecycle Manager, the `ssl_truststore` and `ssl_truststore_password` fields are automatically propagated in `cluster_name.conf` with the corresponding values from `ssl_keystore` and `ssl_keystore_password` for both opscenterd and the agent: LCM propagates the `ssl_keystore` value into `ssl_keystore` and `ssl_truststore`; and the `ssl_keystore_password` value into `ssl_keystore_password` and `ssl_truststore_password`.

**Note:** For information about creating keystores and truststores, see [Enabling client-to-node encryption in OpsCenter](page 90).

**a.** Enter the **OpsCenter Keystore Path**, which is the SSL keystore location for OpsCenter (opscenterd) to use for connecting to the monitored DSE cluster. The value entered in the UI populates the `[cassandra] ssl_keystore` property in the OpsCenter cluster configuration file (`cluster_name.conf`).

**b.** Enter the **Password** for the **OpsCenter Keystore Path**. The value entered in the UI populates the `[cassandra] ssl_keystore_password` property in `cluster_name.conf`.

**c.** Enter the **OpsCenter Truststore Path**, which is the SSL truststore location for OpsCenter (opscenterd) to use for connecting to the monitored DSE cluster. This value should be the same as **OpsCenter Keystore Path** if the same file is used as both the keystore and the truststore (that is, there is not a separate truststore). The value entered in the UI populates the `[cassandra] ssl_truststore` property in `cluster_name.conf`.

**d.** Enter the **Password** for the **OpsCenter Truststore Path**. This value should be the same as password for the **OpsCenter Keystore Path** if the same
Using OpsCenter

file is used as both the keystore and the truststore (that is, there is not a separate truststore). The value entered in the UI populates the `ssl_truststore_password` property in `cluster_name.conf`.

e. Enter the **Agent Keystore Path**, which is the SSL keystore location for each agent to use for connecting to the monitored DSE cluster. The value entered in the UI populates the `[agents] ssl_keystore` property in `cluster_name.conf`.

f. Enter the **Password** for the **Agent Keystore Path**. The value entered in the UI populates the `[agents] ssl_keystore_password` property in `cluster_name.conf`.

g. Enter the **Agent Truststore Path**, which is the SSL truststore location for each agent to use for connecting to the monitored DSE cluster. This value should be the same as password for the **Agent Keystore Path** if the same file is used as both the keystore and the truststore (that is, there is not a separate truststore). The value entered in the UI populates the `[agents] ssl_truststore_password` property in `cluster_name.conf`.

h. Enter the **Password** for the **Agent Truststore Path**. This value should be the same as password for the **Agent Keystore Path** if the same file is used as both the keystore and the truststore (that is, there is not a separate truststore). The value entered in the UI populates the `[agents] ssl_truststore_password` property in `cluster_name.conf`.

9. Click **Next**.

   The **Set Up Agents** dialog appears.

   ![Set Up Agents dialog](image)

10. **Install or start agents automatically** is selected by default. Click **Next**.

   If your environment requires manual installation of agents, click **Install agents manually**. Click the link to access the instructions for manually installing agents *(page 63)*.

   The **Agents Credentials** dialog appears.
11. Enter SSH credentials to connect to the nodes:

   **Important**: You must specify either a sudo password for the indicated user, or a private SSH key. The installation will fail without one of these values.

   a. Enter a **Username**. The user must have root or sudo privileges.

   b. Enter a sudo **Password** for the specified user.

   c. Enter a **Private Key**.

   Entered credentials information is not saved or stored.

12. Click **Submit**.

   The **Agents tab (page 230)** opens, starts installing the agents, and displays the progress of the agent installation. The agent services might go up and down during the installation process.

   When the installation process successfully completes, the DataStax agents are deployed and configured for each managed node in the DataStax Enterprise cluster.
Using OpsCenter

Related information:
Disconnecting a cluster from OpsCenter and Lifecycle Manager [Disconnect a cluster from monitoring within OpsCenter and managing within Lifecycle Manager. Disconnecting a cluster from OpsCenter does not delete the cluster itself.](page 280)

**Disconnecting a cluster from OpsCenter and Lifecycle Manager**

Disconnect a cluster from monitoring within OpsCenter and managing within Lifecycle Manager. Disconnecting a cluster from OpsCenter does not delete the cluster itself.

Disconnecting a cluster also removes the cluster, its datacenters, and nodes from management within Lifecycle Manager. The cluster can be reconnected (page 274) for OpsCenter monitoring and then reimported into LCM (page 569) at any time.

1. From the OpsCenter Monitoring main menu, click **Settings#Cluster Connections**.

   The **Edit Cluster Connections Settings** dialog appears.
2. Select the cluster to disconnect from the **Cluster** list.

3. Click **Disconnect Cluster**.
   
   A prompt requests confirming the disconnection.
   
   ![Disconnect Cluster](image)

4. Click **Disconnect** to confirm disconnecting the cluster from monitoring within OpsCenter. The cluster, its datacenters, and nodes are also removed from management within Lifecycle Manager.

## Rebalancing a cluster overview

Cluster rebalancing ensures that each non-virtual node in a DataStax Enterprise cluster manages an equal amount of data. Currently, OpsCenter only supports rebalancing on clusters using the random partitioner or murmur 3 partitioner. Ordered partitioners are not supported. A rebalance is usually required only when the cluster topology has changed in some way, such as nodes were added or removed, or the replica placement strategy was changed. Configure an alert *(page 283)* to notify you when a cluster requires rebalancing. If using role-based security *(page 101)*, set permission to rebalance a cluster in the Cluster Topology section of the Role dialog.

A cluster is considered balanced when each node is responsible for an equal range of data. OpsCenter determines cluster balance by evaluating the partitioner tokens assigned to each node to make sure that the data ranges each node is responsible for are evenly distributed. Even though a cluster is considered balanced, it is still possible that some nodes have more data relative to others because only the number of rows (not the size of rows) managed by each node is taken into account.

The optimal path to rebalance clusters with around 100 nodes or less is determined by calculating the number of moves required and how much streaming data those moves would entail. If a cluster contains more than around 100 nodes, the optimal path is calculated based on simply the number of moves to expedite the rebalancing process.

When rebalancing a cluster, OpsCenter performs the following actions:

- Calculates appropriate token ranges for each node and identifies nodes that need to move.
- Makes sure that there is appropriate free space to perform the rebalancing.
Using OpsCenter

- Moves nodes one node at a time so as to lessen the impact on the cluster workloads. A move operation involves changing the partitioner token assignment for the node, thus changing the range of data that the node is responsible for. A move streams data from other nodes.
- Runs cleanup after a move is complete on a node. A cleanup operation removes rows that a node is no longer responsible for.

Rebalancing a cluster

Rebalance a non-vnode cluster to ensure each node in a DataStax Enterprise cluster manages an equal amount of data. Rebalancing a cluster is not applicable to vnodes.

1. Select the cluster to rebalance and click **Cluster Actions#Rebalance Cluster**. The Rebalance Cluster menu option is disabled for clusters using vnodes.
   
   OpsCenter checks if the token ranges are evenly distributed across the nodes in the cluster. If the cluster is already balanced, a message indicates rebalancing is not necessary at this time. If the cluster requires a rebalance, the Rebalance Cluster dialog appears showing the Proposed and Current topology.

2. Indicate a delay in seconds between node moves and click **Start Rebalance**. Check the status of the rebalance in the Activities area.
   
   Skipping a disk space check is not recommended.
3. If you cancel a rebalance operation before all nodes are moved, click **Rebalance Cluster** again to resume.

**Configuring an alert for rebalancing a cluster**

Configure an alert to rebalance a cluster. Clusters using vnodes do not require a manual rebalance. Rebalancing a cluster is a DataStax Enterprise feature only.

1. Click the **Alerts** menu.

2. In the Active Alerts dialog, click **Manage Alerts**.
   
   The Add Alert dialog appears.

3. In the **Notify me when** menu, select **Cluster Out of Balance**.

   ![Add Alert dialog](image)

4. Specify the notification criteria and click **Save Alert**.

**Restarting a cluster**

Restart an entire cluster in OpsCenter monitoring. Each node in the cluster restarts in a sequential rolling fashion after a sleep time elapses. Optionally, drain each node before stopping and restarting each node in the cluster. Some operations such as enabling commit log backups prompt you to perform a rolling restart. There are rolling restart configuration options available for fine-tuning your environment.

**Adjusting rolling restart configuration options**

If your environment experiences restart node failures, adjustments might be necessary to the default rolling restart configuration values in cluster_name.conf:

**[cassandra]** `rolling_restart_error_threshold`

A rolling restart will be cancelled if the number of errors during the restart reaches this number. This helps prevent having too many nodes down in your cluster if something catastrophic happens during a rolling restart. Default: 1

**[cassandra]** `rolling_restart_retry_attempts`

The maximum number of connection retry attempts after restarting a Cassandra node. Default: 25.
**[cassandra] rolling_restart_retry_delay**
The number of seconds to wait between retry attempts when connecting to
Cassandra after restarting a node. Default: 5.

**[cassandra] restart_delay**
During a rolling restart, the time in seconds OpsCenter waits after sending the
command to stop Cassandra before sending the command to start it again. The
default is 30 seconds.

cluster_name.conf

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

- **Package installations**: `/etc/opscenter/clusters/cluster_name.conf`
- **Tarball installations**: `install_location/conf/clusters/cluster_name.conf`

1. Click **Restart** from the **Cluster Actions** menu.

   The Rolling Restart dialog appears.

   ![Rolling Restart Dialog](image)

2. Set the amount of time to wait after restarting each node. The default is 60 seconds.

3. Select whether to drain the nodes before stopping.

4. Click **Restart Cluster**.

   A message at the top of the screen indicates the rolling restart is in progress.

5. To view the progress, click **Show Details** in the message, or click **Activities** in the
left navigation pane. The Activities icon reflects the number of operations currently in
progress. A cluster successfully restarted message indicates when the restart cluster
operation has completed.
Related information:
Restarting DSE on a node

[Restart the DataStax Enterprise service on any node.](page 264)

### Changing the display name of a cluster

Change the display name of a cluster as it displays in both OpsCenter and Lifecycle Manager. Lifecycle Manager displays both the original name and the display name alias. For example, the cluster alias (cluster schema name) displays in the Clusters (page 554) workspace.

Changing the display name does not change the actual cluster name in the schema. Differentiating display names for clusters can be helpful when distinguishing between clusters with the same name; such as in Dev, Test, and Live environments.

1. Click the arrow to show the cluster menu if it is not already displayed in the left navigation pane.

2. Click the arrow next to the cluster that you want to edit the display name for.

**Tip:** If the cluster has a long name that is not entirely visible, click and drag the border to resize the navigation menu.
3. Click **Edit Display Name**.
   The Edit Display Name dialog appears.

4. Enter the name you want displayed for the cluster in the **Cluster Name** box.

5. Click **Save Cluster Name**.
   The cluster display name changes throughout the OpsCenter and Lifecycle Manager.

### Downloading diagnostic data

Download a compressed tarball that contains diagnostic information about the OpsCenter daemon and all the nodes in a specific cluster. Examine the diagnostic data and provide the diagnostic tarball to DataStax Support to facilitate resolving any issues.

**Note:** If downloading the tarball times out, increase the default value of the `diagnostic_tarball_download_timeout` option in `cluster_name.conf`. Increasing the default value is recommended for DSE multi-instance clusters or for slower machines and connections.

`cluster_name.conf`

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

- **Package installations:** `/etc/opscenter/clusters/cluster_name.conf`
- **Tarball installations:** `install_location/conf/clusters/cluster_name.conf`

1. Click **Help#Diagnostics**.
   The **Collect Diagnostic Data** dialog appears.
2. Click **Download**.

3. If DataStax Enterprise was installed via the Installer-No Services or tarball or the `system.log` is not located in the default directory, OpsCenter cannot detect its location and prompts you to enter the full path to the `system.log` location. Enter the full path to the DataStax Enterprise system log in the **System Log Location** box and click **Save**.

   ![DSE System Log Location](image)

   **Note:** If you do not enter the system log location, the system log information is not included in the diagnostic report tarball.

4. A message indicates OpsCenter is collecting cluster data. Save the tarball to your local machine.

   `diagnostics.tar.gz`

   Depending on your browser settings, you might be prompted for a file directory to save the tarball in. For assistance with troubleshooting issues, provide the diagnostic tarball to DataStax Support.

**Diagnostic tarball reference**

Reference information about the contents of the diagnostic tarball. Read the [Diagnostic Tarball Goldmine article](link) in the DataStax Support blogs for highlights and a useful summary of the OpsCenter diagnostic tarball files.

In the collapsed view, each cluster diagnostic file contains the main directories and files:

- `nodes` directory
- `cluster_info.json` file
Using OpsCenter

• **opscenterd directory**

The directory structure after expanding the downloaded tarball:
Using OpsCenter

Diagnostic tarball files and directories

The directory structure, files, and their contents vary depending on the cluster and node configurations in an environment, and the installed versions of the DataStax Enterprise products.

The following directories only exist for nodes with Solr or Spark workloads:

- /conf/solr/, /logs/solr/, /logs/solr/tomcat/ for Solr nodes
- /conf/spark and /logs/spark for Spark nodes

Refer to the following table for links to more details about each file present in the diagnostic files directory. The sections below provide descriptions and examples.

<table>
<thead>
<tr>
<th>Main directories and files</th>
<th>Description</th>
<th>Files</th>
<th>Subdirectories</th>
</tr>
</thead>
<tbody>
<tr>
<td>cluster_info.json (page 291) file</td>
<td>Configuration and version information about the cluster.</td>
<td>See subdirectories.</td>
<td>nodes, opsccenterd</td>
</tr>
</tbody>
</table>
| node directory (page 293) | Subdirectories named for each node in the cluster. | • agent_version.json (page 293)  
• agent-metrics.json (page 293)  
• blockdev_report (page 294)  
• java_heap.json (page 301)  
• java_system_properties.json (page 301)  
• machine-info.json (page 304)  
• os-info.json (page 306)  
• process_limits (page 309) | cassandra-cli (page 294)  
conf (page 294)  
driver (page 300)  
dsetool (page 300)  
logs (page 302) (more subdirectories)  
nodetool (page 304)  
ntp (page 306)  
os-metrics (page 306)  
spark (page 299)  
solr (page 309) |
<table>
<thead>
<tr>
<th>Main directories and files</th>
<th>Description</th>
<th>Files</th>
<th>Subdirectories</th>
</tr>
</thead>
</table>
| opscnderd directory (page 310) | Log files, cluster configuration file, agent information, Best Practice Rules configuration, and status for the OpsCenter daemon. | - agent_requests.json (page 310)  
- agent_status.json (page 310)  
- best_practice_rules.json (page 310)  
- conf.json (page 312)  
- gc.log (page 313)  
- logback.xml (page 314)  
- node_info.json (page 318)  
- opscnderd.log (page 320)  
- repair_service_incremental.json (page 320)  
- repair_service subrange.json (page 321)  
- repair_service.log (page 321) | clusters: contains the cluster_name.conf file for the cluster. For more details, see Cassandra connection properties (page 196). |

**cluster_info.json**

Contains configuration and version information about the cluster, such as: cassandra version, number of cores, cluster operating system, OpsCenter version and os, and so forth.

An example:

```json
{
    "avg_token_count": 1,
    "bdp_version": [  
        "6.0.0",
        null  
    ],
    "cassandra_versions": [  
        "4.0.0.1935",
        null  
    ],
    "cluster_cores": 2,
    "cluster_instance_types": [  
        "m3.large",
        null  
    ],
    "cluster_os": [  
        "linux",
        "Ubuntu"  
    ],
    "cassandra_version": "4.0.0.1935",
    "cluster_os": [  
        "linux",
        "Ubuntu"
    ]
}
```
"14.04",
  "amd64"
],
[ null, null, null, null
],
"cluster_ram": 7985,
"columnfamily_count": 11,
"config_diff": {
  "cassandra": [
    "seed_hosts"
  ],
  "destinations": [
    "active"
  ],
  "webserver": [
    "interface"
  ]
},
"cql3_cf_count": 11,
"dc_count": 1,
"free_space": null,
"is_enterprise": true,
"keyspace_count": 6,
"node_count": 3,
"opscenter_arch": "",
"opscenter_cores": null,
"opscenter_instance_type": "m3.large",
"opscenter_os": "linux",
"opscenter_os_sub": "debian",
"opscenter_os_version": "jessie/sid",
"opscenter_ram": 7985,
"opscenter_version": "6.5.0SNAPSHOT",
"opscenterd_install_type": "package",
"partitioner": "org.apache.cassandra.dht.Murmur3Partitioner",
"python_version": "jython-2.7.1",
"rack_map": {
  "Cassandra.rack1": 3
},
"separate_storage": false,
"snitch": null,
"strategy_options": [
  "{class=org.apache.cassandra.locator.SimpleStrategy, replication_factor=1}\",
  "{class=org.apache.cassandra.locator.LocalStrategy}\",
  "{class=org.apache.cassandra.locator.LocalStrategy}\",
  "{class=org.apache.cassandra.locator.EverywhereStrategy}\",
  "{class=org.apache.cassandra.locator.EverywhereStrategy}\",
  "{class=org.apache.cassandra.locator.EverywhereStrategy}\"
Using OpsCenter

nodes diagnostic files
List of folders and files within each node folder.

agent_version.json file
Path: /nodes/node_folder_name/agent_version.json
The `agent_version.json` file indicates the version of the agent installed on a node.

6.1.2

agent-metrics.json file
Path: /nodes/node_folder_name/agent-metrics.json
Metrics collected from the node by the agent.

An excerpt:

```json
{
  "cassandra": {
    "histogram-size": {
      "count": 5825973,
      "description": "Compressed size of histograms after serialization",
      "max": 132,
      "mean": 1.9609843069629802,
      "min": 1,
      "p50": 1.0,
      "p75": 1.0,
      "p95": 5.0,
      "p98": 8.0,
      "p99": 23.0,
    }
  }
}
```
blockdev_report file

Path: /nodes/node_folder_name/blockdev_report

Contains a report on various statistics for block devices used by the operating system.

An example:

<table>
<thead>
<tr>
<th>RO</th>
<th>RA</th>
<th>SSZ</th>
<th>BSZ</th>
<th>StartSec</th>
<th>Size</th>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>rw</td>
<td>256</td>
<td>512</td>
<td>4096</td>
<td>0</td>
<td>34359738368</td>
<td>/dev/vda</td>
</tr>
<tr>
<td>rw</td>
<td>256</td>
<td>512</td>
<td>4096</td>
<td>2048</td>
<td>34358165504</td>
<td>/dev/vda1</td>
</tr>
</tbody>
</table>

cassandra-cli folder

Path: /nodes/node_folder_name/cassandra-cli

The cassandra folder contains the following files:

- describe_cluster
- show_keyspaces

The contents of both of these files state: The removal of Thrift in DSE 5.0 also removes support for cassandra-cli

cnf folder

Path: /nodes/node_folder_name/conf

Configuration directory for all config files relevant to a node.

The conf folder contains the following subfolders:

- agent folder (page 294)
- cassandra folder (page 296)
- dse folder (page 299)
- spark folder (page 299)
- solr folder (page 300)
- system folder (page 300)

Tip: Use LCM Config Profiles (page 594) to centrally manage Cassandra, DSE, Spark, and other configuration files.

agent folder

Path: /nodes/node_folder_name/conf/agent/agentaddress.yaml
**Note:** The `agentaddress.yaml` file is located underneath the `/nodes/node_folder_name/conf` folder but not within it (or the agent folder) for diagnostic tarball files generated in OpsCenter versions earlier than 6.1.1. Those diagnostic files generated for OpsCenter version 6.1.1 and later are contained within the agent folder.

The `agentaddress.yaml` file contains agent configuration settings in `address.yaml` explicitly configured at the node level. Depending on the environment and its configuration, additional files such as `log4j.properties` could appear in the agent folder.

**Note:** The agent does not use logback for logs; it still uses log4j.

**Example** `agentaddress.yaml`:

```
stomp_interface: 10.200.181.112
use_ssl: 0
```

**Example** `log4j.properties`:

```
# Based on the example properties given at http://logging.apache.org/log4j/1.2/manual.html
# Set root logger level to DEBUG and its only appender to A1.
log4j.rootLogger=INFO,R,stdout
log4j.additivity.org.apache=false
log4j.additivity.org.eclipse.jetty=false
log4j.additivity.com.datastax.driver=false
# Silence "missing LZ4" warning
log4j.additivity.com.datastax.driver.core.FrameCompressor=false
# stdout
log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.layout=opsagent.AlternatingEnhancedPatternLayout
log4j.appender.stdout.layout.MainPattern=%5p [%t] %d{ISO8601} %m%n %throwable{200}
log4j.appender.stdout.layout.AlternatePattern=%5p [%t] %d{ISO8601} %m%n %throwable{3}
log4j.appender.stdout.layout.ToMatch=com.datastax.driver
# rolling log file
log4j.appender.R=org.apache.log4j.RollingFileAppender
log4j.appender.R.maxFileSize=20MB
log4j.appender.R.maxBackupIndex=5
log4j.appender.R.layout=opsagent.AlternatingEnhancedPatternLayout
log4j.appender.R.layout.MainPattern=%5p [%t] %d{ISO8601} %m%n %throwable{200}
log4j.appender.R.layout.AlternatePattern=%5p [%t] %d{ISO8601} %m%n %throwable{3}
log4j.appender.R.layout.ToMatch=com.datastax.driver
log4j.appender.R.File=/var/log/datastax-agent/agent.log
```
cassandra folder

Path: /nodes/node_folder_name/conf/cassandra

The cassandra folder contains the following files:

- cassandra-env.sh: Shell script file for the Cassandra environment used for adjusting JVM options, heap size, and setting JMX properties.
- cassandra.yaml: Configuration settings file for Cassandra.
- commitlog_archiving.properties: Properties file for commitlog archiving.

**Tip:** Use LCM Config Profiles (page 594) to centrally manage cassandra.yaml and cassandra-env.sh files.

**Note:** A configuration file explicitly not managed at this time by Lifecycle Manager is commitlog_archiving.properties, which is used for configuring commit log archive (page 384) and PIT restore for the Backup Service. This file is managed instead from within the Backup Service (page 380).

Example excerpt cassandra.yaml:

```yaml
# Cassandra storage config YAML

# NOTE:
# See http://wiki.apache.org/cassandra/StorageConfiguration for
# full explanations of configuration directives
# /NOTE

# The name of the cluster. This is mainly used to prevent machines in
# one logical cluster from joining another.
cluster_name: sunshine

# This defines the number of tokens randomly assigned to this node on
# the ring
# The more tokens, relative to other nodes, the larger the proportion of
# data that this node will store. You probably want all nodes to have the
# same number
# of tokens assuming they have equal hardware capability.
# If you leave this unspecified, Cassandra will use the default of 1
# token for legacy compatibility,
# and will use the initial_token as described below.
# Specifying initial_token will override this setting on the node's
# initial start,
# on subsequent starts, this setting will apply even if initial token is
# set.
# If you already have a cluster with 1 token per node, and wish to
```
# multiple tokens per node, see http://wiki.apache.org/cassandra/Operations
num_tokens: 1
...

### Example excerpt: commitlog_archiving.properties:

```
# Licensed to the Apache Software Foundation (ASF) under one
# or more contributor license agreements. See the NOTICE file
# distributed with this work for additional information
# regarding copyright ownership. The ASF licenses this file
# to you under the Apache License, Version 2.0 (the
# "License"); you may not use this file except in compliance
# with the License. You may obtain a copy of the License at
#
#   http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing,
# software distributed under the License is distributed on an
# "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
# KIND, either express or implied. See the License for the
# specific language governing permissions and limitations
# under the License.

# commitlog archiving configuration. Leave blank to disable.

# Command to execute to archive a commitlog segment
# Parameters: %path => Fully qualified path of the segment to archive
# %name => Name of the commit log.
# Example: archive_command=/bin/ln %path /backup/%name
#
# commitlog archiving configuration. Leave blank to disable.

# Command to execute to archive a commitlog segment
# Parameters: %path => Fully qualified path of the segment to archive
# %name => Name of the commit log.
# Example: archive_command=/bin/cp -f %path /backup/%name
#
# Limitation: *_command= expects one command with arguments. STDOUT
# and STDIN or multiple commands cannot be executed. You might want
# to script multiple commands and add a pointer here.
archive_command=

# Command to execute to make an archived commitlog live again.
# Parameters: %from is the full path to an archived commitlog segment
# (from restore_directories)
# %to is the live commitlog directory
# Example: restore_command=/bin/cp -f %from %to
restore_command=

# Directory to scan the recovery files in.
restore_directories=
```
Example excerpt `cassandra-env.sh`:

```bash
# Licensed to the Apache Software Foundation (ASF) under one
# or more contributor license agreements. See the NOTICE file
# distributed with this work for additional information
# regarding copyright ownership. The ASF licenses this file
# to you under the Apache License, Version 2.0 (the
# "License"); you may not use this file except in compliance
# with the License. You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
# implied.
# See the License for the specific language governing permissions and
# limitations under the License.

calculate_heap_sizes()
{
    case "\`uname\`" in
        Linux)
            system_memory_in_mb=`free -m | awk '/:/ {print $2;exit}'`
            system_cpu_cores=`egrep -c 'processor([[:space:]]+):.*' /proc/cpuinfo`
            ;;
        FreeBSD)
            system_memory_in_bytes=`sysctl hw.physmem | awk '{print $2}'`
            system_memory_in_mb=`expr $system_memory_in_bytes / 1024 / 1024`
            system_cpu_cores=`sysctl hw.ncpu | awk '{print $2}'`
            ;;
        SunOS)
            system_memory_in_mb=`prtconf | awk '/Memory size:/ {print $3}'`
            system_cpu_cores=`psrinfo | wc -l`
            ;;
        Darwin)
            system_memory_in_bytes=`sysctl hw.memsiz | awk '{print $2}'`
            system_memory_in_mb=`expr $system_memory_in_bytes / 1024 / 1024`
            system_cpu_cores=`sysctl hw.ncpu | awk '{print $2}'`
            ;;
        *)
            # assume reasonable defaults for e.g. a modern desktop or
            # cheap server
            system_memory_in_mb="2048"
            system_cpu_cores="2"
            ;;
    esac
}
```
Using OpsCenter

```bash
esac

    # some systems like the raspberry pi don't report cores, use at
    least 1
    if [ "${system_cpu_cores}" -lt "1" ]
      then
          system_cpu_cores="1"
    fi

...  
```

dse folder

Path: `/nodes/node_folder_name/conf/dse`

The dse folder contains the following files:

- `dse.yaml`: Configuration settings file for DSE. See `dse.yaml` configuration file
- `logback.xml`: Configured logging files. See Configuring logging.

**Tip:** Use LCM Config Profiles *(page 594)* to centrally manage `dse.yaml` files.

location.json file

Path: `/nodes/node_folder_name/conf/location.json`

**Note:** Located underneath the `/nodes/node_folder_name/conf/dse` folder but not within it.

This file indicates the location of the `dse.yaml` and `cassandra.yaml` files on the node. The location path is also indicative of the installation type.

Example (package installation):

```json
{
    "dse" : "/etc/dse/dse.yaml",
    "cassandra" : "/etc/dse/cassandra/cassandra.yaml"
}
```

A tarball installation would indicate `installation_location/resources/dse/conf/dse.yaml`; and `installation_location/resources/cassandra/conf/cassandra.yaml`.

spark folder

Path: `/nodes/node_folder_name/conf/spark`

Contains Spark configuration files:

- `dse-spark-env.sh`
- `hive-site.xml`
- `logback-spark.xml`
Using OpsCenter

- logback-spark-executor.xml
- logback-sparkR.xml
- logback-spark-server.xml
- spark-daemon-defaults.conf
- spark-defaults.conf
- spark-env.sh

**Tip:** Use LCM Config Profiles *(page 594)* to centrally manage spark configuration files.

solr folder

Path: /nodes/node_folder_name/solr

Contains the schema.xml and solrconfig.xml files for each category. See also *(page 309)* the /node/solr/index_size.json file.

system folder hosts file

Path: /nodes/node_folder_name/conf/system/hosts

The system folder contains the hosts file derived from etc/hosts. The hosts file is an operating system plain text file that maps hostnames to IP addresses. The hosts file could be managed by a third party configuration management systems such as puppet.

driver folder

Path: /nodes/node_folder_name/driver

The driver folder contains the following files:

- **metadata:** Contains the cluster name and partitioner information.
- **schema:** Contains the schema with all CREATE statements.

dsetool folder

Path: /nodes/node_folder_name/dsetool

The dsetool folder contains the following files:

- **ring:** Lists the nodes in the ring.
- **sparkmaster:** Deprecated. Please use dse client-tool instead.

Example excerpt *listjt*:

```sh
/usr/bin/dsetool --host=127.0.0.1 --jmxport=7199 listjt
exit status: 1
```

stdout:
usage: dsetool [-short <arg>] [-long=<arg>] <command> [command-args]
-a,--jmxusername <arg> JMX user name
Using OpsCenter

-\texttt{b,--jmxpassword <arg>} \textit{JMX password}
-\texttt{-c,--cassandra_port <arg>} \textit{Cassandra port to use}
-\texttt{--cipher-suites <arg>} \textit{Comma separated list of SSL cipher suites for connection to Cassandra when SSL is enabled}
-\texttt{-f,--config-file <arg>} \textit{DSE configuration file}

\textbf{java_heap.json file}

Path: /nodes/node\_folder\_name/java\_heap.json

\textit{Note: Located underneath the /nodes/dsetool folder but not within it.}

Shows heap and non-heap memory usage. For more information, see \textit{tuning the Java heap}.

\textbf{Example:}

```json
{
    "HeapMemoryUsage" : {
        "committed" : 2092957696,
        "init" : 2092957696,
        "max" : 2092957696,
        "used" : 1234174816
    },
    "NonHeapMemoryUsage" : {
        "committed" : 128671744,
        "init" : 2555904,
        "max" : -1,
        "used" : 124666688
    }
}
```

\textbf{java_system_properties.json file}

Path: /nodes/node\_folder\_name/java\_system\_properties.json

\textit{Note: Located underneath the /nodes/dsetool folder but not within it.}

Shows Java system properties.

\textbf{Example excerpt:}

```json
{
    "java.rmi.server.hostname" : "10.200.181.112",
    "java.vendor.url.bug" : "http://bugreport.sun.com/bugreport/",
    "com.sun.management.jmxremote.authenticate" : "false",
    "cassandra.config.loader" : "com.datastax.bdp.config.DseConfigurationLoader",
    "java.vm.name" : "Java HotSpot(TM) 64-Bit Server VM",
    "java.vm.version" : "25.40-b25",
    "java.specification.name" : "Java Platform API Specification",
    "java.specification.version" : "1.7.0_45",
    "java.runtime.name" : "Java(TM) SE Runtime Environment (build 25.40-b25)"
}
```
Using OpsCenter

```json
"cassandra.custom_query_handler_class" : "com.datastax.bdp.cassandra.cql3.DseQueryHandler",
"java.io.tmpdir" : "/tmp",
"java.runtime.name" : "Java(TM) SE Runtime Environment",
"sun.java.command" : "com.datastax.bdp.DseModule",
"sun.java.launcher" : "SUN_STANDARD",
"java.vendor" : "Oracle Corporation",
"os.version" : "3.13.0-133-generic",
...
```

logs folder

Path: `/nodes/node_folder_name/logs`

The logs folder contains the following subfolders and files, depending on the configured workloads:

- **cassandra folder**: Contains the `debug.log`, `gremlin.log`, `output.log`, and `system.log` files.
- **opsagent folder**: Contains the `agent.log`.
- **solr folder**: Contains the `solrvalidation.log` and the `tomcat` folder of its logs.

**Tip:** View certain logs (Cassandra System Log, Cassandra Debug Log, and OpsCenter Agent Log) from within the OpsCenter UI by configuring the logs labs feature *(page 22)* available in OpsCenter.

Example excerpt from a `debug.log` for a node:

```
DEBUG [PerDiskMemtableFlushWriter_0:45] 2018-01-26 14:52:45,433
   Memtable.java:485 - Completed flushing /var/lib/cassandra/data/system/local-7ad54392bddd35a684174e047860b377/mc-55-big-Data.db (0.104KiB)
   for commitlog position CommitLogPosition(segmentId=1516899136469, position=31359084)
   ColumnFamilyStore.java:1228 - Flushed to [BigTableReader(path='/var/lib/cassandra/data/system/local-7ad54392bddd35a684174e047860b377/mc-55-big-Data.db')] (1 sstables, 5.111KiB), biggest 5.111KiB, smallest 5.111KiB
DEBUG [COMMIT-LOG-ALLOCATOR] 2018-01-26 15:00:41,021
   AbstractCommitLogSegmentManager.java:109 - No segments in reserve; creating a fresh one
   OutboundTcpConnection.java:445 - Attempting to connect to /10.200.182.90
   OutboundTcpConnection.java:552 - Done connecting to /10.200.182.90
DEBUG [RMI TCP Connection(1463)-127.0.0.1] 2018-01-26 15:12:14,116
   StorageProxy.java:2642 - Schemas are in agreement.
...```

Example excerpt from an `output.log` for a node:

```
```
Example excerpt from a system.log for a node (after cassandra.yaml loaded):

```
INFO [main] 2018-01-25 16:47:18,603  DseConfig.java:402 - CQL slow log is enabled
INFO [main] 2018-01-25 16:47:18,604  DseConfig.java:403 - CQL system info tables are not enabled
INFO [main] 2018-01-25 16:47:18,604  DseConfig.java:404 - Resource level latency tracking is not enabled
INFO [main] 2018-01-25 16:47:18,605  DseConfig.java:405 - Database summary stats are not enabled
INFO [main] 2018-01-25 16:47:18,605  DseConfig.java:406 - Cluster summary stats are not enabled
INFO [main] 2018-01-25 16:47:18,605  DseConfig.java:407 - Histogram data tables are not enabled
INFO [main] 2018-01-25 16:47:18,605  DseConfig.java:408 - User level latency tracking is not enabled
INFO [main] 2018-01-25 16:47:18,606  DseConfig.java:410 - Spark cluster info tables are not enabled
INFO [main] 2018-01-25 16:47:18,610  DseConfig.java:448 - This instance appears to have 1 thread per CPU core and 2 total CPU threads.
```

Example excerpt from an agent.log for a node:

```
INFO [qtp192788371-31108] 2017-07-28 23:00:00,022 HTTP request started:
{"protocol":"HTTP/1.1","remote-addr":"10.200.175.206","params":
{},"headers":
{"user-agent":"http-kit/2.0","host":"10.200.175.206:61621","accept-encoding":
"gzip, deflate","content-length":"2","opscenter-id":"0f61c8368c834d3a9e4d9e8713e884bb",
```
machine_info.json file

Path: /nodes/node_folder_name/machine_info.json

Note: Located underneath the /nodes/logs folder but not within it.

Shows the processor architecture and memory microcircuit of the CPU for a machine.

Example:

```
{
    "arch" : "amd64",
    "memory" : 7985
}
```
	nodetool folder

Path: /nodes/node_folder_name/nodetool

The set of nodetool commands that OpsCenter executes is predetermined and controlled
by the DataStax agent code. The nodetool operations do not depend on node workload or
anything else such as nodetool commands that were executed externally from OpsCenter
using the nodetool utility CLI. For more information, see the nodetool utility in the DSE
Admin documentation.

The nodetool folder contains the following files:

- cfstats
- compactionhistory
- compactionstats
- describecluster
- getcompactionthroughput
- getstreamthroughput
- gossipinfo
- info
- netstats
- proxyhistograms
- ring
- status
- statusbinary
- tpstats
• **version**: Release Version of Cassandra, such as 4.0.0.1935.

**Examples:**

**getstreamthroughput:**

<table>
<thead>
<tr>
<th>Current stream throughput: 200 Mb/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current streaming connections per host: 200</td>
</tr>
</tbody>
</table>

**gossipinfo:**

```
/10.200.179.234
  generation:1510023125
  heartbeat:683548
  STATUS:23:NORMAL,-9223372036854775808
  LOAD:683492:5.80418858E8
  SCHEMA:19:7af56410-33a6-38ed-980a-d07dbbaf831
  DC:45:Cassandra
  RACK:17:rack1
  RELEASE_VERSION:4:4.0.0.1935
  NATIVE_TRANSPORT_ADDRESS:3:10.200.179.234
  X_11_PADDING:92140:
  {"dse_version":"6.0.0","workloads":"Cassandra","workload":"Cassandra","active":"true","server_id":"FA-16-3E-CA-BB-55","graph":false,"health":0.9} 
```

```
/10.200.179.235
  generation:0
  heartbeat:0
  TOKENS: not present
```

```
/10.200.179.236
  generation:1510023127
  heartbeat:683533
  STATUS:23:NORMAL,-3074457345618258603
  LOAD:683481:5.87518118E8
  SCHEMA:19:7af56410-33a6-38ed-980a-d07dbbaf831
  DC:53:Cassandra
  RACK:17:rack1
  RELEASE_VERSION:4:4.0.0.1935
  NATIVE_TRANSPORT_ADDRESS:3:10.200.179.236
  X_11_PADDING:92067:
  {"dse_version":"6.0.0","workloads":"Cassandra","workload":"Cassandra","active":"true","server_id":"FA-16-3E-CA-BB-55","graph":false,"health":0.9} 
```

```
NET_VERSION:1:256
HOST_ID:2:9440f6c1-4d01-4216-ad9b-9d5c71afce6e
NATIVE_TRANSPORT_READY:58:true
NATIVE_TRANSPORT_PORT:5:9042
STORAGE_PORT:7:7000
STORAGE_PORT_SSL:8:7001
JMX_PORT:9:7199
TOKENS:22:<hidden> 
```

```
/10.200.179.235
  generation:0
  heartbeat:0
  TOKENS: not present
/10.200.179.236
  generation:1510023127
  heartbeat:683533
  STATUS:23:NORMAL,-3074457345618258603
  LOAD:683481:5.87518118E8
  SCHEMA:19:7af56410-33a6-38ed-980a-d07dbbaf831
  DC:53:Cassandra
  RACK:17:rack1
  RELEASE_VERSION:4:4.0.0.1935
  NATIVE_TRANSPORT_ADDRESS:3:10.200.179.236
  X_11_PADDING:92067:
  {"dse_version":"6.0.0","workloads":"Cassandra","workload":"Cassandra","active":"true","server_id":"FA-16-3E-CA-BB-55","graph":false,"health":0.9} 
```

```
NET_VERSION:1:256
HOST_ID:2:d71bb70d-4940-4eb2-addf-72897dd803f0
NATIVE_TRANSPORT_READY:65:true
NATIVE_TRANSPORT_PORT:5:9042
```

```
ntp folder

Path: /nodes/node_folder_name/ntp

Contains files for NTP (Network Time Protocol) for clock synchronization. Synchronized clocks are critical for consistent data determined by timestamps. The diagnostic tarball runs nptstat and npttime.

The ntp folder contains the following files:

- nptstat: Reports the synchronisation state of the NTP daemon running on the local machine. Shows statistics for the NTP synchronization that indicates polling interval and time accuracy lifespan.
- npttime: Monitors drift and offset from an NTP server. Shows some information about kernel parameters used by the NTP system.

**Important:** If clocks are not properly synchronized, or the system does not have nptstat and npttime installed or in the correct path, the Best Practice Rule for clocks out of sync (page 510) fails. Clock drift can interfere with LCM generating TLS certificates (page 579). NTP synchronizes all participating computers to within a few milliseconds of Coordinated Universal Time (UTC). The rule for UTC checks (page 510) that clocks across the nodes are in Coordinated Universal Time (UTC).

Example nptstat:

synchronised to NTP server (10.200.175.206) at stratum 1
  time correct to within 24 ms
  polling server every 60 s

Example npttime:

ntp_gettime() returns code 0 (OK)
  time dd33417a.f6cc3dd4 Mon, Aug 14 2017 19:44:55.964, (.964054877),
  maximum error 106330 us, estimated error 100 us, TAI offset 0

ntp_adjtime() returns code 0 (OK)
  modes 0x0 (),
  offset 81.298 us, frequency -4.200 ppm, interval 1 s,
  maximum error 106330 us, estimated error 100 us,
  status 0x2001 (PLL,NANO),
  time constant 6, precision 0.001 us, tolerance 500 ppm,

os-info.json file

Path: /nodes/node_folder_name/os-info.json
Note: Located underneath the /nodes/ntp folder but not within it.

The operating system information file os-info.json shows the installed operating system and its version.

Example:

```
{
   "sub_os" : "CentOS Linux",
   "os_version" : "7.2.1511"
}
```

os-metrics folder

Path: /nodes/node_folder_name/os-metrics

The os-metrics folder contains the following files:
- cpu.json
- disk_space.json
- disk.json
- load_avg.json
- memory.json

Tip: Set up OS metrics dashboard graphs (page 325) in OpsCenter monitoring. Refer to the OpsCenter metrics reference (page 358) for information about the available OS metrics (page 349). Also, view available memory in use, load, and more information in the Node Details (page 230) dialog in OpsCenter monitoring.

Example:

cpu.json

```
{
   "$user" : 2.5,
   "$nice" : 0.0,
   "$system" : 1.0,
   "$iowait" : 0.0,
   "$steal" : 0.0,
   "$idle" : 96.5
}
```

disk_space.json

```
{
   "free" : {
      "/dev/vda1" : 2.59
   },
   "used" : {
      "/dev/vda1" : 27.51
   }
}
```
Using OpsCenter

disk.json shows writes per second, reads per second, average request size, and so forth when data is available. This file contains metrics for the performance of I/O-related devices on a system. The output is similar to that when the `iostat` command is run.

```json
{
    "w/s" : {
        "vda" : 0.0
    },
    "await" : {
        "vda" : 0.0
    },
    "w-await" : {
        "vda" : 0.0
    },
    "wMB/s" : {
        "vda" : 0.0
    },
    "wrqm/s" : {
        "vda" : 0.0
    },
    "rMB/s" : {
        "vda" : 0.0
    },
    "r-await" : {
        "vda" : 0.0
    },
    "%util" : {
        "vda" : 0.0
    },
    "rrqm/s" : {
        "vda" : 0.0
    },
    "r/s" : {
        "vda" : 0.0
    },
    "svctm" : {
        "vda" : 0.0
    },
    "avgrq-sz" : {
        "vda" : 0.0
    },
    "avgqu-sz" : {
        "vda" : 0.0
    }
}
```

load_avg.json
memory.json

```json
{
    "used" : 4800,
    "free" : 201,
    "shared" : 0,
    "buffers" : 69,
    "cached" : 2913
}
```

process limits file

Path: `/nodes/node_folder_name/process_limits`

**Note:** Located underneath the `/nodes/os-metrics` folder but not within it.

Example:

```
```

solr folder index size json file

Path: `/nodes/node_folder_name/solr/index_size.json`

Contains the `index_size.json` file. If the node is not configured as a solr workload type, this file is empty.

See also (page 300) the `/solr` folder in the `/conf` directory.

```json
{
    "ax.account_freq_accessed" : 4523176,
    "ax.account" : 6106829,
    "ax.tn_activation_event" : 35541859,
    "ax.tn_by_partition" : 4282176,
    "ax.account_recent_accessed" : 274820,
    "cdr.call_details" : 19409157,
    "ax.account_history" : 2191447655,
    "ax.rate_center_by_prefix" : 36048878,
    "ax.management_user" : 153750
}
```
Using OpsCenter

opscenterd folder of diagnostic files

This section of the diagnostic tarball directory contains files that provide information relevant to the OpsCenter daemon opscenterd.

Agent requests json file

Path: /opscenterd/agent_requests.json

The `agent_requests.json` file lists a success or failure status for the agent requests associated with each node.

```
{
    "10.200.175.206": "success",
    "10.200.175.207": "success"
}
```

Agent status json file

Path: /opscenterd/agent_status.json

The `agent_status.json` file lists status for the agent associated with each node. Similar information can be viewed in the Agent Status (page 230) UI of OpsCenter. An excerpt:

```
{
    "10.200.175.206": {
        "agent_install_type": "package",
        "agent_status": {
            "condition": "ALL_OK",
            "http": {
                "status": "up",
                "updated_at": 1502135084
            },
            "install_status": {
                "error-message": null,
                "state": null
            },
            "jmx": {
                "status": "up",
                "updated_at": 1502135084
            }
        }
    },
    ...
```

Best practice rules json file

Path: /opscenterd/best_practice_rules.json

The `best_practice_rules.json` file lists status for the enabled Best Practice Rules. For more information, see Best Practice Service (page 502). An excerpt:

```
{
```
"check-2i-cardinality": {
  "agents-are-compatible": true,
  "alert-level": "alert",
  "category": "Performance Service - Table Metrics",
  "description": "Checks for secondary indexes with too many distinct values.",
  "display-name": "Secondary indexes cardinality",
  "enabled_by_default": true,
  "errors": {
    "node-errors": [
      "10.200.175.206",
      "10.200.175.207"
    ]
  },
  "importance": "low",
  "name": "check-2i-cardinality",
  "recommendation": "Consider denormalizing the indexed data.",
  "run_time": "2017-08-08 19:00:37.640000",
  "scope": "cluster-and-node",
  "status": "Failed",
  "suggested_interval": "hourly",
  "version": "5.0.0"
},
...

Clusters folder config files

Path: /opscenterd/clusters/cluster_name.conf

Subfolder of the cluster config files for each cluster in an OpsCenter instance. For details about the possible configurations, see Cluster configuration properties (page 196). An excerpt:

```
[destinations]
active =

[kerberos]
default_service =
opscenterd_client_principal =
opscenterd_keytab_location =
agent_keytab_location =
agent_client_principal =

[agents]
ssl_keystore_password =
ssl_keystore =
backup_staging_dir = /tmp

[jmx]
password =
port = 7199
username =
```
Warning: The AWS credentials and bucket names are stored in cluster_name.conf (with the exception of ad hoc backups). Be sure to use proper security precautions to ensure that this file is not readable by unauthorized users.

Conf json file

Path: /opscenterd/conf.json

The conf.json file is a JSON representation of the config that was passed into the in-memory representation of a cluster in opscenterd. The contents represent opscenterd.conf. An excerpt (note the diagnostic_tarball_download_timeout):

```json
{
    "agent_config": {},
    "agents": {
        "agent_aggregation_flush": "600",
        "agent_certfile": "/var/lib/opscenter/ssl/agentKeyStore.der",
        "agent_install_mute_period": "120",
        "agent_install_poll_period": "5",
        "agent_install_timeout_period": "1800",
        "agent_keyfile": "/var/lib/opscenter/ssl/agentKeyStore",
        "agent_keyfile_raw": "/var/lib/opscenter/ssl/agentKeyStore.key",
        "api_port": "61621",
        "backup_staging_dir": "/tmp",
        "call_agent_retry": "3",
        "concurrent_agent_requests": "10",
        "concurrent_settings_requests": "10",
        "concurrent_snapshot_list_requests": "1",
        "config_sleep": "420",
        "diagnostic_tarball_download_timeout": "120",
        "ec2_metadata_api_host": "169.254.169.254",
        "http_poll_period": "60",
        "http_timeout": "10",
        "incoming_interface": "0.0.0.0",
        "incoming_port": "61620",
        "not_seen_threshold": "180",
        "remote_backup_region": "us-west-1",
        "restore_req_update_period": "",
        "scp_executable": "/usr/bin/scp",
        "snapshot_wait": "60",
        "ssh_executable": "/usr/bin/ssh",
        "ssh_keygen_executable": "/usr/bin/ssh-keygen",
        "ssh_keyscan_executable": "/usr/bin/ssh-keyscan",
        "ssh_port": "22",
        "ssh_sys_known_hosts_file": "/etc/ssh/ssh_known_hosts",
        "ssh_user_known_hosts_file": "/.ssh/known_hosts",
        "ssl_certfile": "/var/lib/opscenter/ssl/opscenter.der",
        "ssl_keyfile": "/var/lib/opscenter/ssl/opscenter.key",
        "ssl_keystore": "",
        "ssl_keystore_password": ""
    }
}
```
Using OpsCenter

```
"storage_ssl_keystore": ",
"storage_ssl_keystore_password": ",
"tmp_dir": "/usr/share/opscenter/tmp/",
"use_ssl": "False"
},
"authentication": {
  "authentication_method": "DatastaxEnterpriseAuth",
  "enabled": "False",
  "passwd_db": "/etc/opscenter/passwd.db",
  "password_hash_type": "bcrypt+blake2b-512",
  "sqlite_connection_timeout": "5",
  "sqlite_max_active_connections": "200",
  "sqlite_timeout": "10",
  "timeout": "0"
},
"backups": {
  "failure_threshold": "50",
  "restore_init_throttle": "20",
  "restore_sleep": "5"
},
"bestpractice": {
  "results_ttl": "2419200"
},
...
```

gc logs

Path: /opscenterd/gc.log.n

The gc logs record garbage collection activity. Look at the logs marked as current first.

The number and max size of the GC log files are configurable via JVM command-line parameters (page 174). The default (used by the OpsCenter start/stop script) is to allow for no more than 5 log files, each with a maximum size of 1M. The gc logs are named gc.log.0, gc.log.1, gc.log.2, gc.log.3, and gc.log.4.

An excerpt:

```
2017-08-08 21:51:45 GC log file created /var/log/opscenter/gc.log.4
Java HotSpot(TM) 64-Bit Server VM (25.40-b25) for linux-amd64 JRE (1.8.0_40-b25), built on Feb 10 2015 21:29:53 by "java_re" with gcc 4.3.0 20080428 (Red Hat 4.3.0-8)
Memory: 4k page, physical 8176868k(185744k free), swap 0k(0k free)
```
Using OpsCenter

logback

Path: /opscenterd/logback.xml

The logback.xml configuration file for OpsCenter. See OpsCenter logback.xml configuration (page 215) for more details.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!--
Logback configuration file for OpsCenter.

Common options that you may want to change include:

    file - This is the name and location of the active log file that is currently
           being written to. This maps to the log_path property in previous versions
           of OpsCenter. If you change this property, you may want to also change
           fileNamePattern.

    fileNamePattern - This is the name, location and pattern of log files after they
                        exceed the rolling policy. If you change this property, you may want to also
                        change file.

    maxIndex - This is the number of rolled log files to keep. This maps to the max_rotate
               property in previous versions of OpsCenter. The default value is 10.
```
maxFileSize - This is the file size that will cause the current log file to roll into an archived file. This maps to the log_length property in previous versions of OpsCenter. The default is '10MB'.

level - This is the minimum logging level that will be included in the log files along with all higher logging levels. Valid values are TRACE, DEBUG, INFO, WARN and ERROR. Unlike previous versions of OpsCenter logging, each logger can have a different level associated with it. Changing the level property on the <root> element is equivalent to setting the level property in previous versions of OpsCenter.

Additional details on advanced configuration options can be found in the Logback manual at http://logback.qos.ch/manual/configuration.html.

