OpsCenter 6.0 User Guide
(Earlier version)
# Table of Contents

**About OpsCenter** ........................................................................................................................ 10  
  New features ................................................................................................................................. 10  
  DataStax OpsCenter Policy Changes ............................................................................................ 15  
  Key features ................................................................................................................................. 16  
  OpsCenter architecture overview ................................................................................................. 17  
  Send DataStax feedback about OpsCenter .................................................................................... 18  

**OpsCenter Release Notes** ......................................................................................................... 20  
  OpsCenter 6.0.12 Release Notes .................................................................................................. 20  
  OpsCenter 6.0.11 Release Notes .................................................................................................. 21  
  OpsCenter 6.0.10 Release Notes .................................................................................................. 21  
  OpsCenter 6.0.9 Release Notes .................................................................................................... 23  
  OpsCenter 6.0.8 Release Notes .................................................................................................... 25  
  OpsCenter 6.0.7 Release Notes .................................................................................................... 26  
  OpsCenter 6.0.6 Release Notes .................................................................................................... 27  
  OpsCenter 6.0.5 Release Notes .................................................................................................... 29  
  OpsCenter 6.0.4 Release Notes .................................................................................................... 30  
  OpsCenter 6.0.3 Release Notes .................................................................................................... 33  
  OpsCenter 6.0.2 Release Notes .................................................................................................... 36  
  OpsCenter 6.0.1 Release Notes ..................................................................................................... 38  
  OpsCenter 6.0.0 Release Notes ..................................................................................................... 39  

**Installing OpsCenter** .................................................................................................................. 48  
  Installing OpsCenter from the RPM package ............................................................................. 48  
  Installing OpsCenter from the Debian package ......................................................................... 49  
  Installing OpsCenter with a tarball on any Linux distribution .................................................. 51  
  Uninstalling OpsCenter ................................................................................................................ 52  
  Installing DataStax Agents .......................................................................................................... 53  
    Installing DataStax agents automatically .................................................................................... 54
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manually installing DataStax agents</td>
<td>57</td>
</tr>
<tr>
<td>Setting permissions to run the agent as a different user</td>
<td>60</td>
</tr>
<tr>
<td>Configuring JAVA_HOME for DataStax agents</td>
<td>62</td>
</tr>
<tr>
<td>OpsCenter Reference</td>
<td>62</td>
</tr>
<tr>
<td>OpsCenter ports reference</td>
<td>62</td>
</tr>
<tr>
<td>Installation and configuration locations</td>
<td>69</td>
</tr>
<tr>
<td>Starting, stopping, and restarting OpsCenter</td>
<td>69</td>
</tr>
<tr>
<td>Starting and restarting DataStax agents</td>
<td>70</td>
</tr>
<tr>
<td>Upgrading OpsCenter</td>
<td>72</td>
</tr>
<tr>
<td>Configuring OpsCenter</td>
<td>73</td>
</tr>
<tr>
<td>OpsCenter Security</td>
<td>73</td>
</tr>
<tr>
<td>Configuring SSL/TLS between OpsCenter and the DataStax Agents</td>
<td>73</td>
</tr>
<tr>
<td>Connect to DSE with client-to-node encryption in OpsCenter and the DataStax Agents</td>
<td>80</td>
</tr>
<tr>
<td>Enabling HTTPS for the OpsCenter server</td>
<td>83</td>
</tr>
<tr>
<td>Configuring OpsCenter for Kerberos authentication</td>
<td>86</td>
</tr>
<tr>
<td>Editing OpsCenter cluster connections for authentication or encryption</td>
<td>90</td>
</tr>
<tr>
<td>Configuring role-based security</td>
<td>92</td>
</tr>
<tr>
<td>Authenticating OpsCenter users with LDAP</td>
<td>102</td>
</tr>
<tr>
<td>Encrypting sensitive configuration values</td>
<td>112</td>
</tr>
<tr>
<td>Configuring security logging</td>
<td>117</td>
</tr>
<tr>
<td>Configuring alerts for events</td>
<td>118</td>
</tr>
<tr>
<td>SNMP alerts overview</td>
<td>119</td>
</tr>
<tr>
<td>Enabling SMTP email alerts</td>
<td>123</td>
</tr>
<tr>
<td>Enabling alerts posted to a URL</td>
<td>125</td>
</tr>
<tr>
<td>Configuring data collection and expiration</td>
<td>131</td>
</tr>
<tr>
<td>Controlling data collection</td>
<td>131</td>
</tr>
<tr>
<td>Storing collection data on a separate cluster</td>
<td>133</td>
</tr>
<tr>
<td>OpsCenter definition files for supporting updated DSE versions</td>
<td>138</td>
</tr>
</tbody>
</table>
Updating and configuring definitions files properties................................. 139
Automatic failover overview........................................................................... 141
Enabling automatic failover........................................................................... 143
Failover configuration options reference......................................................... 146
Back up critical configuration data................................................................. 146
Configuring named route linking................................................................... 148
Sending OpsCenter metrics to a Graphite server........................................... 149
Configuring the OpsCenter JVM.................................................................... 150
Configuring the DataStax Agent JVM.............................................................. 151
Setting and securing the tmp directory for the agent....................................... 152
Configuration files.......................................................................................... 153
OpsCenter configuration properties................................................................. 154
Cluster configuration properties....................................................................... 167
DataStax Agent configuration........................................................................... 176
OpsCenter logback.xml configuration.............................................................. 185
Customize scripts for starting and stopping DataStax Enterprise.................. 188
Example configuration scenarios...................................................................... 188
Configuring for multiple regions..................................................................... 188
Configuring for very large clusters................................................................. 190
Using OpsCenter............................................................................................. 193
OpsCenter Workspace Overview..................................................................... 193
Node monitoring and administration............................................................... 195
Ring View......................................................................................................... 195
List View........................................................................................................... 199
Agents View.................................................................................................... 201
Node management operations......................................................................... 206
Managing keyspaces and tables..................................................................... 219
Keyspaces........................................................................................................ 219
Browsing data........................................................................................................................................... 223
Cluster administration.............................................................................................................................. 223
  Adding an existing cluster................................................................................................................... 223
  Removing a cluster from OpsCenter and Lifecycle Manager monitoring........................................... 227
Rebalancing a cluster overview............................................................................................................ 228
Restarting a cluster.................................................................................................................................. 230
Changing the display name of a cluster................................................................................................. 232
Downloading diagnostic data.................................................................................................................. 233
Generating a cluster report.................................................................................................................... 234
Dashboard performance metrics........................................................................................................... 235
  Performance metrics overview........................................................................................................ 235
  Cluster performance metrics............................................................................................................ 242
Pending task metrics.............................................................................................................................. 243
Table performance metrics................................................................................................................... 246
Tiered storage performance metrics....................................................................................................... 247
Search performance metrics.................................................................................................................. 250
Graph metrics.......................................................................................................................................... 250
Operating system performance metrics............................................................................................... 254
Alert metrics.............................................................................................................................................. 256
OpsCenter Metrics Tooltips Reference.................................................................................................. 263
DSE Management Services.................................................................................................................... 277
Backup Service....................................................................................................................................... 277
  Backup Service Overview................................................................................................................. 277
  Why use the Backup Service?........................................................................................................... 280
Configuring the Backup Service........................................................................................................... 281
Backing up a cluster............................................................................................................................... 287
Restoring a cluster.................................................................................................................................. 297
Cloning cluster data............................................................................................................................... 309
Backing up and restoring DSE Graphs in OpsCenter..................................................318
Viewing backup and restore history...........................................................................321
Deleting backup data................................................................................................322
Troubleshooting Backup Service errors...................................................................323
Repair Service...........................................................................................................324
How the Repair Service works..................................................................................324
Incremental repairs...................................................................................................326
Starting and stopping the Repair Service.................................................................327
Checking repair progress.........................................................................................331
Advanced Repair Service configuration.................................................................331
Troubleshooting Repair Service errors....................................................................333
Capacity Service.......................................................................................................334
Forecasting trends for metric graphs........................................................................335
Advanced forecast configuration.............................................................................336
Best Practice Service.................................................................................................337
Configuring Best Practice service rules.................................................................338
Monitoring the results of Best Practice service scans..........................................338
Best Practice Rules Reference................................................................................339
Performance Service.................................................................................................352
Performance Service Overview..............................................................................352
Why use the OpsCenter Performance Service?......................................................355
Enabling the OpsCenter Performance Service.......................................................356
Disabling the OpsCenter Performance Service......................................................358
Setting permissions for the OpsCenter Performance Service...............................358
Tuning a database cluster with the Performance Service........................................360
Troubleshooting......................................................................................................379
High CPU usage by opscenterd..............................................................................379
Troubleshooting SSL..............................................................................................380
Troubleshooting SSL connections to Cassandra............................................. 380
Troubleshooting SSL validation for self-signed certificates................................. 382
Data must be a byte string pyopenssl error...................................................... 382
Browser issues................................................................................................. 382
Zero nodes detected in cluster or Loading OpsCenter screen hanging................. 382
Internet Explorer web browser not supported.................................................... 383
Troubleshooting SSTables.................................................................................. 383
Problems running sstableloader......................................................................... 383
The SSTables in this snapshot ‘<tag>’ are not compatible....................................... 384
Troubleshooting collecting or displaying metrics................................................ 384
  Limiting the metrics collected by OpsCenter.................................................... 384
OpsCenter data growing too large...................................................................... 384
Cannot create a keyspace.................................................................................... 385
DataStax agent port setting conflict.................................................................. 385
Java not installed or JAVA_HOME environment variable not set.......................... 385
Insufficient user resource limits errors............................................................... 385
Installing EPEL on CentOS 5.x or RHEL 5.x...................................................... 386
Error getting version update information............................................................ 386
OpsCenter cannot start up.................................................................................. 386
Lifecycle Manager............................................................................................ 387
Lifecycle Manager Overview.............................................................................. 387
  Accessing OpsCenter Lifecycle Manager......................................................... 388
  Getting started workflow.................................................................................. 390
Lifecycle Manager: Clusters.............................................................................. 391
  Cluster Topology Overview............................................................................. 391
  Importing an unmanaged cluster..................................................................... 407
  Cloning a managed cluster, datacenter, or node.............................................. 409
Run an Install and Configure Jobs Overview.................................................... 410
About OpsCenter

DataStax 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 387) (LCM) provisions DataStax Enterprise (DSE) clusters and centrally manages cluster configurations.

The OpsCenter Compatibility chart (page 302) indicates which versions of OpsCenter are compatible with the various DataStax Enterprise versions.

New features in DSE OpsCenter 6.0

New features

The following new and improved features are highlighted for the current DataStax Enterprise (DSE) OpsCenter version 6.0 release.
### Table 1: New and improved OpsCenter features

<table>
<thead>
<tr>
<th>Lifecycle Manager <em>(page 387)</em></th>
<th>Deploy and centrally manage configurations for DataStax Enterprise clusters version 4.7 and later using Lifecycle Manager <em>(page 387)</em>:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Efficiently prevent configuration drift by defining configuration profiles <em>(page 428)</em> that apply to the cluster, datacenter, or node level. Enforce configurations that adhere to the desired baseline configurations for datacenter workloads.</td>
</tr>
<tr>
<td></td>
<td>• Manage SSH credentials <em>(page 426)</em> for machine access when provisioning and managing clusters. Mix and match credential assignments between different clusters and datacenters. Create repositories to access the DataStax Repository or your own mirror for downloading DataStax Enterprise packages. Credential information is encrypted and securely stored.</td>
</tr>
<tr>
<td></td>
<td>• After running <em>(page 410)</em> an install, configure, or import cluster job; view the Jobs Summary <em>(page 415)</em> and drill into details for deep transparency on the status and history of all 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 <em>(page 415)</em> without the immediate need to scour various logs for information. Agent install jobs are also visible in the Jobs workspace. Troubleshoot and terminate <em>(page 419)</em> any jobs that take an excessive time to execute.</td>
</tr>
<tr>
<td>DataStax Enterprise 5.0</td>
<td>Support for provisioning and monitoring DataStax Enterprise 5.0 *(page ) clusters:</td>
</tr>
<tr>
<td></td>
<td>• Support for provisioning and monitoring DSE Graph *(page ). See Configuring Lifecycle Manager for DSE Graph <em>(page 438)</em>.</td>
</tr>
<tr>
<td></td>
<td>• Performance metrics <em>(page 247)</em> for monitoring <em>(page 248)</em> DSE Tiered Storage. Configure the strategy and tiers <em>(page 440)</em> within a Lifecycle Manager config profile.</td>
</tr>
<tr>
<td></td>
<td>• OpsCenter support for monitoring DSE Multi-Instance *(page ) nodes. See Configuring DataStax Agents for Multi-Instance Nodes <em>(page 184)</em> to take advantage of monitoring multi-instance nodes.</td>
</tr>
<tr>
<td>DataStax agents overhaul</td>
<td>Simplified and streamlined the installing, upgrading, and troubleshooting of DataStax agents into a superlative workflow:</td>
</tr>
<tr>
<td></td>
<td>• DataStax agent installs <em>(page 53)</em>: Revamped agent architecture and vastly improved the automatic agent install <em>(page 54)</em> experience.</td>
</tr>
<tr>
<td></td>
<td>• Agent Status <em>(page 201)</em>: More comprehensive details about agent status display in the Agents tab within the Nodes section of the OpsCenter monitoring UI.</td>
</tr>
<tr>
<td></td>
<td>• Troubleshoot <em>(page 204)</em> agent issues. Launch the set up and upgrade agent process from within the Agents view.</td>
</tr>
<tr>
<td></td>
<td>• Add an Agent Issues alert <em>(page 203)</em> and configure alert notification <em>(page 118)</em> for additional awareness of agent issues.</td>
</tr>
</tbody>
</table>
Significant enhancements for alerts include:

- **SNMP (page 119):** SNMP (Simple Network Management Protocol) trap alerts support for monitoring events.
- **Configurable content type and fields for POST URL alerts (page 125):** Both JSON and form URL-encoded content types are supported.
- **Miscellaneous convenient alert plugin enhancements:**
  - Cluster-level granularity available for all alert plugins. Configure alert notifications by one or more or all clusters. Indicate specific clusters for SNMP, email, and POST URL alerts. By default, alerts are fired for events on all clusters.
  - More flexible email alert templates, including full support for multiple email addresses per configuration.

Add backup locations on local filesystems (page 293). See adding a backup location (page 293) for more information on the available backup location options.

Configure logback.xml (page 185) in OpsCenter to suit your logging requirements. Configure security logging (page 117) to record user activity in OpsCenter.

Send metrics (page 149) collected by OpsCenter to your configured Graphite monitoring solution. Graphite server support is an OpsCenter monitoring labs feature in development available for use now.

**Updates from previous OpsCenter versions**

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

**Java version**

Oracle Java SE Runtime Environment 8 (JRE or JDK) or OpenJDK 8. Earlier or later versions are not supported. See installing the Oracle JDK (page 7) or OpenJDK.

**SSL configuration changes**

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

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

See troubleshooting SSL connections (page 1).

**SSL certificate chains issue**

SSL certificate chains do not work properly in OpsCenter versions 6.0.0, 6.0.1, and 6.0.2. OpsCenter does not start up if HTTPs is enabled and the SSL certificates use intermediate
certificates (CA chains). For details about the opscenterd.log errors as a result of this issue, refer to the KB article available from DataStax Support.

Upgrading to 6.0.3 is necessary to alleviate the issue. A patch is also available from DataStax Support.

**LDAP configuration changes**

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

- `ssl_cacert`
- `ssl_cert`
- `ssl_key`
- `tls_reqcert`
- `tls_demand`
- `debug_ssl`
- `opt_referrals`

The removed configuration options have been replaced with:

- `truststore`
- `truststore_type`
- `truststore_pass`

The optional LDAP configuration option `user_memberof Stores_dn` has been added for OpsCenter version 6.0.9 and later. If your organization had difficulty getting LDAP to work with `memberof_search` rather than a `directory_search`, try configuring with the `user_memberof Stores_dn`. For more information, see configuring LDAP (page ).

**User password hash for OpsCenter authentication**

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

**Password database ownership**

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

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

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

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

Kerberos configurations

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

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

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

New configuration options were added to address.yaml:

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

**Diagnostic tarball configurable timeout**

The diagnostic_tarball_download_timeout configuration option has been added to allow configuring a timeout when generating a diagnostics tarball. Increasing the default value might be necessary on slower machines or for multi-instance clusters.
The `tarball_process_timeout` option has been removed. The option was actually an agent installation option that is no longer used due to improvements in the agent installation workflow.

**Deprecated OpsCenter APIs**

The following methods have been removed from Managing Cluster Configurations:

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

**Warnings on deprecated DataStax Enterprise metrics**

After upgrading a DataStax Enterprise cluster, OpsCenter detects any obsolete metrics in use within dashboard graph presets or alert rules. When first starting OpsCenter after an upgrade, warning icons indicate which graphs have unknown metrics. See Working with metrics performance graphs (page 271) for information about deleting unknown metrics.

**Metrics inserted asynchronously**

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

- `async_queue_size`
- `async_pool_size`

**Disabling all Best Practice Service rules**

If all Best Practice rules are disabled, the Best Practice Service (page 441) is considered disabled by OpsCenter. Any new Best Practice rules are not enabled by default.

**DataStax OpsCenter Policy Changes**

Starting with OpsCenter version 6.0, OpsCenter will only be compatible with DataStax Enterprise (DSE) clusters. DataStax will discontinue OpsCenter compatibility with:

- Open Source Software (OSS) Cassandra clusters
- DataStax Distributions of Cassandra (DDC) clusters, formerly known as DataStax Community (DSC)
Customers currently using OpsCenter to provision, manage, and maintain their OSS Cassandra, DSC, and DDC clusters must look for other tools to continue their management and maintenance activities. Please see Planet Cassandra for some alternatives.

## 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 235)*.  
|                      | Ability to add and edit graphs *(page 236)* in the dashboard.  
|                      | Monitoring capabilities of DSE In-Memory tables *(page 210)*.  
|                      | View the Spark console *(page 214)*.  
|                      | Hadoop Job Tracker integration. |
| Configuration, security, and administration | Basic cluster configuration *(page 73)*.  
|                       | Security *(page 73)* 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 223)*, using simple point-and-click actions.  
|                       | Multiple cluster management from a single OpsCenter instance using agents *(page 53)*.  
|                       | Automatic failover *(page 141)* from the primary OpsCenter to a backup OpsCenter instance.  
|                       | Rebalance *(page 228)* data across a cluster after adding or removing non-virtual nodes.  
|                       | Manage multiple nodes simultaneously *(page 206)* for certain bulk operations.  
|                       | Downloadable Cluster report *(page 234)*.  
|                       | Generate a diagnostics tarball *(page 233)* to send to support for further troubleshooting. |
| Alerts *(page 118)* | Alert warnings of events and impending issues.  
|                      | Built-in external notification capabilities and customizable templates. |
### Metrics (page 235)

Metrics are collected (page 131) 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.

### DataStax Enterprise Management Services

DSE Management Services (page 277):
- **Backup Service (page 277)** - allows automatic or manual backup and restore of data in clusters.
- **Repair Service (page 324)** - continuously runs and performs repair operations across a DataStax Enterprise cluster.
- **Capacity Service (page 334)** - understand cluster performance trends at a glance and plan for future capacity with forecasting.
- **Best Practice Service (page 337)** - schedule pre-defined best practice rules that check various properties of clusters and environments.
- **Performance Service (page 352)** - monitor performance metrics and quickly troubleshoot issues with suggested recommendations.

### Lifecycle Manager (page 387)

Simplify deploying DataStax Enterprise clusters. Centrally manage cluster, datacenter, and node configuration.

### 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.
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 (page ).

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 DataStax 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.

OpsCenter 6.0.12 Release Notes

Compatibility

To see which versions of DataStax Enterprise are supported with OpsCenter 6.0, refer to the OpsCenter Compatibility chart. For upgrade instructions, see the DataStax OpsCenter Upgrade Guide.

Be sure to check out the New features section for more details.

Repair Service

- Fixed an issue where restarting OpsCenter would not restart Repair Service. (OPSC-12318)

Core

- Updated project dependencies to no longer be susceptible to CVE-2016-1000031. (OPSC-13404)
- Updated Mina library to address issues with high CPU usage in opscenterd. (OPSC-13678)
- Fixed an issue with agent startup related to networking components. (OPSC-13825)

Backup Service

- Corrected an issue that prevented restores to DSE 4.8 clusters with optional client-to-server ssl. (OPSC-13155)

Provisioning

- LCM now allows debug log to be disabled through the logback.xml configuration. (OPSC-13109)
- 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)
• Fixed an issue where the LCM default download URLs for Oracle Java were no longer valid. (OPSC-13832)

**OpsCenter 6.0.11 Release Notes**

**Compatibility**

To see which versions of DataStax Enterprise are supported with OpsCenter 6.0, refer to the [OpsCenter Compatibility chart](#). For upgrade instructions, see the [DataStax OpsCenter Upgrade Guide](#).

**Core**

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

**Lifecycle Manager (LCM)**

• Fixed issue where Oracle no longer hosts the version of Java that LCM attempts to download by default. (OPSC-13332)
• Fixed issue where LCM is unable to install JCE unlimited cryptography policies for Java 8 u151 and u152. (OPSC-13332)

**OpsCenter 6.0.10 Release Notes**

**Compatibility**

To see which versions of DataStax Enterprise are supported with OpsCenter 6.0, refer to the [OpsCenter Compatibility chart](#). For upgrade instructions, see the [DataStax OpsCenter Upgrade Guide](#).

**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 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)
• OpsCenter login page escapes html characters on login error message, preventing XSS attacks. (OPSC-11843)
• Added the X-XSS-Protection to content responses to enable the XSS auditor in various supported browsers. (OPSC-11845)
• Attempting to login 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)
• Updated solr-cores and range-list routes in agent to address intermittent errors. (OPSC-11674)
• Added Coordinator Read Latency table metric. (OPSC-11509)
• Fixed a bug during agent install that is triggered when no events have been reported. (OPSC-11640)
• 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 an issue where requests would timeout when using AD 2012 with SSL. (OPSC-11995)
• Fixed bug where dashboard would not display when non-ASCII characters present in username. (OPSC-12064)
• Fixed an issue which prevented CQL Solr queries from being reflected in OpsCenter metrics. (OPSC-12071)

Backup Service
• Added support for cloning from a backup using Local FS. (OPSC-6135)
• Removed checks preventing users from restoring backups across major versions. This was in place because previous, 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)
• 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 an issue where materialized views were offered as tables that could be restored directly. (OPSC-11917)
• Fixed an issues that caused the destination dialog to appear multiple times when restoring from an ad hoc S3 backup. (OPSC-11569)
• Fixed an issue where commit logs would be transferred to S3 incomplete. Users with commit log archiving currently active will need to disable then enable commit log archiving for this change to take effect. (OPSC-11903)
• Fixed an issue that required the user to provide a value for local_interface when configuring agents for use on dense nodes. (OPSC-11777)
• Fixed an issue that prevented some snapshots from being displayed in the restore dialog. (OPSC-12168)

Lifecycle Manager (LCM)
• 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)
• Added support to detect and log when openssl and keytool binaries are missing on the LCM node. (OPSC-12210)
• Eliminated SQLITE_BUSY errors when using LCM. (OPSC-11885)

OpsCenter 6.0.9 Release Notes

Compatibility

To see which versions of DataStax Enterprise are supported with OpsCenter 6.0, see the OpsCenter Compatibility chart (page ).

For upgrade instructions, see the DataStax OpsCenter Upgrade Guide (page ).

Known Issues

**Important:** Please be sure to review the list of Known Issues in OpsCenter 6.0 (page 45) before running on a production cluster.

Core

• 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)
• Improved error messaging when the OpsCenter keystore file cannot be read due to bad permissions. (OPSC-9300)
• Added support for using rpc_interface or listen_interface to extract address information rather than rpc_address or listen_address. (OPSC-10873)
• Added http headers to defend against browser-based attacks involving framing and content-sniffing. Embedding OpsCenter within an iframe is no longer possible. (OPSC-11393)
• Added the find-keytool file to the OpsCenter tarball install. That file was added to DEB and RPM packages in OpsCenter version 6.0.6. (OPSC-11627)
• The login page to OpsCenter now benefits from a higher api timeout value, which allows for slower authentication methods. (OPSC-10664)
• The LDAP search_password is properly redacted at the DEBUG logging level. (OPSC-11230)
• Fixed memberof search support when memberof attribute stores a DN (set user_memberof_stores_dn = True in LDAP section of opscenterd.conf). (OPSC-9651)
• Fixed an issue where calculating the next run time of Best Practice rules caused timeouts on the Services page and Best Practice page. (OPSC-10053)
• Fixed the use of cluster names in API calls. (OPSC-10408)
• Fixed a connection leak when the OpsCenter UI was reloaded. (OPSC-11246)
• Fix a diagnostics tarball failure on a stats exception. (OPSC-11253)
• The OpsCenter UI now fully respects the session timeout specified under [authentication] in opscenterd.conf. (OPSC-11410)
• Fixed a potential SQL injection exploit in the OpsCenter SQL authentication (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)
Lifecycle Manager (LCM)

- Improved error messages when meld fails to communicate over HTTP (whether communicating back to LCM, externally to download the JRE, and so forth). (OPSC-8327)
- Improved error reporting when Lifecycle Manager initialization failures prevent OpsCenter from starting. (OPSC-10487)
- Lifecycle Manager jobs give a useful error when 2 or more nodes share the same SSH management address. (OPSC-10992)

Monitoring

- Added Stream Data In/Out metrics for monitoring data streamed between nodes historically. (OPSC-11625)
- The OpsCenter Performance Service exposes the users who are running detected slow queries. (OPSC-10056)
- There is a new property in the posturl.conf file, request_timeout, that can be set higher if customers are seeing alert requests timing out. The default value is the same as the value for connection_timeout and should be suitable for most cases. (OPSC-10907)
- Fixed an issue resulting in incorrect parsing of compaction updates. (OPSC-10862)
- Fixed a bug causing permanent loss of a small amount of metric data when opscenterd was stopped. (OPSC-11293)

Backup Service

- The Backup Service verifies there is sufficient disk space for a restore. Any nodes with insufficient disk space and the amount of space required are displayed in the Restore Report. (OPSC-4543)
- The OpsCenter Backup Service now accepts an optional ending slash / on Local FS destination paths. (OPSC-10681)
- In cases where the agent didn't have write permissions to the Cassandra data directories, the backup service was failing silently and indicating success. This caused restores from those backups to fail when trying to load the schema.json file for a keyspace. This issue has been resolved and backups will fail when the permissions of the Cassandra data directories are incorrectly configured. (OPSC-11499)
- The backup schedule UI now updates when a backup completes. (OPSC-11260)
- Fixed an OpsCenter UI overlap fix on the Backup Service Activity page. (OPSC-10954)
- Fixed an issue where backup destinations were being logged as invalid config keys. (OPSC-11458)
- Fixed an issue where empty keyspaces in a snapshot to a destination would prevent the snapshot from being restored. (OPSC-11513)
- Fixed an issue that caused Solr cores to not be recreated on restore under some configurations. (OPSC-11527)
- Fixed S3 bucket validation delays that led to timeouts in Backup and Restore operations. (OPSC-11574)
• Fixed an issue on timeouts at the beginning of the restore process, where the Backup Service unnecessarily gathered the size of each SSTable in the S3 bucket before performing a restore. (OPSC-11688)

Repair Service
• Added a guard against OpsCenter starting the Repair Service after it has already been started. (OPSC-12116)
• Fixed an issue for OpsCenter versions 6.0.6 through 6.0.8 that repair operations specified at the table level ran on the keyspace level. (OPSC-11584)

OpsCenter 6.0.8 Release Notes

Compatibility
To see which versions of DataStax Enterprise are supported with OpsCenter 6.0, see the OpsCenter Compatibility chart (page ).

For upgrade instructions, see the DataStax OpsCenter Upgrade Guide (page ).

Known Issues

Important: Please be sure to review the list of Known Issues in OpsCenter 6.0 (page 45) before running on a production cluster.

Core
• Removed the retry_delay config option from the [cassandra] and [storage_cassandra] sections of cluster_name.conf because the option was unused. (OPSC-10666)
• RN: Added rolling_restart_retry_delay and rolling_restart_retry_attempts config options to the [cassandra] section of cluster_name.conf to allow customizing the timeout when waiting for nodes to restart. (OPSC-10666)
• All memberOf attribute values are now evaluated during LDAP memberOf authentication. (OPSC-10893)
• Fixed an issue where setting the kerberos_keytab_location and kerberos_client_principal configuration options in address.yaml prevented the agent from starting. (OPSC-10860)
• Removed MaxPermSize from the JVM_OPTIONS for OpsCenter. (OPSC-10878)
• Fixed an issue that prevented automatic detection of a change in Spark masters in some situations. (OPSC-10288)
• Fixed an issue where an agent running on a distressed or down DSE node would improperly handle resources, which resulted in high CPU usage. (OPSC-10895)
• The datastax-agent package can now be installed on centrify-managed systems that have users and groups preconfigured. (OPSC-11002)
• Fixed a security issue where an url injection could be used to redirect to any page after login. (OPSC-11105)
• Fixed an issue where finding the location of Java would sometimes fail prematurely. (OPSC-11227)

Lifecycle Manager (LCM)
• Lifecycle Manager sets the rack name correctly during cluster import. (OPSC-10639)
• Fixed a bug where Use Configured HTTP Proxy could not be unchecked in the Repository dialog. (OPSC-11072)

Monitoring
• Added LiveScannedHistogram metric to track the number of cells scanned during a read. (OPSC-10786)
• The disk partition name now appears in disk-related alerts. (OPSC-10514)
• Fixed a bug in the message format of push notifications for histogram-based alerts. (OPSC-10798)
• Fixed SNMP RFC adherence by adding sysUpTime to trap data. (OPSC-10927)

Backup Service
• Fixed an issue where corrupt S3 backups caused a null-pointer exception; a helpful error message is returned instead of an NPE. (OPSC-10386)
• Fixed an issue that would prevent restoring from Other Location from displaying errors and snapshots. (OPSC-10791)

Repair Service
• Better handle legacy JMX (prior to DSE 5.0) and current JMX repairs. (OPSC-11146)
• Removed an internal ID cache timeout that caused a repair that ran over ten minutes to timeout after an hour. (OPSC-11186)
• Fixed an uncaught exception when invoking get_num_repair_tasks by preventing NoneType object is not iterable exception when seed nodes are unavailable. (OPSC-11047)

OpsCenter 6.0.7 Release Notes

Compatibility
To see which versions of DataStax Enterprise are supported with OpsCenter 6.0, see the OpsCenter Compatibility chart (page ).

For upgrade instructions, see the DataStax OpsCenter Upgrade Guide (page ).

Known Issues

Important: Please be sure to review the list of Known Issues in OpsCenter 6.0 (page 45) before running on a production cluster.

LDAP Security
• Fixed an LDAP login issue when anonymous bind is enabled on the LDAP server. (OPSC-11111)
OpsCenter Release Notes

Note: DataStax recommends upgrading to this new version 6.0.7 to avoid any potential security issues. Please contact DataStax Support for more details.

OpsCenter 6.0.6 Release Notes

Compatibility

To see which versions of DataStax Enterprise are supported with OpsCenter 6.0, see the OpsCenter Compatibility chart (page 28).

For upgrade instructions, see the DataStax OpsCenter Upgrade Guide (page 34).

Known Issues

Important: Please be sure to review the list of Known Issues in OpsCenter 6.0 (page 45) before running on a production cluster.

Core

- Added support for Debian 8 (Jessie). (OPSC-9345)
- Improved Repair dialog keyspace picker. (OPSC-8157)
- Added a check during opscenterd startup that checks the permissions on the system temp directory and displays an error if opscenterd does not have access. The old behavior was to store the temporary files in /tmp but if opscenterd did not have permissions to write, read, or execute on that directory, then opscenterd would not start and would fail to log any errors. (OPSC-9446)
- Improved logging if there is an error when connecting to the DSE cluster. (OPSC-10725)
- The ldap truststore_pass config property will now be encrypted properly. (OPSC-10655)
- OpsCenter will now properly redirect to its login page when an unauthenticated user tries to view the spark master url through OpsCenter. (OPSC-10418)
- OpsCenter now properly decrypts the LDAP search_pass parameter for LDAP connections. (OPSC-10700)
- Fixed OpsCenter post-install script for Debian version 8, Jessie. (OPSC-10564)
- Fixed an issue where the /logout endpoint did not expire the server-side session. (OPSC-5945)
- Fixed an issue where OpsCenter would incorrectly log that a node has moved when the cluster was using vnodes. (OPSC-7620)
- Fixed an issue which prevented some config overrides (including CASSANDRA_CONF) from being respected in some cases. (OPSC-10511)
- Fixed an issue where a rolling restart would not complete in OpsCenter if client-to-node encryption was enabled on the cluster and there was a separate storage cluster. (OPSC-10687)
- Fixed an issue where OpsCenter failed to retrieve the diagnostic tarballs from the agents if SSL was enabled between OpsCenter and the agents. (OPSC-10701)
• Fixed an issue where if a Cassandra query failed in OpsCenter or the DataStax Agent, it would not be retried. This issue could cause the DataStax Agent to fail to respond to any http requests and eventually lead to high CPU on the process. (OPSC-10868)
• Fixed editing of keyspace settings. (OPSC-10793)

Lifecycle Manager (LCM)
• After modifying the Config Profile of an existing cluster in LCM, the Cluster Connection Settings in OpsCenter are now automatically updated after running a configure job. (OPSC-8544)
• LCM errors quickly and with a clear message in cases where custom directories and mount-points are missing. (OPSC-9148)
• LCM now includes tooltips that describe the purpose of each workspace. (OPSC-9760)
• Updated LCM definitions to add DSE 5.0.3 support. (OPSC-10448)
• LCM now warns the user when adding a seed node to an existing datacenter. Such nodes will fail to bootstrap and will require a manual repair immediately after installation is complete. (OPSC-10478)
• LCM now properly formats job event stacktraces with line breaks. (OPSC-9190)
• LCM prompt widgets now properly escape html tags on cluster, datacenter, and node model names. (OPSC-9279)
• Added missing configuration variables to LCM definitions for dse-spark-env.sh. (OPSC-10605)
• LCM now uses broadcast_rpc_address (instead of rpc_address) to change the Cassandra password. This fixes the password change feature for public/private IP cloud installations. (OPSC-10801)
• The truststore and truststore_password fields for client_encryption_options are now included in the Config Profile in LCM regardless of the value of require_client_auth. Omitting the truststore_password field would cause DSE to crash on startup when config_encryption was enabled. (OPSC-10807)
• Clicks to the right of a checkbox will no longer toggle the checkbox in Config Profiles. (OPSC-10830)
• Fixed an issue where LCM failed to install DSE 4.x packages on apt-based systems. (OPSC-10330)
• Fixed an issue where some attributes displayed out-of-date values in LCM cluster, datacenter, and node details pages. (OPSC-10439)
• Fixed an issue where DSE would fail to start, but LCM would report the job as successful. (OPSC-10520)
• Fixed bug in how config profile lists are rendered. (OPSC-10533)

Monitoring and Best Practice Service
• OpsCenter now supports using an http/https proxy in its posturl plugins (page 125). (OPSC-10644)
• Added hover text to subtitles in zoomed graph dialog. (OPSC-3391)
• Dashboard widget text is now selectable. (OPSC-4585)
• Changed the way we calculate heap size for the Solr heap best practice rule to use GiB to be consistent with how Java reports its heap information. (OPSC-10726)
• Best Practices list now maintains scroll position during a refresh. (OPSC-7433)
• Updated the Best Practice Service rule for Solr heap size to be aware of garbage collector type when determining recommended heap size. (OPSC-10300)
• Updated the Best Practice Service rule for vnodes on search nodes to recommend the correct number of vnodes based on the version of DSE running on the search node. (OPSC-10365)
• Fix an issue when a histogram overflows causes serialization to fail. (OPSC-10335)

Backup Service
• The API now checks the keyspace and destination parameters when creating or updating backup jobs. Any specified keyspaces and destinations must exist before the job change. (OPSC-9639)
• Improved error messages when a restore fails because nodes are down. (OPSC-10204)
• Improved the error message when an Amazon S3 bucket name is too short. (OPSC-10760)
• All Keyspace backup jobs that include empty keyspaces will now show the correct status instead of always reporting an error. (OPSC-8894)
• Removing the throttle option from cluster config file now results in using the unthrottled_default value instead of throwing an exception. (OPSC-10576)
• Added support for unicode character in solr config files. (OPSC-10632)
• Improved the View Details dialog for backup and restore errors. (OPSC-10727)
• Fixed a bug that prevented pit restore precheck errors from appearing in the backup history. (OPSC-6638)
• Fixed a bug where the status dialog for a point in time restore would not display information while running a restore containing multiple keyspaces. (OPSC-7325)
• Fixed an issue where users may observe a large number of log messages about requests to /pit-cleanup if there are a large number of existing commitlogs in the staging directory. (OPSC-8349)
• Fixed an issue where the DataStax Agent could run out of memory during transfers to Amazon S3 when using compression. (OPSC-10235)
• Fixed an issue that would sometimes cause enabling PIT restore to fail. (OPSC-10268)

Repair Service
• Improved handling around error conditions for JMX notifications. (OPSC-10390)
• If repairing a range is postponed because of a downed node, the downed node is now included in the log message. (OPSC-10513)

OpsCenter 6.0.5 Release Notes

Compatibility

To see which versions of DataStax Enterprise are supported with OpsCenter 6.0, see the OpsCenter Compatibility chart (page ).

For upgrade instructions, see the DataStax OpsCenter Upgrade Guide (page ).
Known Issues

**Important:** Please be sure to review the list of Known Issues in OpsCenter 6.0 (page 45) before running on a production cluster.

Provisioning

- Fixed LCM repository authentication bug when special characters exist in credentials (as with most DataStax Academy usernames). (OPSC-10817)

OpsCenter 6.0.4 Release Notes

Compatibility

To see which versions of DataStax Enterprise are supported with OpsCenter 6.0, see the OpsCenter Compatibility chart (page ).

For upgrade instructions, see the DataStax OpsCenter Upgrade Guide (page ).

Known Issues

**Important:** Please be sure to review the list of Known Issues in OpsCenter 6.0 (page 45) before running on a production cluster.

Core

- OpsCenter LDAP now filters multiple user LDAP groups by known OPSC roles if necessary during authentication. (OPSC-6946)
- Improved a misleading error message for OpsCenter schema creation failures. (OPSC-9208)
- Upgraded OpsCenter to use the latest version of the Cassandra Java Driver (3.1.0). (OPSC-9232)
- When HTTPS is enabled, OpsCenter now issues cookies with the Secure flag. (OPSC-8713)
- The Add Cluster dialog now requires cassandra keystore and agent keystore fields when client-to-node encryption is enabled. (OPSC-10150)
- Appropriately cleaned up an issue with the tiered storage metric collector that caused OOM problems. (OPSC-10491)
- Fixed a bug where a user's role was incorrectly updated during a password change by an admin user. (OPSC-8860)
- Fixed an issue where OpsCenter could timeout while loading the events page. (OPSC-9762)
- Fixed an issue where if client-to-node encryption was enabled for an OpsCenter monitored cluster and there was also a separate storage cluster, the agents were unable to connect to the storage cluster. (OPSC-10114)
- Fixed an issue where the DataStax Agent would incorrectly bind its http server to broadcast_rpc_address for Cassandra when rpc_address was set to 0.0.0.0. Now the
DataStax agent binds to whatever value `rpc_address` is set to unless it is overridden using the config parameter `agent_rpc_interface` in `address.yaml`. (OPSC-10243)

- Fixed a bug where an improperly specified `user_search_filter` for LDAP would cause a stack trace to be logged that hid the proper LDAP error. (OPSC-10303)
- Fixed a race condition when both an agent and its DSE node were down. The issue caused incorrectly reporting the node as up when the agent status changed. (OPSC-10336)
- Fixed an issue that now allows sensitive config values to be specified in and properly decrypted using the agent `address.yaml` file. (OPSC-10396)
- Fixed a bug in OpsCenter and the Datastax Agent where both systems could not authenticate with internal auth if it was specified as the secondary authentication method in DataStax Unified Authentication. (OPSC-10456)
- Fixed an issue where the Spark Web UI proxy would be unusable when authentication was enabled. (OPSC-6606)
- Fixed an issue that prevented proper startup of the HTTP reverse proxy to the Spark master. This issue only affected Analytics clusters running Spark. (OPSC-10221)

Monitoring

- Added the tombstones per read metric. (OPSC-2069)
- The `not_seen_threshold` config value now also resets node health to unknown when it times out, in addition to marking agent connections as down. (OPSC-9844)
- Improve memory use when many files are being streamed to or from Cassandra. (OPSC-10565)
- Updated the ALLOW FILTERING description for the Best Practice Service rule as appropriate for certain analytic workloads. (OPSC-10297)
- OpsCenter now supports named parameters for the Best Practices Service prepared statements rule check. (OPSC-7323)
- Lowered the default for the `slow_query_fetch_size` configuration option available in `address.yaml` to alleviate potential stack overflows. (OPSC-7471)
- Fixed a bug in the old metrics api when returning device-specific metrics. (OPSC-8885)
- Fix 400 error with cluster-metrics API calls. (OPSC-9392)
- Fixed a bug for Read and Write Latency graphs that caused no data to be displayed when multiple nodes were selected. (OPSC-9568)
- Fixed a bug where the default table metric was not selected when View Metrics was clicked in the Data workspace. Added validation to the Add Metric dialog that warns the user a table selection is ignored if a non-table metric is selected. (OPSC-9766)
- Fixed a bug where users were repeatedly informed about unavailable metrics in dashboard presets. (OPSC-10040)
- Fixed an issue when restarting agents that caused the agent to lose track of the list of configured alerts. (OPSC-10579)
- Fixed a rendering issue with the vnode ring view. (OPSC-9212)
- Fixed an issue where schedules for the Best Practice service were re-enabled after an OpsCenter restart if all of the rules were disabled in the previous session. (OPSC-10157)
• Fixed Solr filterCache best practice rule to account for DSE SolrFilterCache class. The best practice for the SolrFilterCache is to have a highWaterMarkMB setting of 256 (MB). (OPSC-6994)

Backup Service
• Improved the error message and its presentation for certain backup failures cases. (OPSC-9989)
• Improved feedback when OpsCenter does not have write permissions to an S3 bucket. OpsCenter now presents an error if the agents fail to upload a backup to S3 due to permission errors. (OPSC-7029)
• Clicking save on the scheduled backup job dialog now waits until the job has been successfully saved before closing the dialog. (OPSC-9607)
• Cleaned up dirwatch threads when reconnecting to the Opscenter daemon. (OPSC-10528)
• Fixed a bug that caused excessive logging about unexpected files encountered during the backup staging process. (OPSC-4415)
• Fixed an issue with the Backup Report and status if an S3 destination of a scheduled backup is deleted before the backup has fully completed. (OPSC-6712)
• Fixed an issue preventing point in time restore if the commit logs were archived to multiple destinations and at least one destination enabled compression. (OPSC-10293)
• Fixed an issue causing errors when backing up DSE 5.0+ nodes that were still using an older (pre-DSE 5.0) data directory structure/format. (OPSC-10309)
• Fixed a regression where backups did not appear in the restore dialog. (OPSC-10668)

Lifecycle Manager (LCM)
• Invalid package repository credentials in LCM now have a user-friendly error message. (OPSC-6554)
• The meld remote executor used during LCM jobs has changed format from a .py text file to a .pyz zip application. This has no customer-facing impact except in cases where sudoers has required customization to allow meld to execute. (OPSC-7612)
• Improved the error message when the DSE package cannot be found in the package repository during an LCM job. (OPSC-8391)
• Editing an LCM config profile and navigating away without saving now warns users of unsaved changes. (OPSC-8424)
• Password fields under LCM config profiles are now password input types. (OPSC-10379)
• Reduced the likelihood of apt and yum corruption when terminating LCM jobs. (OPSC-9453)
• LCM database operations resulting in a "database is locked" error will be retried several times before failing. (OPSC-9567)
• LCM does not display clusters for importing if the clusters are not available for monitoring. (OPSC-9625)
• Added SSH configuration options (page 420) to alleviate connection issues when running an LCM job. (OPSC-9652)
• Enabled LCM to downgrade agent packages, which prevents job errors when installing new versions of DSE using an older version of OpsCenter. (OPSC-9749)
• Improved the ability of LCM to scale to larger job sizes. (OPSC-10493)
• Fixed a positioning issue with action menus on the Clusters workspace of LCM. (OPSC-8988)
• Fixed an issue that caused LCM clusters (in the Clusters workspace pane) to report that their most recently executed job was successful even though it had failed. (OPSC-9254)
• Fixed an issue with the auto_bootstrap setting when importing clusters into LCM. The issue caused restarting Cassandra to fail after running a configure job. (OPSC-10151)
• Fix rendering bug affecting some config profile values. (OPSC-10215)
• Fixed a LCM job error when the meld remote executor takes too long to start up. (OPSC-10461)

Repair Service
• The Repair Service now continues to run when nodes go offline (page 325). (OPSC-6452)
• If the Repair Service is unable to complete the repair operation within the time to completion window, the Services page now presents a link to the documentation on tuning the Repair Service in the informational status tooltip. (OPSC-6458)

OpsCenter 6.0.3 Release Notes

Compatibility
To see which versions of DataStax Enterprise are supported with OpsCenter 6.0, see the OpsCenter Compatibility chart (page 280).
For upgrade instructions, see the DataStax OpsCenter Upgrade Guide (page 477).

Known Issues

Important: Please be sure to review the list of Known Issues in OpsCenter 6.0 (page 45) before running on a production cluster.

Core
• The [cassandra].log_location property in cluster_name.conf (cassandra_log_location in address.yaml) now expects a directory path (for example, /var/log/cassandra/) instead of the path directly to system.log (for example, /var/log/cassandra/system.log). This change is backwards compatible with existing settings that point directly to system.log. (OPSC-9399)
  Note: This only affects clusters that have a custom path for DSE logs.
• Cluster configuration filenames are now validated and will not be loaded if they contain characters other than a-z, 0-9, or an underscore. (OPSC-10192)
• The cassandra cluster (cluster_name.conf) configuration property ssl_truststore_password is now included in the list of configuration property values that are encrypted when config_encryption_active is set to true. (OPSC-9747)
• The storage_cassandra cluster properties ssl_keystore_password and ssl_truststore_password are now included in the list of configuration property values that are encrypted when config_encryption_active is set to true. (OPSC-10219)
• Improved some DataStax Agent log messages in states when the DSE node is down or unreachable. (OPSC-9917, OPSC-9940, OPSC-10027, OPSC-10028)
• Fixed the Show and Hide Cluster feature in the navigation menu. (OPSC-9478)
• Fixed an issue displaying messages in the security log that contain certain Unicode characters. (OPSC-8846)
• Fix a logging bug that prevented STOMP error details from appearing in logs. (OPSC-9242)
• Fixed an issue where a warning banner for a cluster persisted after removing that cluster from OpsCenter. (OPSC-9272)
• Fixed an issue where the List and Agent tabs in the Nodes section were not clickable. (OPSC-9626)
• Fixed an issue where OpsCenter would not start up if HTTPS was enabled and the certificate files contain intermediate certificates. (OPSC-9999)
• Fixed an issue with an incorrect cluster name being displayed in the agent warning banner. (OPSC-10167)
• Fixed issue in error handling that prevented the OpsCenter web server from cleanly shutting down and instead resulted in a System is not defined error message. (OPSC-10188)
• Fixed a bug in sensitive config encryption, introduced in v. 6.0.0, which yielded encrypted values unable to be decrypted by the OpsCenter Agent. For CFB-mode of encryption, this is due to the agent expecting a CFB-8 byte block cipher of encryption when in fact a CFB full block cipher was being used. For the other three supported cipher modes, the bug is due to incompatible plaintext padding. (OPSC-10244)
• Fixed an issue with initialization vector randomization when encrypting configuration fields. (OPSC-10311)

Monitoring
• A new status has been added in the Ring view for nodes with an unknown health status due to agent connection issues. Nodes with unknown health information are now shown with a gray color and dashed outline. (OPSC-8955)
• An uncaught exception is now prevented if either or both of the optional parameters (start and end) are omitted when invoking a metric API endpoint. (OPSC-9727)
• Fixed a memory leak in the agent caused when unable to load previous state because of CASSANDRA-12053. (OPSC-9712)
• Fixed an error that prevented viewing dashboard graphs when the graph scale is set to monthly. (OPSC-9995)
• Fixed an issue with the Clocks in UTC Best Practice Rule that caused a false failure when the timezone is set to Etc/UTC. (OPSC-10224)

Backup Service
• Improved the layout of the Restore Schema Validation comparison dialog for restoring a backup. (OPSC-7248)
• Improved error messages for failed backups due to unavailable nodes and/or agents. (OPSC-9573)
• Fixed an issue that prevented backup and restore of keyspaces or tables with long names (between 32 and the maximum of 48 characters). (OPSC-9563)
• Fixed an issue with a confusing Edit Backup Schedule workflow that led to the following error: Destination missing: OPSC_ON_SERVER. (OPSC-5257)
• Fixed a bug where formatting characters sometimes appeared in restore logging statements. (OPSC-8749)
• Fixed an issue that prevented the display of all available backups for some clusters when using the Other Locations tab to restore. (OPSC-9117)
• Fixed an issue where some failure conditions in backups would result in no backup activity entries being created. (OPSC-9151)
• Fixed a race condition that sometimes caused the first backup to fail after a restart of an agent. (OPSC-9598)
• Fixed an intermittent issue preventing the restore of tables with solr cores. (OPSC-9601)
• Fixed an issue where backups older than one week would not be displayed in the Activity tab of the Backup Service. (OPSC-9860)
• Fixed an error when an agent is not connected while attempting a restore of a backup made to a Local FS destination. (OPSC-10111)
• Fixed an issue causing errors when backing up DSE 5.0+ nodes that were still using an older (pre-DSE 5.0) data directory structure/format. (OPSC-10309)

Lifecycle Manager
• OpsCenter Lifecycle Manager now provides a return link to OpsCenter Monitoring on its left navigation panel. (OPSC-10011)
• LCM now prompts for an optional sudo username when creating ssh credentials. Jobs running with these credentials will run with sudo using the provided user. (OPSC-9231)
• Improved automatic updates of data in the UI to account for changes in other tabs. (OPSC-9306)
• Basic auth passwords are now masked in log messages when setting repositories in LCM. (OPSC-9532)
• LCM API now supports filtering on fields of all types (int, boolean, and so forth). (OPSC-9619)
• Fixed layout issues in the LCM UI when viewing within narrow browser windows. (OPSC-8998)
• Fixed an issue causing errors when clearing optional fields in various UI forms. (OPSC-9301)
• Fixed an issue causing a single cluster to appear duplicated in the Clusters list after import. (OPSC-9383)
• Fixed an issue causing the memtable_cleanup_threshold property in cassandra.yaml to not accept float values. (OPSC-9565)
• Fixed an issue preventing Cluster Import from working when OpsCenter Authentication is enabled. (OPSC-9812)
• Fixed an issue preventing DSEFS from working when using a default Config Profile. The default values for the data_directories option of DSEFS are now present in the dse.yaml generated by LCM. (OPSC-10162)
• Fixed an issue preventing proper agent configuration when installing a node using LCM with SSL communication enabled between opscenredd and the agent ([agents].use_ssl). (OPSC-10200)
• Fixed a bug that prevented Lifecycle Manager installations that used a package proxy on RPM-based systems. (OPSC-10380)

Repair Service
• Clarified the error message when a repair will not complete within the specified time window and the revised estimate for completion is specified. (OPSC-6817)
• The Repair Service no longer consider repairs as failed when JMX notifications are lost. (OPSC-10112)
• Fixed a race condition that prevented the Repair Service from restarting after a topology change. (OPSC-9244)

OpsCenter 6.0.2 Release Notes

Compatibility
To see which versions of DataStax Enterprise are supported with OpsCenter 6.0, see the OpsCenter Compatibility chart (page ).
For upgrade instructions, see the DataStax OpsCenter Upgrade Guide (page ).

Known Issues

Important: Please be sure to review the list of Known Issues in OpsCenter 6.0 (page 45) before running on a production cluster.

Core
• Added ability to resize the Opcenter cluster panel on the left navigation panel to fully view long cluster names. (OPSC-9380)
• New best practice rule checks that all nodes are in the UTC timezone. (OPSC-4955)
• New best practice rule to checks if DSE and agents run as the same user. (OPSC-8086)
• More logging statements that indicate the start and end of the data-collector, and log the process of a file being added to the diagnostics tarball. (OPSC-8964)
• Deleted the unused configuration item protocol_version from opscenredd configuration files. (OPSC-9080)
• Fixed an issue where the agents sometimes had difficulty connecting to DSE in multi-datacenter clusters when connecting with consistency level LOCAL.ONE (CASSANDRA-12053). (OPSC-9659)
• Fixed an issue where email alerts were not sent when OpsCenter was configured to send alerts in TLS-only mode. (OPSC-9451)
• Fixed an issue where OpsCenter failed to remove a cluster from monitoring if it could not first establish a connection. (OPSC-9580)
• Fixed an issue where popup blockers could prevent a user from downloading the diagnostics tarball. (OPSC-8914)
• Fix dsetool arguments in diagnostics generation when jmx authentication is enabled. (OPSC-6080)
• Fixed an issue that prevented DSE defaults from being included in the diagnostic tarball for DSE 5.0 package installs. (OPSC-9339)
• Fix issues when setting rollup_rate_unit agent configuration option in address.yaml. (OPSC-9394)
• Fixed an issue where changes to a backup schedule might not appear in the Backup dialog if the schedule changes have not yet been saved. (OPSC-7527)
• Fixed an issue with sorting the display in tables on the Node Details dialog. (OPSC-9041)

Lifecycle Manager
• Improved resiliency of Lifecycle Manager in situations where there is high latency between the OpsCenter daemon and nodes in the cluster. This release improves upon the problem; however, there are still known issues in high latency scenarios that will be addressed in a future release. (OPSC-8851)
• Added seconds to the time displayed on the Job Events list. (OPSC-8971)
• Eliminated unnecessary reliance on the shell when invoking subprocesses on LCM target nodes; audited inputs in remaining cases. (OPSC-8372)
• Improved responsiveness in LCM form field inputs. (OPSC-9291)
• Attempting an agent install without a password or private key now has a visible and useful error. (OPSC-9413)
• LCM now reloads definitions automatically after an update. No restart is required. (OPSC-9468)
• Fixed issue with some LCM screens not working when HTTPS was enabled. (OPSC-9757)
• Fixed issue resulting in superfluous "meld error" events. (OPSC-9473)
• Fixed a race condition between successive jobs in Lifecycle Manager when the first job fails or is terminated. (OPSC-9061)
• Fixed rendering bug in Safari 8. (OPSC-9123)
• Fixed a harmless javascript stack trace when opening the Add Node dialog. (OPSC-9280)
• Fixed issue with some Config Profile properties, including heap size, being disabled and not editable. (OPSC-9546)
• Fixed a bug that caused some config profile values, including g1-gc-opts, not to be committed when saved. (OPSC-9556)

OpsCenter 6.0.1 Release Notes

Compatibility

To see which versions of DataStax Enterprise are supported with OpsCenter 6.0, see the OpsCenter Compatibility chart (page ).

For upgrade instructions, see the DataStax OpsCenter Upgrade Guide (page ).

Known Issues

**Important:** Please be sure to review the list of Known Issues in OpsCenter 6.0 (page 45) before running on a production cluster.

Core

• The diagnostics tarball now includes the debug.log and gremlin.log files. (OPSC-7133)

• Upgraded to the latest version of the Cassandra Java Driver (3.0.3), which fixed an issue where schema agreement was never reached if null entries were present in the system.peers or system.local tables (JAVA-1202). This issue prevented some clusters from being added to OpsCenter. (OPSC-9397)

• Fixed an issue where authenticated users would receive a non-descriptive error when logging in a second time. (OPSC-4796)

• Fixed an issue where the jython script would sometimes fail to run on Ubuntu. (OPSC-9036)

• Fixed an issue preventing usernames for authenticated users from appearing in the OpsCenter events table for corresponding actions that took place during a session. (OPSC-9114)

• Fixed an issue where opscenterd and the agent sometimes failed to resolve symlinks when searching for the Java install location. (OPSC-9344)

• Fixed an issue when OpsCenter authentication and https were both enabled, which caused the user to be redirected to an invalid login url (random port). (OPSC-9379)

Backup Service

• Fixed an issue that would fail the backup if the cluster had a temporary schema disagreement at the same time the backup started. The backup process now pauses for a short period while waiting for the schema disagreement to resolve. If the schema disagreement does not resolve, the backup fails with an explicit message to that effect. (OPSC-8952)

• Fixed an issue with re-creating tables during restore when those tables include user-defined types (UDT), user-defined functions (UDF), and user-defined aggregates (UDA). (OPSC-9261)

• Fixed an issue where a backup to a destination would fail if the keyspace contained a materialized view. (OPSC-9328)
• Fixed an issue when using the Local File System (LFS) destination that prevented the
restore action from restoring all sstables associated with a backup. Specifically, it would
only restore the files discoverable on a single node, so that if the folder specified was
local only (as opposed to NFS) or otherwise could only discover a subset of the files in
the backup, the restore operation would report success but only restore a portion of the
required files. This issue did not affect the snapshot process and does not represent a
corruption of the backed-up data. (OPSC-9543)
• Fixed an issue that caused backups to fail at the OpsCenter daemon for search-enabled
nodes in some situations. (OPSC-9595)

Lifecycle Manager
• Fixed an issue preventing Lifecycle Manager from properly installing DSE versions 4.7
and 4.8 in Debian/Ubuntu systems. (OPSC-9409)
• Fixed an issue causing address.yaml to be owned by the root user when the agent
is installed for the first time via OpsCenter. The address.yaml file is now included in the
datastax-agent packages (.rpm/.deb) by default and is owned by the cassandra user.
(OPSC-9336)

OpsCenter 6.0.0 Release Notes

Highlights
• The opscenterd process now runs on the JVM.
• Vastly improved the visibility, display, and behavior of agent status and installation.
• Alerts improvements, including SNMP (page 119) alerts, HipChat integration and more
flexibility.
• Backup and Restore to and from Local Directories (page 293).
• The new Lifecycle Manager (page 387) is now available, allowing users to create
new clusters with the click of a button, expand existing clusters, and centrally manage
configuration for all of their nodes.
• OpsCenter 6.0 now only supports DataStax Enterprise clusters. Attempts to add a non-
DSE cluster error gracefully. Any currently configured non-DSE clusters do not prevent
opscenterd from starting up; however, OpsCenter 6.0 does not monitor any non-DSE
clusters.

Be sure to check out the New features (page 10) section for more details.

Compatibility
To see which versions of DataStax Enterprise are supported with OpsCenter 6.0, see the
OpsCenter Compatibility chart (page ).

For upgrade instructions, see the DataStax OpsCenter Upgrade Guide (page ).