```xml
<configuration>
  <appender name="opscenterd_log" class="ch.qos.logback.core.rolling.RollingFileAppender">
    <file>/var/log/opscenter/opscenterd.log</file>
    <encoder>
      <charset>UTF-8</charset>
      <pattern>%date{ISO8601, UTC} [%X{cluster_id:-opscenter}]
        %5level: %msg \n(%thread)\n%exception{20}</pattern>
    </encoder>
    <rollingPolicy class="ch.qos.logback.core.rolling.FixedWindowRollingPolicy">
      <fileNamePattern>/var/log/opscenter/opscenterd.%i.log</fileNamePattern>
      <minIndex>1</minIndex>
      <maxIndex>10</maxIndex>
    </rollingPolicy>
    <triggeringPolicy class="ch.qos.logback.core.rolling.SizeBasedTriggeringPolicy">
      <maxFileSize>10MB</maxFileSize>
    </triggeringPolicy>
  </appender>

  <appender name="STDOUT" class="ch.qos.logback.core.ConsoleAppender">
    <encoder>
      <charset>UTF-8</charset>
      <pattern>%date{ISO8601, UTC} [%X{cluster_id:-opscenter}]
        %5level: %msg \n(%thread)\n%exception{20}</pattern>
    </encoder>
    <filter class="ch.qos.logback.classic.filter.ThresholdFilter">
      <level>INFO</level>
    </filter>
  </appender>
</configuration>
```
<appender name="repair_log"
class="ch.qos.logback.classic.sift.SiftingAppender">
  <discriminator>
    <key>cluster_id</key>
    <defaultValue>unknown</defaultValue>
  </discriminator>
  <sift>
    <appender name="repair_log_${cluster_id}" class="ch.qos.logback.core.rolling.RollingFileAppender">
      <file>/var/log/opscenter/repair_service/${cluster_id}.log</file>
      <encoder>
        <charset>UTF-8</charset>
        <pattern>%date{ISO8601, UTC} [%{repair_type:-repair_service}] [%5level:] %msg \(%thread\)%n%exception{20}</pattern>
      </encoder>
      <rollingPolicy class="ch.qos.logback.core.rolling.FixedWindowRollingPolicy">
        <fileNamePattern>/var/log/opscenter/repair_service/${cluster_id}.%i.log</fileNamePattern>
        <minIndex>1</minIndex>
        <maxIndex>10</maxIndex>
      </rollingPolicy>
      <triggeringPolicy class="ch.qos.logback.core.rolling.SizeBasedTriggeringPolicy">
        <maxFileSize>10MB</maxFileSize>
      </triggeringPolicy>
      <filter class="ch.qos.logback.classic.filter.ThresholdFilter">
        <level>INFO</level>
      </filter>
    </appender>
  </sift>
</appender>

<appender name="http_log"
class="ch.qos.logback.core.rolling.RollingFileAppender">
  <file>/var/log/opscenter/http.log</file>
  <encoder>
    <charset>UTF-8</charset>
    <pattern>%date{ISO8601, UTC} [%{cluster_id}] [%5level:] %msg \(%thread\)%n%exception{20}</pattern>
  </encoder>
  <rollingPolicy class="ch.qos.logback.core.rolling.FixedWindowRollingPolicy">
    <fileNamePattern>/var/log/opscenter/http.%i.log</fileNamePattern>
    <minIndex>1</minIndex>
    <maxIndex>10</maxIndex>
  </rollingPolicy>
  <triggeringPolicy class="ch.qos.logback.core.rolling.SizeBasedTriggeringPolicy">
    <maxFileSize>10MB</maxFileSize>
  </triggeringPolicy>
  <filter class="ch.qos.logback.classic.filter.ThresholdFilter">
    <level>INFO</level>
  </filter>
</appender>
Node information json file

Path: /opscenterd/node_info.json

Contains complete information about each node; including but not limited to:

- note IP
- agent JVM version
- graph enablement status
- keyspace sizes
- version information for Cassandra, DSE, Search, Spark

An excerpt:

```json
{
    "10.139.48.107": {
        "agent_jvm_version": "1.8.0_101",
        "alias": null,
        "data_held": 2113845533,
        "dc": "entcasprdtopdc1",
        "devices": {
            "commitlog": "dm_3",
            "data": [
                "dm_3"
            ],
            "other": [
                "dm_15",
                "dm_14",
                "dm_13",
                "dm_12",
                "dm_11",
                "dm_10",
                "dm_9",
                "dm_8",
                "dm_7",
                "dm_6",
                "dm_5",
                "dm_4",
                "dm_2",
                "dm_1",
                "dm_0",
                "sda",
                "sdc",
                "sdb"
            ],
            "saved_caches": "dm_3"
        },
        "ec2": {
            "ami-id": null,
            "instance-id": null,
            "instance-type": null,
```
Using OpsCenter

OpsCenter Daemon (opscenterd) log
Path: /opscenterd/opscenterd.log

The opscenterd.log file is the log for all processes running on the OpsCenter daemon (opscenterd). An excerpt:

... 2017-07-22 04:31:00,015 [sunshine]  INFO: Scheduled job 4d55b512-1e8e-4689-844a-b38a67f5dc98 finished (MainThread) 2017-07-22 04:44:00,003 [sunshine]  INFO: Starting scheduled job 4d55b512-1e8e-4689-844a-b38a67f5dc98 (MainThread) 2017-07-22 04:44:00,011 [sunshine]  INFO: The best practice rule 'Replication factor out of bounds' has failed. (MainThread) ...

Repair Service incremental persistence json file
Path: /opscenterd/repair_service_incremental.json
The persistence file for incremental repairs (page 468). The Repair Service (page 462) periodically generates json files for job persistence. See Persisted repair state when restarting opscenterd (page 471).

Example:

```json
{"start_timestamp": 1515614238, "job_state": "success"}
```

**Repair Service subrange persistence json file**

Path: /opscenterd/repair_service_subrange.json

The persistence file for subrange repairs (page 464). The Repair Service (page 462) periodically generates json files for job persistence. See Persisted repair state when restarting opscenterd (page 471).

Example:

```json
{"start_timestamp": 1515615524, "parallel_tasks": 1, "job_state": "running"}
```

**Repair Service log**

Path: /opscenterd/repair_service.log

Note: The location of the repair_service.log in your OpsCenter installation directories rather than the diagnostic tarball directories is /var/log/opscenter for package installations, and install_location/log for tarball installations.

The repair_service.log records the Repair Service (page 462) repair processes and configuration. For more information, see Logging for the Repair Service (page 486).

An excerpt:

```
2017-08-06 16:00:41,501 [repair_service] INFO: Initializing Repair Service with configuration: [('persist_directory', './repair_service/'), ('restart_period', '300'), ('cluster_stabilization_period', '30'), ('single_task_err_threshold', '10'), ('max_parallel_repairs', '0'), ('max_pending_repairs', '5'), ('single_repair_timeout', '3600'), ('min_repair_time', '5'), ('prioritization_page_size', '512'), ('offline_splits', '256'), ('min_throughput', '512'), ('num_recent_throughput', '500'), ('error_logging_window', '86400'), ('snapshot_override', 'False'), ('ignore_keyspaces', ''), ('ignore_tables', ''), ('incremental_repair_tables', 'OpsCenter.settings, OpsCenter.backup_reports'), ('incremental_repair_datacenters', ''), ('incremental_sleep', '3600'), ('incremental_threshold', '1'), ('incremental_err_alert_threshold', '20'), ('time_to_completion_target_percentage', '65'), ('tokenranges_http_timeout', '30'), ('persist_period', '300'),
```
Creating an alternate directory for diagnostic information

When a request is issued for diagnostic information, OpsCenter requests each DataStax Agent to run diagnostic commands, and collect logs and configuration files. Each DataStax Agent reports diagnostic data to OpsCenter, which temporarily places that data in the /tmp directory to construct a compressed tarball.

Depending on the OpsCenter server configuration, the /tmp directory might not have capacity to store the collected data from the DataStax Agents. This lack of space can result in different errors (page 322), causing the diagnostic collection to fail.

Complete the following steps to provide an alternate working directory for OpsCenter to use for diagnostic data collection.

1. Create a new directory on the file system, other than /tmp.

   $ mkdir diagnostic_directory

2. Grant the OpsCenter user full read and write permissions to the directory. In some cases, it might be necessary to grant permissions on the parent directory as well.

   $ sudo chmod -R 660 diagnostic_directory

3. Open the opscenter script in a terminal or text editor.
   - Package installs: /usr/share/opscenter/bin
   - Tarball installs: install_location/bin

   $ vim opscenter

4. Add the following line before the line indicating RUN_IN_BACKGROUND=1.

   export TMPDIR="path_to/diagnostic_directory"

5. Save and close the opscenter script.

6. Restart OpsCenter (page 75) for the changes to take effect.

7. Download diagnostic data (page 286) as a compressed tarball.

   Example: Errors caused by lack of space in temporary file system

   Depending on the version of OpsCenter installed, the following errors can display if insufficient space is available to generate the diagnostic tarball.
Not enough space to store all data for diagnostic tarball

Can't write out diagnostic file for OpsCenter machine, because it will fill up a disk

In addition to the previous error messages being displayed in the OpsCenter console, the following messages can be reported in opscenterd.log.

Not enough space to store all data for diagnostic tarball

Not enough space to write the diagnostic file out on OpsCenter machine

**Generating a cluster report**

To generate a report about the cluster being monitored, click **Help#Report** at the top right menu of the OpsCenter interface.

The report shows the version of OpsCenter, number of clusters and nodes being managed, total CPU cores being managed, total gigabytes of storage used, name of the cluster, and information about nodes in the cluster.

<table>
<thead>
<tr>
<th>Managed Clusters Report</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Report Date:</strong> 2016-05-25</td>
</tr>
<tr>
<td><strong>OpsCenter Version:</strong> 6.0.0</td>
</tr>
</tbody>
</table>

- **Total clusters managed:** 1
- **Total nodes managed:** 2
- **Total CPU cores managed:** 4
- **Total storage used:** 0 GB

<table>
<thead>
<tr>
<th>Cluster Name: prodwest</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Number of Nodes:</strong> 2</td>
</tr>
</tbody>
</table>

**Node Listing:**

- **Name:** agrodwest-ef61347-1 - 10.260.182.220
  - **Cassandra Version:** 2.1.14.1272
  - **DataStax Enterprise Version:** 4.8.7
  - **Available Memory:** 7985 MB
  - **Number of CPU Cores:** 2
  - **Operating System:** Linux
  - **Space Used:** 0 GB / 0 GB

- **Name:** agrodwest-ef61347-2 - 10.260.182.221
  - **Cassandra Version:** 2.1.14.1272
  - **DataStax Enterprise Version:** 4.8.7
  - **Available Memory:** 7985 MB
  - **Number of CPU Cores:** 2
  - **Operating System:** Linux
  - **Space Used:** 0 GB / 0 GB

The node information includes:

- Node name and IP address
- Cassandra and DataStax Enterprise software versions
• Available memory
• Number of CPU cores
• Operating system running on each node
• Space used

You can print the report or save as a file format supported by your browser.

Dashboard performance metrics

Monitor performance metrics in the OpsCenter Dashboard. Real-time and historical performance metrics are available at different granularities: cluster-wide, per node, per table (column family), or per storage tier.

Tip: For more comprehensive monitoring of DSE cluster performance, take advantage of the features provided by the OpsCenter Performance Service (page 521). The OpsCenter Performance Service combines OpsCenter metrics with CQL-based diagnostic tables populated by the DSE Performance Service to help understand, tune, and optimize cluster performance.

Performance metrics overview

Select Dashboard to view these types of metrics:

• Cluster Performance Metrics
• Pending Task Metrics
• Table (Column Family) Metrics

When adding a graph to the dashboard, choose the metric and the source that OpsCenter uses to collect the data for the graph:

• Cluster-wide
• All nodes
• The node running OpsCenter

Several commonly used performance metrics graphs are displayed initially. Data appears in the graphs after you set alerts.

Click the magnifying glass icon at the top left of a graph to open it in a larger dialog for easier viewing of the details.

Note: When a graph is zoomed, it does not auto-update.

You can delete, clone, rename, choose the default view of graphs, and share graphs with other users.

Export dashboard configurations to import into other clusters or OpsCenter instances.
For automated guidance with setting up performance monitoring metrics for DataStax Enterprise, see the Performance Service (page 521).

**Working with metrics performance graphs**

Add, edit, or remove performance metrics graphs on the OpsCenter Monitoring Dashboard. After an upgrade, adjust or remove graphs that contain unknown obsolete metrics.

Graphs can be added containing multiple metrics provided the metrics use the same unit. For example, a graph can contain multiple metrics showing utilization as a percentage, like CPU and disk utilization. Other metrics such as write or read requests for a cluster or the operating system load for a node cannot be added to the utilization percentage graph. Metrics can be added to a graph for a cluster or for one or more nodes.

**Note:** After an upgrade, some obsolete metrics might become unknown to the current version of OpsCenter. Unknown metrics are indicated with a red warning icon by the graph and metric titles. If a graph does not contain any known metrics, the graph displays **No Valid Metrics** as its title.

Remove (page 328) the obsolete graph, or remove any obsolete metrics from dashboard graphs by deleting the unknown metric in the Metrics On This Graph dialog.

1. Click *cluster name*#Dashboard from the left navigation panes.
2. Click **Add Graph**.

The Add Metric dialog appears.

3. Select the metric to add from the **Metric** list.

   **Tip:** To search for a metric, begin typing the metric you want to search for and matching metrics populate in the list. To view its description, hover over a metric. To make multiple selections, press and hold the **Cmd** key (Mac) or **Ctrl** key (Windows/Linux) to keep the list open for multiple selections.

4. Select the nodes to monitor in the **Node** list. Select an individual node, multiple nodes, all nodes within a datacenter, all nodes, or cluster-wide as appropriate.
5. To specify particular tables, click **Table**.

6. Click **Add Metric**.

   The **Metrics on this graph** dialog appears.

7. Make any changes to the graph metrics:
   - To add additional metrics that are measured using the same unit, click the **Add Metric** link.
   - To edit metric details such as the applicable nodes or tables, click the **Edit** icon.
   - To delete a metric, click the **Trash** icon.

8. When you are done, click **Save Graph** to display the graph showing the defined metrics.

9. To edit the metrics displayed in a graph, click the menu on the upper right next to the graph title and click **Edit Graph**.
Using OpsCenter

Hover over the metric in the graph legend to view its descriptions.

10. To delete a graph, click **Delete Graph** from the graph menu.

11. There are also widgets that display information on Alerts, Cluster Health, and Storage Capacity. To enable or disable the Alerts, Cluster Health, or Storage Capacity widgets, click **Add Widget** and select the widget you want to enable or disable.

Organizing performance metrics presets

Clone, rename, share, or delete performance metrics views on the Dashboard. Saving groups of graphs with named presets allows customizing, organizing, and viewing different groups of related metrics graphs for analysis. *Add metrics in performance graphs (page 325)* for each preset view as you prefer.

1. Click **Dashboard** in the left navigation pane.

2. At the top of the Dashboard page, hover on the preset tab and click the drop-down arrow to open the menu.
3. To clone a preset view of metrics graphs:
   
a. Click **Clone**.

b. For multiple clusters, select either **To This Cluster** to clone the preset view for the current cluster, or **To Different Cluster** to clone the view to a different cluster.

c. In the Save Preset dialog, enter a name for the preset and click **Save**.
   - If the preset already exists in the destination cluster, a dialog prompts you to overwrite or cancel.
   - If there are any issues with incompatible schema or metrics, a warning appears. Click **Continue Clone** to proceed with cloning the preset, or click **Cancel**.

   ![Warning: Incompatible Graphs]

   After the clone, rectify any issues displayed in the Error Loading Graph panels by either editing the graph to remove the incompatible metrics or by removing the graph entirely.

4. To set the default preset, click **Make Default**.

5. To rename a preset, click **Rename**, enter a new name and click **Save**.

6. To delete a preset, click **Delete**. The original installed default view cannot be deleted.

7. (Admins only) To share a preset view, click **Share with all users**. A globe icon in the view tab indicates the preset view is visible to all users.
The Share... menu option is not available if authentication is disabled.

8. To view another preset, click the preset name tab at the top of the Dashboard. Tabs appear in alphabetical order.

Exporting and importing dashboard presets

Export dashboard configurations to conveniently import into other clusters or other OpsCenter instances. The dashboard configuration exports and imports as a JSON file.

Note: The export and import of presets is currently a labs feature. To enable the feature, add the following to opscenterd.conf and restart opscenterd (page 75):

```
[labs]
enable_dashboard_preset_import_export = True
```

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

- Package installations: /etc/opscenter/opscenterd.conf
- Tarball installations: install_location/conf/opscenterd.conf

1. Go to the Dashboard tab that you want to export. Hover on the preset tab and click the drop-down arrow to open the menu. Click Export from the preset menu.

The dashboard preset .json file downloads to your computer.

2. Go to the cluster or instance into which you want to import the presets.

3. Click the Import link on the Dashboard.
The Import Dashboard Preset dialog appears.

4. Enter a name for the preset.

5. Click Choose File and select the exported .json file.

6. Click Import.

   The dashboard preset is imported into its new location.

**Cluster performance metrics**

Cluster metrics monitor cluster performance at a high level. Cluster metrics are aggregated across all nodes in the cluster. OpsCenter tracks a number of cluster-wide metrics for read performance, write performance, memory, and capacity. Watching for variations in cluster performance can signal potential performance issues that might require further investigation. For general performance monitoring, watch for spikes in read and write latency, along with an accumulation of pending operations. Drilling down on high-demand tables can further pinpoint the source of performance issues with an application.

**Data Size [data-load]**
The live disk space used by all tables on a node.

**Heap Used [heap-used]**
Average amount of Java heap memory used.

**JVM CMS Collection Count [cms-collection-count]**
Number of concurrent mark sweep garbage collections performed per second.

**JVM CMS Collection Time [cms-collection-time]**
Average number of milliseconds spent performing CMS garbage collections per second.

**JVM G1 Old Collection Count [g1-old-collection-count]**
Number of G1 old generation garbage collections performed per second.

**JVM G1 Old Collection Time [g1-old-collection-time]**
Average number of milliseconds spent performing G1 old generation garbage collections per second.

**JVM G1 Young Collection Count [g1-young-collection-count]**
Number of G1 young generation garbage collections performed per second.

**JVM G1 Young Collection Time [g1-young-collection-time]**
Average number of milliseconds spent performing G1 young generation garbage collections per second.

**JVM ParNew Collection Count [par-new-collection-count]**
Number of ParNew garbage collections performed per second. ParNew collections pause all work in the JVM but should finish quickly.

**JVM ParNew Collection Time [par-new-collection-time]**
Average number of milliseconds spent performing ParNew garbage collections per second. ParNew collections pause all work in the JVM but should finish quickly.

**Read Requests [read-ops]**
The number of read requests per second on the coordinator nodes, analogous to client reads. Monitoring the number of requests over a given time period reveals system read workload and usage patterns.

**Read Request Latency (percentiles) [read-histogram]**
The min, median, max, 90th, and 99th percentiles of a client reads. The time period starts when a node receives a client read request, and ends when the node responds back to the client. Depending on consistency level and replication factor, this may include the network latency from requesting the data’s replicas.

**Total Bytes Compacted [total-bytes-compacted]**
Number of bytes compacted per second.

**Total Compactions [total-compactions-completed]**
Number of sstable scans per second that could result in a compaction.

**Write Requests [write-ops]**
The number of write requests per second on the coordinator nodes, analogous to client writes. Monitoring the number of requests over a given time period reveals system write workload and usage patterns.

### Pending task metrics

Pending task metrics track requests that have been received by a node but are waiting to be processed. An accumulation of pending tasks on a node can indicate a potential bottleneck in performance and should be investigated.
Cassandra maintains distinct thread pools for different stages of execution. Each of these thread pools provide granular statistics on the number of pending tasks for that particular process. Accumulating pending tasks is indicative of a cluster that is not keeping up with the workload. Pending tasks are usually caused by a lack of (or failure of) cluster resources such as disk bandwidth, network bandwidth, or memory.

**Pending task metrics for writes**

Pending task metrics for writes indicate that write requests are arriving faster than they can be handled.

**Flushes Pending**

The flush process flushes memtables to disk as SSTables. This metric shows the number of memtables queued for the flush process. The optimal number of pending flushes is 0 (or at most a very small number). A value greater than 0 indicates either I/O contention or degrading disk performance (see disk metrics such as disk latency, disk throughput, and disk utilization for indications of disk health).

**Repl. (Replicate) on Write Tasks Pending**

When an insert or update to a row is written, the affected row is replicated to all other nodes that manage a replica for that row. This is called the `ReplicateOnWriteStage`. This metric tracks the pending tasks related to this stage of the write process. During low or moderate write load, you should see 0 pending replicate on write tasks (or at most a very low number). A continuous high number signals a need to investigate disk I/O or network contention problems.

**Pending task metrics for reads**

Pending read and compaction tasks indicate I/O contention and can manifest in degrading read performance.

**Read Requests Pending**

The number of read requests that have arrived into the cluster but are waiting to be handled. During low or moderate read load, you should see 0 pending read operations (or at most a very low number). A continuous high number of pending reads signals a need for more capacity in a cluster or to investigate disk I/O contention. Pending reads can also indicate an application design that is not accessing data in the most efficient way possible.

**Read Repair Tasks Pending**

The number of read repair operations that are queued and waiting for system resources in order to run. The optimal number of pending read repairs is 0 (or at most a very small number). A value greater than 0 indicates that read repair operations are in I/O contention with other operations. If this graph shows high values for pending tasks, this may suggest the need to run a node repair to make nodes consistent. Or, for tables where your requirements can tolerate a certain degree of stale data, you can lower the value of the table parameter `read_repair_chance`.

**Compactions Pending**

...
An upper bound of the number of compactions that are queued and waiting for system resources in order to run. This is a worst-case estimate. The compactions pending metric is often misleading. An unrealistic, high reading often occurs. The optimal number of pending compactions is 0 (or at most a very small number). A value greater than 0 indicates that read operations are in I/O contention with compaction operations, which usually manifests itself as declining read performance. This is usually caused by applications that perform frequent small writes in combination with a steady stream of reads. If a node or cluster frequently displays pending compactions, that is an indicator that you might need to increase I/O capacity by adding nodes to the cluster. You can also try to reduce I/O contention by reducing the number of insert/update requests (have your application batch writes for example), or reduce the number of SSTables created by increasing the memtable size and flush frequency on your tables.

Pending task metrics for cluster operations

Pending task metrics for cluster operations can indicate a backup of cluster operational processes such as those maintaining node consistency, system schemas, fault detection, and inter-node communications. Pending tasks for resource-intensive operations such as repair, bootstrap, or decommission are normal and expected while that operation is in progress, but should continue decreasing at a steady rate in a healthy cluster.

**TP: Manual Repair Tasks Pending [pending-anti-entropy-stage]**
Repair tasks pending, such as handling the merkle tree transfer after the validation compaction.

**TP: Gossip Tasks Pending [pending-gossip-stage]**
Number of gossip messages and acknowledgments queued and waiting to be sent or received.

**Hinted handoff pending**
While a node is offline, other nodes in the cluster save hints about rows that were updated during the time the node was unavailable. When a node comes back online, its corresponding replicas begin streaming the missed writes to the node to catch it up. The hinted handoff pending metric tracks the number of hints that are queued and waiting to be delivered after a failed node is back online again. High numbers of pending hints are commonly seen when a node is brought back online after some downtime. Viewing this metric can help you determine when the recovering node has been made consistent again. Hinted handoff is an optional feature of Cassandra. Hints are saved for a configurable period of time (an hour by default) before they are dropped. This prevents a large accumulation of hints caused by extended node outages.

**TP: Migrations Pending [pending-migration-stage]**
Number of pending tasks from system methods that modified the schema.

**TP: Misc. Tasks Pending [pending-misc-stage]**
Number of pending tasks from infrequently run operations, such as taking a snapshot or processing the notification of a completed replication.

**Streams Pending**
The progress of rows of data being streamed from the sending node. Streaming of data between nodes happens during operations such as bootstrap and decommission when one node sends large numbers of rows to another node.

Table performance metrics

Table metrics allow drilling down and locating specific areas of application workloads that are the source of performance issues. If you notice a performance trend at the OS or cluster level, viewing table metrics can provide a more granular level of detail.

The metrics for KeyCache Hits, RowCache Hits, and SSTable Size can only be viewed on a single table at a time. Otherwise, all table metrics are available for specific tables as well as for all tables on a node. In addition to monitoring read latency, write latency and load on a table, monitor the hit rates on the key and row caches for tables that rely on caching for performance. The more requests that are served from the cache, the faster the response times. Viewing SSTable Size and SSTable Count for a specific table (or counts for all tables) can help with compaction tuning.

OpsCenter has been optimized to efficiently handle thousands of tables. If a table experiences a dramatic dip in performance, check the Pending Tasks metrics (page 332) for a backup in queued operations.

Table metrics are prefaced with TBL.

**TBL: Local Writes [cf-write-ops]**
Local write requests per second. Local writes update the table's memtable and appends to a commitlog.

**TBL: Local Write Latency (percentiles) [cf-local-write-latency]**
The min, median, max, 90th, and 99th percentile of the response times to write data to a table's memtable. The elapsed time from when the replica receives the request from a coordinator and returns a response.

**TBL: Local Reads [cf-read-ops]**
Local read requests per second. Local reads retrieve data from a table's memtable and any necessary SSTables on disk.

**TBL: Local Read Latency (percentiles) [cf-local-read-latency]**
The min, median, max, 90th, and 99th percentile of the response time to read data from the memtable and sstables for a specific table. The elapsed time from when the replica receives the request from a coordinator and returns a response.

**Read Request Latency (percentiles) [read-histogram]**
The min, median, max, 90th, and 99th percentiles of a client reads. The time period starts when a node receives a client read request, and ends when the node responds back to the client. Depending on consistency level and replication factor, this may include the network latency from requesting the data’s replicas.

**TBL: Live Disk Used [cf-live-disk-used]**
Disk space used by live SSTables. There might be obsolete SSTables not included.

**TBL: Total Disk Used [cf-total-disk-used]**
Disk space used by a table by SSTables, including obsolete ones waiting to be garbage collected.

**TBL: SSTables per Read (percentiles) [cf-sstables-per-read]**
The min, median, max, 90th, and 99th percentile of how many SSTables are accessed during a read. Includes sstables that undergo bloom-filter checks, even if no data is read from the sstable.

**TBL: Cell Count (percentiles) [cf-column-count]**
The min, median, max, 90th, and 99th percentile of how many cells exist in partitions for this table.

**TBL: Partition Size (percentiles) [cf-partition-size]**
The min, median, max, 90th, and 99th percentile of the size (in bytes) of partitions of this table.

**TBL: Pending Reads/Writes**
The number of pending reads and writes on a table. Pending operations are an indication that Cassandra is not keeping up with the workload. A value of zero indicates healthy throughput. If out-of-memory events become an issue in your Cassandra cluster, it might help to check cluster-wide pending tasks for operations that could be clogging throughput.

**TBL: Bloom Filter Space Used [cf-bf-space-used]**
The total size of all the SSTables' bloom filters for this table.

**TBL: Bloom Filter False Positives [cf-bf-false-positives]**
Number of bloom filter false positives per second.

**TBL: Bloom Filter False Positive Ratio [cf-bf-false-ratio]**
Percentage of bloom filter lookups that resulted in a false positive.

**TBL: Bloom Filter Off Heap [cf-bf-offheap]**
Total off heap memory used by bloom filters from all live SSTables in a table.

**TBL: Index Summary Off Heap [cf-index-summary-offheap]**
Total off heap memory used by the index summary of all live SSTables in a table.

**TBL: Compression Metadata Off Heap [cf-compression-data-offheap]**
Total off heap memory used by the compression metadata of all live SSTables in a table.

**TBL: Memtable Off Heap [cf-memtable-offheap]**
Off heap memory used by a table's current memtable.

**KeyCache Requests [key-cache-requests]**
The number of key cache requests per second. This metric only applies to SSTables created by DSE versions earlier than 6.0.

**KeyCache Hits [key-cache-hits]**
The number of key cache hits per second. This will avoid possible disk seeks when finding a partition in an SSTable. This metric only applies to SSTables created by DSE versions earlier than 6.0.

**KeyCache Hit Rate [key-cache-hit-rate]**
The percentage of key cache lookups that resulted in a hit. This metric only applies to SSTables created by DSE versions earlier than 6.0.

**RowCache Requests [row-cache-requests]**
The number of row cache requests per second.

**RowCache Hits [row-cache-hits]**
The number of row cache hits per second.

**RowCache Hit Rate [row-cache-hit-rate]**
Tiered storage performance metrics

Storage tier metrics allow drilling down into specific storage tiers for a table to monitor the data distribution.

To view a storage metric, a table and tier number must be specified. DSE identifies tiers by number. Zero represents the tier with the newest data. The number increases by one for each older tier.

**Note:** An additional implicit tier exists that represents the oldest data. For instance, if `max_tier_ages` for a table is set to 60,120, there are three tiers: Tier 0 would be for data newer than 60 seconds, tier 1 would be for data between 60 seconds and 120 seconds, and tier 2 would be for data older than 120 seconds.

For more information, see [DSE Tiered Storage](#).

**TIER: Max Data Age [cf-tier-max-data-age]**

Timestamp in local server time that represents an upper bound to the newest piece of data stored in the SSTable. When a new SSTable is flushed, it is set to the time of creation. When an SSTable is created from compaction, it is set to the max of all merged SSTables.

**TIER: sstables [cf-tier-sstables]**

Number of SSTables in a tier for a table.

**TIER: Total Disk Used [cf-tier-size]**

Disk space used by a table by SSTables for the tier.

Configuring tiered storage metric graphs

Configure tiered storage graphs to visually monitor data distribution and performance metrics for storage tiers. Add graphs for the following metrics:

- **TIER: Max Data Age** *(page 369)*
- **TIER: Total Disk Used** *(page 369)*
- **TIER: SSTables** *(page 369)*

Prerequisites:

- For centralized configuration convenience, configure the strategy and tiers *(page 610)* in Lifecycle Manager and run a configuration job *(page 576)* to push the configuration to all applicable nodes.
- Apply tiered storage to a table schema and define the maximum age of data in each tier:

```
CREATE TABLE ks.tbl (k INT, c INT, v INT, PRIMARY KEY (k, c))
WITH COMPACTON= '{'class':'TieredCompactionStrategy',
    'tiering_strategy': 'DateTieredStorageStrategy',
    'config': 'strategy1',
    'max_tier_ages': '3600,7200'};
```
See DSE Tiered Storage for complete details.

1. Click **Cluster#Dashboard**.

2. Clone *(page 329)* the Default preset tab and give it a name such as **Tiered storage**.

3. Click **Add Graph**.

   The Add Metric dialog appears.

4. Complete the dialog for each metric graph:
   a. Choose the TIER metric from the **Metric** list.
   b. Choose the nodes to monitor from the **Node** list.
   c. Choose the tables to monitor from the **Tables** list.
   d. Choose the tiers to monitor from the **Tiers** list.
   e. Click **Add Metric**.

   The graph appears in the dashboard.
f. Repeat these steps for each tier metric and the corresponding nodes, tables, or tiers that you need to monitor.

### Configuring tiered storage alerts

Configure tiered storage alerts to monitor data distribution and performance metrics for storage tiers. Add graphs for the following metrics:

- **TIER: Total Disk Used** *(page 369)*
- **TIER: SSTables** *(page 369)*

**Note:** Alerts are not supported for **TIER: Max Data Age**.

**Prerequisites:**

- For centralized configuration convenience, configure the strategy and tiers *(page 610)* in Lifecycle Manager and *run a configuration job* *(page 576)* to push the configuration to all applicable nodes.
- Apply tiered storage to a table schema and define the maximum age of data in each tier:

```java
CREATE TABLE ks.tbl (k INT, c INT, v INT, PRIMARY KEY (k, c))
WITH COMPACTION={'class':'TieredCompactionStrategy',
'tiering_strategy': 'DateTieredStorageStrategy',
'config': 'strategy1',
'max_tier_ages': '3600,7200'};
```

See [DSE Tiered Storage](#) for complete details.

1. Click **Alerts** from the main OpsCenter monitoring menu.
The **Active Alerts** dialog appears.

2. Click **Manage Alerts** > **Add Alert**.
   The Add Alert dialog appears.

3. In the **Notify me when** menu, select **Advanced** > **DSE** > **TIER: metric name**.

4. Specify the notification criteria, the table, and the tier on which to notify as well as the notification frequency.
5. Click **Save Alert**.

### Message latency metrics

Monitor messaging latency by adding dashboard graphs ([page 341](#)) and alerts ([page 343](#)) for latency metrics at the datacenter or node level. Messaging latency metrics are available for DSE versions 5.1.0 and later.

**Datacenter Messaging Latency** [cross-dc-latency]

The min, median, max, 90th, and 99th percentiles of the message latency between nodes in the same or different destination datacenter. This metric measures how long it takes a message from a node in the source datacenter to reach a node in the destination datacenter. Selecting a destination node within the source datacenter yields lower latency values.

**Node Messaging Latency** [cross-node-latency]

The min, median, max, 90th, and 99th percentiles of the latency of messages between nodes. The time period starts when a node sends a message and ends when the current node receives it.

### Adding dashboard graphs for datacenter and node messaging latency

Add graphs to the OpsCenter Monitoring Dashboard to monitor latency between datacenters and nodes. Messaging latency metrics are available for DSE versions 5.1.0 and later.

1. Click **Cluster#Dashboard**.

2. Click **Add Graph**.

   The **Add Metric** dialog appears.

3. Complete the dialog for each metric graph:

   a. Choose **Datacenter Messaging Latency** from the **Metric** list.

   ![Add Metric Dialog](image)

   b. Choose the nodes to monitor from the **Node** list. Available options:
Using OpsCenter

- Cluster-wide
- All nodes
- Datacenter (aggregate)
- Datacenter (individual nodes)
- A single specific node

**Note:** If multiple selections are made in the Node list, reported latencies are aggregated for all selections.

c. Select the datacenter from the Source list.

d. Click Add Metric.

e. Repeat the procedure for the Node Message Latency metric. Selecting a source datacenter is not applicable to node-level graphs.

![Add Metric](image)

f. Add as many variations of these metrics as required for your environment. When finished, click Save Graph.

   The graphs appear in the dashboard.
Adding alerts for DC and node message latency

Add alerts for notification when datacenter and node messaging latencies reach unacceptable levels for your environment. Messaging latency metrics are available for DSE versions 5.1.0 and later.

1. Click **Alerts** from the main OpsCenter monitoring menu.
   
   The **Active Alerts** dialog appears.

2. In the **Active Alerts** dialog, click **Manage Alerts**.
   
   The **Manage Alerts Rules** dialog appears.

3. If applicable (multiple clusters are monitored in the OpsCenter instance), select the cluster from the **Cluster** list.

4. Click **Add Alert**.
   
   The **Add Alert** dialog appears.

5. In the **Notify me when** list, select **Advanced#Cassandra#Datacenter Messaging Latency**.
6. Specify the notification criteria:

   a. Choose above or below. Default: above.

   b. Enter a value for ms/op (milliseconds per operation).

   c. Select the elapsed time for how long the condition should be in the defined state before alerting. Default: 5 minutes.

   d. Select the percentile. Choose: min (default), 50th, 90th, 99th, or max.

   e. Select the Source Datacenter.

   f. Select the notification frequency from Notify every. Default: 5 minutes.

   g. Click Save Alert.

      The alert is added to the Manage Alert Rules dialog.
7. Repeat this procedure for the **Node Messaging Latency** alert. **Select Source Datacenter** is not applicable to node-level alerts.

When the alert criteria is met, the fired alert is recorded in the Event Log and sent via any other alert mechanism configured *(page 138)* for your environment, such as email or posted URLs.

---

**Search performance metrics**

Metrics for monitoring DSE Search performance.

For more information, see *About DSE Search*.

**Search: Core Size [solr-index-size]**

Size of the Solr core on disk.

**Note:** The solr-index-size (displayed as Search: Core Size) metric in the OpsCenter Monitoring UI is unavailable for DSE versions 5.1.0 through 5.1.3.

**Search: Errors [solr-errors]**
Errors per second that occur for a specific Solr core/index.

**Search: Request Latency [solr-avg-time-per-req]**
Average time a search query takes in a DSE cluster using DSE Search.

**Search: Requests [solr-requests]**
Requests per second made to a specific Solr core/index.

**Search: Timeouts [solr-timeouts]**
Timeouts per second on a specific Solr core/index.

### Graph metrics

Reference of available graph metrics in OpsCenter monitoring for DSE Graph. Alerts are also available for all Graph metrics.

**Note:** DSE Graph metrics are available for DSE versions 5.0 and later. There is one exception: The Graph Request Latencies metric is only supported in OpsCenter for DSE versions 5.0.5 and later due to the underlying JMX MBean for the metric being renamed from datastore-latencies to request-latencies in DSE.

Metrics for monitoring DSE Graph in OpsCenter include:

- Cache metrics (page 346): Cache metrics bound to a specific DSE graph. Per node per graph instance metrics.
- Request Latencies (page 346): Histogram of request latencies.
- Threadpool metrics (page 347): Global-level threadpool metrics for tasks that have active, pending, and completed threads.

#### Graph cache metrics

Metrics for hits and misses to the adjacency and index caches used by DSE Graph.

**Note:** The graph must be configured at schema creation time to use the adjacency or index caches. For more information, see Caching edges and properties.

**Graph: Adjacency Cache Hits [graph-adjacency-cache-hit]**
Number of hits against the adjacency cache for this graph.

**Graph: Adjacency Cache Misses [graph-adjacency-cache-miss]**
Number of misses against the adjacency cache for this graph.

**Graph: Index Cache Hits [graph-index-cache-hit]**
Number of hits against the index cache for this graph.

**Graph: Index Cache Misses [graph-index-cache-miss]**
Number of misses against the index cache for this graph.

#### Graph transaction latencies

**Note:** The Graph Request Latencies metric is only supported in OpsCenter 6.1 and later for DSE versions 5.0.5 and later due to the underlying JMX MBean for the metric being renamed from datastore-latencies to request-latencies in DSE.

**Graph: Request Latencies [graph-request-latencies]**
The min, median, max, 90th, and 99th percentile of request latencies during the period.

Graph threadpool metrics

Graph query threads, Graph scheduled threads, Graph system threads, and Graph Gremlin worker metrics for pending, active, and completed threads in the respective threadpools. Threadpool metrics are global across all graphs.

**Graph TP: Graph Scheduled Threads Pending** [pending-graph-scheduled-threads]
Number of pending tasks in the GraphScheduledThreads thread pool.

**Graph TP: Graph Scheduled Threads Active** [active-graph-scheduled-threads]
Number of active tasks in the GraphScheduledThreads thread pool.

**Graph TP: Graph Scheduled Threads Completed** [completed-graph-scheduled-threads]
Number of tasks completed by the GraphScheduledThreads thread pool.

**Graph TP: Graph System Threads Pending** [pending-graph-system-threads]
Number of pending tasks in the GraphSystemThreads thread pool.

**Graph TP: Graph System Threads Active** [active-graph-system-threads]
Number of active tasks in the GraphSystemThreads thread pool.

**Graph TP: Graph System Threads Completed** [completed-graph-system-threads]
Number of tasks completed by the GraphSystemThreads thread pool.

**Graph TP: Gremlin Worker Threads Pending** [pending-gremlin-worker-threads]
Number of pending tasks in the GremlinWorkerThreads thread pool.

**Graph TP: Gremlin Worker Threads Active** [active-gremlin-worker-threads]
Number of active tasks in the GremlinWorkerThreads thread pool.

**Graph TP: Gremlin Worker Threads Completed** [completed-gremlin-worker-threads]
Number of tasks completed by the GremlinWorkerThreads thread pool.

Dashboard Graphs for Graph Metrics

Dashboard graphs are available for all Graph metrics in OpsCenter. Click `Cluster > Dashboard > Add Graph`. Select the metric from the **Metric** list in the **Add Metric** dialog:

![Add Metric Dialog](image)

The **Graph** list is only applicable to the per graph cache metrics *(page 346)*.

After adding metrics and clicking **Save Graph**, the graphs appear on the monitoring dashboard.
Tip: Clone (page 329) the Default preset tab and give it a name such as DSE Graph to organize your dashboard.

For longer graph titles that are truncated in the dashboard, hover over the graph title to view it in its entirety:

Graph Alerts

Alerts are available for all graph metrics. Click Alerts#Notify me when#Advanced#Graph#Graph metric in the Add Alert dialog.
Define the notification criteria for each alert required for monitoring your DSE Graph environment.

![Add Alert Window](image)

**Operating system performance metrics**

As with any database system, Cassandra performance greatly depends on underlying systems on which it is running. Monitoring Cassandra nodes for increasing disk and CPU utilization can help identify and remedy issues before performance degrades to unacceptable levels. The graphs in OpsCenter provide a quick way to view variations in OS metrics at a glance, and drill-down for specific data points. Especially in systems with heavy write loads, monitoring disk space is also important because it allows for advanced expansion planning while there is still adequate capacity to handle expansion and rebalancing operations.

System metrics are prefaced with OS.

**OS: Memory**

Shows memory usage metrics in megabytes.

- **Linux** - Shows how much total system memory is currently used, cached, buffered or free.
- **Windows** - Shows the available physical memory, the cached operating system code, and the allocated pool-paged-resident and pool-nonpaged memory.
- **Mac OS X** - Shows free and used system memory.

**OS: CPU**

Shows average percentages for CPU utilization metrics, which is the percentage of time the CPU was idle subtracted from 100 percent. CPU metrics can be useful for determining the origin of CPU performance reduction.

- **Linux** - Shows how much time the CPU devotes to system and user processes, to tasks stolen by virtual operating systems, to waiting for I/O to complete, and to processing nice tasks. High percentages of nice might indicate that other processes are crowding out Cassandra processes, while high percentages of iowait might indicate I/O contention. On fully virtualized environments like Amazon EC2, a Cassandra cluster
under load might show high steal values while other virtual processors use the available
system resources.
• Windows and Mac OS X - Shows how much time the CPU spends on user processes
  and system processes.

OS: Load

The amount of work that a computer system performs. An idle computer has a load number
of 0 and each process using or waiting for CPU time increments the load number by 1.
Any value above one indicates that the machine was temporarily overloaded and some
processes were required to wait. Shows minimum, average, and maximum OS load
expressed as an integer.

OS: Disk usage (GB)

Tracks growth or reduction in the amount of available disk space used. If this metric
indicates a growth trend leading to high or total disk space usage, consider strategies to
relieve it, such as adding capacity to the cluster. DataStax recommends leaving 30-50% free
disk space for optimal repair and compaction operations.

OS: Disk Usage (percentage)

The percentage of disk space that is being used by Cassandra at a given time. When
Cassandra is reading and writing heavily from disk, or building SSTables as the final product
of compaction processes, disk usage values may be temporarily higher than expected.

OS: Disk Throughput

The average disk throughput for read and write operations, measured in megabytes per
second. Exceptionally high disk throughput values may indicate I/O contention. This is
typically caused by numerous compaction processes competing with read operations.
Reducing the frequency of memtable flushing can relieve I/O contention.

OS: Disk Rates

• Linux and Windows - Averaged disk speed for read and write operations.
• Mac OS X - Not supported.

OS: Disk Latency

• Linux and Windows - Measures the average time consumed by disk seeks in
  milliseconds. Disk latency is among the higher-level metrics that may be useful
to monitor on an ongoing basis by keeping this graph posted on your OpsCenter
  performance console. Consistently high disk latency may be a signal to investigate
  causes, such as I/O contention from compactions or read/write loads that call for
  expanded capacity.
• Mac OS X - Not supported.

OS: Disk Request Size

• Linux and Windows - The average size in sectors of requests issued to the disk.
• Mac OS X - Not supported.
OS: Disk Queue Size
- Linux and Windows - The average number of requests queued due to disk latency issues.
- Mac OS X - Not supported.

OS: Disk Utilization
- Linux and Windows - The percentage of CPU time consumed by disk I/O.
- Mac OS X - Not supported.

Alert metrics

From the Alerts area of OpsCenter, configure alert thresholds for Cassandra cluster-wide, table, and operating system metrics. This proactive monitoring feature is available for DataStax Enterprise clusters.

Commonly watched alert metrics

Commonly watched metrics are available from the main Notify me when choice menu in the Add Alert dialog.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node down</td>
<td>When a node does not respond to requests, OpsCenter marks the node as down. To determine whether a node is down, each agent gets a list of nodes that its node suspects are down based on information from Cassandra returned via JMX. Based on that information, Opscenterd determines whether a node is truly down based on status reported by other nodes, or if a node is simply flapping and erroneously reporting all other nodes down. Nodes marked with a down status are clearly indicated in the Nodes Ring View (page 224). For even more awareness and visibility, see adding an alert for down nodes (page 242) for further instructions.</td>
</tr>
<tr>
<td>Write requests</td>
<td>The number of write requests per second. Monitoring the number of writes over a given time period can give you an idea of system write workload and usage patterns.</td>
</tr>
<tr>
<td>Write request latency</td>
<td>The response time (in milliseconds) for successful write operations. The time period starts when a node receives a client write request, and ends when the node responds back to the client.</td>
</tr>
<tr>
<td>Read requests</td>
<td>The number of read requests per second. Monitoring the number of reads over a given time period can give you an idea of system read workload and usage patterns.</td>
</tr>
<tr>
<td>Read request latency</td>
<td>The response time (in milliseconds) for successful read operations. The time period starts when a node receives a client read request, and ends when the node responds back to the client.</td>
</tr>
</tbody>
</table>
Using OpsCenter

<table>
<thead>
<tr>
<th>Metric</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU usage</td>
<td>The percentage of time that the CPU was busy, which is calculated by subtracting the percentage of time the CPU was idle from 100 percent.</td>
</tr>
<tr>
<td>Load</td>
<td>Load is a measure of the amount of work that a computer system performs. An idle computer has a load number of 0 and each process using or waiting for CPU time increments the load number by 1.</td>
</tr>
</tbody>
</table>

Advanced Cassandra alert metrics

To access Advanced Cassandra metrics, choose Advanced#Cassandra in the Add Alert dialog.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heap max</td>
<td>The maximum amount of shared memory allocated to the JVM heap for Cassandra processes.</td>
</tr>
<tr>
<td>Heap used</td>
<td>The amount of shared memory in use by the JVM heap for Cassandra processes.</td>
</tr>
<tr>
<td>JVM CMS collection count</td>
<td>The number of concurrent mark-sweep (CMS) garbage collections performed by the JVM per second.</td>
</tr>
<tr>
<td>JVM ParNew collection count</td>
<td>The number of parallel new-generation garbage collections performed by the JVM per second.</td>
</tr>
<tr>
<td>JVM CMS collection time</td>
<td>The time spent collecting CMS garbage in milliseconds per second (ms/sec).</td>
</tr>
<tr>
<td>JVM ParNew collection time</td>
<td>The time spent performing ParNew garbage collections in ms/sec.</td>
</tr>
<tr>
<td>Data size</td>
<td>The size of table data (in gigabytes) that has been loaded/inserted into Cassandra, including any storage overhead and system metadata.</td>
</tr>
<tr>
<td>Compactions pending</td>
<td>The number of compaction operations that are queued and waiting for system resources in order to run. The optimal number of pending compactions is 0 (or at most a very small number). A value greater than 0 indicates that read operations are in I/O contention with compaction operations, which usually manifests itself as declining read performance.</td>
</tr>
<tr>
<td>Total bytes compacted</td>
<td>The number of SSTable data compacted in bytes per second.</td>
</tr>
<tr>
<td>Total compactions</td>
<td>The number of compactions (minor or major) performed per second.</td>
</tr>
<tr>
<td>Metric</td>
<td>Definition</td>
</tr>
<tr>
<td>------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Flush sorter tasks pending</td>
<td>The flush sorter process performs the first step in the overall process of flushing memtables to disk as SSTables. The optimal number of pending flushes is 0 (or at most a very small number).</td>
</tr>
<tr>
<td>Flushes pending</td>
<td>The flush process flushes memtables to disk as SSTables. This metric shows the number of memtables queued for the flush process. The optimal number of pending flushes is 0 (or at most a very small number).</td>
</tr>
<tr>
<td>Gossip tasks pending</td>
<td>Cassandra uses a protocol called gossip to discover location and state information about the other nodes participating in a Cassandra cluster. In Cassandra, the gossip process runs once per second on each node and exchanges state messages with up to three other nodes in the cluster. Gossip tasks pending shows the number of gossip messages and acknowledgments queued and waiting to be sent or received. The optimal number of pending gossip tasks is 0 (or at most a very small number).</td>
</tr>
<tr>
<td>Hinted hand-off pending</td>
<td>While a node is offline, other nodes in the cluster will save hints about rows that were updated during the time the node was unavailable. When a node comes back online, its corresponding replicas will begin streaming the missed writes to the node to catch it up. The hinted hand-off pending metric tracks the number of hints that are queued and waiting to be delivered once a failed node is back online again. High numbers of pending hints are commonly seen when a node is brought back online after some down time. Viewing this metric can help you determine when the recovering node has been made consistent again.</td>
</tr>
<tr>
<td>Internal response pending</td>
<td>The number of pending tasks from various internal tasks such as nodes joining and leaving the cluster.</td>
</tr>
<tr>
<td>Manual repair tasks pending</td>
<td>The number of operations still to be completed when you run anti-entropy repair on a node. It will only show values greater than 0 when a repair is in progress. It is not unusual to see a large number of pending tasks when a repair is running, but you should see the number of tasks progressively decreasing.</td>
</tr>
<tr>
<td>Memtable postflushers pending</td>
<td>The memtable post flush process performs the final step in the overall process of flushing memtables to disk as SSTables. The optimal number of pending flushes is 0 (or at most a very small number).</td>
</tr>
<tr>
<td>Migrations pending</td>
<td>The number of pending tasks from system methods that have modified the schema. Schema updates have to be propagated to all nodes, so pending tasks for this metric can manifest in schema disagreement errors.</td>
</tr>
<tr>
<td>Miscellaneous tasks pending</td>
<td>The number of pending tasks from other miscellaneous operations that are not ran frequently.</td>
</tr>
</tbody>
</table>
Using OpsCenter

<table>
<thead>
<tr>
<th>Metric</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read requests pending</td>
<td>The number of read requests that have arrived into the cluster but are waiting to be handled. During low or moderate read load, you should see 0 pending read operations (or at most a very low number).</td>
</tr>
<tr>
<td>Read repair tasks pending</td>
<td>The number of read repair operations that are queued and waiting for system resources in order to run. The optimal number of pending read repairs is 0 (or at most a very small number). A value greater than 0 indicates that read repair operations are in I/O contention with other operations.</td>
</tr>
<tr>
<td>Replicate on write tasks pending</td>
<td>When an insert or update to a row is written, the affected row is replicated to all other nodes that manage a replica for that row. This is called the ReplicateOnWriteStage. This metric tracks the pending tasks related to this stage of the write process. During low or moderate write load, you should see 0 pending replicate on write tasks (or at most a very low number).</td>
</tr>
<tr>
<td>Request response pending</td>
<td>Streaming of data between nodes happens during operations such as bootstrap and decommission when one node sends large numbers of rows to another node. The metric tracks the progress of the streamed rows from the receiving node.</td>
</tr>
<tr>
<td>Streams pending</td>
<td>Streaming of data between nodes happens during operations such as bootstrap and decommission when one node sends large numbers of rows to another node. The metric tracks the progress of the streamed rows from the sending node.</td>
</tr>
<tr>
<td>Write requests pending</td>
<td>The number of write requests that have arrived into the cluster but are waiting to be handled. During low or moderate write load, you should see 0 pending write operations (or at most a very low number).</td>
</tr>
</tbody>
</table>

Advanced table alert metrics

To access Advanced Tables metrics, choose Advanced#Tables in the Add Alert dialog.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local writes</td>
<td>The write load on a table measured in operations per second. This metric includes all writes to a given table, including write requests forwarded from other nodes.</td>
</tr>
<tr>
<td>Local write latency</td>
<td>The response time in milliseconds for successful write operations on a table. The time period starts when nodes receive a write request, and ends when nodes respond.</td>
</tr>
<tr>
<td>Local reads</td>
<td>The read load on a table measured in operations per second. This metric includes all reads to a given table, including read requests forwarded from other nodes.</td>
</tr>
<tr>
<td>Metric</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Local read latency</td>
<td>The response time in microseconds for successful read operations on a table. The time period starts when a node receives a read request, and ends when the node responds.</td>
</tr>
<tr>
<td>Table key cache hits</td>
<td>The number of read requests that resulted in the requested row key being found in the key cache.</td>
</tr>
<tr>
<td>Table key cache requests</td>
<td>The total number of read requests on the row key cache.</td>
</tr>
<tr>
<td>Table key cache hit rate</td>
<td>The key cache hit rate indicates the effectiveness of the key cache for a given table by giving the percentage of cache requests that resulted in a cache hit.</td>
</tr>
<tr>
<td>Table row cache hits</td>
<td>The number of read requests that resulted in the read being satisfied from the row cache.</td>
</tr>
<tr>
<td>Table row cache requests</td>
<td>The total number of read requests on the row cache.</td>
</tr>
<tr>
<td>Table row cache hit rate</td>
<td>The key cache hit rate indicates the effectiveness of the row cache for a given table by giving the percentage of cache requests that resulted in a cache hit.</td>
</tr>
<tr>
<td>Table bloom filter space used</td>
<td>The size of the bloom filter files on disk.</td>
</tr>
<tr>
<td>Table bloom filter false positives</td>
<td>The number of false positives, which occur when the bloom filter said the row existed, but it actually did not exist in absolute numbers.</td>
</tr>
<tr>
<td>Table bloom filter false positive ratio</td>
<td>The fraction of all bloom filter checks resulting in a false positive.</td>
</tr>
<tr>
<td>Live disk used</td>
<td>The current size of live SSTables for a table. It is expected that SSTable size will grow over time with your write load as compaction processes continue doubling the size of SSTables. Monitor the current state of compaction for a given table using this metric together with SSTable count.</td>
</tr>
<tr>
<td>Total disk used</td>
<td>The current size of the data directories for the table including space not reclaimed by obsolete objects.</td>
</tr>
<tr>
<td>SSTable count</td>
<td>The current number of SSTables for a table. When table memtables are persisted to disk as SSTables, this metric increases to the configured maximum before the compaction cycle is repeated. Monitor the current state of compaction for a given table using this metric together with live disk used.</td>
</tr>
<tr>
<td>Pending reads and writes</td>
<td>The number of pending reads and writes on a table. Pending operations indicate Cassandra is not keeping up with the workload. A value of zero indicates healthy throughput.</td>
</tr>
</tbody>
</table>

**Advanced system alert metrics**

Configure advanced system metrics for memory, CPU, and disk metrics on Linux or Mac OS X. As with any database system, Cassandra performance greatly depends on
underlying systems on which it is running. Before configuring advanced system metric alerts, you should first have an understanding of the baseline performance of your hardware and the averages of these system metrics when the system is handling a typical workload.

To access Advanced System metrics, choose Advanced#System in the Add Alert dialog.

### Linux memory metrics

<table>
<thead>
<tr>
<th>Metric</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory free</td>
<td>System memory that is not being used.</td>
</tr>
<tr>
<td>Memory used</td>
<td>System memory used by application processes.</td>
</tr>
<tr>
<td>Memory buffered</td>
<td>System memory used for caching file system metadata and tracking in-flight pages.</td>
</tr>
<tr>
<td>Memory shared</td>
<td>System memory that is accessible to CPUs.</td>
</tr>
<tr>
<td>Memory cached</td>
<td>System memory used by the OS disk cache.</td>
</tr>
</tbody>
</table>

### Linux CPU metrics

<table>
<thead>
<tr>
<th>Metric</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Idle</td>
<td>Percentage of time the CPU is idle.</td>
</tr>
<tr>
<td>Iowait</td>
<td>Percentage of time the CPU is idle and there is a pending disk I/O request.</td>
</tr>
<tr>
<td>Nice</td>
<td>Percentage of time spent processing prioritized tasks. Nice tasks are also counted in system and user time.</td>
</tr>
<tr>
<td>Steal</td>
<td>Percentage of time a virtual CPU waits for a real CPU while the hypervisor services another virtual processor.</td>
</tr>
</tbody>
</table>
### Using OpsCenter

<table>
<thead>
<tr>
<th>Metric</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>System</td>
<td>Percentage of time allocated to system processes.</td>
</tr>
<tr>
<td>User</td>
<td>Percentage of time allocated to user processes.</td>
</tr>
</tbody>
</table>

#### Linux Disk metrics

<table>
<thead>
<tr>
<th>Metric</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disk usage</td>
<td>Percentage of disk space Cassandra uses at a given time.</td>
</tr>
<tr>
<td>Free disk space</td>
<td>Available disk space in GB.</td>
</tr>
<tr>
<td>Used disk space</td>
<td>Used disk space in GB.</td>
</tr>
<tr>
<td>Disk read throughput</td>
<td>Average disk throughput for read operations in megabytes per second.</td>
</tr>
<tr>
<td></td>
<td>Exceptionally high disk throughput values may indicate I/O contention.</td>
</tr>
<tr>
<td>Disk write throughput</td>
<td>Average disk throughput for write operations in megabytes per second.</td>
</tr>
<tr>
<td>Disk read rate</td>
<td>Averaged disk speed for read operations.</td>
</tr>
<tr>
<td>Disk write rate</td>
<td>Averaged disk speed for write operations.</td>
</tr>
<tr>
<td>Disk latency</td>
<td>Average time consumed by disk seeks in milliseconds.</td>
</tr>
<tr>
<td>Disk request size</td>
<td>Average size in sectors of requests issued to the disk.</td>
</tr>
<tr>
<td>Disk queue size</td>
<td>Average number of requests queued due to disk latency.</td>
</tr>
<tr>
<td>Disk utilization</td>
<td>Percentage of CPU time consumed by disk I/O.</td>
</tr>
</tbody>
</table>

#### Mac OS X memory metrics

<table>
<thead>
<tr>
<th>Metric</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Free memory</td>
<td>System memory that is not being used.</td>
</tr>
<tr>
<td>Used memory</td>
<td>System memory that is being used by application processes.</td>
</tr>
</tbody>
</table>

#### Mac OS X CPU metrics

<table>
<thead>
<tr>
<th>Metric</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Idle</td>
<td>Percentage of time the CPU is idle.</td>
</tr>
<tr>
<td>System</td>
<td>Percentage of time allocated to system processes.</td>
</tr>
<tr>
<td>User</td>
<td>Percentage of time allocated to user processes.</td>
</tr>
</tbody>
</table>

#### Mac OS X disk metrics

<table>
<thead>
<tr>
<th>Metric</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disk usage</td>
<td>Percentage of disk space Cassandra uses at a given time.</td>
</tr>
<tr>
<td>Metric</td>
<td>Definition</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Free space</td>
<td>Available disk space in GB.</td>
</tr>
<tr>
<td>Used disk space</td>
<td>Used disk space in GB.</td>
</tr>
<tr>
<td>Disk throughput</td>
<td>Average disk throughput for read/write operations in megabytes per second. Exceptionally high disk throughput values may indicate I/O contention.</td>
</tr>
</tbody>
</table>

**Partition name limitation for disk usage alerts**

When configuring a disk usage alert for multiple nodes, the partition names must be named the same on every node. For example, if you want to alert when disk space on your commit log partition is running low, the commit log partition must be named the same on every node. Currently, the OpsCenter UI only displays the drives from one of the nodes.

**OpsCenter Metrics Tooltips Reference**

Metrics are available to add to any graph. View descriptions of any metric by hovering over a metric in the Add Metric dialog, or by hovering over a graph legend.
The following list of metric descriptions available in tooltips is provided for your convenience:

**Write Requests [write-ops]**
The number of write requests per second on the coordinator nodes, analogous to client writes. Monitoring the number of requests over a given time period reveals system write workload and usage patterns.

**Write Request Latency (percentiles) [write-histogram]**
The min, median, max, 90th, and 99th percentiles of a client writes. The time period starts when a node receives a client write request, and ends when the node responds back to the client. Depending on consistency level and replication factor, this may include the network latency from writing to the replicas.

**Write Failures [write-failures]**
The number of write requests on the coordinator nodes that fail due to errors returned from replicas.

**Write Timeouts [write-timeouts]**
The number of server write timeouts per second on the coordinator nodes.

**Write Unavailable Errors [write-unavailables]**
The number of write requests per second on the coordinator nodes, that fail because not enough replicas are available.
Read Requests [read-ops]
The number of read requests per second on the coordinator nodes, analogous to client reads. Monitoring the number of requests over a given time period reveals system read workload and usage patterns.

Read Request Latency (percentiles) [read-histogram]
The min, median, max, 90th, and 99th percentiles of a client reads. The time period starts when a node receives a client read request, and ends when the node responds back to the client. Depending on consistency level and replication factor, this may include the network latency from requesting the data’s replicas.

Read Failures [read-failures]
The number of read requests on the coordinator nodes that fail due to errors returned from replicas.

Read Timeouts [read-timeouts]
The number of server read timeouts per second on the coordinator nodes.

Read Unavailable Errors [read-unavailables]
The number of read requests per second on the coordinator nodes, that fail because not enough replicas are available.

Non Heap Committed [nonheap-committed]
Allocated memory, guaranteed for Java nonheap.

Non Heap Max [nonheap-max]
Maximum amount that the Java nonheap can grow.

Non Heap Used [nonheap-used]
Average amount of Java nonheap memory used.

Heap Committed [heap-committed]
Allocated memory guaranteed for the Java heap.

Heap Max [heap-max]
Maximum amount that the Java heap can grow.

Heap Used [heap-used]
Average amount of Java heap memory used.

JVM CMS Collection Count [cms-collection-count]
Number of concurrent mark sweep garbage collections performed per second.

JVM ParNew Collection Count [par-new-collection-count]
Number of ParNew garbage collections performed per second. ParNew collections pause all work in the JVM but should finish quickly.

JVM CMS Collection Time [cms-collection-time]
Average number of milliseconds spent performing CMS garbage collections per second.

JVM ParNew Collection Time [par-new-collection-time]
Average number of milliseconds spent performing ParNew garbage collections per second. ParNew collections pause all work in the JVM but should finish quickly.

JVM G1 Old Collection Count [g1-old-collection-count]
Number of G1 old generation garbage collections performed per second.

JVM G1 Old Collection Time [g1-old-collection-time]
Average number of milliseconds spent performing G1 old generation garbage collections per second.

JVM G1 Young Collection Count [g1-young-collection-count]
Number of G1 young generation garbage collections performed per second.

JVM G1 Young Collection Time [g1-young-collection-time]
Average number of milliseconds spent performing G1 young generation garbage collections per second.

**Data Size [data-load]**
The live disk space used by all tables on a node.

**Total Bytes Compacted [total-bytes-compacted]**
Number of bytes compacted per second.

**Total Compactions Completed [actual-total-compactions-completed]**
Number of compaction tasks completed per second.

**Total Compactions [total-compactions-completed]**
Number of sstable scans per second that could result in a compaction.

**Compactions Pending [pending-compaction-tasks]**
Estimated number of compactions required to achieve the desired state. This includes the pending queue to the compaction executor and additional tasks that may be created from their completion.

**Task Queues [all-pending]**
Aggregate of thread pools pending queues that can be used to identify where things are backing up internally. This doesn't include pending compactions because it includes an estimate outside of the task queue or the hinted hand off queue, which can be in constant state of being on.

**Dropped Messages: All [all-dropped]**
Aggregate of all messages that have been dropped server-side due to not having been processed before their respective timeout.

**Dropped Messages: Counter Mutations [dropped-counter-mutations]**
Mutation was seen after the timeout (write_request_timeout_in_ms) so was thrown away. This client might have timed out before it met the required consistency level, but might have succeeded as well. Hinted handoffs and read repairs should resolve inconsistencies but a repair can ensure it.

**Dropped Messages: Mutations [dropped-mutations]**
Mutation was seen after the timeout (write_request_timeout_in_ms) so was thrown away. This client might have timed out before it met the required consistency level, but might have succeeded as well. Hinted handoffs and read repairs should resolve inconsistencies but a repair can ensure it.

**Dropped Messages: Reads [dropped-reads]**
A local read request was received after the timeout (read_request_timeout_in_ms) so it was thrown away because it would have already either been completed and sent to client or sent back as a timeout error.

**Dropped Messages: Ranged Slice Reads [dropped-ranged-slice-reads]**
A local ranged read request was received after the timeout (range_request_timeout_in_ms) so it was thrown away because it would have already either been completed and sent to client or sent back as a timeout error.

**Dropped Messages: Read Repairs [dropped-read-repairs]**
The Mutation was seen after the timeout (write_request_timeout_in_ms) so was thrown away. With the read repair timeout, the node still exists in an inconsistent state.

**TP: Flushes Pending [pending-flushes]**
Number of memtables queued for the flush process. A flush sorts and writes the memtables to disk.

**TP: Gossip Tasks Pending [pending-gossip-stage]**
Using OpsCenter

Number of gossip messages and acknowledgments queued and waiting to be sent or received.

**TP: Internal Responses Pending [pending-internal-response-stage]**
Number of pending tasks from internal tasks, such as nodes joining and leaving the cluster.

**TP: Manual Repair Tasks Pending [pending-anti-entropy-stage]**
Repair tasks pending, such as handling the merkle tree transfer after the validation compaction.

**TP: Cache Cleaning Pending [pending-cache-cleanup-stage]**
Tasks pending to clean row caches during a cleanup compaction.

**TP: Post Flushes Pending [pending-memtable-post-flush]**
Tasks related to the last step in flushing memtables to disk as SSTables. Includes removing unnecessary commitlog files and committing Solr-based secondary indexes.

**TP: Migrations Pending [pending-migration-stage]**
Number of pending tasks from system methods that modified the schema.

**TP: Misc. Tasks Pending [pending-misc-stage]**
Number of pending tasks from infrequently run operations, such as taking a snapshot or processing the notification of a completed replication.

**TP: Read Repair Tasks Pending [pending-read-repair-stage]**
Number of read repair operations in the queue waiting to run.

**TP: Request Responses Pending [pending-request-response-stage]**
Number of pending callbacks to execute after a task on a remote node completes.

**TP: Validation Executor Pending [pending-validation-executor]**
Pending task to read data from sstables and generate a merkle tree for a repair.

**TP: Compaction Executor Pending [pending-compaction-executor]**
Pending compactions that are known. This metric could deviate from "pending compactions," which includes an estimate of tasks that these pending tasks might create after completion.

**TP: Pending Range Calculator Pending [pending-pending-range-calculator]**
Pending tasks to calculate the ranges according to bootstrapping and leaving nodes.

**TP: Flushes Active [active-flushes]**
Up to memtable_flush_writers concurrent tasks to flush and write the memtables to disk.

**TP: Gossip Tasks Active [active-gossip-stage]**
Number of gossip messages and acknowledgments actively being sent or received.

**TP: Internal Responses Active [active-internal-response-stage]**
Number of active tasks from internal tasks, such as nodes joining and leaving the cluster.

**TP: Manual Repair Tasks Active [active-anti-entropy-stage]**
Repair tasks active, such as handling the merkle tree transfer after the validation compaction.

**TP: Cache Cleaning Active [active-cache-cleanup-stage]**
Tasks to clean row caches during a cleanup compaction.

**TP: Post Flushes Active [active-memtable-post-flush]**
Tasks related to the last step in flushing memtables to disk as SSTables. Includes removing unnecessary commitlog files and committing Solr-based secondary indexes.

**TP: Migrations Active [active-migration-stage]**
Number of active tasks from system methods that modified the schema.

**TP: Misc. Tasks Active [active-misc-stage]**
Number of active tasks from infrequently run operations, such as taking a snapshot or processing the notification of a completed replication.

**TP: Read Repair Tasks Active [active-read-repair-stage]**
Number of read repair operations actively being run.

**TP: Request Responses Active [active-request-response-stage]**
Number of callbacks to being executed after a task on a remote node is completed.

**TP: Validation Executor Active [active-validation-executor]**
Active task to read data from sstables and generate a merkle tree for a repair.

**TP: Compaction Executor Active [active-compaction-executor]**
Active compactions that are known.

**TP: Pending Range Calculator Active [active-pending-range-calculator]**
Active tasks to calculate the ranges according to bootstrapping and leaving nodes.

**TP: Flashes Completed [completed-flushes]**
Number of memtables flushed to disk since the nodes start.

**TP: Gossip Tasks Completed [completed-gossip-stage]**
Number of gossip messages and acknowledgments recently sent or received.

**TP: Internal Responses Completed [completed-internal-response-stage]**
Number of recently completed tasks from internal tasks, such as nodes joining and leaving the cluster.

**TP: Manual Repair Tasks Completed [completed-anti-entropy-stage]**
Repair tasks recently completed, such as handling the merkle tree transfer after the validation compaction.

**TP: Cache Cleaning Completed [completed-cache-cleanup-stage]**
Tasks to clean row caches during a cleanup compaction.

**TP: Post Flashes Completed [completed-memtable-post-flush]**
Tasks related to the last step in flushing memtables to disk as SSTables. Includes removing unnecessary commitlog files and committing Solr-based secondary indexes.

**TP: Migrations Completed [completed-migration-stage]**
Number of completed tasks from system methods that modified the schema.

**TP: Misc. Tasks Completed [completed-misc-stage]**
Number of completed tasks from infrequently run operations, such as taking a snapshot or processing the notification of a completed replication.

**TP: Read Repair Tasks Completed [completed-read-repair-stage]**
Number of read repair operations recently completed.

**TP: Request Responses Completed [completed-request-response-stage]**
Number of completed callbacks executed after a task on a remote node is completed.

**TP: Validation Executor Completed [completed-validation-executor]**
Completed tasks to read data from sstables and generate a merkle tree for a repair.

**TP: Compaction Executor Completed [completed-compaction-executor]**
Completed compactions.
TP: Pending Range Calculator Completed [completed-pending-range-calculator]
   Completed tasks to calculate the ranges according to bootstrapping and leaving
   nodes.

KeyCache Hits [key-cache-hits]
   The number of key cache hits per second. This will avoid possible disk seeks when
   finding a partition in an SSTable. This metric only applies to SSTables created by
   DSE versions earlier than 6.0.

KeyCache Requests [key-cache-requests]
   The number of key cache requests per second. This metric only applies to
   SSTables created by DSE versions earlier than 6.0.

KeyCache Hit Rate [key-cache-hit-rate]
   The percentage of key cache lookups that resulted in a hit. This metric only applies
   to SSTables created by DSE versions earlier than 6.0.

RowCache Hits [row-cache-hits]
   The number of row cache hits per second.

RowCache Requests [row-cache-requests]
   The number of row cache requests per second.

RowCache Hit Rate [row-cache-hit-rate]
   The percentage of row cache lookups that resulted in a hit.

Native Clients [native-connections]
   The number of clients connected using the native protocol.

Read Repairs Attempted [read-repair-attempted]
   Number of read requests where the number of nodes queried possibly exceeds the
   consistency level requested in order to check for a possible digest mismatch.

Asynchronous Read Repairs [read-repaired-background]
   Corresponds to a digest mismatch that occurred after a completed read, outside of
   the client read loop.

Synchronous Read Repairs [read-repaired-blocking]
   Corresponds to the number of times there was a digest mismatch within the
   requested consistency level and a full data read was started.

TBL: Local Writes [cf-write-ops]
   Local write requests per second. Local writes update the table's memtable and
   appends to a commitlog.

TBL: Local Write Latency (percentiles) [cf-local-write-latency]
   The min, median, max, 90th, and 99th percentile of the response times to write data
   to a table's memtable. The elapsed time from when the replica receives the request
   from a coordinator and returns a response.

TBL: Local Reads [cf-read-ops]
   Local read requests per second. Local reads retrieve data from a table's memtable
   and any necessary SSTables on disk.

TBL: Local Read Latency (percentiles) [cf-local-read-latency]
   The min, median, max, 90th, and 99th percentile of the response time to read data
   from the memtable and sstables for a specific table. The elapsed time from when
   the replica receives the request from a coordinator and returns a response.

TBL: Live Disk Used [cf-live-disk-used]
   Disk space used by live SSTables. There might be obsolete SSTables not included.

TBL: Total Disk Used [cf-total-disk-used]
Disk space used by a table by SSTables, including obsolete ones waiting to be garbage collected.

**TBL: SSTable Count [cf-live-sstables]**
Total number of SSTables for a table.

**TBL: SSTables per Read (percentiles) [cf-sstables-per-read]**
The min, median, max, 90th, and 99th percentile of how many SSTables are accessed during a read. Includes sstables that undergo bloom-filter checks, even if no data is read from the sstable.

**TBL: Partition Size (percentiles) [cf-partition-size]**
The min, median, max, 90th, and 99th percentile of the size (in bytes) of partitions of this table.

**TBL: Cell Count (percentiles) [cf-column-count]**
The min, median, max, 90th, and 99th percentile of how many cells exist in partitions for this table.

**TBL: Bloom Filter Space Used [cf-bf-space-used]**
The total size of all the SSTables' bloom filters for this table.

**TBL: Bloom Filter False Positives [cf-bf-false-positives]**
Number of bloom filter false positives per second.

**TBL: Bloom Filter False Positive Ratio [cf-bf-false-ratio]**
Percentage of bloom filter lookups that resulted in a false positive.

**Search: Requests [solr-requests]**
Requests per second made to a specific Solr core/index.

**Search: Request Latency [solr-avg-time-per-req]**
Average time a search query takes in a DSE cluster using DSE Search.

**Search: Errors [solr-errors]**
Errors per second that occur for a specific Solr core/index.

**Search: Timeouts [solr-timeouts]**
Timeouts per second on a specific Solr core/index.

**Search: Core Size [solr-index-size]**
Size of the Solr core on disk.

**OS: Memory (stacked) [os-memory]**
Stacked graph of used, cached, and free memory.

**OS: Memory (stacked) [os-memory-osx]**
Stacked graph of used and free memory.

**OS: Memory Free [os-memory-free]**
Total system memory currently free.

**OS: Memory Used [os-memory-used]**
Total system memory currently used.

**OS: Memory Shared [os-memory-shared]**
Total amount of memory in shared memory space.

**OS: Memory Buffered [os-memory-buffers]**
Total system memory currently buffered.

**OS: Memory Cached [os-memory-cached]**
Total system memory currently cached.

**OS: Memory (stacked) [os-memory-win]**
Stacked graph of committed, cached, paged, non-paged, and free memory.

**OS: Memory Available [os-memory-avail]**
Available physical memory.
OS: Memory Committed [os-memory-committed]
  Memory in use by the operating system.
OS: Pool Paged Resident Memory [os-memory-pool-paged]
  Allocated pool-paged-resident memory.
OS: Pool Nonpaged Memory [os-memory-pool-nonpaged]
  Allocated pool-nonpaged memory.
OS: System Cache Resident Memory [os-memory-sys-cache-resident]
  Memory used by the file cache.
OS: CPU (stacked) [cpu]
  Stacked graph of iowait, steal, nice, system, user, and idle CPU usage.
OS: CPU (stacked) [cpu-osx]
  Stacked graph of idle, user, and system CPU usage.
OS: CPU (stacked) [cpu-win]
  Stacked graph of user, privileged, and idle CPU usage.
OS: CPU User [os-cpu-user]
  Time the CPU devotes to user processes.
OS: CPU System [os-cpu-system]
  Time the CPU devotes to system processes.
OS: CPU Idle [os-cpu-idle]
  Time the CPU is idle.
OS: CPU Iowait [os-cpu-iowait]
  Time the CPU devotes to waiting for I/O to complete.
OS: CPU Steal [os-cpu-steal]
  Time the CPU devotes to tasks stolen by virtual operating systems.
OS: CPU Nice [os-cpu-nice]
  Time the CPU devotes to processing nice tasks.
OS: CPU Privileged [os-cpu-privileged]
  Time the CPU devotes to processing privileged instructions.
OS: Load [os-load]
  Operating system load average. One minute value parsed from /proc/loadavg on Linux systems.
OS: Disk Usage (%) [os-disk-usage]
  Disk space used by Cassandra at a given time.
OS: Disk Free [os-disk-free]
  Free space on a specific disk partition.
OS: Disk Used [os-disk-used]
  Disk space used by Cassandra at a given time.
OS: Disk Read Throughput [os-disk-read-throughput]
  Average disk throughput for read operations.
OS: Disk Write Throughput [os-disk-write-throughput]
  Average disk throughput for write operations.
OS: Disk Throughput [os-disk-throughput]
  Average disk throughput for read and write operations.
OS: Disk Read Rate [os-disk-read-rate]
  Rate of reads per second to the disk.
OS: Disk Writes Rate [os-disk-write-rate]
  Rate of writes per second to the disk.
OS: Disk Latency [os-disk-await]
Average completion time of each request to the disk.

**OS: Disk Request Size [os-disk-request-size]**
- Average size of read requests issued to the disk.

**OS: Disk Request Size [os-disk-request-size-kb]**
- Average size of read requests issued to the disk.

**OS: Disk Queue Size [os-disk-queue-size]**
- Average number of requests queued due to disk latency issues.

**OS: Disk Utilization [os-disk-utilization]**
- CPU time consumed by disk I/O.

**OS: Net Received [os-net-received]**
- Speed of data received from the network.

**OS: Net Sent [os-net-sent]**
- Speed of data sent across the network.

**OS: Net Sent [os-net-sent-win]**
- Speed of data sent across the network.

**OS: Net Received [os-net-received-win]**
- Speed of data received from the network.

**Speculative Retries [speculative-retries]**
- Number of speculative retries for all column families.

**TBL: Speculative Retries [cf-speculative-retries]**
- Number of speculative retries for this table.

**Stream Data Out - Total [stream-out-total]**
- Data streamed out from this node to all other nodes, for all tables.

**Stream Data In - Total [stream-in-total]**
- Data streams in to this node from all other nodes, for all tables.

**Hint Creation Rate [hint-creation-rate]**
- Rate at which new individual hints are stored on this node, to be replayed to peers.

**TBL: Bloom Filter Off Heap [cf-bf-offheap]**
- Total off heap memory used by bloom filters from all live SSTables in a table.

**TBL: Index Summary Off Heap [cf-index-summary-offheap]**
- Total off heap memory used by the index summary of all live SSTables in a table.

**TBL: Compression Metadata Off Heap [cf-compression-data-offheap]**
- Total off heap memory used by the compression metadata of all live SSTables in a table.

**TP: Memtable Reclaims Pending [memtable-reclaim-pending]**
- Waits for current reads to complete and then frees the memory formerly used by the obsoleted memtables.

**TP: Memtable Reclaims Active [memtable-reclaim-active]**
- Waits for current reads to complete and then frees the memory formerly used by the obsoleted memtables.

**TP: Memtable Reclaims Completed [completed-memtable-reclaim]**
- Waits for current reads to complete and then frees the memory formerly used by the obsoleted memtables.

**TBL: Memtable Off Heap [cf-memtable-offheap]**
- Off heap memory used by a table's current memtable.

**TBL: Total Memtable Heap Size [cf-all-memtables-heapsize]**
- An estimate of the space used in JVM heap memory for all memtables. This includes ones that are currently being flushed and related secondary indexes.
TBL: Total Memtable Live Data Size [cf-all-memtables-livedatasize]
An estimate of the space used for 'live data' (off-heap, excluding overhead) for all memtables. This includes ones that are currently being flushed and related secondary indexes.

TBL: Total Memtable Off-Heap Size [cf-all-memtables-offheapsize]
An estimate of the space used in off-heap memory for all memtables. This includes ones that are currently being flushed and related secondary indexes.

In-Memory Percent Used [in-memory-percent-used]
The percentage of memory allocated for in-memory tables currently in use.

TBL: Partition Count [cf-row-size]
Approximate number of partitions. This may be off given duplicates in memtables and sstables are both counted and there is a very small error percentage inherited from the HyperLogLog data structure.

Write Request Latency [write-latency-legacy]
Deprecated. The median response times (in milliseconds) of a client write. The time period starts when a node receives a client write request, and ends when the node responds back to the client. Depending on consistency level and replication factor, this may include the network latency from writing to the replicas.

Read Request Latency [read-latency-legacy]
Deprecated. The median response times (in milliseconds) of a client read. The time period starts when a node receives a client read request, and ends when the node responds back to the client. Depending on consistency level and replication factor, this may include the network latency from requesting the data's replicas.

View Write Latency (percentiles) [view-write-histogram]
The min, median, max, 90th, and 99th percentiles of the time from when base mutation is applied to memtable until CL.ONE is achieved on the async write to the tables materialized views. An estimate to determine the lag between base table mutations and the views consistency.

View Write Successes [view-replicas-success]
The number of view mutations sent to replicas that have been acknowledged.

View Write Pending [view-replicas-pending]
The number of view mutations sent to replicas where the replicas acknowledgement hasn't been received.

TP: Hint Dispatcher Pending [pending-hint-dispatcher]
Pending tasks to send the stored hinted handoffs to a host.

TP: Hint Dispatcher Active [active-hint-dispatcher]
Up to max_hints_delivery_threads tasks, each dispatching all hinted handoffs to a host.

TP: Hint Dispatcher Completed [completed-hint-dispatcher]
The number of tasks to transfer hints to a host that have completed.

TP: Index Management Pending [pending-secondary-index-management]
Any initialization work when a new index instance is created. This may involve costly operations such as (re)building the index.

TP: Index Management Active [active-secondary-index-management]
Any initialization work when a new index instance is created. This may involve costly operations such as (re)building the index.

TP: Index Management Completed [completed-secondary-index-management]
Any initialization work when a new index instance is created. This may involve costly operations such as (re)building the index.

**TBL: Tombstones per Read (percentiles) [cf-tombstones-per-read]**

The min, median, max, 90th, and 99th percentile of how many tombstones are read during a read.

**TBL: Local Write Latency [cf-write-latency-legacy]**

*Deprecated.* Median response time to write data to a table's memtable. The elapsed time from when the replica receives the request from a coordinator and returns a response.

**TBL: Local Read Latency [cf-read-latency-legacy]**

*Deprecated.* Median response time to read data from the memtable and SSTables for a specific table. The elapsed time from when the replica receives the request from a coordinator and returns a response.

**TBL: Coordinator Read Latency (percentiles) [cf-coordinator-read-latency]**

The min, median, max, 90th, and 99th percentiles of client reads on this table. The time period starts when a node receives a client read request, and ends when the node responds back to the client. Depending on consistency level and replication factor, this may include the network latency from requesting the data's replicas.

**TBL: Coordinator Read Requests [cf-coordinator-read-ops]**

The number of read requests per second for a particular table on the coordinator nodes. Monitoring the number of requests over a given time period reveals table read workload and usage patterns.

**Cells Scanned (percentiles) [cells-scanned-during-read]**

The min, median, max, 90th, and 99th percentile of how many cells were scanned during a read.

**TBL: Cells Scanned (percentiles) [cf-cells-scanned-during-read]**

The min, median, max, 90th, and 99th percentile of how many cells were scanned during a read.

**TIER: Total Disk Used [cf-tier-size]**

Disk space used by a table by SSTables for the tier.

**TIER: sstables [cf-tier-sstables]**

Number of SSTables in a tier for a table.

**TIER: Max Data Age [cf-tier-max-data-age]**

Timestamp in local server time that represents an upper bound to the newest piece of data stored in the SSTable. When a new SSTable is flushed, it is set to the time of creation. When an SSTable is created from compaction, it is set to the max of all merged SSTables.

**Graph: Adjacency Cache Hits [graph-adjacency-cache-hit]**

Number of hits against the adjacency cache for this graph.

**Graph: Adjacency Cache Misses [graph-adjacency-cache-miss]**

Number of misses against the adjacency cache for this graph.

**Graph: Index Cache Hits [graph-index-cache-hit]**

Number of hits against the index cache for this graph.

**Graph: Index Cache Misses [graph-index-cache-miss]**

Number of misses against the index cache for this graph.

**Graph: Request Latencies [graph-request-latencies]**

The min, median, max, 90th, and 99th percentile of request latencies during the period.
Graph TP: Graph Scheduled Threads Pending [pending-graph-scheduled-threads]
Number of pending tasks in the GraphScheduledThreads thread pool.

Graph TP: Graph Scheduled Threads Active [active-graph-scheduled-threads]
Number of active tasks in the GraphScheduledThreads thread pool.

Graph TP: Graph Scheduled Threads Completed [completed-graph-scheduled-threads]
Number of tasks completed by the GraphScheduledThreads thread pool.

Graph TP: Graph System Threads Pending [pending-graph-system-threads]
Number of pending tasks in the GraphSystemThreads thread pool.

Graph TP: Graph System Threads Active [active-graph-system-threads]
Number of active tasks in the GraphSystemThreads thread pool.

Graph TP: Graph System Threads Completed [completed-graph-system-threads]
Number of tasks completed by the GraphSystemThreads thread pool.

Graph TP: Gremlin Worker Threads Pending [pending-gremlin-worker-threads]
Number of pending tasks in the GremlinWorkerThreads thread pool.

Graph TP: Gremlin Worker Threads Active [active-gremlin-worker-threads]
Number of active tasks in the GremlinWorkerThreads thread pool.

Graph TP: Gremlin Worker Threads Completed [completed-gremlin-worker-threads]
Number of tasks completed by the GremlinWorkerThreads thread pool.

Percent Data Repaired [percentage-repaired]
Percentage of data (uncompressed) marked as repaired across all non-system tables on a node. Tables with a replication factor of 1 are excluded.

TBL: Percent Data Repaired [cf-percentage-repaired]
Percentage of data (uncompressed) marked as repaired for a given table on a node. This metric is only meaningful for replication factor > 1.

Read Requests - Local Node Non Replica [read-coordinator-nonreplica]
Rate of coordinated reads to a node where that node is not a replica for that partition.

Read Requests - Preferred Other Replicas [read-coordinator-preferother]
Rate of coordinated reads to a node where that node did not choose itself as a replica for the read request.

Hints on Disk [hints-on-disk]
The number of hints currently stored on disk, to be replayed to peers.

Hint Replay Success Rate [hint-replay-success-rate]
Rate of successful individual hint replays to peers. If one or more individual hints fail to replay in a batch, the successful hints in that batch will be replayed again and double counted in this metric.

Hint Replay Error Rate [hint-replay-error-rate]
Rate of failed individual hint replays. Replay of a single hint can fail more than once if retried.

Hint Replay Timeout Rate [hint-replay-timeout-rate]
Rate of timed out individual hint replays. Replay of a single hint can timeout more than once if retried.

Hint Replay Received Rate [hint-replay-received-rate]
Rate of successful individual hints replayed to this node, from other peers.

Node Messaging Latency [cross-node-latency]
The min, median, max, 90th, and 99th percentiles of the latency of messages between nodes. The time period starts when a node sends a message and ends when the current node receives it.

**Datacenter Messaging Latency [cross-dc-latency]**
The min, median, max, 90th, and 99th percentiles of the message latency between nodes in the same or different destination datacenter. This metric measures how long it takes a message from a node in the source datacenter to reach a node in the destination datacenter. Selecting a destination node within the source datacenter yields lower latency values.

**NodeSync: Data Repaired [nodesync-data-repaired]**
Bytes of data that were inconsistent and needed synchronization.

**NodeSync: Data Validated [nodesync-data-validated]**
Bytes of data checked for consistency.

**NodeSync: Repair Data Sent [nodesync-repair-data-sent]**
Total bytes of data transferred between all nodes during synchronization.

**NodeSync: Objects Repaired [nodesync-objects-repaired]**
Number of rows and range tombstones that were inconsistent and needed synchronization.

**NodeSync: Objects Validated [nodesync-objects-validated]**
Number of rows and range tombstones checked for consistency.

**NodeSync: Repair Objects Sent [nodesync-repair-objects-sent]**
Total number of rows and range tombstones transferred between all nodes during synchronization.

**NodeSync: Processed Pages [nodesync-processed-pages]**
Number of pages (internal groupings of data) processed.

**NodeSync: Full In Sync Pages [nodesync-full-in-sync-pages]**
Number of processed pages that were not in need of synchronization.

**NodeSync: Full Repaired Pages [nodesync-full-repaired-pages]**
Number of processed pages that were in need of synchronization.

**NodeSync: Partial In Sync Pages [nodesync-partial-in-sync-pages]**
Number of in sync pages for which a response was gotten from only a partial number of replicas.

**NodeSync: Partial Repaired Pages [nodesync-partial-repaired-pages]**
Number of repaired pages for which a response was gotten from only a partial number of replicas.

**NodeSync: Uncompleted Pages [nodesync-uncompleted-pages]**
Number of processed pages not having enough responses to perform synchronization.

**NodeSync: Failed Pages [nodesync-failed-pages]**
Number of processed pages for which an unknown error prevented proper synchronization completion.

**NodeSync TBL: Data Repaired [nodesync-tbl-data-repaired]**
Bytes of data that were inconsistent and needed synchronization.

**NodeSync TBL: Data Validated [nodesync-tbl-data-validated]**
Bytes of data checked for consistency.

**NodeSync TBL: Repair Data Sent [nodesync-tbl-repair-data-sent]**
Total bytes of data transferred between all nodes during synchronization.

**NodeSync TBL: Objects Repaired [nodesync-tbl-objects-repaired]**
Number of rows and range tombstones that were inconsistent and needed synchronization.

**NodeSync TBL: Objects Validated [nodesync-tbl-objects-validated]**
Number of rows and range tombstones checked for consistency.

**NodeSync TBL: Repair Objects Sent [nodesync-tbl-repair-objects-sent]**
Total number of rows and range tombstones transferred between all nodes during synchronization.

**NodeSync TBL: Processed Pages [nodesync-tbl-processed-pages]**
Number of pages (internal groupings of data) processed.

**NodeSync TBL: Full In Sync Pages [nodesync-tbl-full-in-sync-pages]**
Number of processed pages that were not in need of synchronization.

**NodeSync TBL: Full Repaired Pages [nodesync-tbl-full-repaired-pages]**
Number of processed pages that were in need of synchronization.

**NodeSync TBL: Partial In Sync Pages [nodesync-tbl-partial-in-sync-pages]**
Number of in sync pages for which a response was gotten from only a partial number of replicas.

**NodeSync TBL: Partial Repaired Pages [nodesync-tbl-partial-repaired-pages]**
Number of repaired pages for which a response was gotten from only a partial number of replicas.

**NodeSync TBL: Uncompleted Pages [nodesync-tbl-uncompleted-pages]**
Number of processed pages not having enough responses to perform synchronization.

**NodeSync TBL: Failed Pages [nodesync-tbl-failed-pages]**
Number of processed pages for which an unknown error prevented proper synchronization completion.

**TP: Authentication Active [active-authentication]**
Authentication Active

**TP: Authentication Completed [completed-authentication]**
Authentication Completed

**TP: Execute Statement Active [active-execute-statement]**
Execute Statement Active

**TP: Execute Statement Completed [completed-execute-statement]**
Execute Statement Completed

**TP: Unknown Active [active-unknown]**
Unknown Active

**TP: Unknown Completed [completed-unknown]**
Unknown Completed

**TP: Truncate Active [active-truncate]**
Truncate Active

**TP: Truncate Completed [completed-truncate]**
Truncate Completed

**TP: Counter Acquire Lock Active [active-counter-acquire-lock]**
Counter Acquire Lock Active

**TP: Counter Acquire Lock Completed [completed-counter-acquire-lock]**
Counter Acquire Lock Completed

**TP: CAS Active [active-cas]**
CAS Active

**TP: CAS Completed [completed-cas]**
CAS Completed
TP: Write Switch For Memtable Active [active-write-switch-for-memtable]
  Write Switch For Memtable Active
TP: Write Switch For Memtable Completed [completed-write-switch-for-memtable]
  Write Switch For Memtable Completed
TP: Read Disk Async Active [active-read-disk-async]
  Read Disk Async Active
TP: Read Disk Async Completed [completed-read-disk-async]
  Read Disk Async Completed
TP: Timed Unknown Active [active-timed-unknown]
  Timed Unknown Active
TP: Timed Unknown Completed [completed-timed-unknown]
  Timed Unknown Completed
TP: Timed Timeout Active [active-timed-timeout]
  Timed Timeout Active
TP: Timed Timeout Completed [completed-timed-timeout]
  Timed Timeout Completed
TP: Write Defragment Active [active-write-defragment]
  Write Defragment Active
TP: Write Defragment Completed [completed-write-defragment]
  Write Defragment Completed
TP: Read Secondary Index Active [active-read-secondary-index]
  Read Secondary Index Active
TP: Read Secondary Index Completed [completed-read-secondary-index]
  Read Secondary Index Completed
TP: Write Defragment Total Blocked [total-blocked-write-defragment]
  Total Write Defragment Blocked
TP: Write Defragment Pending [pending-write-defragment]
  Write Defragment Pending
TP: View Build Executor Active [active-view-build-executor]
  View Build Executor Active
TP: View Build Executor Completed [completed-view-build-executor]
  View Build Executor Completed
TP: View Build Executor Pending [pending-view-build-executor]
  View Build Executor Pending
TP: Background IO Stage Active [active-background-io-stage]
  Background IO Stage Active
TP: Background IO Stage Completed [completed-background-io-stage]
  Background IO Stage Completed
TP: Background IO Stage Pending [pending-background-io-stage]
  Background IO Stage Pending
TP: Repair Tasks Active [active-repair-task]
  Repair Tasks Active
TP: Repair Tasks Completed [completed-repair-task]
  Repair Tasks Completed
TP: Repair Tasks Pending [pending-repair-task]
  Repair Tasks Pending
TP: Auth Stage Active [active-auth-stage]
Auth Stage Active
TP: Auth Stage Completed [completed-auth-stage]
  Auth Stage Completed
TP: Auth Stage Pending [pending-auth-stage]
  Auth Stage Pending
TP: Frame Decode Total Blocked [total-blocked-frame-decode]
  Total Frame Decode Blocked
TP: Write Remote Total Blocked [total-blocked-write-remote]
  Total Write Remote Blocked
TP: Frame Decode Completed [completed-frame-decode]
  Frame Decode Completed
TP: Read Local Pending [pending-read-local]
  Read Local Pending
TP: Write Remote Completed [completed-write-remote]
  Write Remote Completed
TP: Read Remote Pending [pending-read-remote]
  Read Remote Pending
TP: Read Range Remote Pending [pending-read-range-remote]
  Read Range Remote Pending
TP: Read Range Local Completed [completed-read-range-local]
  Read Range Local Completed
TP: Write Local Pending [pending-write-local]
  Write Local Pending
TP: Read Range Nodesync Active [active-read-range-nodesync]
  Read Range Nodesync Active
TP: Network Backpressure Active [active-network-backpressure]
  Network Backpressure Active
TP: Read Range Nodesync Completed [completed-read-range-nodesync]
  Read Range Nodesync Completed
TP: Read Local Active [active-read-local]
  Read Local Active
TP: Read Timeout Completed [completed-read-timeout]
  Read Timeout Completed
TP: Read Internal Completed [completed-read-internal]
  Read Internal Completed
TP: Read Local Completed [completed-read-local]
  Read Local Completed
TP: Read Range Local Total Blocked [total-blocked-read-range-local]
  Total Read Range Local Blocked
TP: Read Range Internal Completed [completed-read-range-internal]
  Read Range Internal Completed
TP: Hint Dispatch Active [active-hint-dispatch]
  Hint Dispatch Active
TP: Write Internal Active [active-write-internal]
  Write Internal Active
TP: Hint Response Completed [completed-hint-response]
  Hint Response Completed
TP: Write Internal Completed [completed-write-internal]
Write Internal Completed
TP: Read Speculate Active [active-read-speculate]
    Read Speculate Active
TP: Network Backpressure Completed [completed-network-backpressure]
    Network Backpressure Completed
TP: Read Range Remote Completed [completed-read-range-remote]
    Read Range Remote Completed
TP: Read Timeout Active [active-read-timeout]
    Read Timeout Active
TP: Read Internal Active [active-read-internal]
    Read Internal Active
TP: Read Speculate Completed [completed-read-speculate]
    Read Speculate Completed
TP: Hint Dispatch Completed [completed-hint-dispatch]
    Hint Dispatch Completed
TP: Write Local Completed [completed-write-local]
    Write Local Completed
TP: Read Range Remote Total Blocked [total-blocked-read-range-remote]
    Total Read Range Remote Blocked
TP: Hint Dispatch Total Blocked [total-blocked-hint-dispatch]
    Total Hint Dispatch Blocked
TP: Read Local Total Blocked [total-blocked-read-local]
    Total Read Local Blocked
TP: Read Range Internal Active [active-read-range-internal]
    Read Range Internal Active
TP: Read Remote Active [active-read-remote]
    Read Remote Active
TP: Write Remote Active [active-write-remote]
    Write Remote Active
TP: Frame Decode Active [active-frame-decode]
    Frame Decode Active
TP: Read Range Local Active [active-read-range-local]
    Read Range Local Active
TP: Network Backpressure Pending [pending-network-backpressure]
    Network Backpressure Pending
TP: Write Local Active [active-write-local]
    Write Local Active
TP: Read Range Local Pending [pending-read-range-local]
    Read Range Local Pending
TP: Hint Response Active [active-hint-response]
    Hint Response Active
TP: Read Range Nodesync Total Blocked [total-blocked-read-range-nodesync]
    Total Read Range Nodesync Blocked
TP: Read Range Remote Active [active-read-range-remote]
    Read Range Remote Active
TP: Network Backpressure Total Blocked [total-blocked-network-backpressure]
    Total Network Backpressure Blocked
TP: Write Local Total Blocked [total-blocked-write-local]
Total Write Local Blocked

**TP: Frame Decode Pending [pending-frame-decode]**
Frame Decode Pending

**TP: Write Remote Pending [pending-write-remote]**
Write Remote Pending

**TP: Read Range Nodessync Pending [pending-read-range-nodessync]**
Read Range Nodessync Pending

**TP: Hint Dispatch Pending [pending-hint-dispatch]**
Hint Dispatch Pending

**TP: Read Remote Total Blocked [total-blocked-read-remote]**
Total Read Remote Blocked

**TP: Read Remote Completed [completed-read-remote]**
Read Remote Completed

**TP: Eventloop Spin Active [active-eventloop-spin]**
Eventloop Spin Active

**TP: Read Deferred Completed [completed-read-deferred]**
Read Deferred Completed

**TP: Authorization Completed [completed-authorization]**
Authorization Completed

**TP: Batch Replay Completed [completed-batch-replay]**
Batch Replay Completed

**TP: Write Await Commitlog Segment Active [active-write-await-commitlog-segment]**
Write Await Commitlog Segment Active

**TP: Eventloop Park Active [active-eventloop-park]**
Eventloop Park Active

**TP: Read Switch For Response Active [active-read-switch-for-response]**
Read Switch For Response Active

**TP: NodeSync Validation Active [active-nodesync-validation]**
NodeSync Validation Active

**TP: Read Switch For Iterator Active [active-read-switch-for-iterator]**
Read Switch For Iterator Active

**TP: Batch Remove Active [active-batch-remove]**
Batch Remove Active

**TP: Batch Replay Active [active-batch-replay]**
Batch Replay Active

**TP: Read Range Switch For Response Active [active-read-range-switch-for-response]**
Read Range Switch For Response Active

**TP: Write Switch For Response Active [active-write-switch-for-response]**
Write Switch For Response Active

**TP: Batch Remove Completed [completed-batch-remove]**
Batch Remove Completed

**TP: Batch Store Response Completed [completed-batch-store-response]**
Batch Store Response Completed

**TP: Write Memtable Full Active [active-write-memtable-full]**
Write Memtable Full Active

**TP: Lwt Propose Pending [pending-lwt-propose]**
Lwt Propose Pending

**TP: Write Await Commitlog Sync Active [active-write-await-commitlog-sync]**
Write Await Commitlog Sync Active

TP: NodeSync Validation Completed [completed-nodesync-validation]
  NodeSync Validation Completed

TP: Lwt Commit Completed [completed-lwt-commit]
  Lwt Commit Completed

TP: Read Switch For Response Completed [completed-read-switch-for-response]
  Read Switch For Response Completed

TP: Eventloop Yield Active [active-eventloop-yield]
  Eventloop Yield Active

TP: Lwt Prepare Active [active-lwt-prepare]
  Lwt Prepare Active

TP: Lwt Propose Completed [completed-lwt-propose]
  Lwt Propose Completed

TP: Batch Store Pending [pending-batch-store]
  Batch Store Pending

TP: Read Switch For Iterator Completed [completed-read-switch-for-iterator]
  Read Switch For Iterator Completed

TP: Lwt Prepare Pending [pending-lwt-prepare]
  Lwt Prepare Pending

TP: Write Memtable Full Completed [completed-write-memtable-full]
  Write Memtable Full Completed

TP: Truncate Pending [pending-truncate]
  Truncate Pending

TP: Read Deferred Pending [pending-read-deferred]
  Read Deferred Pending

TP: Eventloop Spin Completed [completed-eventloop-spin]
  Eventloop Spin Completed

TP: Write Switch For Response Completed [completed-write-switch-for-response]
  Write Switch For Response Completed

TP: Eventloop Park Completed [completed-eventloop-park]
  Eventloop Park Completed

TP: Lwt Propose Active [active-lwt-propose]
  Lwt Propose Active

TP: Lwt Prepare Completed [completed-lwt-prepare]
  Lwt Prepare Completed

TP: Authorization Active [active-authorization]
  Authorization Active

TP: Eventloop Yield Completed [completed-eventloop-yield]
  Eventloop Yield Completed

TP: Batch Store Completed [completed-batch-store]
  Batch Store Completed

TP: Batch Store Active [active-batch-store]
  Batch Store Active

TP: Batch Remove Pending [pending-batch-remove]
  Batch Remove Pending

TP: Lwt Commit Active [active-lwt-commit]
  Lwt Commit Active

TP: Lwt Commit Pending [pending-lwt-commit]
Using OpsCenter

Lwt Commit Pending
TP: Write Await Commitlog Segment Completed [completed-write-await-commitlog-segment]
  Write Await Commitlog Segment Completed
TP: Read Range Switch For Response Completed [completed-read-range-switch-for-response]
  Read Range Switch For Response Completed
TP: Batch Store Response Active [active-batch-store-response]
  Batch Store Response Active
TP: Write Await Commitlog Sync Completed [completed-write-await-commitlog-sync]
  Write Await Commitlog Sync Completed
TP: Read Deferred Active [active-read-deferred]
  Read Deferred Active
TP: Batch Remove Total Blocked [total-blocked-batch-remove]
  Total Batch Remove Blocked
TP: Read Deferred Total Blocked [total-blocked-read-deferred]
  Total Read Deferred Blocked
TP: Lwt Commit Total Blocked [total-blocked-lwt-commit]
  Total Lwt Commit Blocked
TP: Lwt Propose Total Blocked [total-blocked-lwt-propose]
  Total Lwt Propose Blocked
TP: Truncate Total Blocked [total-blocked-truncate]
  Total Truncate Blocked
TP: Lwt Prepare Total Blocked [total-blocked-lwt-prepare]
  Total Lwt Prepare Blocked
TP: Batch Store Total Blocked [total-blocked-batch-store]
  Total Batch Store Blocked

Dropped Messages: Materialized View Mutations [dropped-view-mutations]
  Mutation of Materialized View was seen after the timeout
  (write_request_timeout_in_ms) so was thrown away. This client might have timed
out before it met the required consistency level, but might have succeeded as well.
  Hinted handoffs and read repairs should resolve inconsistencies but a repair can
ensure it.

Dropped Messages: Lightweight Transactions [dropped-lwt]
  Lightweight Transaction was seen after the timeout (write_request_timeout_in_ms)
so was thrown away. This client might have timed out before it met the required
consistency level, but might have succeeded as well. Hinted handoffs and read
repairs should resolve inconsistencies but a repair can ensure it.

Dropped Messages: Hinted Handoffs [dropped-hints]
  Hinted Handoff was seen after the timeout (write_request_timeout_in_ms) so
was thrown away. Repairing the data or using NodeSync, should resolve data
inconsistencies.

Dropped Messages: Truncate Operations [dropped-truncates]
  Truncate operation was seen after the timeout (truncate_request_timeout_in_ms)
so was thrown away.

Dropped Messages: Snapshot Requests [dropped-snapshots]
  Snapshot Request was seen after the timeout (request_timeout_in_ms) so was
thrown away. Snapshot should be retried.
Dropped Messages: Schema Changes [dropped-schemas]
Schema change was seen after the timeout (request_timeout_in_ms) so was thrown away. Schema agreement may not have been reached immediately, but this will eventually resolve itself.

Dropped Messages: Repair Messages [dropped-repairs]
Repair message was seen after the timeout so was thrown away.

Dropped Messages: Miscellaneous [dropped-other]
Miscellaneous message was seen after the timeout so was thrown away.

Dropped Messages: NodeSync [dropped-node-sync]
NodeSync message was seen after the timeout so was thrown away.

Dropped Messages: Batch Store [dropped-batch-store]
Batch store message was seen after the timeout so was thrown away.
DSE Management Services

Backup Service

The OpsCenter Backup Service allows scheduling an automatic backup or running a manual backup of DSE cluster data. Backup from all the keyspaces in a cluster to specific keyspaces. Backup data by specific datacenters for improved efficiency. You can perform both local and remote backups, point-in-time restores, and restoring to a different cluster (that is, cloning data). Backup data is stored locally on each node (On Server), and optionally in a local filesystem or cloud-based storage services such as Amazon S3.

Backup Service overview

Use OpsCenter to schedule and manage backups, and restore from those backups, across all registered DataStax Enterprise clusters. The Backup Service:

- Performs all functions using the REST API (page 627) or visually through the OpsCenter UI
- Delivers smart backups that always ensure full data protection, including backups of commit logs
- Backs up data to a local server (On Server), Amazon S3, or a custom location on the local filesystem
- Compresses backup files to save storage
- Allows specifying retention policies on scheduled backups
- Easily lets admins carry out full, table-level, or point-in-time restores for a cluster
- Notifies operations staff should backup or restore operations fail
- Supports cloning data between clusters (such as copy data from a production cluster to a development cluster) or from a defined other location (Amazon S3 or Local FS)
- Provides detailed backup and restore reports and history

A backup is a snapshot of all on-disk data files (SSTable files) stored in the data directory. Backups are stored locally on each node (On Server), and you can specify additional locations such as a local filesystem or in cloud backup services like Amazon S3 where the snapshot data is copied.

Backups can be taken per datacenter, per keyspace, for selected multiple keyspaces, or for all keyspaces in the cluster while the system is online.

**Note:** Consider the following caveats when creating and restoring backups:

- Restoring a snapshot that contains only the system keyspace is not allowed. There must be both system and non-system keyspaces, or only non-system keyspaces in the snapshot you want to restore.
- Restoring a snapshot that does not contain a table definition is not allowed.
• Restoring from a backup while Kerberos is enabled is not currently supported by OpsCenter.
• Restoring a snapshot to a location with insufficient disk space fails. The Restore Report indicates which nodes do not have sufficient space and how much space is necessary for a successful restore. For more information and tips for preventative measures, see Monitoring sufficient disk space for restoring backups (page 427).

There must be enough free disk space on the node to accommodate making snapshots of data files. Configure the free disk space threshold (page 387) to prevent backups from starting if there is insufficient disk space below a specified percentage. A single snapshot requires little disk space. However, snapshots cause disk usage to grow more quickly over time because a snapshot prevents obsolete data files from being deleted. Specify how long to retain the snapshot data by setting a retention policy for each backup location.

Note: OpsCenter data backups do not show or manage manual snapshots taken using the nodetool snapshot command.

If a cluster includes DSE Search or DSE Analytics nodes, a backup job that includes keyspaces with DSE Search data or Analytics nodes will save the Search and Analytics data. Any Solr indexes are recreated upon restore.

OpsCenter intelligently stores the backup data to prevent duplication of files. A backup first flushes all in-memory writes to disk, then makes a hard link of the SSTable files for each keyspace. Unlike traditional backup systems that use full backups and then incremental backups with deltas based on the last full backup, the OpsCenter approach allows you to fully recreate the state of the database at the time of each backup without duplicating files. If you have configured an additional Local FS or S3 location, OpsCenter creates a manifest for each backup that contains a list of the SSTables in that backup, and only uploads new SSTable files.

You can schedule backups to run automatically on a recurring interval, or manually run one-off backups on a scheduled or ad hoc basis.

Commit log backups for point-in-time restores

In addition to keyspace backups, commit log backups are also available in the backup service to facilitate point-in-time (page 421) restores for finer-grained control of the backup data. Point-in-time restores are available after enabling (page 384) commit log backups in conjunction with keyspace backups. Similar to keyspace backups, the commit log archives are retained based on a configurable retention policy.

Note: Point-in-time restores are only supported if the cluster topology is unchanged since the point-in-time to which you want to restore a backup.

Backup retention policies

Each scheduled backup has a retention policy that defines how OpsCenter handles the files for older backup data. The default policy is to retain On Server backup files for 30 days.
Amazon S3 and Local FS default retention policy is to Retain all. For each scheduled backup task and configured location, you can set a configurable time period for which to retain the snapshot data. OpsCenter supports minutes, hours, days, and weeks for the retention time period. For example, you can define a retention policy that removes snapshot data older than 30 days, or 26 weeks, or 3 hours. If you want to keep all backups, OpsCenter has a Retain All policy that retains the backup files indefinitely.

When a backup that was configured with a time-limited retention policy completes, OpsCenter scans the snapshot data for outdated files that do not belong to other snapshots and removes them at the next scheduled backup.

For example, a scheduled backup sends data to S3, runs weekly, and has a retention policy of removing backups older than 3 days. The layout in the S3 bucket is as follows:

```
mybucket/
snapshots/
 node-id1/
 sstables/
   MyKeyspace-MyTable-ic-4-Data.db
   MyKeyspace-MyTable-ic-5-Data.db
   MyKeyspace-MyTable-ic-6-Data.db
   MyKeyspace-MyTable-ic-7-Data.db
   ...
1234-ABCD-2015-01-25-01-00/
   backup.json #includes 4-Data and 5-Data
   MyKeyspace/schema.json
1234-ABCD-2015-02-01-01-00/
   backup.json #includes 5,6,7-Data
   MyKeyspace/schema.json
```

After the February 1 backup completes, OpsCenter scans the SSTables for outdated files according to the retention policy. The January 25 backup files can be removed by OpsCenter. Because `MyKeyspace-MyTable-ic-4-Data.db` was in the January 25 backup but not in the February 1 backup, it will be removed. Even though `MyKeyspace-MyTable-ic-5-Data.db` was in the January 25 backup, it is also in the latest backup, so it will be retained until it meets its defined retention policy.

**Backing up to Amazon S3**

When you add an Amazon S3 bucket as an additional location for storing backup snapshots, the agent sends the snapshot files to the S3 bucket automatically. All SSTables for a particular node and table are only stored once in S3 to optimize storage space.

**Important:** The Backup Service requires control over the data and structure of its destination locations. The AWS S3 bucket and the Local file system destinations must be dedicated for use only by OpsCenter. Any additional directories or files in those destinations can prevent the Backup Service from properly conducting a Backup or Restore operation.
The backup files are stored in S3 in the following hierarchy:

```
mybucket/
  snapshots/
     node-id1/
        sstables/
           MyKeyspace-MyTable-ic-5-Data.db
               ...
           MyKeyspace-MyTable-ic-5-TOC.txt
           MyKeyspace-MyTable-ic-6-Data.db
               ...
        1234-ABCD-2014-10-01-01-00/
            backup.json
            MyKeyspace/schema.json
        1234-ABCD-2014-09-30-01-00/
            backup.json
            MyKeyspace/schema.json
     node-id2/
        sstables/
           MyKeyspace-MyTable-ic-1-Data.db
               ...
           MyKeyspace-MyTable-ic-2-Data.db
               ...
        1234-ABCD-2014-10-01-01-00/
            backup.json
            MyKeyspace/schema.json
        1234-ABCD-2014-09-30-01-00/
            backup.json
            MyKeyspace/schema.json
commitlogs/
     node1/
        1435432324_Commitlog-3-1432320421.log
        1435433232_Commitlog-3-1432320422.log
        ...
```

The `backup.json` file contains metadata about which of the backed up SSTables are included in that backup.

**Note:** The Backup Service switched to the AWS SDK as of OpsCenter version 6.1. With the current heap default *(page 174)*, a maximum SSTable file size of 1 TB is supported when backing up to S3.

If OpsCenter encounters an error when backing up to S3, it retries the backup a configurable number of times (3 by default) unless it encounters an unrecoverable error such as invalid AWS credentials.
Warning: The AWS credentials and bucket names are stored in cluster_name.conf (with the exception of ad hoc backups). Be sure to use proper security precautions to ensure that this file is not readable by unauthorized users.

Why use the Backup Service?

The Backup Service was designed to manage enterprise-wide backup and restore operations for DataStax Enterprise clusters. While some administrators and operations staff believe that backup is not needed because of powerful and flexible replication capabilities in DSE, proper backup and restore procedures are still very important to implement for production clusters.

While replication does provide for copies of data to exist in multiple locations, datacenters, and cloud availability zones, all operations performed in a cluster are replicated, including operations that result in lost or incorrect data. For example, if a table is mistakenly dropped, if data is accidentally deleted, or if cluster data becomes corrupted, those adverse events will be replicated to all other copies of that data. In such cases, there is no way to recover the lost or uncorrupted data without a backup of the data. The Backup Service provides a simple interface for scheduling regular or one-off backups of all or specific keyspaces in a cluster, and for recovering data from the stored backups.

DataStax strongly recommends that organizations using DataStax Enterprise create a good backup and recovery plan using the Backup Service. Testing backup and restore operations on a non-production cluster is also recommended to ensure that the disaster recovery plan deployed for your organization actually works as intended.

Quick Video Tour: Backup Service

View a DataStax Academy video for a brief introduction to backup, restore, and cloning data. Be sure you have a DataStax Academy account and credentials to view this and other videos.

Backup, Restore, and Cloning using OpsCenter Backup Service with Amazon S3

Configuring the Backup Service

Configuring commit log backups

Commit log backups facilitate restoring backup data to a particular point-in-time (page 421).

If commit log archiving is enabled on a cluster, OpsCenter does not automatically enable commit log archiving on the new nodes when adding new nodes (page 563) to a cluster using Lifecycle Manager. The Commit Log Archiving Setting Enabled Consistency Rule in the Best Practice Service (page 502) checks to make sure all nodes within a cluster have Commit Log Archiving enabled.
Important: Enabling commit log archiving modifies the Apache Cassandra™ commitlog_archiving.properties configuration file. The archive_commitlog.sh is created when commit log archiving is enabled. If you add nodes to a cluster and commit log archiving is enabled, you must manually copy commitlog_archiving.properties and archive_commitlog.sh to the new nodes prior to starting DataStax Enterprise, or re-enable commit log archiving.

Prerequisites:
- Java 7 or later is required when backing up commit logs.

1. Click **cluster name** Services.
2. Click the **Details** link for the Backup Service.
   The **Activity** tab appears.
3. Click the **Settings** tab.

4. Click the **Configure** link for the **Commit log backup** feature.
   The **Configure commit log backup** dialog appears.
5. Set the slider to On.

6. If you want to backup the commit logs to a local filesystem location or to a cloud storage provider such as Amazon S3, click Add Location. For more information about Locations and Retention Policy, see adding a backup location (page 406).

7. Enter the path where you want to store the commit logs on each node in Backup Directory.

The location set under Backup Directory must be writable both by the user running DataStax Enterprise and the agent user. The DataStax agent user and the DataStax Enterprise user are by default the same user.

Important: If the location runs out of disk space, the backup fails. Monitor the free disk space using available alerts. Configure the free disk space threshold (page 387) to prevent backups from starting if there is not enough free space available. Commit log files record every mutation of the data in a cluster, and can grow quite large over time (see Logging writes and memtable storage). The disk performance of the commit log backup location is extremely important because the disk write speed can likely be a bottleneck for write-heavy use cases, and read performance a bottleneck for restore operations.

8. Click Save.

9. After enabling (or disabling) commit log backups, OpsCenter prompts you to confirm a rolling restart (page 283) of the cluster.
The **Status** column displays On for the **Status** of the Commit Log Backup feature.

![DataStax OpsCenter 6.1 Snapshot](image)

10. Click **View Archive Times** at any time to view the data and time for the archived commit logs.

The **Commit log backups** dialog displays the Last Archived date and time for each node. When restoring a backup to a specific point in time, all nodes in the cluster must have a commit log backup for the desired time.

![Commit log backups](image)

**Configuring the free disk space threshold for backups**

Set a threshold for free disk space that must be available before a backup operation can start. Ensuring there is sufficient free disk space helps prevent backup failures.

- To keep apprised of free disk space status, set up the Free Disk Space alert: **Alerts**#Manage Alerts #Add Alerts#Notify... Advanced #System#Free Disk Space (GB).
- To monitor free disk space on a specific disk partition in a dashboard graph view: Click **Add Graph**#Add Metric#OS: Disk Free.
1. Click `cluster name`Services.

2. Click the Details link for the Backup Service.

3. Click the Settings tab.

4. Click the Configure link for the Disk Space Threshold feature.

   The **Configure Disk Space Threshold** dialog appears.
5. Select **Do not backup if free space is below** % and enter the percentage of free space that must be available before a backup operation can start.

6. Click **Save**.

   If a backup is attempted when the free disk space is below the configured threshold, the backup fails and an **Error** dialog displays the free disk space for each node.

![Error dialog](image)

**What's next:**

In addition to configuring the free disk space threshold for backups, set up alerts for free and used disk space to **monitor free space when restoring** *(page 427)* backups.

**Configuring encryption key storage for backups**

Configure whether OpsCenter stores the encryption keys for each node along with the SSTables. When tables in a cluster use the DataStax Enterprise **Transparent data encryption**, encryption keys from each node are stored in remote locations alongside the data. Encryption key storage is enabled by default in the OpsCenter Backup Service and is highly recommended.

**Note:** Any encryption keys associated with a table will be backed up.

1. Click **cluster name**#Services.

2. Click the **Details** link for the Backup Service.

   The **Activity** tab appears.

3. Click the **Settings** tab.
4. Click the **Configure** link for **Encryption Key Storage**.

5. Set the slider to **On** to enable or **Off** to disable storing encryption keys alongside the backup data.

If Encryption Key Storage is enabled and a cluster has encrypted keyspaces, the encryption key for each node is stored in the backup location along with the data. If you disable this option, you must ensure that the encryption key is available on all nodes prior to restoring encrypted tables.

6. Click **Save**.

**Configuring custom scripts to run before and after backups**

Configure custom scripts to run before or after a backup. Specify custom scripts in the Pre- and Post-Backup Script fields in the Create Backup dialog.

Custom backup scripts must be located in:

- **Package installs**: `/usr/share/datastax-agent/bin/backup-scripts`
- **Tarball installs**: `install location/agent/bin/backup-scripts`
The `backup-scripts` directory also contains example scripts. The scripts must be executable, and run as the DataStax agent user (by default the Cassandra user). Any custom scripts should exit with a status of 0 if all operations completed successfully. Otherwise, the script should exit with a non-zero status to indicate a failure.

Post-backup scripts are sent a list of files in the backup to `stdin`, one file per line, but do not have any arguments passed to them.

**Configuring restore to continue after a download failure**

Override the default agent configuration to allow the restore operation to continue after a file download fails. If a file fails to download from S3 during a restore, the restore operation fails by default.

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

- **Package installations**: `/var/lib/datastax-agent/conf/address.yaml`
- **Tarball installations**: `install_location/conf/address.yaml`

1. Open the agent configuration file `address.yaml` and set the `restore_on_transfer_failure` option to `true`:

```
restore_on_transfer_failure: true
```

2. Restart *(page 76)* the agent.

**Backing up a cluster**

OpsCenter provides a way to schedule and run backup operations on a cluster. OpsCenter allows organizations to run one-time backup jobs as well as schedule backup jobs to run at a later date and on a recurring basis. Commit log backups facilitate restoring backup data to a particular date and time.

**Note:** Consider the following caveats when creating and restoring backups:

- Restoring a snapshot that contains only the system keyspace is not allowed. There must be both system and non-system keyspaces, or only non-system keyspaces in the snapshot you want to restore.
- Restoring a snapshot that does not contain a table definition is not allowed.
- Restoring from a backup while Kerberos is enabled is not currently supported by OpsCenter.
- Restoring a snapshot to a location with insufficient disk space fails. The Restore Report indicates which nodes do not have sufficient space and how much space is necessary for a successful restore. For more information and tips
for preventative measures, see Monitoring sufficient disk space for restoring backups (page 427).

Creating a recurring scheduled backup

Create a scheduled backup job to automatically run periodic backups.

Prerequisites:

Important: The Backup Service requires control over the data and structure of its destination locations. The AWS S3 bucket and the Local file system destinations must be dedicated for use only by OpsCenter. Any additional directories or files in those destinations can prevent the Backup Service from properly conducting a Backup or Restore operation.

1. Click cluster name#Services.

2. Click the Details link for the Backup Service.

   The Activity tab appears.

3. In the Scheduled Backups tab, click Create New.

   The Create Backup dialog appears with Schedule selected as the Type.
4. Select a date, time, and repeat frequency for the backup schedule.

5. Make any other selections using the following table for guidance.

Table 11: Create Backup fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Type of backup job to create: Run now or Scheduled.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| **Schedule** | Schedule backup settings by date and time. Only appears when **Schedule** is selected as the Type. Available settings:  
  - Date: Day, Month, and Year.  
  - Time and Timezone: GMT is the default timezone. To change the timezone, click **GMT**, select the country and timezone, and click **Save**.  
  - Repeat or Don't Repeat settings: Use to schedule a backup job on a regular periodic basis, or run a scheduled backup during off hours. |
| **Keyspaces** | Required. Allows selecting one or more or all keyspaces for a backup.  
When backing up a DSE Graph database, select your graph keyspace. When a keyspace is created in DSE Graph, it is actually comprised of three keyspaces that OpsCenter bundles together because all three keyspaces are required for a DSE Graph backup.  
**Tip:** Click the plus (+) icon to select one or more options. To keep the list open when making multiple selections, press and hold the Cmd key (Mac) or Ctrl key (Windows/Linux). |
| **Datacenters** | Required. The Datacenters field only appears if there are multiple datacenters configured for a cluster. Allows selecting one or more or all datacenters for a backup.  
Running a backup by a specific datacenter improves backup performance under most circumstances, because the process eliminates backing up data redundantly since the data is replicated to other datacenters anyway.  
**Warning:** When limiting a backup by datacenter, if the SimpleStrategy replication strategy is used, or the NetworkTopologyStrategy is used with a replication factor of 0 for a datacenter, the snapshot might not contain all the data for the cluster.  
**Tip:** Click the plus (+) icon to select one or more options. To keep the list open when making multiple selections, press and hold the Cmd key (Mac) or Ctrl key (Windows/Linux). |
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alert on Failure</td>
<td>Activates an alert upon backup failure that sends an alert to the event log. Default: False. <strong>Tip:</strong> Enabling Alert on Failure is particularly useful for monitoring scheduled overnight backups. Configure alerts for events (<a href="#">page 138</a>) and route the alerts through email, or post URLs (<a href="#">page 145</a>) to a chat room such as HipChat or Slack.</td>
</tr>
<tr>
<td>Current Data Size</td>
<td>Current data size is a best estimate of how large the snapshot will be based on the currently selected keyspace or keyspaces. The information is based on table metrics collected by OpsCenter. Current Data Size is a dynamic read-only field that displays Unknown when size information is not available for the selected keyspaces. An Unknown Data Size typically displays when either a system keyspace is selected or keyspace size information has not been collected yet.</td>
</tr>
<tr>
<td>Location</td>
<td>Snapshots are saved to the node's snapshot directory for the table being saved. For example, <code>/var/lib/cassandra/data/OpsCenter/settings/snapshots</code>. Add more locations to back up to and restore from, including Local file system and Amazon S3 locations, by clicking <strong>Add Location</strong>. Set a retention policy for the backup data in each location of a backup job. See <a href="#">Adding a backup location</a>.</td>
</tr>
<tr>
<td>Staging Directory</td>
<td>The location of the Backup Service staging directory for backups. The default location is <code>/var/lib/datastax-agent/tmp</code>. To change the default location of the directory, indicate the path using the <code>tmp_dir</code> (<a href="#">page 211</a>) configuration option in <code>address.yaml</code>.</td>
</tr>
<tr>
<td>Advanced Options</td>
<td>Expands the Pre-Backup Script and Post-Backup Script fields for running scripts before or after (or both) snapshots are taken. The script must be located in the appropriate install directory on each node. Custom backup scripts must be located in:</td>
</tr>
</tbody>
</table>
|                       | • **Package installs:** `/usr/share/datastax-agent/bin/backup-scripts`  
|                       | • **Tarball installs:** `install location/agent/bin/backup-scripts`  
|                       | For details, see Configuring custom scripts to run before and after backups ([page 390](#)).                                                                                                                                                                                                                                               |
| Create Backup         | Creates a scheduled or runs an ad hoc backup job immediately depending on the selected backup Type (Scheduled or Run Now).                                                                                                                                                                                                                   |
6. Click **Create Backup**.

The scheduled backup appears in the **Scheduled Backups** tab, which displays the keyspaces, datacenters, backup locations, backup schedule, and the timestamp of the last scheduled backup. You can also edit or delete scheduled backup (page 396)s from the Scheduled Backups tab.

**Editing or deleting scheduled backups**

Change the schedule or other details of a scheduled backup. Delete a scheduled backup job.

1. Click **cluster name**#Services.

2. Click the **Details** link for the Backup Service.
   
   The **Activity** tab appears.

3. Click the **Scheduled Backups** tab.

   - To edit a backup, click the **Edit** link for the scheduled backup job you want to edit. Make any changes in the Create Backup dialog and click **Save Backup**.
   - To delete a backup, click the **Delete** link for the scheduled backup job you want to delete. Click **Delete Backup** in the prompt dialog.

**Scheduling a one-off backup**

Schedule a backup to run once later, such as during off-peak hours, rather than running an ad hoc backup (page 400) now.

Prerequisites:
**Important:** The Backup Service requires control over the data and structure of its destination locations. The AWS S3 bucket and the Local file system destinations must be dedicated for use *only* by OpsCenter. Any additional directories or files in those destinations can prevent the Backup Service from properly conducting a Backup or Restore operation.

1. Click *cluster name*Services.

2. Click the Details link for the Backup Service.
   
   The Activity tab appears.

3. In the Scheduled Backups tab, click Create New.
   
   The Create Backup dialog appears with Schedule selected as the Type.

4. Select a date and time for the backup.

5. In the Schedule section, select the Don't Repeat option for the frequency.
6. Make any other selections using the following table for guidance.

**Table 12: Create Backup fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type</strong></td>
<td>Type of backup job to create: Run now or Scheduled.</td>
</tr>
</tbody>
</table>
| **Schedule** | Schedule backup settings by date and time. Only appears when Schedule is selected as the Type. Available settings:  
  - Date: Day, Month, and Year.  
  - Time and Timezone: GMT is the default timezone. To change the timezone, click GMT, select the country and timezone, and click Save.  
  - Repeat or Don't Repeat settings: Use to schedule a backup job on a regular periodic basis, or run a scheduled backup during off hours. |
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| Keyspaces     | Required. Allows selecting one or more or all keyspaces for a backup.  
|               | When backing up a DSE Graph database, select your graph keyspace. When a keyspace is created in DSE Graph, it is actually comprised of three keyspaces that OpsCenter bundles together because all three keyspaces are required for a DSE Graph backup.  
|               | **Tip:** Click the plus (+) icon to select one or more options. To keep the list open when making multiple selections, press and hold the Cmd key (Mac) or Ctrl key (Windows/Linux).                                                                                                                                                        |
| Datacenters   | Required. The Datacenters field only appears if there are multiple datacenters configured for a cluster. Allows selecting one or more or all datacenters for a backup. Running a backup by a specific datacenter improves backup performance under most circumstances, because the process eliminates backing up data redundantly since the data is replicated to other datacenters anyway.  
|               | **Warning:** When limiting a backup by datacenter, if the SimpleStrategy replication strategy is used, or the NetworkTopologyStrategy is used with a replication factor of 0 for a datacenter, the snapshot might not contain all the data for the cluster.  
|               | **Tip:** Click the plus (+) icon to select one or more options. To keep the list open when making multiple selections, press and hold the Cmd key (Mac) or Ctrl key (Windows/Linux).                                                                                                                                                        |
| Alert on Failure | Activates an alert upon backup failure that sends an alert to the event log. Default: False.  
<p>|               | <strong>Tip:</strong> Enabling Alert on Failure is particularly useful for monitoring scheduled overnight backups. [Configure alerts for events](page 138) and route the alerts through email, or [post URLs](page 145) to a chat room such as HipChat or Slack.                                                                                                                                 |
| Current Data Size | Current data size is a best estimate of how large the snapshot will be based on the currently selected keyspace or keyspaces. The information is based on table metrics collected by OpsCenter. Current Data Size is a dynamic read-only field that displays Unknown when size information is not available for the selected keyspaces. An Unknown Data Size typically displays when either a system keyspace is selected or keyspace size information has not been collected yet. |</p>
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Snapshots are saved to the node’s snapshot directory for the table being saved. For example, /var/lib/cassandra/data/OpsCenter/settings/snapshots. Add more locations to back up to and restore from, including Local file system and Amazon S3 locations, by clicking Add Location. Set a retention policy for the backup data in each location of a backup job. See Adding a backup location (page 406).</td>
</tr>
<tr>
<td>Staging Directory</td>
<td>The location of the Backup Service staging directory for backups. The default location is /var/lib/datastax-agent/tmp. To change the default location of the directory, indicate the path using the tmp_dir (page 211) configuration option in address.yaml.</td>
</tr>
</tbody>
</table>
| Advanced Options    | Expands the Pre-Backup Script and Post-Backup Script fields for running scripts before or after (or both) snapshots are taken. The script must be located in the appropriate install directory on each node. Custom backup scripts must be located in:  
  • Package installs: /usr/share/datastax-agent/bin/backup-scripts  
  • Tarball installs: install location/agent/bin/backup-scripts  
  For details, see Configuring custom scripts to run before and after backups (page 390).  
| Create Backup       | Creates a scheduled or runs an ad hoc backup job immediately depending on the selected backup Type (Scheduled or Run Now).                                                                                     |

7. Click Create Backup.

**Running an ad hoc backup**

Run an ad hoc backup. A one-time backup runs only once and starts immediately after creating the job in the OpsCenter UI. You can also schedule a one-time backup (page 396) that does not repeat if you do not want to run the backup now.

**Prerequisites:**

**Important:** The Backup Service requires control over the data and structure of its destination locations. The AWS S3 bucket and the Local file system destinations must be dedicated for use only by OpsCenter. Any additional directories or files in those destinations can prevent the Backup Service from properly conducting a Backup or Restore operation.
1. Click *cluster name#Services*.

2. Click the **Details** link for the Backup Service.
   The **Activity** tab appears.

3. In the **Activity** tab, click **Create Backup**.
   The Create Backup dialog appears. The **Run Now** option is already selected for the backup Type.

4. Make any other selections using the following table for guidance.

   **Table 13: Create Backup fields**
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Type of backup job to create: <strong>Run now</strong> or <strong>Scheduled</strong>.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| **Schedule** | Schedule backup settings by date and time. Only appears when **Schedule** is selected as the Type. Available settings:  
  • Date: Day, Month, and Year.  
  • Time and Timezone: GMT is the default timezone. To change the timezone, click **GMT**, select the country and timezone, and click **Save**.  
  • Repeat or Don't Repeat settings: Use to schedule a backup job on a regular periodic basis, or run a scheduled backup during off hours. |
| **Keyspaces** | Required. Allows selecting one or more or all keyspaces for a backup.  
When backing up a DSE Graph database, select your graph keyspace. When a keyspace is created in DSE Graph, it is actually comprised of three keyspaces that OpsCenter bundles together because all three keyspaces are required for a DSE Graph backup.  
**Tip:** Click the plus (+) icon to select one or more options. To keep the list open when making multiple selections, press and hold the Cmd key (Mac) or Ctrl key (Windows/Linux). |
| **Datacenters** | Required. The Datacenters field only appears if there are multiple datacenters configured for a cluster. Allows selecting one or more or all datacenters for a backup. Running a backup by a specific datacenter improves backup performance under most circumstances, because the process eliminates backing up data redundantly since the data is replicated to other datacenters anyway.  
**Warning:** When limiting a backup by datacenter, if the SimpleStrategy replication strategy is used, or the NetworkTopologyStrategy is used with a replication factor of 0 for a datacenter, the snapshot might not contain all the data for the cluster.  
**Tip:** Click the plus (+) icon to select one or more options. To keep the list open when making multiple selections, press and hold the Cmd key (Mac) or Ctrl key (Windows/Linux). |
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alert on Failure</td>
<td>Activates an alert upon backup failure that sends an alert to the event log. Default: False.</td>
</tr>
<tr>
<td>Current Data Size</td>
<td>Current data size is a best estimate of how large the snapshot will be based on the currently selected keyspace or keyspaces. The information is based on table metrics collected by OpsCenter. Current Data Size is a dynamic read-only field that displays Unknown when size information is not available for the selected keyspaces. An Unknown Data Size typically displays when either a system keyspace is selected or keyspace size information has not been collected yet.</td>
</tr>
<tr>
<td>Location</td>
<td>Snapshots are saved to the node's snapshot directory for the table being saved. For example, /var/lib/cassandra/data/opscenter/settings/snapshots. Add more locations to back up to and restore from, including Local file system and Amazon S3 locations, by clicking Add Location. Set a retention policy for the backup data in each location of a backup job. See Adding a backup location (page 406).</td>
</tr>
<tr>
<td>Staging Directory</td>
<td>The location of the Backup Service staging directory for backups. The default location is /var/lib/datastax-agent/tmp. To change the default location of the directory, indicate the path using the tmp_dir (page 211) configuration option in address.yaml.</td>
</tr>
<tr>
<td>Advanced Options</td>
<td>Expands the Pre-Backup Script and Post-Backup Script fields for running scripts before or after (or both) snapshots are taken. The script must be located in the appropriate install directory on each node.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Create Backup</td>
<td>Creates a scheduled or runs an ad hoc backup job immediately depending on the selected backup Type (Scheduled or Run Now).</td>
</tr>
</tbody>
</table>
5. Click **Create Backup**.

The ad hoc backup starts running. A progress dialog displays the status of the backup operation. If the progress dialog is closed, you can continue to view the operation in the **Activities** tab.

**Syncing a snapshot to a location**

Manually synchronize an On Server backup snapshot to a location.

1. Click *cluster name*#Services.

2. Click the **Details** link for the Backup Service.

3. In the **Activity** tab, click on an On Server backup type that completed successfully.

   The **Backup Report** dialog appears.

   ![Backup Report](image)

4. Click **Sync to Location**.

   The **Sync Snapshot to Location** dialog appears.
5. Click **Add Location**.
   The **Add Location** dialog appears. Add a location *(page 406)* to sync the
   snapshot to. After clicking **Save Location**, the defined location displays in the **Sync
   Snapshot to Location** dialog.

6. Select an **Alert on Failure** option.

7. Click **Sync Snapshot**.
   The **Sync Snapshot Report** dialog displays the progress.
Adding a backup location

Add or edit an additional backup location beyond On Server. Additional locations include Amazon S3 or a Local file system. Adding another backup location besides On Server is a recommended best practice.

**Important:** The Backup Service requires control over the data and structure of its destination locations. The AWS S3 bucket and the Local file system destinations must be dedicated for use only by OpsCenter. Any additional directories or files in those destinations can prevent the Backup Service from properly conducting a Backup or Restore operation.

Backup location options include:

- **On Server** is the default backup location option and is primarily a staging directory. Because the On Server backup data has to reside on the same disks as the original data, there is an underlying hard link between those directories that provides for very fast transfer when backing up or restoring data. While this option does not prevent data loss in the event of disk failure, it provides the ability to quickly restore data, such as in situations where a keyspace was inadvertently dropped.

- **A Local FS** backup location performs a copy operation of the data rather than a hard link like On Server, because the expectation is that the Local FS location mounts to a different drive.
• **Amazon S3 (Simple Storage Service)** is a cloud file storage web service offered by Amazon. For more information about S3, see the Amazon documentation.

### Adding a local file system backup location

Add a local file system location (such as NFS mount) when creating or editing (page 396) a backup job. Backup to and restore from a custom local directory.

Backups to different external locations are supported. When the snapshots are uploaded, they are further placed in subdirectories based on a unique identifier for the node. Because of this scheme, any number of clusters or nodes could refer to the same external mount point. If you want the nodes to back up to different locations, simply mount the same named directory on different nodes to different external locations.

**Prerequisites:**

The path for the local file system should at least be mounted to a different drive. Mounting to an external drive or shared file system is preferable and recommended. Not doing so for local file system backups results in copying the tables between locations on the hard disk, which requires double the snapshot size in disk space. A best practice is to avoid backups on the same server located where the original data exists.

**Configuration approaches for the backup location:**

- **On Server** is essentially the same increased speed and inherent data loss risk as Local FS on the same drive.
- Local FS on a different drive but the same server is an improvement in backup safety over On Server or Local FS on the same drive.
- Local FS on a shared or distributed network drive provides the best safety for backups, with some tradeoff in decreased speed due to distribution latency.

A Local FS destination must be unique and self-contained. Any defined directory destination cannot be contained within another backup destination. For example, if you set up a backup location to mydir1, do not designate another backup location to mydir1/mysubdir1. If you set up a mydir1/mysubdir1 location, do not set up another location as mydir1/mysubdir1/mydeeper/subdir1. Subdirectories are supported; however, filepaths cannot share any portion of a backup destination.

**Important:** The Backup Service requires control over the data and structure of its destination locations. The AWS S3 bucket and the Local file system destinations must be dedicated for use only by OpsCenter. Any additional directories or files in those destinations can prevent the Backup Service from properly conducting a Backup or Restore operation.

1. **Access the Create (or Edit (page 396)) Backup dialog:**
   a. Click **cluster name##Services**.
   b. Click the **Details** link for the Backup Service.
c. In the Activity tab, click Create Backup.

2. In the Create or Edit Backup dialog, click +Add Location.
   The Add Location dialog appears.

3. Select Local FS in Location.

4. Enter the Path to the local file system location.

5. Select any throttling or compressing of the data:
   a. To avoid saturating your network, set a maximum upload rate. Select Throttle transfer rate and set the maximum MB per second.
   b. To compress the backup data, select Enable compression. Compression reduces the amount of data going through your network and reduces the disk and data usage but increases the CPU load for the server.

6. For scheduled backups, indicate how long the snapshot data should be retained by selecting a Retention Policy (page 381). Retain All (default) saves the snapshot data indefinitely. Or, define a set period of time. After the snapshot data is older than the time set in Retention Policy, the snapshot data is deleted.

   DataStax strongly recommends setting a retention policy to periodically remove backups. This practice helps to avoid long-term performance issues caused by an excessive number of backups.

   Note: Setting a Retention Policy is not available for an ad hoc (Run Now) backup.
7. Click **Save Location**.

   The newly added Local FS location displays in the **Location** pane of the Create or Edit Backup dialog.

   ![Location pane](image)

   Click the edit icon to edit a location and its retention policy if applicable. Click the delete icon to delete a location. The On Server location cannot be deleted.

8. Click **Save Backup**, or **Create Backup** as applicable.

**Adding an Amazon S3 backup location**

Add an Amazon S3 backup location. For more details, see [backing up to Amazon S3](page 382).

**Prerequisites:**

- Java 8 is required to store at an S3 location.
- Make sure you have the proper AWS IAM privileges.
- An S3 bucket destination must be unique and self-contained. Any defined destination cannot be contained within another backup destination. For example, if you set up a backup location to mybucket1, do not designate another backup location to mybucket1/myfolder1. If you set up a mybucket1/myfolder1 location, do not set up another location as mybucket1/myfolder1/mysubfolder1. Folders are supported; however, bucket paths cannot share any portion of a backup destination.
- As a recommended best practice, limit an S3 bucket to a single keyspace for OpsCenter backups. Because every backup job gathers a list of all existing data files before the transfer to S3 can start, that process can take more time as the number of files grow in the bucket.

   **Important:** Moving backup files from Amazon S3 to Amazon Glacier is not supported by the OpsCenter Backup Service.

**Important:** The Backup Service requires control over the data and structure of its destination locations. The AWS S3 bucket and the Local file system destinations must be dedicated for use *only* by OpsCenter. Any additional directories or files in those destinations can prevent the Backup Service from properly conducting a Backup or Restore operation.
**Warning:** The AWS credentials and bucket names are stored in cluster_name.conf (with the exception of ad hoc backups). Be sure to use proper security precautions to ensure that this file is not readable by unauthorized users.

1. Access the Create (or Edit *(page 396)*) Backup dialog:
   a. Click *cluster name*#Services.
   b. Click the Details link for the Backup Service.
   c. In the Activity tab, click Create Backup.

2. In the Create or Edit Backup dialog, under Location, click +Add Location.
   The Add Location dialog appears.

   ![Add Location dialog]

   3. Select Amazon S3 as the backup Location.
   4. Enter the S3 Bucket name.

   **Note:** The bucket name must be at least 4 characters long. Bucket names should only contain lowercase letters, numbers, and hyphens. Additionally, OpsCenter requires that bucket prefixes contain only lowercase letters, numbers, and safe characters. See the S3 guidelines for more details about bucket naming restrictions.

   **Tip:** To indicate a bucket subfolder location, delineate the bucket name from the folder name with a forward slash (/) character. Example: mybucket/
myfolder/mysubfolder. Remember that slashes are not allowed within bucket or folder names themselves.

5. Select the source type of your AWS credentials.

   **Warning:** The AWS credentials and bucket names are stored in `cluster_name.conf` (with the exception of ad hoc backups). Be sure to use proper security precautions to ensure that this file is not readable by unauthorized users.

<table>
<thead>
<tr>
<th>User-Supplied Credentials</th>
<th>Enter your <strong>AWS key</strong> and <strong>AWS Secret</strong>. See <a href="#">AWS Access Keys</a>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS Credential Provider chain</td>
<td>Use the default credential provider chain to locate AWS credentials. See <a href="#">Working with AWS Credentials</a>.</td>
</tr>
</tbody>
</table>

6. Select any throttling, compressing, or encryption of the data:

   a. To avoid saturating your network, set a maximum upload rate. Select **Throttle S3 transfer rate** and set the maximum MB per second.

      **Note:** When the AWS CLI ([page 24](#)) labs feature is enabled, the S3 throttle is ignored at this time. A tooltip also mentions this current limitation. See [Tuning throttling when using AWS CLI](#).

   b. To compress the backup data, select **Enable compression**. Compression reduces the amount of data going through your network and reduces the disk and data usage but increases the CPU load for the server.

   c. To enable server-side S3 encryption (SSE-S3), select **Enable S3 server-side encryption**. Server-side encryption encrypts each file in the backup set with a unique key, as well as the key itself, using a 256-bit AES cypher. The key is encrypted with a master key that is regularly rotated. Enabling server-side encryption increases the security of your backup files, but increases the time it takes to complete a backup. For more information on S3 server-side encryption, see [Using Server Side Encryption](#).

   d. To backup nodes running in multiple regions to a single bucket, select **Enable S3 transfer acceleration**. Instead of traffic crossing over the internet, acceleration mode uses Amazon CloudFront to cache S3 requests. Because the CloudFront servers are closer to the nodes in each region, the backup latency is reduced.

      **Note:** Enabling S3 transfer acceleration can cause performance degradation, and might slow a standard backup configuration. Use
this option only if backing up nodes in multiple regions to a single bucket.

7. For scheduled backups, indicate how long the snapshot data should be retained by selecting a Retention Policy (page 381). Retain All (default) saves the snapshot data indefinitely. Or, define a set period of time. After the snapshot data is older than the time set in Retention Policy, the snapshot data is deleted.

DataStax strongly recommends setting a retention policy to periodically remove backups. This practice helps to avoid long-term performance issues caused by an excessive number of backups.

   **Note:** Setting a Retention Policy is not available for an ad hoc (Run Now) backup.

8. Click **Save Location**.

   The newly added S3 location displays in the **Location** pane of the Create or Edit Backup dialog.

   ![Location pane](image)

   Click the edit icon to edit a location and its retention policy if applicable. Click the delete icon to delete a location. The On Server location cannot be deleted.

9. Click **Save Backup**, or **Create Backup** as applicable.

**Bulk uploading S3 backups using the AWS CLI**

*address.yaml*

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

- **Package installations:** `/var/lib/datastax-agent/conf/address.yaml`
- **Tarball installations:** `install_location/conf/address.yaml`

*cluster_name.conf*

The location of the *cluster_name.conf* file depends on the type of installation:
• Package installations: /etc/opscenter/clusters/cluster_name.conf
• Tarball installations: install_location/conf/
  clusters/cluster_name.conf

Use the AWS CLI instead of the AWS SDK when bulk loading backups to Amazon S3 locations. Using the AWS CLI rather than the AWS SDK can result in a performance increase, with a noticeable decrease in the time it takes to complete a backup. This is an OpsCenter Labs feature (that is, under ongoing development but available for use). The feature is available in OpsCenter versions 6.1.3 and later.

For more information, see AWS CLI in the Amazon documentation.

**Note:** When the AWS S3 CLI is enabled, the S3 throttling setting is ignored by OpsCenter during backups. See Tuning throttling *(page 25)* for AWS CLI.

**Prerequisites:**

• Install the AWS CLI package on every node. DataStax recommends using the Amazon bundled installer method and upgrading to the latest version of AWS CLI if it is already installed. See Install the AWS CLI using the bundled installer in the Amazon documentation for installation procedures.

  **Tip:** As a recommended best practice for OpsCenter, install the AWS CLI bundle using APT as follows:

  ```
  sudo apt-get install -y unzip
  unzip awscli-bundle.zip
  sudo ./awscli-bundle/install -i /usr/local/aws -b /usr/local/bin/aws
  ```

  **Important:** Regardless of the install procedure used, make sure that the AWS CLI package is installed in the PATH of the cassandra user, or whichever user the DataStax agent runs as *(page 67).*

• Add an S3 location *(page 409)* for backups.

1. Open cluster_name.conf for editing. Substitute `cluster_name` with the name of your cluster. Setting agent options through the cluster configuration file sets the corresponding property in address.yaml on every node.

   If necessitated by your environment, open address.yaml for editing and configure at the node level. Do so for every node that requires a specific configuration override.

2. Add the following configuration option:

   ```
   [labs]
   ```
3. Save the configuration file or files.

4. Restart (page 75) the OpsCenter daemon.

5. If you made changes to address.yaml (page 205), restart (page 76) the DataStax agents.

What’s next: Tune throttling (page 25) for AWS CLI

**Tuning throttling when using AWS CLI**

Use alternative throttle options when using the AWS CLI for bulk uploads because the OpsCenter S3 throttle is ignored at this time when the OpsCenter AWS CLI for S3 Labs feature (page 24) is enabled.

1. Adjust the max_concurrent_requests available in the AWS SDK. Refer to the AWS CLI S3 configuration documentation for details.

2. If necessary, use a tool such as Trickle to limit bandwidth.

**Restoring a cluster**

Restore data to a cluster from local keyspace backups and backups stored to cloud storage providers like Amazon S3. These restores can be from a particular point-in-time if commit log backups are enabled.

When performing a restore operation, you can restore all the keyspaces from a backup or select specific keyspaces and tables.

**Note:** Consider the following caveats when creating and restoring backups:

- Restoring a snapshot that contains only the system keyspace is not allowed. There must be both system and non-system keyspaces, or only non-system keyspaces in the snapshot you want to restore.
- Restoring a snapshot that does not contain a table definition is not allowed.
- Restoring from a backup while Kerberos is enabled is not currently supported by OpsCenter.
- Restoring a snapshot to a location with insufficient disk space fails. The Restore Report indicates which nodes do not have sufficient space and how much space is necessary for a successful restore. For more information and tips for preventative measures, see Monitoring sufficient disk space for restoring backups (page 427).

When restoring from backups stored on Amazon S3, OpsCenter chooses an agent to determine which nodes in the cluster have data that needs to be restored. The SSTables stored in the S3 bucket are sorted into directories with the node ID of the original node. If
the cluster topology is unchanged from when the backup was taken, OpsCenter instructs
each node to restore the set of SSTables that were stored on that node before. If the cluster
topology has changed since the backup was completed, OpsCenter attempts to match
the SSTables to the node that originally stored the SSTable, and distributes the remaining
SSTables to the remaining nodes to balance the load evenly.

**Note:** The Restore feature of the Backup Service leverages the sstableloader utility,
which currently requires enabling the thrift server on all nodes before restoring.

### Restoring from a backup

Restore data from any local or Amazon S3 backups that have been run by OpsCenter. You
cannot use the OpsCenter Backup Service to restore from snapshots run with nodetool.
You can pick any subset of tables that exist in the snapshot to restore.

**Note:** If the backup contains encrypted tables created prior to DataStax Enterprise
4.0.4 or 4.5.2, you will not be able to restore the snapshot. Due to a bug in
Cassandra, backups containing encrypted table data from versions prior to 4.0.4 and
4.5.2 do not contain the necessary keys to restore the backup.

**Prerequisites:**

- To restore an encrypted backup, the agent must be granted password-less sudo
  access on the DataStax Enterprise nodes. This has already been granted if you used
  OpsCenter to install the agents. If you are running the agent as a different user than
  DataStax Enterprise and need to restore encrypted tables, you must manually restore
  the system_key table.
- The Restore feature of the Backup Service leverages the sstableloader utility, which
currently requires enabling the thrift server on all nodes before restoring. Before
  restoring, ensure the thrift server is enabled on all nodes.
- When restoring tables that are Solr cores, if the table does not already exist, it will be
  automatically re-created as a CQL table. If you require this to be a thrift-based table,
  manually recreate the table prior to restoring. If you are restoring data from a thrift table
  that no longer exists, you are responsible for creating the table prior to restoring.

**Important:** The Backup Service requires control over the data and structure of its
destination locations. The AWS S3 bucket and the Local file system destinations
must be dedicated for use only by OpsCenter. Any additional directories or files
in those destinations can prevent the Backup Service from properly conducting a
Backup or Restore operation.

**Note:** Consider the following caveats when creating and restoring backups:

- Restoring a snapshot that contains only the system keyspace is not allowed.
  There must be both system and non-system keyspaces, or only non-system
  keyspaces in the snapshot you want to restore.
- Restoring a snapshot that does not contain a table definition is not allowed.
- Restoring from a backup while Kerberos is enabled is not currently supported by
  OpsCenter.
• Restoring a snapshot to a location with insufficient disk space fails. The Restore Report indicates which nodes do not have sufficient space and how much space is necessary for a successful restore. For more information and tips for preventative measures, see Monitoring sufficient disk space for restoring backups (page 427).

1. Click **cluster name** Services.

2. Click the **Details** link for the Backup Service.

3. In the **Activity** tab, click **Restore Backup**.

   The Restore from Backup dialog appears.

4. Select the backup to restore in the list of backups and click **Next**.

   a. The **Backups** tab lists the available keyspace backups, including both scheduled and manual backups.

   b. If you are restoring from a location that is not listed in the **Backups** tab, select **Other Location**.
Selecting a backup from Other Location is most commonly used when cloning a cluster (page 435), but can be used when this OpsCenter instance is not aware of the backup location.

5. Select the tables included in the backup you want to restore. Click the keyspace name to include all the tables in the keyspace. Click All Keyspaces to restore all the keyspaces.

![Restore from Backup](image)

To select only specific tables, expand the keyspace name and select the tables.

**Note:** Consider the following caveats when creating and restoring backups:

- Restoring a snapshot that contains only the system keyspace is not allowed. There must be both system and non-system keyspaces, or only non-system keyspaces in the snapshot you want to restore.
- Restoring a snapshot that does not contain a table definition is not allowed.
- Restoring from a backup while Kerberos is enabled is not currently supported by OpsCenter.
- Restoring a snapshot to a location with insufficient disk space fails. The Restore Report indicates which nodes do not have sufficient space and how much space is necessary for a successful restore. For more information and tips for preventative measures, see Monitoring sufficient disk space for restoring backups (page 427).
6. Under **Location**, select the target cluster for the restored data.

   • The **Location** list is only available when there are multiple clusters and both clusters are managed by the same instance of OpsCenter.
   
   • If you select a different cluster than the one that was backed up, the data is cloned to the selected cluster. See [cloning cluster data](page 429).

   **Note:** Restoring encrypted tables *(page 389)* to a different cluster does not work unless the encryption keys are identical, which is typically not the case.

7. To remove the existing keyspace data before the data is restored, select **Truncate/delete existing data before restore**. This completely removes any updated data in the cluster for the keyspaces you are restoring.

8. To prevent overloading the network, set a maximum transfer rate for the restore. Select **Throttle stream throughput at ____ MB** and set the maximum MB per second.

9. Change the staging directory if necessary by setting the `backup_staging_directory` configuration option in `address.yaml` *(page 205)*.

10. Click **Restore Backup**.

    The **Confirm Restore** dialog appears.

    ![Confirm Restore](image)

    **Warning:** If a value was not set for throttling stream output in 8 *(page 418)*, a warning message indicates the consequences of unthrottled restores. The throttle warning only appears for versions of DSE from 4.8.7 and later. Either click Cancel and set the throttle value in the **Restore from Backup** dialog, set the values in `cassandra.yaml` *(stream_throughput_outbound_megabits_per_sec and ...)}
inter_dc_stream_throughput_outbound_megabits_per_sec), or proceed anyway at risk of network bottlenecks.

**Tip:** If you are using LCM to manage DSE cluster configuration, update Cluster Communication settings in `cassandra.yaml` in the config profile (page 594) for the cluster and run a configuration job (page 576). Stream throughput (not inter-dc) is already set to 200 in LCM defaults.

11. Click **Start Restore** to confirm the restore.

   If the pre-restore checks detected schema differences that could not automatically be validated, the Restore Schema Validation dialog appears. Possible issues are listed and a comparison of the backup and current schema are presented side-by-side.
12. Review the information to determine what adjustments if any need to be made to the current schema:

- To rectify the schema issues and try the restore again afterward, click **Cancel**.
- To proceed despite the schema mismatch, click **Continue Restore**.

**Warning:** Attempting to restore a backup with an incompatible schema might result in corrupt or inaccessible data. Before forcing the restore, you might want to back up your current data.

After the restore starts, the **Restore Report** displays detailed information about the progress and status of the restore. The Restore Report dialog can be closed at any time without impacting the restore process. Reopen the report by clicking on the **In Progress** restore in the **Activity** tab. View the Restore Report for any completed restore by clicking on the restore of interest in the Activity tab.
If there is insufficient disk space to restore the backup, the restore fails. See Monitoring sufficient disk space for restoring backups (page 427).

**Restoring a backup to a specific point-in-time**

For a point-in-time restore, OpsCenter intelligently chooses which snapshots and commit logs to restore from based on the date and time you are restoring the cluster to. If an acceptable combination of snapshots and commit logs cannot be found, the restore fails. A detailed error message is visible in the Activity section of the OpsCenter UI.

**dse-env.sh**

The location of the `dse-env.sh` file depends on the type of installation:

- **Package installations:** `/etc/dse/dse-env.sh`
- **Tarball installations:** `install_location/dse/dse-env.sh`

**Prerequisites:**

- For point-in-time restores to work, you must have enabled commit log backups (page 384) and performed at least one snapshot backup (page 391) before the time to which you are restoring.
• The Restore feature of the Backup Service leverages the sstableloader utility, which currently requires enabling the thrift server on all nodes before restoring. Before restoring, ensure the thrift server is enabled on all nodes.

   **Note:** The thrift server is only required for DSE versions earlier than 5.0 (DSE 4.8.x versions).

• When performing a point-in-time restore, the cluster topology must not have changed since the backup. Attempting to perform a point-in-time restore on a cluster whose topology has changed results in a failure. DataStax strongly recommends performing a snapshot backup **both before and after** changing the cluster topology. After changing the topology, you can then restore the cluster based on that backup. If reverting to the previous topology, you can use the backup with the original topology to restore the cluster.

• **Known limitations:**
  # Point-in-time restore cannot restore commit logs for keyspaces or tables that would have to be recreated in Cassandra 2.1 and later, and DataStax Enterprise 4.7 and later.
  # Point-in-time restore fails if any tables were recreated during the time period of the actual point-in-time restore.

**Important:** The Backup Service requires control over the data and structure of its destination locations. The AWS S3 bucket and the Local file system destinations must be dedicated for use only by OpsCenter. Any additional directories or files in those destinations can prevent the Backup Service from properly conducting a Backup or Restore operation.

**Note:** Consider the following caveats when creating and restoring backups:

• Restoring a snapshot that contains only the system keyspace is not allowed. There must be both system and non-system keyspaces, or only non-system keyspaces in the snapshot you want to restore.
• Restoring a snapshot that does not contain a table definition is not allowed.
• Restoring from a backup while Kerberos is enabled is not currently supported by OpsCenter.
• Restoring a snapshot to a location with insufficient disk space fails. The Restore Report indicates which nodes do not have sufficient space and how much space is necessary for a successful restore. For more information and tips for preventative measures, see Monitoring sufficient disk space for restoring backups (*page 427*).

1. Click **cluster name**#Services.
2. Click the **Details** link for the Backup Service.
3. Click **Restore Backup**.
The **Restore from Backup, Step 1 of 2: Select Backup** dialog appears.

4. Click the **Point In Time** tab.

![Restore from Backup, Step 1 of 2: Select Backup dialog](image)

5. Complete your selections:

   a. **Point in Time**: Set the date and time to which you want to restore your data.

   b. **Commit logs location**: Select the location of the commit logs; either On Server, Local FS, or another location on Amazon S3. The location of commit logs is configured when enabling commit log backups *(page 384)*.

   c. **Backup location**: Select the location of the snapshot; either On Server, Local FS, or Amazon S3. Click the **Add Location** link to add another location *(page 406)*.

   d. Click **Next**.

      The **Restore from Backup, Step 2 of 2: Configure and Restore** dialog appears.
6. Complete your selections:

   a. **Keyspaces**: Select the tables included in the backup you want to restore. Click the keyspace name to include all the tables in the keyspace. Click **All Keyspaces** to restore all the keyspaces.

To select only specific tables, expand the keyspace name and select the tables.

   **Note**: Consider the following caveats when creating and restoring backups:

   - Restoring a snapshot that contains only the system keyspace is not allowed. There must be both system and non-system keyspaces, or only non-system keyspaces in the snapshot you want to restore.
   - Restoring a snapshot that does not contain a table definition is not allowed.
   - Restoring from a backup while Kerberos is enabled is not currently supported by OpsCenter.
• Restoring a snapshot to a location with insufficient disk space fails. The Restore Report indicates which nodes do not have sufficient space and how much space is necessary for a successful restore. For more information and tips for preventative measures, see Monitoring sufficient disk space for restoring backups (page 427).

b. Under Location: Select the target cluster for the restored data.
   • The Location list is only available when there are multiple clusters and both clusters are managed by the same instance of OpsCenter.
   • If you select a different cluster than the one that was backed up, the data is cloned to the selected cluster. See cloning cluster data (page 429).

   **Note:** Restoring encrypted tables to a different cluster does not work unless the encryption keys are identical, which is typically not the case.

c. To remove the existing keyspace data before the data is restored, select **Truncate/delete existing data before restore.** This completely removes any updated data in the cluster for the keyspaces you are restoring.

d. To prevent overloading the network, set a maximum transfer rate for the restore. Select **Throttle stream throughput at ____ MB** and set the maximum MB per second.

e. Change the staging directory if necessary by setting the **backup_staging_directory configuration option in address.yml (page 205).**

f. Click the **Edit Restart Settings** link to adjust settings for the rolling restart (page 283).

7. Click **Restore Backup.**
   The **Confirm Restore** dialog appears.
Warning: If a value was not set for throttling stream output in page 418, a warning message indicates the consequences of unthrottled restores. The throttle warning only appears for versions of DSE from 4.8.7 and later. Either click Cancel and set the throttle value in the Restore from Backup dialog, set the values in cassandra.yaml (stream_throughput_outbound_megabits_per_sec and inter_dc_stream_throughput_outbound_megabits_per_sec), or proceed anyway at risk of network bottlenecks.

Tip: If you are using LCM to manage DSE cluster configuration, update Cluster Communication settings in cassandra.yaml in the config profile (page 594) for the cluster and run a configuration job (page 576). Stream throughput (not inter-dc) is already set to 200 in LCM defaults.
Click **Start Restore** to confirm when prompted.

OpsCenter retrieves the backup data and sends the data to the nodes in the cluster. A snapshot restore is completed first, following the same process as a normal snapshot restore. After the snapshot restore successfully completes, OpsCenter instructs all agents in parallel to download the necessary commit logs, followed by a rolling commit log replay across the cluster. Each node is configured for replay and restarted after the previous node finishes successfully.

If an error occurs during a point-in-time restore for a subset of tables, you might need to manually revert changes made to some cluster nodes. To clean up a node, edit `dse-env.sh` and remove the last line that specifies `JVM_OPTS`. For example:

```
export JVM_OPTS="$JVM_OPTS -Dcassandra.replayList=Keyspace1.Standard1"
```

### Monitoring sufficient disk space for restoring backups

OpsCenter monitoring provides multiple features to proactively prevent disk space issues. Sufficient disk space is necessary for restoring backups successfully.

Before downloading a snapshot from a destination to use for a restore, OpsCenter (as of version 6.0.9 or later) verifies whether each node has sufficient disk space required for the download. The disk space check looks up the partition for the agent `tmp_dir` and verifies there is sufficient space on that partition. If necessary, change the agent download directory using `tmp_dir` (page 211) in `address.yaml`. Attempting a restore to a location with insufficient disk space fails. The Restore Report indicates which nodes do not have sufficient space and how much space is necessary on each for a successful restore.

**Tip:** When creating a backup, the **Create Backup** dialog dynamically displays the size of the selected data to back up in the **Current Data Size** field.

### address.yaml

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

- **Package installations:** `/var/lib/datastax-agent/conf/address.yaml`
- **Tarball installations:** `install_location/conf/address.yaml`
Storage Capacity Widget

Monitor free disk space for backups and restores by observing the **Storage Capacity** widget in the **Dashboard**. The **Storage Capacity** widget is enabled by default in the **Dashboard**. When storage capacity becomes limited, add capacity or clear space as necessary.

**Note:** Monitoring disk space for restoring backups assumes the `tmp_dir` resides on the same partition as the backup data.

<table>
<thead>
<tr>
<th>Plentiful free disk space</th>
<th>Low free disk space</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Storage Capacity" /></td>
<td><img src="image2" alt="Storage Capacity" /></td>
</tr>
</tbody>
</table>

**Tip:** Keep abreast of capacity requirements using the **Capacity Service** *(page 498)*. See **Forecasting** *(page 499)*.

Free and Used Disk Space Alerts

Add the **Free Disk Space** alert to notify you when free disk space goes below an acceptable threshold for an environment. Click **Alerts #Advanced#System#System Free Disk Space** from the **Notify me when** menu. You could also set up a **Used Disk Space (GB)** alert.

**Note:** Alerting about disk space for restoring backups assumes the `tmp_dir` resides on the same partition as the backup data.
Cloning cluster data

The OpsCenter Backup Service provides methods of loading data backed up from one cluster in a different target cluster. The Backup Service can load backed up data automatically from backup locations that are accessible to both locations, such as when backing up to S3 or if a Local FS location path points to an NAS device shared by both clusters. The Backup Service can also detect and load backup data for non-centralized Local FS locations if the user copies the appropriate directories from the origin cluster to the target cluster.

There are a few workflows supported in OpsCenter for cloning cluster data:

- Cloning from a defined Other Location allows cloning data from any previous snapshot that a cluster in an OpsCenter instance can access, regardless of the source of that snapshot. This workflow can be thought of as a pull process of cloning in that OpsCenter fetches what it needs from a location to clone the data. The location to clone from can be an Amazon S3 or a Local FS location. When cloning from a Local FS location, ensure that all files from the original backup are available at the specified path on the nodes of the target cluster.

- Cloning a backup from one cluster to another cluster, both of which are managed within the same OpsCenter instance. This workflow can be thought of as a push process of cloning in that an OpsCenter instance pushes a backup it is aware of in one cluster to another cluster managed within that instance.

**Note:** Cloning encrypted tables *(page 389)* to a different cluster does not work unless the encryption keys are identical, which is typically not the case.

**Cloning cluster data from a defined other location**

Clone cluster data from one DSE cluster to another using the Restore Backup feature in OpsCenter. This workflow does not require the source and target clusters to both be managed by the same OpsCenter instance. This workflow does not necessarily even
require another existing cluster instance. The data can be cloned providing it was backed up to an available Local FS or an Amazon S3 location.

This procedure steps you through the basic required selections in each of the three restore dialogs presented during the workflow. For information on the available optional fields, refer to the Restore from backup dialogs fields (page 457).

Prerequisites:

To clone the cluster data, there must be an existing backup of the cluster to a Local FS or an Amazon S3 location. See adding a backup location (page 406).

Note: Cloning encrypted tables (page 389) to a different cluster does not work unless the encryption keys are identical, which is typically not the case.

Important: The Backup Service requires control over the data and structure of its destination locations. The AWS S3 bucket and the Local file system destinations must be dedicated for use only by OpsCenter. Any additional directories or files in those destinations can prevent the Backup Service from properly conducting a Backup or Restore operation.

1. Click the target cluster name Services.

2. Click the Details link for the Backup Service.

3. In the Activity tab, click Restore Backup.

4. Click the Other Location tab.

   The Step 1 of 3: Select Backup Restore from Backup dialog appears.
5. Select the **Location**. Available options are:
   - **Amazon S3** (default): Go to 6 *(page 431)*.
   - **Local FS**: Go to 7 *(page 431)*.

6. If the location is **Amazon S3**:
   a. Enter the **S3 Bucket** name.
   b. Enter your AWS credentials in **AWS Key** and **AWS Secret**.

7. If the location is **Local FS**, enter the **Path** to the backups.

8. Click **Next**.
   The **Step 2 of 3: Select Backup Version** dialog appears populated with the available backups at the selected location.
9. Select the backup to restore and click **Next**.

   The **Step 3 of 3: Configure and Restore** dialog appears.

10. Select the keyspaces or tables from the available **Keyspaces**.
To select only specific tables, expand the keyspace name and select the tables.

**Note:** Consider the following caveats when creating and restoring backups:

- Restoring a snapshot that contains only the system keyspace is not allowed. There must be both system and non-system keyspaces, or only non-system keyspaces in the snapshot you want to restore.
- Restoring a snapshot that does not contain a table definition is not allowed.
- Restoring from a backup while Kerberos is enabled is not currently supported by OpsCenter.
- Restoring a snapshot to a location with insufficient disk space fails. The Restore Report indicates which nodes do not have sufficient space and how much space is necessary for a successful restore. For more information and tips for preventative measures, see Monitoring sufficient disk space for restoring backups *(page 427).*

11. In the **Location** list, select the cluster to clone the data to.

12. Click **Restore Backup**.

   The **Confirm Restore** dialog appears.

   ![Confirm Restore dialog](image)

   **Warning:** If a value was not set for throttling stream output in 8 *(page 418)*, a warning message indicates the consequences of unthrottled restores. The throttle warning only appears for versions of DSE from 4.8.7 and later. Either click **Cancel** and set the throttle value in the **Restore from Backup** dialog, set the values in *cassandra.yaml* (stream_throughput_outbound_megabits_per_sec and...
inter_dc_stream_throughput_outbound_megabits_per_sec), or proceed anyway at risk of network bottlenecks.

**Tip:** If you are using LCM to manage DSE cluster configuration, update Cluster Communication settings in `cassandra.yaml` in the **config profile (page 594)** for the cluster and **run a configuration job (page 576)**. Stream throughput (not inter-dc) is already set to 200 in LCM defaults.

13. Review the information to determine what adjustments if any need to be made to the current schema:
   - To rectify the schema issues and try the restore again afterward, click **Cancel**.
   - To proceed despite the schema mismatch, click **Continue Restore**.

   **Warning:** Attempting to restore a backup with an incompatible schema might result in corrupt or inaccessible data. Before forcing the restore, you might want to back up your current data.

The progress and details of the restore operation are displayed in the **Restore Report**.
Cloning cluster data from clusters managed by the same OpsCenter instance

Clone cluster data from one DSE cluster to another using the Restore Backup feature in OpsCenter. This workflow requires the source and target clusters to both be managed by the same OpsCenter instance.

**Prerequisites:**

To clone the cluster data, there must be an existing backup of the cluster to a Local FS or an Amazon S3 location. See adding a backup location (page 406).

**Note:** Cloning encrypted tables (page 389) to a different cluster does not work unless the encryption keys are identical, which is typically not the case.

**Important:** The Backup Service requires control over the data and structure of its destination locations. The AWS S3 bucket and the Local file system destinations must be dedicated for use only by OpsCenter. Any additional directories or files in those destinations can prevent the Backup Service from properly conducting a Backup or Restore operation.

1. Click the source *cluster name* #Services.
2. Click the **Details** link for the Backup Service.

3. In the **Activity** tab, click **Restore Backup**.
   
The Restore from Backup dialog appears.

![Restore from Backup dialog](image)

4. In the **Backups** tab, select the backup that contains the data you want to clone and click **Next**.
   
   **Note:** On Server backups are not eligible for cloning.

5. Select the tables included in the backup you want to restore. Click the keyspace name to include all the tables in the keyspace. Click **All Keyspaces** to restore all the keyspaces.
To select only specific tables, expand the keyspace name and select the tables.

**Note:** Consider the following caveats when creating and restoring backups:

- Restoring a snapshot that contains only the system keyspace is not allowed. There must be both system and non-system keyspaces, or only non-system keyspaces in the snapshot you want to restore.
- Restoring a snapshot that does not contain a table definition is not allowed.
- Restoring from a backup while Kerberos is enabled is not currently supported by OpsCenter.
- Restoring a snapshot to a location with insufficient disk space fails. The Restore Report indicates which nodes do not have sufficient space and how much space is necessary for a successful restore. For more information and tips for preventative measures, see [Monitoring sufficient disk space for restoring backups](page 427).

6. **Under Location**, select the target cluster for the restored data. Select a different cluster than the one that was backed up to clone the data to the cluster.

   The **Location** list is only available when there are multiple clusters and both clusters are managed by the same instance of OpsCenter.
7. To remove the existing keyspace data before the data is restored, select **Truncate/delete existing data before restore**. This completely removes any updated data in the cluster for the keyspaces you are restoring.

8. To prevent overloading the network, set a maximum transfer rate for the restore. Select **Throttle stream throughput at ____ MB** and set the maximum MB per second.

9. Change the staging directory if necessary by setting the `backup_staging_directory` configuration option in `address.yaml` (page 205).

10. Click **Restore Backup**.
    
    The **Confirm Restore** dialog appears.

    **Warning:** If a value was not set for throttling stream output in 8 (page 418), a warning message indicates the consequences of unthrottled restores. The throttle warning only appears for versions of DSE from 4.8.7 and later. Either click Cancel and set the throttle value in the **Restore from Backup** dialog, set the values in `cassandra.yaml` (`stream_throughput_outbound_megabits_per_sec` and `inter_dc_stream_throughput_outbound_megabits_per_sec`), or proceed anyway at risk of network bottlenecks.

    **Tip:** If you are using LCM to manage DSE cluster configuration, update Cluster Communication settings in `cassandra.yaml` in the **config profile** (page 594) for the cluster and run a **configuration job** (page 576). Stream throughput (not inter-dc) is already set to 200 in LCM defaults.
11. Click **Start Restore** to confirm the restore.

If the pre-restore checks detected schema differences that could not automatically be validated, the Restore Schema Validation dialog appears. Possible issues are listed and a comparison of the backup and current schema are presented side-by-side.
12. Review the information to determine what adjustments if any need to be made to the current schema:

- To rectify the schema issues and try the restore again afterward, click Cancel.
- To proceed despite the schema mismatch, click Continue Restore.

**Warning:** Attempting to restore a backup with an incompatible schema might result in corrupt or inaccessible data. Before forcing the restore, you might want to back up your current data.

The details and progress of the restore operation are displayed in a progress dialog, and also appear in the **Backup Activity** of the target cluster. If you close the progress dialog, track the progress and status of the restore in the target cluster's **Backup Activity** section.

The progress and details of the restore operation are displayed in the **Restore Report**.

![Restore Report](image)

**Backing up and restoring DSE Graphs in OpsCenter**

Follow these instructions to backup and restore DSE Graphs using the OpsCenter Backup Service. The steps you need to follow depend on:

- the versions of DSE and OpsCenter.
- the type of restore operation (to the same cluster or cloned data to a different cluster).
• whether or not a graph was dropped.

The order in which the steps are performed are outlined in the **DSE Graph guidelines (page 441)** table.

DataStax highly recommends upgrading to OpsCenter 6.5 to take advantage of improved and simplified backups and restores for DSE Graphs. For instance, a DSE Graph version 5.x can be restored in DSE 6.x if the 5.x backup was taken in OpsCenter version 6.5.x. The `graph_name_pvt` table has been removed for DSE versions 6.0 and later. When restoring a DSE 5.0.x or DSE 5.1.x graph, the `pvt` table is removed during the restore process by OpsCenter 6.5 and later. To restore a graph backup from DSE 5.0.x or 5.1.x to DSE 6.0, the backup must be taken in OpsCenter 6.5.

Fewer steps are required for restoring backups when using the latest version of OpsCenter (6.5 or later), as is evident from the guidelines below.

**Table 14: DSE Graph backup and restore guidelines**

<table>
<thead>
<tr>
<th>OpsCenter versions</th>
<th>DSE versions</th>
<th>Operations</th>
<th>Steps Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.5.x</td>
<td>5.0.x, 5.1.x, 6.0.x</td>
<td>Backup and restore to same cluster</td>
<td>1 (page 441), 2 (page 442)</td>
</tr>
<tr>
<td>6.5.x</td>
<td>5.0.x, 5.1.x, 6.0.x</td>
<td>Backup and restore to different cluster (clone (page 429))</td>
<td>1 (page 441), 2 (page 442), 7 (page 443)</td>
</tr>
<tr>
<td>6.1.x</td>
<td>5.0.x, 5.1.x</td>
<td>Same cluster, does not involve a manually dropped graph</td>
<td>1 (page 441), 2 (page 442), 6 (page 443)</td>
</tr>
<tr>
<td>6.1.x</td>
<td>5.0.x, 5.1.x</td>
<td>Same cluster, manually dropped graph (graph is recreated)</td>
<td>3 (page 442), 1 (page 441), 2 (page 442), 5 (page 442), 6 (page 443)</td>
</tr>
<tr>
<td>6.1.x</td>
<td>5.0.x, 5.1.x</td>
<td>Backup and restore to different cluster (clone)</td>
<td>4 (page 442), 1 (page 441), 2 (page 442), 5 (page 442), 6 (page 443), 7 (page 443)</td>
</tr>
</tbody>
</table>

1. In OpsCenter (version 6.5 highly recommended), **create a backup (page 391)** of the DSE Graph: Run an ad hoc backup, create a one-off scheduled backup, or set up a recurring scheduled backup.

**Note:** If a DSE Graph backup was created in an OpsCenter version prior to 6.5 (6.0.x or 6.1.x) for a DSE version earlier than 6.0 (5.0.x or 5.1.x), OpsCenter cannot restore that legacy backup into DSE 6.0 because the data needed is not present in the older backup schemas. In this case, simply upgrade to OpsCenter 6.5 and create a backup for the DSE 5.x graph in OpsCenter 6.5.
2. Restore the backup to the same cluster or a different (clone) cluster as appropriate. See Restoring a cluster *(page 414)* and Cloning cluster data *(page 429)*.

3. (If applicable step, see table guidelines OpsCenter 6.1.x and 6.0.x only) Manually create the *graph_name_pvt* keyspace using DataStax Studio:

```sql
CREATE KEYSPACE "<graph_name_pvt>" WITH replication = {
'class': 'NetworkTopologyStrategy', 'originalcluster': '1'} AND
durable_writes = true;
```

Replace *originalcluster* with the original datacenter name.

*Note:* The *graph_name* and *graph_name_system* keyspaces already exist; only the *graph_name_pvt* keyspace needs to be created.

4. (If applicable step, see table guidelines OpsCenter 6.1.x and 6.0.x clone restores only) Create the 3 graph keyspaces using DataStax Studio:

```sql
CREATE KEYSPACE "<graph_name>" WITH replication = {
'class': 'NetworkTopologyStrategy', 'restorecluster': '1'} AND
durable_writes = true;
CREATE KEYSPACE "<graph_name_system>" WITH replication = {
'class': 'NetworkTopologyStrategy', 'restorecluster': '1'} AND
durable_writes = true;
CREATE KEYSPACE "<graph_name_pvt>" WITH replication = {
'class': 'NetworkTopologyStrategy', 'restorecluster': '1'} AND
durable_writes = true;
```

Replace *<graph_name>* with the graph’s name and *restorecluster* with the new datacenter name. When cloning to a different cluster, the keyspaces must be manually created, because otherwise OpsCenter Restore Backup creates these keyspaces with the source cluster’s datacenter designations and the graph data will not be copied.

5. (If applicable step, see table guidelines OpsCenter 6.1.x and 6.0.x only) Insert a *dse_system.shared_data* record using DataStax Studio:

```sql
INSERT INTO dse_system.shared_data (dataspace, valid_until, namespace, Name, last_updated, Json, type, written_on) VALUES ('Cluster', 13814000-1dd2-11b2-0000-000000000000, 'system', 'DSE_GRAPH_QUICKSTART', now(),
```
Insert a new record in the `dse_system.shared_data` table of the destination cluster with values obtained from the source cluster:

```
insert into dse_system.shared_data
values 'json',
now() ;
```

6. (If applicable step, see table guidelines OpsCenter 6.1.x and 6.0.x only) If a graph has multiple search indexes against a single vertex label, or a search index and materialized views against a vertex label, you must manually recreate the graph indexes and materialized views. Manually recreate the indexes (search, materialized views) in the original source cluster using DataStax Studio. See Adding index schema.

7. (If applicable step, see table guidelines for cloned clusters) Ensure that the topology of the destination (target) cluster keyspaces is the same as the topology of source cluster keyspaces.

**Viewing backup and restore history**

OpsCenter tracks all in-process and completed backup and restore operations. View the status of the current and recent jobs, page through completed jobs, and view the detailed status of a particular backup or restore operation in the Activity tab. If OpsCenter was recently upgraded, synchronize the backup activity (page 444).

The details of all completed backup and restore operations are stored in the OpsCenter keyspace in the `backup_reports` table. The data is stored whether or not the operation was successful.

1. Click **cluster name** Services.

2. Click the **Details** link for the Backup Service.
   
   The Activity tab appears.

3. Click the **Previous** and **Next** links to peruse backup activity.
Synchronizing backup data after an upgrade

The first time the Backup Service starts, it scans for existing backups, including backups from versions of OpsCenter prior to 5.1 (the release when the Backup Service made its debut), and populates the `backup_reports` table. If there is no backup activity, or OpsCenter was recently upgraded, the Activity page indicates there is not any backup activity available at this time.

- If the OpsCenter instance has no prior version, create a backup *(page 391)* to get started.
- If OpsCenter was just upgraded, click the link to synchronize activity.
1. Click `cluster name`Services.

2. Click the Details link for the Backup Service. The Activity tab appears.

3. To manually synchronize the `backup_reports` table, click the inline sync activity link or Actions#Synchronize Data. The Synchronize Backup History Data dialog appears.

4. Select the locations whose history you want to synchronize and click Sync Backup Data.

**Deleting backup data**

Delete backup data that your organization no longer needs to retain. The Backup Service retains the backup history.

1. Click `cluster name`Services.
2. Click the **Details** link for the Backup Service.
   The **Activity** tab appears.

3. In the Activity tab, click the backup for which you want to delete data.
   The Backup Report dialog appears.

4. Click **Delete Backup Data**.
   A dialog prompts you to confirm permanently deleting the data for the backup.

5. Click **Delete Backup Data**.
   The Status column in the Activity tab indicates Deletion Complete for the backup.

### Backup Service configuration options reference

The Backup Service configuration options reference lists the available configuration options in alphabetical order for each configuration file. See [Configuration files for OpsCenter](page 180) more information about configuration files and their precedence behavior.

OpsCenter daemon configuration options for the Backup Service

The following options are available in opscenterd.conf:

**[backup_service] auto_sync_retry_attempts**

The number of times to auto retry sending a backup to a destination if there are failures. Default: 3.
[backup_service] auto_sync_retry_delay
   The amount of time in minutes to wait between retry attempts. Default: 60.

Cluster configuration options for the Backup Service

The following options are available in cluster_name.conf:

[agents] backup_staging_dir
   This path specifies the directory where commit logs archived to by cassandra.
   From there the commit logs are sent to the configured destinations and the
   backup_storage_dir.

[agents] concurrent_snapshot_list_requests
   The number of concurrent get snapshot info requests. The default value is 1.

[agents] remote_backup_region
   The S3 region region to connect to for remote backup/restore. The default value is
   us-west-1.

[agents] restore_req_update_period
   The frequency (in seconds) that a restore reports progress back to OpsCenter.
   The default value is automatically calculated based on cluster size. To optimize
   performance, larger clusters have a longer threshold in which restore progress is
   sent to the UI. Configure this property accordingly for more or less frequent updates
   during a restore. Note: Lower numbers might impact performance during a restore.

[agents] snapshot_wait
   See [cassandra] snapshot_threshold

[backups] failure_threshold
   The percentage of the cluster that can fail to respond before a remote destination
   restore action fails. Default: 50.

[backups] restore_init_throttle
   The number of DataStax Agents on which OpsCenter will concurrently start the
   restore process. The default value is 20.

[backups] restore_sleep
   How long OpsCenter will sleep between batches of starting the restore process, set
   in restore_init_throttle. The default value is 5 seconds.

[cassandra] snapshot_threshold
   The number of nodes in the cluster before OpsCenter will switch from running a
   backup immediately upon receiving a request to scheduling the backup to run after
   the next full minute plus any time set in snapshot_wait. The default value is 10. If
   there are less than 10 nodes in the cluster then OpsCenter will tell all nodes to take
   a snapshot as soon as it is requested. If there are more than 10 nodes, OpsCenter
   will tell all nodes to take a snapshot at the current time rounded to the next minute,
   plus snapshot_wait seconds.

[destinations] active
   Specifies the names of destinations to back up to. The destination names should
   not have any spaces and should be comma-delimited.

Agent configuration options for the Backup Service

The following agent configuration options are available in address.yaml:

backup_staging_dir
**backup_staging_dir**
The directory used for staging commit logs to be backed up. The default location is 
/var/lib/datastax-agent/commitlogs/. Example: 
`backup_staging_dir: /var/lib/datastax-agent/commitlogs/`

**backup_storage_dir**
The directory used for storing backup files. The default location is /var/lib/datastax-agent/backups/. Example: `backup_storage_dir: /var/lib/datastax-agent/backups/`

**destination_pretest_timeout**
The maximum amount of time in seconds to verify a destination can be written to and read from. Default: 60. Example: `destination_pretest_timeout: 60`

**max_file_transfer_attempts**
The maximum number of attempts to upload a file or create a remote destination. Default: 3. Example: `max_file_transfer_attempts: 30`

**remote_backup_region**
The AWS region to use for remote backup transfers. Default: us-west-1. Example: `remote_backup_region: us-west-1`

**remote_backup_retries**
The number of attempts to make when file download fails during a restore. Default: 3. Example: `remote_backup_retries: 3`

**remote_backup_timeout**
The timeout in milliseconds for the connection used to push backups to remote destinations. Default: 1000. Example: `remote_backup_timeout: 1000`

**remote_verify_initial_delay**
Initial delay in milliseconds to wait before checking if a file was successfully uploaded during a backup operation. This configuration option works in conjunction with the `remote_verify_max` option to distinguish between broken versus tardy backups when cleaning up SSTables. The `remote_verify_initial_delay` value doubles each time a file transfer validation failure occurs until the value exceeds the `remote_verify_max` value. Default: 1000 (1 second). Example: `remote_verify_initial_delay: 1000`

**remote_verify_max**
The maximum time period to wait after a file upload completed but is still unreadable from the remote destination. When this delay is exceeded, the transfer is considered failed. This configuration option works in conjunction with the `remote_verify_initial_delay` option to distinguish between broken versus tardy backups when cleaning up SSTables. Default: 30000 (30 seconds). Example: `remote_verify_max: 30000`

**restore_on_transfer_failure**
When set to true, a failed file transfer from the remote destination will not halt the restore process. A future restore attempt uses any successfully transferred files. Default: false. Example: `restore_on_transfer_failure: false`

**restore_req_update_period**
The frequency in seconds with which status updates are sent to opscenterd during Restore operations in the Backup Service. Default: 60. Example: `restore_req_update_period: 60`

**sstableloader_max_heap_size**
The maximum heap size used by the sstableloader during restore operations. Only supported with DSE 4.8.4+. Default: 256M. Example:
```
sstableloader_max_heap_size: 256M
```

**tmp_dir**
The location of the temp directory used as the staging directory for Backup Service backups and for Repair Service persistence files. The default location is /var/lib/datastax-agent/tmp. Example:
```
tmp_dir: /var/lib/datastax-agent/tmp/
```

**address.yaml**
The location of the `address.yaml` file depends on the type of installation:
- Package installations: /var/lib/datastax-agent/conf/address.yaml
- Tarball installations: `install_location/conf/address.yaml`

**opscenterd.conf**
The location of the `opscenterd.conf` file depends on the type of installation:
- Package installations: /etc/opscenter/opscenterd.conf
- Tarball installations: `install_location/conf/opscenterd.conf`

**cluster_name.conf**
The location of the `cluster_name.conf` file depends on the type of installation:
- Package installations: /etc/opscenter/clusters/cluster_name.conf
- Tarball installations: `install_location/conf/clusters/cluster_name.conf`

### Backup and Restore fields reference

Backup and Restore fields descriptions

Reference of fields that are UI controls in the Backup Service. The fields can appear in both Backup and Restore dialogs. The fields that appear in a given dialog depend on the type of task you are performing with the Backup Service, and other factors such as cluster topology (datacenters, for instance).
## Backup fields

<table>
<thead>
<tr>
<th>Create (or Edit) Scheduled Backup dialog</th>
<th>Create (or Edit) Backup Run Now dialog</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Create Backup</strong></td>
<td><strong>Create Backup</strong></td>
</tr>
<tr>
<td><strong>Type:</strong></td>
<td><strong>Type:</strong></td>
</tr>
<tr>
<td>Run Now</td>
<td>Run Now</td>
</tr>
<tr>
<td>Schedule</td>
<td>Schedule</td>
</tr>
<tr>
<td>1/12/2017</td>
<td>1/12/2017</td>
</tr>
<tr>
<td>11:00 PM</td>
<td>11:00 PM</td>
</tr>
<tr>
<td>US Central</td>
<td>US Central</td>
</tr>
<tr>
<td>Repeat every 1 Days</td>
<td>Repeat every 1 Days</td>
</tr>
<tr>
<td>Keyspaces: *</td>
<td>Keyspaces: *</td>
</tr>
<tr>
<td>All Keyspaces</td>
<td>Click to choose one or more keyspaces</td>
</tr>
<tr>
<td>Datacenters: *</td>
<td>Datacenters: *</td>
</tr>
<tr>
<td>Analytic</td>
<td>Datacenters: *</td>
</tr>
<tr>
<td><strong>Alert on Failure:</strong></td>
<td><strong>Alert on Failure:</strong></td>
</tr>
<tr>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>Current Data Size: 5 MB</td>
<td>Current Data Size: 5 MB</td>
</tr>
<tr>
<td>Location:</td>
<td>Location:</td>
</tr>
<tr>
<td>LOCATION</td>
<td>LOCATION</td>
</tr>
<tr>
<td>RETENTION</td>
<td>RETENTION</td>
</tr>
<tr>
<td>On Server</td>
<td>On Server</td>
</tr>
<tr>
<td>Retain all</td>
<td>Retain all</td>
</tr>
<tr>
<td>Amazon S3 - mybucket</td>
<td>Amazon S3 - mybucket</td>
</tr>
<tr>
<td>Retain all</td>
<td>Retain all</td>
</tr>
<tr>
<td>Local FS - /mylocalbackupdir</td>
<td>Local FS - /mylocalbackupdir</td>
</tr>
<tr>
<td>Retain all</td>
<td>Retain all</td>
</tr>
<tr>
<td>+ Add Location</td>
<td>+ Add Location</td>
</tr>
<tr>
<td>Staging Directory:</td>
<td>Staging Directory:</td>
</tr>
<tr>
<td>/var/lib/dtestax-agent/tmp/</td>
<td>/var/lib/dtestax-agent/tmp/</td>
</tr>
<tr>
<td>This directory will be used to store...</td>
<td>This directory will be used to store...</td>
</tr>
<tr>
<td>Advanced Options</td>
<td>Advanced Options</td>
</tr>
<tr>
<td>Create Backup</td>
<td>Create Backup</td>
</tr>
<tr>
<td>Cancel</td>
<td>Cancel</td>
</tr>
</tbody>
</table>
### Table 15: Create Backup fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type</strong></td>
<td>Type of backup job to create: <strong>Run now</strong> or <strong>Scheduled</strong>.</td>
</tr>
</tbody>
</table>
| **Schedule** | Schedule backup settings by date and time. Only appears when **Schedule** is selected as the Type. Available settings:  
  - Date: Day, Month, and Year.  
  - Time and Timezone: GMT is the default timezone. To change the timezone, click **GMT**, select the country and timezone, and click **Save**.  
  - Repeat or Don't Repeat settings: Use to schedule a backup job on a regular periodic basis, or run a scheduled backup during off hours. |
| **Keyspaces** | Required. Allows selecting one or more or all keyspaces for a backup.  
When backing up a DSE Graph database, select your graph keyspace. When a keyspace is created in DSE Graph, it is actually comprised of three keyspaces that OpsCenter bundles together because all three keyspaces are required for a DSE Graph backup.  
**Tip:** Click the plus (+) icon to select one or more options. To keep the list open when making multiple selections, press and hold the Cmd key (Mac) or Ctrl key (Windows/Linux). |
| **Datacenters** | Required. The Datacenters field only appears if there are multiple datacenters configured for a cluster. Allows selecting one or more or all datacenters for a backup. Running a backup by a specific datacenter improves backup performance under most circumstances, because the process eliminates backing up data redundantly since the data is replicated to other datacenters anyway.  
**Warning:** When limiting a backup by datacenter, if the SimpleStrategy replication strategy is used, or the NetworkTopologyStrategy is used with a replication factor of 0 for a datacenter, the snapshot might not contain all the data for the cluster.  
**Tip:** Click the plus (+) icon to select one or more options. To keep the list open when making multiple selections, press and hold the Cmd key (Mac) or Ctrl key (Windows/Linux). |
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Alert on Failure**       | Activates an alert upon backup failure that sends an alert to the event log. Default: False.  
**Tip:** Enabling Alert on Failure is particularly useful for monitoring scheduled overnight backups. Configure alerts for events (page 138) and route the alerts through email, or post URLs (page 145) to a chat room such as HipChat or Slack. |
| **Current Data Size**      | Current data size is a best estimate of how large the snapshot will be based on the currently selected keyspace or keyspaces. The information is based on table metrics collected by OpsCenter. Current Data Size is a dynamic read-only field that displays **Unknown** when size information is not available for the selected keyspaces. An Unknown Data Size typically displays when either a system keyspace is selected or keyspaces size information has not been collected yet. |
| **Location**               | Snapshots are saved to the node’s snapshot directory for the table being saved. For example, /var/lib/cassandra/data/OpsCenter/settings/snapshots. Add more locations to back up to and restore from, including Local file system and Amazon S3 locations, by clicking Add Location. Set a retention policy for the backup data in each location of a backup job. See Adding a backup location (page 406). |
| **Staging Directory**      | The location of the Backup Service staging directory for backups. The default location is /var/lib/datastax-agent/tmp. To change the default location of the directory, indicate the path using the tmp_dir (page 211) configuration option in address.yaml. |
| **Advanced Options**       | Expands the Pre-Backup Script and Post-Backup Script fields for running scripts before or after (or both) snapshots are taken. The script must be located in the appropriate install directory on each node.  
Custom backup scripts must be located in:  
- **Package installs:** /usr/share/datastax-agent/bin/backup-scripts  
- **Tarball installs:** install location/agent/bin/backup-scripts  
For details, see Configuring custom scripts to run before and after backups (page 390). |
| **Create Backup**          | Creates a scheduled or runs an ad hoc backup job immediately depending on the selected backup Type (Scheduled or Run Now). |
Sync Snapshot to Location dialog

The **Sync Snapshot to Location** dialog is accessible by clicking **Sync to Location** from the Backup Report dialog.

![Sync Snapshot to Location dialog](image)

**Table 16: Sync Snapshot to Location fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Displays the locations configured by clicking the <strong>+Add Location</strong> link.</td>
</tr>
<tr>
<td></td>
<td>Activates an alert if the sync snapshot operation fails.</td>
</tr>
<tr>
<td>Sync Snapshot</td>
<td><strong>Syncs a snapshot</strong> <em>(page 404)</em> to the designated location.</td>
</tr>
</tbody>
</table>

Add Location fields

The **Add Location** dialog is accessible by clicking Add Location from Create Backup, Commit log backup, Restore backup, or Sync Snapshot to Location dialogs.
### Table 17: Add Location dialog fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Location</strong></td>
<td>Specifies the location to backup data to and restore data from. Available options:</td>
</tr>
<tr>
<td></td>
<td>• Amazon S3</td>
</tr>
<tr>
<td></td>
<td>• Local FS</td>
</tr>
<tr>
<td></td>
<td>Required.</td>
</tr>
<tr>
<td><strong>Path</strong></td>
<td>Only applicable to the Local FS location. The path to a local file system location such as an NFS mount. The path must be unique and can include subdirectories. Required.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>S3 bucket</strong></td>
<td>Only applicable to an Amazon S3 location. The Amazon S3 bucket name. The bucket name must be unique.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The bucket name must be at least 4 characters long. Bucket names should only contain lowercase letters, numbers, and hyphens. See the S3 guidelines for more details about bucket naming restrictions.</td>
</tr>
<tr>
<td></td>
<td><strong>Tip:</strong> To indicate a bucket subfolder location, delineate the bucket name from the folder name with a forward slash (/) character. Example: <code>mybucket/myfolder/mysubfolder</code>. Remember that slashes are not allowed within bucket or folder names themselves.</td>
</tr>
<tr>
<td></td>
<td>Required.</td>
</tr>
<tr>
<td><strong>AWS Key</strong></td>
<td>Only applicable to an Amazon S3 location. Your AWS Key to access your resources on AWS. Required.</td>
</tr>
<tr>
<td><strong>AWS Secret</strong></td>
<td>Only applicable to an Amazon S3 location. Your AWS Secret to access your resources on AWS. Required.</td>
</tr>
<tr>
<td><strong>Throttle S3 transfer rate</strong></td>
<td>Only applicable to an Amazon S3 location. Sets a maximum upload rate at the specified MB/s to avoid saturating your network. Optional.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> When the AWS CLI (page 24) labs feature is enabled, the S3 throttle is ignored at this time. A tooltip also mentions this current limitation. See Tuning throttling when using AWS CLI (page 25).</td>
</tr>
<tr>
<td><strong>Throttle transfer rate</strong></td>
<td>Only applicable to a Local FS location. Sets a maximum upload rate at the specified MB/s to avoid saturating your network. Optional.</td>
</tr>
<tr>
<td><strong>Enable compression</strong></td>
<td>Decompresses a backup that was created with compression enabled. Compression reduces the amount of data going through the network and reduces the disk and data usage but increases the CPU load for the server. Optional.</td>
</tr>
<tr>
<td><strong>Enable S3 server-side encryption</strong></td>
<td>Only applicable to an Amazon S3 location. Enables server-side S3 encryption (SSE-S3). Server-side encryption encrypts each file in the backup set with a unique key, as well as the key itself, using a 256-bit AES cypher. The key is encrypted with a master key that is regularly rotated. Enabling server-side encryption increases the security of your backup files, but increases the time it takes to complete a backup. For more information on S3 server-side encryption, see <a href="https://docs.aws.amazon.com/AmazonS3/latest/dev/UsingServerSideEncryption.html">https://docs.aws.amazon.com/AmazonS3/latest/dev/UsingServerSideEncryption.html</a>. Optional.</td>
</tr>
</tbody>
</table>
RetentionPolicy (page 381) is only available for scheduled backups. Available options:

- **Retain all**: Retains all backups. Default for S3 or Local FS.
- **Retain for**: Retains backups for the specified period of time and then deletes the backups. Available options:
  - # Minutes
  - # Hours
  - # Days (On Server default 30 days)
  - # Weeks

Optional.

Save Location

Saves the location for backup and restores.

Configure commit log backup dialog for point in time restore

![Configure commit log backup dialog](image)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retention Policy</td>
<td><strong>Retain Policy (page 381)</strong> is only available for scheduled backups. Available options:</td>
</tr>
<tr>
<td></td>
<td>- <strong>Retain all</strong>: Retains all backups. Default for S3 or Local FS.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Retain for</strong>: Retains backups for the specified period of time and then deletes the backups. Available options:</td>
</tr>
<tr>
<td></td>
<td>- # Minutes</td>
</tr>
<tr>
<td></td>
<td>- # Hours</td>
</tr>
<tr>
<td></td>
<td>- # Days (On Server default 30 days)</td>
</tr>
<tr>
<td></td>
<td>- # Weeks</td>
</tr>
<tr>
<td></td>
<td>Optional.</td>
</tr>
</tbody>
</table>

Save Location

Saves the location for backup and restores.

Table 18: Configure commit log backup dialog fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>On</td>
<td>Toggles commit log backups <strong>On</strong> or <strong>Off</strong>. Configure commit log backups (page 384) to facilitate point in time restores (page 421).</td>
</tr>
<tr>
<td>Commit log locations</td>
<td>Configured locations for commit log backups. Clicking <strong>+Add Location</strong> opens the Add Location (page 453) dialog to configure additional locations (page 406) beyond On Server, such as Amazon S3 or Local FS.</td>
</tr>
</tbody>
</table>
The path to the location where commit logs are stored on each node. The directory must be writable both by the user running DataStax Enterprise and the agent user, which are by default the same user.

**Note:** The **Backup Directory** and the **tmp_dir (page 211)** must be located on the same filesystem so that a hard link can be established between them.

**Restore from backup dialogs fields**

The Restore process steps through two or more dialogs depending on the selections and type of restore or clone. Depending on the selected backup location, the dialog title is Step 2 or 3 of 3. Selecting Other Location is a 3-step process. Follow the task procedures for step-by-step instructions.

**Table 19: Restore fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Backups</strong></td>
<td>List of available backups by date, time, and location to select for a restore.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>PIT Restore</strong> tab</td>
<td><strong>Point in Time</strong> Sets the date and time to restore data.</td>
</tr>
<tr>
<td></td>
<td><strong>Commit logs Location</strong> Selects the location of the commit logs; either On Server, Local FS, or another location on Amazon S3. The location of commit logs is configured when enabling commit log backups <em>(page 384)</em>.</td>
</tr>
<tr>
<td></td>
<td><strong>Backup Location</strong> Selects the location of the snapshot. Available locations: On Server, Local FS path, Amazon S3 bucket. The +Add location link is available to add more locations <em>(page 406)</em>.</td>
</tr>
<tr>
<td><strong>Other Location</strong> tab</td>
<td>The Other Location tab allows specifying the same fields as available in the Add Location <em>(page 453)</em> dialogs.</td>
</tr>
</tbody>
</table>
### Configure and Restore dialogs

Depending on the selected backup location, the dialog title is Step 2 or 3 of 3. Selecting Other Location is a 3-step process.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keyspaces</td>
<td>Required. Allows selecting one or more or all keyspaces for a restore.</td>
</tr>
<tr>
<td></td>
<td>When restoring a DSE Graph database, select your graph keyspace. When a keyspace is created in DSE Graph, it is actually comprised of three keyspaces that OpsCenter bundles together because all three keyspaces are required for a DSE Graph backup.</td>
</tr>
<tr>
<td></td>
<td><strong>Tip:</strong> Click the plus (+) icon to select one or more options. To keep the list open when making multiple selections, press and hold the Cmd key (Mac) or Ctrl key (Windows/Linux).</td>
</tr>
<tr>
<td>(Cluster) Location</td>
<td>The location of the cluster to which to restore or clone data. Only clusters managed by a singular OpsCenter instance appear in this list.</td>
</tr>
<tr>
<td>Truncate/delete existing data before restore</td>
<td>Removes the existing keyspace data before the data is restored or cloned.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Throttle stream throughput | Sets a maximum transfer rate at the specified MB per second for the restore. Throttling prevents overloading the network. Recommended. When a throttle value is not set, the following warning displays in the Confirm Restore dialog for versions of DSE from 4.8.7 and later:  
An outgoing stream throttle value was not specified for this restore. If cassandra.yaml does not have values set for stream_throughput_outbound_megabits_per_sec and inter_dc_stream_throughput_outbound_megabits_per_sec, the restore will not be throttled. Previous versions of Apache Cassandra would limit these streams to no more than 200 MB/sec. Unthrottled restores may consume large amounts of network resources. Plan accordingly.  
Throttle options display in the Restore Report.                                                                 |
| Staging Directory      | The location of the Backup Service staging directory for backups and restores. The default location is /var/lib/datastax-agent/tmp. To change the default location of the directory, indicate the path using the tmp_dir (page 211) configuration option in address.yaml. |
| Restore Backup          | Begins a restore backup job immediately.                                                                                                                                                    |
| Start Restore           | Confirms the restore operation from the Confirm Restore dialog.                                                                                                                                 |

**Troubleshooting Backup Service errors**

If you encounter errors when backing up or restoring using the Backup Service, follow these instructions to make sure your environment is configured correctly.

**Amazon S3 errors**

If you are using an Amazon S3 location for storing backups or commit logs, you might encounter errors if the permissions or authentication keys have been changed since the backup job was created. Input updated authentication tokens with permissions to write to the specified bucket name.

**Important:** The Backup Service requires control over the data and structure of its destination locations. The AWS S3 bucket and the Local file system destinations must be dedicated for use only by OpsCenter. Any additional directories or files in those
destinations can prevent the Backup Service from properly conducting a Backup or Restore operation.

Insufficient privileges as an AWS IAM (Identity Access Management) user causes an authentication error that displays the message "Unable to authenticate against AWS with the provided key and secret." Ensure the user has base privileges as well as the privilege for the ListAllMyBuckets action:

```
"Action": "s3:ListAllMyBuckets"
```

Actions to consider for granting AWS IAM user privileges:

- CreateBucket
- GetBucketLocation
- ListBucket
- ListAllMyBuckets
- PutObject
- GetObject
- DeleteObject

For more information, refer to Amazon AWS IAM documentation.

Agent errors

All the nodes in your cluster must use Java 7+ for the Backup Service to work.

The agent and DataStax Enterprise user either must be the same (the default, starting in OpsCenter 5.1), or the agent user must have the correct permissions (page 67) to read and modify the files owned by the DataStax Enterprise user.

Repair Service

The Repair Service performs repair operations across a DataStax Enterprise cluster in a minimally impactful manner. The Repair Service runs in the background, repairing small chunks of a cluster to alleviate the pressure and potential performance impact of having to periodically run repair on entire nodes.

The Repair Service cyclically repairs a DataStax Enterprise (DSE) cluster within the specified time to completion. Any anticipated overshoot (page 471) of the targeted completion time is communicated with a revised estimate.

**Note:** DataStax Enterprise 4.8 or later is required for running the OpsCenter Repair Service 6.1 or later.

Repair Service overview

The Repair Service runs repair operations, which synchronize the most current data across nodes and their replicas, including repairing any corrupted data encountered at the filesystem level. By default, the Repair Service runs subrange repairs for most tables, and
can be configured to run incremental repairs on certain tables. Distributed subrange repair is an alternative implementation of subrange repairs within the Repair Service, intended to better scale for large clusters.

To determine which type of repair to use, and when to use it, follow these guidelines:

- If data is relatively static, configure incremental repair for those tables or datacenters.
- If data is dynamic and constantly changing, use subrange repairs, excluding keyspaces and tables as appropriate for an environment.
- If repairing a very large cluster, and the opscenterd process becomes a bottleneck for timely subrange repairs, use distributed subrange repairs.

**Warning:** Do not run a manual node repair using `nodetool repair` or OpsCenter while the Repair Service is running.

Subrange repairs

Subrange repairs *(page 464)* repair a portion of the data that a node is responsible for. Subrange repairs are analogous to specifying the `-st` and `-et` options on the `nodetool repair` command, only the Repair Service determines and optimizes the start and end tokens of a subrange for you. The main benefit of subrange repair is more precise targeting of repairs while avoiding overstreaming.

Distributed subrange repairs

Distributed subrange repairs *(page 466)* are designed for repairing large clusters. Rather than rely on opscenterd to coordinate subrange repairs, opscenterd instead instructs agents to simply repair a list of one or more entire keyspaces. The agent handles the details of repairing the keyspaces on a per-subrange basis so that opscenterd coordinates a much smaller list of more coarsely-grained repair tasks, the details of which are handled by each agent.

Incremental repairs

Incremental repairs *(page 468)* only repair data that has not been previously repaired on tables reserved and configured for incremental repair.

Subrange repairs operate on an exclusion (opt out) basis that can exclude certain keyspaces and tables. Ignored tables for subrange repairs consist of those reserved by OpsCenter and those configured by admins. Incremental repairs operate on an inclusion (opt in) basis. Only those keyspaces and tables designated for incremental repairs are processed during an incremental repair. Tables flagged for incremental repair include those built-in by OpsCenter and those configured by admins.

If data is relatively static, configure incremental repair for those tables or datacenters. If data is dynamic and constantly changing, use subrange repairs, excluding keyspaces and tables as appropriate for an environment.

There is no crossover between subrange and incremental repairs: keyspaces and tables are either repaired by a subrange or an incremental repair. Subrange and incremental repairs are mutually exclusive at a table level. The Repair Service runs both repair types
simultaneously. Each repair type has its own timeline, which is tracked in their respective individual subrange and incremental progress bars in the Repair Status (page 474) summary.

Parallel vs. sequential validation compaction processing

The Repair Service runs validation compaction in parallel by default rather than sequentially because sequential processing take considerably more time. The snapshot_override setting controls whether validation compactions for both subrange and incremental repairs are processed in parallel or sequentially. See Running validation compaction sequentially (page 490).

Conditions under which the Repair Service does not run

A cluster with a single node is not eligible for repairs. Repairs make node replicas consistent; therefore, there must be at least two nodes to exchange Merkle trees during the repair process.

Subrange repairs overview

Subrange repairs repair a portion of the data that a node is responsible for. After an entire cluster has been repaired, including any independently running incremental repairs, the Repair Service recalculates the list of subranges to repair and starts over. Repairing an entire cluster one time is referred to as a repair cycle.

Excluding keyspaces and tables

Specify entire keyspaces or certain tables for subrange repairs to ignore, in addition to the default system keyspaces, certain rollup tables, and so forth. See Excluding keyspaces or tables from subrange repairs (page 483).

View excluded tables (page 478) in the repair status details.

Prioritizing tasks

When running subrange repairs, the Repair Service determines which nodes have the least traffic in terms of compactions or streaming between nodes. Both streaming and compaction activities are represented in the Repair Service Status. The prioritization_page_size option limits the number of possible repair tasks for the Repair Service to review when selecting a low impact repair. Increasing the page size is more CPU-intensive for the Repair Service but could result in more optimal dispatching of repairs to the cluster.

Warning: The prioritization_page_size is an expert option that should not be changed without guidance from DataStax Support.

Offline splits

Offline splits refers to offline task generation (determining splits for subrange repairs) by the Repair Service when a node is down or unavailable.
Ideally, during planning of a subrange repair, the Repair Service in the OpsCenter daemon retrieves the token subrange splits from each OpsCenter agent in the cluster, since each agent is able to retrieve the necessary data from its node to determine the optimal set of subrange splits for each keyspace to repair. However, if either the agent or node is offline or unavailable, the Repair Service falls back to splitting the token range for that node. This is less than optimal because the OpsCenter daemon cannot access the information about counts and sizes of partitions that belong to a token range for an unavailable node.

The `offline_splits` option controls the number of subranges per keyspace to split the primary range into for a node. The goal for each subrange is to have no more than approximately 32,000 partitions per keyspace. It is most optimal to repair a subrange that contains 32,000 partitions because that is the largest number of partitions in a range that can be repaired in a single attempt without streaming more data than necessary between nodes.

The default for the `offline_splits` option is 256. For sparsely populated clusters, the default might suffice. For clusters having much more densely populated nodes, it could make sense to increase the default value. The `system.size_estimates` table is regenerated every five minutes, and gives some indication of how many partitions are contained within each node’s primary range for each keyspace and table.

Configuration options for offline splits and its related options are considered expert-level options (page 493) that should not be adjusted without guidance from DataStax Support.

The Repair Service log (page 486) indicates if offline splits had to be used for any node.

Throttling subrange repair time

The Repair Service automatically throttles subrange repairs when the current repair cycle is estimated to finish significantly before the deadline specified by the time to completion.

The `time_to_completion_target_percentage` configuration option controls the frequency and pace of the subrange repair process. The throttle slows repairs or reduces parallel repair processes as necessary to prevent overloading the cluster while still completing a repair cycle within the specified time window designated by the `Time to completion` value. The default value for the target percentage to complete a repair is 65%.

Because certain repair config options are tempered by the percentage option, a judicious approach to configuring advanced repair options can optimize repair performance for various production environments and avoid issues due to misconfiguration. The majority of default settings typically do not require adjustment unless advised by a DataStax Support professional.

If there are any issues with the Repair Service configuration, the Repair service not configured correctly rule in the Best Practice Service (page 502) fails and provides guidance as to incorrectly configured options, unless the rule has been turned off.

Caution: DataStax recommends only manually adjusting the `max_parallel_repairs`, changing `min_repair_time` and other advanced (page 490) or expert (page 494) options only if the `time_to_completion_percentage`
throttle is not in use. See Adjusting or disabling the throttle for subrange repairs (page 488).

Note: Incremental repairs are exempt from this throttle.

See Adjusting or disabling the throttle for subrange repairs (page 488) for more information.

Calculating parallel repairs

The Repair Service uses an average of the throughput of recent repairs to calculate the average throughput. The average throughput is used to dynamically determine the number of parallel repairs required to complete a repair during a current cycle. The num_recent_throughputs option determines the maximum number of recent throughputs used to calculate average throughput. The default value is 500. Calculating parallel repairs also depends on a corresponding minimum throughput value before commencing its calculation. The min_throughput option represents the throughput required for any given repair task to be considered when determining the number of parallel repairs. The default value is 512 bytes/sec.

Maximum pending repairs

Before issuing a new subrange repair, the Repair Service checks for the number of repairs both running or waiting to run. If the configured maximum pending repairs (page 189) threshold would be exceeded, the repair skips that node for the time being to avoid overwhelming an already swamped node. The repair task is moved to the back of the pending repair tasks queue to retry later and an alert is fired.

The Attempts column in the table details pane of the Repair Service Status page displays the number of repair attempts.

Subrange repair status

View progress, statistics, and details for subrange repairs in the Repair Status (page 474) tab.

Distributed subrange overview

Distributed subrange repair is an alternative implementation of subrange repairs within the OpsCenter Repair Service, intended to scale for large clusters.

Note: The distributed subrange repair feature is available for OpsCenter version 6.1.7 only.

With subrange repairs, metadata about cluster token ranges must be retrieved and processed, which requires splitting the ranges into appropriate subranges prior to starting repair operations. The amount of metadata to process is proportional to the size of the cluster. Therefore, for large clusters, the OpsCenter daemon (opscenterd) process can become a bottleneck when attempting to processing sizeable metadata.
With distributed subrange repairs, much of the metadata processing and repair planning is moved to the DataStax agents, relieving opscenterd and allowing the repair service to scale better for large clusters. In this method of repair, opscenterd instructs OpsCenter agents to repair a list of one or more entire keyspaces, and the agents repair the keyspaces on a per-subrange basis.

You must opt into using distributed subrange repairs by setting `enable_distributed_subrange_repair` to `True` in the `opscenterd.conf` configuration file. See Enabling distributed subrange repairs (page 484).

**Important:** After changing the value for `enable_distributed_subrange_repair`, restart OpsCenter (page 75) for the changes to take effect.

Differences from subrange repairs

Distributed subrange repair is designed to run as fast as possible to expedite repair of large clusters. Because distributed subrange repair is optimized for performance, it cannot be tuned in the way that subrange repairs can. For example, the `time_to_completion_target_percentage` parameter has no effect on distributed subrange repairs.

However, distributed subrange repair will honor the `min_repair_time` property to provide a limited amount of throttling. Each DataStax agent ensures that the individual JMX repair operations will not occur more frequently than the time set by this property.

Limitations

Because of the distributed and dynamic nature of distributed subrange repair, the OpsCenter Repair Service is unable to provide a precise estimated time to completion for a running distributed subrange repair job. This limitation exists because distributed subrange repair does not process all token subrange metadata at the beginning of the job (like in subrange repair). Instead, each DataStax agent processes its own subset of that metadata dynamically, as necessary.

The benefit of this processing method is reducing latency and eliminating error handling changes that can stem from processing all metadata at the beginning of the job. The drawback is the inability to calculate estimates for data and time remaining in the repair process.

In OpsCenter, the distributed repair progress bar indicates a rough measurement of progress for data synchronization to complete, but without the fine-grained measure of progress (time and bytes remaining) available in subrange repairs.
opscenterd.conf

The location of the `opscenterd.conf` file depends on the type of installation:

- **Package installations:** `/etc/opscenter/opscenterd.conf`
- **Tarball installations:** `install_location/conf/opscenterd.conf`

**Incremental repairs overview**

Incremental repairs

Incremental repairs *(page 468)* only repair data that has not been previously repaired on tables reserved and configured for incremental repair.

After incremental repairs have completed for an entire cluster, the Repair Service **sleeps** *(page 469)* for an appointed time. When the **incremental threshold** *(page 469)* of unrepaired data is reached (DSE 5.1 and later), it triggers an incremental repair only on designated tables that meet the criteria. Repairing an entire cluster one time is referred to as a **repair cycle**.

Incremental repairs run in a singular sequential manner and do not run in parallel. The Repair Service coordinates incremental repairs with subrange repairs. If the **max_parallel_repairs** option is set to 1 *(page 493)*, subrange repairs and incremental repairs alternate running tasks one-at-a-time, waiting for a subrange repair to complete before starting an incremental repair and vice versa. Doing so can be helpful for isolating repair issues.

**Restricting by datacenter and racks**

Specify the datacenters and racks by which to restrict incremental repairs using the `incremental_repair_datacenters` option. Restricting repairs by datacenter or racks
improves repair performance in a multi-DC cluster with replicated keyspaces in both datacenters. Repairs complete faster with fewer repair tasks to process.

Including tables

Specify the specific tables to include for incremental repairs in the incremental_repair_tables option. The OpsCenter.settings and OpsCenter.backup_reports tables are included by default.

Threshold of unrepaired data

The Repair Service only repairs a table designated as candidate for incremental repair if the amount of unrepaired data is above a certain threshold, which is 1 KB by default. Configure the threshold with the incremental_threshold option.

**Note:** The incremental repair threshold option is only applicable for DSE versions 5.1 and later.

When the DSE version is at least 5.1 or later, the Repair Service takes an extra step of excluding any tables from the incremental_repair_tables option that do not meet the threshold criteria. When an incremental repair ends, the Repair Service checks every table in the incremental repair tables list against the threshold before starting the next repair on tables that qualify for repair. The threshold option allows for more selective incremental repairs.

If the DSE version is earlier than 5.1, the Repair Service always repairs every table configured for incremental repairs.

Ignore incremental errors threshold

The threshold for ignoring errors before alerting is set to a default of 20. Configure the threshold with the incremental_err_alert_threshold to adjust the tolerated level of incremental repair error alerts for your environment.

Sleep between incremental repair cycles

After completing all incremental repairs, the Repair Service suspends incremental repairs for a fixed interval (one hour by default) until starting again. The sleep time can be configured with the incremental_sleep option.

Incremental repair progress

Observe the progress of incremental repairs using the SSTable repaired metrics available in the dashboard graphs. See Tracking repaired SSTables for incremental repairs (page 481).

The Repair Status (page 474) tab displays a progress bar when an incremental repair is running.

Incremental tables are clearly indicated in the task details pane.
For more information and configuration examples, see Configuring incremental repairs (page 479).

**Repair Service behavior during environment changes**

The following sections provide details on how the Repair Service behaves when there are changes in the environment such as topology changes, down nodes, and OpsCenter restarts.

**Cluster topology changes**

The Repair Service is nearly immediately aware of any topology changes to a cluster. When a change in cluster topology occurs, the Repair Service stops its current repair cycle and waits for the ring to stabilize before restarting a new cycle. The restart period is controlled by the `restart_period` configuration option, which defaults to 300 seconds (5 minutes). While paused, the Repair Service checks the state of the cluster periodically using this period of time until it is able to reactivate.

Before resuming repairs, the Repair Service checks every 30 seconds by default for the cluster state. After the cluster has stabilized, the checks for cluster stabilization cease until the next time `opscenterd` is restarted. Configure the interval for the stable cluster check with the `cluster_stabilization_period` option.

Topoogy changes include:
- Nodes moving within a cluster
- Nodes joining a cluster
- Nodes leaving a cluster

**Schema changes**

When a schema change happens, the Repair Service pauses for five minutes by default, then starts back up and immediately begins repairing new keyspaces or tables. Schema changes include adding, changing, or removing keyspaces or tables.

**Down nodes or replicas**

A repair cannot run if any of the nodes in the replica set for that range are down. In the case where an entire rack or data center goes down, it is likely that no repair operations can be successfully run on the cluster. When one or more nodes are down, the Repair Service continues to run repairs for ranges and keyspaces unaffected by the down nodes.

When there are no runnable repair operations remaining, the Repair Service waits for 10 seconds and checks again. The Repair Service repeats this for up to the value configured for the `max_down_node_retry` option, which defaults to three hours based on the `max_hint_window_in_ms` property in `cassandra.yaml`, and then starts a new cycle. After the `max_hint_window_in_ms` is exceeded for a down node, the recovery process for that node is to rebuild rather than rely on hint replay. Therefore the Repair Service starts a new cycle to ensure that any available ranges continue to be repaired and are not blocked by down nodes.
**Note:** To mitigate the performance implications of scanning the entire list of remaining repair tasks, the scan for available ranges only scans the first `prioritization_page_size` tasks (default: 512). The order of these tasks is random, so if no available ranges are found in the first `prioritization_page_size`, it is unlikely there are any available ranges.

Persisted repair state when restarting `opscenterd`

At the end of each persist period (one hour by default), the current state of the Repair Service is persisted locally on the `opscenterd` server in the persist directory location. The persist period frequency can be configured with the `persist_period` option. The persist directory location can be configured with the `persist_directory` option. When `opscenterd` is restarted, the Repair Service resumes where it left off based on the persisted state information.

**Attention:** If automatic failover *(page 166)* is configured, be sure to mirror the repair service persist directory *(page 170)*.

For more information on repair service continuity during a failure, see *failover aftereffects* *(page 168)*.

**Estimating remaining repair time**

If the Repair Service anticipates it cannot complete a repair cycle within the allotted time to completion due to throughput, it displays a warning message and a newly estimated time remaining to complete the repair cycle. The Repair Service does not adjust the configured time to completion; it reports the revised estimate for completion without stopping the repair in progress.

When the Repair Service estimates that it will not finish a repair cycle within the configured `time_to_completion`, it triggers an ALERT in the OpsCenter Event Log. The alert is also visible in the `opscenterd.log`, as well as the Event Log in the Activities section of the OpsCenter UI. If *email alerts* *(page 143)* or *post-url alert notifications* *(page 145)* are configured, the alert notifications are emailed or posted.

The `error_logging_window` *(page 190)* configuration property controls both how often to log the message and how often to fire the alert if the Repair Service continues to estimate that it will not finish a repair in time.

**Parameters**

The `time_to_completion` parameter is the maximum amount of time it takes to repair the entire cluster one time.

**Note:** Typically, you should set the **Time to Completion** to a value lower than the lowest grace seconds before garbage collection setting `(gc_grace_seconds)` on your tables. The default for `gc_grace_seconds` is 10 days (864000 seconds). OpsCenter provides an estimate by checking `gc_grace_seconds` across all tables and calculating 90% of the lowest value. The default estimate for the time to completion
based on the typical grace seconds default is 9 days. For more information about configuring grace seconds, see \texttt{gc\_grace\_seconds} in the CQL documentation.

The Repair Service might run multiple subrange repairs in parallel, but runs as few as needed to complete within the amount of time specified. The Repair Service always avoids running more than one repair within a single replica set; there is no overlap in repairs between replica sets.

**Turning the Repair Service on**

Configure the days for the repair cycle to complete and enable the Repair Service for a cluster. The Repair Service takes a single parameter, \texttt{time\_to\_completion}, which is the maximum amount of time it takes to repair an entire cluster per repair cycle.

The Repair Service does not shut itself off. If necessary, turn off (page 473) the Repair Service.

**Warning:** Do not run a manual repair operation from Node administration or using the command line while the Repair Service is On.

The Repair service not enabled rule in the Best Practice Service (page 502) fails until the Repair Service is turned on, unless the rule has been turned off.

1. In the left navigation pane, click **Services**.

2. Click **Configure** for the Repair Service.

   The **Settings** tab appears.
3. The Repair Service must be off to enter a completion value. Enter a value for **Time to completion** field if your environment requires adjusting the value estimated by the Repair Service.

   **Note:** Typically, you should set the **Time to Completion** to a value lower than the lowest grace seconds before garbage collection setting (`gc_grace_seconds`) on your tables. The default for `gc_grace_seconds` is 10 days (864000 seconds). OpsCenter provides an estimate by checking `gc_grace_seconds` across all tables and calculating 90% of the lowest value. The default estimate for the time to completion based on the typical grace seconds default is 9 days. For more information about configuring grace seconds, see `gc_grace_seconds` in the CQL documentation.

   The frequency of repair cycles during the time to completion is controlled by the `time_to_completion_target_percentage` option. For more information, see Adjusting or disabling the throttle for subrange repairs (*page 488*).

4. Click the **Off** button to toggle the setting to **On**.

   The Repair Services starts and opens the **Status** tab.

   For more information about the repair status dashboard, see Viewing repair status (*page 474*).

**Turning the Repair Service off**

Stopping the Repair Service cancels any pending repairs. When the Repair Service is turned back on again, a new cycle of repairs begins.
Turn off the Repair Service if you need to adjust the **Time to completion** parameter value. If for some reason you need to perform a manual repair operation, be sure to turn the Repair Service off.

**Warning:** Do not run a manual repair operation from Node administration or using the command line while the Repair Service is On.

While the Repair Service is off, the **Repair service not enabled rule** in the Best Practice Service *(page 502)* will continue to fail unless the rule has been turned off.

<table>
<thead>
<tr>
<th>Config Advisor</th>
<th>Status</th>
<th>Schedule</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Repair service not enabled</td>
<td>HIGH</td>
<td>Every Day 5:00AM GMT</td>
<td>Configure</td>
</tr>
<tr>
<td>Repair service not configured correctly</td>
<td>HIGH</td>
<td>Every Day 5:00AM GMT</td>
<td>Configure</td>
</tr>
</tbody>
</table>

1. In the left navigation pane, click **Services**.
2. Click **Details** for the Repair Service.
3. Click the **Settings** tab.
4. Click the **On** button to toggle the setting to **Off**.

Any repairs that were in progress when the Repair Service was stopped continue running until completion. Any pending repairs are cancelled. The Status for the Repair Service on the Services page displays Off.

**Viewing repair status**

Access the Status Tab for the Repair Service

To access the Repair Status details:

- **Turn the Repair Service on** *(page 472)*. It immediately activates the Repair Service and opens the Status tab.
- If you are elsewhere in the monitoring application and the Repair Service is already activated, click **Details** for the Repair Service. The Repair Status tab displays full details for repair processes.

**Tip:** Information to fully understand all aspects of repair status are readily available from within the Repair Status tab. Hover over areas of the Status page to view inline information. Click tooltip icons to access short descriptions about an item. Click the **Read more** links to access the relevant Repair Service documentation.

**Monitor repair status**

Monitor the progress of incremental and subrange repairs in the Status tab. After turning on the Repair Service, the Repair Service Status is either Active or Paused:
When the Repair Service is actively processing repairs, the **Repair Service Status** indicates **Active**. The progress graphics and statistics reflect real-time measurements of repairs.

The Repair Service Status appears **Paused** in response to cluster or schema change events.

The repair process performs validation compaction *(page 476)* and streams *(page 476)* data to and from other nodes in the cluster when synchronizing replicas. Those activities when active are visible in their respective panes.

---

**Status pane**

Indicates whether the Repair Service Status is Active or Paused.

**View repair progress and statistics**

The progress and statistics pane displays progress bars for subrange and incremental repairs. A pie chart represents Completed, In Progress, and Failed repair tasks thus far. Remaining tasks are not represented in the pie chart; they are represented in the progress bars. The remaining time until the incremental and subrange repairs are completed is indicated underneath each respective progress bar.
DSE Management Services

**Note:** The Total Repairs value represents the number and percentage of the grand total of repair tasks for the current repair cycle. Repair tasks for each category's count can represent an aggregate of the tasks shown in the Table Repair Tasks pane. Repair tasks in a particular category might not equal the total number of tasks displayed in the Table Repair Tasks (page 477) pane because multiple tables might be aggregated into a single repair task. The number of tasks in the Table Repair Tasks pane are displayed and counted in all rows for tables within the range of a repair task.

View validation compactions

The Validation Compactions pane displays the progress of any validation compactions per node for both incremental and subrange repairs. In the absence of compaction activity, the No active validation compactions status is displayed.

**Note:** If repairs are configured for Running validation compaction sequentially (page 490), compaction progress is considerably slower, impacting both subrange and incremental repairs.

A validation compaction reads and generates a hash for every row in the stored tables, adds the result to a Merkle tree, and returns the tree to the initiating node as part of the underlying Merkle tree comparison process.

View streaming activity

The Streams pane displays an aggregate of streaming activity progress per node. The streams could be comprised of hundreds of files. When actively streaming data, the nodes from which the streams originate and their target node are shown along with progress bars for each node receiving streamed replica data. Otherwise, the No active streams status is displayed.
View repair tasks per table

The Table Repair Tasks pane provides insight into keyspace tables that are being repaired (or not if excluded), status summary, attempts at repair for skipped tasks, the type of repair, average repair time. To discover more:

- View keyspaces and tables excluded (page 478) from repairs, grouped by the exclusion criteria.
- View details (page 479) of repair tasks at the individual table level. Click a row to view repair task details isolated per keyspace table in the Repair Tasks for keyspace.table dialog.

Each column is sortable. Click a column heading to sort its column contents. The Status column provides visual status indicators along with a summary of completed, running, or pending repair tasks. Any task with errors displays a red explanation point.

The Total Attempts column indicates how many attempts (retries) the repair service has made before temporarily skipping the task. The skipped task is added to the end of the queue to retry later. The default maximum is 10 attempts. When that maximum is reached, an alert is fired and the Repair Service abandons any further repair attempts for that task. In the above graphic, 0/10 indicates all repair tasks completed without the need for any retry.
attempts. Configure the maximum attempts with the `single_task_err_threshold` (*page 189*) option.

Incremental repair tables are opted in for repair as mentioned in the incremental repairs overview (*page 468*). There are a few OpsCenter keyspace tables that are hard-coded for incremental repairs: `OpsCenter.backup_reports` and `OpsCenter.settings` tables. The incremental tooltip flags these as special tables and provides a link to documentation to configure additional tables or datacenters (*page 479*) to include in incremental repairs. Any tables configured by OpsCenter admins appear in the tasks pane sans the tooltip.

Incremental repairs have their own threshold setting for alerting about failed repair tasks. The default is 20. Configure with the `incremental_err_alert_threshold` (*page 190*) option.

Observe the progress of incremental repairs using the SSTable repaired metrics available in the dashboard graphs. See Tracking repaired SSTables for incremental repairs (*page 481*).

**View keyspaces and tables excluded from repairs**

Excluding keyspaces and tables from unnecessary repairs makes repair processes more focused, efficient, and faster with less workload impact on DSE clusters.

A link is available above the Table Repair Tasks pane for viewing keyspaces and tables excluded from subrange repairs. Click the View excluded tables link. The Excluded Keyspaces and Tables dialog displays the keyspaces excluded due to RF=1, system keyspaces, reserved tables, or those specifically configured for subrange repairs (*page 483*) to ignore.

**Note:** If using authentication, be sure to change the replication strategy and replication factor for the dse_security and system_auth keyspaces so that those keyspaces are included in repairs. See Managing keyspaces and tables (*page 265*).
View details for repair tasks

Click any row of the **Table Repair Tasks** pane to view more details about a particular task.

The **Repair Tasks for** `keyspace.table` dialog provides details for the number of Succeeded, Failed, Running, Pending, and Aborted repair tasks for each repair-eligible table in a keyspace. The Average Repair Time and number of Attempts (*configurable* ([page 189](#))) for the repair task are also shown.

**Basic repair configuration**

**Configuring incremental repairs**

The Repair Service runs an **incremental repair** ([page 468](#)) on a user-configured set of tables. For DSE 5.1 and later, OpsCenter starts an incremental repair when the incremental threshold of 1 KB (default) of unrepaired data is detected on designated tables. The Repair
Service sleeps for an hour between completed incremental repair cycles. If the number of errors during an incremental repair exceeds its threshold, an alert is sent to the Event Log.

**Prerequisites:**

Manually migrate tables to use incremental repair. Any incorrectly formatted table logs an error. For information on migrating to incremental repairs in DSE, see [migrating to incremental repairs](#).

- Update the list of tables to include in incremental repairs using the `incremental_repair_tables` configuration option.
  
  **Note:** The `OpsCenter.settings` and `OpsCenter.backup_reports` tables are included in incremental repairs by default.

- Adjust the default thresholds to trigger incremental repairs and error alerts only if necessary for your environment.
- Set the default sleep time between ending and starting a subsequent incremental repair only if necessary for your environment.

**opscenterd.conf**

The location of the `opscenterd.conf` file depends on the type of installation:

- **Package installations:** `/etc/opscenter/opscenterd.conf`
- **Tarball installations:** `install_location/conf/opscenterd.conf`

**cluster_name.conf**

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

- **Package installations:** `/etc/opscenter/clusters/cluster_name.conf`
- **Tarball installations:** `install_location/conf/clusters/cluster_name.conf`

**Configuration options for incremental repairs**

The following options are currently configurable by adding a `[repair_service]` section to the `opscenterd.conf` file to apply to all clusters, or per cluster by adding the section to the `cluster_name.conf` file. Settings in `cluster_name.conf` override any settings in `opscenterd.conf`. After changing configuration, restart `opscenterd` (page 75).

- **[repair_service] incremental_repair_datacenters**
  
  Restricts incremental repairs by datacenters or racks. Setting this option improves performance by limiting the repair requests to only those replicas within the datacenters and any specified racks. Example: `dc1,dc2:rack1,dc2:rack2`. The default behavior sends repair requests to all datacenters and racks for all replicas.

- **[repair_service] incremental_repair_tables**
  
  The list of keyspaces and tables to include in incremental repairs. The `OpsCenter.settings` and `OpsCenter.backup_reports` tables are included by default. Example: `keyspace1.standard1, keyspace1.standard2`.

- **[repair_service] incremental_sleep**
  
  The number of seconds to pause after completing all incremental repairs for the cluster. Default: 3600 (1 hour).
[repair_service] incremental_threshold

The minimum number of bytes required to consider a table for incremental repairs (DSE 5.1+ only). The default value of 1 byte means that if there is any unrepaired data in a table, the Repair Service will run an incremental repair. Be cautious of setting this value too high. If not enough data is written to exceed the threshold in the gc_grace_seconds period, deletes might be lost. Default: 1.

[repair_service] incremental_err_alert_threshold

The threshold for the number of errors during incremental repair to ignore before alerting that incremental repair seems to be failing more than an acceptable amount. Default: 20.

1. Open for editing opscenterd.conf for all clusters, or cluster_name.conf for a specific cluster.

2. Set the following incremental options for your environment requirements in the [repair_service] section:

   The following example restricts incremental repairs by datacenter (dc1) and rack (rack1), lists the tables to perform incremental repairs on, doubles the sleep between incremental repairs to 2 hours, bumps the threshold to 2 KB of unrepaired data for triggering an incremental repair for a DSE version 5.1 cluster, and doubles the default error threshold to 40 errors before sending an alert:

   ```
   [repair_service]
   incremental_repair_datacenters=dc1:rack1
   incremental_repair_tables=OpsCenter.settings,OpsCenter.backup_reports,keyspace1.standard1,keyspace2.standard2
   incremental_sleep=7200
   incremental_threshold=2
   incremental_err_alert_threshold=40
   ```

   **Caution:** Exercise caution when setting the incremental_threshold option. Setting the threshold too high might result in lost deletes during repairs. If deletes are not properly replicated, deleted data could be resurrected (also referred to as zombie data).

3. Restart opscenterd (page 75).

### Tracking repaired SSTables for incremental repairs

Configure dashboard graphs to visually monitor the progress of repaired SSTables at the node and table levels during an incremental repair. The Repair Service uses the SSTables Repaired metrics to determine whether it should run incremental repairs, based on the configured value of the incremental_threshold (page 190) option.

Add graphs for the following metrics:

**Percent Data Repaired [percentage-repaired]**

Percentage of data (uncompressed) marked as repaired across all non-system tables on a node. Tables with a replication factor of 1 are excluded.

**TBL: Percent Data Repaired [cf-percentage-repaired]**
DSE Management Services

Percentage of data (uncompressed) marked as repaired for a given table on a node. This metric is only meaningful for replication factor > 1.

The SSTables Repaired metrics are only applicable to DSE versions 5.1 and later.

**Note:** The SSTables Repaired metrics are only meaningful for keyspaces that have a replication factor of 2 or greater. The percentage repaired for a keyspace that has an RF=1 displays percentage repaired as 0%.

**Prerequisites:** Configure tables to include in incremental repairs. Adjust the incremental threshold if necessary. See *Configuring incremental repairs (page 479).*

1. Click **Cluster#Dashboard**.

2. **Clone** *(page 329)* the Default preset tab and give it a name such as **Repairs**.

3. Click **Add Graph**.

   The **Add Metric** dialog appears.

4. Choose the **SSTables Repaired** metric from the **Metric** list.

   **Tip:** Type **ss** to quickly locate the metric.

5. Choose the nodes to monitor from the **Node** list.

6. Click **Add Metric**.

   The metric is added to the dashboard.

7. Repeat **3** *(page 482)* through **6** *(page 482)* for the **TBL: SSTables Repaired metric** and select the tables to monitor from the **Tables** list.

8. Repeat this procedure for all DSE 5.1 clusters that you need to monitor for incremental repairs.
Excluding keyspaces or tables from subrange repairs

Exclude entire keyspaces or specific tables within a keyspace from unnecessary repairs during a subrange repair. Exclude entire keyspaces and all its tables within that keyspace, or exclude specific tables within a keyspace. Doing so reduces use of system resources during the repair process, and can shorten the repair cycle time. The OpsCenter Repair Service already excludes certain system keyspaces from repair by default. Keyspaces with a replication factor of 1 are always automatically excluded from repairs.

View excluded keyspaces and tables (page 478) from within the Repair Status (page 474) tab.

Examples of keyspaces that are candidates for exclusion are those for which data loss is not a concern, such as those that summarize data duplicated or rebuilt elsewhere. Additional scenarios for excluding keyspaces might those with be a wide variety in gc_grace_seconds table values, and those keyspaces that are maintained by a custom script.
The location of the `opscenterd.conf` file depends on the type of installation:

- **Package installations**: `/etc/opscenter/opscenterd.conf`
- **Tarball installations**: `install_location/conf/opscenterd.conf`

1. Open for editing `opscenterd.conf` for all clusters, or `cluster_name.conf` for a specific cluster.

2. Set the following options for your environment requirements in the `[repair_service]` section:
   - **[repair_service] ignore_keyspaces**
     The list of keyspaces to ignore during a subrange or distributed subrange repair, in addition to system keyspaces or those with a replication factor of 1.
   - **[repair_service] ignore_tables**
     The list of keyspace-qualified table names to exclude during a subrange or distributed subrange repair, including reserved tables, those that reside in a system keyspace, and tables belonging to keyspaces with a replication factor of 1. Example: `keyspace1.standard1, keyspace1.standard2`.

   The following configuration example excludes both specific keyspace in their entirety (keyspaces 1, 2, and 3); and excludes specific tables within keyspaces (tables X, Y, and X in keyspaces A and B):
   ```
   [repair_service]
   ignore_keyspaces=keyspace1,keyspace2,keyspace3
   ignore_tables=mykeyspaceA.tableX,mykeyspaceA.tableY,mykeyspaceB.tableZ
   ```

3. Restart `opscenterd` (page 75).

   The Repair Service ignores the specified keyspaces or tables during repairs.

### Enabling distributed subrange repairs

Enable distributed subrange repairs for large clusters that have large numbers of standard subrange repair tasks. The distributed subrange repair feature is available for OpsCenter versions 6.1.7 and 6.1.8. The feature is not available in 6.5.0.

The location of the `opscenterd.conf` file depends on the type of installation:

- **Package installations**: `/etc/opscenter/opscenterd.conf`
- **Tarball installations**: `install_location/conf/opscenterd.conf`

1. Open `opscenterd.conf` for editing.

2. Set the following option in the `[repair_service]` section:
[repair_service] enable_distributed_subrange_repair

Specifies whether to enable the Distributed Subrange Repair job type in the Repair Service, which is designed to better scale for large, dense clusters that require an extremely large number of individual subrange repairs. Default: False.

[repair_service]
enable_distributed_subrange_repair=True

3. **Restart opscenterd (page 75).**

   Track the progress of distributed subrange repairs in the Repair **Status** panel.

   **Important:** The percentage of the time to complete repairs might not update because the [repair_service] time_to_completion_target_percentage (page 190) parameter is not supported for distributed subrange repairs.

   ![Repair Status Panel](image)

   Track the progress of distributed subrange repairs in the Repair panel within the **Nodes** summary area.
Logging for the Repair Service

Repair Service log names and locations

All Repair Service activity is logged by default to a log file in the `repair_service` directory applicable to the install type and each cluster name:

**Package installs**

```shell
/var/log/opscenter/repair_service/<cluster_name>.log
```

**Tarball installs**

```shell
<install_location>/log/repair_service/<cluster_name>.log
```

Rotating the Repair Service log

The log file is automatically rotated at ~9.5MB (`<maxFileSize>` logback setting), keeping up to ten rotated logs by default (`<maxIndex>` logback setting).

Configuring the Repair Service log

The Repair Service log options and levels are configurable using `logback.xml` (**page 215**).

The following example shows a standard configuration for a package install. Replace `cluster_name` with the name of your cluster. The log level is set at the INFO level by default. The example shows temporarily setting the level to DEBUG level for troubleshooting purposes.