Known Issues

Important: Please be sure to review the list of Known Issues in OpsCenter 6.0 (page
45) before running on a production cluster.
Core

- Added full support for the newly released DataStax Enterprise 5.0. (OPSC-5562)
- OpsCenter now runs on the JVM. See Configuring the OpsCenter JVM (page 150). (OPSC-4915)
- Python has been removed as a requirement for opsc entered to run. (OPSC-7368)
- If you are planning on or are currently using the opsc_system_key_tool to add encryption to your opsc entered configuration file, note that your key size is limited to 128 bits unless the Java JCE Policy files Extension is installed for your JDK. If you are already using the encryption feature and your system key is currently greater than 128 bits, you must ensure you have this extension installed for the encryption feature to continue functioning properly. (OPSC-5985)
- There is now a startup.log (page 70) file that gets created upon startup of the opsc entered process. This log file can contain debug information or stack traces if opsc entered fails to start up before a normal opscentered.log is created. (OPSC-7565)
- Improved the encryption strength of passwords (page 101) stored in passwd.db when using OpsCenter authentication. Passwords are re-encrypted when users log in again. (OPSC-4400)
- More detailed agent status can now be found in the Agents (page 201) tab in the Nodes section. (OPSC-7364)
- The Agents dialog has been replaced by a less obtrusive banner at the top of the UI. (OPSC-7385)
- The Add Cluster workflow now incorporates the option to install agents automatically or manually. (OPSC-7388, OPSC-7815)
- The automated agent install process using SSH has been overhauled to be idempotent and more robust. (OPSC-7399)
- Improved automatic detection of local node properties in the agent; including but not limited to file location, permission detection, and handling. (OPSC-2445)
- The datastax_agent_monitor process has been removed. The process existed to automatically restart the DataStax agent if it crashed. Users should now use the third-party tool of their choice to accomplish automatically restarting the DataStax agent if so desired. (OPSC-6901)
- Added stronger validation of configuration options in address.yaml. If invalid properties are found, the agent shuts down. (OPSC-6945)
- Reduced the verbosity of log messages that stated a message was received from an agent opsc entered could not recognize. These messages are now batched and logged only every 10 minutes by default. (OPSC-7006)
- Improved error handling when LDAP connection issues are encountered during login. (OPSC-5839)
- The API timeout for managing a new but existing cluster has been increased from 30 seconds to 5 minutes. The extra time allows clusters with schemas that are still settling to be added successfully. (OPSC-9203)
- Trace logging was added for all opsc enteredd incoming stomp messages. (OPSC-8664)
- Sensitive information is now *REDACTED* from logs rather than excluded completely. (OPSC-6572)
• The Cluster Report now opens in a new tab rather than automatically downloading a PDF. Users can save and export the page in whatever format is supported by their OS and browsers. (OPSC-7852)
• Added a timeout to schema agreement to improve the robustness of creating a schema upon startup. (OPSC-8253)
• Automatic definition file updates now leverage the list of SSL/TLS certificate authorities built-in to the JVM rather than a bundled certificate. (OPSC-6782)
• The standalone installer (.run for linux, .dmg for OS X) for OpsCenter has been discontinued. (OPSC-9044)
• Native browser auto-complete has been disabled for cluster username and password fields. (OPSC-7619)
• Fixed an issue where opscenterd was running as root when installed using an RPM package. The opscenter user is now created upon install. Note that any OpsCenter files on custom paths might need updated permissions. (OPSC-5487)
• Fixed an issue with the Manage Roles button being disabled when one or more DataStax Enterprise clusters are down. (OPSC-4396)
• Fixed an issue with the Multiple Versions Detected dialog displaying on the Dashboard when restarting nodes. (OPSC-1407)
• Fixed an issue with writing multiple `stomp_interface` properties to `address.yaml` during failover. (OPSC-5048)
• Fixed an issue that prevented scrolling the cluster navigation list when necessary. (OPSC-5465)
• Fixed an issue that required restarting opscenterd after JMX authentication was enabled on a cluster. (OPSC-5524)
• Fixed an issue causing browsers to incorrectly autofill some credential fields in the Edit Cluster Connection Settings dialog. (OPSC-5886)
• Fixed an issue where updating Edit Cluster Connection Settings through the UI would remove other properties set manually in `cluster_name.conf`. (OPSC-6078)
• Fixed an issue causing the `startup_sleep` property to not be respected. This property controls an optional delay between clusters when opscenterd is starting up to alleviate performance issues. (OPSC-7334)
• Fixed an issue in tarball installations that excluded `opscenter_system_key_tool`. (OPSC-7347)
• Fixed an issue with the Role list not selecting the correct role when editing a user. (OPSC-7861)
• Fixed an issue where opscenterd would not use the broadcast address to connect to DataStax agents when it was set. (OPSC-7897)
• Fixed an issue that prevented trying to create the OpsCenter keyspace again if it failed. (OPSC-8336)
• Fixed an issue editing keyspace replication when datacenter names contained hyphens. (OPSC-6137)
• Fixed an issue with the repair service causing errors via overlapping repairs. This fixes the "Cannot start multiple repair sessions over the same sstables" error. (OPSC-8202)
Fixed an issue where Repair Service would block on replication settings of `system_distributed` (it was not ignored as with all other system keyspaces).
(OPSC-7993)

**Monitoring**

- When upgrading DSE, alerts that exist for metrics that no longer exist in DSE will be automatically deleted. Users will be notified of this and the automatic removal of deprecated graphs on Dashboard presets when the OpsCenter UI is loaded.
  (OPSC-7763)
- Non-percentile latency (that is, Read/Write Request Latency) metrics were removed from DSE and also OpsCenter. A placeholder metric generated from the percentile latency metrics will take its place but are calculated a little differently. The average metric is actually the median now, and the minimum and maximum are the actual minimum and maximum occurring latencies instead of the minimum and maximum collected averages.
  (OPSC-8458)
- The Nodes section is now shown by default rather than the Dashboard when loading the OpsCenter monitoring UI. (OPSC-8234)
- Added support for DSE Tiered storage including metrics for storage tiers (page 247).
  (OPSC-7458)
- Added SNMP (page 119) integration in alert notifications. (OPSC-309)
- Added a new Agent Issue (page 203) alert type. Users can now be proactively alerted when an agent installation or configuration may need attention. (OPSC-1862)
- Added HipChat (page 129) integration to alert notifications. (OPSC-2750)
- Added ability to export metrics directly to Graphite (page 149).
  (OPSC-4499)
- Global dashboard now displays a warning message when Opscenter cannot connect to a cluster. (OPSC-6966)
- Threshold information is now included in the body of email alerts. For example: (Current value is 30; threshold is >10).
  (OPSC-3827)
- Cluster name has been added as a property in POST URL alert notifications.
  (OPSC-4786)
- Multiple recipients can now be specified in email alerts. (OPSC-5193)
- Preset labels on the dashboard range selector are now marked bold upon selection.
  (OPSC-5760)
- Added ability to set up alerts on percentile metrics, such as read and write latency.
  (OPSC-6791)
- Creating and updating alerts with a duration set to zero now returns an HTTP 400 response. (OPSC-7474)
- Added validation to start and end timestamps on /new-metrics API call. (OPSC-7666)
- Values shown on Thread Pool Stats are now color-coded to bring attention to important values.
  (OPSC-4641)
- Fixed an issue causing the Activity listing to reset scroll position upon update in the Activities tab. (OPSC-7519)
- Fixed issues with and updated schema for maintaining rollup states across agent restarts. (OPSC-4190)
• Fixed an issue causing zoomed Dashboard graphs to not reflect the proper graph parameters in some cases. (OPSC-7478)
• Fixed an issue that caused the Down Node alert to not disable properly. (OPSC-7766)
• Fixed an issue that caused graphs using All Nodes to improperly load. (OPSC-8035)
• Fixed an issue that caused the Storage Capacity dashboard widget to appear blank. The issue was due to OpsCenter failing to parse mounted filesystems in some environments. (OPSC-8215)
• Fixed an issue with mini-graph label layouts on the Overview page in Firefox. (OPSC-6024)
• Fixed UI layout overlap issues when nodes had many tasks running at the same time. (OPSC-5374)
• Fixed an issue where filters on the Nodes List View would not be cleared until users clicked away from the list in the UI. (OPSC-6399)
• Fixed an issue with node load decimal precision being too long. (OPSC-7969)
• Fixed an issue that prevented Search metrics from being available when running Search Analytics workloads. (OPSC-5002)

Backup Service
• Added the ability to back up to and restore from a user-defined directory on a local filesystem. (OPSC-5185)
• Added the ability to back up multiple keyspaces in a single backup. (OPSC-7712)
• Browsing S3 backups to restore now leverages the `default_api_timeout` setting for environments where the API call takes longer than 60 seconds. (OPSC-8863)
• Added new `sstableloader_max_heap_size` property to agent configuration to increase the `MAX_HEAP_SIZE` of `sstableloader` during restore. (OPSC-7225)
• Improved connection retry handling when checking if a blob exists in S3. (OPSC-7146)
• Improved error reporting during restore when `sstableloader` runs out of memory. (OPSC-7180)
• Exposed new configuration options to increase the history of job statuses stored on the agent. More information about the symptoms and solution can be found here: https://support.datastax.com/hc/en-us/articles/206456076. (OPSC-6917)
• Data size for backups now displays in all cases. (OPSC-7686)
• Scheduled backups can no longer be created in the past, which prevents accidental unexpected behavior. (OPSC-7421)
• Provided clear messaging when attempting to back up very large keyspaces that might exceed currently configured limits. (OPSC-7537)
• Attempting to restore from a deleted backup is no longer allowed. (OPSC-7647)
• Improved validation handling for S3 bucket names that are not lowercase. (OPSC-8015)
• A manual backup deletion has been re-labelled "Deletion Complete" in the Backup Service activity table. The backup report for a deleted backup now contains the heading "Pre-Deletion Summary" and label "This backup has been deleted". (OPSC-8656)
• Fixed issues when running many backup and restore operations back-to-back. (OPSC-7125)
• Fixed an issue with lost+found directories causing backups to fail. Only directories for existing tables are scanned. (OPSC-5389)
• Fixed an issue that caused Data Size to always appear as N/A on the Restore Status dialog. (OPSC-4498)
• Fixed an issue when restoring to a cluster with client-to-node encryption enabled and a custom keystore path. (OPSC-6692)
• Fixed error handling when restoring a nonexistent table. (OPSC-7094)
• Fixed an issue with post-backup scripts causing files to be sent as a JSON blob rather than a file-per-newline. (OPSC-7108)
• Fixed an issue with some multi-part S3 backups failing when using S3 server-side encryption. (OPSC-7247)
• Fixed consistency issues with PIT restores. (OPSC-7639)
• Fixed an issue that created some duplicate directories during backups. (OPSC-7655)
• Fixed an issue when backing up DataStax Enterprise encrypted tables multiple times. (OPSC-7709)
• Fixed an issue in cleaning up destination after a Restore. (OPSC-7767)
• Fixed an issue with the display of Restore dialog pushing buttons beyond view. (OPSC-7858)
• Fixed the display of the Restore Report dialog after an S3 bucket has been removed. (OPSC-8212)
• Fixed an issue that caused the agent config property backup_file_queue_max to not be respected. (OPSC-8868)
• Fixed issues with the Backup Service and Best Practice Service that corrupted schedules and expected jobs to fire in the past, which caused those services to run more aggressively than they should. (OPSC-7350)

Diagnostic Tarball
• The cluster_name.conf file is now included in the diagnostic tarball. (OPSC-7157)
• The agent address.yaml configuration is now included in the diagnostic tarball. (OPSC-7177)
• Changed the way a new browser tab is opened for the Diagnostics Tarball as a workaround to popup blockers in some browsers. (OPSC-8869)
• Added a diagnostic_tarball_download_timeout config property that allows users to increase the timeout for downloading information from a single node. (OPSC-8891)
• Fixed an issue with the diagnostic tarball not collecting cqlsh output for clusters running DSE 4.7 or later. (OPSC-7053)

Performance Service
• Those that use Lifecycle Manager and want to persist Performance Object settings in DSE are notified to make those changes manually in Lifecycle Manager rather than through the Performance Service. The Performance Service still applies any changes via JMX, but the changes do not persist after a DSE restart. (OPSC-8355)
• Fixed some issues with link rendering. (OPSC-5787, OPSC-7404)
• Fixed an issue that truncated the titles of some graphs in the Performance Service. (OPSC-5788)
• Fixed an issue that caused stack overflows in some Performance Service edge cases. Most users would not see the symptoms of this issue. (OPSC-7648)

Best Practice Service
• Fixed an issue with properly showing disabled Best Practice Service rules. (OPSC-5779)
• Fixed an issue with the "Security superuser has default setting" Best Practice Service rule that prevented the rule from warning properly. (OPSC-7281)
• Fixed issues with the Backup Service and Best Practice Service that corrupted schedules and expected jobs to fire in the past, which caused those services to run more aggressively than they should. (OPSC-7350)

Known Issues
The following are new issues that exist in OpsCenter 6.0.x 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, please contact DataStax Support for assistance.

• Users may observe a large number of log messages about requests to /pit-cleanup if there are a large number of existing commitlogs 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 opscentered.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 opscentered to hang on startup in some slower environments. (OPSC-9314; see More Details)
• Failure to follow the required prerequisite instructions to install Oracle Java SE Runtime Environment 8 (JRE or JDK) before installing OpsCenter 6.0 on Ubuntu 16.04 results in installation of OpenJDK 9, which is not currently supported. (OPSC-10778)
• Kerberos authentication will not work when the rpc_address setting in cassandra.yaml is 0.0.0.0. The symptom of this issue is reflected in the Agent Status view, which shows the storage database as up but the monitored database as down according to the agent. (OPSC-11217)
• 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)
• (Applicable to OpsCenter version 6.0.10) When running DSE nodes that use two different network interfaces to separate client traffic from internode traffic, the OpsCenter agent will fail to establish a STOMP connection. For more details, please see the KB support article. (OPSC-13016)
• Lifecycle Manager (LCM) (page 387)
  # Under certain circumstances, OpsCenter Lifecycle Manager may fail to install java unless the OpsCenter version being used is at least 6.0.11 in the 6.0.x series. For more details, please see the KB support article. (OPSC-13332)
Lifecycle Manager is not currently compatible with DSE Configuration Encryption. See Encrypted DSE configuration values for more details. (OPSC-7529)

OPSC-8851, in 6.0.2, Improved resiliency of Lifecycle Manager in situations where there is high latency between the OpsCenter daemon and nodes in the cluster. This release improves upon the problem; however, there are still known issues in high latency scenarios that will be addressed in a future release. (OPSC-9853)

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.

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

Known Issues Fixed in OpsCenter 6.0.8

- Kerberos: Setting the kerberos_keytab_location and kerberos_client_principal in the agent address configuration prevented the agent from starting properly. For more details and a workaround for OpsCenter versions prior to 6.0.8, see the KB support article. (OPSC-10860)

Known Issues Fixed in OpsCenter 6.0.6

- When modifying the Config Profile of an existing cluster in Lifecycle Manager, the Cluster Connection Settings in OpsCenter are now automatically updated after running a configure job. (OPSC-8544)
- Fixed an issue where OpsCenter failed to retrieve the diagnostic tarballs from the agents if SSL was enabled between OpsCenter and the agents. (OPSC-10701)

Known Issues Fixed in OpsCenter 6.0.5

- Fixed LCM repository authentication bug when special characters exist in credentials (as with most DataStax Academy usernames). (OPSC-10817)

Known Issues Fixed in OpsCenter 6.0.3

- Any encrypted config values generated since Opscenter 6.0 (and prior to fix in 6.0.3) will need to be re-encrypted. (OPSC-10244).
- Backups/Restores will not work with keyspace names longer than 32 characters on DSE 4.7 or 4.8. (OPSC-9563)
- The Repair Service fails to auto-restart after a node is decommissioned. Manually starting the Repair Service resolves this issue. (OPSC-9244)
- Some items in the Lifecycle Manager UI may not automatically update if they are modified outside of the current UI session; for example, via the API directly or in another UI session. If multiple users might be concurrently modifying the same cluster, please be sure to refresh the UI before making any changes. (OPSC-9306)
Known Issues Fixed in OpsCenter 6.0.2

- g1-gc-opts in `cassandra-env.sh` are not immediately editable when using G1 garbage collection by default. To workaround this issue, change the garbage collector to something other than G1 and back again. (OPSC-9556)
- Max heap size in `cassandra-env.sh` is not editable using the LCM UI. To workaround this issue, users can set the `-Xmx` and `-Xms` JVM properties directly via `additional-jvm-opts` further down on the `cassandra-env.sh` section in the Config Properties page. (OPSC-9546)
- The `use_tls` setting in email alerts does not currently work as expected. Users can still configure email alerts to work with TLS-enabled servers by setting `use_ssl=1` and `use_tls=0`. Please contact DataStax Support if you have any issues. (OPSC-9451)
- Automatic definition file updates are not dynamically reloaded for new versions of DSE. If you see an error for "Unsupported or invalid version of DSE" in the UI, try restarting opscenterd. (OPSC-9468)
- Some users may see intermittent job failures with an IncompleteRead error (OPSC-8851; see More Details)
- The LCM UI has some rendering issues in older versions of Safari (<=8). The workaround is to use a newer version of Safari or another supported browser. (OPSC-9123)

Known Issues Fixed in OpsCenter 6.0.1

- opscenterd fails to properly resolve relative symlinks to Java. (OPSC-9344; see More Details)
- When installing an agent on a node for the first time, `address.yaml` is owned by the root user. The only OpsCenter functionality this affects directly is Automatic Failover, which will not work until ownership or permissions are updated. If the agent has previously been installed on the node, ownership is not affected. (OPSC-9336; see More Details)
- S3 and Local FS backups fail for keyspaces leveraging the new Materialized Views feature in DSE 5.0. On Server backups are not affected. (OPSC-9328; see More Details)
- Users must ensure tables that leverage the new User Defined Aggregates and User Defined Functions features in DSE 5.0 exist prior to running a restore. OpsCenter cannot automatically re-create these tables, but can successfully restore the data to existing tables. (OPSC-9261; see More Details)
Installing OpsCenter

Installing OpsCenter from the RPM package

Install the DataStax 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 (page ).

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.
- Oracle Java SE Runtime Environment 8 (JRE or JDK) or OpenJDK 8. Earlier or later versions are not supported. See installing the Oracle JDK (page ) or OpenJDK.
  
  **Tip:** Use OpsCenter Lifecycle Manager to automatically manage Java and JCE installs (page 442) 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.
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 154) 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 223) an existing cluster or provision a new cluster in Lifecycle Manager (page 391).

Installing OpsCenter from the Debian package

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

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

The OpsCenter Debian and Ubuntu packaged releases runs 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:

- Be sure your platform is supported (page ).
• Aptitude Package Management (APT) application.
• Root or sudo access.
• **Oracle Java SE Runtime Environment 8 (JDK)** (page 381) (1.8.0_40 minimum) or OpenJDK 8. Earlier or later versions are not supported.
• Python 2.7.x
• DataStax recommends using a recent version of one of the major web browsers. OpsCenter does not support Internet Explorer or Microsoft Edge.

Also see [Recommended production settings](#) and the **DataStax Enterprise Reference Architecture** white paper.

### Table 2: Hardware requirements

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Minimum</th>
<th>Production</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPUs</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>Memory</td>
<td>8 GB</td>
<td>24 GB</td>
</tr>
<tr>
<td>Data directory</td>
<td>20 GB</td>
<td>200 GB</td>
</tr>
<tr>
<td>Commit log directory</td>
<td>20 GB</td>
<td>200 GB</td>
</tr>
<tr>
<td>Saved caches directory</td>
<td>20 GB</td>
<td>200 GB</td>
</tr>
<tr>
<td>Logs directory</td>
<td>20 GB</td>
<td>200 GB</td>
</tr>
</tbody>
</table>

Production requirements depend on the volume of data and workload.

Minimum hardware requirements for the machine on which OpsCenter runs:

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

**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. Verify that a required version of Java is installed:

   ```
   java -version
   ```

   If not Oracle Java 8, or OpenJDK, see [Installing the JDK](#).

   **Important:** Package management tools do not install Oracle Java.

In a terminal window:
2. 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

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

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

4. 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 154) 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 223) an existing cluster or provision a new cluster in Lifecycle Manager (page 391).

Installing OpsCenter with a tarball on any Linux distribution

Install DataStax 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 (page ).

Prerequisites:

Minimum hardware requirements for the machine on which OpsCenter runs:

   • 2 CPU cores
Installing OpsCenter

- 2 GB of RAM available to OpsCenter

Permission and software requirements:

- Oracle Java SE Runtime Environment 8 (JRE or JDK) or OpenJDK 8. Earlier or later versions are not supported. See installing the Oracle JDK (page 400) or OpenJDK.

  **Tip:** Use OpsCenter Lifecycle Manager to automatically manage Java and JCE installs (page 442) 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.

   ```
   $ curl -L http://downloads.datastax.com/enterprise/opscenter.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 223) an existing cluster or provision a new cluster in Lifecycle Manager (page 391).

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 48) package.

1. Stop (page 69) OpsCenter.

2. Open a terminal and enter the following command:

   ```
   $ sudo yum remove opscener
   ```

Uninstalling an OpsCenter Debian package
Use this method if OpsCenter was installed using a Debian (page 49) package.

1. Stop (page 69) OpsCenter.

2. Open a terminal and enter the following command:

   ```
   $ sudo apt-get purge opscener
   ```

Uninstalling an OpsCenter binary tarball
Use this method if OpsCenter was installed using a tarball (page 51).

1. Stop (page 69) OpsCenter.

2. Open a terminal and enter the following command:

   ```
   $ 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.
Installing OpsCenter

If there are any issues with installing agents, a banner at the top of the OpsCenter workspace is displayed. The banner cannot be dismissed until the 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.

![Agent Status Banner]

Clicking the **# problems** link opens the **Agent Status** (page 201) tab where you can view more detailed information about agent status, view troubleshooting suggestions, and access the **Install Agents** button to retry installing the agents.

**Related information:**

- **Agents View** [View the status of agents. Agent status automatically updates as the information becomes available within OpsCenter.] (page 201)

**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 223), 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 201) 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 154), or accepted the default SSH port (22) for node-agent communication.
- DataStax Enterprise (page ) 4.7 or greater.

1. Install OpsCenter (page 48).

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](http://opscenter_host:8888), where *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 57)*.

   The Install Agents - Credentials dialog appears.

   ![Install Agents - Credentials dialog](image)

8. Enter SSH credentials to connect to your nodes:

   a. Enter a **Username**. The user must have root or sudo privileges.

   b. A sudo password can be entered in the **Password** box.

   c. Enter a **Private Key**.
No information entered in the Install Agents - Credentials dialog is saved or stored.

9. Click Install. The Agents tab (page 201) opens and displays the progress of the agent installation.

DataStax agents have been deployed and configured for each managed node in the DataStax Enterprise cluster.

If you were unable to install the agents through the OpsCenter UI, follow the instructions to manually install the agents (page 57).

**Manually installing DataStax agents**

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 (page ) 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 (page ) 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.

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

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.
   
   # yum install datastax-agent

4. In address.yaml (page 176), 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 (page ) 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 (page ) 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 176) 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 176).

   ```
   $ 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` (page ) 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` (page ) 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.
Installing OpsCenter

- 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).

   ```bash
   $ 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.

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

3. Install the DataStax agent.

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

4. In `address.yaml (page 176)`, set `stomp_interface` to the IP address that OpsCenter is using. You might have to create the `address.yaml` file.

   ```bash
   $ 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 176)`.

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

6. Start the DataStax agent.

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

**Setting permissions to run the agent as a different user**

Running the agent as the same user running DataStax Enterprise is highly recommended because directory and file permissions do not need to be set manually. By default, the DataStax agent is installed with the Installer-Services or package installations and runs as the same user as DataStax Enterprise, which is cassandra. When installing the agent from...
Installing OpsCenter

Installer-No Services or tarball installations, you must manually configure the agent and DataStax Enterprise as different users.

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

<table>
<thead>
<tr>
<th>Feature functionality</th>
<th>Permissions required</th>
</tr>
</thead>
<tbody>
<tr>
<td>General agent functionality</td>
<td>Read permission to <code>cassandra.yaml</code></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 umask <em>(page 62)</em> must also be set to accommodate group permissions for new tables and data.</td>
</tr>
<tr>
<td></td>
<td>• If commitlog archiving <em>(page 281)</em> 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 4: 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 Installer-Services or package installations: /etc/dse</td>
</tr>
<tr>
<td></td>
<td>• DataStax Enterprise Installer-No Services or tarball installations: <em>install_location/conf</em></td>
</tr>
<tr>
<td>Data directories</td>
<td>Default: <code>/var/lib/cassandra</code></td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: Location is user-configurable; set in <code>cassandra.yaml</code>.</td>
</tr>
<tr>
<td>Commitlog archiving script</td>
<td>• Agent package install: <code>/usr/share/datastax-agent/bin/archive_commitlog.sh</code></td>
</tr>
<tr>
<td></td>
<td>• Agent tarball install: <code>install_location/bin/archive_commitlog.sh</code></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.

```bash
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:

```bash
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:

```bash
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 5: 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 154) option in <code>opscenterd.conf</code>.</td>
</tr>
</tbody>
</table>

```
<table>
<thead>
<tr>
<th>Port/Protocol</th>
<th>Source</th>
<th>Destination</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<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] port</code> (page 154) option in <code>opscenterd.conf</code>.</td>
<td></td>
</tr>
<tr>
<td>7199/TCP</td>
<td>DataStax agent</td>
<td>DSE</td>
<td>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 <code>[jmx] port</code> (page 168) option in <code>cluster_name.conf</code>, or if necessary, in <code>address.yaml</code>.</td>
</tr>
<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 (page 73). When SSL is on, 8888 redirects to 8443 and serves no content. Configure with the <code>[webserver] ssl_port</code> (page 155) in <code>opscenterd.conf</code>.</td>
</tr>
</tbody>
</table>
## Installing OpsCenter

<table>
<thead>
<tr>
<th>Port/Protocol</th>
<th>Source</th>
<th>Destination</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSE nodes</td>
<td>OpsCenter server</td>
<td></td>
<td><strong>LCM Meld configs and job-events.</strong> 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] <code>ssl_port (page 155)</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><strong>SMTP for email alerting.</strong> See [Enabling SMTP email alerts](page 123). Configure in one or more uniquely named <code>config_location/event-plugins/email.conf</code> files.</td>
</tr>
</tbody>
</table>
| 9042/TCP           | OpsCenter server      | DSE         | **Native transport:** The native transport port for the cluster configured in `native_transport_port` in `cassandra.yaml`. Port 9042 must be open from the OpsCenter server to all DSE nodes. Configure with the `cql_port (page 168)` option in `cluster_name.conf`.  
**Note:** The port must also be open on a [storage cluster](page 133). Configure with the [storage_cassandra] `cql_port (page 170)`. |
| 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.                                                                 |

---

**Note:** The port must also be open on a [storage cluster](page 133). Configure with the [storage_cassandra] `cql_port (page 170)`.
<table>
<thead>
<tr>
<th>Port/Protocol</th>
<th>Source</th>
<th>Destination</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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 159) 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 177) 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 172) in cluster_name.conf or in address.yaml. |
| 22/TCP       | OpsCenter server | LCM deployment targets and agent install deployment targets | 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 158) in opscenterd.conf.  
**Note:** The SSH Management Port can be overridden at the cluster, datacenter, or node level in the Clusters (page 391) workspace of Lifecycle Manager (page 387). |
<table>
<thead>
<tr>
<th>Port/Protocol</th>
<th>Source</th>
<th>Destination</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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 139). Configure with the [definitions] download_port (page 156) 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 102). Configure with the [ldap] server_port (page 157) 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 157) option in opscenterd.conf.</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 214) in the Spark console access from the Node Details dialog within OpsCenter. Configure with the [spark] master_http_port (page 173) option in cluster_name.conf. For information about defining a datacenter in OpsCenter LCM for Spark workloads, see Adding a datacenter (page 398). For information about using Spark with the DataStax Enterprise platform, see analyzing data using Spark (page ).</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>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 161) option in opscenterd.conf.</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 119) traps. Configure with one or more uniquely named config_location (page 69)/event-plugins/snmp.conf 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 149) metrics. Configure with the [labs] graphite_port (page 175) option in cluster_name.conf or address.yaml if necessary.</td>
</tr>
<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 445) section of the Config Profile in the Lifecycle Manager UI.</td>
</tr>
<tr>
<td>50030/TCP</td>
<td>OpsCenter server</td>
<td>DSE</td>
<td>Hadoop: Sets the Hadoop HTTP job tracker port. Configure with the [hadoop] job_tracker_http_port (page 173) in cluster_name.conf.</td>
</tr>
</tbody>
</table>
### 50031/TCP
- **Source**: Management workstations
- **Destination**: OpsCenter server
- **Description**: Hadoop: OpsCenter HTTP proxy for Hadoop Job Tracker. The `opscenterd` daemon listens on this port for incoming HTTP requests from the browser when viewing the Hadoop Job Tracker page directly. Configure with the `opscenterd.conf` option in `opscenterd.conf`.

### 9260/TCP
- **Source**: OpsCenter server
- **Destination**: Hadoop Job Tracker client
- **Description**: Hadoop: Hadoop Job Tracker website port. The Job Tracker listens on this port for HTTP requests. If initiated from the OpsCenter UI, these requests are proxied through the `opscenterd` daemon; otherwise, they come directly from the browser. Configure with the `[hadoop] job_tracker_port` option in `cluster_name.conf`.

---

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>/var/lib/opscenter</td>
<td>SSL certificates for encrypted agent/dashboard communications</td>
</tr>
<tr>
<td>/var/log/opscenter</td>
<td>Log directory</td>
</tr>
<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 for running OpsCenter as a service:

```
/etc/init.d
```

- **Tarball installs**

  Starting opscenterd:

  ```
  $ install_location/bin/opscenter ##Use -f to start in the foreground.
  ```

  Stopping and restarting opscenterd:

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

     ```
     $ ps -ef | grep opscenter
     $ sudo kill pid
     ```

  2. To start OpsCenter:

     ```
     $ install_location/bin/opscenter ##Use -f to start in the foreground.
     ```

**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

**Starting and restarting DataStax agents**

- **To start the DataStax agent:**

  ```
  # Package installations:
  $ sudo service datastax-agent start
  ```
**Note:** The DataStax agent starts automatically.

# Tarball installations:

```
$ install_location/bin/datastax-agent ##Use -f to start in the foreground.
```

- **To restart the DataStax agent:**
  - **Package installations:**
    ```
    $ sudo service datastax-agent restart
    ```
  - **Tarball installations:**

    1. Find the DataStax agent Java process ID (PID), kill the process using its PID number:
       ```
       $ ps -ef | grep datastax-agent
       $ sudo kill <pid>
       ```

    2. Start the agent:
       ```
       $ install_location/bin/datastax-agent ##Use -f to start in the foreground.
       ```
Upgrading OpsCenter

See the Upgrade Guide (page) for detailed instructions on upgrading OpsCenter.
Configuring OpsCenter

OpsCenter Security

Configuring SSL/TLS between OpsCenter and the DataStax Agents

OpsCenter uses 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. 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 free from network threats.
- In a situation where you are not concerned about someone listening to OpsCenter traffic.

Otherwise, you should enable SSL.

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

The SSL key generation process is done automatically at package install time. Unless you need to regenerate SSL key files, skip to 4 (page 75) to edit the proper configuration files to enable SSL.

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

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

[agents] ssl_certfile

The location of the SSL certificate used for SSL traffic between OpsCenter and the 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.

Prerequisites:
Configuring OpsCenter

opscenterd.conf
The location of the opscenterd.conf (page 72) 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 (page 72) file depends on the type of installation:

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

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

   **Note:** The setup.py script requires the /tmp directory to have exec permissions. Make sure that the /tmp directory does not have the noexec flag set. See Setting and securing the tmp directory for the agent (page 152).

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.

2. **Copy the OpsCenter SSL private key opscenter.key and the OpsCenter SSL certificate opscenter.der to /var/lib/opscenter/ssl:**

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

   The OpsCenter SSL private key and certificate are located in the following directory: /var/lib/opscenter/ssl

3. **Change ownership of the OpsCenter SSL private key opscenter.key and the OpsCenter SSL certificate opscenter.der to the opscenter user and the opscenter group:**

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

Page 74
$ sudo chown opscenter:opscenter /var/lib/opscenter/ssl/opscenter.der

The OpsCenter SSL private key and certificate are readable by the OpsCenter daemon process.

4. 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 177) for more details on the `use_ssl` option.

$ sudo vi /etc/opscenter/opscenterd.conf

```plaintext
[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 # for non-Java clients
agent_certfile = /var/lib/opscenter/ssl/agentKeyStore.der
```

5. Restart (page 69) the OpsCenter daemon.

6. 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 54) procedure.

   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`.

   ```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.
Configuring OpsCenter

- `sudo vi /var/lib/datastax-agent/conf/address.yaml`
  - `use_ssl: 1`

- d. Restart the agent.
  - `sudo service datastax-agent restart`

7. After `opscenterd` and all agents have been configured and restarted, verify proper agent connection through the Agent Status (page 201) 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`

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:

   ```bash
   $ 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.

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

   ```yaml
   [agents]
   ```
use_ssl = true

You can also configure the locations of the agent keystore or truststore and the associated keystore password generated in 1 (page 76).

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

```
[agents]
use_ssl = true
opscenter_ssl_keystore = /etc/opscenter/ssl/agentKeyStore
opscenter_ssl_keystore_password = new_password
opscenter_ssl_truststore = /etc/opscenter/ssl/agentKeyStore
```

See configuring the agent for ssl (page 177) for more details on the use_ssl option.

3. Restart (page 69) 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 54) 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 package installs, or install_location/ssl/agentKeyStore for tarball installs.

```
$ 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.

```
$ 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 install_location/conf/address.yaml
```
Configuring OpsCenter

use_ssl: 1

d. Restart the agent.

$ 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 201) tab.

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

opscenterd.conf

The location of the opscenterd.conf (page ) 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 (page ) 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 69) 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 54) 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.

```bash
$ sudo vi /var/lib/opscenter/address.yaml

use_ssl: 0
```

c. Restart the agent.

```bash
$ 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 201) 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`

`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.

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

[agents]
use_ssl = false
```

2. Restart (page 69) 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 54) 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 201) 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. 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 (page ) must be enabled on the DSE cluster.
- The keystores and truststores (optional) must be created for all DSE nodes. For more information, see Preparing server certificates for SSL encryption (page ) in the DSE documentation.
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.

   ```bash
   $ 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.

   ```bash
   $ 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.

   ```bash
   $ 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 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. 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.

   a. Configure the client-to-node settings using the **Edit Cluster Connection Settings** dialog in the OpsCenter monitoring UI. See Configuring cluster settings (page 90).

   b. Configure 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 133) (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.

```
cluster_name.conf

[cassandra]
# Note: If the truststore and keystore are the same file, enter the same path/password
# for both the keystore and truststore
```
Configuring OpsCenter

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
ssl_keystore = /path/to/dse/truststore/file/on/agent/machine
ssl_keystore_password = password_of_keystore

# 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

7. Restart OpsCenter (page 69).

Enabling HTTPS for the OpsCenter server

opscenterd.conf

The location of the opscenterd.conf (page ) 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 84).

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
#ssl_port = 8443

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 62)* 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 69).**

**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:**

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 `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 `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 83](#)) and `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.

**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 OpsCenter configuration file, `opscenterd.conf`.

2. In the `[webserver]` section, make the following changes as appropriate:
   
   a. If HSTS is enabled with a long max age, lower the `hsts_max_age` value.

   b. Save `opscenterd.conf` and **restart OpsCenter ([page 69](#)).**

   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 `opscenterd.conf` again. Comment out the ssl and max age options. Set `hsts_enabled` to **False** (or comment out) as shown in the following example:

```
[webserver]
```
4. Save `opscenterd.conf` and restart OpsCenter (page 69).

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 OpsCenter for Kerberos authentication**

OpsCenter can use Kerberos to authenticate to DataStax Enterprise clusters. If a DataStax Enterprise cluster uses Kerberos authentication, you need to create and configure the OpsCenter principals before adding the cluster to OpsCenter.

**Prerequisites:** Configure DSE for Kerberos. For more information, see Authenticating a cluster with Kerberos (page ) in the DataStax Enterprise Documentation and review the Kerberos tutorial (page ).

**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. Create an `opscenterd` principal and register it with DataStax Enterprise.

   ```bash
   $ cqlsh
cqlsh> create user 'opscenterd/Kerberos host@Kerberos domain';
   ```

   To view the users who are on the node, run the `list users` command in `cqlsh`.

   ```bash
   $ cqlsh
cqlsh> list users;
   ```

2. Create service principals for the OpsCenter agent user running on each node and register them with DataStax Enterprise. The default user name is `cassandra`. 
$ cqlsh

cqlsh> create user 'cassandra/Kerberos host@Kerberos domain';

Note: If you require running the agent as a different user than cassandra, see setting permissions (page 60) to run the agent as a different user.

3. **Create keytabs** for the cassandra principals and move them to the location of the datastax-agent directory on each node.

   Place the keytabs in the location configured for your environment. The default location is /usr/share/datastax-agent/krb5.keytab for package installations, or install_location/datastax-agent/krb5.keytab for tarball installations. The path to the keytabs is configurable using either the [kerberos] opscenterd_keytab_location (page 174) in cluster_name.conf, or the DataStax Agent configuration options (address.yaml) (page 180) in address.yaml.

4. Change the owner of the keytabs and the /datastax-agent directory to the cassandra user.

   The following example changes ownership of the /datastax-agent directory and keytabs for the default location in a package installation. Replace the paths below with your configured keytab location as appropriate for your installation type.

   
   ```
   $ sudo chown cassandra /usr/share/datastax-agent /usr/share/datastax-agent/krb5.keytab
   ```

5. When adding the cluster as described in adding an existing cluster (page 223), select **DSE security (kerberos) is enabled on my cluster**.

   The Kerberos fields expand.
6. Enter the information as applicable to your environment.
   
   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 client principal for the OpsCenter process/machine to use. Example: `opscenterd@YOUR_REALM`.
   
   c. Enter the location of the keytab OpsCenter machine, which contains credentials for the `opscenter_client_principal`.
   
   d. Enter the client principal for the DataStax Agent process/machine to use. Example: `agent@YOUR_REALM`.
   
   e. Enter the location of the keytab on the DataStax Agent machines, which contains credentials for the `agent_client_principal`. Example: `/path/to/keytab.keytab`.

**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 90) 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] job_tracker_client_principal
```
The job tracker client principal in Kerberos. (Example: user@realm).

[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

**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 [Configure capabilities by manually modifying the opscenterd.conf, cluster_name.conf, and address.yaml configuration files.](#) (page 153)

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:
Configuring OpsCenter

```yaml
[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 69).**

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 150).

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

The connections settings for a cluster define how OpsCenter connects to a cluster. For example, if you have enabled authentication or encryption on a cluster, you need to specify that information in the cluster connection settings.

1. **Select the cluster you want to edit from the** Cluster **menu.**

2. **Click** Settings#Cluster Connections. The Edit Cluster Connection Settings dialog appears.
3. If applicable, change the IP addresses of cluster nodes.

4. If applicable, change the JMX and Native Transport listen port (page 62) numbers.

5. If applicable, click Add credentials to add or edit the user credentials if the JMX or Native Transport ports require authentication.

6. If applicable to your environment, select DSE security (kerberos) is enabled on my cluster and complete the following 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 client principal for the OpsCenter process/machine to use. Example: `opscenterd@YOUR_REALM`.

c. Enter the location of the keytab OpsCenter machine, which contains credentials for the `opscenter_client_principal`.

d. Enter the client principal for the DataStax Agent process/machine to use. Example: `agent@YOUR_REALM`.

e. Enter the location of the keytab on the DataStax Agent machines, which contains credentials for the `agent_client_principal`. Example: `/path/to/keytab.keytab`.

For more information, see **Authenticating with Kerberos** (page 432) and the Kerberos tutorial (page 243) in the DataStax Enterprise documentation.

7. **Note**: For more information, see **Enabling client-to-node encryption in OpsCenter** (page 80)

Select **Client-to-node encryption is enabled on my cluster**. Enter the following for OpsCenter to use to directly connect to Cassandra:

- If the paths are the same for the OpsCenter Keystore and Truststore (default), select **My Keystore and Truststore are the same** and enter the path and password.
- If the paths are not the same, enter the **OpsCenter Keystore Path**, **Truststore Path**, and the corresponding **Password** for each.

8. Click **Save Cluster**.

**Configuring 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 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. A role can be applied per cluster.
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 define custom roles for users. The permissions of the custom user roles are applied per cluster. Any functionality in OpsCenter that a user does not have permission for appears as gray and unavailable to that logged in user.

**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.

Role permissions

When defining custom roles, each role can have specific permissions enabled for that role. Role permissions are applied per cluster.
### Table 6: 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 53)</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 90)</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 277)</em></td>
<td>Allows users to perform backups and restores.</td>
</tr>
<tr>
<td>Best Practice Service <em>(page 337)</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 324)</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 352)</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 362)</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 and stop DSE nodes. Start and stop nodes from the Other Actions menu options available in the List view <em>(page 199)</em>, or from the Actions menu in the Node Details view.</td>
</tr>
<tr>
<td>Cleanup</td>
<td>Allows users to run a cleanup on one or more keyspaces.</td>
</tr>
<tr>
<td>Compact</td>
<td>Allows users to run compaction on a keyspaces and their tables. Major compactions are not recommended unless there is a compelling reason to do so.</td>
</tr>
<tr>
<td>Drain</td>
<td>Allows users to drain a node. The Drain option is available from the Actions menu in the Node Details dialog view, and also available when restarting DSE on a node <em>(page 217)</em>.</td>
</tr>
<tr>
<td>Permission</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Flush</td>
<td>Allows users to flush a keyspace and its tables. Flushing a keyspace might affect system performance when there are many live, large memtables.</td>
</tr>
<tr>
<td>Garbage Collection</td>
<td>Allows users to perform garbage collection on nodes. Running GC causes a spike in latency. The Perform GC option is available from the Other Actions menu in the List view (page 199), or from the Actions menu in the Node Details dialog view.</td>
</tr>
<tr>
<td>Repair</td>
<td>Allows users to run an ad hoc repair operation on selected nodes in the List view.</td>
</tr>
<tr>
<td>Data</td>
<td></td>
</tr>
<tr>
<td>Data Explorer</td>
<td>Deprecated. As a result of moving from thrift to native transport, the Data Explorer feature has been removed from OpsCenter. The Data Explorer feature in OpsCenter has been deprecated in favor of DataStax DevCenter, a visual CQL tool. Find more information about DevCenter and a link to download at <a href="http://www.datastax.com/what-we-offer/products-services/devcenter">http://www.datastax.com/what-we-offer/products-services/devcenter</a>.</td>
</tr>
<tr>
<td>Modify Schema</td>
<td>Allows users to modify the tables (page 221) (delete or truncate) in the Data workspace of OpsCenter.</td>
</tr>
<tr>
<td>Truncate Data</td>
<td>Allows users to truncate data (page 222) in a table.</td>
</tr>
<tr>
<td>View Schema</td>
<td>Allows users to view the CQL statements for the schema in the Data workspace of OpsCenter.</td>
</tr>
<tr>
<td>Cluster Topology</td>
<td></td>
</tr>
<tr>
<td>Add Nodes</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>Rebalance Cluster (non-vnode)</td>
<td>Allows users to rebalance a non-vnode cluster (page 228). Not applicable to vnodes.</td>
</tr>
<tr>
<td>Move</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 Move option from the Other Actions menu available in the List view (page 199), or from the Actions menu in the Node Details dialog view.</td>
</tr>
<tr>
<td>Decommission</td>
<td>Allows users to decommission a node from the Actions menu in the Node Details dialog view.</td>
</tr>
</tbody>
</table>
## Configuring OpsCenter

### Permission

<table>
<thead>
<tr>
<th>Permission</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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`. Changing the default admin password is strongly recommended the first time you log in.

If you enable OpsCenter authentication, DataStax strongly recommends enabling SSL communication (page 73) 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 69).

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 96) 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 101) 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.

   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]

   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.
Configuring OpsCenter

2. To add a user:
   a. Click **Add User**.
      The **Add User** dialog appears.

   b. Enter the username, password, and select a role for the user.

c. Click **Save**.

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

5. To edit a role:

a. Click **Manage Roles**.

The **Manage Roles** dialog appears.

![Manage Roles dialog](image)

b. To edit an already-existing role, click the **Edit** icon.

The **Edit Role** dialog appears.

![Edit Role dialog](image)

c. Select the cluster to apply role permissions to.
d. Select the options the user role has permissions for.

e. Click Save. Repeat these steps to apply role permissions for each cluster.

6. To add a role and assign it to users:

a. Click Add Role.

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

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

d. In the Role list, select the role, and click Save.

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

7. To 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 (page ) 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.
Configuring OpsCenter

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.

2. **Restart OpsCenter (page 69).**

**Configuring the user password hash algorithm**

Configure the algorithm to hash user passwords 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.

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

2. **Restart OpsCenter (page 69).**

3. Instruct users to log in again (page 96) 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.

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. 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.

LDAP configuration is extremely flexible with many configuration options possible within OpsCenter. To peruse all of the available configuration options, see OpsCenter configuration properties (page 154). 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 96) and subsequently switched to implementing LDAP, delete the old passwd.db.
- Roles: If there are LDAP roles, create and mirror in OpsCenter the user role names (page 109) 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.

**Note:** Users can only have one role in OpsCenter, regardless of whether they have multiple roles in LDAP. If OpsCenter roles match multiple user roles from the group_search_filter_with_dn filter, an authentication failure
results because OpsCenter cannot determine which role to grant the user. See troubleshooting LDAP (page 110).

1. Open the opscenterd.conf file for editing.

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

<table>
<thead>
<tr>
<th>Option</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>passwd_db</td>
<td>./ passwd.db (provide only because it contains the required OpsCenter user role information)</td>
</tr>
<tr>
<td>enabled</td>
<td>Set to True to enable LDAP authentication.</td>
</tr>
<tr>
<td>authentication_method</td>
<td>Set to LDAP, regardless if configuring Active Directory.</td>
</tr>
</tbody>
</table>

```
[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>Option</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>server_host</td>
<td>The host name of the LDAP server.</td>
</tr>
<tr>
<td>server_port</td>
<td>The port on which the LDAP server listens. For example, 389 or 636. 389 (page 62) is the default port for non-SSL LDAP and AD. 636 (page 62) is the default port for SSL LDAP and AD. For more information about ports, see OpsCenter ports (page 62).</td>
</tr>
<tr>
<td>uri_scheme</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 used with the -H flag and with the URI ldap.conf(5) option. Defaults to ldap for ldap_security = None; defaults to ldaps for ldap_security = SSL or TLS.</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</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. <strong>Note:</strong> If the <code>search_dn</code> and <code>search_password</code> (that constitute the search user entry point for locating users in LDAP) are omitted from the configuration, LDAP attempts to make an <em>anonymous bind</em> to perform the user search.</td>
</tr>
<tr>
<td><strong>search_password</strong></td>
<td>The password of the <code>search_dn</code> user.</td>
</tr>
<tr>
<td><strong>user_search_base</strong></td>
<td>The search base for your domain, used to look up users. Set the <code>ou</code> and <code>dc</code> elements for your LDAP domain. For example, this can be set to <code>ou=users,dc=domain,dc=top level domain</code>. More specifically: <code>ou=users,dc=example,dc=com</code>. Active Directory uses a different user search base. For example: <code>CN=search,CN=Users,DC=Active Directory domain name,DC=internal</code>. More specifically: <code>CN=search,CN=Users,DC=example-sales,DC=internal</code>.</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 <code>(uid={0})</code>, which looks for a user by unique user identifier. The value of the <code>{0}</code> variable is the username provided when logging in to OpsCenter. When using Active Directory, set the filter to <code>{sAMAccountName={0})</code>. <strong>Note:</strong> There is a known limitation in OpsCenter when using search filters for Active Directory. See troubleshooting LDAP (page 110).</td>
</tr>
<tr>
<td><strong>group_search_base</strong></td>
<td>The ldap search base used to find a group. Example: <code>ou=groups,dc=qaldap,dc=datastax,dc=lan</code></td>
</tr>
</tbody>
</table>
**group_search_filter**

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 AD implementations that have this configuration option already set, the `group_search_filter_with_dn` overwrites the returned value with the user's DN.

**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 the deprecated `group_search_filter`. Example: (member={0}).

**group_name_attribute**

The ldap field name used to identify a group's name. For example: cn.

**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

**usermemberof_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.
<table>
<thead>
<tr>
<th><strong>group_search_type</strong></th>
<th>Defines how group membership is determined for a user. Available options:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• <code>directory_search:</code> <em>(default)</em> Performs a subtree search of <code>group_search_base</code> using <code>group_search_filter</code> to filter the results.</td>
</tr>
<tr>
<td></td>
<td>• <code>memberof_search:</code> Gets groups from the <code>user_memberof_attribute</code> of a user. Using this option requires the directory server to have memberof support. When using the <code>memberof_search</code> rather than <code>directory_search</code> for group searches, you do not need to specify the <code>group_search_base</code> or <code>group_search_filter</code> options.</td>
</tr>
</tbody>
</table>
### 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`, `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).

### truststore_pass

The password to access the truststore.
**connection_timeout**  The number of seconds to wait before concluding that the LDAP server is down. Default: 20 seconds.

The following example configuration reflects a typical SSL LDAP (OpenLDAP or Oracle) implementation. The **server_port** value of 636 (page 10) is for an SSL configuration. If the **search_dn** and **search_password** 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 and group categories to authenticate a user (**user_search_base and user_search_filter; group_search_base and group_search_filter**). The **group_search_type** is **directory_search**. Note that **#user_search_base** and **#user_search_filter** are commented out in lines 14 and 15 because they are only applicable to 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={0})
14 #user_search_base = CN=search,CN=Users,DC=datastax,DC=internal
   # AD base
15 #user_search_filter = (sAMAccountName={0}) # AD filter
16 group_search_base = ou=users,dc=devldap,dc=datastax,dc=lan
17 group_search_filter_with_dn = (member={0})
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. Note that unlike the LDAP example above for OpenLDAP or Oracle, this AD config makes use of **user_search_base** line 12 and **user_search_filter** in line 13 for Active Directory configuration options. Also note the different user search base for AD shown in line 12.

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.

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

[ldap]
server_host = mywin2008.myCompany.lan
server_port = 636
uri_scheme = ldap
search_dn = CN=Administrator,CN=Users,DC=prodwin2008,DC=datastax,DC=lan
search_password = ****
user_search_base = CN=Users,DC=prodwin2008,DC=datastax,DC=lan # AD base
user_search_filter = (sAMAccountName={0}) # AD filter
directory admin_group_name = superusers
group_search_type = memberof_search
group_name_attribute = cn
user_memberof_attribute = memberof
user_memberof_stores_dn = True
ldap_security = SSL_TLS
truststore_type = JKS
truststore = /tmp/path_to_truststore_win2008
truststore_pass = secret
```

4. Restart OpsCenter (page 69) for the changes to take effect.

**Adding a role for an LDAP user**

When an LDAP user has been assigned LDAP roles, exactly one of those roles must also exist 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 roles assigned to a user. An LDAP user can belong to multiple LDAP roles; however, only one OpsCenter role can be named in the list of roles returned by the group search. OpsCenter grants the matching role to the user.

The group_search_type property indicates which method is used to determine LDAP group membership:

- If using directory_search, the group_search_filter_with_dn must return a list of LDAP roles that matches exactly one of the OpsCenter roles.
- If using memberof_search, the list of LDAP roles from the user's memberof attribute must match exactly one of the OpsCenter roles.

When LDAP is enabled, only role editing is supported in OpsCenter role-based security (page 92). Creating or editing users is disabled when LDAP is enabled because the users originate from the LDAP and are managed therein. When creating or editing user
Configuring OpsCenter

roles, OpsCenter LDAP supports non-ASCII character sets for the role name. Since 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:** After you configure the admin role in the `admin_group_name` configuration option, log in with a user that has that role so that 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:

```xml
<logger name="org.apache.directory" level="INFO" additivity="false"/>
```

to

```xml
<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 109)* 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 more than one defined OpsCenter role.**

The list of User's groups returned from LDAP must contain exactly one of the roles configured in OpsCenter. OpsCenter allows only one matching role. A user can only have one assigned role in OpsCenter so as to apply the proper permissions associated with the user role.

- If you are using a `group_search_type` of `directory_search`, modify your `group_search_filter_with_dn` to reduce the number of groups (roles) returned, so that OpsCenter finds only one match. The role corresponds to the user's assigned OpsCenter role.

- 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 has no more than one OpsCenter role.

For more details, see role prerequisites (page 102) and creating a role (page 109) for an LDAP user.

**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.

For more details, see role prerequisites (page 102) and creating a role (page 109) for an LDAP user.

**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 Opscenter. The list of User's groups returned from LDAP must contain exactly one of the roles configured in OpsCenter. See role prerequisites (page 102) and creating a role (page 109) 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.
Configuring OpsCenter

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.
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

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 on-the-fly, then transmitted and written in an encrypted state
to the relevant configuration files. Manually editing configuration files requires manually
encrypting the value and pasting it in the appropriate location. Use the OpsCenter system
key tool to manually encrypt (page 116) configuration values.

System encryption key
An 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. Decrypting values is not supported.
Table 7: System key encryption modes and strengths

<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.

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
- `[agents]`: ssl_keystore_password (monitored cluster), storage_ssl_keystore_password (storage cluster)
- `[agent_kerberos]`: keytab, ticket_cache
- `[ldap]`: search_password

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

**Note:** If you set the 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 agent configuration fields that require encryption include:

- `jmx_pass`
- `cassandra_pass`
- `monitored_cassandra_pass`
- `ssl_keystore_password` (storage cluster)
Configuring OpsCenter

- **monitored_ssl_keystore_password** *(monitored cluster)*

**cluster_name.conf**

The location of the **cluster_name.conf** *(page 72)* 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** *(page 72)* 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** *(page 72)* 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** *(page 72)* 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 69).**

**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 116)** 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`

**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:

   ```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]
   # Specifies whether opscenter should attempt to decrypt sensitive config values
   config_encryption_active = True
   # Name of the system key used to encrypt/decrypt stored passwords.
   config_encryption_key_name = opsc_system_key
   # Path to the encryption key. If left blank, the directory of opscenterd.conf will be used
   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.

2. 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.

   ```sh
   $ cp local/opsc_system_key ../agent/local
   ```
3. Stop OpsCenter *(page 69)* and stop the agents *(page 70).*

4. 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 pasting it in the appropriate location.

1. In your opscenterd directory, run the system key tool with the `value` parameter.

   ```
   pathTo/opscenterd $ bin/opscenter_system_key_tool value
   ```

2. Enter and confirm the value to encrypt.

   The system key tool displays the encrypted value.

3. 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.

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

4. Repeat these steps for each configuration value that requires encryption.

5. Restart OpsCenter *(page 69).*

**Deactivating configuration encryption**

Follow these steps to temporarily deactivate configuration encryption for any needed troubleshooting. After completing troubleshooting, activate configuration encryption *(page 114)* again.

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

**opscenterd.conf**

The location of the `opscenterd.conf` *(page )* file depends on the type of installation:
Configuring OpsCenter

- **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 69).

### 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
- 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 185).

**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:

```
<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>
```
3. To set structured data output, set \%X{structured} as shown in the following example:

```xml
<pattern>%date{ISO8601, UTC} \[%X{cluster_id}\] %msg ==%X{structured}== \{%thread\}%n%exception{20}</pattern>
```

4. **Restart OpsCenter (page 69).**

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 119) 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 8: 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...
Configuring OpsCenter

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, with the exception of an additional time field.

Table 9: OpsCenter MIB fields

<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/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 (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 10: SNMP configuration fields**

<table>
<thead>
<tr>
<th><strong>SNMP configuration options</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>enabled</td>
<td>Enables (1) or disables (0) the SNMP plugin instance. Default: 0 (disabled).</td>
</tr>
</tbody>
</table>
   | levels                        | 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:  
   |                               | • DEBUG  
   |                               | • INFO  
   |                               | • WARN  
   |                               | • ERROR  
   |                               | • CRITICAL  
   |                               | • ALERT  
   | clusters                      | 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. |
## Configuring OpsCenter

### SNMP configuration options

| **engine_id** | 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](https://tools.ietf.org/html/rfc3411). |
| **Format Schemes** | The fifth octet in the engine_id determines the format scheme that specifies the nature of the remaining trailing octets. Available format schemes: |
| | • 01: IPv4 Address scheme |
| | • 02: IPv6 Address scheme |
| | • 03: MAC Address scheme |
| | • 04: Text Address scheme |
| | • 05: Octets scheme (default) |
| **target_ip** | IPv4 address of the SNMP trap target. Default: 127.0.0.1 |
| | **Note:** Only IPv4 is supported at this time. |
| **target_port** | Listening port on the SNMP manager to receive SNMP traps. Default: 162. |
| **use_snmpv3** | 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). |
| **community_name** | SNMPv1/2c-specific configuration. Community name for authentication with the SNMP trap target. Default: public. |
| **user** | Username for SNMPv3 authentication. Default: opscusername. |
| **auth_protocol** | SNMPv3 authentication hashing algorithm. Specifies the authentication protocol to use with SNMPv3. Available options: |
| | • SHA (default) |
| | • MD5 |
| | • NoAuth |
| **auth_key** | SNMPv3 authentication key. Authentication key as configured for your OpsCenter installation in your monitoring software. Default: auth1key. |
### SNMP configuration options

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>privacy_protocol</td>
</tr>
<tr>
<td>SNMPv3 privacy protocol. Available standards-compliant options:</td>
</tr>
<tr>
<td>• AES (default)</td>
</tr>
<tr>
<td>• DES</td>
</tr>
<tr>
<td>• NoPriv</td>
</tr>
<tr>
<td>Available Extended Security Options (ESO) from the SNMP Research Group:</td>
</tr>
<tr>
<td>• 3DES</td>
</tr>
<tr>
<td>• AES192</td>
</tr>
<tr>
<td>• AES256</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>privacy_key</td>
</tr>
<tr>
<td>Privacy key as configured for your OpsCenter install in your monitoring software. Default: privkey1.</td>
</tr>
</tbody>
</table>

### Enabling SMTP email alerts

OpsCenter can send alerts to multiple email addresses. To enable email alerts, edit the `config_location (page 69)/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.
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 69) 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=

smtp_host=smtp.gmail.com
smtp_port=465
smtp_user=mercury@gmail.com
smtp_pass=*********
smtp_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
```
Configuring OpsCenter

subject={[level_str]} OpsCenter Event on {cluster} - {message}

template=Message: {message}
    Time:    {datetime}
    Level:   {level_str}
    Cluster: {cluster}
    Node:    {node}
    User:    {user}

A custom email template example:

template=Hey support staff! Wanted to let you know that {cluster}
says, {message} This is
{level_str} level, and you may want to check out http://
support.lan/opscenter_alerts_guide for further information.

An example email alert:

[INFO] OpsCenter Event on C1 - OpsCenter shutting down.
June 29, 2015 at 8:46 AM

Hey support staff! Wanted to let you know
that C1 says, OpsCenter shutting down.
This is INFO level, and you may want to check
out http://support.lan/opscenter_alerts_guide
for further information.

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

```
target_node=None&event_source=OpsCenter&success=None&level=1&level_str=INFO&api_source_ip=None +starting+up.&source_node=None
```

The request body contains fields described in Alerts (page 119) 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 69) 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.

**opscenterd.conf**

The location of the **opscenterd.conf (page )** 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]`:

```
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 69)** the OpsCenter daemon.

9. **Verify (page 128)** 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=CRITICAL,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:
```
Configuring OpsCenter

```bash
#   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:

```bash
fields=message={message}
message_type=CRITICAL
```

Example form data displaying the message and cluster name:

```json
FORM DATA: array (
   'event-msg' => 'OpsCenter starting up.',
   'mixedKey' => 'cluster-C1',
   'textKey' => 'value',
)
```

Example JSON data displaying the message and cluster name:

```json
JSON DATA: {"event-msg": "OpsCenter starting up.", "mixedKey": "cluster:C1", "textKey": "value"}
```

Verifying that events are posting correctly

Set preferences to specify handling URL events posting on the receiving side. Follow the PHP script scenario to verify events are posting correctly.

1. Post events to a file such as /tmp/events on the web server host.

2. Create a script.
   
   URL: http://10.1.1.11/postOPScevents.php

   ```php
   <?php
   file_put_contents( '/tmp/events', print_r ( $_POST,true ), FILE_APPEND );
   ```
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:

```plaintext
Array
  ( [api_source_ip ] => 67.169.50.240
    [target_node ] => None
    [event_source ] => OpsCenter
    [user ] => None
    [ time ] => 131102598851602
    [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 HipChat room**

Post URL alerts to one or more HipChat rooms using the HipChat API v2. The HipChat event plugin posts an event message as customized in the message field to the HipChat room using the identity provided by the authorization token.

Multiple instances of the `config_location (page 69)` `hipchat_posturl.conf` file results in multiple instances of the event plugin. The HipChat 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 `hipchat_posturl1.conf`, `hipchat_posturl2.conf`, and so on. Multiple plugin instances facilitate flexibility when setting up alerts to different HipChat rooms for development and production clusters. All OpsCenter event plugins support multiple configurations files and subsequent plugin instances.

**Prerequisites:**

1. Create an authorization token with send notification (scope) privileges to the HipChat API.

2. Ensure Notifications are enabled for the appropriate HipChat user profiles.

3. Obtain the room_id for the HipChat rooms you intend to send alerts to. To locate a list of room IDs for your organization, hit this endpoint with your auth token.
Configuring OpsCenter

that has the View Room scope: https://api.hipchat.com/v2/room?max-results=1000&auth_token=your_auth_token.

1. In a terminal, open hipchat_posturl.conf for editing.

2. In the [posturl] section, 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. Enter your auth_token (API key) to access the HipChat API.

5. Enter the ID of the HipChat room to post to in the room_id option.

6. Save hipchat_posturl.conf and restart (page 69) the OpsCenter daemon.

7. Verify that the expected HipChat room received notifications of OpsCenter starting up.

Example hipchat_posturl.conf:

```conf
[posturl]
enabled=1

# levels can be comma-delimited list of any of the following:
# DEBUG, INFO, WARN, ERROR, CRITICAL, ALERT
# If left empty, all levels will be posted
levels=CRITICAL,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=prod1,prod2,prod3

# You can learn more about how to get authentication tokens at
# https://www.hipchat.com/docs/apiv2/auth
auth_token=your_hipchat_auth_token

# You can find a list of room IDs for your organization at
# https://api.hipchat.com/v2/room?max-results=1000&auth_token=<your_auth_token>
room_id=123

# 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}.
message=<b>{level_string} from {cluster}</b>:  {message}
```
# Background color of the message posted in the HipChat room.
# Valid values: yellow (default), green, red, purple, gray, random.
color=purple

# Determines how the message is treated by HipChat applications
# html:
#   Message is rendered as HTML and receives no special treatment.
#   Must be valid HTML and entities must be escaped (e.g.: '&amp;' instead of '&').
#   May contain basic tags: a, b, i, strong, em, br, img, pre, code, lists, tables.
# text:
#   Message is treated just like a message sent by a user.
#   Can include @mentions, emoticons, pastes, and auto-detected URLs (Twitter, YouTube, images, etc).
# Valid values: html (default), text.
format=html

---

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 help 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):

[cassandra_metrics]
Configuring OpsCenter

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.

class_name.conf

The location of the class_name.conf (page) 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 175)

Changing performance data expiration times

class_name.conf

The location of the class_name.conf (page) 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]
### 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`

If you do not want OpsCenter to store data in an OpsCenter keyspace on the DataStax Enterprise cluster being monitored, you can store the data on a separate DataStax Enterprise cluster. OpsCenter supports connecting to a DataStax Enterprise storage cluster when SSL is enabled.