```xml
<appender name="cluster_name_log" class="ch.qos.logback.core.rolling.RollingFileAppender">
  <file>./var/log/opscenter/repair_service/cluster_name.log</file>
  <encoder>
    <pattern>%date{ISO8601, UTC} [%X{cluster_id:-opscenterd}]
%5level: %msg (%thread)%n%exception{20}</pattern>
  </encoder>
  <rollingPolicy class="ch.qos.logback.core.rolling.FixedWindowRollingPolicy">
    <maxFileSize>10MB</maxFileSize>
    <maxIndex>10</maxIndex>
  </rollingPolicy>
</appender>
```
Basic Repair Service configuration reference

This reference contains the basic Repair Service options without defaults that can be configured by all customers for their unique environment, such as which tables to include in incremental repairs, or which keyspaces or tables to exclude from subrange repairs. Subrange repairs exclude certain system keyspaces and tables by default. Incremental repairs include certain OpsCenter tables by default. Any tables being repaired incrementally are not subjected to subrange repairs.

The following options are currently configurable by adding a `[repair_service]` section to the `opscenterd.conf` file to apply to all clusters, or per cluster by adding the section to the `cluster_name.conf` file. Settings in `cluster_name.conf` override any settings in `opscenterd.conf`. After changing configuration, restart `opscenterd` (page 75).

If there are any issues with the Repair Service configuration, the Repair service not configured correctly rule in the Best Practice Service (page 502) fails and provides guidance as to incorrectly configured options, unless the rule has been turned off.

Basic configuration for subrange repairs

Flag which keyspaces or tables therein to exclude from subrange repairs:

```
[repair_service] ignore_keyspaces
```

The list of keyspaces to ignore during a subrange or distributed subrange repair, in addition to system keyspaces or those with a replication factor of 1.

```
[repair_service] ignore_tables
```

The list of keyspace-qualified table names to exclude during a subrange or distributed subrange repair, including reserved tables, those that reside in a system keyspace, and tables belonging to keyspaces with a replication factor of 1. Example: keyspace1.standard1, keyspace1.standard2.
Basic configuration for incremental repairs

Flag which tables or datacenters to include in incremental repairs:

**[repair_service] incremental_repair_datacenters**
Restricts incremental repairs by datacenters or racks. Setting this option improves performance by limiting the repair requests to only those replicas within the datacenters and any specified racks. Example: dc1,dc2:rack1,dc2:rack2. The default behavior sends repair requests to all datacenters and racks for all replicas.

**[repair_service] incremental_repair_tables**
The list of keyspaces and tables to include in incremental repairs. The OpsCenter.settings and OpsCenter.backup_reports tables are included by default. Example: keyspace1.standard1, keyspace1.standard2.

**Note:** The OpsCenter.settings and OpsCenter.backup_reports tables are included in incremental repairs by default.

**Note:** The OpsCenter.settings and OpsCenter.backup_reports tables are flagged as special incremental repair tables in the Table Repair Tasks pane (page 477) within the Repair Service Status page.

For more information about other incremental config options, see Configuring incremental repairs (page 479).

**Advanced repair configuration**

**Adjusting or disabling the throttle for subrange repairs**

Adjust or disable the throttle for the subrange repair cycle.

**Note:** This feature is only applicable to subrange repairs. It has no impact on incremental repairs.

The Repair Service automatically throttles subrange repairs when the current repair cycle is estimated to finish significantly before the deadline specified by the time to completion. The throttle sleeps longer between repairs to alleviate load on the server during the repair process, dynamically tuning a repair job to run within the allotted time. Without the throttle enabled, repair processes that complete quickly unnecessarily repeats the repair. For instance, if it takes 3 days to repair a cluster and Repair Service is configured to complete in 9 days, a throttled Repair Service spreads out the repair over the 9 day period. In contrast, an unthrottled service would run 3 repair cycles in that same 9 day period.

The `time_to_completion_target_percentage` configuration option controls the frequency and pace of the subrange repair process. The throttle slows repairs or reduces parallel repair processes as necessary to prevent overloading the cluster while still completing a repair cycle within the specified time window designated by the **Time to completion** value. The default value for the target percentage to complete a repair is 65%.
Adjusting this setting is not typically required. The throttle is particularly useful for smaller clusters that have a subrange repair that completes within a short period of time, such as one day.

Disable the throttle only if you do not have any concerns about server workload impact from running repairs with greater frequency, or if you intend to configure advanced options such as `max_parallel_repairs` or `min_repair_time`.

**opscenterd.conf**

The location of the `opscenterd.conf` file depends on the type of installation:

- **Package installations:** `/etc/opscenter/opscenterd.conf`
- **Tarball installations:** `install_location/conf/opscenterd.conf`

**cluster_name.conf**

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

- **Package installations:** `/etc/opscenter/clusters/cluster_name.conf`
- **Tarball installations:** `install_location/conf/clusters/cluster_name.conf`

1. Open for editing `opscenterd.conf` for all clusters, or `cluster_name.conf` for a specific cluster.

2. Adjust the configuration as appropriate for your environment:

   Setting the percentage to zero disables the throttle:
   
   ```
   [repair_service]
   time_to_completion_target_percentage=0
   ```

   Disabling the throttle is not recommended unless you intend to configure options such as `maximum_parallel_repairs` manually. See Setting the maximum for parallel subrange repairs *(page 493).*

   After disabling the throttle, the Repair Service logs messages that inform you about the behavior change.

   Setting the percentage to 100 spaces out subrange repairs over the entire time to completion window at the potential risk of not completing the repairs on time:
   
   ```
   [repair_service]
   time_to_completion_target_percentage=100
   ```

   At 100% targeted percentage, the workload is spread out over a longer period of time, which lowers the impact on the cluster at any given time. However, missing the time to completion deadline is more likely without the extra time buffer afforded by the target default of 65%.
3. Restart opscenterd (page 75).

4. Monitor the subrange repair progress on the Status tab (page 474).

5. Review the repair service log (page 486) messages for awareness about the throttle recalibration impact on your environment.

**Running validation compaction sequentially**

Set the `snapshot_override` configuration option to true to run validation compaction sequentially for both subrange and incremental repairs. The repair takes a snapshot of all data on all the nodes and then each node validates the snapshot one at a time.

Switching the repair snapshot behavior might be necessary for latency sensitivity reasons. The snapshot override option determines the manner in which Merkle trees are built during validation compaction: all at once in parallel (default); or one node at a time to reduce impact on server load.

**Attention:** Be aware that enabling the snapshot override results in considerably slower repair completion times. Roughly estimate the slowdown as equivalent to the replication factor. For instance, with an RF of 3, the repairs builds the Merkle tree three times, one at a time, so sequential compaction could take at least three times as long as parallel.

**opscenterd.conf**

The location of the `opscenterd.conf` file depends on the type of installation:

- **Package installations:** `/etc/opscenter/opscenterd.conf`
- **Tarball installations:** `install_location/conf/opscenterd.conf`

1. Open `opscenterd.conf` for editing.

2. Set the `snapshot_override` option to True:
   
   ```
   [repair_service]
   snapshot_override=True
   ```

3. Restart opscenterd (page 75).

4. Monitor the subrange and incremental repair progress on the Status tab (page 474). Be sure to observe the slower progress of validation compactions in the Validation Compactions pane. After differences are determined by Merkle tree comparisons, the streaming progress continues at its regular pace as shown in the Streams pane.

**Advanced Repair Service configuration reference**

The following options are currently configurable by adding a `[repair_service]` section to the `opscenterd.conf` file to apply to all clusters, or per cluster by adding the section to...
the cluster_name.conf file. Settings in cluster_name.conf override any settings in opscenterd.conf. After changing configuration, restart opscenterd (page 75).

If there are any issues with the Repair Service configuration, the Repair service not configured correctly rule in the Best Practice Service (page 502) fails and provides guidance as to incorrectly configured options, unless the rule has been turned off.

The configuration options prefixed with incremental_* are only applicable to incremental repairs.

[repair_service] cluster_stabilization_period
The frequency in seconds the repair service checks for cluster stability before making repairs. This check begins when repair service is activated (either by a user or after an OpsCenter restart) and repeats until the cluster is stable. Default: 30.

[repair_service] error_logging_window
The frequency in seconds to log errors and trigger alerts after exceeding time_to_completion. Default: 86400 (1 day).

[repair_service] incremental_err_alert_threshold
The threshold for the number of errors during incremental repair to ignore before alerting that incremental repair seems to be failing more than an acceptable amount. Default: 20.

[repair_service] incremental_sleep
The number of seconds to pause after completing all incremental repairs for the cluster. Default: 3600 (1 hour).

[repair_service] incremental_threshold
The minimum number of bytes required to consider a table for incremental repairs (DSE 5.1+ only). The default value of 1 byte means that if there is any unrepaired data in a table, the Repair Service will run an incremental repair. Be cautious of setting this value too high. If not enough data is written to exceed the threshold in the gc_grace_seconds period, deletes might be lost. Default: 1.

[repair_service] max_down_node_retry
The maximum number of attempts to retry a repair task when a node containing a replica is down. The default is 1080 retry attempts. Retries occur every 10 seconds. The default 1080 retries elapses after 10800 seconds (3 hours), which corresponds to the default cassandra hinted-handoff expiration. Example: To double the time allowed to attempt repairs on a down node or replica to 6 hours, set the number of retries to 2160. Default: 1080.

[repair_service] max_pending_repairs
The maximum number of pending repairs allowed to be running on a node at one time. Default: 5.

[repair_service] min_repair_time
The minimum length of time in seconds for a repair to complete. If a repair finishes sooner it will be padded with a sleep. Default: 5.

[repair_service] persist_directory
The location in which to store a file with the current repair service status. The default location is /var/lib/opscenter/repair_service for package installations and install_location/repair_service for tarball installations.

[repair_service] persist_period
The minimum number of seconds between Repair Service writing the persist file to disk. Default: 3600 (1 hour). This parameter applies to subrange and incremental repairs only, and is not applicable to distributed subrange repairs.

[repair_service] restart_period
The period of time in seconds repair service pauses in response to certain events before verifying the cluster stability and restarting repairs. Default: 300 (5 minutes).

[repair_service] single_repair_timeout
The maximum length of time for a repair to complete, in seconds. Default: 3600 (1 hour).

[repair_service] single_task_err_threshold
The maximum number of times to retry a repair task before temporarily skipping the task and moving on to the next task. The skipped task is moved to the end of the repairs queue to retry later. After the maximum retries is reached, an alert is fired. Default: 10.

[repair_service] snapshot_override
Specifies whether to override the default snapshot repair behavior. Specifying this option as True runs validation compaction sequentially rather than in parallel. Default: False.

[repair_service] time_to_completion_target_percentage
A percentage of the time to completion that the repair service should target, including slowing down or reducing parallelism as necessary to avoid overtaxing the cluster. Default: 65. This parameter applies to subrange and incremental repairs only, and is not applicable to distributed subrange repairs.

opscenderd.conf

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

- Package installations: /etc/opscenter/opscenderd.conf
- Tarball installations: install_location/conf/opscenderd.conf

cluster_name.conf

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

- Package installations: /etc/opscenter/clusters/cluster_name.conf
• Tarball installations: `install_location/conf/clusters/cluster_name.conf`

**Expert repair configuration**

**Setting the maximum for parallel subrange repairs**

Set the maximum number of subrange repairs to run in parallel to tune slow running repairs, or to troubleshoot repairs.

When the `max_parallel_repairs` option is unspecified or set to 0 (default), the Repair Service calculates the correct number of maximum repairs to run in parallel. The basic calculation is ceiling(total # nodes in cluster / maximum total RF). The calculation prevents replica sets from overlapping during repairs.

**Caution:** DataStax recommends only manually adjusting the `max_parallel_repairs`, changing `min_repair_time` and other advanced (page 490) or expert (page 494) options only if the `time_to_completion_percentage` throttle is not is use. See Adjusting or disabling the throttle for subrange repairs (page 488).

1. Open for editing `opscenterd.conf` for all clusters, or `cluster_name.conf` for a specific cluster.

2. Adjust the configuration as appropriate for your environment:

   Setting the `max_parallel_repairs` to 0 (or leaving it blank) makes the Repair Service dynamically calculate the number of subrange repairs to run in parallel based on the formula previously described:

   ```
   [repair_service]
   max_parallel_repairs=0
   ```

   This is the default behavior for determining maximum parallel subrange repairs. DataStax recommends using the default dynamic setting for maximum parallel repairs in conjunction with the throttle provided by the `time_to_completion_target_percentage` option.

   Setting the `max_parallel_repairs` to 1 forces the Repair Service to run only one repair task at a time:

   ```
   [repair_service]
   max_parallel_repairs=1
   ```

   Subrange repairs and incremental repairs alternate running tasks one at a time. The Repair Service waits for a subrange repair to complete before starting an incremental repair and vice versa. Forcing the repairs to process tasks one at a time and alternate between incremental and subrange repairs can be helpful when trying to isolate issues during troubleshooting.
If subrange repairs are running slowly with the dynamically calculated value (default behavior with 0 or unset as shown in the first example), manually set the number of maximum parallel repairs:

```
[repair_service]
max_parallel_repairs=4
```

Experiment with adjusting this value until repairs are processing as expected for your environment.

3. Restart opscenterd *(page 75).*

4. Monitor the repair progress on the Status tab *(page 474).*

5. Review the repair service log *(page 486)* messages for awareness about the impact the configuration change has on your environment.

**Expert Repair Service configuration reference**

Expert repair configuration options

There are some configuration options exposed only for adjustment with the assistance of an expert in DataStax Support.

**[repair_service] max_parallel_repairs**

The maximum number of subrange repairs to run in parallel. If unspecified or set to 0, the Repair Service calculates the correct number of maximum repairs to run in parallel. Default: 0.

**[repair_service] min_throughput**

The minimum throughput in bytes needed to calculate parallel repairs. See also num_recent_throughputs. Default: 512. This parameter applies to subrange and incremental repairs only, and is not applicable to distributed subrange repairs.

**[repair_service] num_recent_throughputs**

The maximum most recent completed repair throughputs used to calculate the average repair throughput, which is then used to determine how many parallel repairs are needed. See also min_throughput. Default: 500. This parameter applies to subrange and incremental repairs only, and is not applicable to distributed subrange repairs.

**[repair_service] offline_splits**

The minimum number of subrange splits for a node to have per keyspace when falling back to offline splits if the node is unable to communicate with its agent to get more optimal splits. Default: 256. This parameter applies to subrange repairs only.

**[repair_service] prioritization_page_size**

The maximum number of remaining subrange repair tasks for the repair service to evaluate when choosing the next low-impact repair to run. Default: 512. This parameter applies to subrange repairs only.

**[repair_service] tokenranges_http_timeout**
The timeout in seconds for the HTTP call to the agent to retrieve node token range splits. The default value (30) is deliberately set higher than the default value for DataStax Agents http_timeout in clusternname.conf (10). Default: 30. This parameter applies to subrange repairs only, and is not applicable to incremental or distributed subrange repairs.

[repair_service] tokenranges_partitions
Target number of partitions for each range in a subrange repair. This value is used by the Repair Service to create repair tasks, and is set to the max DSE merkle tree depth by default. Default: 1048576.

Note: Do not set tokenranges_partitions higher than default. See Tuning Repair Service for multi-datacenter environments (page 495).

Tuning Repair Service for multi-datacenter environments

When running the Repair Service on a multi-datacenter cluster, consider the number of total repair tasks and over-streaming.

Reduce the number of repair tasks

A single repair task is made up of at least six network requests between any two peers. Reducing the total number of repair tasks can drastically reduce network overhead and the time to complete a full Repair Service cycle. The number of repair tasks is controlled by how many partitions are targeted for each subrange. If there are more partitions in a subrange, each subrange is larger, which means fewer total subranges. The tokenranges_partitions property controls the targeted partition count.

Avoid over-streaming

Over-streaming occurs when a subrange is repaired that contains more partitions than the maximum merkle tree depth. This occurs if the tokenranges_partitions is set too high.

Guidelines for tuning

- Never set tokenranges_partitions higher than the default 1048576, which is max-merkle-tree-depth of 220.
- Test the tuning on the cluster prior to production. Look for the total number of repair tasks, average repair task time, and impact on cluster performance.
- If single repair tasks take longer than 20-30 minutes and a full Repair Service cycle is within gc_grace_seconds, halve the tokenranges_partitions and re-test.
- To check for over-streaming, ensure the following line does not exist in system.log:

  Range X with Y partitions require a merkle tree with depth Z but the maximum allowed depth for this range is 20.
Note: X, Y, and Z are variables.

**Expeditied Repair Service configuration**

To repair one or more nodes as quickly as possible without regard to application performance, use the Repair Service to repair the entire cluster. Tell the Repair Service to run as fast as it can without severely impacting cluster performance.

1. Stop the Repair Service.

2. Restart Repair Service with a very low time-to-complete value, such as .01 days.

   **Note:** The Repair Service logs will warn that the Repair Service cannot be completed within .01 days. This is expected. The Repair Service will continue to run.

   The Repair Service will run its maximum number of parallel repair tasks and avoid any delays between tasks. The status of the running repair cycle can be seen in the Repair Service UI as a progress bar. Once this cycle completes, a new cycle will start and reset the progress bar.

3. Once one full cycle has been completed with this time-to-complete setting, restart the Repair Service with your standard time-to-complete value.

**Troubleshoot Repair Service errors**

To resolve errors, try adjusting the configuration options (page 490) in the [repair_service] section of opscenterd.conf or cluster_name.conf as appropriate for your environment. Errors encountered when running the Repair Service can include:

**Error of a single repair task**

When a single repair task fails, the repair is skipped temporarily and added to the end of the queue of repairs and retried later. If a single repair fails ten times (default), the Repair Service fires an alert. Adjust this setting with the single_task_err_threshold (page 189) option.

The Repair Attempts display in the Table Repair Tasks (page 477) pane of the Repair Service Status (page 474) page.

**Incremental error alert threshold exceeded**

By default, the number of failed incremental repair attempts defaults to 20 before sending an alert that there could be a problem with incremental repair. Adjust this setting with the incremental_err_alert_threshold (page 190) option.

**Offline splits**

At the beginning of each cycle, the Repair Service attempts to generate intelligent subrange splits based on the system.size_estimates table. The subrange splits cannot happen when a node or agent is down or unavailable. If a node or agent is unavailable when the subrange determination happens, Offline Splits are used.
In large or dense clusters, these offline subrange calculations can often be inefficient. The best way to detect that a Repair Service cycle has fallen back to offline splits is to monitor the Repair Service (page 486) log for using offline task generation. If offline splits are detected, restart the Repair Service once all nodes/agents are up and available.

**Repair history tables**

DSE stores repair events details in the system_distributed.repair_history and system_distributed.parent_repair_history tables. By default these tables have no time to live (TTL), which can lead to significant unnecessary disk usage because of the number of repair tasks being run continuously.

Manually set a TTL on these tables based on your needs. In most cases, the TTL should be larger than gc_grace_seconds, but not more than needed for debugging purposes.

**Skipping range because pending repairs exceeds the max repairs**

The Repair Service skips repairing a range if pending repairs exceed the maximum pending repairs, which is 5 by default. The Repair Service immediately moves the skipped repair task to the end of the repair queue and fires an alert. At your discretion, you might want to restart any stalled nodes. Adjust this setting with the max_pending_repairs (page 189) option.

**Timeouts**

The Repair Service times out a single repair task after one hour by default. This counts towards an error for that repair task and it is placed at the end of the queue of repairs and retried later. Adjust this setting with the single_repair_timeout (page 189) option.

**Too many repairs in parallel**

The Repair Service errors if it has to run too many repairs in parallel. By default, this error happens if it estimates that it needs to run more than one repair in a single replica set to complete on time. Try increasing the Time to completion parameter. If that does not resolve the issue, try adjusting the max_parallel_repairs (page 189) option. See Setting the maximum for parallel subrange repairs (page 493).

**Caution:** DataStax recommends only manually adjusting the max_parallel_repairs, changing min_repair_time and other advanced (page 490) or expert (page 494) options only if the time_to_completion_percentage throttle is not in use. See Adjusting or disabling the throttle for subrange repairs (page 488).

**opscenterd.conf**

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

- **Package installations:** /etc/opscenter/opscenterd.conf
- **Tarball installations:** install_location/conf/opscenterd.conf
cluster_name.conf

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

- Package installations: /etc/opscenter/clusters/cluster_name.conf
- Tarball installations: install_location/conf/clusters/cluster_name.conf

Learn more about repairs

DataStax learning resources about repairs

Although the Repair Service features provided by OpsCenter are automated and implemented a bit differently than standard repairs in DSE Cassandra, it is still helpful to become familiar with the underlying processes involved in a repair. Here are some links to learn more if you are not familiar with repair intricacies and the corresponding manual repair procedures:

- **DataStax Academy**: View the Run a repair operation course.
- **DataStax Developer Blogs**:
  # Read about efficiencies of incremental repairs.
  # Read about Repair in Cassandra.
  # Read about subrange repair in Advanced repair techniques
  # Read tips about interpreting repair logs
- **DataStax Documentation**: Review the content for manual repair procedures:
  # Repairing nodes
  # Nodetool repair reference
  # KB article: Manually tune data streaming for repair

**Warning**: Do not run a manual repair operation from Node administration or using the command line while the Repair Service is On.

Capacity Service

Using trend analysis and forecasting, the Capacity Service helps you understand how a cluster is performing within its current environment and workload, and gain a better sense of how time affects those trends, both past and future. Several types of metrics are collected (page 151) by the Capacity Service, including Cassandra-specific and platform-specific metrics (for example, disk and network metrics), at both the node and table level (where applicable). These metrics are stored in Cassandra on the cluster being managed by default. That metrics data can be stored on a separate dedicated cluster (page 154) as well.

Trend Analysis

The Trend Analysis component of the Capacity Service allows viewing historical metrics for any node or table, as well as aggregates across an entire cluster.
Forecasting

The Forecast feature (page 499) allows viewing a predicted trend for any metric, based on collected historical data. A predicted trend uses polynomial curve fitting against historical data. By default, the polynomial equation is degree 3. If the predicted results need adjustment or improvement, send the results to DSE OpsCenter using the Help us improve these results, and contact DataStax Support for assistance.

Forecasting trends for metric graphs

Use forecasting to predict trends in metric graphs based on historical data.

1. In the Dashboard, locate the metric graph to forecast and click Forecast from the graph menu.

   ![Forecast graph metric dialog appears.]

2. Select the date and time for the end of the predicted trend timeframe.
   The end date and time must be a minimum of two days into the future, up to a maximum of one year.

   ![Select a date and time to forecast your data]

3. Click Forecast.
   The forecast displays the predicted metric trend for the selected timeframe.
The example forecast above shows that the node disk usage will double from 20% to 40% in one month. In this case, admins should consider adding capacity now.

4. To help DataStax improve your predicted trends, click the Help us improve these results link to forward the results to the DSE OpsCenter team.

Advanced forecast configuration

The default forecasting options are configurable by adding a [forecasting] section to the opscenterd.conf file to propagate to all clusters, or adjusting the defaults per cluster in cluster_name.conf.
opscenterd.conf

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

- Package installations: /etc/opscenter/opscenterd.conf
- Tarball installations: install_location/conf/opscenterd.conf

cluster_name.conf

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

- Package installations: /etc/opscenter/clusters/cluster_name.conf
- Tarball installations: install_location/conf/clusters/cluster_name.conf

1. To propagate the configuration to all clusters, open opscenterd.conf for editing.

2. Add a [forecasting] section with the following available configuration options:

   [forecasting]
   range_multiplier
   The multiplier for the query range needed to produce forecasts. The default multiplier is 3.

   [forecasting] required_data_percentage
   Minimum percent of past data required to forecast. The default value is 0.5.

   To generate a meaningful prediction, historical data is analyzed for a longer period than the range being forecasted. The default range multiplier is three times larger than the range being forecasted. For example, to forecast one month into the future, three months of data is analyzed.

   The following example decreases the default range multiplier for the forecast timeframe, and decreases the required percentage amount of historical data, which might be necessary for a newer cluster to forecast a trend. If there is not enough data when forecasting, an error indicates less than 50 percent of the required past data was available. After a cluster has more historical data to work with, increasing the forecasting configuration values can increase the accuracy of forecasted trends.

   [forecasting]
   range_multiplier = 3
   required_data_percentage = 0.25

3. To adjust the defaults per cluster, open cluster_name.conf for editing and adjust the forecasting options as appropriate per cluster.

4. Save the configuration file or files.
5. Restart (page 75) the OpsCenter daemon.

**Best Practice Service**

The Best Practice service periodically scans clusters to automatically detect issues that affect a cluster's health. It includes a set of rules organized into categories that are used to analyze the nodes in a cluster and report back any deviations from the best practice recommendations. The status dialog for each rule includes recommendations for fixing any problems. See Monitoring the results of Best Practice service scans (page 503).

The Best Practice Service is enabled by default and cannot be disabled in its entirety. Individual rules can be turned on or off as appropriate for an environment. The status summary of passing or failing active rules is shown on the main Services page in the Status column.

Also by default, the majority of best practice rules are enabled (with the exception of the Sensitive Config Value Encryption rule in the Security Advisor) and configured to run at 5:00 AM GMT. Configure (page 502) which rules to run on a schedule appropriate for your organization.

If a best practice rule fails, it sends an alert (page 351). Similar to other alerts, you can configure notification settings for these rules.

**Configuring Best Practice service rules**

Best Practice service rules can be enabled, scheduled, and disabled from OpsCenter. Configure which rules are active when scanning a cluster and how often the rule should run.

1. Click `cluster name` Services.

2. Click the Details link for the Best Practice Service.

3. Click the category name to show the rules for that category.

4. Most rules are enabled by default. To disable a rule:
a. Click **Turn Rule Off**.

b. Click **Disable Rule** to confirm disabling the rule. You can re-enable the rule at any time.

5. To enable a rule:

   a. Click **Turn Rule On**.

   b. Select a date and time to start the rule, and indicate how often the rule should run.

   c. Click **Save Rule**.

6. To change the schedule for scanning a rule:

   a. Click the **Configure** link for the rule.

   b. Modify the date, time to start the rule, timezone, or the frequency.

   c. Click **Save Rule**.

---

**Monitoring the results of Best Practice service scans**

Monitoring Best Practice rules

View the status of, troubleshoot, and configure Best Practice Rules from the details page. Within each category, the rule name, importance level, current status of the last scan, and schedule are shown. The **Configure** and **Turn rule off** links enable fast access to configuring rules or turning rules off and on.

The total number of active rules that have passed or failed displays at the top right. Clicking the Pass or Fail number filters the rules to display only those that have either passed or failed.

Click **Expand All** to view the all of the rules within each category. Click **Collapse All** to view only the Best Practice categories.
After an active rule has completed a scan based on its configured schedule, its status is displayed as either Passed if all nodes in the cluster successfully complied with the rule or Failed if one or more nodes did not pass. The active rule status includes:

- **Failed**: The rule did not pass the last scan. Investigate the failed rule.
- **Passed (with warning)**: The rule passed but due to most likely an agent connection issue, the rule could not run on one or more nodes within a cluster. Investigate the rules that pass with a warning. Be sure to check agent status (page 230).
- **Passed (no issues)**: The rule ran successfully without any issues.

**Rule importance**

Rule importance categories include the following levels in order from most to least critical:

- High
- Medium
- Low
- Info (informational only)
Any rules that are deemed high importance should be investigated and resolved before those of lower priority.

Viewing rule status and logs

Click **Passed** or **Failed** to get more info on a rule. The window displays a description of the rule, the importance level of the rule, and the date and time that the last scan was run in the **Status** view. If there are failures, the window provides a detailed message about the rule failure, including what caused the rule to fail and how to correct the failure.

![Replication factor out of bounds](image)

To view log history of a rule, click **Logs**. Click a Run Time item to view its details.
Best Practice Rules Reference

Reference of available rules in the Best Practice Service organized in alphabetical order by each Advisor section.

Backup Advisor

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description/Recommendation</th>
<th>Importance</th>
<th>Scope</th>
<th>Interval (default)</th>
<th>Alert Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto Snapshot not enabled</td>
<td>Checks to make sure auto snapshot isn't turned off in production.</td>
<td>High</td>
<td>Node</td>
<td>Daily</td>
<td>Info</td>
</tr>
<tr>
<td>Rule</td>
<td>Description/Recommendation</td>
<td>Importance</td>
<td>Scope</td>
<td>Interval (default)</td>
<td>Alert Level</td>
</tr>
<tr>
<td>------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td><strong>Auto snapshot is not enabled and can lead to data loss on truncation or drop. Please update your cassandra.yaml to enable auto_snapshot and prevent data loss.</strong> <strong>Tip:</strong> Use LCM Config Profiles (<a href="#">page 594</a>) to enable auto_snapshot in the Snapshots section of cassandra.yaml. The auto_snapshot setting is enabled by default in LCM config profiles.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Commit Log Archiving has been turned off due to inconsistent settings for all nodes in the cluster.</strong></td>
<td>High</td>
<td>Node, Cluster</td>
<td>Hourly</td>
<td>Alert</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> This rule is available in OpsCenter versions 6.1 and later.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Commit Log Archiving is not enabled for all nodes within the cluster, which can result in data loss when performing a Point-in-Time restore. **Turn Commit Log Archiving on (<a href="#">page 384</a>) again so that all nodes in the cluster have the enabled setting consistent for Commit Log Archiving.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Config Advisor**

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description/Recommendation</th>
<th>Importance</th>
<th>Scope</th>
<th>Interval (default)</th>
<th>Alert Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Repair service not enabled</td>
<td>Verifies that the repair service is enabled.</td>
<td>High</td>
<td>Cluster</td>
<td>Daily</td>
<td>Info</td>
</tr>
<tr>
<td></td>
<td>Running regular repair ensures data consistency across a cluster. <strong>Enable the repair service (<a href="#">page 462</a>).</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rule</td>
<td>Description/Recommendation</td>
<td>Importance</td>
<td>Scope</td>
<td>Interval (default)</td>
<td>Alert Level</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Repair service not configured correctly</td>
<td>Verifies that the repair service is configured correctly for your cluster. For more information, see <a href="#">basic</a>, <a href="#">advanced</a>, and <a href="#">expert</a> repair configuration.</td>
<td>High</td>
<td>Cluster</td>
<td>Daily</td>
<td>Info</td>
</tr>
<tr>
<td>Security not enabled for DataStax agents</td>
<td>Checks that OpsCenter authentication is enabled in conjunction with SSL between daemon and agent.</td>
<td>High</td>
<td>Cluster</td>
<td>Daily</td>
<td>Alert</td>
</tr>
<tr>
<td>Swap space is enabled</td>
<td>Checks that you do not have swap space enabled on any node. Swap space should not be used in a production environment.</td>
<td>Medium</td>
<td>Node</td>
<td>Daily</td>
<td>Alert</td>
</tr>
<tr>
<td>Seed node configuration</td>
<td>In each DC, there should be at least two seed nodes present, if there are at least two nodes present in the DC. IPs should be used rather than hostnames. All nodes should have the same seed list.</td>
<td>Low</td>
<td>Node, Cluster</td>
<td>Daily</td>
<td>Alert</td>
</tr>
</tbody>
</table>
### Network Advisor

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description/Recommendation</th>
<th>Importance</th>
<th>Scope</th>
<th>Interval (default)</th>
<th>Alert Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>To correct this, please use the same seed list of IPs on all nodes. <strong>Tip:</strong> If using LCM, adjust the [seed nodes](page 563) in the appropriate [LCM Config Profile](page 594).</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** When the `listen_address` field in `cassandra.yaml` file is left blank, OpsCenter agents default to the same listen address as DSE in OpsCenter version 6.1.2 and later.

### OpsCenter Config Advisor

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description/Recommendation</th>
<th>Importance</th>
<th>Scope</th>
<th>Interval (default)</th>
<th>Alert Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpsCenter Failover Enabled</td>
<td>DataStax recommends configuring OpsCenter failover for high availability.</td>
<td>Low</td>
<td>OpsC</td>
<td>Daily</td>
<td>Alert</td>
</tr>
</tbody>
</table>
### DSE Management Services

#### Rule

<table>
<thead>
<tr>
<th>Description/Recommendation</th>
<th>Importance</th>
<th>Scope</th>
<th>Interval</th>
<th>Alert Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>There is no backup OpsCenter configured. Please enable failover (page 166) for OpsCenter.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### OS Advisor

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description/Recommendation</th>
<th>Importance</th>
<th>Scope</th>
<th>Interval</th>
<th>Alert Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clocks in cluster out of sync</td>
<td>Checks that clocks across the cluster are in sync within a 2 second tolerance. The total drift across cluster exceeds the tolerance of 2 seconds; please sync clocks on your nodes. <strong>Warning</strong>: Clock drift can cause issues when LCM attempts to generate SSL certificates (page 579). Keeping clocks synchronized is critical to ensure accurate timestamps for database operations and logging.</td>
<td>High</td>
<td>Node, Cluster</td>
<td>Daily</td>
<td>Alert</td>
</tr>
<tr>
<td>Cassandra-user and agent-user match</td>
<td>Checks that cassandra and agent are run as the same user. Cassandra and agent are not run as the same user. Please ensure that Cassandra and agent are run as the same user.</td>
<td>High</td>
<td>Node</td>
<td>Daily</td>
<td>Alert</td>
</tr>
<tr>
<td>Clocks in UTC</td>
<td>Checks that clocks across the nodes are in Coordinated Universal Time (UTC). All the nodes are not in Coordinated Universal Time (UTC). Please ensure that all nodes are in UTC.</td>
<td>Low</td>
<td>Node</td>
<td>Daily</td>
<td>Alert</td>
</tr>
<tr>
<td>Rule</td>
<td>Description/Recommendation</td>
<td>Importance</td>
<td>Scope</td>
<td>Interval (default)</td>
<td>Alert Level</td>
</tr>
<tr>
<td>-------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------</td>
<td>-------</td>
<td>--------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Require Oracle Java</td>
<td>Checks to make sure that Oracle Java is being used on the node. Unsupported JDK is in use on the node. Oracle/Sun Hotspot JDK is the preferred JDK to use and well-tested in DataStax Enterprise. Switch to Oracle Hotspot JDK if you're currently using OpenJDK (as the default Java environment coming from the Linux OS). Tip: Use LCM Config Profiles to manage Java installations <em>(page 618)</em>.</td>
<td>Medium</td>
<td>Node</td>
<td>Daily</td>
<td>Alert</td>
</tr>
</tbody>
</table>

**Performance Advisor**

Rules for read and write to node performance (Performance Advisor not to be confused with the Performance Services *(page 521)*).

**Tip:** Use LCM Config Profiles *(page 594)* to adjust request timeout settings in cassandra.yaml settings and run a configuration job *(page 572)*.

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description/Recommendation</th>
<th>Importance</th>
<th>Scope</th>
<th>Interval (default)</th>
<th>Alert Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read request timeout not optimal</td>
<td>Checks that the read request timeout on your nodes is not set above recommended values. Significantly increasing the read request timeout on your nodes is not recommended. Please update cassandra.yaml on your nodes and lower the value of read_request_timeout_in_ms. Tip: Set the value in the Timeouts pane of an LCM Config Profile <em>(page 594)</em> and run a configure job <em>(page 576)</em>.</td>
<td>Medium</td>
<td>Node</td>
<td>Daily</td>
<td>Alert</td>
</tr>
<tr>
<td>Rule</td>
<td>Description/Recommendation</td>
<td>Importance</td>
<td>Scope</td>
<td>Interval (default)</td>
<td>Alert Level</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------</td>
<td>-------</td>
<td>--------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Write request timeout not optimal</td>
<td>Checks that the write request timeout on your nodes is not set above recommended values.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Significantly increasing the write request timeout on your nodes is not recommended. Please update cassandra.yaml on your nodes and lower the value of write_request_timeout_in_ms.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Tip:</strong> Set the value in the <strong>Timeouts</strong> pane of an LCM Config Profile <em>(page 594)</em> and run a configure job <em>(page 576)</em>.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Range request timeout not optimal</td>
<td>Checks that the range request timeout on your nodes is not set above recommended values.</td>
<td>Medium</td>
<td>Node</td>
<td>Daily</td>
<td>Alert</td>
</tr>
<tr>
<td></td>
<td>Significantly increasing the range request timeout on your nodes is not recommended. Please update cassandra.yaml on your nodes and lower the value of range_request_timeout_in_ms.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Tip:</strong> Set the value in the <strong>Timeouts</strong> pane of an LCM Config Profile <em>(page 594)</em> and run a configure job <em>(page 576)</em>.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Performance Service - Slow Queries Advisor**

For more information, see Slow Queries *(page 528)* in the Performance Service *(page 521)*.
<table>
<thead>
<tr>
<th>Rule</th>
<th>Description/Recommendation</th>
<th>Importance</th>
<th>Scope</th>
<th>Interval (default)</th>
<th>Alert Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use prepared statements</td>
<td>Prepared statements reduce the workload on the coordinator by removing the overhead of parsing the query. Use prepared statements for your queries.</td>
<td>Medium</td>
<td>Cluster</td>
<td>Hourly</td>
<td>Info</td>
</tr>
<tr>
<td>Avoid ALLOW FILTERING</td>
<td>Checks that ALLOW FILTERING is not used in queries. ALLOW FILTERING causes a query to scan all data within a token range, which might be desired with analytic workloads but is not recommended for non-analytic workloads. ALLOW FILTERING can cause long running queries and consume excessive system resources. If using ALLOW FILTERING outside of an analytics workload, please consider a new data model based on the query pattern instead.</td>
<td>Medium</td>
<td>Cluster</td>
<td>Hourly</td>
<td>Info</td>
</tr>
<tr>
<td>Avoid using large batches</td>
<td>Using large batches seems like an optimization but doing so puts extra load on the coordinator, which can cause hotspots in the cluster. Queries run faster after breaking large batches into individual queries and distributing them to different nodes. Break the batches into individual queries and distribute them to different nodes.</td>
<td>Medium</td>
<td>Cluster</td>
<td>Hourly</td>
<td>Info</td>
</tr>
<tr>
<td>Use counter instead of count</td>
<td>A count(*) query can be expensive, even with smaller limits. Replace the logic with a counter you maintain.</td>
<td>Medium</td>
<td>Cluster</td>
<td>Hourly</td>
<td>Info</td>
</tr>
<tr>
<td>Rule</td>
<td>Description/Recommendation</td>
<td>Importance</td>
<td>Scope</td>
<td>Interval (default)</td>
<td>Alert Level</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------------------------------------------------------------------------------------</td>
<td>------------</td>
<td>--------</td>
<td>-------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Minimize keys in IN clause</td>
<td>Huge IN clauses give the impression of a singular query but the clauses actually execute as multiple queries. Make individual async queries distributed amongst more coordinators.</td>
<td>Medium</td>
<td>Cluster</td>
<td>Hourly</td>
<td>Info</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance Service - Table Metrics Advisor</td>
<td>For more information, see <strong>Table Metrics (page 535)</strong> in the <strong>Performance Service (page 521)</strong>.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wide partitions</td>
<td>Checks for excessively wide partitions. Excessively wide partitions have a negative impact on performance and are not recommended. A partition is considered to be wide when the size is greater than 100 MB. Excessively wide partitions have a negative impact on performance and are not recommended. Consider remodeling your data to break up wide partitions.</td>
<td>Low</td>
<td>Node, Cluster</td>
<td>Hourly</td>
<td>Alert</td>
</tr>
<tr>
<td>Secondary indexes cardinality</td>
<td>Checks for secondary indexes with too many distinct values. High-cardinality secondary indexes can have a negative impact on system performance. Consider denormalizing the indexed data.</td>
<td>Low</td>
<td>Node, Cluster</td>
<td>Hourly</td>
<td>Alert</td>
</tr>
<tr>
<td>Tombstone count</td>
<td>Number of tombstones processed during reads. Too many tombstones can cause a degradation of performance. This can even lead to query failures.</td>
<td>Low</td>
<td>Node, Cluster</td>
<td>Hourly</td>
<td>Alert</td>
</tr>
<tr>
<td>Rule</td>
<td>Description/Recommendation</td>
<td>Importance</td>
<td>Scope</td>
<td>Interval (default)</td>
<td>Alert Level</td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------</td>
<td>---------</td>
<td>-------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Compaction Strategy</td>
<td>The compaction strategy you use should be based on your data and environment. This Best Practice rule is set to run so that you are aware of the importance of choosing a compaction strategy. If you have already chosen the correct compaction strategy based on your environment, please disable this rule if you do not want to see a reminder about compaction strategy again. Choose the compaction strategy that best fits your data and environment. See Compaction strategies.</td>
<td>Low</td>
<td>Cluster</td>
<td>Hourly</td>
<td>Alert</td>
</tr>
</tbody>
</table>

Performance Service - Thread Pools Advisor

For more information, see Thread Pool Statistics *(page 542)* in the Performance Service *(page 521)*.

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description/Recommendation</th>
<th>Importance</th>
<th>Scope</th>
<th>Interval (default)</th>
<th>Alert Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read Stage</td>
<td>Number of pending reads. Too many pending reads, which could be related to disk problems, poor tuning, or cluster overload. Consider adding new nodes, tuning the system, and revisiting your data model. If not CPU or IO bound, try increasing concurrent_reads.</td>
<td>Low</td>
<td>Node</td>
<td>Hourly</td>
<td>Alert</td>
</tr>
<tr>
<td>Mutation Stage</td>
<td>Number of pending mutations. Too many pending mutations; which could be related to disk problems, poor tuning, or cluster overload. Please consider adding new nodes, tuning the system, and revisiting your data model. If not CPU or IO bound, try increasing concurrent_writes.</td>
<td>Low</td>
<td>Node</td>
<td>Hourly</td>
<td>Alert</td>
</tr>
<tr>
<td>Rule</td>
<td>Description/Recommendation</td>
<td>Importance</td>
<td>Scope</td>
<td>Interval (default)</td>
<td>Alert Level</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------------------------------------------------------------------</td>
<td>------------</td>
<td>-------</td>
<td>--------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>ReplicateOnWriteStress</td>
<td>Be careful when using CL.ONE counter increments because it has an async task, which involves a read, kicked off to run after the increment is completed. Too many processes in this pool will begin to block writes. Reduce the use of CL.ONE counter increments or upgrade to Cassandra 2.1 or higher.</td>
<td>Medium</td>
<td>Node</td>
<td>Hourly</td>
<td>Info</td>
</tr>
<tr>
<td>Replication factor out of bounds</td>
<td>Checks that your cluster does not have a replication factor higher than it can support. Lists keyspaces that have a total RF higher than the number of nodes. Please update the replication factor for the appropriate keyspaces, or add additional nodes to your cluster.</td>
<td>Info</td>
<td>Cluster</td>
<td>Daily</td>
<td>Info</td>
</tr>
<tr>
<td>SimpleSnitch usage found</td>
<td>Checks to make sure SimpleSnitch isn't used in production. SimpleSnitch is not recommended for production clusters because it does not recognize datacenter or rack information. Please update the snitch to a topology-enabled snitch.</td>
<td>Medium</td>
<td>Node</td>
<td>Daily</td>
<td>Info</td>
</tr>
<tr>
<td>SimpleStrategy keyspace usage found</td>
<td>Checks that you are not using SimpleStrategy for any keyspaces in a multi-datacenter environment.</td>
<td>Medium</td>
<td>Cluster</td>
<td>Daily</td>
<td>Alert</td>
</tr>
</tbody>
</table>
## Search Advisor

Advice for Solr search nodes. For more information, see [DSE Search](#).

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description/Recommendation</th>
<th>Importance</th>
<th>Scope</th>
<th>Interval (default)</th>
<th>Alert Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vnodes enabled on Search nodes</td>
<td>Checks that vnodes are not in use on DataStax Enterprise search nodes for version 4.8 and below, or checks that there are either 16 or 32 vnodes on DataStax Enterprise search nodes for version 5.0 and above.</td>
<td>High</td>
<td>Node</td>
<td>Daily</td>
<td>Alert</td>
</tr>
<tr>
<td>Search nodes enabled with bad autocommit</td>
<td>Checks to see if a running Solr node has autocommit within 5-10 seconds.</td>
<td>Medium</td>
<td>Cluster</td>
<td>Daily</td>
<td>Alert</td>
</tr>
<tr>
<td>Search nodes enabled with query result cache</td>
<td>Checks to see if a running Solr node has query result cache disabled.</td>
<td>Medium</td>
<td>Cluster</td>
<td>Daily</td>
<td>Alert</td>
</tr>
</tbody>
</table>

Please update the replication strategies ([page 179](#)) of the relevant keyspace(s) to use NetworkTopologyStrategy.
<table>
<thead>
<tr>
<th>Rule</th>
<th>Description/Recommendation</th>
<th>Importance</th>
<th>Scope</th>
<th>Interval (default)</th>
<th>Alert Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search nodes with bad filter cache size</td>
<td>Checks to see if filter cache size is optimized for a running Solr node. Please modify your filter cache size attribute to 128 if using solr.LRUCache. Otherwise, if using solr.search.SolrFilterCache, modify the highWaterMarkMB attribute to 256.</td>
<td>Medium</td>
<td>Cluster</td>
<td>Daily</td>
<td>Alert</td>
</tr>
<tr>
<td>Search nodes enabled with row cache</td>
<td>Checks to see if a Solr node has row cache enabled. For optimizing memory use for DSE search with Solr, the row cache should be disabled. Edit the cassandra.yaml file and disable the row cache. <strong>Tip:</strong> If using LCM, adjust the value in the Caches pane of cassandra.yaml in the appropriate LCM Config Profile (page 594) and run a configure job (page 576).</td>
<td>Medium</td>
<td>Node</td>
<td>Daily</td>
<td>Alert</td>
</tr>
<tr>
<td>Rule</td>
<td>Description/Recommendation</td>
<td>Importance</td>
<td>Scope</td>
<td>Interval (default)</td>
<td>Alert Level</td>
</tr>
<tr>
<td>------</td>
<td>-----------------------------</td>
<td>------------</td>
<td>-------</td>
<td>-------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Search nodes have default key cache size</td>
<td>Checks to see if a Solr node has key cache set to default size. For optimizing memory use for DSE search with Solr, the key cache size should be set to its default size. Edit the cassandra.yaml file and ensure the key cache size is set to the recommended default size. <strong>Tip</strong>: If using LCM, adjust the value in the Caches pane of cassandra.yaml in the appropriate LCM Config Profile (page 594) and run a configure job (page 576).</td>
<td>Medium</td>
<td>Node</td>
<td>Daily</td>
<td>Alert</td>
</tr>
<tr>
<td>Search nodes have improper heap size</td>
<td>Checks to see if a Solr node has enough heap space. For optimizing memory use for DSE search with Solr, the heap should be set to at least 14GB. Set the Solr node max heap to at least 14GB.</td>
<td>Medium</td>
<td>Node</td>
<td>Daily</td>
<td>Alert</td>
</tr>
</tbody>
</table>

**Security Advisor**

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description/Recommendation</th>
<th>Importance</th>
<th>Scope</th>
<th>Interval (default)</th>
<th>Alert Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security keyspace not properly replicated</td>
<td>Checks that the auth keyspace is replicated correctly when using PasswordAuthenticator. Please increase the replication of the system_auth keyspace.</td>
<td>High</td>
<td>Node, Cluster</td>
<td>Daily</td>
<td>Alert</td>
</tr>
<tr>
<td>Security superuser has default setting</td>
<td>Checks that the default cassandra superuser and password has been changed from the default.</td>
<td>High</td>
<td>Cluster</td>
<td>Daily</td>
<td>Alert</td>
</tr>
<tr>
<td>Rule</td>
<td>Description/Recommendation</td>
<td>Importance</td>
<td>Scope</td>
<td>Interval (default)</td>
<td>Alert Level</td>
</tr>
<tr>
<td>------</td>
<td>-----------------------------</td>
<td>------------</td>
<td>-------</td>
<td>--------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Security superuser has default setting. Please update the password for the user ‘cassandra’.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Improper Security authentication setting</td>
<td>Checks that the cassandra authentication is enabled and not set to AllowAllAuthenticator. AllowAllAuthenticator performs no security checks and is not recommended. Please update cassandra.yaml on your nodes and change authenticator from org.apache.cassandra.auth.AllowAllAuthenticator to org.apache.cassandra.auth.PasswordAuthenticator. <strong>Tip:</strong> Change the authenticator in the <strong>Security</strong> pane of cassandra.yaml in the appropriate LCM Config Profile (page 594).</td>
<td>Medium</td>
<td>Node</td>
<td>Daily</td>
<td>Alert</td>
</tr>
<tr>
<td>Incorrect OpsCenter authentication setting</td>
<td>Checks that the OpsCenter authentication is not set to the default if you are using DatastaxEnterpriseAuth. Please change the default password of the admin user for OpsCenter authentication (page 101).</td>
<td>High</td>
<td>Cluster</td>
<td>Daily</td>
<td>Alert</td>
</tr>
<tr>
<td>Sensitive Config Value Encryption</td>
<td>It is recommended to enable encryption of sensitive config values in cassandra.yaml.</td>
<td>Medium</td>
<td>Node</td>
<td>Daily</td>
<td>Info</td>
</tr>
<tr>
<td>Rule</td>
<td>Description/Recommendation</td>
<td>Importance</td>
<td>Scope</td>
<td>Interval (default)</td>
<td>Alert Level</td>
</tr>
<tr>
<td>------</td>
<td>-----------------------------</td>
<td>------------</td>
<td>-------</td>
<td>-------------------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td>Config value encryption is not enabled. The rule failed on the following nodes: listed failed nodes. In dse.yaml, set config_encryption_active to true and use dsetool encryptconfigvalue to create encrypted config values for the sensitive fields. For more information, see config_encryption_active and Transparent data encryption. <strong>Tip:</strong> If using LCM, adjust the dse.yaml in the Encryption settings pane of the appropriate LCM Config Profile <em>(page 594).</em></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Performance Service**

**Performance Service Overview**

The OpsCenter Performance Service provides visual monitoring of diagnostics collected through the DSE Performance Service, displays alerts, and provides recommendations for optimizing cluster performance. The OpsCenter Performance Service combines OpsCenter metrics with CQL-based diagnostic tables populated by the DSE Performance Service to help understand, tune, and optimize cluster performance.

Performance Service overview panels

Enable the performance metrics panels and visually analyze the results within OpsCenter. Currently, the performance overview panels include:

- **Slow Queries** *(page 528)*: Identify slow queries on a cluster to easily find and tune poorly performing queries.
- **Table Metrics** *(page 535)*: Displays metrics to discover and diagnose poorly performing tables.
- **Thread Pool Statistics** *(page 542)*: Displays information about Cassandra system-level details such as thread pools.
Each panel contains a read-only high-level summary with pertinent sortable columns. Clicking on the title bar of a panel opens its expanded performance page:

Alerts panel

On each performance page in OpsCenter, an alerts panel feed displays triggered alerts. Any alerts you configure specifically for your environment appear in the panel.

**Note:** Alerts for the OpsCenter Performance Service are not pre-configured.

For guidance on configuring alerts for Performance Service, refer to the sample setup scenario sections:
• Slow query alert setup example (page 533)
• Table metrics alert setup example (page 539)
• Thread pool statistics alert setup example (page 545)

Recommendations panel

On each performance page in OpsCenter, a panel provides recommendations to rectify and improve performance issues.

Best Practice Service - Performance Service Advisors

The OpsCenter Performance Service integrates with the Best Practice Service (page 502). Any rules that fail send an alert to the relevant Alerts panel in each performance page, and generate suggestions to remediate issues in the Recommendations panels. The Best Practice Service monitors rules configured for the Performance Service Advisors:

• Performance Service - Slow Queries Advisor (page 534)
• Performance Service - Table Stats Advisor (page 540)
• Performance Service - Thread Pools Advisor (page 546)
By default, all rules are enabled and scheduled to run every hour. Adjust the rule configuration (page 502) to suit the requirements of your environment.

**Why use the OpsCenter Performance Service?**

The OpsCenter Performance Service expedites investigating and troubleshooting performance issues with your clusters. While valuable information is available using only the DSE Performance Service, the diagnostic information must be accessed from a command line, which can be more time-consuming than viewing a visual presentation of performance indicators automatically extracted and presented in a graphical fashion. Moreover, the onus to interpreting the information and deciding on remedial action is solely on the administrator.

The OpsCenter Performance Service works in conjunction with the DSE Performance Service, combining OpsCenter performance metrics (page 324) with CQL-based diagnostic tables populated by the DSE Performance Service. Context-specific recommendations on possible causes of and potential avenues to fixing performance issues greatly minimizes the amount of time spent manually troubleshooting performance problems. The Performance Service harnesses the power of its Best Practice Service (page 502) and Alerts (page 138) features to feed the Recommendations and Alerts panels in the performance pages.

In addition, using the OpsCenter Performance Service eliminates the need for custom scripting and scheduling to detect problem nodes.

Selectively enable and disable the Performance Service as necessary to accommodate testing and peak production times in your environment.

**Enabling the OpsCenter Performance Service**

Enable the OpsCenter Performance Service. Enabling each performance object also enables its default configuration. Disabling the OpsCenter Performance Service is recommended during peak production time.

**Note:** The Slow Query Log is enabled by default in the OpsCenter Performance Service based on the default settings in DataStax Enterprise (DSE).

**Prerequisites:** When enabling or disabling the OpsCenter Performance Service, the JMX interface is used to immediately toggle the Service. This behavior is the same as when using the `dsetool perf` command to temporarily change the running parameters for the DSE Performance Service.
For clusters not managed using Lifecycle Manager (LCM), OpsCenter programmatically sets a value in `dse.yaml`, which causes the file to lose comments and formatting. If there are comments and formatting you want to retain or refer to later, back up the `dse.yaml` file for safekeeping.

**Important:** If a cluster is being managed within LCM, OpsCenter does not automatically modify `dse.yaml` for the Performance Service. You must manually update the associated settings in the configuration profile (page 594) for the cluster in Lifecycle Manager (page 549).

1. Click **cluster name** > **Services**.

2. Click the **Configure** or **Details** link as appropriate for the Performance Service.
   The Overview page prompts you to enable metrics.

3. Click the **Enable metrics** link to view performance data.
   The Settings tab appears where you can enable and configure the performance objects.
4. Click the toggle in the Status column to **On** for the performance objects you want to enable.

   The performance objects can be turned on and used independently. Turning on a performance service object enables the associated settings in the dse.yaml file.

   A dialog prompts you to confirm. Toggling performance objects from off to on has the following effect:

   **Clusters not managed using LCM**

   For clusters not managed using LCM, changes are pushed to dse.yaml on every node, which overwrites any comments or formatting in the dse.yaml file. To retain your original dse.yaml, make a backup of the file as mentioned in the above prerequisite (**page 525**).

   **Clusters managed using LCM**

   For clusters managed using LCM, you must manually update any configuration profiles to persist changes to dse.yaml related to the performance service. See **Editing a configuration profile (page 603)**.

5. Click **Change** to proceed.

   No further configuration is required; however, you can adjust the default configuration parameters:

   - Configuring the slow query log (**page 529**)
   - Configuring table metrics (**page 535**)
Disabling the OpsCenter Performance Service

After tuning queries and optimizing the database with recommendations from the Performance Service, disable the OpsCenter Performance Service during peak production time.

1. Click \texttt{cluster name}\#Services.

2. Click the \texttt{Configure} link next to the Performance Service.

3. On the Settings tab, turn all performance objects to Off.

Setting permissions for the OpsCenter Performance Service

Optionally, set OpsCenter Performance Service permissions for each user role and cluster. Grant configuration and CQL tracing privileges. You must have the admin role to grant role privileges.

\textbf{Prerequisites:} Enabling authentication in OpsCenter (page 105)

1. Click \texttt{Settings}\#Users & Roles.
   
   The Users & Roles dialog appears.

2. Click \texttt{Manage Roles}.
   
   The Manage Roles dialog appears.

3. Select the edit icon for the role.
   
   The Edit Role dialog appears.
4. In the Services area, select the Performance Service permissions to grant to the role:
   - **Performance Service Configuration**: Enables configuration permissions for the OpsCenter Performance Service metrics pages, which allows Configuring the slow query log (page 529), Configuring table metrics (page 535), and Configuring thread pool statistics (page 542).
   - **Performance Service CQL Tracing**: Enables the Trace feature (page 531) for the Slow Query Log.

5. Click **Save**. If applicable, repeat for each cluster in your environment.

Related information:
*Configuring OpsCenter role-based security* [OpsCenter allows enabling user authentication, defining custom roles, managing users, and designating permissions.] (page 101)

**Tuning a database cluster with the Performance Service**

Tune a DataStax Enterprise cluster by following recommendations from the OpsCenter Performance Service.

**Slow Queries**

View, troubleshoot, and trace slow queries in the Slow Queries page. Examine the slow query log to identify and trace queries that take an excessive time to execute.
The query panel indicates when no slow queries are detected.

**Configuring the slow query log**

Configure the slow query log parameters in the Performance Service. Examine the slow query log to identify and track queries that take an excessive time to execute. Slow queries are candidates for performance optimization. When the slow query log is enabled, OpsCenter records any queries that take longer than the allotted threshold.

**Prerequisites:**

- **Note:** Review and if warranted, update the default replication for `dse_perf` keyspace. The default value might need to be increased.

1. Click `cluster name` #Services.

2. Click the **Configure** or **Details** link as appropriate for the Performance Service.

3. Click the **Settings** tab.

4. Click the **Configure** link for the Slow Query Log.
   The Configure Slow Query Log dialog appears.
5. Click the button to the On position.

6. Enter a Threshold value to override the default. Queries that take longer than the allotted threshold value are recorded in the slow query log.
   
   To prevent excess overhead, the threshold must be higher than 15 ms.

7. Enter a TTL in Time To Live to override the default. The TTL indicates how long the recorded slow query should stay in Cassandra in seconds.

8. Click Save.

**Viewing slow queries**

View any slow queries along with alerts and recommendations for improving query performance. Any CQL tables with queries that take longer than the configured threshold appear in the Slow Queries list. System and OpsCenter keyspaces are excluded from the Slow Queries list.

**Note:** The ability to trace queries (page 531) requires permission (page 527) granted in your user role.

Version-specific information:

- The User column displays the users who are running slow queries for OpsCenter versions 6.0.9 and later.
- When viewing slow queries for DSE versions 5.1 and later, slow queries are only shown for data since the last time the agent was restarted. For versions of DSE earlier than 5.1, all historical slow query data is available for slow queries. For more information, see the known issue (page 52) and referenced KB article.
1. Click `cluster name#Services`.

2. Click the **Configure** link for the Performance Service.
   The Overview tab displays the performance panels *(page 521)*.

3. Click the title bar of the **Slow Queries** panel.
   The Slow Queries performance details page appears.

4. Sort the slow queries by any column header.

**Tracing slow queries**

Trace slow queries to view the stages in a query and determine where performance bottlenecks occur.

**Prerequisites:**

Access to the trace query feature in OpsCenter requires permission granted *(page 527)* to a user role. Those users with the admin role have full privileges.

**Warning:** When initially accessing the Slow Query console, a warning dialog informs you of potential consequences to data or performance when tracing queries. Review the information, select **I Understand** and click **Close** to continue.
1. Click $\textit{cluster name}$\#Services.
2. Click the \textit{Configure} link for the Performance Service.
   The Overview tab displays the \textit{performance panels} (\textit{page 521}).
3. Click the Slow Queries panel.
4. Select a query from the Slow Queries list.
   The CQL query appears with syntax highlighting in the query box. Selecting a query from the list automatically selects the keyspace for the query.
5. Click \textit{Trace}.
   The query executes and displays each stage in the query, including timestamp and elapsed time in microseconds. To view the longest running stages of the query, sort the query by the \textit{Elapsed Time} column. IP addresses for any nodes in the Source list are highlighted in red.
6. To make adjustments to the query, select an option from the \textit{Consistency} or \textit{Coordinator} lists and experiment with running additional traces.
   Generally, selecting a keyspace is not applicable to tracing a query selected from the Slow Queries list because that was already done for that query. If you have tables named the same but located in different keyspaces, select the \textit{Keyspace} to trace. The Slow Queries excludes system or OpsCenter keyspaces.
7. To run an ad hoc query:
   a. Enter the CQL statement directly in the query box.
b. Make any selections from the **Consistency**, **Keyspace**, or **Coordinator** lists.

c. Click **Trace**.

**Example alert setup scenario for slow queries**

Follow these guidelines to configure latency alerts for monitoring slow queries. Any alerts triggered for read and write latencies appear in the Slow Queries Alerts panel within the Slow Queries performance page. Some suggested basic alerts to configure include:

- **Write Request Latency** - above 50 ms/op for more than one minute; also for more than 5 minutes
- **Read Request Latency** - above 50 ms/op for more than one minute; also for more than 5 minutes
- **Local Write Latency** - above 50 ms/op for more than one minute; also for more than 5 minutes

Local latency alerts are available from **Advanced#Tables**. Configure the pertinent latency alerts of interest with thresholds appropriate for your environment.

1. Click the **Alerts** menu.

2. In the Active Alerts dialog, click **Manage Alerts**.
   
   The Add Alert dialog appears.

3. In the **Notify me when** menu, choose a **r/w latency metric name**.
   
   Click **Advanced#Tables** to set up table latency alerts.

4. Select the alert notification criteria:
   
   - Select the nodes to monitor for cluster-level alerts.
   - Select the table for local latency alerts.
5. Click **Save Alert**.
   The alert displays in the Manage Alert Rules list. Repeat these steps for each additional alert.

### Configuring rules for the slow queries advisor

Configure rules for the Performance Service - Slow Queries Advisor in the Best Practice Service. The rules generate recommendations for the OpsCenter Performance Service. By default, all rules are enabled and scheduled to run every hour.

1. Click `cluster name`\#Services.

2. Click the Details link for the Best Practice Service.

3. Click to expand the **Performance Service - Slow Queries Advisor** panel.

4. Click the **Configure** link for the rule you want to adjust.
5. Click **Save Rule**. Repeat these steps for each rule you want to adjust.

### Slow query recommendations

Recommendations that appear for assistance with resolving slow query issues are described in the table. The recommendations appear in the Recommendations panel when the Performance Service detects known issues in a CQL query statement.

<table>
<thead>
<tr>
<th>Recommendation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avoid ALLOW FILTERING</td>
<td>Checks that ALLOW FILTERING is not used in queries. Please consider a new data model based on the query pattern instead of using ALLOW FILTERING.</td>
</tr>
<tr>
<td>Avoid using large batches</td>
<td>Using large batches seems like an optimization but doing so puts extra load on the coordinator, which can cause cluster hotspots in the cluster. Queries run faster after breaking large batches into individual queries and distributing them to different nodes.</td>
</tr>
<tr>
<td>Minimize keys used within the IN clause</td>
<td>Huge IN clauses can give the impression of a singular query but the clauses actually execute as multiple queries. Make individual async queries distributed amongst more coordinators.</td>
</tr>
<tr>
<td>Use counter instead of count</td>
<td>A count(*) query can be expensive, even with smaller limits. Replace the logic with a counter you maintain.</td>
</tr>
<tr>
<td>Use prepared statements for your queries</td>
<td>Prepared statements reduce the workload on the coordinator by removing the overhead of parsing the query.</td>
</tr>
</tbody>
</table>

### Table Metrics

View table performance and potential issues at a glance. View table metrics charts for one or more nodes.

#### Configuring table metrics

Enable and configure the table histogram table options.

1. Click **cluster name**Services.

2. Click the **Configure** or **Details** link as appropriate for the Performance Service.

3. Click the **Settings** tab.
4. Click the **Configure** link for Table Metrics.

The Configure Table Metrics Options dialog appears.

5. Enter a **Refresh Rate** in seconds that specifies how often DataStax Enterprise updates info in dse_perf tables. Setting the refresh rate to 60 seconds or greater is recommended to match the minimum default frequency at which OpsCenter collects and aggregates metrics data.

6. Enter a **Retention Count** value that specifies the number of snapshots retained at any given time.
7. Click **Save**.

**Viewing table metrics**

View concentrated table metrics and charts thereof, along with alerts and recommendations for improving table performance. Table (column family) metrics are pre-populated in the lower half of the Table Metrics Performance page.

1. Click **cluster name#Services**.

2. Click the **Configure** link for the Performance Service.
   
   The Overview tab displays the **performance panels (page 521)**.

3. Click the title bar of the **Table Metrics** panel.

![Table Metrics](image)

The Table Metrics performance page appears.
4. Select an option from the **Table Details For** list. The Table Metrics view updates and highlights the associated table row.

5. The Table Metrics performance details page provides multiple viewing options:
   - Date and time range: Select the date and time and click **Update**. Click **Current** to view real-time data.
• Time Range: Select the granularity for the data that displays in the metrics charts. Available options are by 20 minutes, Hour, Day, Week, or Month.

• By default, all nodes are shown. Select a specific node from the Node list. Select a few nodes to view a visual side-by-side comparison.

Example alert setup scenario for table metrics

Follow these guidelines to configure alerts for monitoring table performance. Any triggered alerts appear in the Table Metrics Alerts panel within the Table Metrics performance page. Some suggested basic alerts to enable and configure include:

• Any alerts on the Table (formerly Column Family) metrics shown in the charts area of the Table Metrics performance page:
  # TBL: Local Read Latency
  # TBL: Local Write Latency
  # TBL: Total Disk Used
  # TBL: Cell Count
  # TBL: Partition Size
  # TBL: SSTables per Read
  # TBL: SSTable Count

• Cluster-level metrics shown in the charts area of the Table Metrics performance page:
  # Write Request Latency - above 50 ms/op for more than one minute; also for more than 5 minutes
  # Read Request Latency - above 50 ms/op for more than one minute; also for more than 5 minutes

Table and local latency alerts are available from Advanced#Tables. Configure alerts of interest with thresholds appropriate for your environment.

1. Click the Alerts menu.

2. In the Active Alerts dialog, click Manage Alerts.
   The Add Alert dialog appears.
3. In the **Notify me when** menu, click **Advanced#Tables#TBL: metric name**.

4. Select the alert notification criteria:
   - Select the table for local latency alerts.
   - Select the nodes to monitor for cluster-level alerts.

![Add Alert Dialogue Box](image.png)

5. Click **Save Alert**.
   The alert displays in the Manage Alert Rules list. Repeat these steps for each additional alert.

### Configuring rules for the table metrics advisor

Configure rules for the Performance Service - Table Metrics Advisor in the Best Practice Service. The rules generate recommendations for the OpsCenter Performance Service. By default, all rules are enabled and scheduled to run every hour.

1. Click **cluster name#Services**.

2. Click the **Details** link for the Best Practice Service.

3. Click to expand the **Performance Service - Table Metrics Advisor** panel.

![Performance Service - Table Metrics Advisor Table](image.png)

4. Click the **Configure** link for the rule you want to adjust.
5. Click **Save Rule**. Repeat these steps for each rule you want to adjust.

**Tables metrics recommendations**

Recommendations for assistance with troubleshooting table performance issues appear in the Recommendations panel when the Performance Service detects known issues in the table metrics.

<table>
<thead>
<tr>
<th>Recommendation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compaction Strategy</td>
<td>Choose the compaction strategy that best fits your data and environment. Learn more about compaction strategies. The compaction strategy you use should be based on your data and environment. This Best Practice rule is set to run so that you are aware of the importance of choosing a compaction strategy. If you have already chosen the correct compaction strategy based on your environment, please disable this rule (page 502) if you do not want to see a reminder about compaction strategy again.</td>
</tr>
<tr>
<td>Secondary indexes cardinality</td>
<td>Checks for secondary indexes with too many distinct values. High cardinality secondary indexes can have a negative impact on system performance. Consider denormalizing the indexed data.</td>
</tr>
<tr>
<td>Tombstones count</td>
<td>Number of tombstones processed during reads. Too many tombstones can cause a degradation of performance, which can even lead to query failures. Consider updating your data model to decrease the amount of deleted data. For TTLed time series data, consider reducing gc_grace_seconds and evaluating different compaction strategies.</td>
</tr>
</tbody>
</table>
Wide partitions

Checks for excessively wide partitions. Excessively wide partitions have a negative impact on performance and are not recommended. Consider remodeling your data to break up wide partitions. A partition is considered to be wide when the size is greater than 100 MB.

### Thread Pool Statistics

View data for thread pool statistics at a high or detailed level. A node has multiple thread pools available to improve and manage memory consumption. Thread pools execute tasks in pending and active queues.

### Configuring thread pool statistics

Configure logging system information such as thread pool statistics in the `dse_perf` keyspace. The only option to configure is the refresh rate for updating the system information tables.

1. Click `cluster name#Services`.
2. Click the `Configure` or `Details` link as appropriate for the Performance Service.
3. Click the `Configure` link for Thread Pool Stats.
   
   The Configure System Info Options dialog appears.

4. Enter a Refresh Rate at which DataStax Enterprise updates the system information tables. Setting the refresh rate to 60 seconds or greater is recommended to match the minimum default frequency at which OpsCenter collects and aggregates metrics data.
5. Click **Save**.

### Viewing thread pool statistics

View details for thread pools statistics for a specific node or an entire cluster. Thread pool statistics are generally intended for tracking node rather than cluster activity. Thread pool alerts and any recommendations for improving cluster performance display in the respective panels. Specify the date and time range granularity to investigate the thread pool stats. The name of the thread pool statistic is displayed in the view, along with the Active, Pending, Completed, Blocked, and Total Blocked columns that indicate the counts for each queue. The Pending column is sortable in both the overview mini-panel and the main Thread Pools page. The Thread Pool Dropped Tasks and Pending Queues graphs provide a visualization of the metrics.

1. Click **cluster name** Services.

2. Click the Configure link for the Performance Service.

   The Overview tab displays the performance panels (*page 521*).

3. Click the title bar of the **Thread Pools** panel.

![Thread Pools Table](image)

The Thread Pools performance details page appears.
4. The Thread Pools page provides multiple options for viewing thread pool data:

- **Date and time range for data:** Select the date and time and click **Update**. Click **Current** to view real-time data.

- **Time Range:** Select the granularity for the data that displays in the thread pools statistics table. Available options are by 20 minutes, Hour, Day, Week, or Month. The selected granularity affects the view and the options available in the Slider Step list.

- **Select the increment by which to move the slider bar for viewing the range of data over time from the Slider Step list.** Click on the bar or drag to dynamically change the view of the statistics activities.

- **Select either a cluster-wide view or a specific node from the Node list.**
5. Click the magnifier option in the metrics panels to view expanded details. Change the date and time range or the Graph Scale granularity.

Example alert setup scenario for thread pool statistics

Follow these steps to access and configure TP statistic alerts. Some suggested basic alerts to configure include:

- TP: Dropped Counter Mutations - above 0 for more than 1 minute
- TP: Dropped Mutations - above 0 for more than 1 minute
- TP: Migrations Pending - above 2 for more than 1 minute

There are many TP statistic alerts available. Configure the alerts of interest with thresholds appropriate for your environment.

1. Click the Alerts menu.

2. In the Active Alerts dialog, click Manage Alerts.
   
   The Add Alert dialog appears.

3. In the Notify me when menu, choose Advanced#Cassandra#TP: metric name.
4. Select the alert notification criteria and the nodes to monitor.

5. Click **Save Alert**.

   The alert displays in the Manage Alert Rules list. Repeat these steps for each additional alert.

**Configuring rules for the thread pools advisor**

Configure rules for the Performance Service - Table Metrics Advisor in the Best Practice Service. The rules generate recommendations for the OpsCenter Performance Service. By default, all rules are enabled and scheduled to run every hour.

1. Click `cluster name#Services`.

2. Click the **Details** link for the Best Practice Service.
3. Click to expand the **Performance Service - Table Metrics Advisor** panel.

![Performance Service - Table Metrics Advisor panel]

4. Click the **Configure** link for the rule you want to adjust.

![Configure Rule: Mutation Stage]

5. Click **Save Rule**. Repeat these steps for each rule you want to adjust.

**Thread pool statistics recommendations**

Recommendations that appear for assistance with resolving thread pool issues are described in the table. The recommendations appear in the Recommendations panel when the Performance Service detects known issues in the CQL statement of a query.

<table>
<thead>
<tr>
<th>Recommendation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mutation stage</td>
<td>Number of pending mutations. Too many pending mutations, which could be related to disk problems, poor tuning, or cluster overload. Consider adding new nodes, tuning the system, and revisiting your data model. If not CPU or IO bound, try increasing concurrent_writes.</td>
</tr>
<tr>
<td>Pending reads</td>
<td>Number of pending reads. Too many pending reads, which could be related to disk problems, poor tuning, or cluster overload. Consider adding new nodes, tuning the system, and revisiting your data model. If not CPU or IO bound, try increasing concurrent_reads.</td>
</tr>
<tr>
<td>ReplicateOnWriteStage</td>
<td>Be careful when using CL.ONE counter increments because it has an async task, which also has a read in it, kicked off to run after the increment is completed. Too many processes in this pool will begin to block writes. Reduce the use of CL.ONE counter increments or upgrade to Cassandra 2.1 or higher.</td>
</tr>
</tbody>
</table>
Troubleshooting OpsCenter

Issues experienced with DataStax OpsCenter, including suggested solutions or workarounds.

See troubleshooting OpsCenter.
Lifecycle Manager

Lifecycle Manager overview

What is Lifecycle Manager?

Lifecycle Manager (LCM) is a powerful provisioning and configuration management system designed for ease of use with DataStax Enterprise (DSE) clusters. Graphical workflows enable efficient installation and configuration of DataStax Enterprise, empowering your organization to effectively manage DataStax Enterprise clusters without requiring extensive platform expertise.

The Lifecycle Manager cluster topology model allows completely defining the cluster configuration including datacenter and node topology. The LCM model works in conjunction with a flexible configurable profile system that integrates deeply with the full spectrum of DSE settings. The declarative model facilitates inheritance of configuration profiles and more when installing and configuring DataStax Enterprise at the cluster, datacenter, or node level:

- Efficiently monitor and prevent configuration drift by defining configuration profiles that apply to the cluster, datacenter, or node level. Enforce uniform configurations that adhere to the desired baseline configurations for the workload of each datacenter.
- Securely stored credentials automate access to machines and package repositories without the need to repeatedly enter credentials during installation and configuration jobs.

Monitor job status with unprecedented access and deep transparency into each recorded and timestamped step of the deploy process. Drill into job details to troubleshoot provisioning and configuring jobs from the convenience of the Jobs workspace (page 580) without the immediate need to scour various logs for information.

Note: The data (cluster topology models, configuration profiles, credentials, repositories, job history, and so forth) for Lifecycle Manager is stored in the lcm.db database. Your organization is responsible for backing up the lcm.db database. You must also configure failover (page 169) to mirror the lcm.db.

How many nodes can Lifecycle Manager support when creating DataStax Enterprise clusters?

Lifecycle Manager supports creating DataStax Enterprise clusters with up to 300 nodes. Larger clusters cannot be managed using the LCM web UI. The API can be used directly to attempt to manage larger clusters, provided sufficient heap memory is available.

Does OpsCenter and LCM support decommission of nodes?

OpsCenter supports decommissioning a node (page 256). LCM does not currently support node decommission.

Does OpsCenter and LCM support multi-instance nodes?

OpsCenter supports monitoring multi-instance nodes.
Lifecycle Manager

**Note:** Lifecycle Manager does not currently support managing DataStax Enterprise Multi-Instance nodes (also referred to as dense nodes).

Which snitches does LCM support?

At this time, only the GossipingPropertyFileSnitch (GPFS) is supported for managing or importing DataStax Enterprise clusters in Lifecycle Manager.

What partitioners does LCM support?

- Murmur3Partitioner
- RandomPartitioner

What operating systems does LCM support?

Lifecycle Manager runs on OpsCenter-supported Linux environments only.

**Note:** As of OpsCenter and Lifecycle Manager 6.1.3 and later, LCM automatically performs an OS supported platform check for the version of DSE being installed. For details, see [Supported OS platform check for DSE installs (page 573)](#).

lcm.db

The location of the Lifecycle Manager database lcm.db depends on the type of installation:

- **Package installations:** /var/lib/opscenter/lcm.db
- **Tarball installations:** install_location/lcm.db

**Note:** The data (cluster topology models, configuration profiles, credentials, repositories, job history, and so forth) for Lifecycle Manager is stored in the lcm.db database. Your organization is responsible for backing up the database. You must also configure failover (page 169) to mirror the lcm.db.

**Accessing OpsCenter Lifecycle Manager**

Navigate to the Lifecycle Manager application for installing and configuring DataStax Enterprise clusters. Return to OpsCenter Monitoring after provisioning clusters.

**Prerequisites:**

Lifecycle Manager performs provisioning and system configuration actions, which require secrets such as SSH credentials and DSE user passwords.

- Authentication is necessary for limiting access to LCM to authorized individuals only. See [Configuring OpsCenter role-based security (page 101)](#) for details enabling OpsCenter authentication.

  **Note:** Only OpsCenter users assigned an Admin role (page 101) have access to Lifecycle Manager.
• Enabling TLS is necessary to protect these secrets during network transit, see Configuring SSL/TLS between OpsCenter and the DataStax Agents (page 82) for details.

1. Access Lifecycle Manager:
   • When you first launch OpsCenter, the Welcome to DSE OpsCenter dialog appears. Click Create a new cluster. Lifecycle Manager launches in another browser window.

   ![Lifecycle Manager](image1)

   • Click Lifecycle Manager from the OpsCenter navigation menu.

     **Note:** Accessing Lifecycle Manager requires the Admin role if OpsCenter authentication (page 101) is enabled. If the Lifecycle Manager menu is gray and unavailable, contact your OpsCenter admin.

   ![Lifecycle Manager](image2)

   Lifecycle Manager launches in another browser window.

   ![Lifecycle Manager](image3)

   • Within OpsCenter Lifecycle Manager, click OpsCenter Monitoring to return to monitoring DSE clusters in DSE OpsCenter.
2. Should you happen to open Lifecycle Manager in multiple browser windows or tabs, a dialog appears.

To navigate to your desired location:

- Click the re-open Lifecycle Manager link to open Lifecycle Manager again.
- Click Reload the page to open OpsCenter monitoring.
- Click Go Back to return to the Welcome to DSE OpsCenter dialog.

Getting started workflow

Welcome to Lifecycle Manager! If you do not have any existing clusters in OpsCenter to manage in LCM, a getting started page appears:

Click the information (i) icon for each workspace title to view a summary about its purpose in LCM:
View a short video that demonstrates getting started:

The initial Lifecycle Manager workflow:

After creating SSH credentials \(^1\) \((\text{page } 553)\), defining configuration profiles \(^2\) \((\text{page } 553)\), adding a repository \(^3\) \((\text{page } 553)\), and building the declarative cluster topology model \(^4\) \((\text{page } 553)\), you are ready to run an install job to install and configure DataStax Enterprise (DSE) and monitor the install job \(^5\) \((\text{page } 553)\).

Follow the linked workflow steps to quickly get started working with Lifecycle Manager:

**Prerequisites:**

Bring your own instances on a supported platform.

Follow the linked workflow steps to quickly get started working with Lifecycle Manager:

1. Add SSH credentials \((\text{page } 591)\) so LCM can remotely log in to target machines when performing installation and configuration activities.

2. Add a configuration profile \((\text{page } 600)\) that defines the required DSE configuration on the development, test, and production clusters for your organization. If the configuration of a cluster is intentionally heterogeneous and not uniform at all levels, you can create multiple configuration profiles to apply individually at the cluster, datacenter, or node levels.

3. Add a repository \((\text{page } 587)\) so LCM can download the DSE software onto target machines from either the public DataStax repo or an internal install repo mirror set up by your organization.

4. Define the topology \((\text{page } 554)\) of the cluster:
   a. Add a cluster \((\text{page } 556)\).
   b. Add its datacenters \((\text{page } 560)\).
   c. Add its nodes \((\text{page } 563)\).

5. Run an installation job \((\text{page } 574)\) of DataStax Enterprise at the cluster level. Lifecycle Manager installs and configures DataStax Enterprise on all datacenters and nodes within the cluster. With Lifecycle Manager, there is deep transparency into the progress of each
installation and configuration job available when viewing job summary and details (page 580).

What's next:

As a cluster grows and configuration options for various workloads require adjustments, running a configure job applies the configuration profiles across the cluster topology. To update the configuration of a cluster, edit its configuration profile (page 603) and run a configuration job (page 576) at the cluster level. LCM efficiently deploys the configuration changes across the cluster without duplicating work already completed in previous jobs.

Lifecycle Manager: Clusters

Create and manage the DataStax Enterprise cluster topology model in the Clusters workspace of Lifecycle Manager. Run install and configure jobs at the cluster, datacenter, or node level.

Cluster topology overview

The Lifecycle Manager (LCM) topology model consists of cluster, datacenter, and node entities. The model facilitates installing and configuring DataStax Enterprise at the cluster, datacenter, and node levels. When installing and configuring DSE clusters, the model provides flexibility and powerful inheritance mechanisms.

LCM requires manually defining the cluster topology for new DataStax Enterprise clusters, or for clusters whose nodes do not use the same SSH credentials because the automatic cluster import (page 569) process requires using a singular SSH credential. Automatically importing the cluster for existing DSE clusters imports the cluster topology and constructs the cluster model entities on your behalf in the Clusters workspace. The logical LCM model should reflect the actual physical topology of a cluster.

If someone manually changes cluster topology without using LCM, you must update the logical topology model in LCM to reflect those physical changes. For instance, when a node is decommissioned in OpsCenter, you must manually delete the node in the model. It is not currently possible to decommission systems directly using LCM.

Note: If you neglect to update the corresponding LCM model, when running the next configure or install job, LCM attempts to restore the old topology, with unpredictable results.

When deleting (that is, ceasing to manage) an entity from the topology model in LCM, you are simply removing management of the entity from LCM. Deleting a cluster, datacenter, or node in LCM does not affect the physical systems. Deleting entities from the LCM topology model causes LCM to stop managing and ignore them. The physical cluster, datacenters, nodes, and the corresponding entities in OpsCenter are not affected.

Note: The data (cluster topology models, configuration profiles, credentials, repositories, job history, and so forth) for Lifecycle Manager is stored in the lcm.db
database. Your organization is responsible for backing up the lcm.db database. You must also configure failover (page 169) to mirror the lcm.db.

Lifecycle Manager: Clusters Workspace

Use the Clusters workspace to import existing clusters, provision new clusters, manage the cluster topology, and run configure jobs. Manually adding entities in the model must be performed in order:

1. **Add clusters (page 556)**
2. **Add datacenters (page 560)**
3. **Add nodes (page 563)**

Datacenters can inherit certain shared settings from a cluster. Nodes can inherit certain shared settings from a datacenter or a cluster.

The following graphic shows the fully expanded and populated Clusters, Datacenters, and Nodes panes prior to running an install and configure job:

![Lifecycle Manager: Clusters Workspace](image)

**Table 20: Topology status legend**

<table>
<thead>
<tr>
<th>Status</th>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not run</td>
<td><img src="image" alt="red flag" /></td>
<td>An install job has not been run on the topology entity. See running a job (page 572).</td>
</tr>
<tr>
<td>Import (unmanaged cluster)</td>
<td><img src="image" alt="red plus sign" /></td>
<td>An existing cluster is not being managed by Lifecycle Manager. Click <strong>Start Managing</strong> and follow instructions to <strong>automatically import (page 569)</strong> the cluster.</td>
</tr>
</tbody>
</table>
### Status

<table>
<thead>
<tr>
<th>Status</th>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Success</td>
<td>✅</td>
<td>The job ran successfully on a topology entity (cluster, datacenter, or node).</td>
</tr>
<tr>
<td>Failure</td>
<td>😞 (red universal no access symbol)</td>
<td>The job run on a topology entity (cluster, datacenter, or node) failed. Investigate the issue by drilling into the job details (page 582). Try running the job again.</td>
</tr>
</tbody>
</table>

### lcm.db

The location of the Lifecycle Manager database lcm.db depends on the type of installation:

- **Package installations:** /var/lib/opscenter/lcm.db
- **Tarball installations:** install_location/lcm.db

**Note:** The data (cluster topology models, configuration profiles, credentials, repositories, job history, and so forth) for Lifecycle Manager is stored in the lcm.db database. Your organization is responsible for backing up the database. You must also configure failover (page 169) to mirror the lcm.db.

### Adding a cluster

Add a cluster to centrally manage installs and configurations within Lifecycle Manager for DataStax Enterprise clusters. By default, all datacenters within a cluster inherit the configuration profile from the cluster level, unless overridden when creating the datacenter model. By default, all nodes within a datacenter inherit the configuration profile from the datacenter level, unless overridden when creating the node models. SSH settings can also be inherited or overridden at the datacenter or node levels. Repositories are only specified at the cluster level and are inherited by datacenters and nodes.

**Prerequisites:**

- Add a Configuration Profile (page 600)
- Add a Repository (page 587)
- Add SSH Credentials (page 591)

**Important:** When enabling node-to-node encryption on an existing cluster, the cluster will experience a network partition during the transition, leading to temporary loss of consistency. If possible, choose whether to employ node-to-node encryption when first creating the cluster. See Configuring SSL/TLS for DSE using LCM (page 604).

1. Click Clusters from the Lifecycle Manager navigation menu. If you do not have any clusters yet, click the adding a cluster link on the Getting Started (page 552) page.

2. Click the + (plus) sign above the Clusters pane.
The Add Cluster dialog appears.

3. Enter a name for the cluster. The name reflects the cluster name in both DataStax Enterprise and Lifecycle Manager.

   The cluster name cannot be edited after clicking Save because LCM populates the `cluster_name` field in `cassandra.yaml` with the entered value.

4. Select predefined SSH Credentials to use for accessing the machines that nodes reside on.

5. Specify a SSH Management Port to override the default of 22.

6. Select a predefined Config Profile to apply at the cluster level. If config profiles are not defined for datacenter or node levels, those levels inherit from the cluster.

7. Enter a descriptive comment about the cluster. Example: Dev cluster, or Prod cluster.

8. Select a predefined Repository from which to access DataStax Enterprise install packages. Datacenters and nodes inherit the repository from the cluster level. Defining a repository for the datacenter or node level is not necessary.
9. Click **Save**.

The newly defined cluster displays in the Clusters pane. Click the cluster title to view its details. The Datacenters pane becomes visible.

What's next:

1. Add another cluster. Repeat this procedure as necessary.

2. Add datacenters *(page 560)* to the cluster.

3. Add nodes *(page 563)* to the datacenters.

4. Run an installation job *(page 574)*. After Lifecycle Manager successfully creates a cluster during an install job, LCM automatically adds the cluster to the OpsCenter workspace for monitoring and management.

**Editing a cluster**

Edit cluster details and run a job to update a cluster.

1. Click **Clusters** from the Lifecycle Manager navigation menu.

2. Select the cluster to edit in the Clusters pane and choose **Edit** from the Clusters pane menu.

   The Edit Cluster dialog appears.
3. Make any changes, such as selecting a new Config Profile to apply. If the cluster has an alias, the **Display Name** appears as a read-only field above the **Cluster Name**. To edit or remove the alias, see Changing the display name of a cluster (page 285).

4. Click **Save**.

**What's next:** Depending on the edit, run an install (page 574) or configure (page 576) job at the cluster level.

**Deleting a cluster**

Stop managing and remove a cluster from Lifecycle Manager. Removing a cluster also removes its datacenters and nodes. Removing a cluster does not actually physically delete a cluster; it simply removes it from management by Lifecycle Manager.

1. Click **Clusters** from the Lifecycle Manager navigation menu.

2. Select the cluster to delete in the Clusters pane and choose **Delete** from the Clusters pane menu. A dialog prompts you to confirm no longer managing the cluster within Lifecycle Manager.
3. Click **Delete**.
   The cluster no longer appears in the Clusters pane. Any datacenters or nodes in the cluster are also removed from Lifecycle Manager.

**Adding a datacenter**

Define the datacenter for a cluster topology. Add datacenters to a cluster.

**Prerequisites:**
- Add a config profile (*page 600*)
- Add SSH credentials (*page 591*)
- Add a cluster (*page 556*)

1. Click **Clusters** in the Lifecycle Manager navigation menu. In the Clusters pane, select the cluster to which you want to add a datacenter.
   Details about the cluster are displayed. The Datacenters pane becomes visible.

2. Click the + (plus) sign above the Datacenters pane.
   The Add Datacenter dialog appears.
3. Enter a unique **Name** for the datacenter.

   The datacenter name cannot be edited after clicking Save. Datacenter names must be unique within a cluster.

4. Consult the following table for assistance with completing the fields:

   **Table 21: Datacenter fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
   | Name                   | Enter a name for the datacenter. Datacenter names must be unique within a cluster. Required.  
   |                         | **Note:** Changing the name of a datacenter after the datacenter entity was created and saved is not supported in DataStax Enterprise or Lifecycle Manager. |
   | Config Profile         | Select a configuration file ([page 594](#)) to apply at the datacenter level to override inheriting the configuration profile from the cluster level. |
   | SSH Credentials        | Select an SSH credential ([page 591](#)) to apply at the datacenter level to override inheriting the credentials from the cluster level. |
   | SSH Management Port    | Enter a port value if you do not want to inherit the value from the cluster. Default value for SSH port at the cluster level is 22. |
   | Comment                | Enter a comment about the datacenter that identifies its purpose or location. For example, DC East Coast Production DSE v5. |
   | Workload               | Select the workload type. Node workload type must be homogeneous within each datacenter. Available options:  
   |                         | • Cassandra (default)  
   |                         | • Solr  
   |                         | • Spark  
   |                         | • Spark + Solr  
   | DSE Graph (DSE 5.0+ only) | Select the DSE Graph option if the datacenter is for a DSE Graph database. |

5. Click **Save**.

   The newly added datacenter displays in the Datacenters pane of the Clusters page. The Datacenter Details display in the details area next to the cluster model panes.
What's next:

1. Add another datacenter. Repeat this procedure as necessary.

2. Add nodes (page 563) to the datacenters.

3. Run an installation job (page 574).

   **Important:** After the new datacenter has been added and the nodes are all online, you must change the replication strategy either manually with CQL or using OpsCenter (page 265). Run nodetool rebuild to propagate the datacenter with data. LCM does not perform those operations.

**Editing a datacenter**

Edit datacenter details and run a job to update a datacenter.

   **Note:** Changing the name of a datacenter after the datacenter entity was created and saved is not supported in DataStax Enterprise or Lifecycle Manager.

1. Click **Clusters** from the Lifecycle Manager navigation menu.

2. Select a cluster in the Clusters pane.
   
   The Datacenters pane appears.

3. Select the datacenter to edit in the Datacenters pane and choose **Edit** from the Datacenters pane menu.
   
   The Edit Datacenter dialog appears.
4. Make any necessary changes, such as selecting a new Config Profile to apply.

5. Click **Save**.

**What's next:** Depending on the edit, run an install *(page 574)* or configure *(page 576)* job at the datacenter level.

**Deleting a datacenter**

Delete a datacenter from managing within Lifecycle Manager. Removing a datacenter also removes its child nodes.

1. Click **Clusters** from the Lifecycle Manager navigation menu.

2. Select a cluster in the Clusters pane.
   
   The Datacenters pane appears.

3. Select the datacenter to unmanage in the Datacenters pane.

4. Click **Delete** from the Datacenters pane menu.
   
   A dialog prompts you to confirm removing the datacenter and its nodes from management within LCM.

5. Click **Delete**.
   
   The datacenter no longer appears in the Datacenters pane, and its nodes no longer appear in the Nodes pane.

**Adding a node**

Add nodes to a datacenter to manage install and config jobs from Lifecycle Manager.
Lifecycle Manager

**Note:** Lifecycle Manager does not currently support managing DataStax Enterprise Multi-Instance nodes (also referred to as *dense* nodes).

**Prerequisites:**

- Add a cluster *(page 556).*
- Add a datacenter *(page 560).*
- If you need a Configuration Profile specialized at the node level, define a node-specific config profile *(page 600).*
- If you need an SSH Credential to specify access and privileges at the node level, add an SSH credential *(page 591)* for that purpose.
- An installed version of Python 2.6 through 2.7 is required on the target nodes. LCM does not automate the installation of Python. Install jobs fail if Python is not installed.

**Note:** Repositories *(page 586)* are inherited from the cluster.

**Procedure:**

1. Click **Clusters** in the Lifecycle Manager navigation menu.

2. Select the cluster and its datacenter to which you want to add a node.

3. Click the + plus sign above the Nodes pane.

   The Add Node dialog appears. The *asterisks indicate required fields.* The Address fields display the default values that are used if no values are entered.
4. Enter a **Name** for the node.
   The node name must be unique per datacenter ID.

5. Enter the **SSH IP Address**.

6. Consult the following table for assistance with completing the fields and overriding their defaults:

**Table 22: Node fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name*</td>
<td>Enter a name for the node. Nodes names must be unique within a datacenter. Required.</td>
</tr>
<tr>
<td>Rack</td>
<td>Enter the name of the rack. If a rack is not entered, the rack name default is rack. Example: rack1.</td>
</tr>
<tr>
<td>SSH IP Address*</td>
<td>Default: SSH IP Address specified for the cluster. Required.</td>
</tr>
<tr>
<td>SSH Management Port</td>
<td>Enter a port value if you do not want to inherit the value from the datacenter. Default value for SSH port at the cluster level is 22.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SSH Credentials</td>
<td>Select an <a href="page591">SSH credential (page 591)</a> to apply at the node level to override inheriting the credentials from the datacenter level.</td>
</tr>
<tr>
<td>Config Profile</td>
<td>Select a <a href="page594">configuration file (page 594)</a> to apply at the node level, which overrides inheriting the configuration profile from the cluster or datacenter levels. Defaults to the datacenter config profile.</td>
</tr>
<tr>
<td>Listen Address</td>
<td>Default: SSH IP Address. The IP address or hostname that Cassandra binds to for connecting to other Cassandra nodes.</td>
</tr>
<tr>
<td>Broadcast Address</td>
<td>Default: Listen Address. The broadcast address is the IP address a node tells other nodes in the cluster to contact it by. A broadcast address allows public and private addresses to be different. For example, use the broadcast address in topologies where not all nodes have access to other nodes by their private IP addresses.</td>
</tr>
<tr>
<td>RPC Address</td>
<td>Default: SSH IP Address.</td>
</tr>
<tr>
<td>Broadcast RPC Address</td>
<td>Default: RPC Address.</td>
</tr>
<tr>
<td>Comment</td>
<td>Enter an optional comment about the node.</td>
</tr>
</tbody>
</table>
## Seed Node

Select a seed node option:

- **Automatically choose**: (Default) Allow Lifecycle Manager to select a seed node. LCM designates 1 node per rack in each datacenter, or two seeds per datacenter if a single rack is defined.

- **Make this a seed node**: Select to designate the node as a seed node used by gossip for bootstrapping new nodes joining a cluster.
  
  **Warning**: When adding a node to an existing datacenter that has already been converged (that is, an install job has already been run) in LCM, a tooltip warning appears: New nodes that list themselves as seeds will fail to bootstrap and will require immediately running a repair on the node. DataStax recommends designating the node as a seed after the node has bootstrapped into the cluster.

- **Do not make this a seed node**: Select to explicitly exclude the node from seed node designation.

After adding a node but before running a job, the Node Details displays the status as not a seed node. The true seed node status is not displayed until after running an install job.

**Note**: It is a best practice to designate more than one seed node per datacenter (ideally two per datacenter); however, do not make all nodes seed nodes. Please read Internode communications (gossip).

---

7. Click **Save**.
   
   The newly added node displays in the Nodes pane of the Clusters page. The Node Details display in the details area next to the cluster model panes.

**What’s next:**

1. Continue adding nodes to the datacenter until done. Repeat the steps in this procedure.

2. Run an installation job *(page 574)*.

### Editing a node

Edit node details and run a job to update a node.

1. Click **Clusters** from the Lifecycle Manager navigation menu.

2. Select a cluster in the Clusters pane.
The Datacenters pane appears.

3. Select a datacenter in the Datacenters pane.
   The Nodes pane appears.

4. Select the node to edit in the Nodes pane and choose **Edit** from the Nodes pane menu.
   The Edit Node dialog appears.

5. Make any necessary changes, such as selecting a new Config Profile to apply or changing the Seed Node status.

6. Click **Save**.

**What's next:** Depending on the edit, run an **installation** (page 574) or **configuration** (page 576) job at the node level.

**Deleting a node**

Remove a node from a cluster topology in Lifecycle Manager.

1. Click **Clusters** from the Lifecycle Manager navigation menu.

2. Select a cluster in the Clusters pane.
   The Datacenters pane appears.
3. Select a datacenter in the Datacenters pane.
   The Nodes pane appears.

4. Select the node to unmanage in the Nodes pane.

5. Click **Delete** from the Nodes pane menu.
   A dialog prompts you to confirm removing the node from management within LCM.

6. Click **Delete**.
   The node no longer appears in the Nodes pane.

**Importing an unmanaged cluster**

Import the topology of a DataStax Enterprise cluster that already exists in OpsCenter and is being monitored in OpsCenter into Lifecycle Manager so you can start centrally managing configurations. Lifecycle Manager imports the cluster topology and automatically builds the topology model in the Clusters (page 554) workspace. If a cluster is not connected to OpsCenter, the cluster is not displayed in LCM and is not available for import.

**Note:** Lifecycle Manager does not currently support managing DataStax Enterprise Multi-Instance nodes (also referred to as dense nodes).

**Note:** At this time, Lifecycle Manager only supports importing DataStax Enterprise clusters installed from Debian (.deb) or RedHat (.rpm) packages.

**Prerequisites:**

- Add **SSH credentials** (page 591).
- Ensure all nodes in the cluster you want to import use the same SSH credentials. If some nodes require different credentials, see **manually importing a cluster** (page 571).
- Only the **GossipingPropertyFileSnitch (GPFS)** is supported for managing or importing DataStax Enterprise clusters in Lifecycle Manager.
- Lifecycle Manager currently does not support configuring nodes by network interface (**listen_interface** or **rpc_interface**). If a cluster is configured with those options, the import will fail. Reconfigure the nodes by **listen_address** and **rpc_address** rather than the corresponding interface options before importing the cluster.
- Datacenters must consist of a homogeneous node workload type. Mixed-load datacenters cannot be imported into Lifecycle Manager.
- If **DSE Configuration Encryption** is enabled, it must be disabled prior to importing the cluster into LCM.

1. Click **Clusters** from the Lifecycle Manager navigation menu.
   Clusters that are not being managed by LCM are indicated with a red plus sign in the Clusters pane.
2. In the Clusters pane, select the cluster to manage.
   The Automatic Cluster Import dialog appears.

3. Select the SSH credentials (page 591) defined for the cluster.

4. Read the warning messages. Follow the instructions that are applicable to your environment:
   a. If nodes do not use the same SSH credentials, you must follow the instructions to manually import the cluster (page 571).

5. Select acknowledgement of the warnings.
6. Click **Submit**.

   The import cluster job begins processing. Follow the progress in the **Jobs** (page 580) workspace. If unsuccessful, a dialog provides a link to review the job details.

What’s next:

- Create the necessary **configuration profiles** (page 600) and associate them with the cluster, datacenter, or nodes in the **Clusters** (page 554) workspace.
- **Add a repository** (page 587) and associate it with the cluster in the **Clusters** (page 554) workspace for running future install jobs (page 572).

**Manually creating a cluster topology**

Manually create the cluster topology model that reflects an existing cluster when nodes do not have similar SSH credentials required for an **automatic import** (page 569) of the cluster topology.

1. **Add SSH credentials** (page 591) so LCM can remotely log in to target machines when performing installation and configuration activities.

2. **Add a configuration profile** (page 600) that defines the required DSE configuration on the development, test, and production clusters for your organization. If the configuration of a cluster is intentionally heterogeneous and not uniform at all levels, you can create multiple configuration profiles to apply individually at the cluster, datacenter, or node levels.

3. **Add a repository** (page 587) so LCM can download the DSE software onto target machines from either the public DataStax repo or an internal install repo mirror set up by your organization.

4. Define the **topology** (page 554) of the cluster:
   
   a. **Add a cluster** (page 556).
   
   b. **Add its datacenters** (page 560).
   
   c. **Add its nodes** (page 563).

5. **Run an installation job** (page 574) of DataStax Enterprise at the cluster level. Lifecycle Manager installs and configures DataStax Enterprise on all datacenters and nodes within the cluster. With Lifecycle Manager, there is deep transparency into the progress of each installation and configuration job available when viewing **job summary and details** (page 580).

**Cloning a managed cluster, datacenter, or node**

Clone an existing cluster, datacenter, or node.
1. Click **Clusters** in the Lifecycle Manager navigation menu.

2. Select the cluster to clone in the **Managed Clusters** list. If cloning a datacenter, select the datacenter within the cluster. If cloning a node, select the node within the datacenter.

3. Click the drop menu and choose **Clone**.

**Related information:**

**Cloning a cluster** [Clone cluster data from one DSE cluster to another using the Restore Backup feature in OpsCenter. This workflow requires the source and target clusters to both be managed by the same OpsCenter instance.] *(page 435)*

**Run an Install and Configure Jobs Overview**

Jobs are launched from the **Clusters** *(page 554)* workspace of Lifecycle Manager. Monitor install, configure, and import jobs in the **Jobs** *(page 580)* workspace of Lifecycle Manager.

Lifecycle Manager runs jobs concurrently for different clusters; however, jobs for the same cluster execute sequentially and remain in the Pending state while other install or configure jobs are currently running.

The primary job types you can run on an entity in the LCM topology model (that is; cluster, datacenter, or node) are Install and Configure:

- **Install Job:** An Install job downloads, installs, and configures DataStax Enterprise versions 4.7 and later onto your pre-launched instances. LCM efficiently skips work that is already completed. For example, DSE is not downloaded again if the correct version of DSE is already installed on a target node, but configure steps are performed if necessary. If a DSE package is already installed but is a different version than specified in the install job, the install job fails. Install jobs are idempotent operations and can be safely rerun, ensuring your cluster continues to operate according to your desired configuration. If a job fails for some transient reason, it can be rerun and LCM efficiently completes the remaining work. Prior to OpsCenter provisioning with LCM, manually removing all traces of DataStax Enterprise packages from the affected nodes was required after a failed provisioning attempt.

  **Note:** If the version of DataStax Enterprise associated with a configuration profile being pushed differs from the installed version, the job fails. Version upgrades or downgrades of DataStax Enterprise are not supported within Lifecycle Manager.

- **Configure Job:** A Configure job pushes an associated configuration profile to the appointed nodes and restarts the cluster. An Install Job runs a Configure job in addition to installation.

**Importing an unmanaged cluster** *(page 569)* is also tracked in the Jobs summary and details.

After Lifecycle Manager successfully creates a cluster during an install job, LCM automatically adds the cluster to the OpsCenter workspace for monitoring and management.
Meld Remote Execution

At the beginning of each job, an ephemeral and stateless binary program \( \text{meld.pyz} \) is copied to the target node to execute the job. As a prerequisite, Python 2.7 must be available on the target node.

Supported OS platform check for DSE installs

As of OpsCenter version 6.1.3 and later, LCM automatically performs an OS supported platform check for the version of DSE being installed. Success or failures are logged and are also visible when drilling into the Job Details (page 582):

The OS platform check can be disabled at your own risk by setting the \texttt{disable\_platform\_check} option to \texttt{True} in the [lifecycle\_manager] section of the \texttt{opscenterd.conf} file.

\texttt{opscenterd.conf}

The location of the \texttt{opscenterd.conf} file depends on the type of installation:

- **Package installations**: \texttt{/etc/opscenter/opscenterd.conf}
- **Tarball installations**: \texttt{install\_location/conf/opscenterd.conf}

Related information:
Lifecycle Manager: Jobs [View a summary and drill into details of install, configure, and import jobs in Lifecycle Manager. Monitor the progress of running jobs. Troubleshoot failed and abort idle jobs.] (page 580)

Running a DSE installation job using LCM

Submit a DataStax Enterprise install job to run on a cluster, datacenter, or node. An install job includes running a configuration job (page 576) and by default, installing Java and the JCE Policy files (page 618) required to enable unlimited strength encryption.

Lifecycle Manager runs jobs concurrently for different clusters; however, jobs for the same cluster execute sequentially and remain in the Pending state while other install or configure jobs are currently running.

Lifecycle Manager pushes configuration jobs to a single node at a time and restarts DataStax Enterprise on that node. For a newly added cluster, the very first install job runs on several concurrent nodes. The concurrency speeds the initial install process and is safe because the new cluster cannot serve clients until it has been installed for the first time. After that, install and configure jobs proceed one-node-at-a-time to ensure cluster availability. The job does not progress to the next node until the current node successfully restarts (that is, the node is responding on the `native_transport_port`). By default, the job gracefully stops prematurely if a job fails for a single node. Jobs that are already running on nodes are allowed to finish, but the job does not continue running on any remaining nodes for that job. Doing so prevents any potential configuration problems from bringing down multiple nodes, or even the entire cluster. If required, override this default behavior with the **Continue on error** option, which will attempt to run the job on all nodes, regardless of failure.

Install jobs that expand an existing cluster are throttled to one node at a time to prevent too much data from streaming concurrently.

**Note:** If the version of DataStax Enterprise associated with a configuration profile being pushed differs from the installed version, the job fails. Version upgrades or downgrades of DataStax Enterprise are not supported within Lifecycle Manager.

**Prerequisites:**

Complete the following tasks before running any install jobs in Lifecycle Manager.

1. Create all credentials, both SSH (page 591) and repositories (page 586).

2. Define configuration profiles (page 594).

3. Build the cluster topology model (page 554) or import (page 569) an existing model.

4. An installed version of Python 2.6 through 2.7 is required on the target nodes. LCM does not automate the installation of Python. Install jobs fail if Python is not installed.
5. Check the clock drift rule (page 510) in the Best Practice Service (page 502) to ensure clocks are in sync before proceeding. Clock drift can interfere with LCM generating TLS certificates.

6. Ensure that the SSH server on each node allows file transfer:
   • For OpsCenter 6.0.0-6.0.x, 6.1.0-6.1.x, and 6.5.0-6.5.3, the SSH server on target nodes must allow SFTP transfers.
   • For OpsCenter 6.5.4-6.5.x, the SSH server on the target node must allow file transfer by either SCP or SFTP. LCM tries SFTP first and falls back to SCP.

LCM does not create data directories or manage their permissions. See Creating custom data directories (page 578) for steps to use a custom data directory.

1. Click Clusters from the Lifecycle Manager navigation menu.

2. Select the cluster, datacenter, or node to run an install on.

3. Click Install from the drop menu.

   The Install Job dialog appears.

4. Upon encountering an error on any given node, the job ceases running on additional nodes. Any nodes that are already running continue to completion. To override this default behavior, select Continue on error, which will continue running the job on subsequent nodes until all are finished.

5. Enter a description about the job.
6. Enter a password in the **New DSE password** box to replace the default password for the cassandra user.

7. If LDAP authentication is configured for a DSE cluster, you are prompted to enter your **LDAP Username** and **LDAP Password** the first time an install job is run on the cluster. Providing the LDAP credentials is necessary for OpsCenter to connect to the cluster for monitoring.

8. Select an **auto_bootstrap** option in `cassandra.yaml`. To override the LCM smart default, choose True or False as required.
   - **LCM Default**: Following best practices for data integrity, sets `auto_bootstrap` to True for new nodes, requiring new nodes to be started sequentially. The default job concurrency policy ensures that nodes start sequentially.
     
     This default is different from previous OpsCenter versions.

     **Warning**: When adding a node to an existing datacenter that has already been converged (that is, an install job has already been run) in LCM, a tooltip warning appears: New nodes that list themselves as seeds will fail to bootstrap and will require immediately running a repair on the node. DataStax recommends designating the node as a seed after the node has bootstrapped into the cluster.
   - **True**: Explicitly sets `auto_bootstrap` to True.
   - **False**: Explicitly sets `auto_bootstrap` to False.

9. Click **Submit**. The job is submitted. A dialog informs you the job is in the queue to run.

10. Click **View Job Summary** to navigate quickly to the **Jobs (page 580)** page to monitor the job progress. Click **Close** if you do not want to immediately monitor the job and prefer to remain in the Clusters workspace.

**Running a configuration job**

Synchronizing a managed cluster, datacenter, or node pushes a configuration profile or updates to configuration profiles (page 603) to designated targets. When Lifecycle Manager pushes its version of a configuration, it overwrites any configuration changes that were not made within Lifecycle Manager. Running a configure job realigns configuration and ensures uniform configuration.

**Note**: If the version of DataStax Enterprise associated with a configuration profile being pushed differs from the installed version, the job fails. Version upgrades or downgrades of DataStax Enterprise are not supported within Lifecycle Manager.
Lifecycle Manager runs jobs concurrently for different clusters; however, jobs for the same cluster execute sequentially and remain in the Pending state while other install or configure jobs are currently running.

Lifecycle Manager pushes configuration jobs to a single node at a time and restarts DataStax Enterprise on that node. For a newly added cluster, the very first install job runs on several concurrent nodes. The concurrency speeds the initial install process and is safe because the new cluster cannot serve clients until it has been installed for the first time. After that, install and configure jobs proceed one-node-at-a-time to ensure cluster availability. The job does not progress to the next node until the current node successfully restarts (that is, the node is responding on the `native_transport_port`). By default, the job gracefully stops prematurely if a job fails for a single node. Jobs that are already running on nodes are allowed to finish, but the job does not continue running on any remaining nodes for that job. Doing so prevents any potential configuration problems from bringing down multiple nodes, or even the entire cluster. If required, override this default behavior with the **Continue on error** option, which will attempt to run the job on all nodes, regardless of failure.

**Prerequisites:**

1. Create all credentials, both SSH (page 591) and repositories (page 586).
2. Define configuration profiles (page 594).
3. Build the cluster topology model (page 554) or import (page 569) an existing model.
4. Run an installation job (page 574) for the initial installation of DSE.

   **Note:** If importing an existing cluster (page 569) into LCM, run an install job after the import. A configuration job fails if an install job has not preceded the configuration job.

1. Click **Clusters** in the Lifecycle Manager navigation menu.
2. Select the cluster, datacenter, or node to run a configuration for.
3. Click **Configure** from the drop menu.

   The **Run Configuration Job** dialog appears.
4. Upon encountering an error on any given node, the job ceases running on additional nodes. Any nodes that are already running continue to completion. To override this default behavior, select **Continue on error**, which will continue running the job on subsequent nodes until all are finished.

5. Enter a description about the job.

6. Enter a password in the **New DSE password** box to replace the default password for the cassandra user.

7. Click **Submit**.

   The job is submitted. A dialog informs you the job is in the queue to run.

8. Click **View Job Summary** to navigate quickly to the **Jobs (page 580)** page to monitor the job progress. Click **Close** if you do not want to immediately monitor the job and prefer to remain in the Clusters workspace.

---

**Creating custom data directories**

LCM does not create data directories or manage their permissions. Complete the following steps to use a custom data directory.

**Note:** The following commands use the `useradd` and `groupadd` commands. These examples are not comprehensive for all environments, but instead provide an illustrative example. Use whichever common utility is employed by your organization to complete these steps.

1. Create the **cassandra user and group**:
2. Create the custom data directory and assign ownership and permissions to the `cassandra` user:

```
$ mkdir /opt/my-custom-dir
$ chown cassandra:cassandra /opt/my-custom-dir
$ chmod 750 /opt/my-custom-dir
```

3. If additional custom directories are required, repeat the previous step to create the directories and assign proper ownership and permission.

**Downloading the generated CA cert**

Download the CA certificate automatically generated by Lifecycle Manager after enabling client-to-node encryption. LCM automates the process of setting up SSL certificates using an internal certificate authority (page 599). Configure your CQL clients to trust certificates signed by the certificate authority.

**Prerequisites:**

1. Enable client-to-node encryption (page 598) in the configuration profile (page 594) associated with the cluster.

2. If the cert was not generated, an error message in both the `opscenterd.log` and the job event details (page 582) indicate the SSL certificate is not yet valid. Ensure that there is not any clock drift, which can interfere with generating the cert chain properly. Check the clock drift rule (page 510) in the Best Practice Service (page 502) to ensure clocks are in sync.

**opscenterd.log**

The location of the `opscenterd.log` file depends on the type of installation:

- Package installations: `/var/log/opscenter/opscenterd.log`
- Tarball installations: `install_location/log/opscenterd.log`

1. In the **Clusters** workspace of Lifecycle Manager, select the cluster in the **Clusters** pane. The **Cluster Details** pane for the cluster appears.
2. In the **Cluster Details** pane, click the **Download Cert** link for CA Certificate.

   The browser downloads the certificate file. By default, the DSE client CA certificate file provided by LCM is named *cacert* and has a PEM format.

3. Use the CA Certificate to configure CQL clients to communicate over SSL/TLS. The process for configuring each CQL client is unique. Refer to the steps for configuring SSL/TLS for *cqlsh* as an example.

   For example, using CQLSH client, you can (SSL) access DSE nodes with the following command:

   ```
   SSL_CERTFILE=cacert cqlsh --ssl <DseNode_Host>
   ```

   Clients are able to connect to the DataStax Enterprise cluster via CQL over SSL/TLS.

### Lifecycle Manager: Jobs

View a summary and drill into details of install, configure, and import jobs in Lifecycle Manager. Monitor the progress of running jobs. Troubleshoot failed and abort idle jobs.

**Related information:**

- [Run an Install and Configure Jobs Overview](#) [Jobs are launched from the Clusters workspace of Lifecycle Manager. Monitor install, configure, and import jobs in the Jobs workspace of Lifecycle Manager.] (page 572)

### Monitoring Job Status Overview

View a summary of install, configure, and import jobs in Lifecycle Manager. View a summary of the status of all jobs and details about the job type, target, and its job ID. Monitor the progress of a running job. Abort a job that might be taking an excessive time to execute.

Deep transparency into currently running, completed, or failed jobs is available. Drill into details of a failed job to conveniently troubleshoot the root cause of the failure from within the Jobs workspace before resorting to reviewing logs. Investigate any failed jobs or those that are taking an excessive time to run.
The status of each job is clearly indicated with icons and their descriptions in the Status column of the Jobs workspace. Refer to the Job status legend for further details:

Table 23: Job status legend

<table>
<thead>
<tr>
<th>Status</th>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not Run</td>
<td>![icon]</td>
<td>The initial install job has not been run. This indicator displays in the Clusters workspace.</td>
</tr>
<tr>
<td>Pending</td>
<td>![icon]</td>
<td>The job is in the queue waiting to run.</td>
</tr>
<tr>
<td>Running</td>
<td>![icon]</td>
<td>The job is currently running.</td>
</tr>
<tr>
<td>Success (Completed)</td>
<td>![icon]</td>
<td>The job ran successfully.</td>
</tr>
<tr>
<td>Failure</td>
<td>![icon]</td>
<td>The job failed. Investigate the issue by drilling into the job details. Try running the job again. <strong>Note:</strong> An ORPHANED status only appears in logs upon startup if there were any jobs left in a RUNNING status. An ORPHANED status indicates a job failed because OpsCenter was restarted while a job was running. A WILL_FAIL status in the logs indicates that a job was marked early in processing as guaranteed to fail, which might be informative from an API troubleshooting perspective. The ORPHANED and WILL_FAIL statuses only appear in logs and do not appear in the UI.</td>
</tr>
<tr>
<td>Idle</td>
<td>![icon]</td>
<td>A job was actively running but has at least one node that failed to recently report progress. An idle job is still running and will never automatically fail, since a node could be successfully executing a slow operation. To stop a job in an idle status, you must <strong>manually abort</strong> <em>(page 584)</em> the job. <strong>Note:</strong> The defaults for timing out a job and marking it as idle can be changed with idle timeout configuration options in the <code>[lifecycle_manager]</code> section of opscenterd.conf. For details, see configuring idle timeout thresholds for jobs <em>(page 585).</em></td>
</tr>
</tbody>
</table>
Aborted

The job was manually aborted (page 584). Aborted jobs appear in logs with a TERMINATED status.

The Jobs page keeps a complete historical record of all jobs performed in Lifecycle Manager.

**Note:** The data (cluster topology models, configuration profiles, credentials, repositories, job history, and so forth) for Lifecycle Manager is stored in the lcm.db database. Your organization is responsible for backing up the lcm.db database. You must also configure failover (page 169) to mirror the lcm.db.

**lcm.db**

The location of the Lifecycle Manager database lcm.db depends on the type of installation:

- **Package installations:** /var/lib/opscenter/lcm.db
- **Tarball installations:** install_location/lcm.db

**Note:** The data (cluster topology models, configuration profiles, credentials, repositories, job history, and so forth) for Lifecycle Manager is stored in the lcm.db database. Your organization is responsible for backing up the database. You must also configure failover (page 169) to mirror the lcm.db.

**opscenterd.conf**

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

- **Package installations:** /etc/opscenter/opscenterd.conf
- **Tarball installations:** install_location/conf/opscenterd.conf

**Related information:**
[Run an Install and Configure Jobs Overview](#) [Jobs are launched from the Clusters workspace of Lifecycle Manager. Monitor install, configure, and import jobs in the Jobs workspace of Lifecycle Manager.] (page 572)

**Viewing job details**

View details about currently running or already completed jobs in Lifecycle Manager. Use the filter lists to quickly locate the job you want to view.

1. Click **Jobs** in the Lifecycle Manager navigation menu.

   The Lifecycle Manager: Jobs page appears.
2. If necessary, use the filter lists to quickly locate the job. Specify your filter criteria in each list and click **Filter**. Filter by:
   - **Clusters**: All clusters or a specific cluster.
   - **Datacenters**: All datacenters or a specific datacenter.
   - **Nodes**: All nodes or a specific node.
   - **Job Type**: All Types, Install, Install Agent, Configure, or Import.

3. Click the **Created On** column title to quickly sort jobs by date created in ascending or descending order.

4. Click the view details icon to access more details for any event within a job.

   The details page for the job appears.

5. Drill deeper into more granular details about a job. Click the view details icon for the job item you want to view.
6. When you are done viewing job details, click the back arrow icons to navigate back to the main jobs page.

**Abort a job**

Abort a pending, running, or an idle install, configuration, or import job in Lifecycle Manager. Abort a job that might be taking an excessive time to execute. If a job has hit the idle threshold, the Status column indicates the job is IDLE.

1. Click **Jobs** in the Lifecycle Manager navigation menu.
   The Lifecycle Manager: Jobs page appears.

2. If necessary, use the filter lists to quickly locate the job. Specify your filter criteria in each list and click **Filter**. Filter by:
   - **Clusters**: All clusters or a specific cluster.
   - **Datacenters**: All datacenters or a specific datacenter.
   - **Nodes**: All nodes or a specific node.
   - **Job Type**: All Types, Install, Install Agent, Configure, or Import.
3. Click the details icon in the Actions column to access the details page for a job. The details page for the job appears.

![](image)

4. Click **Abort**. The job ends. Aborted jobs display an aborted status.

### Configuring SSH connection and idle timeout thresholds for LCM jobs

Adjust the default timeout values for SSH connection attempts and idle status during Lifecycle Manager jobs as necessary for your environment. If jobs are timing out prematurely, increase the idle thresholds as appropriate for the job type:

- installation job
- restarting a node during a job
- configuration job on a node
- importing a cluster job

**opscenterd.conf**

The location of the `opscenterd.conf` file depends on the type of installation:

- Package installations: `/etc/opscenter/opscenterd.conf`
- Tarball installations: `install_location/conf/opscenterd.conf`

1. Open `opscenterd.conf` for editing.

2. Adjust the available SSH connection values shown below in the `[lifecycle_manager]` section:

```
[lifecycle_manager] ssh_connect_timeout_in_seconds
The number of seconds to wait for the SSH server on a target node to respond to a connection attempt during an LCM job. Default: 30 seconds.