**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.

**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.

Version availability for configuration options:
Configuring OpsCenter

- The SSL configuration options are only applicable to OpsCenter versions 5.1.1 and later.
- The `monitored_*` and `storage_ssl_keystore*` settings are only applicable to OpsCenter versions 5.2.2 and later.
- The `rolling_restart_retry_*` settings for both `[cassandra]` (monitored) and `[storage_cassandra]` sections are only applicable to OpsCenter versions 6.0.8 and later.

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`.
- If you are using SSL to access the storage cluster and have a CER-encoded certificate, use the following command to convert it:

  ```
  $ openssl x509 -inform der -in certificate.cer -out certificate.pem
  ```

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 = opsusr
password = opscenter
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`.

---

Page 134

OpsCenter 6.0 User Guide (Earlier version)
Available cluster configuration options for monitored and storage clusters:

[agents] ssl_keystore
   The SSL keystore location for agents to use to connect to CQL on the
   monitored cluster.

[agents] ssl_keystore_password
   The SSL keystore password for agents to use to connect to CQL on the
   monitored cluster.

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

[agents] storage_ssl_keystore_password
   The SSL keystore password for 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.

[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 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.

[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.

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:

```yaml
[agents]
ssl_keystore = /etc/dse/keystore
ssl_keystore_password = example
storage_ssl_keystore = /etc/dse/keystore_storage
storage_ssl_keystore_password = example2

[storage_cassandra]
username = foo
password = bar

[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.

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**
Configuring OpsCenter

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.]

**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.]

5. Restart OpsCenter *page 69* for the changes to take effect.

**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 filesystem locations:

- `/etc/opscenter/conf/definitions` for package installs
- `install_location/conf/definitions` for tarball installs

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 140) or disable (page 140) the automatic downloads altogether.

**Updating and configuring definitions files properties**

Updating and configuring definition files overview

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 140); otherwise, you must manually download (page 140) and update the definitions file.

*Note:* After manually downloading or changing properties in the definitions file, restart OpsCenter (page 69) 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.

```plaintext
[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_dir  
    The file system location where definition files are stored. The default location is `/etc/opscenter/definitions` for package installations and `install_location/conf/definitions` for tarball installations.

  auto_update  
    Specifies whether OpsCenter should automatically attempt to periodically update the definition files. The default value is True.

  download_host  
    The host that definition file update requests will be made to. The default host is `opscenter.datastax.com`.

  download_port  
    The port used to request definition file updates on `download_host`. The default port is 443.

  download_filename
```

OpsCenter 6.0 User Guide (Earlier version)
Configuring OpsCenter

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.

```
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 (page ).

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/6.0.0/definition_files.tgz
Note: The URL depends on the currently installed version of OpsCenter. Replace 6.0.0 as shown in the above URL with your currently installed version of OpsCenter.

2. If applicable, remove any existing files in the directory specified by definitions_dir in opscenerd.conf.

   The definitions directory defaults to /etc/opscenter/definitions/ for Debian or RPM installs; and to ./conf/definitions/ for tarball installs.

3. Unpack the definitions files tarball in the directory specified by definitions_dir in opscenerd.conf.

4. Restart opscenerd (page 69).

opscenerd.conf

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

- Package installations: /etc/opscenter/opscenerd.conf
- Tarball installations: install_location/conf/opscenerd.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 337) recommends enabling failover. When no backup is configured, the rule fails and sends an alert. After enabling failover (page 143), 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 in address.yaml to connect to the backup instance instead of the failing
primary instance.

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 145) 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 (uuid), 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 145).

Failover after-effects

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 147). 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 387) 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 410)* again allows the job to proceed to completion.

**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 *(page 99)*.

**address.yaml**

The location of the *address.yaml* *(page 293)* file depends on the type of installation:

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

**Enabling automatic failover**

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 OpsCenter 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 primary OpsCenter instance to the backup OpsCenter instance using the method you prefer, such as NFS mount or rysnc. See Backing up critical configuration data *(page 146)* for instructions on backing up the active OpsCenter instance to the standby instance.
3. 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.

**Package installation:**

```
*/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
```

**Tarball installation:**

```
*/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
```

**Warning:** When a failover occurs, you must manually stop the sync scripts on the former primary OpsCenter instance and start the sync scripts on the new primary OpsCenter instance. Failure to do so will result in configuration changes on the new primary OpsCenter instance being overwritten by stale files from the former primary OpsCenter instance.

4. If you want to override the default values, edit the [failover] (page 146) section of the OpsCenter configuration file `opscenterd.conf`.

**Note:** Making any changes to the `opscenterd.conf` file requires restarting OpsCenter (page 69).

**Table 11: 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 instance sends a heartbeat to the backup OpsCenter instance.</td>
<td>10</td>
</tr>
<tr>
<td>heartbeat_reply_period</td>
<td>Frequency (in seconds) with which the OpsCenter backup instance sends a heartbeat to the primary OpsCenter instance.</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: Do not mirror the failover_configuration_directory (page 163) across OpsCenter installs when configuring OpsCenter to support failover. | /var/lib/opscenter/failover/ (package installs)  
install_location/failover/ (tarball installs) |
| override_primary_redirect_url | If set, overrides the primary OpsCenter URL automatically generated during failover. Example: https://opscenter:80. This option provides more flexibility for setting a port in the redirect URL. | Empty. Allows OpsCenter to generate the redirect URL for the newly primary OpsCenter instance switched from backup OpsCenter instance status. |

5. On the backup OpsCenter instance, 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:

    $ 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 instance can take over as the primary OpsCenter instance, the backup OpsCenter instance deletes the primary_opscenter_location file in the event of a failover. After a failover, recreate
Configuring OpsCenter

the primary_opscenter_location file on the newly designated backup OpsCenter instance.

Failover configuration options reference

Reference of available configuration options in opscenterd.conf for OpsCenter failover.

[failover] heartbeat_period
How often OpsCenter should heartbeat to the backup.

[failover] heartbeat_reply_period
How often a backup OpsCenter should heartbeat to the primary Opscenter.

[failover] heartbeat_fail_window
The amount of time required before a lack of heartbeat triggers failover

[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.

Example non-default configuration

The default values for failover configuration can be adjusted for your environment requirements. Open opscenterd.conf for editing and add a [failover] section:

```
[failover]
heartbeat_period = 20
heartbeat_reply_period = 360
heartbeat_fail_window = 120
failover_configuration_directory = /myrsyncdir/opscenter/failover/
override_primary_redirect_url = https://opscenter:9999
```

The example increases the default values, sets a custom directory location for failover configuration (package installation), and overrides the primary redirect URL with a non-standard port.

Restart OpsCenter (page 69) after making configuration changes.

Backing up critical configuration data

The main OpsCenter configuration directory contains critical data, such as the opscenterd.conf configuration file. The cluster topology, configuration profiles, credentials, repositories, and job history for Lifecycle Manager (LCM) is stored in the lcm.db database. Your organization is responsible for backing up these files, in addition to the critical configuration data outlined in this procedure.

Important: DataStax strongly recommends enabling automatic failover (page 143) from the primary OpsCenter instance to a designated backup OpsCenter instance.
The following steps indicate mirroring data between a primary OpsCenter instance and a backup OpsCenter instance.

1. Mirror the contents of the main configuration directory on the primary OpsCenter instance to the backup OpsCenter instance.
   - /etc/opscenter (package installs)
   - install_location/conf (tarball installs)

   ```
   $ scp /etc/opscenter/* secondary:/etc/opscenter
   ```

   **Note:** Do not mirror the `failover_configuration_directory (page 163)` across OpsCenter installs when configuring OpsCenter to support failover.

2. Mirror the contents of the `persist_directory (page 161)` 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 installation:
   - /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.

3. Back up the Lifecycle Manager database, encryption key, and security certificates:

   a. 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
      ```

   b. 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
      ```
c. If Lifecycle Manager has generated any certificates (page 414) for clusters configured to use node-to-node or client-to-node encryption (page 432), mirror the Lifecycle Manager certificate authority.

- /var/lib/opscenter/ssl/lcm (package installs)
- install_location/ssl/lcm/cacerts (tarball installs)

```bash
$ scp -r /var/lib/opscenter/ssl/lcm secondary:/var/lib/opscenter/ssl/
```

4. If SSL is enabled, mirror the contents of the SSL configuration directory on the primary OpsCenter instance to the backup OpsCenter instance.

- /var/lib/opscenter/ssl (package installs)
- install_location/ssl (tarball installs)

```bash
$ scp /var/lib/opscenter/ssl/* secondary:/var/lib/opscenter/ssl
```

5. If OpsCenter role-based security (page 92) is enabled, mirror the roles and password database passwd.db:

- /etc/opscenter/passwd.db (package installs)
- install_location/passwd.db (tarball installs)

```bash
$ scp /etc/opscenter/passwd.db secondary:/etc/opscenter/passwd.db
```

## Configuring named route linking

opscenterd.conf

The location of the opscenterd.conf (page ) 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.
Configuring OpsCenter

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 69)*.

Sending OpsCenter metrics to a Graphite server

`address.yaml`

The location of the `address.yaml` *(page*) 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` *(page*) 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.

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. Do so for every node that requires a specific configuration override.

2. Add the following configuration options:

   ```yaml
   [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.
   ```

   ```yaml
   [labs]
   graphite_host = 127.0.0.1
   graphite_port = 2003
   graphite_prefix = opscenter
   ```

3. Save the configuration file or files.

4. Restart (page 69) the OpsCenter daemon.

5. If you made changes to `address.yaml` (page 176), restart (page 70) the DataStax agents.

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 69).

**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:

```bash
$ export OPSC_JVM_OPTS=-Xmx2048m
$ sudo service opscenterd restart
```

Update the maximum heap size JVM argument for a package installation in `bin/opscenter`:

```bash
$ sudo service opscenterd restart
```

### 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 should require very little heap. The agent ships with a default max heap size of 128M. The most common cause of memory issues on the agent are clusters with a large number of tables. Each additional table in the cluster increases the number of metrics collected as well as the size of the metadata collected for cluster operations such as backups. Clusters that have more than 100 tables might require updates to the agent max heap size. If an agent is experiencing OutOfMemory issues, try changing the \(-Xmx\) property in `datastax-agent-env.sh` to 512 (-Xmx512M). If OutOfMemory issues persist, try increasing heap size one more time to \(-Xmx1024M\). 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.

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`

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 a different user (page 60).

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.
Configuring OpsCenter

JVM_OPTS="$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 70).

Configuration files

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 90) 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 cluster_name.conf.

• 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 cluster_name.conf configuration file took precedence over settings in address.yaml. In OpsCenter version 5.2 and going forward, the reverse is true: address.yaml settings take precedence over cluster_name.conf. To summarize, OpsCenter 5.2 and later configuration files precedence, settings in address.yaml override settings in 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 (page ).

cluster_name.conf

The location of the cluster_name.conf (page ) file depends on the type of installation:
Configuring OpsCenter

- **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`

**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`

**OpsCenter configuration properties**

**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`

These properties are configured in the `opscenterd.conf` file.

**Note:** After changing properties in this file, **restart OpsCenter (page 69)** 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 83).

```
[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 decrypt 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 /etc/opscenter/definitions for package installations and install_location/conf/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
   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 92) 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

[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] 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 opscentered.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 agents. The default interface is 0.0.0.0.

[agents] use_ssl
Specifies whether traffic between OpsCenter and the 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 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 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 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 agents when using SSL communication between OpsCenter and the agents. The default location is /var/lib/opscenter/ssl/agentKeyStore for package installations and install_location/ssl/agentKeyStore for tarball installations.

[agents] agent_keyfile_raw
The raw key that is stored in the java key store from agent_keyfile. This key is needed by non java clients that wish to communicate with OpsCenter.

[agents] agent_certfile
The location of the certfile sent to the agents when using SSL communication between OpsCenter and the 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.
Configuring OpsCenter

[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 used for temporary files used by OpsCenter. 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:

```plaintext
interval = 0
```

Additional configuration metric collection properties are available in Metrics Collection Properties (page 175).

[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.

[hadoop] base_job_tracker_proxy_port
The port to use for job tracker information. The interface, SSL key, and SSL cert are taken from the webserver section. The default port is 50031.

[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.

[cloud] accepted_certs
The location of the SSL CA certificate file used when provisioning new clusters or using the Backup Service. The default location is /var/lib/opscenter/ssl/cacert.der.

[labs] orbited_longpoll
This option increases the time between polling requests to orbited for data updates.

[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.

[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
How often, in seconds, to write the state to the persistence file for the repair service. The default value is 300 (5 minutes).

[repair_service] restart_period
How often in seconds to restart repairs. The default value is 300 (5 minutes).

[repair_service] cluster_stabilization_period
How often in seconds repair service checks for cluster state before resuming.

[repair_service] ks_update_period
The maximum age, in seconds, of a cached version of the current keyspace schema. The default value is 300 (5 minutes).

[repair_service] single_task_err_threshold
The number of times to retry a repair task before moving on to the next task. The default value is 10.

[repair_service] max_err_threshold
The maximum number of times to fail on a repair before cancelling the repair attempt. Errors during incremental repair do not count towards this threshold. The default value is 100.

[repair_service] max_parallel_repairs
The maximum number of repairs to run in parallel. If unspecified or set to 0, the system attempts to calculate 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.

[repair_service] max_pending_repairs
The maximum pending repairs allowed to be running on a node at one time. The default value is 5.

[repair_service] alert_on_repair_failure
Whether there should be alerts fired when a repair task fails. Defaults to true.

[repair_service] single_repair_timeout
The maximum length of time for a repair to complete, in seconds. The default value is 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. The default value is 5.

[repair_service] min_throughput
The minimum throughput needed to calculate parallel repairs. The default value is 512.

[repair_service] num_recent_throughput
The number of recent throughputs used to calculate the average throughput, which is then used to determine how many parallel repairs are needed. The default value is 20.

[repair_service] repair_estimation_factor
Estimated reduced efficiency due to other issues such as concurrent compaction.

[repair_service] error_logging_window
The desired amount of time in seconds between errors for exceeding max_parallel_repairs. Defaults to 86400

[repair_service] incremental_repair_tables
The list of keyspaces and tables to include in incremental repairs. (e.g. Keyspace1.Standard1, Keyspace1.Standard2)

[repair_service] incremental_range_repair
Determines whether incremental repairs run as subrange repairs (True) or repair the node's entire range (False). Default: False.

[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.

[repair_service] snapshot_override
Specifies whether to override the default snapshot repair behavior. The default value is False. Specifying this option as either True or False will always modify the behavior of the repair service. Snapshot repair can only be configured for Cassandra versions 2.0.11 and greater.

[u] 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.

[u] max_metrics_requests
The maximum concurrent metrics requests from the OpsCenter UI to opscenterd. The default value is 16.

[u] node_detail_refresh_delay
The time in seconds between polling calls to update node details. The default value is 5.

[u] 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.

[failover] heartbeat_reply_period
How often a backup OpsCenter should heartbeat to the primary Opscenter.

[failover] heartbeat_fail_window
The amount of time required before a lack of heartbeat triggers failover

[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.

[lifecycle_manager] db_location
The location of the database used for storing lifecycle management information.

[lifecycle_manager] cacerts_directory
Configuring OpsCenter

The directory to use when storing SSL-related files that are automatically generated during provisioning tasks.

```
lifecycle_manager] cacerts_truststore_location
  The location of the truststore containing the root CA cert for signing certificates.
```

```
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 management information. The encryption key is automatically generated if it does not already exist.
```

```
lifecycle_manager] job_manager_polling_interval
  Polling frequency in seconds when performing lifecycle management 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 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. The default value is 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. The default value is 3.
```

```
lifecycle_manager] ssh_retry_delay_in_seconds
  The number of seconds to wait between SSH connection retries during an LCM job. The default value is 10 seconds.
```

```
[agents] path_to_installscript
  The location of the script used to install agents. The default location is /usr/share/opscenter/agent/bin/install_agent.sh for package installations
```
Configuring OpsCenter

and `install_location/agent/bin/install_agent.sh` for tarball installations.

**[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. This value is used when installing the DataStax Agent automatically through OpsCenter and is written as the `stomp_interface` property in `address.yaml`. It is also used by the remote script deployed by the Lifecycle Manager (LCM) to communicate status back to the central opscenterd process. 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 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 69):

```
[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**
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**

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

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

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 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] 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 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] 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 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
Configuring OpsCenter

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**
The port used by 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 agents to use to connect to CQL on the monitored cluster.

**[agents] ssl_keystore_password**
The SSL keystore password for agents to use to connect to CQL on the monitored cluster.

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

**[agents] storage_ssl_keystore_password**
The SSL keystore password for agents to use to connect to CQL 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 agents for most HTTP operations. The default value is 10.

**[agents] concurrent_settings_requests**
The number of concurrent 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 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
Configuring OpsCenter

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 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.

[hadoop] job_tracker_port
Sets the Hadoop job tracker port. The default port is 9260.

[hadoop] job_tracker_http_port
Sets the Hadoop HTTP job tracker port. The default port is 50030.

[hadoop] job_tracker_proxy_port
Overrides the proxy port for job tracker. Use to prevent the proxy port from autoincrementing.

[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.
Configuring OpsCenter

[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] job_tracker_client_principal
   The job tracker client principal in Kerberos. (Example: user@realm).

[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 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

**[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.

**[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**

---

OpsCenter 6.0 User Guide (Earlier version)
Configuring OpsCenter

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 131)

---

**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 57), stomp_interface is the only property in most environments that needs to be explicitly configured. When automatically installing agents (page 54), stomp_interface is configured for you.

For more information about viewing agent status and troubleshooting agent issues, see Agents View (page 201).

---

**address.yaml**

The location of the address.yaml (page ) 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 (page ) 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 (page ) 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

**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`

**opscenter_ssl_keystore**
The SSL keystore location that the agents use to connect to opscenterd. Example:
`opscenter_ssl_keystore: /etc/opscenter/conf/.keystore`

**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**
The path to the truststore file that the agents use to connect to opscenterd.
Example: `opscenter_ssl_truststore: /etc/opscenter/conf/.truststore`
Configuring OpsCenter

**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`

**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`

**ec2_metadata_api_host**
The ec2 metadata api host, used to determine information about this node, if it is on ec2. Example: `ec2_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 tables (formerly referred to as column families) ignored by metrics collection. Example: `metrics_ignored_keyspaces: ks1, ks2, ks3`

**metrics_ignored_column_families**
Configuring OpsCenter

A comma separated list of column families that will be ignored by metric 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 133). 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.]`

**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.]

**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`

**kerberos_client_principal**
The Kerberos client principal to use when using Kerberos authentication within DSE. Example: `kerberos_client_principal: cassandra@hostname`

**storage_keyspace**
The keyspace that the agent will use to store data. Example: `storage_keyspace: OpsCenter`

**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: `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: `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: `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: `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: `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: `dse_binary_location: /usr/bin`

**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

hadoop_conf_location
The location of directory that holds hadoop-env.sh. When not set, the agent
attempts to auto-detect the location. Example: hadoop_conf_location: /etc/dse/hadoop

hadoop_log_location
The location of directory that holds hadoop logs. When not set, the agent attempts
to auto-detect the location. Example: hadoop_log_location: /var/log/hadoop/

userlogs

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,
elevated privileges. Default is true. Example: runs_sudo: true

destinations
Backup and restore destination definitions. Each destination is an
entry in the map with the destination id as the key and a map of
options specific to the type of destination. Example: destinations:
{"4798b1cd3a145f0b4fa8ef7b3e20309": {:throttle_bytes_per_second
"0", :path "s3-bucket"", :server_side_encryption "False", :provider
"s3", :access_key "key", :access_secret "secret"}}

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. Example:
backup_staging_dir: /var/lib/datastax-agent/clogs/

tmp_dir
The location of the Backup Service staging directory for backups. 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`

**remote_backup_retry_delay**
The delay in milliseconds between remote backup retries. Default: 5000. Example: `remote_backup_retry_delay: 5000`

**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`

**backup_file_queue_max**
The maximum number of files that may be queued for an upload to a remote destination. Increasing this number consumes more memory. Default: 10000. Example: `backup_file_queue_max: 10000`

**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: 30. 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`

**max-seconds-to-sleep**
When stream throttling is configured in Backup Service transfers to or from a remote destination, this setting acts as a cap on how long to sleep when throttling. The cap prevents prematurely closing connections due to inactivity. Default: 25 (seconds). Example: `max-seconds-to-sleep: 25`

**read-buffer-size**
The buffer size to read off the disk. Increasing this number may improve transfer speed but will consume more memory. Example: `read-buffer-size: 1000000`

**write-buffer-size**

The buffer size to write to the remote destination. Increasing this number may improve transfer speed, but will limit the ability of the throttler to slow transfers.

Example: `write-buffer-size: 10000`

**unthrottled-default**

A very large number used for bytes per second if no throttle is selected. Default: `10000000000`. Example: `unthrottled-default: 10000000000`

**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`

**multipart-chunk-size**

The chunk size used for ec2 s3 file transfers in bytes. Example: `multipart-chunk-size: 5000000`

**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 opscner 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`
**Configuring OpsCenter**

- **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. 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`

### Configuring DataStax Agents for Multi-Instance Nodes

Configure DataStax Agents for DataStax Enterprise Multi-Instance (page 58) (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 58) 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 Backup Service staging directory for backups. 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. Example: `backup_staging_dir: /var/lib/datastax-agent/clogs/`
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_address` must be unique for each agent.

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: `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: `stomp_interface: 127.0.0.1`

6. **Restart the DataStax Agents** *(page 70)*.

**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 69)* 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 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. Below is a sample default block.

```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 [http://logback.qos.ch/manual/appenders.html#SizeBasedTriggeringPolicy](http://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.

```
<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.jboss.netty" 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>
```
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
   - `stop-cassandra.example`: example shutdown script
   ```
   $ 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.
   ```
   $ 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 206)

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**
     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>private IP: 10.11.12.10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Node 1</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>public IP: 198.51.100.1</td>
<td>private IP: 10.11.12.1</td>
</tr>
<tr>
<td>Cassandra (cassandra.yaml)</td>
<td></td>
</tr>
<tr>
<td>broadcast_address: 198.51.100.1</td>
<td>listen_address: 10.11.12.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Agent (address.yaml)</th>
<th></th>
</tr>
</thead>
</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 387) 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 = 20
   ```

2. Open opscenterd.conf for editing and adjust the following settings:

   ```
   [agents]
   not_seen_threshold = 620
   http_poll_period = 500
   ```

   ```
   [ui]
   default_api_timeout = 10
   ```
3. **Restart OpsCenter (page 69).**
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 394) in Lifecycle Manager or manage an existing (page 223) 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 90) or remove the cluster (page 227) from OpsCenter.
  - # **Users & Roles** - Manage users (page 97) with role-based authentication.
- **Help** - Information resources for OpsCenter:
  - # **Help Center** - opens the current DataStax OpsCenter documentation.
  - # **Feedback** opens a form for sending feedback to the OpsCenter team.
  - # **Report** generates an HTML report (page 234) with information on the managed clusters.
  - # **Diagnostics** downloads a tarball that contains diagnostic information (page 233) about the nodes in a cluster.
- **Username** - If authentication (page 96) 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 387) for managing installation and configuration for DataStax Enterprise clusters.

- **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 195) or List (page 199) View) and perform certain maintenance operations (page 206) on cluster nodes. View the status of agents, troubleshoot, and install agents from the Agents (page 201) tab.

- **Dashboard** - View information about the clusters managed by OpsCenter and monitor a number of performance metrics (page 235). Real-time and historical performance metrics are available at different granularities: cluster-wide, per node, or per table. Organize your dashboards using presets (page 239), which you can also export and import (page 241) 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 keyspace and tables (page 219) (column families) within those keyspaces.

- **Services** - Enable DataStax Enterprise Management Services (page 277); including the Backup (page 277), Repair (page 324), Capacity (page 334), Best Practice (page 337), and Performance (page 352) services.
• **Hide/Show Clusters** - Toggle to hide or show the list of clusters in the left navigation pane.

## Node monitoring and administration

### 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 199) 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 206) on the node.
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.

<table>
<thead>
<tr>
<th>Outgoing</th>
<th>DC</th>
<th>Incoming</th>
<th>DC</th>
<th>Progress</th>
</tr>
</thead>
<tbody>
<tr>
<td>machine1</td>
<td>Cassandra</td>
<td>machine2</td>
<td>Cassandra</td>
<td>Running 40%</td>
</tr>
<tr>
<td>machine1</td>
<td>Cassandra</td>
<td>machine3</td>
<td>Cassandra</td>
<td>Running 40%</td>
</tr>
<tr>
<td>machine1</td>
<td>Cassandra</td>
<td>101.202.203.108</td>
<td>Cassandra</td>
<td>Running 40%</td>
</tr>
<tr>
<td>machine1</td>
<td>Cassandra</td>
<td>101.202.203.109</td>
<td>Cassandra</td>
<td>Running 40%</td>
</tr>
<tr>
<td>machine1</td>
<td>Cassandra</td>
<td>101.202.203.110</td>
<td>Cassandra</td>
<td>Running 40%</td>
</tr>
<tr>
<td>machine1</td>
<td>Cassandra</td>
<td>101.202.203.111</td>
<td>Cassandra</td>
<td>Running 40%</td>
</tr>
<tr>
<td>machine1</td>
<td>Cassandra</td>
<td>101.202.203.112</td>
<td>Cassandra</td>
<td>Running 40%</td>
</tr>
</tbody>
</table>
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 (page), 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 (page) 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.
Using OpsCenter

Select a check box next to the node or nodes to perform an operation on. Select the box next to the Datacenter column 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 206).

Filtering nodes

Filter the list of nodes by Datacenter, Name, Token, Status, and Load (CPU) 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:

- **Datacenter**: Filter by datacenter node type category of All, Analytics, Cassandra, or Solr.
- **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.

- **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 206) on the node.

View node details such as Status, Capacity, Uptime, and Memory Usage, including In Memory if in use.

---

Agents View

View the status of agents. Agent status automatically updates in the list view as the information becomes available within OpsCenter.

To access the Agents view:

- Click cluster#Nodes#Agents tab.

- If displayed in the Agent issues detected banner at the top of the OpsCenter workspace, click the Show Details#problems link.
Any DataStax Agents that need attention appear at the top of the list. If there is a problem with an agent connection to OpsCenter, the issue and status is clearly indicated in the appropriate column. When available, additional error details display by hovering the mouse pointer over the warning icon.

A Set Up Agents button is available to queue installation for any nodes that currently do not have the agent installed. The Agent Status column provides feedback about the individual agent. Expected values include:

- **Pending**: Agent installation is waiting for installation.
- **Installing**: Agent installation is in progress.
- **Error**: Agent installation failed. Hovering the mouse over the warning icon displays error message details.
- **Inactive**: Agent installation completed without error but OpsCenter detected no activity from the agent.
- **Active**: Agent installation completed without error and OpsCenter has detected some activity from the agent.

**Note**: Agents may still have problems with individual services with an Agent Status of Active. This is expected when agents are running but may be misconfigured.

- 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.
Using OpsCenter

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 53)

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 201).

**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.

**Viewing agent issues**

The Agents page provides comprehensive information about agent status; including troubleshooting information, links to documentation, and buttons that launch the agent install process:
Table 12: 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 54) or start agents, or follow the documentation to manually install agents (page 57).</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 Upgrade Guide (page ) to upgrade the current version of OpsCenter.</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>Issues</td>
<td>Recommendations</td>
</tr>
<tr>
<td>---------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------</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 73) 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 133) 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 195) views, you can add an alert (page 209) for nodes marked as down and configure alert notification (page 118) using email or POSTing to a URL.</td>
</tr>
</tbody>
</table>
| JMX is misconfigured.                                               | Check the JMX settings configured in address.yaml (page 176). 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 settings on the node. |

**Related information:**

Starting and restarting DataStax agents [Commands for starting and restarting DataStax agents for each type of installation.] (page 70)

**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, 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 section.

Managing single nodes

To run an operation on a single node, click that node from Ring View (page 195) or List View (page 199) and choose an action from the **Actions** menu from the node details view:

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 or stops the DataStax Enterprise process on the node.

**Restart**
Restarts the DataStax Enterprise process on the node. If running on multiple nodes, each node is started as soon as the start command for the previous node returns. If you want the operation to wait for thrift to be active on each node before continuing, use the **Rolling Restart** action.

**Cleanup**
Removes rows for which the node is no longer responsible. This is usually done after changing the partitioner tokens or the replication options for a cluster.

**Compact**
Perform a major compaction, which is not a recommended procedure in most DataStax Enterprise clusters.

Flush
Flushes to disk as persistent SSTables the recent writes currently stored in memory (memtables).

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.

Perform GC
Forces the Java Virtual Machine (JVM) on the selected node to perform a garbage collection (GC).

Decommission (single node only)
Removes a node from the cluster and streams its data to neighboring replicas.

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. Draining a node 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 enabled if vnodes are enabled.

Adding an alert for down nodes
Add an alert to notify when a node has been marked as down (page 256) 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. In the **Notify me when** menu, choose **Node Down**.

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**.

**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:

   ```yaml
   alias: nodeName1
   ```

3. **Restart the agent. (page 70)**

4. Repeat the above steps for each node.

**Monitoring in-memory usage**

Monitor in-memory usage from within OpsCenter. More information about creating or altering a table to use in-memory is available in the DataStax Enterprise documentation DSE In-Memory section.

A metric and an alert are available for monitoring in-memory usage:

- The **In-Memory Percent Used alert** (page 212) 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 236) 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**:

To view the in-memory usage of a node:

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

2. Click the node to view its details.
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 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 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.

### 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.

**Viewing Spark information**

**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.

*Note:* Clusters containing both Hadoop and Spark nodes are not supported for viewing Spark Details in OpsCenter (v 5.1.2). Future versions of OpsCenter will accommodate viewing Spark Worker Details directly from within OpsCenter, and viewing both Spark and Hadoop details for clusters that contain both node types.

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 210) for faster identification.

3. Under Spark Master, click the **View Spark Console** link.
The Spark console launches in another browser window.
Configuring an alert for KMIP errors

Configure an alert to monitor KMIP server status. For more information, see configuring KMIP to use off-server encryption keys (page ... ) in the DataStax Enterprise documentation. 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.

---

**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 199) 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 230)

**Decommission a node**

Decommission a node using OpsCenter Nodes administration.

1. Click *cluster name* # Nodes.

2. In the Ring *(page 195)* or List *(page 199)* view, select the node to decommission.

   The Node Details dialog appears.

   ![Node Details dialog](image)

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 70).

### Managing keyspaces and tables

The Data Explorer feature in OpsCenter has been removed and deprecated in favor of DataStax DevCenter, a visual CQL tool. Find more information about DevCenter, see DevCenter.

#### Keyspaces

Click **Data** in the left pane to list the keyspaces in the cluster that you are monitoring. View, edit, and delete keyspaces. Adding a keyspace in OpsCenter is no longer supported as of version 5.2. Add new keyspaces in DataStax DevCenter instead. The Data Explorer feature in OpsCenter has been removed and deprecated in favor of DataStax DevCenter, a visual CQL tool. Find more information about DevCenter, see DevCenter.

### Managing a keyspace

To manage a keyspace:

1. Click **Data** in the left navigation pane.
   
   The list of Keyspaces appears.
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 dialog appears.

![Edit Keyspace dialog](image)

4. To delete a keyspace, click **Delete Keyspace**.

5. To view table properties, select a table under Tables. For more information, see **Managing tables** (page 221).

### 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.

To manage a table:

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.
Using OpsCenter

Warning: Removal of the data is irreversible.

Browsing data

Note: The Data Explorer feature in OpsCenter has been removed and deprecated in favor of DataStax DevCenter, a visual CQL tool. Find more information about DevCenter and a link to download at http://www.datastax.com/what-we-offer/products-services/devcenter.

Cluster administration

OpsCenter manages multiple DataStax Enterprise clusters with a single install of the central opscenterd server. The OpsCenter Compatibility chart (page ) 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 407) 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.

If commitlog archiving is enabled on a cluster, OpsCenter does not automatically enable commitlog archiving on the new nodes when adding new nodes (page 401) to a cluster using Lifecycle Manager. If you add nodes to a cluster and commitlog archiving is enabled, you must manually copy commitlog_archiving.properties to the new nodes prior to starting DataStax Enterprise, or re-enable commit log archiving.

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 you are not using the default JMX or Native Transport ports (page 62), enter the appropriate port numbers.

5. If required, click **Add Credentials** and enter the username and password for JMX or Native Transport ports.

6. If applicable to your environment, select **DSE security (kerberos) is enabled on my cluster** and complete the following 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 client principal for the OpsCenter process/machine to use. Example: `opscenterd@YOUR_REALM`.

   c. Enter the location of the keytab OpsCenter machine, which contains credentials for the `opscenter_client_principal`.

   d. Enter the client principal for the DataStax Agent process/machine to use. Example: `agent@YOUR_REALM`.

   e. Enter the location of the keytab on the DataStax Agent machines, which contains credentials for the `agent_client_principal`. Example: `/path/to/keytab.keytab`.

   For more information, see **Authenticating with Kerberos (page )** and the **Kerberos tutorial (page )** in the DataStax Enterprise documentation.

7. **Note:** For more information, see **Enabling client-to-node encryption in OpsCenter (page 80)**

   Select **Client-to-node encryption is enabled on my cluster**. Enter the following for OpsCenter to use to directly connect to Cassandra:

   - If the paths are the same for the OpsCenter Keystore and Truststore (default), select **My Keystore and Truststore are the same** and enter the path and password.

   - If the paths are not the same, enter the **OpsCenter Keystore Path**, **Truststore Path**, and the corresponding **Password** for each.

8. Click **Next**.
The Set Up Agents dialog appears.

9. **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 57).

The Install Agents - Credentials dialog appears.

10. Enter SSH credentials to connect to your nodes:

   a. Enter a **Username**. The user must have root or sudo privileges.

   b. A sudo password can be entered in the **Password** box.

   c. Enter a **Private Key**.

No information entered in the Install Agents - Credentials dialog is saved or stored.
11. Click **Install**. The **Agents tab (page 201)** opens and displays the progress of the agent installation.

### Removing a cluster from OpsCenter and Lifecycle Manager monitoring

Remove a cluster from monitoring within the OpsCenter UI and Lifecycle Manager UI. Disconnecting a cluster from OpsCenter does not delete the cluster itself.

1. From the OpsCenter monitoring main menu, click **Settings#Cluster Connections**.
   The **Edit Cluster Connections Settings** dialog appears.

2. Select the cluster you want to delete from the **Cluster** list.

3. Click **Disconnect Cluster**.
   A prompt requests you to confirm the delete.
4. Click **Disconnect** to confirm removing the cluster from monitoring within the OpsCenter UI. The cluster and its datacenters and nodes are also removed from monitoring 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 230) to notify you when a cluster requires rebalancing. If using role-based security (page 92), 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.
- 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]

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.
Using OpsCenter

Related information:
Restarting DSE on a node [Restart the DataStax Enterprise service on any node.] (page 217)

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 391) 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.

![Edit Cluster Display Name dialog](image)

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**.

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.

### 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.
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 352)*. 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:
Using OpsCenter

- 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 352).

### 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 239](#)) 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**.

![Graph Legend](image)

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.

![Add Widget](image)

**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 236) 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.

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 69):

```plaintext
[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.
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**
The live disk space used by all tables on a node.

**Heap Used**
Average amount of Java heap memory used.

**JVM CMS Collection Count**
Number of concurrent mark sweep garbage collections performed per second.

**JVM CMS Collection Time**
Average number of milliseconds spent performing CMS garbage collections per second.

**JVM G1 Old Collection Count**
Number of G1 old generation garbage collections performed per second.

**JVM G1 Old Collection Time**
Average number of milliseconds spent performing G1 old generation garbage collections per second.

**JVM G1 Young Collection Count**
Number of G1 young generation garbage collections performed per second.

**JVM G1 Young Collection Time**
Average number of milliseconds spent performing G1 young generation garbage collections per second.

**JVM ParNew 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**
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**
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)**
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**
Number of bytes compacted per second.

**Total Compactions**
Number of sstable scans per second that could result in a compaction.

**Write Requests**
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**
Repair tasks pending, such as handling the merkle tree transfer after the validation compaction.

**TP: Gossip Tasks Pending**
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**
Number of pending tasks from system methods that modified the schema.

**TP: Misc. Tasks Pending**
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 243) for a backup in queued operations.

Table metrics are prefaced with TBL.

**TBL: Local Writes**
Local write requests per second. Local writes update the table's memtable and appends to a commitlog.

**TBL: Local Write Latency (percentiles)**
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**
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)**
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)**
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**
Disk space used by live SSTables. There might be obsolete SSTables not included.

**TBL: Total Disk Used**
Disk space used by a table by SSTables, including obsolete ones waiting to be garbage collected.

**TBL: SSTables per Read (percentiles)**
Using OpsCenter

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)**
The min, median, max, 90th, and 99th percentile of how many cells exist in partitions for this table.

**TBL: Partition Size (percentiles)**
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**
The total size of all the SSTables' bloom filters for this table.

**TBL: Bloom Filter False Positives**
Number of bloom filter false positives per second.

**TBL: Bloom Filter False Positive Ratio**
Percentage of bloom filter lookups that resulted in a false positive.

**TBL: Bloom Filter Off Heap**
Total off heap memory used by bloom filters from all live SSTables in a table.

**TBL: Index Summary Off Heap**
Total off heap memory used by the index summary of all live SSTables in a table.

**TBL: Compression Metadata Off Heap**
Total off heap memory used by the compression metadata of all live SSTables in a table.

**TBL: Memtable Off Heap**
Off heap memory used by a table's current memtable.

**KeyCache Requests**
The number of key cache requests per second.

**KeyCache Hits**
The number of key cache hits per second. This will avoid possible disk seeks when finding a partition in an SSTable.

**KeyCache Hit Rate**
The percentage of key cache lookups that resulted in a hit.

**RowCache Requests**
The number of row cache requests per second.

**RowCache Hits**
The number of row cache hits per second.

**RowCache Hit Rate**
The percentage of row cache lookups that resulted in a hit.

### 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](page 275).

**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**
Number of SSTables in a tier for a table.

**TIER: Total Disk Used**
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 275])
- **TIER: Total Disk Used** ([page 275])
- **TIER: SSTables** ([page 275])

**Prerequisites:**

- For centralized configuration convenience, configure the strategy and tiers ([page 440]) in Lifecycle Manager and run a configuration job ([page 413]) 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 COMPACTION={
      'class': 'TieredCompactionStrategy',
      'tiering_strategy': 'DateTieredStorageStrategy',
      'config': 'strategy1',
      'max_tier_ages': '3600,7200';
  }
  ```

See [DSE Tiered Storage](page 275) for complete details.

1. Click **Cluster#Dashboard**.

2. **Clone** ([page 240]) 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.
Repeat these steps for each tier metric and the corresponding nodes, tables, or tiers that you need to monitor.

Search performance metrics

Metrics for monitoring DSE Search (page 250) performance include:

**Search: Requests**
Requests per second made to a specific Solr core/index.

**Search: Request Latency**
Average time a search query takes in a DSE cluster using DSE Search.

**Search: Errors**
Errors per second that occur for a specific Solr core/index.

**Search: Timeouts**
Timeouts per second on a specific Solr core/index.

**Search: Core Size**
Size of the Solr core on disk.

Graph metrics

Reference of available graph metrics in OpsCenter monitoring for DSE Graph (page 250). 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 251):** Cache metrics bound to a specific DSE graph. Per node per graph instance metrics.

• **Request Latencies (page 251):** Histogram of request latencies.

• **Threadpool metrics (page 251):** 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 (page )* in the DSE Graph documentation.

**Graph: Adjacency Cache Hits**
Number of hits against the adjacency cache for this graph.

**Graph: Adjacency Cache Misses**
Number of misses against the adjacency cache for this graph.

**Graph: Index Cache Hits**
Number of hits against the index cache for this graph.

**Graph: Index Cache Misses**
Number of misses against the index cache for this graph.

Graph transaction latencies

**Note:** The Graph Request Latencies metric is only supported in OpsCenter versions 6.0.10, 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**
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 Query Threads Pending**
Number of pending tasks in the GraphQueryThreads thread pool.

**Graph TP: Graph Query Threads Active**
Number of active tasks in the GraphQueryThreads thread pool.

**Graph TP: Graph Query Threads Completed**
Number of tasks completed by the GraphQueryThreads thread pool.

**Graph TP: Graph Scheduled Threads Pending**
Number of pending tasks in the GraphScheduledThreads thread pool.

**Graph TP: Graph Scheduled Threads Active**
Number of active tasks in the GraphScheduledThreads thread pool.
Using OpsCenter

**Graph TP: Graph Scheduled Threads Completed**
Number of tasks completed by the GraphScheduledThreads thread pool.

**Graph TP: Graph System Threads Pending**
Number of pending tasks in the GraphSystemThreads thread pool.

**Graph TP: Graph System Threads Active**
Number of active tasks in the GraphSystemThreads thread pool.

**Graph TP: Graph System Threads Completed**
Number of tasks completed by the GraphSystemThreads thread pool.

**Graph TP: Gremlin Worker Threads Pending**
Number of pending tasks in the GremlinWorkerThreads thread pool.

**Graph TP: Gremlin Worker Threads Active**
Number of active tasks in the GremlinWorkerThreads thread pool.

**Graph TP: Gremlin Worker Threads Completed**
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 251).*

After adding metrics and clicking Save Graph, the graphs appear on the monitoring dashboard.

**Tip:** Clone *(page 240)* 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.
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.
Using OpsCenter

- 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 195). For even more awareness and visibility, see adding an alert for down nodes (page 209) 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>
### Metric Definition

<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 processflushes 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>
<tr>
<td>Metric</td>
<td>Definition</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<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
Using OpsCenter

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.

![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. Niced 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 commitlog partition is running low, the commitlog 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.
Using OpsCenter

The following list of metric descriptions available in tooltips is provided for your convenience:

**Write Requests**
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)**
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**
The number of write requests on the coordinator nodes that fail due to errors returned from replicas.

**Write Timeouts**
The number of server write timeouts per second on the coordinator nodes.

**Write Unavailable Errors**
The number of write requests per second on the coordinator nodes, that fail because not enough replicas are available.
Read Requests
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)
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
The number of read requests on the coordinator nodes that fail due to errors returned from replicas.

Read Timeouts
The number of server read timeouts per second on the coordinator nodes.

Read Unavailable Errors
The number of read requests per second on the coordinator nodes, that fail because not enough replicas are available.

Non Heap Committed
Allocated memory, guaranteed for Java nonheap.

Non Heap Max
Maximum amount that the Java nonheap can grow.

Non Heap Used
Average amount of Java nonheap memory used.

Heap Committed
Allocated memory guaranteed for the Java heap.

Heap Max
Maximum amount that the Java heap can grow.

Heap Used
Average amount of Java heap memory used.

JVM CMS Collection Count
Number of concurrent mark sweep garbage collections performed per second.

JVM ParNew 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
Average number of milliseconds spent performing CMS garbage collections per second.

JVM ParNew 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
Number of G1 old generation garbage collections performed per second.

JVM G1 Old Collection Time
Average number of milliseconds spent performing G1 old generation garbage collections per second.

JVM G1 Young Collection Count
Number of G1 young generation garbage collections performed per second.

JVM G1 Young Collection Time
Using OpsCenter

Average number of milliseconds spent performing G1 young generation garbage collections per second.

Data Size
The live disk space used by all tables on a node.

Total Bytes Compacted
Number of bytes compacted per second.

Total Compactions Completed
Number of compaction tasks completed per second.

Total Compactions
Number of sstable scans per second that could result in a compaction.

Compactions Pending
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
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.

TP: Dropped Tasks
Aggregate of different messages that might be thrown away.

TP: 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.

TP: 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.

TP: 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.

TP: 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.

TP: Dropped Paged Range Reads
A local paged read request was received after the timeout (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.

TP: Dropped Request Responses
A response to a request was received after the timeout (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.

TP: 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: Flashes Pending**
Number of memtables queued for the flush process. A flush sorts and writes the memtables to disk.

**TP: Gossip Tasks Pending**
Number of gossip messages and acknowledgments queued and waiting to be sent or received.

**TP: Internal Responses Pending**
Number of pending tasks from internal tasks, such as nodes joining and leaving the cluster.

**TP: Manual Repair Tasks Pending**
Repair tasks pending, such as handling the merkle tree transfer after the validation compaction.

**TP: Cache Cleaning Pending**
Tasks pending to clean row caches during a cleanup compaction.

**TP: Post Flashes Pending**
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**
Number of pending tasks from system methods that modified the schema.

**TP: Misc. Tasks Pending**
Number of pending tasks from infrequently run operations, such as taking a snapshot or processing the notification of a completed replication.

**TP: Read Requests Pending**
Number of pending read requests. Read requests read data off of disk and deserialize cached data.

**TP: Read Repair Tasks Pending**
Number of read repair operations in the queue waiting to run.

**TP: Request Responses Pending**
Number of pending callbacks to execute after a task on a remote node completes.

**TP: Write Requests Pending**
Number of write requests received by the cluster and waiting to be handled.

**TP: Validation Executor Pending**
Pending task to read data from sstables and generate a merkle tree for a repair.

**TP: Compaction Executor Pending**
Pending compactions that are known. This may deviate from "pending compactions" which includes an estimate of tasks that these pending tasks may create after completion.

**TP: Pending Range Calculator Pending**
Pending tasks to calculate the ranges according to bootstrapping and leaving nodes.

**TP: Native Transport Requests Pending**
Native Transport Requests Requests Pending

**TP: Flashes Active**
Up to memtable_flush_writers concurrent tasks to flush and write the memtables to disk.

**TP: Gossip Tasks Active**
Number of gossip messages and acknowledgments actively being sent or received.

**TP: Internal Responses Active**
Number of active tasks from internal tasks, such as nodes joining and leaving the cluster.

**TP: Manual Repair Tasks Active**
Repair tasks active, such as handling the merkle tree transfer after the validation compaction.

**TP: Cache Cleaning Active**
Tasks to clean row caches during a cleanup compaction.

**TP: Post Flushes Active**
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**
Number of active tasks from system methods that modified the schema.

**TP: Misc. Tasks Active**
Number of active tasks from infrequently run operations, such as taking a snapshot or processing the notification of a completed replication.

**TP: Read Requests Active**
Number of active read requests. Read requests read data off of disk and deserialize cached data.

**TP: Read Repair Tasks Active**
Number of read repair operations actively being run.

**TP: Request Responses Active**
Number of callbacks to being executed after a task on a remote node is completed.

**TP: Write Requests Active**
Number of write requests being handled.

**TP: Validation Executor Active**
Active task to read data from sstables and generate a merkle tree for a repair.

**TP: Compaction Executor Active**
Active compactions that are known.

**TP: Pending Range Calculator Active**
Active tasks to calculate the ranges according to bootstrapping and leaving nodes.

**TP: Native Transport Requests Active**
Native Transport Requests Requests Active

**TP: Flushes Completed**
Number of memtables flushed to disk since the nodes start.

**TP: Gossip Tasks Completed**
Number of gossip messages and acknowledgments recently sent or received.

**TP: Internal Responses Completed**
Number of recently completed tasks from internal tasks, such as nodes joining and leaving the cluster.

**TP: Manual Repair Tasks Completed**
Repair tasks recently completed, such as handling the merkle tree transfer after the validation compaction.
**TP: Cache Cleaning Completed**  
Tasks to clean row caches during a cleanup compaction.

**TP: Post Flushes Completed**  
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**  
Number of completed tasks from system methods that modified the schema.

**TP: Misc. Tasks Completed**  
Number of completed tasks from infrequently run operations, such as taking a snapshot or processing the notification of a completed replication.

**TP: Read Requests Completed**  
Number of completed read requests. Read requests read data off of disk and deserialize cached data.

**TP: Read Repair Tasks Completed**  
Number of read repair operations recently completed.

**TP: Request Responses Completed**  
Number of completed callbacks executed after a task on a remote node is completed.

**TP: Write Requests Completed**  
Number of write requests received by the cluster that have been handled.

**TP: Validation Executor Completed**  
Completed tasks to read data from sstables and generate a merkle tree for a repair.

**TP: Compaction Executor Completed**  
Completed compactions.

**TP: Pending Range Calculator Completed**  
Completed tasks to calculate the ranges according to bootstrapping and leaving nodes.

**TP: Native Transport Requests Completed**  
Native Transport Requests Requests Completed

**TP: Native Transport Requests Blocked**  
Native Transport Requests Requests Blocked

**TP: Total Native Transport Requests Blocked**  
Total Native Transport Requests Requests Blocked

**KeyCache Hits**  
The number of key cache hits per second. This will avoid possible disk seeks when finding a partition in an SSTable.

**KeyCache Requests**  
The number of key cache requests per second.

**KeyCache Hit Rate**  
The percentage of key cache lookups that resulted in a hit.

**RowCache Hits**  
The number of row cache hits per second.

**RowCache Requests**  
The number of row cache requests per second.

**RowCache Hit Rate**  
The percentage of row cache lookups that resulted in a hit.

**Native Clients**
The number of clients connected using the native protocol.

**Thrift Clients**
The number of clients connected via thrift.

**Read Repairs 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**
Corresponds to a digest mismatch that occurred after a completed read, outside of the client read loop.

**Synchronous Read Repairs**
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**
Local write requests per second. Local writes update the table's memtable and appends to a commitlog.

**TBL: Local Write Latency (percentiles)**
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**
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)**
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**
Disk space used by live SSTables. There might be obsolete SSTables not included.

**TBL: Total Disk Used**
Disk space used by a table by SSTables, including obsolete ones waiting to be garbage collected.

**TBL: SSTable Count**
Total number of SSTables for a table.

**TBL: SSTables per Read (percentiles)**
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)**
The min, median, max, 90th, and 99th percentile of the size (in bytes) of partitions of this table.

**TBL: Cell Count (percentiles)**
The min, median, max, 90th, and 99th percentile of how many cells exist in partitions for this table.

**TBL: Bloom Filter Space Used**
The total size of all the SSTables' bloom filters for this table.

**TBL: Bloom Filter False Positives**
Number of bloom filter false positives per second.

**TBL: Bloom Filter False Positive Ratio**
Percentage of bloom filter lookups that resulted in a false positive.

**Search: Requests**
Requests per second made to a specific Solr core/index.

**Search: Request Latency**
Average time a search query takes in a DSE cluster using DSE Search.

**Search: Errors**
Errors per second that occur for a specific Solr core/index.

**Search: Timeouts**
Timeouts per second on a specific Solr core/index.

**Search: Core Size**
Size of the Solr core on disk.

**OS: Memory (stacked)**
Stacked graph of used, cached, and free memory.

**OS: Memory (stacked)**
Stacked graph of used and free memory.

**OS: Memory Free**
Total system memory currently free.

**OS: Memory Used**
Total system memory currently used.

**OS: Memory Shared**
Total amount of memory in shared memory space.

**OS: Memory Buffered**
Total system memory currently buffered.

**OS: Memory Cached**
Total system memory currently cached.

**OS: Memory (stacked)**
Stacked graph of committed, cached, paged, non-paged, and free memory.

**OS: Memory Available**
Available physical memory.

**OS: Memory Committed**
Memory in use by the operating system.

**OS: Pool Paged Resident Memory**
Allocated pool-paged-resident memory.

**OS: Pool Nonpaged Memory**
Allocated pool-nonpaged memory.

**OS: System Cache Resident Memory**
Memory used by the file cache.

**OS: CPU (stacked)**
Stacked graph of iowait, steal, nice, system, user, and idle CPU usage.

**OS: CPU (stacked)**
Stacked graph of idle, user, and system CPU usage.

**OS: CPU (stacked)**
Stacked graph of user, privileged, and idle CPU usage.

**OS: CPU User**
Time the CPU devotes to user processes.

**OS: CPU System**
Time the CPU devotes to system processes.

**OS: CPU Idle**
Using OpsCenter

- **Time the CPU is idle.**
- **OS: CPU Iowait**
  - Time the CPU devotes to waiting for I/O to complete.
- **OS: CPU Steal**
  - Time the CPU devotes to tasks stolen by virtual operating systems.
- **OS: CPU Nice**
  - Time the CPU devotes to processing nice tasks.
- **OS: CPU Privileged**
  - Time the CPU devotes to processing privileged instructions.
- **OS: Load**
  - Operating system load average.
- **OS: Disk Usage (%)**
  - Disk space used by Cassandra at a given time.
- **OS: Disk Free**
  - Free space on a specific disk partition.
- **OS: Disk Used**
  - Disk space used by Cassandra at a given time.
- **OS: Disk Read Throughput**
  - Average disk throughput for read operations.
- **OS: Disk Write Throughput**
  - Average disk throughput for write operations.
- **OS: Disk Throughput**
  - Average disk throughput for read and write operations.
- **OS: Disk Read Rate**
  - Rate of reads per second to the disk.
- **OS: Disk Writes Rate**
  - Rate of writes per second to the disk.
- **OS: Disk Latency**
  - Average completion time of each request to the disk.
- **OS: Disk Request Size**
  - Average size of read requests issued to the disk.
- **OS: Disk Request Size**
  - Average size of read requests issued to the disk.
- **OS: Disk Queue Size**
  - Average number of requests queued due to disk latency issues.
- **OS: Disk Utilization**
  - CPU time consumed by disk I/O.
- **OS: Net Received**
  - Speed of data received from the network.
- **OS: Net Sent**
  - Speed of data sent across the network.
- **OS: Net Sent**
  - Speed of data sent across the network.
- **OS: Net Received**
  - Speed of data received from the network.
- **Speculative Retries**
  - Number of speculative retries for all column families.
- **TBL: Speculative Retries**
Number of speculative retries for this table.

**Stream Data Out - Total**
Data streamed out from this node to all other nodes, for all tables.

**Stream Data In - Total**
Data streams in to this node from all other nodes, for all tables.

**TBL: Bloom Filter Off Heap**
Total off heap memory used by bloom filters from all live SSTables in a table.

**TBL: Index Summary Off Heap**
Total off heap memory used by the index summary of all live SSTables in a table.

**TBL: Compression Metadata Off Heap**
Total off heap memory used by the compression metadata of all live SSTables in a table.

**TP: Counter Mutations Pending**
Pending tasks to execute local counter mutations.

**TP: Counter Mutations Active**
Up to concurrent_counter_writes running tasks that execute local counter mutations.

**TP: Counter Mutations Completed**
Number of local counter mutations that have been executed.

**TP: Memtable Reclams Pending**
Waits for current reads to complete and then frees the memory formerly used by the obsoleted memtables.

**TP: Memtable Reclams Active**
Waits for current reads to complete and then frees the memory formerly used by the obsoleted memtables.

**TP: Memtable Reclams Completed**
Waits for current reads to complete and then frees the memory formerly used by the obsoleted memtables.

**TBL: Memtable Off Heap**
Off heap memory used by a table's current memtable.

**TBL: Total Memtable Heap Size**
An estimate of the space used in JVM heap memory for all memtables. This includes ones that are currently being flushed and related secondary indexies.

**TBL: Total Memtable Live Data Size**
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 indexies.

**TBL: Total Memtable Off-Heap Size**
An estimate of the space used in off-heap memory for all memtables. This includes ones that are currently being flushed and related secondary indexies.

**In-Memory Percent Used**
The percentage of memory allocated for in-memory tables currently in use.

**TBL: Partition Count**
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**
**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**

**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)**

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**

Number of view mutations sent to replicas that have been acknowledged.

**View Write Pending**

Number of view mutations sent to replicas where the replicas acknowledgement hasn't been received.

**TP: View Mutations Pending**

Number of mutations to apply locally after modifications to a base table.

**TP: View Mutations Active**

Number of mutations to being applied locally after modifications to a base table.

**TP: View Mutations Completed**

Number of mutations applied locally after modifications to a base table.

**TP: Hint Dispatcher Pending**

Pending tasks to send the stored hinted handoffs to a host.

**TP: Hint Dispatcher Active**

Up to max_hints_delivery_threads tasks, each dispatching all hinted handoffs to a host.

**TP: Hint Dispatcher Completed**

Number of tasks to transfer hints to a host that have completed.

**TP: Index Management Pending**

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**

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**

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)**

The min, median, max, 90th, and 99th percentile of how many tombstones are read during a read.

**TBL: Local Write Latency**

**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
  **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)
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
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)
The min, median, max, 90th, and 99th percentile of how many cells were scanned during a read.

TBL: Cells Scanned (percentiles)
The min, median, max, 90th, and 99th percentile of how many cells were scanned during a read.

TIER: Total Disk Used
Disk space used by a table by SSTables for the tier.

TIER: sstables
Number of SSTables in a tier for a table.

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
Number of hits against the adjacency cache for this graph.

Graph: Adjacency Cache Misses
Number of misses against the adjacency cache for this graph.

Graph: Index Cache Hits
Number of hits against the index cache for this graph.

Graph: Index Cache Misses
Number of misses against the index cache for this graph.

Graph: Request Latencies
The min, median, max, 90th, and 99th percentile of request latencies during the period.

Graph TP: Graph Query Threads Pending
Number of pending tasks in the GraphQueryThreads thread pool.

Graph TP: Graph Query Threads Active
Number of active tasks in the GraphQueryThreads thread pool.

Graph TP: Graph Query Threads Completed
Number of tasks completed by the GraphQueryThreads thread pool.

Graph TP: Graph Scheduled Threads Pending
Number of pending tasks in the GraphScheduledThreads thread pool.

Graph TP: Graph Scheduled Threads Active
Number of active tasks in the GraphScheduledThreads thread pool.

**Graph TP: Graph Scheduled Threads Completed**
Number of tasks completed by the GraphScheduledThreads thread pool.

**Graph TP: Graph System Threads Pending**
Number of pending tasks in the GraphSystemThreads thread pool.

**Graph TP: Graph System Threads Active**
Number of active tasks in the GraphSystemThreads thread pool.

**Graph TP: Graph System Threads Completed**
Number of tasks completed by the GraphSystemThreads thread pool.

**Graph TP: Gremlin Worker Threads Pending**
Number of pending tasks in the GremlinWorkerThreads thread pool.

**Graph TP: Gremlin Worker Threads Active**
Number of active tasks in the GremlinWorkerThreads thread pool.

**Graph TP: Gremlin Worker Threads Completed**
Number of tasks completed by the GremlinWorkerThreads thread pool.

**Node Messaging 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**
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.

**SSTables Repaired**
The percentage of bytes that have been repaired. Calculated as ratio of the uncompressed size of repaired sstables to the total size of all sstables.

**TBL: SSTables Repaired**
The percentage of bytes that have been repaired for this table. Calculated as ratio of the uncompressed size of repaired sstables to the total size of all sstables.
DSE Management Services

Backup Service

The OpsCenter Backup Service allows you to create automatic or manual backups of your cluster data, from all the keyspaces in a cluster to specific keyspaces. You can perform both local and remote backups, point-in-time restores, and restoring to a different cluster (or "cloning" a cluster). Backup data is stored locally on each node, and optionally in cloud-based storage services like Amazon S3.

Backup Service Overview

Using OpsCenter, you can 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 449) 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, Amazon S3, or a custom location on the local filesystem
- Compresses backup files to save storage
- Allows specifying retention policies on 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 database clusters (e.g., copy a production cluster to a development cluster)
- Provides detailed backup and restore reports

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, and you can specify additional locations in cloud backup services like Amazon S3 where the snapshot data will be copied. Backups can be taken per keyspace, for selected multiple keyspaces, or for all keyspaces in the cluster while the system is online.

If your cluster includes DSE Search or DSE Analytics nodes, a backup job that includes keyspaces with DSE Search data or the cfs keyspace for Analytics nodes will save the Search and Analytics data. Any Solr indexes will be recreated on 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, this 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 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, or manually run one-off backups on a scheduled or ad hoc basis.

There must be enough free disk space on the node to accommodate making snapshots of your data files. A single snapshot requires little disk space. However, snapshots will cause your disk usage to grow more quickly over time because a snapshot prevents obsolete data files from being deleted. You can specify how long the snapshot data should be retained 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.

In addition to keyspace backups, commitlog backups are also available in the backup service to allow point-in-time restores for finer-grained control of the backed up data. Point-in-time restores are available when you enable commitlog backups in conjunction with keyspace backups. Like keyspace backups, the commitlogs will be retained based on a configurable retention policy.

**Note:** Point-in-time restores are only supported if the cluster topology is unchanged since the time you want to restore.

**Backing up to Amazon S3**

When you add an S3 bucket as an additional location for storing backup snapshots, the agent will send the snapshot files to the S3 bucket automatically. All SSTables for a particular node and table will only be 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:

```plaintext
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
```
The `backup.json` file contains metadata about which of the backed up SSTables are included in that backup.

If OpsCenter encounters an error when backing up to S3, it will retry the backup a user-configurable number of times (3 by default) unless it encounters an unrecoverable error such as invalid AWS credentials.