[lifecycle_manager] ssh_max_attempts
The maximum number of times to attempt an SSH connection to a given target during an LCM job. Default: 3.

[lifecycle_manager] ssh_retry_delay_in_seconds
```


The number of seconds to wait between SSH connection retries during an LCM job. Default: 10 seconds.

The following example doubles the default values for SSH connection attempts by LCM:

```
[lifecycle_manager]
ssh_connect_timeout_in_seconds = 60
ssh_max_attempts = 6
ssh_retry_delay_in_seconds = 20
```

3. Adjust the available default idle timeout values shown below in the [lifecycle_manager] section:

```ini
[lifecycle_manager] node_install_idle_timeout
   Duration of inactivity in seconds before marking an install job as idle. Default: 900 seconds.

[lifecycle_manager] node_restart_idle_timeout
   Duration of inactivity in seconds before marking a node restart job as idle. Default: 120 seconds.

[lifecycle_manager] node_configure_idle_timeout
   Duration of inactivity in seconds before marking a node configuration job as idle. Default: 120 seconds.

[lifecycle_manager] node_cluster_import_idle_timeout
   Duration of inactivity in seconds before marking a cluster import job as idle. Default: 120 seconds.

[lifecycle_manager] node_install_agent_idle_timeout
   Duration of inactivity in seconds before marking an agent install job as idle. Default: 120 seconds.
```

The following example doubles the default timeout values for LCM jobs:

```
[lifecycle_manager]
node_install_idle_timeout = 1800
node_restart_idle_timeout = 240
node_configure_idle_timeout = 240
node_cluster_import_idle_timeout = 240
node_install_agent_idle_timeout = 240
```

4. Restart OpsCenter (page 75).

Lifecycle Manager: Repositories

Repositories Overview

Repositories allow automatically downloading packages from a repository without repeatedly entering your credentials during a DataStax Enterprise install job. Repositories use user-defined credentials to access and download all required packages for installing DataStax Enterprise from the DataStax Repository or from an internal private repository maintained
by your organization. Add a repository to use for automatically downloading Debian or RPM packages within the DataStax Enterprise repository. In addition to repo credentials, you can define custom URLs that point to your own internal mirror of the DataStax Repository. If your organization has an internal mirror of the DataStax Repository, you could also create a DataStax Repo as an alternative repository resource to your private repo. Unless your organization has multiple local mirrors of the DataStax package repository, you probably only need to define one repository.

Optionally, configure a proxy (page 621) to expedite package downloads for targets with limited internet connectivity, which is especially useful for offline installs accessing a custom repository. For a detailed example, see the Knowledge Base article.

Repositories are applied at the cluster level only in the Clusters (page 554) workspace. Datacenters and nodes inherit the repository from the cluster. If you require a different repository per datacenter, you can read more about a potential workaround in this Knowledge Base article.

**Note:** The data (cluster topology models, configuration profiles, credentials, repositories, job history, and so forth) for Lifecycle Manager is stored in the lcm.db database. Your organization is responsible for backing up the lcm.db database. You must also configure failover (page 169) to mirror the lcm.db.

### lcm.db

The location of the Lifecycle Manager database lcm.db depends on the type of installation:

- **Package installations:** /var/lib/opscenter/lcm.db
- **Tarball installations:** install_location/lcm.db

**Note:** The data (cluster topology models, configuration profiles, credentials, repositories, job history, and so forth) for Lifecycle Manager is stored in the lcm.db database. Your organization is responsible for backing up the database. You must also configure failover (page 169) to mirror the lcm.db.

## Adding a repository

Add a repository to use for automatically downloading Debian or RPM packages within the DataStax Enterprise repository. In addition to providing repo credentials, you can define custom URLs that point to your own mirror of the DataStax Repository.

**Note:** The data (cluster topology models, configuration profiles, credentials, repositories, job history, and so forth) for Lifecycle Manager is stored in the lcm.db database. Your organization is responsible for backing up the lcm.db database. You must also configure failover (page 169) to mirror the lcm.db.

### lcm.db

The location of the Lifecycle Manager database lcm.db depends on the type of installation:

- **Package installations:** /var/lib/opscenter/lcm.db
• Tarball installations: `install_location/lcm.db`

**Note:** The data (cluster topology models, configuration profiles, credentials, repositories, job history, and so forth) for Lifecycle Manager is stored in the `lcm.db` database. Your organization is responsible for backing up the database. You must also configure failover (page 169) to mirror the `lcm.db`.

**Prerequisites:** Optionally configure a proxy for package downloads (page 621) for targets with limited internet connectivity.

1. Click **Repositories#Add Repository**.
   
The Add Repository dialog appears.

2. Enter a **Name** for the repository credential.

3. Enter a **Comment** that describes the repository credential.

4. Select a **Repository** option:
   
   - Select **Access DataStax Repo** if using the DataStax Repository for packages (default).
   - Select **Access Private Repo** if your organization has an internal repository for software packages:
     
     1. Enter the **Repository URL**.
     2. Enter the **Repository Key URL** to retrieve a public key from. A repository key URL is the address of the public key of a package repository. You can download the key and use it to verify that packages downloaded from the repository have not been tampered with.
• Select **Manually configure DataStax repo** if the repo is configured externally from LCM. The target nodes are already able to download DataStax Enterprise through their package manager and LCM should not attempt to automatically configure package repositories. This option is useful if DataStax packages have been mirrored to a Red Hat Satellite channel, if a configuration management system other than LCM is used to configure package repositories, or if LCM is not compatible with your repository setup and your organization requires configuring its repo externally from LCM.

5. For the DataStax Repository, using the configured proxy is enabled by default. If you are not using the DataStax Repo, the Use Configured HTTP Proxy option is disabled. Custom package repositories are commonly located internal to the HTTP proxy. If your custom package repository is external to your proxy, select **Use Configured HTTP Proxy** to ensure the package manager on your target nodes uses your configured HTTP proxy. For information on configuring Lifecycle Manager to forward requests to your proxy, see configuring a proxy for package downloads (page 621).

6. Enter the **Username** and **Password** for accessing the DataStax Repository.

   The username and password must be specified if authentication is enabled on the repository. The DataStax Repository always has authentication enabled. For the DataStax Repository, the username and password are the DataStax Academy account credentials you created on the registration page. The repository username and password are used when downloading packages (they are not used when fetching the repository public key).

   **Note:** When configuring credentials in a Repository, special characters such as #, $, and so forth are supported, but non-ascii unicode characters are not. If you have further questions, please contact DataStax Support and reference ticket OPSC-8921.

7. Click **Save**.

   The newly defined repository credential is listed on the Repositories page along with edit and delete options.
What's next: Edit the cluster (page 558) in the Clusters (page 554) workspace to select the repository. The datacenters and nodes inherit the cluster repository.

**Editing a repository**

Edit a repository credential to update repository access. Make any changes to the DataStax Repo, private repo URLs, proxy usage, or credentials.

1. Click **Repositories**.
2. Click the edit icon for the repository you want to edit.
   
   The Edit Repository dialog appears.

   ![Edit Repository Dialog](image)

   - **Name**: DS std repo
   - **Comment**:
   - **Repository** options: Access DataStax Repo, Access Private Repo
   - **Use Configured HTTP Proxy**:
   - **Username**: test
   - **Password**: ********
   - **Save**
   - **Cancel**

3. Make any needed adjustments.
4. Click **Save**.

**Deleting a repository**

Delete a repository. There must be at least one repository defined for Lifecycle Manager to access DataStax Enterprise install packages.

1. Click **Repositories**.
2. Click **Delete** for the repository you want to delete.

   A dialog prompts you to confirm deleting the repository.
3. Click **Delete**.

**Lifecycle Manager: SSH Credentials**

**SSH Credentials Overview**

Lifecycle Manager logs in to machines with the specified credentials during install, configure, and import jobs. Specify an SSH credential at the cluster, datacenter, or node level. An SSH credential applied at the cluster level is used to log into all machines associated with the cluster. Applying a credential at the datacenter level overrides any cluster-level credentials and applies to all nodes associated with the datacenter. A credential applied at the node level has the highest possible precedence. Define unique SSH credentials as needed for your cluster topologies.

**Note:** The data (cluster topology models, configuration profiles, credentials, repositories, job history, and so forth) for Lifecycle Manager is stored in the lcm.db database. Your organization is responsible for backing up the lcm.db database. You must also configure failover (page 169) to mirror the lcm.db.

If there are any SSH connection issues when running a job, increase the SSH connection settings (page 585) for Lifecycle Manager.

**lcm.db**

The location of the Lifecycle Manager database lcm.db depends on the type of installation:

- **Package installations:** /var/lib/opscenter/lcm.db
- **Tarball installations:** install_location/lcm.db

**Note:** The data (cluster topology models, configuration profiles, credentials, repositories, job history, and so forth) for Lifecycle Manager is stored in the lcm.db database. Your organization is responsible for backing up the database. You must also configure failover (page 169) to mirror the lcm.db.

**Adding SSH credentials**

**lcm.db**

The location of the Lifecycle Manager database lcm.db depends on the type of installation:

- **Package installations:** /var/lib/opscenter/lcm.db
- **Tarball installations:** install_location/lcm.db

**Note:** The data (cluster topology models, configuration profiles, credentials, repositories, job history, and so forth) for Lifecycle Manager is stored in the lcm.db database. Your organization is responsible for backing up the database. You must also configure failover (page 169) to mirror the lcm.db.
Add SSH credentials to use for connecting to the target machines being managed by LCM.

The Login User and SSH Login options allow LCM to remotely log in to a machine over SSH using a password or private key. LCM requires superuser privileges to install software and manage configuration on a target machine. The Privileges Escalation options allow specifying SU or SUDO mechanisms to elevate privileges if necessary after the remote login.

**Note:** The data (cluster topology models, configuration profiles, credentials, repositories, job history, and so forth) for Lifecycle Manager is stored in the lcm.db database. Your organization is responsible for backing up the lcm.db database. You must also configure failover ([page 169](#)) to mirror the lcm.db.

All passwords and private keys are encrypted and safely stored by Lifecycle Manager.

1. **Click SSH Credentials/Add Credential.**
   The Add Credential dialog appears.

2. Enter a **Name** for the credential.
3. Enter a **Comment** that describes the credential. Example: *Superuser access to all nodes.*
4. Enter the name of the **Login User**.
5. Select an **SSH Login** type:
   - **Password:** (Default) Enter the **Login Password** associated with the **Login User**.
   - **Private Key:** Paste the private key into **SSH Private Key**. If applicable, enter the **SSH Unlock Passphrase**.
6. For **Escalate Privileges**, select an option to enter any necessary secondary credentials.
   - **SU**: Enter the **SU to this user** (optional to switch to another user’s account; typically the default root) and **SU password**.
   - **SUDO**: (Default) Enter the **SUDO to this user** and **SUDO password** of the Login User.
   - **None**: The login user is root or otherwise has all privileges required to run LCM jobs, including the ability to install packages and start services; no additional SUDO or SU privileges are required beyond the SSH password or private key.

7. Click **Save**.
   The newly defined SSH credential is listed on the SSH Credentials page along with edit and delete options.

**What’s next:** Edit the cluster ([page 558](#)), datacenter ([page 562](#)), or node ([page 567](#)) as appropriate to select the credential in the Clusters ([page 554](#)) workspace.

### Editing SSH credentials

Edit SSH credentials.

1. Click **SSH Credentials**.

2. Click the edit icon for the credential you want to edit.

3. Make any necessary changes.

4. Click **Save**.

### Deleting SSH credentials

Delete SSH credentials.

**Prerequisites**: Swap out the SSH credential wherever it is in use with another SSH credential before deleting it.

1. Click **SSH Credentials**.

2. Click the delete icon for the credential you want to delete.
   A dialog prompts you to confirm deleting the credential.
3. Click **Delete**.

**Lifecycle Manager: Configuration Profiles**

**Configuration profiles overview**

**Purpose of Configuration Profiles**

Define the required configuration profiles to prevent configuration drift for DataStax Enterprise (DSE) clusters. A configuration profile enforces uniform configuration at the cluster, datacenter, or node level.

A **configuration profile** allows defining and centrally managing consistent configuration settings, which prevents **configuration drift**. Configuration drift happens over time as changes are made on a manual rather than an automated basis, and the changes are applied in an inconsistent manner. Configuration drift contributes to failures with high availability and disaster recovery efforts. If a configuration change is made outside of the Lifecycle Manager application, **running a configuration job (page 576)** within LCM overwrites the configuration on the job targets; ensuring that the clusters, datacenters, and nodes are running as specified in the applied configuration profiles.

**Inheritance and Precedence**

Configuration profiles inherit intelligently within the cluster topology. For example, if a configuration profile is not explicitly specified at the datacenter or node level, the configuration profile is inherited from the cluster level. When creating the **cluster topology model (page 554)**, defined configuration profiles can be applied at the cluster, datacenter, or node level. A configuration profile at the node level takes precedence over datacenter or cluster level profiles. Define configuration profiles that reflect the requirements of the workload node type in a datacenter.

When a configuration job is run, configuration profiles specified at different topology levels are merged in a granular manner. For example, consider a cluster with a configuration profile defined and applied at the cluster level that specifies:

- Use the G1 Garbage Collector (g1gc)
- Use a max heap size of 16 GB
- Does not explicitly specify a commit log directory, instead relying on the default value of /var/lib/cassandra/commitlog.

The cluster has two datacenters; DC1 and DC2:

- DC1 has no configuration profile of its own and therefore inherits its configuration profile from its cluster.
- DC2 has a defined configuration profile at the DC level that specifies a maximum heap size of 32 GB, and a commit log directory of /cassandra_data/commitlog.

When the configuration job runs, the resulting configuration of nodes in each datacenter is as follows:
• All nodes in DC1 inherit cluster settings: g1gc; 16 GB max heap size; and uses the default commit log directory of /var/lib/cassandra/commitlog commitlog directory.

• Nodes in DC2 inherit from the cluster and also override cluster settings with the datacenter-level configuration profile: g1gc (inherited from cluster); 32 GB max heap size (DC configuration profile takes precedence over an explicit setting in a cluster-level configuration profile); /cassandra_data/commitlog commit log directory (DC configuration profile takes precedence over an implicit default inherited from a cluster-level configuration profile).

The inheritance and precedence of configuration profiles allows keeping a cluster consistent by inheriting as much as possible from a cluster-level configuration profile, while also providing the flexibility of specifying only the granular settings that differ in higher precedence within configuration profiles applied at the lower, more granular datacenter and node levels.

**Configuration Profile Files**

Each configuration profile is specific to a recent version of DSE. A configuration profile is composed of multiple configuration files for configuring features of DSE clusters:

![Configuration Profiles](image)

Configuration profiles allow customizing settings for the following configuration files:

- **Cassandra** section:
  - # logback.xml
  - # cassandra.yaml
  - # dse.yaml
# cassandra-env.sh
# jvm.options: See Configuring JVM options in LCM (page 616)

**Note:** When you add a configuration profile, DSE authentication is enabled by default for all supported versions of DSE. DSE Authenticator is enabled in dse.yaml for DSE version 5.0 and later. For more information, see Managing DSE Security using LCM (page 597).

Every configuration option in cassandra.yaml and dse.yaml is editable, while other configuration files use a template system that exposes only frequently used settings. Contact DataStax Support to request additional configuration options.

**• Spark** section:
  - # logback-sparkR.xml
  - # dse-spark-env.sh
  - # logback-spark.xml
  - # spark-defaults.conf
  - # spark-env.sh
  - # logback-spark-executor.xml
  - # logback-spark-server.xml
  - # hive-site.xml
  - # spark-daemon-defaults.conf

For more information, see configuring Spark for DSE and configuring Spark logging options in the DSE Administrator documentation.

**• Lifecycle Manager** section:
  - # Package Proxy (page 621): Accelerate package downloads or isolate DataStax Enterprise clusters offline from the internet.
  - # Java Setup (page 618): Automatically manages JRE installs and JCE Policy files.

**Note:** A configuration file explicitly not managed at this time by Lifecycle Manager is commitlog_archiving.properties, which is used for configuring commit log archive (page 384) and PIT restore for the Backup Service. This file is managed instead from within the Backup Service (page 380).

**Note:** The data (cluster topology models, configuration profiles, credentials, repositories, job history, and so forth) for Lifecycle Manager is stored in the lcm.db database. Your organization is responsible for backing up the lcm.db database. You must also configure failover (page 169) to mirror the lcm.db.

lcm.db

The location of the Lifecycle Manager database lcm.db depends on the type of installation:

**• Package installations:** /var/lib/opscenter/lcm.db
• Tarball installations: `install_location/lcm.db`

**Note:** The data (cluster topology models, configuration profiles, credentials, repositories, job history, and so forth) for Lifecycle Manager is stored in the `lcm.db` database. Your organization is responsible for backing up the database. You must also configure failover (page 169) to mirror the `lcm.db`.

## Configuration Known Issues and Limitations

### Encrypted DSE configuration values

Lifecycle Manager is not currently compatible with DSE Configuration Encryption. If DSE configuration encryption is enabled, it must be disabled prior to importing a cluster (page 569) into LCM. Similarly, secret fields in configuration profiles are not encrypted during storage and are returned in API responses to clients that have permission to read configuration profiles. This behavior for configuration profiles is in contrast to secret fields stored on other models such as ssh credentials, repositories, and clusters in that secret fields are encrypted and not returned to API clients.

### DSE Graph properties (DSE 5.0.1+ only)

DSE Graph configuration in `dse.yaml`, which is configurable through LCM Config Profiles. All Graph properties in `dse.yaml` can be managed through the LCM UI with the exception of `gremlin_server.serializers` and `gremlin_server.scriptEngines`. If you are using LCM and need to customize these properties, be sure to leverage the LCM API to make the changes. Future changes to the Config Profile via the LCM UI will retain properties set through the API.

## Managing DSE Security using LCM

### Authentication

Authentication for DataStax Enterprise clusters is enabled by default in a Lifecycle Manager configuration profile. Configuration profiles created for supported DSE versions earlier than DSE version 5.0 enable the **PasswordAuthenticator** option by default. Configuration profiles created for DSE version 5.0 and later use the **DSE Authenticator** by default:
The following links provide more information about the available security options in `cassandra.yaml`:

- authenticator
- authorizer
- role_manager
- permissions_validity_in_ms
- roles_validity_in_ms
- roles_update_interval_in_ms
- server_encryption_options (internode_encryption)
- client_encryption_options

### Node-to-Node Encryption

Lifecycle Manager can configure DataStax Enterprise clusters to use node-to-node encryption. The feature is disabled by default. See Configuring SSL/TLS for DSE using LCM (page 604) for step-by-step instructions for enabling internode encryption using LCM Config Profiles.

When `internode_encryption` is enabled (by selecting a value of `all`, `dc`, or `rack` in the Security panel of `cassandra.yaml` in Config Profile), Lifecycle Manager automates the process of setting up SSL certificates using an internal certificate authority and deploys the resulting keystore and truststore to each node automatically. No further action is necessary beyond running an install or configure job.

**Important:** When enabling node-to-node encryption on an existing cluster, the cluster will experience a network partition during the transition, leading to temporary loss of consistency. If possible, choose whether to employ node-to-node encryption when first creating the cluster (page 556).

### Client-to-Node Encryption

Lifecycle Manager can configure DataStax Enterprise clusters to use client-to-node encryption. The option is disabled by default. See Configuring SSL/TLS for DSE using LCM
Lifecycle Manager (page 604) for step-by-step instructions for enabling client encryption using LCM Config Profiles.

When client-to-node encryption is enabled (by selecting enabled for client_encryption_options in the Security panel of cassandra.yaml in Config Profile), Lifecycle Manager automates the process of preparing server certificates, exactly as it does for node-to-node encryption (page 598).

Some organizations might not want to use the internal certificate authority in Lifecycle Manager, and can manually deploy the keystore and truststore as described for node-to-node encryption.

Before drivers, cqlsh, and other CQL clients can connect to a cluster with client-to-node encryption enabled, they typically must be configured to trust the appropriate certificates. The process is different for each CQL client and Lifecycle Manager does not automatically configure CQL clients. After enabling client-to-node encryption, configure your CQL clients to use the appropriate certificates.

- If certificates were generated by the internal certificate authority in Lifecycle Manager, download the CA certificate (page 579).
- If certificates were generated outside of Lifecycle Manager, acquire the appropriate CA certificate or self-signed certificates. See Using non-LCM generated certificates (page 600).

**Note:** Enabling require_client_auth for client_encryption_options requires special steps due to an LCM limitation. For more information, refer to the Knowledge Base Article.

Internal Certificate Authority generated by LCM

The process of manually preparing certificates and deploying them can be a barrier to the adoption of security features. To simplify deployments, Lifecycle Manager optionally generates certificates using an internal certificate authority.

- When Lifecycle Manager is first started, it creates a self-signed 2048 bit RSA certificate authority that is stored in the [lifecycle_manager].cacerts_directory in opscenterd.conf.
- When running install or configure jobs, Lifecycle Manager generates a keystore and truststore for each node if necessary. Certificate generation occurs if either node-to-node or client-to-node encryption is enabled, and if there is no pre-existing keystore or truststore in the locations specified by the configuration profile.
- When generating a keystore for each node, Lifecycle Manager creates a certificate signing request for the node, signs the request with the internal certificate authority, and packages the resulting certificate in a JKS-formatted keystore.
- When generating a truststore for each node, Lifecycle Manager packages the CA certificate in a JKS-formatted truststore. The same CA is used to sign certificates for all nodes in all clusters, and it enables validation of all automatically generated certificates.
Using non-LCM generated certificates

Some organizations might want to generate certificates for DataStax Enterprise servers using a commercial or enterprise certificate authority external to LifeCycle Manager:

1. Prior to running an install or configure job, prepare keystores and truststores for each node outside of Lifecycle Manager.

2. Deploy the appropriate keystore and truststore to each DataStax Enterprise server using scp, rsync, or some other method of file deployment. The keystore files are commonly deployed to the `/etc/dse/keystores/` directory.

3. Edit the config profile (page 603) in Lifecycle Manager so that the keystore and truststore paths point to the location where the files were deployed as mentioned above; for example `/etc/dse/keystores/server.keystore` and `/etc/dse/keystores/server.truststore`.

4. Edit the config profile in Lifecycle Manager so that the keystore and truststore passwords allow DataStax Enterprise to unlock the files that were manually deployed.

5. Run an install or configure job (page 572). When executing the job, Lifecycle Manager configures each DataStax Enterprise server to use the pre-deployed keystore and truststore you have provided. Lifecycle Manager does not attempt to prepare certificates using the internal certificate authority when it finds a pre-existing keystore and truststore present on a DataStax Enterprise server.

opscenterd.conf

The location of the `opscenterd.conf` file depends on the type of installation:

- **Package installations:** `/etc/opscenter/opscenterd.conf`
- **Tarball installations:** `install_location/conf/opscenterd.conf`

Adding a configuration profile

Add a configuration profile that defines the configuration for DataStax Enterprise clusters. The config profile informs Lifecycle Manager which supported version of DataStax Enterprise to install, configure, and manage. DSE versions 4.7 and later are supported.

**Note:** When you manually upgrade the version of DataStax Enterprise installed on a cluster, you must add a new Config Profile that reflects the actual version of DSE installed and update your clusters, datacenters, and nodes to use the new profile where appropriate. To prevent any potential loss of settings between different versions, the DSE version cannot be edited after saving the Config Profile. LCM jobs (page 572) fail if the installed DSE version does not match the version specified in the Config Profile associated with a cluster, datacenter, or node. Automated DSE upgrades are not supported in Lifecycle Manager at this time.
Note: The data (cluster topology models, configuration profiles, credentials, repositories, job history, and so forth) for Lifecycle Manager is stored in the lcm.db database. Your organization is responsible for backing up the lcm.db database. You must also configure failover (page 169) to mirror the lcm.db.

lcm.db

The location of the Lifecycle Manager database lcm.db depends on the type of installation:

- **Package installations**: /var/lib/opscenter/lcm.db
- **Tarball installations**: install_location/lcm.db

Note: The data (cluster topology models, configuration profiles, credentials, repositories, job history, and so forth) for Lifecycle Manager is stored in the lcm.db database. Your organization is responsible for backing up the database. You must also configure failover (page 169) to mirror the lcm.db.

1. From the Lifecycle Manager navigation menu, click **Config Profiles#Add Config Profile**.
   The Config Profiles page appears.

   ![Lifecycle Manager: Config Profiles](image)

2. Enter a **Name** for the config profile.

3. Select the supported version of DataStax Enterprise (4.7 and later) to associate with the config profile.
**Note:** The DSE Version cannot be edited after saving the Config Profile. If you manually upgrade DSE, you must add a Config Profile with the new DSE version. Apply the Config Profile where appropriate to a cluster, datacenter, or node.

**Tip:** If the OpsCenter machine does not have an internet connection, you can manually update the definitions (page 166) to access the most currently supported versions of DSE.

The Config Profile pane populates with the config files and their corresponding options for the selected DSE version.

4. Enter a comment about the config profile. For example: DSE 5 Graph.

5. If there are any configuration files for which you want to override the default values, select the configuration file in the Config Profile pane to access its configuration options. A subset of the configuration options are available for overriding defaults for each configuration file. If you click save without overriding any defaults or customizing settings, the config profile is created with standard and sensible default settings. You can continue to edit the config profile file and run a configuration job at any time.

6. **Note:** If at any time you navigate away from the Config Profiles workspace after making changes, you are prompted to confirm leaving the page. Click OK to discard your changes and leave the Config Profiles workspace, or click Cancel to retain any changes and remain in the Config Profiles page.

When you are done with the configuration profile changes, click Save.

The newly added configuration profile is listed on the Configuration Profiles page. An edit and delete option are available for each config profile.

**What's next:**

- Continue customizing the sections of the config profile as appropriate for clusters, datacenters, or nodes.
- Configure Java setup (page 618) for automatically installing and managing Java installs and upgrades, and for installing and managing the JCE Policy files (recommended).
- Configure a package proxy (page 621) to cache and accelerate package downloads (recommended), isolate DataStax Enterprise clusters offline from the internet, or both.
- Select the newly added Config Profile and assign it to the appropriate cluster, datacenter, or node in the Clusters (page 554) Workspace. Edit the cluster (page 558), datacenter (page 562), or node (page 567) and select the Config Profile.
- Run a configuration job (page 576) to propagate the configuration profile to its targets.

**Editing a configuration profile**

Edit a configuration profile. After making changes, deploy the new configuration to the cluster, datacenter, or node by running a configuration job.

**Note:** When you manually upgrade the version of DataStax Enterprise installed on a cluster, you must add a new Config Profile that reflects the actual version of DSE installed and update your clusters, datacenters, and nodes to use the new profile where appropriate. To prevent any potential loss of settings between different versions, the DSE version cannot be edited after saving the Config Profile. **LCM jobs (page 572)** fail if the installed DSE version does not match the version specified in the Config Profile associated with a cluster, datacenter, or node. Automated DSE upgrades are not supported in Lifecycle Manager at this time.

1. Click **Config Profiles** from the Lifecycle Manager navigation menu.

2. Click the edit icon for the config profile you want to edit.

3. In the Config Profile pane, select the configuration file to make changes to.
   - The available configuration options appear for the selected configuration file.

4. **Note:** If at any time you navigate away from the Config Profiles workspace after making changes, you are prompted to confirm leaving the page. Click **OK** to discard your changes and leave the Config Profiles workspace, or click **Cancel** to retain any changes and remain in the Config Profiles page.

   When you are done with editing the config profile, click **Save** to save configuration changes, or **Cancel** to cancel.

**What's next:**
- Check config profile usage
Lifecycle Manager

- Run a configuration job (page 576) to apply the config profile changes.

### Configuring SSL/TLS for DSE using LCM

Configure SSL/TLS for a DSE cluster using Lifecycle Manager Config Profile. This is the recommended procedure for a production environment. Follow these steps to enable node-to-node and client-to-node encryption.

When either node-to-node or client-to-node encryption is enabled, LCM creates keystores and truststores for DSE node-to-node and client-to-node SSL/TLS communication with the following default names and locations:

- `/etc/dse/keystores/server.keystore` (node-to-node keystore)
- `/etc/dse/keystores/server.truststore` (node-to-node truststore)
- `/etc/dse/keystores/client.keystore` (client-to-node keystore)
- `/etc/dse/keystores/client.truststore` (client-to-node truststore)

When client-to-node SSL encryption is enabled, both the OpsCenter daemon and DataStax agents also need to be SSL-enabled to properly communicate with DSE server nodes. LCM automatically configures the ssl configuration options for that purpose in the cluster-specific configuration file `cluster_name.conf`:

```plaintext
[agents]
ssl_keystore_password = cassandra
ssl_keystore = /etc/dse/keystores/client.keystore

[cassandra]
ssl_keystore_password = lifecyclemanager
ssl_keystore = /var/lib/opscenter/ssl/lcm/lcm-auto-generated.truststore
```

The keystore file used for DataStax agents to communicate with DSE nodes is exactly the same as that used by a DSE node connecting to other DSE nodes.

The keystore file used for OpsCenter daemon communicating with DSE nodes is automatically generated by the LCM configuration process and put under the folder `/var/lib/opscenter/ssl/lcm`.

For information about manually configuring cluster connections (using the OpsCenter UI), see Editing OpsCenter cluster connections for authentication or encryption (page 93).

### Prerequisites:

Review the LCM documentation for Managing DSE Security using LCM (page 597).

**Important:** When enabling node-to-node encryption on an existing cluster, the cluster will experience a network partition during the transition, leading to temporary loss of consistency. If possible, choose whether to employ node-to-node encryption when first creating the cluster (page 556).
Note: Enabling require_client_auth for client_encryption_options requires special steps due to an LCM limitation. For more information, refer to the Knowledge Base Article.

cluster_name.conf

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

- Package installations: /etc/opscenter/clusters/cluster_name.conf
- Tarball installations: install_location/conf/clusters/cluster_name.conf

1. Click Config Profiles from the Lifecycle Manager navigation menu.

2. Click the edit icon for the config profile you want to edit, or click Add config profile if you have not already created a profile.

3. In the Config Profile pane under the Cassandra section, click cassandra.yaml.

4. In the Security pane under server_encryption_options, select an option for internode_encryption.

   Available options for node-to-node encryption:
   - all: All inter-node communication is encrypted. Recommended and strongest option.
• **dc**: Traffic between DCs is encrypted. Select this option if there is concern about a performance impact of encrypting traffic locally, but encryption is still required for inter-dc traffic that might transit untrusted links.

• **rack**: Traffic between racks is encrypted.

Tip: For more details about available configuration options, see `server_encryption_options` in the DSE Admin documentation.

5. In the **Security** pane, select the **enabled** option for `client_encryption_options`. 
Tip: For more details about available configuration options, see client_encryption_options in the DSE Admin documentation.

6. Click **Save** to save the Config Profile.

**What's next:**

- Go to the Clusters (page 554) workspace in Lifecycle Manager and select the config profile to apply at the cluster level.
- If an install job has not been run yet on the cluster, Run an Install Job (page 574). Otherwise, Run a configure job (page 576) to apply the config profile changes.
- **Monitor the job** (page 580). When the job completes successfully, SSL/TLS setup for the DSE cluster is done.
- **Download the generated CA cert** (page 579) for use with the DSE client SSL connection.

**Configuring Lifecycle Manager for DSE Graph**

Setting up Lifecycle Manager for DSE Graph workloads is fast and easy. Make sure you configure the DSE Graph-specific options when adding a config profile or adding a datacenter.
1. When adding a config profile *(page 600)*:

   a. Select **dse v5.0.1** or later as the **DataStax Enterprise Version**.

   ![Lifecycle Manager: Config Profiles](image)

   b. Configuration options are available in the **DSE Graph** section of **dse.yaml** in a Lifecycle Manager Config Profile. Override the defaults if necessary. For more information about configuration options, see configuring **DSE Graph options** in the DataStax Enterprise documentation.
Note: Some advanced DSE Graph properties can only be set using the API. For more information, see DSE Graph properties (DSE 5.0.1+ only) (page 597).

2. When adding a datacenter (page 560) to a cluster in the Clusters (page 554) workspace, select the DSE Graph (DSE 5.0+ Only) option in the Add Datacenter dialog:
Lifecycle Manager

Configuring tiered storage

DSE Tiered Storage segregates hot and cold data to designated paths on different gradations of storage media, which can facilitate cost savings for data storage. Configure the storage tier strategy and paths to each tier in dse.yaml within an LCM config profile. Run an install or configure job (page 572) in Lifecycle Manager to push the configuration to all applicable nodes.

After completing this procedure, additional steps are required in DSE to apply tiered storage to a table using CREATE or ALTER table statements:

```sql
CREATE TABLE ks.tbl (k INT, c INT, v INT, PRIMARY KEY (k, c))
WITH COMPACTION={"class":"TieredCompactionStrategy",
             "tiering_strategy": "DateTieredStorageStrategy",
             "config": "strategy1",
             "max_tier_ages": '3600,7200'};
```

Configure the age policy for each tier at the schema or local node level. For more information, see configuring and testing configurations for DSE tiered storage.

1. Click Config Profiles from the Lifecycle Manager navigation menu.

2. Click the edit icon for the config profile you want to edit, or click Add config profile if you have not already created a profile.

3. In the Config Profile pane under the Cassandra section, click dse.yaml.

4. Scroll down to the Miscellaneous pane and locate tiered_storage_options.
5. Click + Add an item under the tiered_storage_options option. The Add tiered_storage_options dialog appears.

6. Enter a strategy_name.

For DSE clusters to use the tiered storage strategy, the strategy must be referenced in the config of the table schema. See configuring DSE Tiered Storage in the DSE Admin documentation.

7. Under local_options, click + Add an item to add local configuration options to overwrite the tiered storage settings in the table schema.

   Note: The class or the tiered storage configuration name cannot be overwritten. For more details and an example, see testing configurations in the DSE Admin documentation.

   The Add local_options dialog appears.
8. Enter a **key** and **value** for each local option and click **Save**. For example, type **max_tier_ages** for key and **3600** for its value.

9. Under **tiers**, click **+ Add an item** to define the paths to the storage tiers.

   The **Add tiers** dialog appears.

10. Enter the paths to the directory for each tier in the strategy and click **Save**.

    Tier order of each path matters. Add the paths for each tier in the order that represents each successive tier in a strategy. Each tier added is used in the order listed, with the top tier typically routing data to the fastest storage media.

    **Note:** Ensure the directories referenced in the path already exist. LCM does not create the directories. If the directories do not exist when the installation or configuration job runs, DSE does not start up.

    The strategy and tiers are added to the config profile.

11. Click **Save** to save the Config Profile.

    **What's next:**
• Apply tiered storage to a table schema and define the maximum age of data in each tier. See DSE Tiered Storage for complete details.

• Go to the Clusters (page 554) workspace in Lifecycle Manager and select the config profile to apply at the cluster, datacenter, or node level. Nodes can inherit config profile settings from the cluster or datacenter levels, or have settings at the node level that take precedence.

• Run a configuration job (page 576) to push the configuration to all of the applicable nodes.

• Add dashboard graphs and alerts to monitor data distribution (page 337) amongst storage tiers using the tiered storage metrics (page 337) available in OpsCenter.

### Configuring row-level access control

Row-level access control (RLAC) enforces an organization's authorization policies to data within a table by matching a filter, such as a user or company name, applied to a text-based partition key. RLAC provides more granular security for tables so that only authorized users are able to view or modify subsets of the data. The RLAC feature is useful for multi-tenant applications.

Enable row-level access control in LCM for DSE clusters that have tables with row-level permissions required for access. Follow these steps to make Lifecycle Manager (LCM) aware of and allow use of RLAC configured for DSE clusters provisioned and managed by LCM.

**Note:** RLAC is supported in DSE 5.1 and later, and LCM 6.1 and later.

1. Click **Config Profiles** from the Lifecycle Manager navigation menu.

2. Click the edit icon for the config profile you want to edit, or click **Add config profile** if you have not already created a profile.

3. In the Config Profile pane under the **Cassandra** section, click **cassandra.yaml**. In the **Security** pane:
   
   a. Ensure that the **authenticator** setting is **DseAuthenticator**.

   b. Ensure that the **authorizer** setting is **DseAuthorizer**.
4. In the Config Profile pane under the Cassandra section, click `dse.yaml`.

5. Scroll to the DSE Authorizer Options pane.
   a. Select enabled for authorization options.
   b. Select `allow_row_level_security`.

6. Click Save to save the Config Profile.

What's next:

- Go to the Clusters (page 554) workspace in Lifecycle Manager and select the config profile to apply at the cluster, datacenter, or node level. Nodes can inherit config profile settings from the cluster or datacenter levels, or have settings at the node level that take precedence.
- Run a configuration job (page 576) to push the configuration to all of the applicable nodes.
- Restrict the applicable tables rows and grant permissions to the applicable role names using GRANT or REVOKE statements as required for your environment. For details, see Setting row-level permissions in the DSE Administrator documentation.
• Log in as each user role and run queries to confirm that results represent your defined access permissions. See an example.

**Configuring a JMX Connection to DSE using LCM**

Configure a local connection or remote JMX connection to DSE using Lifecycle Manager.

The types of JMX connections are:

- Local JMX - No Authentication
- Remote JMX - No Authentication
- Remote JMX - DSE Unified Authentication

1. Click **Config Profiles** from the Lifecycle Manager navigation menu.

2. Click the edit icon for the configuration profile you want to edit, or click **Add config profile** if you have not already added a config profile.

3. In the Config Profile pane under the Lifecycle Manager section, click **jvm.options**. The general settings for configuring JVM options appears.

4. Scroll down to the JMX pane. Select the connection type.
To enable local JMX Connection, ensure that `jmx-connection-type` is set to `Local-JMX - No Authentication` (default).

To enable remote JMX Connection without authentication, ensure that `jmx-connection-type` is set to `Remote-JMX - No Authentication`.

To enable remote JMX Connection with DSE Unified Authentication, ensure that `jmx-connection-type` is set to `Remote-JMX - DSE Unified Authentication`.

When using DSE Unified Authentication, by default you can access JMX using the name cassandra and the password of the cassandra role from the DSE cluster.

5. Click **Save**.

What’s next:
- Run a configure job ([page 576](#)) to apply the config profile changes.

**Configuring JVM options in LCM**

Configure JVM options from within Lifecycle Manager.

Note: The `jvm.options` file is available for DSE versions 5.1 and later.

For more information on JVM, see [Tuning Java resources](#).

1. Click **Config Profiles** from the Lifecycle Manager navigation menu.

2. Click the edit icon for the configuration profile you want to edit, or click **Add config profile** if you have not already added a config profile.

3. In the **Config Profile** pane under the Lifecycle Manager section, click **jvm.options**.
   The **General JVM Parameters** pane for configuring JVM options appears.
4. Adjust the default settings as appropriate for your environment.

   **Tip:** Click **Show Field Descriptions** to view available tooltips for fields.

5. Arbitrary JVM options can be passed to DSE on startup. To configure any additional JVM options required for your environment, scroll down to the **Advanced** pane and click **+ Add an item**. Enter a single option in each box exactly as it would appear as a Java command-line argument. For example:

   -XX:MaxJavaStackTraceDepth=1000
   -Dcassandra.maxqueuednative_transport_requests=1024
6. Click **Save** to save the Config Profile.

**What's next:**
- Run a configure job ([page 576](#)) to apply the config profile changes.

**Managing Java installs**

Automatically manage Java installs and upgrades within configuration profiles. Managing Java installs is enabled by default in Lifecycle Manager, which downloads the JRE package and accepts the license on your behalf. The JRE or JDK is required for most DataStax products; including DataStax Enterprise (DSE), Lifecycle Manager (LCM), and OpsCenter. You can also point to your own internal download location for packages.

**Note:** DataStax recommends the latest build of a Technology Compatibility Kit (TCK) Certified OpenJDK version 8.

If there is no Java version found on a target during an install job or configuration job, LCM installs the specified version of Java. If a Java version is found but it does not match the specified version, LCM installs the specified version of Java specified alongside any outdated or incorrect version already installed. If managing Java is disabled, and no version of Java is found on the target, the install job fails.

The version fields in the Java setup area of the configuration profiles are used to dynamically construct a URL string that downloads a particular Java version based on major version number, update version number, and build version number. For example, when you check your Java version at the command prompt, you see something like:

```
$ java -version
java version "1.8.0_60"
Java(TM) SE Runtime Environment (build 1.8.0_60-b27)
Java HotSpot(TM) 64-Bit Server VM (build 25.60-b23, mixed mode)
```

In the above example, the **1.8.0** is the major version, **60** is the update version, and **27** is the build. If your organization requires a version other than the default provided, enter the version information in the java-setup page accessed in the **Lifecycle Manager** section of...
the **Config Profile** pane. For more details about Java version naming conventions, see the Oracle documentation.

If LCM is managing your Java installations, it can also manage installations of the Java Cryptography Extension (JCE) Unlimited Strength Policy files. JCE files enable the use of strong encryption suites.

**Note:** Ensure that usage of the JCE files is legal in your jurisdiction. JCE-based products are restricted for export to certain countries by the U.S. Export Administration Regulations.

**Prerequisites:** Create a config profile *(page 600)* (shell of defaults, optional)

1. Click **Config Profiles** from the Lifecycle Manager navigation menu.

2. Click the edit icon for the config profile you want to edit, or click **Add config profile** if you have not already created a profile.

3. In the **Config Profile** pane under the **Lifecycle Manager** section, click **Java Setup**.
   The settings including defaults for managing Java install versions appears. Click the information (i) icon to view information inline for each available field.

4. Make any changes to the available options in the **Java** pane:
   a. To disable automatically managing Java installs, clear the **manage-java** setting. You might want to disable Java management if you have pre-installed your desired version of Java and do not want LCM to alter your Java deployment.
Provisioning skips installing Java when manage-java is disabled. If managing Java is disabled, and Java is not found to be already installed when running an install job (page 574), the install job fails.

When manage-java is enabled, target nodes download Java directly from Oracle servers. To download the Oracle tarball from a local mirror, enter the complete URL to the tarball. In any case, complete the major-version, update-version, build-version, and build-hash (if applicable) fields so that target nodes can verify that the expected version of Java is available.

b. Enter the version of Java to install in the major-version box. For example, 1.8.0. To override the default, enter the applicable major version number.

c. To override the default, enter the specific update number of a version in the update-version box.

Each Java release is identified by an update-version within its filename. For example, the update-version for server-jre-8u121-linux-x64.tar.gz is 121 as indicated by 8u121 (version 8 update 121).

d. To override the default, enter a value for final released build version in the build-version box.

Each Java release is additionally identified by a build number in its download url. For example, in the download-page at https://www.oracle.com/technetwork/java/javase/downloads/server-jre8-downloads-2133154.html, the url for server-jre-8u121-linux-x64.tar.gz includes .../java/jdk/8u121-b13/..., in which the build number is 13 as indicated by b13 (build 13).

e. To override the default, enter a value (or leave blank as appropriate for older Java releases prior to 8u121-b13) for the build hash-version in the hash-version box.

Java releases from 8u121-b13 onward are additionally identified by a build hash-version in their download url. When downloading older Java releases, the build hash-version can be left blank.

f. To download Java from a custom location rather than directly from Oracle, enter the location in the download-url box.

Important: The *-version fields are still required to construct the particular download version from a custom location.

5. Make any changes to the available options in the JCE Policy pane:

a. To disable downloading JCE, clear the manage-jce-policy setting.
b. If your organization has its own mirror, enter the location in the `jce-download-url` box. If left blank and managing JCE policy is enabled, the JCE is downloaded directly from Oracle.

   **Note:** Kerberos, client-to-node encryption, and node-to-node encryption require JCE installed on every node.

6. Click **Save**.

**What's next:**

- Check config profile usage.
- Configure a package proxy *(page 621)* to cache and expedite package downloads.
- Run a configure job *(page 576)* to apply the config profile changes.

**Configuring a proxy for package downloads**

Configure an http or https proxy for expediting package downloads by way of caching, or for an environment where DataStax Enterprise clusters must be isolated offline from the internet. The package proxy is disabled by default. Enable the package proxy in Lifecycle Manager if your organization manages your own http or https caching proxy such as **Squid**.

1. Click **Config Profiles** from the Lifecycle Manager navigation menu.

2. Click the edit icon for the configuration profile you want to edit, or click **Add config profile** if you have not already added a config profile.

3. In the Config Profile pane under the Lifecycle Manager section, click **Package Proxy**.
   
   The general settings for configuring package proxy appears.
4. Complete the following fields:

   a. Select enabled for package proxy.

   b. Select the http protocol that target nodes use to connect to the proxy.

   c. Enter the host, which is the IP address (recommended) or host name of the proxy itself.

   d. Enter the port number that the proxy is listening on if it is not the default port.

   e. Select whether authentication (above and beyond the DataStax or internal repo authentication (page 586)) to the proxy itself is required to forward requests, and enter the required username and password.

      The only type of supported authentication is HTTP Basic Authentication.

5. Click Save.

What's next:

- Be sure to enable Use Configured HTTP Proxy for the appropriate repository (page 586).
- Check config profile usage.
• Run a configure job (page 576) to apply the config profile changes.

Deleting a configuration profile

Delete a configuration profile. Delete a config profile if it is obsolete or if you prefer a node or datacenter to inherit a config profile from its cluster. Make sure a replacement configuration profile is ready to apply in its stead, especially if you are deleting at the cluster level because datacenters and nodes might be inheriting the configuration profile from their parent entity in the topology model.

Prerequisites:

Check its usage beforehand.

Edit the cluster, datacenter, or node to remove the config profile from use. Select another config profile to use in its stead.

1. From the Lifecycle Manager navigation menu, click **Config Profiles** > **Delete** icon for the configuration profile you want to delete.
   
   A dialog prompts you to confirm deleting the profile.

2. Click **Delete**.

Cloning a configuration profile

Clone a config profile as a starting point for similar but differentiated configurations, to use for testing purposes, or when upgrading DSE.

1.

Checking the usage of a configuration profile

Check where a configuration profile is in use within clusters, datacenters, and nodes. Checking the usage of a config profile is useful when editing a profile for a particular workload, before cloning a profile for similar but differentiated uses, or prior to deleting a profile.

1. From the Lifecycle Manager navigation menu, click **Config Profiles** > **Edit** for the configuration profile you want to edit.

2. Click the **Check Usage** tab.

   The Check Usage tab displays the clusters, datacenters, and nodes where the config profile is currently in use.
Lifecycle Manager configuration options

Reference of configuration options in available opscenterd.conf for Lifecycle Manager. After changing properties in the opscenterd.conf file, restart OpsCenter (page 75) for the changes to take effect.

[lifecycle_manager] db_location
   The location of the lcm.db database used for storing Lifecycle Manager information. Default: /var/lib/opscenter/lcm.db

   Note: The data (cluster topology models, configuration profiles, credentials, repositories, job history, and so forth) for Lifecycle Manager is stored in the lcm.db database. Your organization is responsible for backing up the lcm.db database. You must also configure failover (page 169) to mirror the lcm.db.

[lifecycle_manager] cacerts_directory
   The directory to use when storing SSL-related files that are automatically generated during provisioning tasks. Default: /var/lib/opscenter/ssl/lcm

[lifecycle_manager] cacerts_truststore_location
   The location of the truststore containing the root CA cert for signing certificates. Default: /var/lib/opscenter/ssl/lcm/lcm-auto-generated.truststore

[lifecycle_manager] cacerts_truststore_password
   The password to open the cacerts truststore.

[lifecycle_manager] crypto_key_file
   The key to use to encrypt sensitive data in the database for storing Lifecycle Manager information. The encryption key is automatically generated if it does not already exist. Default: /etc/opscenter/lcm.key

[lifecycle_manager] job_manager_polling_interval
   Polling frequency in seconds when performing Lifecycle Manager tasks. Default: 5 seconds.

[lifecycle_manager] node_install_idle_timeout
Duration of inactivity in seconds before marking an install job as idle. Default: 900 seconds.

[lifecycle_manager] node_restart_idle_timeout
Duration of inactivity in seconds before marking a node restart job as idle. Default: 120 seconds.

[lifecycle_manager] node_configure_idle_timeout
Duration of inactivity in seconds before marking a node configuration job as idle. Default: 120 seconds.

[lifecycle_manager] node_cluster_import_idle_timeout
Duration of inactivity in seconds before marking a cluster import job as idle. Default: 120 seconds.

[lifecycle_manager] node_install_agent_idle_timeout
Duration of inactivity in seconds before marking an agent install job as idle. Default: 120 seconds.

[lifecycle_manager] cassandra_connection_timeout
Duration in seconds to allow for connecting to DataStax Enterprise when provisioning a DataStax Enterprise cluster. Default: 20 seconds.

[lifecycle_manager] meld_directory
The path to the meld provisioning script

[lifecycle_manager] opscenterd_agent_package_directory
The directory lcm will find agent packages in for installing DataStax Agents.

[lifecycle_manager] ssh_connect_timeout_in_seconds
The number of seconds to wait for the SSH server on a target node to respond to a connection attempt during an LCM job. Default: 30 seconds.

[lifecycle_manager] ssh_max_attempts
The maximum number of times to attempt an SSH connection to a given target during an LCM job. Default: 3.

[lifecycle_manager] ssh_retry_delay_in_seconds
The number of seconds to wait between SSH connection retries during an LCM job. Default: 10 seconds.

[lifecycle_manager] disable_platform_check
Platform check ensures DSE compatibility and support. Disable at your own risk. Default: False.

eclipse\eclipse.exe

The location of the eclipse\eclipse.exe file depends on the type of installation:

- Package installations: /etc/opscenter/opscenterd.conf
- Tarball installations: install_location/conf/opscenterd.conf

lcm.db

The location of the Lifecycle Manager database lcm.db depends on the type of installation:

- Package installations: /var/lib/opscenter/lcm.db
- Tarball installations: install_location/lcm.db

Note: The data (cluster topology models, configuration profiles, credentials, repositories, job history, and so forth) for Lifecycle Manager is stored in the lcm.db
database. Your organization is responsible for backing up the database. You must also configure failover (page 169) to mirror the lcm.db.
OpsCenter API reference for developers

The OpsCenter API facilitates the development of websites and programs to retrieve data and perform DataStax Enterprise administrative actions. The OpsCenter API provides the ability to make RESTful requests for programmatically performing the same set of operations as the OpsCenter GUI. As of OpsCenter version 6.0 and later, the OpsCenter API reference includes the Lifecycle Manager API for the LCM GUI.

As of OpsCenter version 6.1 and later, there is an API reference for the DataStax Agent. See Enable the Swagger UI (page 627).

Enable and access the DataStax Agent API

Enable and access the interactive Swagger UI console for the DataStax Agent API. Explore the console to learn about the agent APIs available for scripting purposes, or to use directly with a curl command (page 628).

address.yaml

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

- Package installations: /var/lib/datastax-agent/conf/address.yaml
- Tarball installations: install_location/conf/address.yaml

cluster_name.conf

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

- Package installations: /etc/opscenter/clusters/cluster_name.conf
- Tarball installations: install_location/conf/clusters/cluster_name.conf
1. Open address.yaml for editing.

2. Set the `swagger_enabled` option to `true`.

3. Restart the agents *(page 76)*.

4. Navigate to the UI:

   ```
   http(s)://{agent_ip}:{api_port}/ui
   ```

   Use `https` or `http` as appropriate for your environment. Replace `agent_ip` with the IP address of the particular agent. Replace the `api_port` with the port configured for your environment. Typically, the `api_port` is 61621. Configure the API port using the `api_port` *(page 211)* option in `address.yaml` or `[agents] api_port *(page 200)* in `cluster_name.conf`.

**DataStax Agent API example curl commands**

**Examples**

Use the DataStax Agent API with cURL commands to directly hit an agent for useful information when troubleshooting an issue. Check the size of a diagnostic file before downloading. Determine the root cause of connection issues. Some useful `curl` commands:


- If you see connection-status failed messages in the `opscenterd.log`, get more details using `curl "http://54.152.37.135:61621/v1/connection-status" | json_pp` (piped to JSON pretty print):

  ```
  $ curl "http://54.152.37.135:61621/connection-status" | json_pp
  % Total    % Received % Xferd Average Speed   Time    Time  Time  Time
  Time  Time Left  Speed  Current   Dload  Upload Total Spent
  100   935  100   935    0     0   3571      0 --:--:-- --:--:--  3582
  "storage_cassandra" : {
      "config" : {
        "storage-keyspace" : "OpsCenter",
        "storage-port" : 9042,
        "storage-hosts" : [
            "172.31.6.241"
        ],
        "storage-cassandra-user" : null,
      }
  }
  ```
"error" : "Authentication error on host /172.31.6.241:9042: Host /172.31.6.241:9042 requires authentication, but no authenticator found in Cluster configuration",
"started" : false,

"install-type" : "package",
"stomp" : {
    "started" : true,
    "config" : {
        "error" : null,
        "stomp-interface" : "172.31.0.97",
        "stomp-port" : 61620,
        "use-ssl?" : false
    }
},

"monitored_cassandra" : {
    "config" : {
        "monitored-cassandra-user" : null,
        "monitored-hosts" : [
            "172.31.6.241"
        ],
        "error" : "Authentication error on host /172.31.6.241:9042: Host /172.31.6.241:9042 requires authentication, but no authenticator found in Cluster configuration",
        "monitored-port" : 9042
    },
    "started" : false
},

"http" : {
    "config" : {
        "agent-rpc-interface" : "172.31.6.241",
        "api-port" : 61621,
        "use-ssl" : false
    }
},

"jmx" : {
    "started" : true,
    "config" : {
        "jmx-port" : 7199,
        "error" : null,
        "jmx-user" : null
    }
}

opscenterd.log

The location of the opscenterd.log file depends on the type of installation:

- **Package installations:** /var/log/opscenter/opscenterd.log
- **Tarball installations:** install_location/log/opscenterd.log