The AWS credentials and bucket names are stored in `cluster_name.conf`. Be sure to use proper security precautions to ensure that this file isn't readable by unauthorized users.

**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 backup files for 30 days. For each backup task, you can set a configurable time period in 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 will retain 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 user configured a scheduled backup that sends the data to S3, runs every week, and has a retention policy of removing backups older than 3 days. The layout in the S3 bucket is this:

```
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. 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.

Commitlog backups

Commitlog backups allow you to perform point-in-time restores, where you can specify a particular date and time from which to restore the data. Commitlog backups are configured (page 281) separately from snapshot backups.

custom_name.conf

The location of the custom_name.conf (page ) file depends on the type of installation:

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

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 Cassandra, proper backup and restore procedures are still very important to implement for production clusters.

While replication in Cassandra 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 the cluster’s data is 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 so you can ensure that the disaster recovery plan you deploy actually works as you intend.

**Configuring the Backup Service**

**Enabling commitlog backups**

Commitlog backups facilitate restoring backup data to a particular point in time *(page 303)*.

Commitlog backups are available for versions 4.6 or later of DataStax Enterprise.

**Note:**

Enabling commitlog archiving modifies the Cassandra commitlog_archiving.properties configuration file.

If commitlog archiving is enabled on a cluster, OpsCenter does not automatically enable commitlog archiving on the new nodes when adding new nodes *(page 401)* to a cluster using Lifecycle Manager. If you add nodes to a cluster and commitlog archiving is enabled, you must manually copy commitlog_archiving.properties to the new nodes prior to starting DataStax Enterprise, or re-enable commit log archiving.

**Prerequisites:**

- You must use Java 7 or later when backing up commitlogs.

1. Click *cluster name*#Services.
2. Click the **Configure** link for the Backup Service.
3. Click the **Settings** tab.

![Backup Service Settings](image)
4. Click the **Configure** link for Commitlog Backup.

   The Configure Commitlog Backup dialog appears.

   ![Configure Commitlog Backup dialog](image)

5. Set the slider to **On**.

6. Enter the path where you want to store the commitlogs 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. Starting in OpsCenter 5.1, the agent user and the DataStax Enterprise user are by default the same user.

   If the location runs out of disk space, the backup will fail. Configure the **free disk space threshold** (page 283) to prevent backups from starting if there is not enough free space available. Commitlog files **record every mutation of the data in a cluster** (page ), and can grow quite large over time. The disk performance of the commitlog backup location is extremely important because the disk write speed will likely be a bottleneck for write-heavy use cases, and read performance a bottleneck for restore operations.

7. If you want to backup the commitlogs to a cloud storage provider such as Amazon S3, click **Add Location**.

   **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.

a. In the Add Location window, select the location type under Location and indicate how long the snapshot data should be saved by selecting a Retention Policy.

To save the commitlogs to Amazon S3, select Amazon S3, set the bucket name, and your AWS key and secret. You can set a maximum upload rate to avoid saturating your network by selecting Throttle S3 transfer rate and setting the maximum MB per second to upload the data. To compress the data, select Enable compression. Compressing data reduces the amount of data going through your network, but increases the CPU load for the server.

The Retention Policy can be set to Retain All, which saves the commitlogs indefinitely, or to a set period of time. Once the commitlogs are older than the time set in Retention Policy, the commitlogs will be deleted.

b. Click Save Location.

8. Click Save.

9. After enabling or disabling commitlog backups, OpsCenter prompts you to perform a rolling restart (page 230) of your cluster.

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.

In addition to configuring the free disk space threshold for the backup service, set up alerts for free and used disk space. Add a graph to your dashboard to visually monitor free disk space.

• 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 in a dashboard graph view: Click Add Graph#Add Metric#OS: Disk Free.
1. Click `cluster name#Services`.

2. Click the Configure link for the Backup Service.

3. Click the Settings tab.

4. Click the Configure link for Disk Space Threshold.

   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**.

**Configuring encryption key storage for backups**

Configure whether OpsCenter stores the encryption keys for each node along with the SSTables. When tables in your cluster use the DataStax Enterprise transparent data encryption (page [link](#)), encryption keys from each node are stored in remote locations alongside the data. Encryption key storage is enabled by default and highly recommended.

1. Click **cluster name**\#Services.

2. Click the **Configure** link for the Backup Service.

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 your 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/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 the agent to upload very large files to Amazon S3**

Change the settings on the agent to allow uploading very large files to Amazon S3. The default settings for the DataStax agent prevent it from uploading SSTables that are over a certain size to S3. This limitation is in place to prevent the agent from using too much memory and will be lifted in future versions. The default maximum SSTable size is approximately 150 GB. Increase the maximum SSTable size allowed for uploading by modifying some agent properties on each node. These properties are configured in the `datastax-agent-env.sh` file on each node. The defaults in `datastax-agent-env.sh` are:

```
JVM_OPTS="$JVM_OPTS -Xmx128M -Djclouds.mpu.parts.magnitude=100000 -Djclouds.mpu.parts.size=16777216"
```

To increase the maximum SSTable size that the agent can upload, modify these properties:

```
-Xmx128M
-Djclouds.mpu.parts.size=16777216
```

- The `-Xmx` setting controls the heap size of the agent.
- The `-Djclouds` setting controls the chunk size for files when uploading to S3.

Since S3 supports multipart file uploads with a maximum number of 10,000 parts, the chunk size controls how large a file the agent can upload. Increasing the chunk size...
requires using more memory on the agent, so increasing the agent heap size is also required.

Here are example settings that allow loading 250 GB SSTables:

```
-Xmx256M
-Djclouds.mpu.parts.size=32000000
```

These settings increase the chunk size to 32MB and the heap size to 256MB, which allows for uploading the larger SSTable sizes.

**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`:
   ```yaml
   restore_on_transfer_failure: true
   ```

2. **Restart (page 70)** 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. Commitlog backups allow you to restore the data to a particular date and time.

**Note**: Keep the following caveats in mind 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 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 308).

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 Configure link for the Backup Service.

3. In the Activity tab, click Create Backup.

The Create Backup dialog appears.
4. Select the backup parameters:

   a. Under **Type**, click **Schedule**.

   b. **Schedule**: Select a date, time, and frequency for the backup. GMT is the default timezone. To change the timezone, click **GMT**, select the country and timezone, and click **Save**.

   c. **Keyspaces**: Click the plus (+) icon to select one or more keyspaces to back up. To keep the list open when making multiple selections, press and hold the Cmd key (Mac) or Ctrl key (Windows/Linux). Select **All Keyspaces** to back up all keyspaces.

   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.

   d. **Location**: Snapshots are saved to the node's snapshot directory for the table being saved. For example, `/var/lib/cassandra/data/OpsCenter/settings/snapshots`. To edit the Location and Retention Policy, click the edit icon in the Location pane. Or, to copy the snapshot to an additional location, click **Add Location**. For more information, see **adding a backup location (page 293)**.

   e. Change the staging directory if necessary by setting the `backup_staging_directory` configuration option in `address.yaml` (**page 176**).

   f. To set custom pre- or post-backup scripts, click **Advanced Options**:

      **Pre-Backup Script**: Enter the name of a script to run before the backup starts.

      **Post-Backup Script**: Enter the name of a script to run after the backup is done.

      For details, see **Configuring custom scripts to run before and after backups (page 286)**.

5. Click **Create Backup**.

   The scheduled backup appears in 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 Configure link for the Backup Service.

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 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 Configure link for the Backup Service.

3. In the Scheduled Backups tab, click Create New.
   The Create Backup dialog appears.

4. In the Schedule section, select the Don't Repeat option for the frequency.

5. Make your other selections and 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 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 Configure link for the Backup Service.

3. Click Create Backup.
   The Create Backup dialog appears.
4. For Type, click Run Now.

5. **Keypaces**: Click the plus (+) icon to select one or more keyspaces to back up. To keep the list open when making multiple selections, press and hold the Cmd key (Mac) or Ctrl key (Windows/Linux). Select **All Keypaces** to back up all keyspaces.

6. **Location**: Snapshots are saved to the node’s snapshot directory for the table being saved. For example, /var/lib/cassandra/data/OpsCenter/settings/snapshots. To edit the Location and Retention Policy, click the edit icon in the Location pane. Or, to copy the snapshot to an additional location, click **Add Location**. For more information, see adding a backup location (page 293).

7. Change the staging directory if necessary by setting the backup_staging_directory configuration option in address.yaml (page 176).

8. To set custom pre- or post-backup scripts, click **Advanced Options**:
   
   a. **Pre-Backup Script**: Enter the name of a script to run before the backup starts.
   
   b. **Post-Backup Script**: Enter the name of a script to run after the backup is done.
      
      For details, see Configuring custom scripts to run before and after backups (page 286).

9. 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.

**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 will 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 will mount 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 290) 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.

**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 290)*) Backup dialog:
   a. Click *cluster name*#Services.
   b. Click the Configure 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 in the Path box.

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. Indicate how long the snapshot data should be saved by selecting a **Retention Policy** *(page 279)*. **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.

7. Click **Save Location**.

   The newly added Local FS displays in the Location pane of the Create or Edit Backup dialog.

   ![Location Settings](image)

   Click the edit icon to edit a location. Click the delete icon to delete a location.

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 278)*.

**Prerequisites:**

- Java 7 or later is required to store at an S3 location.
- Make sure you have the proper AWS IAM privileges.

**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 290)*) Backup dialog:

   a. Click **cluster name**#Services.
b. Click the **Configure** 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 **Amazon S3** in Location.

4. Enter the **S3 Bucket** name.

   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.

5. Enter your **AWS key** and **AWS Secret**.

6. Indicate how long the snapshot data should be saved by selecting a **Retention Policy** (page 279). **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.

7. Select any throttling, compressing, or encryption 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.

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 [http://docs.aws.amazon.com/AmazonS3/latest/dev/UsingServerSideEncryption.html](http://docs.aws.amazon.com/AmazonS3/latest/dev/UsingServerSideEncryption.html).

8. Click **Save Location**.

   The newly added S3 location displays in the Location pane of the Create or Edit Backup dialog.

   Click the edit icon to edit a location. Click the delete icon to delete a location.

9. Click **Save Backup**, or **Create Backup** as applicable.

### Restoring a cluster

You can 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 you enabled commitlog backups.

When performing a restore operation, you can restore all the keyspaces from a backup or select specific keyspaces and tables.

**Note:** Keep the following caveats in mind 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 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 308).
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.

1. Click *cluster name* #Services.

2. 
3. Click the **Configure** link for the Backup Service.

4. In the Activity tab, click **Restore Backup**.
   
The Restore from Backup dialog appears.

5. 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 have enabled commitlog backups ([page 281](#)), you can restore from a particular point-in-time by selecting the **Point In Time** tab. For more details, see **Restoring a backup to a specific point-in-time** ([page 303](#)).

   c. If you are restoring from an S3 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 314](#)), but can be used when this OpsCenter instance is not aware of the backup location.

      Enter the name of the bucket under **S3 Bucket**, then the AWS key and secret.
6. 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:** Keep the following caveats in mind 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 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 308).

![Restore from Backup](image)

7. Under **Location**, select the target cluster for the restored data.

- The Location list is only available when 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 will be cloned to the selected cluster.
Note:
Restoring encrypted tables to a different cluster will not work unless the encryption keys are identical, which is typically not the case.

8. 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.

9. 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.

10. Change the staging directory if necessary by setting the `backup_staging_directory` configuration option in `address.yaml (page 176)`.

11. Click **Restore Backup**.

12. Click **Start Restore** to confirm when prompted.

   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.
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.

After the restore starts, the **Restore Report** dialog 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.

**Note:** If you are restoring (essentially cloning) from an S3 backup and you close the Restore Report dialog, you must reopen the Restore Report from the destination cluster.
If there is insufficient disk space to restore the backup, the restore fails. See Monitoring sufficient disk space for restoring backups (page 308).

Restoring a backup to a specific point-in-time

For a point-in-time restore, OpsCenter intelligently chooses which snapshots and commitlogs to restore from based on the date and time you are restoring the cluster to. If an acceptable combination of snapshots and commitlogs cannot be found, the restore will fail and 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 commitlog backups (page 281) and performed at least one snapshot backup (page 287) 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.

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 recommends performing a snapshot backup before changing cluster topology. You can then restore the cluster based on that backup.

**Known limitations:**

- Point-in-time restore cannot restore commitlogs 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.

1. Click *cluster name*\#Services.

2. Click the Configure 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.
5. Complete your selections in the step 1 dialog:

   a. **Point in Time**: Set the date and time to which you want to restore your data.

   b. **Commitlogs Location**: Select the location of the commitlogs, either On Server or another location on S3. The location of commitlogs is configured when enabling commitlog backups (*page 281*).

   c. **Backup Location**: Select the location of the snapshot, either On Server or another location on S3. The backup location was specified when creating the backup.

   d. Click **Next**.

     The Restore from Backup, Step 2 of 2: Configure and Restore dialog appears.
6. Complete your selections in the step 2 dialog:

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**: Keep the following caveats in mind 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 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 308).

b. Location: Select the target cluster for the restored data.
   - The Location list is only available when 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 will be cloned to the selected cluster.

   **Note:**
   Restoring encrypted tables to a different cluster will 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.yaml` (page 176).

f. Click the **Edit Restart Settings** link to adjust settings for the rolling restart (page 230).

7. Click **Restore Backup**.

8. 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 commitlogs, followed by a rolling commitlog replay across the cluster. Each node is configured for replay and restarted after the previous one finishes successfully.

If an error occurs during a point-in-time restore for a subset of tables, you might need to manually the 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:
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 181) 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 (page ) 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.
Plentiful free disk space | Low free disk space
---|---

<table>
<thead>
<tr>
<th>Storage Capacity</th>
<th>Storage Capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Used: 5 GB Free: 26 GB Total: 31 GB 1 of 1 nodes</td>
<td>Used: 320 GB Free: 8 GB Total: 328 GB 4 of 4 nodes</td>
</tr>
</tbody>
</table>

**Tip:** Keep abreast of capacity requirements using the **Capacity Service** *(page 334)*. See **Forecasting** *(page 335)*.

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

Cloning data from a DSE cluster using the Backup Service is a simple way to copy the data from one cluster or location to another cluster. Cloning cluster data is useful for development
purposes, such as updating applications using cloned production data. Cloning cluster data is also useful for testing or performance tuning.

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 285) 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.

**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 293).

  **Note:** Cloning encrypted tables (page 285) 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*.
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 311).
   • Local FS: Go to 7 (page 311).

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:** Keep the following caveats in mind 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 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 308).

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, 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_outboundMegabitsPerSec** and **inter_dc_stream_throughput_outboundMegabitsPerSec**), or the restore will not be throttled. Previous versions of Cassandra would limit these streams to no more than 200 Mbps. Unthrottled restores may consume large amounts of network resources. Plan accordingly.
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 428)* for the cluster and run a configuration job *(page 413)*. 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 293)*.

**Note:** Cloning encrypted tables *(page 285)* 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.

4. In the Backups tab, select the backup that contains the data you want to clone and click Next.

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:** Keep the following caveats in mind 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 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 308*).

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 176)`.

10. Click **Restore Backup**.

11. Click **Start Restore** to confirm when prompted.

   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**.

## 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 318) 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 13: 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 319), 2 (page 319)</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 309))</td>
<td>1 (page 319), 2 (page 319), 7 (page 321)</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 319), 2 (page 319), 6 (page 320)</td>
</tr>
<tr>
<td>OpsCenter versions</td>
<td>DSE versions</td>
<td>Operations</td>
<td>Steps Sequence</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------------------</td>
<td>------------------------------------------------------</td>
<td>----------------</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 319), 1 (page 319), 2 (page 319), 5 (page 320), 6 (page 320)</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 319), 1 (page 319), 2 (page 319), 5 (page 320), 6 (page 320), 7 (page 321)</td>
</tr>
</tbody>
</table>

1. In OpsCenter (version 6.5 highly recommended), create a backup (page 287) 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 297) and Cloning cluster data (page 309).

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(), 0x1f8b0800000000000000abe65250504a2e4a4d2c494d5152b0523034b1b0b0343036b030d13335353600031d909ae292c492d2629012259fccb25425ae5a00e2fb48b33b000000, 'json', now()) ;
```

Insert a new record in the dse_system.shared_data table of the destination cluster with values obtained from the source cluster: select * from dse_system.shared_data from the original source DSE cluster. Get the values for dataspace, valid_until, namespace, and Name fields from the source DSE Graph cluster, and replace the values into the corresponding statement values above for the destination cluster.

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 (page 505).
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.

The details of all completed backup and restore operations are stored in the OpsCenter keyspace in Cassandra in the **backup_reports** table. The data is stored whether or not the operation was successful.

The first time the Backup Service starts, it scans for existing backups, including backups from versions of OpsCenter prior to 5.1, and populates the **backup_reports** table.

1. Click **cluster name**Services.

2. Click the Configure link for the Backup Service.

3. To manually synchronize the **backup_reports** table, click Actions#Synchronize Data.
4. Select the locations whose history you want to synchronize and click **Sync**.

![Synchronize Cassandra Log](image)

**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 **Configure** link for the Backup Service.

3. In the Activity tab, click the backup for which you want to delete data. The Backup Report dialog appears.

![Backup Report](image)

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.

**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 commitlogs, 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](https://docs.aws.amazon.com/IAM/latest/UserGuide/security-iam-managed-policies.html).

**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 60) to read and modify the files owned by the DataStax Enterprise user.

Repair Service

The Repair Service is configured to run continuously and perform repair operations across a DataStax Enterprise cluster in a minimally impactful way. The Repair Service runs in the background, constantly repairing small chunks of a cluster to alleviate the pressure and potential performance impact of having to periodically run repair on entire nodes. When the entire cluster has been repaired, the Repair Service recalculates the list of subranges to repair and starts over. Repairing the entire cluster one time is a cycle.

Note: DataStax Enterprise 4.5 or later is required.

How the Repair Service works

The Repair Service works by repairing small chunks of a cluster in the background. The Repair Service incrementally and cyclically repairs a DataStax Enterprise cluster within the specified completion time. This overview describes the Repair Service behavior and its response to changes in cluster topology or schemas.

The Repair Service uses an average of the throughput of recent repairs to calculate how many parallel repairs OpsCenter can complete in the current cycle. Before issuing a new subrange repair, the Repair Service checks for the number of repairs. If the configured maximum pending repairs (page 162) 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 and an alert is fired.

Parameters

- The time_to_completion parameter is the maximum amount of time it takes to repair the entire cluster once. Typically, you set this value lower than your lowest gc_grace_seconds setting. The default for gc_grace_seconds is 10 days. The service might run multiple repairs in parallel, but runs as few as needed to complete within the amount of time specified. The service always avoids running more than one repair within a single replica set.

Estimating remaining repair time

If the Repair Service anticipates it cannot complete a repair within the allotted time to completion due to throughput errors, it displays a warning message and a newly estimated time remaining to complete the repair. 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 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 123) or post-url alert notifications (page 125) are configured, the alert notifications are emailed or posted.

The error_logging_window (page 162) configuration property controls how often to fire the alert if the Repair Service continues to estimate that it will not finish the repair in time.

Known limitations

If a cluster is datacenter-aware and has any keyspaces using SimpleStrategy, the Repair Service will fail to start. Follow the prompts (page 327) to change the keyspaces to use NetworkTopologyStrategy.

Cluster topology changes

If a change in cluster topology occurs, the Repair Service stops its current cycle and waits for the ring to stabilize before starting a new cycle. The check for topology changes occurs every five minutes. Topology changes include:

- Nodes moving within a cluster
- Nodes joining a cluster
- Nodes leaving a cluster
- Nodes being decommissioned

Schema changes

- Keyspaces added while the repair service is running are repaired when the next subrange repair is started.
- Tables added to existing keyspaces are repaired immediately during the current cycle of the Repair Service.
- Keyspaces or tables can be removed while the Repair Service is running without causing any issues.

Down nodes

When a node or multiple nodes are down, the Repair Service continues to run repairs for ranges and keyspaces unaffected by the down nodes. If there are enough nodes down to make repairing the cluster not possible during the down time, the Repair Service waits up to three hours before restarting a cycle (after repairing as much as possible under the circumstances).

Restarting opscenterd

The current state of the Repair Service is persisted locally on the opscenterd server every five minutes by default. If opscenterd is restarted, the Repair Service resumes where it left off.

**Note:** If automatic failover (page 141) is configured, be sure to mirror the repair service persist directory (page 147).
For more information on repair service continuity during a failure, see failover aftereffects (page 142).

**Incremental repairs**

OpsCenter performs an incremental repair on a user-configured set of tables every time a subrange repair on the mutually exclusive set of other tables is run on a given node. A user-configurable option (`incremental_range_repair`) controls whether to repair just the subrange or the entire range of the node. The default is to repair the entire range of the node.

**Note:** If a cluster is multi-datacenter and there is a keyspace that only exists in one datacenter, it might be a while between incremental repairs in that datacenter because the Repair Service currently repairs an entire datacenter at a time.

After manually migrating a table to use incremental repair, update the user-configured list of tables in the `incremental_repair_tables` configuration option. Any incorrectly formatted table logs an error. Read more about efficient incremental repairs in the DataStax Developer blog post. For information on migrating to incremental repairs in Cassandra, see migrating to incremental repairs (page ).

**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 repair**

The following is 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`.

- **[repair_service] incremental_repair_tables**
  
  The list of keyspaces and tables to include in incremental repairs. (e.g. Keyspace1.Standard1, Keyspace1.Standard2)

- **[repair_service] incremental_range_repair**
  
  Determines whether incremental repairs run as subrange repairs (True) or repair the node’s entire range (False). Default: False.

- **[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.

**Starting and stopping the Repair Service**

**Starting the Repair Service**

Configure the days for the repair cycle to complete and start the Repair Service for a cluster. The Repair Service takes a single parameter, `time_to_completion`, which is the maximum amount of time it takes to repair an entire cluster one cycle.

**Prerequisites:**

In a multi-datacenter environment, the NetworkTopologyStrategy is required as the replication strategy for all keyspaces to run the Repair Service. The Repair Service will not run with SimpleStrategy. Prompts guide you to change the strategy (page 328) from SimpleStrategy to NetworkTopologyStrategy when necessary.

To configure and start the Repair Service:

1. In the left navigation pane, click **Services**.
2. Click **Configure** for the Repair Service.

![cluster0: Services](image)

3. In the **Start Repair Service** dialog, enter a value for **Time to completion** field. The default is 9 days.

   Typically, you should set the **Time to Completion** to a value lower than the lowest `gc_grace_seconds` setting. The default for `gc_grace_seconds` is 10 days.
4. Click **Start Repair Service**.

The Repair Service starts up and updates its status after one minute and every minute thereafter. A progress bar displays the percentage complete for the currently running repair.

---

**Changing the OpsCenter keyspace**


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 221).*
The Edit Keyspace `keyspace_name` dialog for the keyspace appears. The dialog also appears if you attempt to start the Repair Service despite the warning.

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.
   
   Be careful 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**.

6. If there are multiple incompatible keyspaces, the **Error: Incompatible Keyspace** dialog appears after clicking Configure for the Repair Service.

7. Click the **Data section** link to jump directly to the Data section to edit the keyspaces.
8. Repeat the above steps as necessary until all keyspace warnings disappear.

**Stopping the Repair Service**

Stopping the Repair Service stops the service after any repairs in progress complete the currently running repair cycle.

1. In the left navigation pane, click **Services**.

2. Click **Configure** for the Repair Service.

   The **Stop Repair Service** dialog appears.
3. Click **Stop Repair Service**.
   
   Any repairs that were running when the repair service was stopped continue running until completion. Repairs in progress are not cancelled. The Status for the Repair Service on the Services page displays Off.

**Checking repair progress**

The Status column

The **Status** column in **Services#Repair Service** displays the progress of the current cycle as a percentage of subranges that have been fully repaired. The **Status** column updates every minute, or when you click on the **Services** section in the left navigation pane.

![Services page screenshot with Repair Service status](image)

**Logging**

All Repair Service activity is logged to by default to:

**Package installs**

/var/log/opscenter/repair_service/<cluster_name>.log

**Tarball installs**

<install_location>/log/repair_service/<cluster_name>.log

The log file is automatically rotated at ~9.5MB, keeping up to ten rotated logs by default. The Repair Service log options are configurable.

**Advanced Repair Service configuration**

The following is 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`.

[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
How often, in seconds, to write the state to the persistence file for the repair service. The default value is 300 (5 minutes).

[repair_service] restart_period
How often in seconds to restart repairs. The default value is 300 (5 minutes).

[repair_service] cluster_stabilization_period
How often in seconds repair service checks for cluster state before resuming.

[repair_service] ks_update_period
The maximum age, in seconds, of a cached version of the current keyspace schema. The default values is 300 (5 minutes).

[repair_service] single_task_err_threshold
The number of times to retry a repair task before moving on to the next task. The default value is 10.

[repair_service] max_err_threshold
The maximum number of times to fail on a repair before cancelling the repair attempt. Errors during incremental repair do not count towards this threshold. The default value is 100.

[repair_service] max_parallel_repairs
The maximum number of repairs to run in parallel. If unspecified or set to 0, the system attempts to calculate 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.

[repair_service] max_pending_repairs
The maximum pending repairs allowed to be running on a node at one time. The default value is 5.

[repair_service] alert_on_repair_failure
Whether there should be alerts fired when a repair task fails. Defaults to true.

[repair_service] single_repair_timeout
The maximum length of time for a repair to complete, in seconds. The default value is 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. The default value is 5.

[repair_service] min_throughput
The minimum throughput needed to calculate parallel repairs. The default value is 512.

[repair_service] num_recent_throughputs
The number of recent throughputs used to calculate the average throughput, which is then used to determine how many parallel repairs are needed. The default value is 20.

[repair_service] error_logging_window
The desired amount of time in seconds between errors for exceeding max_parallel_repairs. Defaults to 86400

[repair_service] repair_estimation_factor
Estimated reduced efficiency due to other issues such as concurrent compaction.
[repair_service] incremental_repair_tables
   The list of keyspaces and tables to include in incremental repairs. (e.g.
   Keyspace1.Standard1, Keyspace1.Standard2)
[repair_service] incremental_range_repair
   Determines whether incremental repairs run as subrange repairs (True) or repair
   the node's entire range (False). Default: False.
[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.
[repair_service] snapshot_override
   Specifies whether to override the default snapshot repair behavior. The default
   value is False. Specifying this option as either True or False will always modify
   the behavior of the repair service. Snapshot repair can only be configured for
   Cassandra versions 2.0.11 and greater.

opscenterd.conf
The location of the opscenterd.conf (page 331) 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 (page 331) file depends on the type of installation:
   • Package installations: /etc/opscenter/clusters/cluster_name.conf
   • Tarball installations: install_location/conf/clusters/cluster_name.conf

Troubleshooting Repair Service errors

To resolve errors, try adjusting the configuration options (page 331) 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 range repair
   When a single range repair fails, the repair is skipped temporarily and added to
   the end of the queue of repairs and retried later. If a single range fails ten times
   (default), the Repair Service shuts down and fires an alert. Configure this setting
   with the single_task_err_threshold option.

Too many errors in a single run
   After a total of 100 errors (default) during a single run, the Repair Service shuts
   down and fires an ALERT. Configure this setting with the max_err_threshold
   option.

Time-outs
   The Repair Service times out a single repair command after one hour by default.
   This counts towards an error for that repair command and it is placed at the
   end of the queue of repairs and retried later. Configure this setting with the
   single_repair_timeout option.
Too many repairs in parallel
The Repair Service errors and shuts down if it has to run too many repairs in parallel. By default, this happens if it estimates that it needs to run more than one repair in a single replica set to complete on time. Configure this setting with the `max_parallel_repairs` option.

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. Configure this setting with the `max_pending_repairs` option.

Incremental error alert threshold exceeded
By default, the number of failed incremental repair attempts defaults to 20 before sending an alert that there may be a problem with incremental repair. Adjust this setting with the `incremental_err_alert_threshold` option.

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`

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 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 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 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 DataStax 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 dialog](image)

   The 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.

   ![Forecast dialog](image)

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 DataStax 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.

```ini
[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 69) the OpsCenter daemon.

**Best Practice Service**

The Best Practice service periodically scans your clusters to automatically detect issues that affect a cluster’s health. It includes a set of rules, organized into different categories, that are used to analyze the nodes in a cluster and report back any deviations from the best practice recommendations for the clusters. The report includes recommendations on how to alter the node or cluster to fix the problems.

By default, all 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.

You can configure which rules are used when scanning the cluster and how often the clusters will be scanned.

If a best practice rule fails, it sends an alert (page 256). Similar to other alerts, you can configure notification settings for these rules.

Click Best Practice service in the Services section to see the current number of enabled rules that have passed and failed. Click the numbers to filter the list of rules to only passing or failing rules.
Click Configure to manage rules and see more details about each rule.

**Configuring Best Practice service rules**

Best Practice service rules can be enabled, scheduled, and disabled from OpsCenter.

1. Click *cluster name*#Services.

2. Click the **Configure** link for the Best Practice Service.

3. Click the category name to show the rules for that category.

4. 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**.

5. To disable a rule, click **Turn Rule Off**.

6. To change the schedule for running a rule:
   - a. Click the **Configure** link for the rule.
   - b. Modify the date and time to start the rule or how often the rule should run.
   - c. Click **Save Rule**.

**Monitoring the results of Best Practice service scans**

If a rule has been enabled and a scan has completed, the status of the rule 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 total number of rules that have passed and 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 **Passed** or **Failed** to get more info on the rule. The window displays a description of the rule, the importance of the rule, and when the last scan was run. If there are failures, the window provides a detailed message about the rule failure, including which nodes failed the rule and how to correct the failure.

Hover over a rule to view its full description:

### 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>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>Tip:</strong> Use LCM Config Profiles (page 428) 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>
<td></td>
</tr>
<tr>
<td>Commit Log Archiving has been turned off due to inconsistent settings for all nodes in the cluster. <strong>Note:</strong> This rule is available in OpsCenter versions 6.1 and later.</td>
<td>High</td>
<td>Node, Cluster</td>
<td>Hourly</td>
<td>Alert</td>
<td></td>
</tr>
<tr>
<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. <strong>Turn Commit Log Archiving on (page 281) again so that all nodes in the cluster have the enabled setting consistent for Commit Log Archiving.</strong></td>
<td></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 (page 324).</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. It is recommended to enable the OpsCenter repair service to run within the smallest <code>gc_grace</code> window configured on your cluster.</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. Please enable SSL for communicating with agents (page 73).</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. Please disable swap space.</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. 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 401) in the appropriate LCM Config Profile (page 428).</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>
</table>
| Different Listen and RPC Addresses | Checks that if there are multiple network interfaces that Cassandra has been configured to use separate networks for listen and rpc address.  
**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.  
Multiple networks have been detected but you are using the same network for client and internal customer communication. | Medium     | Node   | Daily              | Info        |

## 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>
</table>
| OpsCenter Failover Enabled          | DataStax recommends configuring OpsCenter failover for high availability.  
There is no backup OpsCenter configured. Please enable failover ([page 141](#)) for OpsCenter.                                                                                                                                 | Low        | OpsC   | Daily              | Alert       |

## OS 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>Clocks in cluster out of sync</td>
<td>Checks that clocks across the cluster are in sync within a 2 second tolerance.</td>
<td>High</td>
<td>Node, 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>The total drift across cluster exceeds</td>
<td>exceeds the tolerance of 2 seconds; please sync clocks on your nodes.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cassandra-user and agent-user match</td>
<td>Checks 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></td>
<td>Cassandra and agent are not run as the same user. Please ensure that Cassandra and agent</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>are run as the same user.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Clocks in UTC</td>
<td>Checks that clocks across the nodes are in Coordinated Universal Time (UTC).</td>
<td>Low</td>
<td>Node</td>
<td>Daily</td>
<td>Alert</td>
</tr>
<tr>
<td></td>
<td>All the nodes are not in Coordinated Universal Time (UTC). Please ensure that all nodes</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>are in UTC.</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.</td>
<td>Medium</td>
<td>Node</td>
<td>Daily</td>
<td>Alert</td>
</tr>
<tr>
<td></td>
<td>Unsupported JDK is in use on the node. Oracle/Sun Hotspot JDK is the preferred JDK to use</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>and well-tested in DataStax Enterprise. Switch to Oracle Hotspot JDK if you're currently</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>using OpenJDK (as the default Java environment coming from the Linux OS).</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Tip:</strong> Use LCM Config Profiles to manage Java installations <em>(page 442)</em>.</td>
<td></td>
<td></td>
<td></td>
<td></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 352)*).
Tip: Use LCM Config Profiles (page 428) to adjust request timeout settings in `cassandra.yaml` settings and run a configuration job (page 410).

<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 <code>cassandra.yaml</code> on your nodes and lower the value of <code>read_request_timeout_in_ms</code> (page 209).</td>
<td>Medium</td>
<td>Node</td>
<td>Daily</td>
<td>Alert</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. Significantly increasing the write request timeout on your nodes is not recommended. Please update <code>cassandra.yaml</code> on your nodes and lower the value of <code>write_request_timeout_in_ms</code> (page 209).</td>
<td>Medium</td>
<td>Node</td>
<td>Daily</td>
<td>Alert</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. Significantly increasing the range request timeout on your nodes is not recommended. Please update <code>cassandra.yaml</code> on your nodes and lower the value of <code>range_request_timeout_in_ms</code> (page 209).</td>
<td>Medium</td>
<td>Node</td>
<td>Daily</td>
<td>Alert</td>
</tr>
</tbody>
</table>

Performance Service - Slow Queries Advisor

For more information, see Identifying and tuning slow queries (page 360) in the Performance Service (page 352).
<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>
</tbody>
</table>

Performance Service - Table Metrics Advisor

For more information, see Identifying poorly performing tables *(page 366)* in the Performance Service *(page 352)*.

<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>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.</td>
<td>Low</td>
<td>Node, Cluster</td>
<td>Hourly</td>
<td>Alert</td>
</tr>
<tr>
<td></td>
<td>Excessively wide partitions have a negative impact on performance and are not recommended. Consider remodeling your data to break up wide partitions.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Secondary indexes cardinality</td>
<td>Checks for secondary indexes with too many distinct values.</td>
<td>Low</td>
<td>Node, Cluster</td>
<td>Hourly</td>
<td>Alert</td>
</tr>
<tr>
<td></td>
<td>High-cardinality secondary indexes can have a negative impact on system performance. Consider denormalizing the indexed data.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tombstone count</td>
<td>Number of tombstones processed during reads.</td>
<td>Low</td>
<td>Node, Cluster</td>
<td>Hourly</td>
<td>Alert</td>
</tr>
<tr>
<td></td>
<td>Too many tombstones can cause a degradation of performance. This can even lead to query failures.</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>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 (page ) that best fits your data and environment.</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 Monitoring node thread pool statistics (page 372) in the Performance Service (page 352).

<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 (page 328).

<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>
<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 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 cassandra.yaml in the appropriate LCM Config Profile <em>(page 428).</em></td>
<td>Medium</td>
<td>Node</td>
<td>Daily</td>
<td>Alert</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 cassandra.yaml in the appropriate LCM Config Profile <em>(page 428).</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>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. Security superuser has default setting. Please update the password for the user ‘cassandra’.</td>
<td>High</td>
<td>Cluster</td>
<td>Daily</td>
<td>Alert</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.</td>
<td>Medium</td>
<td>Node</td>
<td>Daily</td>
<td>Alert</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.

Version availability and compatibility:
The OpsCenter Performance Service is available in OpsCenter versions 5.2 and later.
OpsCenter Performance Service is only compatible with DataStax Enterprise versions 4.6 and later.

**Note:** The DSE Performance Service *(page)* is available in DataStax Enterprise versions 4.5 and later.

### 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 360):** Identify slow queries on a cluster to easily find and tune poorly performing queries.
- **Table Metrics (page 366):** Displays metrics to discover and diagnose poorly performing tables.
- **Thread Pool Statistics (page 372):** 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 363)*
- Table metrics alert setup example *(page 369)*
- Thread pool statistics alert setup example *(page 376)*
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 337). 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 365)
- Performance Service - Table Stats Advisor (page 371)
- Performance Service - Thread Pools Advisor (page 377)

By default, all rules are enabled and scheduled to run every hour. Adjust the rule configuration (page 338) 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 (page ), 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 235) 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 337) and Alerts (page 118) 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.

**Prerequisites:** When the OpsCenter Performance Service is enabled, OpsCenter programmatically sets a value in dse.yaml (page ), 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 for safekeeping.

**Note:** If a cluster is being managed within Lifecycle Manager, OpsCenter does not automatically modify dse.yaml for the Performance Service. You must manually update the associated settings in the configuration profile (page 428) for the cluster in Lifecycle Manager (page 387).

1. Click *cluster name*#Services.

2. Click the Configure link 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 buttons 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 from off to on pushes changes 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.

5. Click **Change** to proceed.

   No further configuration is required; however, you can adjust the default configuration parameters:
   - Configuring the slow query log (*page 360*)
   - Configuring table metrics (*page 366*)
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 `cluster name`#Services.

2. Click the `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.

**Prerequisites:** Enabling authentication in OpsCenter *(page 96)*

1. Click `Settings`#Users & Roles.
   The Users & Roles dialog appears.

2. Click `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 360), Configuring table metrics (page 366), and Configuring thread pool statistics (page 372).
   
   - **Performance Service CQL Tracing**: Enables the Trace feature (page 362) for the Slow Query Log.

5. Click **Save**. If applicable, repeat for each cluster in your environment.

**Related information:**
Configuring role-based security [OpsCenter allows enabling user authentication, adding users, and defining custom roles.] (page 92)

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 link 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 362*) requires permission (*page 358*) granted in your user role.

The **User** column displays the users who are running slow queries for OpsCenter versions 6.0.9 and later.

1. Click **cluster name**\#Services.

2. Click the **Configure** link for the Performance Service.
   
   The Overview tab displays the **performance panels** (*page 353*).

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 358) 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.
2. Click the Configure link for the Performance Service.
   The Overview tab displays the performance panels (page 353).

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 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 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 Consistency or 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 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 Configure 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>
</tbody>
</table>
### Recommendation | Description
--- | ---
Avoid using large batches | 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.
Minimize keys used within the IN clause | 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.
Use counter instead of count | A count(*) query can be expensive, even with smaller limits. Replace the logic with a counter you maintain.
Use prepared statements for your queries | Prepared statements reduce the workload on the coordinator by removing the overhead of parsing the query.

### 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** link 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 353).*

3. Click the title bar of the **Table Metrics** panel.
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.

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 Configure link for the Best Practice Service.
3. Click to expand the Performance Service - Table Metrics Advisor panel.

   ![Performance Service - Table Metrics Advisor](image)

4. Click the Configure link for the rule you want to adjust.

   ![Configure Rule: Wide partitions](image)

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. (page ) 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 338) 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>
<tr>
<td>Wide partitions</td>
<td>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.</td>
</tr>
</tbody>
</table>

**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** link 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 353).*

3. Click the title bar of the **Thread Pools** panel.
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.

![Thread Pools](image_url)

• 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 Configure link for the Best Practice Service.
3. Click to expand the **Performance Service - Table Metrics Advisor** panel.

![Performance Service - Thread Pools Advisor](image)

4. Click the **Configure** link for the rule you want to adjust.

![Configure Rule: Mutation Stage](image)

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 Stress</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

High CPU usage by opscenterd

Increasing the nodelist polling period or setting a sleep delay can reduce excessive CPU usage when starting or running opscenterd. In some environments, you might notice CPU usage for the opscenterd spiking dramatically (almost to 100%) upon startup or while it’s already running. Typically, this is caused by the retrieval and parsing of cluster topology performed during startup and every 60 seconds by default while opscenterd is running. When OpsCenter is managing multiple clusters with vnodes enabled, the impact of this CPU spike can cause performance issues or even stop opscenterd from starting up properly.

If your environment is experiencing excessive CPU consumption, try the available workarounds to alleviate the issue.

**Note:** Future versions of OpsCenter will provide a better solution to reduce the CPU usage of this process and should obviate the need for the interim CPU workarounds.

Configuring the polling period for CPU issues while running opscenterd

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`

Increasing the nodelist polling period can reduce CPU usage when running opscenterd. The `nodelist_poll_period` configuration option is applicable to OpsCenter version 5.0.2+

1. Open `cluster_name.conf` for editing.

2. Add a `[collection]` section and set the `nodelist_poll_period` value:

   ```
   [collection]
   nodelist_poll_period = 43200 # this would be every 12 hours
   ```

   The `nodelist_poll_period` represents the interval in seconds that OpsCenter polls the nodes and token lists in a cluster. Polling the node list determines whether there were any topology changes since the last poll. If you do not anticipate any topology changes, set it to a high value.

3. If there were any topology changes and the polling interval is set to a high value, restart `opscenterd`. Otherwise, wait for the next poll.
4. Refresh the browser.

Configuring a sleep delay for CPU issues when starting opscenterd

opscenterd.conf

The location of the opscenterd.conf (page 306) file depends on the type of installation:

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

Configuring a delay between clusters during startup helps alleviate opscenterd CPU usage on startup, allowing OpsCenter to function properly. The startup_sleep configuration option is applicable to OpsCenter version 5.1.1+. 

1. Open opscenterd.conf and set startup_sleep to the following:

```
[clusters]
startup_sleep = 5
```

The sleep value controls how long OpsCenter waits between connecting to clusters on startup. The default value is 0 seconds, which results in no staggered wait between connecting to each cluster. Depending on your environment, you might need to adjust the value accordingly. After configuring the sleep value to a value other than zero, wait until all clusters have started before using the UI or API. Otherwise, OpsCenter can become unresponsive and log multiple errors.

2. Restart opscenterd.

Troubleshooting SSL

Troubleshooting SSL connections to Cassandra

In OpsCenter 6.0, all OpsCenter SSL connections to DataStax Enterprise clusters are validated automatically. Since moving to the JVM, OpsCenter uses a keystore/truststore model to connect to Cassandra, and as such, requires stricter enforcement of SSL certificates than in earlier versions of OpsCenter. Therefore there is no longer an option to disable SSL certificate validation using the ssl_validate option in cluster settings.

To ensure that OpsCenter is set up to work with your client-to-node encryption-enabled cluster, follow these steps:

1. Create a truststore file on the OpsCenter machine using each Cassandra node's certificates.
a. Obtain the public key certificates from each Cassandra node you want to connect to OpsCenter.

b. Import these public key certificates into the truststore file. For example:

   keytool -import -v -trustcacerts -alias node0 -file node0.cer -keystore .truststore

2. Create a client certificate and keystore on the OpsCenter machine:

   keytool -genkey -alias opscenter -keystore keystore.jks
   keytool -export -alias opscenter -file opscenter.cer -keystore keystore.jks

3. If your DataStax Enterprise cluster is set up to require client auth
   (require_client_auth under client-to-node-encryption in cassandra.yaml) when
   using client-to-node encryption, you will need to import the public certificate generated in
   step 2 (page 381) into every node’s truststore so that Cassandra will trust connections
   coming from OpsCenter.

Troubleshooting Connections with OpsCenter to Cassandra with SSL

**OpsCenter shuts down due to invalid property in cluster config: ssl_validate**

This error occurs when your cluster config contains the deprecated ssl_validate
configuration value in your cluster_name.conf configuration file. Remove the ssl_validate
option and **Restart OpsCenter (page 69)**.

**OpsCenter cannot connect to the cluster with No Cassandra connection available error in logs**

While this error can be due to a number of issues with the cluster connection, when working
with SSL, it can be an indicator that the keystore/truststore setup needs tweaking. You might
see errors similar to these in the logs:

   2016-02-04 16:06:53,255 [] DEBUG: Node 127.0.0.1 seems to be down, trying
   next contact point (MainThread)
   2016-02-04 16:06:53,255 [] DEBUG: Unable to connect to any seed nodes,
   tried ['127.0.0.1'] (MainThread)
   2016-02-04 16:06:53,256 [] WARN: No cassandra connection available for
   hostlist ['127.0.0.1'] . Retrying. (MainThread)

These errors indicate that OpsCenter cannot make a connection with the DataStax
Enterprise cluster. To ensure that your SSL setup is correct, check the following:

- Certificates in keystore/truststore are valid (hostnames match and certificates are not
  expired).
- Ensure that all public certificates for each node are in the truststore for OpsCenter.
Troubleshooting

- Ensure that the certificate for OpsCenter is in each node’s truststore if `require_client_auth` is enabled.

**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`

### Troubleshooting SSL validation for self-signed certificates

**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`

Set `ssl_validate` to `False` if you use a self-signed certificate and experience difficulty connecting to a DataStax Enterprise cluster.

1. Open `cluster_name.conf` for editing.

   ```
   [cassandra]
   username =
   ssl_ca_certs = .cer file location
   ssl_validate = False
   ```

2. Restart OpsCenter (page 69).

### Data must be a byte string pyopenssl error

Update your version of openpyssl to version 0.15+ to resolve the "data must be a byte string error" encountered when setting up email or http alerts.

### Browser issues

#### Zero nodes detected in cluster or Loading OpsCenter screen hanging

Some environments might experience connectivity issues with the persistent connection between the browser and `opscenterd`. Symptoms of this issue include a blinking icon near the top right of the OpsCenter UI, and "0 nodes" appears as well. Another symptom is the Loading OpsCenter screen seems stuck and does not load OpsCenter.
Follow the workaround steps to resolve the issue:

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` and add the following:

   ```
   [labs]
   orbited_longpoll = true
   ```

2. Restart `opscenterd`.

3. Refresh the browser.

**Internet Explorer web browser not supported**

If you try to load the OpsCenter client in Microsoft Internet Explorer, a dialog indicates IE is not supported.

OpsCenter is only supported on the latest versions of:

- Apple Safari
- Google Chrome
- Mozilla Firefox

**Troubleshooting SSTables**

**Problems running sstableloader**

Running `sstableloader` results in broken data distribution

Running `sstableloader` on Cassandra versions prior to 1.2.12 and DataStax Enterprise 3.1.4 or earlier results in broken data distribution. For more information, see `CASSANDRA-6272`

We recommend upgrading to DataStax Enterprise 3.1.5 or higher, or Cassandra 1.2.12 or higher. Cassandra 2.0.x is unaffected.

Running `sstableloader` hangs on restore

Running `sstableloader` on DataStax Enterprise 4.0.4 when using `DSEDelegateSnitch` (the default snitch) or `DSESimpleSnitch` will cause the restore to hang. If you find that the OpsCenter restore status is stuck, run the following command on every node in your cluster:
Troubleshooting

The workaround is to modify the group permissions of /var/lib/cassandra/data to give the opscenter-agent group write access, or wait for an update to DataStax Enterprise 4.0.x.

**The SSTables in this snapshot '<tag>' are not compatible**

If you receive an error message that includes "The SSTables in this snapshot '<tag>' are not compatible with the current version of Cassandra", it means you must upgrade your snapshot to the current major version of DataStax Enterprise.

**How to**

1. Log in to each node.

2. Run the sstableupgrade script for every keyspace and table you need to restore; passing it the keyspace, table, and OpsCenter snapshot tag received from the error message.
   
   How you run the script depends on how you installed DataStax Enterprise.

3. Retry the restore from OpsCenter.

**Troubleshooting collecting or displaying metrics**

**Limiting the metrics collected by OpsCenter**

If a cluster keyspace has many tables, the number of metrics OpsCenter collects can become quite large. For information about reducing the number of keyspaces or tables that are monitored, see Controlling data collection (page 131).

**OpsCenter data growing too large**

A bug fixed in 3.2.1 was not setting a TTL (time to live) on metrics data being collected for a managed cluster. Depending on your environment, this could cause some tables in the OpsCenter keyspace to grow too large. The most common offenders typically are the pdps (raw data points) and rollups60 (1m data points) tables.
If any of the tables have grown too large, you can truncate (page 222) them to reclaim the space. If you are not comfortable losing historical data for that granularity (for example, 1m), please contact DataStax support.

**Cannot create a keyspace**

Due to a Python 2.6 or earlier bug, some users experience a problem adding a keyspace using Data Modeling OpsCenter features. OpsCenter cannot save a newly created keyspace. Upgrading Python generally fixes this problem.

**Note:** As of version 5.2, OpsCenter no longer supports adding a keyspace because the Data Explorer feature was removed. The Data Explorer feature in OpsCenter has been deprecated in favor of DataStax DevCenter, a visual CQL tool. Find more information about DevCenter and a link to download at [http://www.datastax.com/what-we-offer/products-services/devcenter](http://www.datastax.com/what-we-offer/products-services/devcenter).

**DataStax agent port setting conflict**

If there are problems with OpsCenter, check for conflicts in port settings (page 62). The DataStax Agent uses port 7199 by default. If you have not changed the default port, check that Cassandra or another process on the node is not set up to use port 7199.

If you set the DataStax Agent port to a host name instead of an IP address, the DNS provider must be online to resolve the host name. If the DNS provider is not online, expect some intermittent problems.

**Java not installed or JAVA_HOME environment variable not set**

If Java is not installed or if OpsCenter cannot find JAVA_HOME, you may see an error such as:

```
/usr/share/datastax-agent/bin/datastax-agent: line 98:exec: -X: invalid option
exec: usage: exec [-cl ] [-a name ] [ command [arguments ... ]] [redirection ... ]
```

To correct this problem, install Java or set JAVA_HOME: `export JAVA_HOME=<path_to_java>`

**Insufficient user resource limits errors**

Insufficient resource limits may result in an insufficient nofiles error:

```
2012-08-13 11:22:51-0400 [] INFO: Could not accept new connection (EMFILE )
```
Troubleshooting

See Recommended settings (page ) under Insufficient user resource limits errors in the Cassandra documentation.

Installing EPEL on CentOS 5.x or RHEL 5.x

Before installing OpsCenter on CentOS 5.x or RHEL 5.x, you must install EPEL (Extra Packages for Enterprise Linux).

To install for both 32- and 64-bit systems:

```
```

**Note:** You do not have to install EPEL on other machines.

Error getting version update information

If an OpsCenter node does not have internet access, or OpsCenter has difficulty with the URL for DataStax updates, the "Error getting version update information" message is displayed. The message simply indicates that notifications when new versions of OpsCenter or DataStax Enterprise are available will not be shown. As a temporary workaround to hiding the error message, you can disable the latest version check.

Add the following to `opscenterd.conf` and restart `opscenterd`:

```
[labs]
latest_version_check = False
```

OpsCenter cannot start up

OpsCenter uses the system temporary directory to load/execute dynamic class files. You may see an error message in the `startup.log` indicating that OpsCenter cannot read/write/execute from that directory. If permissions on the directory cannot be changed, you can use a JVM argument to change the temporary directory that OpsCenter uses. Please refer to the JVM Tuning (page 151) section for more details.

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`
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 415) 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 143) 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 multi-instance nodes?

OpsCenter supports monitoring multi-instance nodes.
Lifecycle Manager

**Note:** Lifecycle Manager does not currently support managing DataStax Enterprise Multi-Instance (page ) nodes (also referred to as dense nodes).

Which snitches does LCM support?

At this time, only the GossipingPropertyFileSnitch (GPFS) (page ) is supported for managing or importing DataStax Enterprise clusters in Lifecycle Manager.

What partitioners does LCM support?

- Murmur3Partitioner (page )
- RandomPartitioner (page )

What operating systems does LCM support?

Lifecycle Manager runs on OpsCenter-supported Linux environments only.

`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 143) 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 role-based security (page 92) for details enabling OpsCenter authentication.

  **Note:** Only OpsCenter users assigned an Admin role (page 92) 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 73) for details.

1. Access Lifecycle Manager:
When you first launch OpsCenter, the Welcome to DataStax OpsCenter dialog appears. Click **Create a new cluster**. Lifecycle Manager launches in another browser window.

Click **Lifecycle Manager** from the OpsCenter navigation menu. Lifecycle Manager launches in another browser window.

Within OpsCenter Lifecycle Manager, click **OpsCenter Monitoring** to return to monitoring DSE clusters in DataStax 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 DataStax 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:

To view a short video that demonstrates getting started, see **Setting up a multi-node cluster using OpsCenter Lifecycle Manager**.

The initial Lifecycle Manager workflow:

After creating SSH credentials *(page 391)*, defining configuration profiles *(2 (page 391))* , adding a repository *(3 (page 391))* , and building the cluster topology model *(4 (page 391))* , you are ready to run an install job to install and configure DataStax Enterprise (DSE), including monitoring the install job *(5 (page 391))* .

**Prerequisites:**

Bring your own instances on a **supported platform** *(page)*.

Follow the linked workflow steps to quickly get started working with Lifecycle Manager:
1. **Add SSH credentials** *(page 426)* so LCM can remotely log in to target machines when performing installation and configuration activities.

2. **Add a configuration profile** *(page 434)* 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 422)* 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 391)* of the cluster:
   a. **Add a cluster** *(page 394)*.
   b. **Add its datacenters** *(page 398)*.
   c. **Add its nodes** *(page 401)*.

5. **Run an installation job** *(page 411)* 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 415)*.

**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 437)* and **run a configuration job** *(page 413)* 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 407) 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 143) 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 394)**
2. **Add datacenters (page 398)**
3. **Add nodes (page 401)**

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:
Table 14: Topology status legend

<table>
<thead>
<tr>
<th>Status</th>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not run</td>
<td>(red flag)</td>
<td>An install job has not been run on the topology entity. See <a href="#">running a job</a> (page 410).</td>
</tr>
<tr>
<td>Import (unmanaged cluster)</td>
<td>(red plus sign)</td>
<td>An existing cluster is not being managed by Lifecycle Manager. Click <strong>Start Managing</strong> and follow instructions to automatically import (page 407) the cluster.</td>
</tr>
<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 417). 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 143) 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 434)
- Add a Repository (page 422)
- Add SSH Credentials (page 426)

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 390) 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 398) to the cluster.

3. Add nodes (page 401) to the datacenters.

4. Run an installation job (page 411). 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 232).

4. Click **Save**.

**What's next:** Depending on the edit, run an install (page 411) or configure (page 413) 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 434](#))
- Add SSH credentials ([page 426](#))
- Add a cluster ([page 394](#))

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 15: 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 428) to apply at the datacenter level to override inheriting the configuration profile from the cluster level. |
   | SSH Credentials   | Select an SSH credential (page 425) 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)  
                    |   • Hadoop (DSE Hadoop, not BYOH)  
                    |   • Solr  
                    |   • Spark  
                    |   • Spark + Solr  
   | DSE Graph (DSE 5.0+ only) | Select the DSE Graph option if the datacenter is for a DSE Graph (page ) 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 401) to the datacenters.

3. Run an installation job (page 411).

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 411)* or configure *(page 413)* 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.
Note: Lifecycle Manager does not currently support managing DataStax Enterprise Multi-Instance (page ) nodes (also referred to as dense nodes).

Prerequisites:

- **Add a cluster (page 394).**
- **Add a datacenter (page 398).**
- If you need a Configuration Profile specialized at the node level, define a node-specific config profile (page 434).
- If you need an SSH Credential to specify access and privileges at the node level, add an SSH credential (page 426) 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 421) are inherited from the cluster.

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 16: 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 rack1. 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 SSH credential (<a href="#">page 425</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 configuration file (<a href="#">page 428</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>
<tr>
<td>Seed Node</td>
<td>Select a seed node option:</td>
</tr>
<tr>
<td></td>
<td>- <strong>Automatically choose</strong>: (Default) Allow Lifecycle Manager to select a seed node. LCM designates two nodes as seeds per datacenter.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Make this a seed node</strong>: Select to designate the node as a seed node used by gossip for bootstrapping new nodes joining a cluster.</td>
</tr>
<tr>
<td></td>
<td><strong>Warning</strong>: 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.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Do not make this a seed node</strong>: Select to explicitly exclude the node from seed node designation.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: 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).</td>
</tr>
</tbody>
</table>
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 411).

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 411) or configuration (page 413) 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 391) 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 (page  ) 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 426).
- 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 409).
- Only the GossipingPropertyFileSnitch (GPFS) (page ) 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 (page ) and rpc_address (page ) 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 (page ) 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 425)* 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 409).*

5. Select acknowledgement of the warnings.
6. Click **Submit**.

The import cluster job begins processing. Follow the progress in the **Jobs (page 415)** workspace. If unsuccessful, a dialog provides a link to review the job details.

**What’s next:**

- Create the necessary configuration profiles (page 434) and associate them with the cluster, datacenter, or nodes in the **Clusters (page 391)** workspace.
- Add a repository (page 422) and associate it with the cluster in the **Clusters (page 391)** workspace for running future install jobs (page 410).

**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 407) of the cluster topology.

1. **Add SSH credentials (page 426)** so LCM can remotely log in to target machines when performing installation and configuration activities.

2. **Add a configuration profile (page 434)** 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 422)** 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 391)** of the cluster:
   - a. **Add a cluster (page 394).**
   - b. **Add its datacenters (page 398).**
   - c. **Add its nodes (page 401).**

5. **Run an installation job (page 411)** 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 415).

**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 314)

Run an Install and Configure Jobs Overview

Jobs are launched from the Clusters (page 391) workspace of Lifecycle Manager. Monitor install, configure, and import jobs in the Jobs (page 415) workspace of Lifecycle Manager.

Lifecycle Manager jobs 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 407) 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.

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 415)

**Running an installation job**

Submit a DataStax Enterprise install job to run on a cluster, datacenter, or node. An install job includes running a configuration job (page 413) and by default, installing Java and the JCE Policy files (page 442) required to enable unlimited strength encryption.

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:**

- All credentials (SSH (page 425) and repositories (page 421)) must be created, configuration profiles (page 428) defined, and a cluster topology model (page 391) built or imported (page 407) before running any install jobs in Lifecycle Manager.
- LCM does not create data directories or manage their permissions on your behalf. If you want to use a custom data directory, please ensure that it exists and is owned by the cassandra user.
- 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.

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.
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**: Sets the option depending on actions within the job. When adding a cluster or datacenter, sets **auto_bootstrap** to False. When adding nodes to an existing datacenter, sets **auto_bootstrap** to True.
     
     **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 415)** 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 437)** 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 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:**

- All credentials ([SSH (page 425)](#)) and [repositories (page 421)](#) must be created, [configuration profiles (page 428)](#) defined, and a [cluster topology model (page 391)](#) built or imported (page 407) before running any install jobs in Lifecycle Manager.
- Run an installation job (page 411) for the initial first-time installation.

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 415)** 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.

**Downloading the generated CA cert**

Download the CA certificate automatically generated by Lifecycle Manager after enabling client-to-node encryption. Lifecycle Manager automates the process of preparing server certificates (page 433) using an internal certificate authority (page 434). Configure your CQL clients to trust certificates signed by the certificate authority.

**Prerequisites**: Enable client-to-node encryption (page 433) in the configuration profile (page 428) associated with the cluster.

1. In the Clusters workspace of Lifecycle Manager, select the cluster in the Clusters pane.
The Cluster Details for the cluster appears.

2. In the Cluster Details pane, click the **Download Cert** link for CA Certificate. The browser downloads the certificate file.

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 (page 410) as an example.

   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 410)

**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 17: Job status legend

<table>
<thead>
<tr>
<th>Status</th>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not Run</td>
<td>⬤</td>
<td>The initial install job has not been run. This indicator displays in the Clusters workspace.</td>
</tr>
<tr>
<td>Pending</td>
<td>🌶</td>
<td>The job is in the queue waiting to run.</td>
</tr>
<tr>
<td>Running</td>
<td>⚙</td>
<td>The job is currently running.</td>
</tr>
<tr>
<td>Success (Completed)</td>
<td>✅</td>
<td>The job ran successfully.</td>
</tr>
<tr>
<td>Failure</td>
<td>⚔</td>
<td>The job failed. Investigate the issue by drilling into the job details. Try running the job again.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><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>⚠</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 manually abort (page 419) the job.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><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 [lifecycle_manager] section of opscenterd.conf. For details, see configuring idle timeout thresholds for jobs (page 420).</td>
</tr>
<tr>
<td>Status</td>
<td>Icon</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Aborted</td>
<td>🔄</td>
<td>The job was manually aborted (<a href="#">page 419</a>). Aborted jobs appear in logs with a TERMINATED status.</td>
</tr>
</tbody>
</table>

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 143](#)) 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 143](#)) to mirror the lcm.db.

**opscenterd.conf**

The location of the opscenterd.conf ([page](#)) 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 410](#))

**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.

   ![Lifecycle Manager: Jobs](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. The default value is 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. The default value is 3.

   `[lifecycle_manager] ssh_retry_delay_in_seconds`
The number of seconds to wait between SSH connection retries during an LCM job. The default value is 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:

   - **[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 69).

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 445) 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 391) 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 143) 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 143) 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 143) 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 143) to mirror the `lcm.db`.

**Prerequisites:** Optionally configure a proxy for package downloads (page 445) 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:
     
     # Enter the **Repository URL**.
     # 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.
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 445).

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 396)* in the Clusters *(page 391)* 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.
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 143) to mirror the lcm.db.

If there are any SSH connection issues when running a job, increase the SSH connection settings (page 420) 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 143) 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 143) 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 143) 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 396), datacenter (page 400), or node (page 405) as appropriate to select the credential in the Clusters (page 391) 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 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 413) 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 config 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 391), 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 config profile defined and applied at the cluster level that specifies:

- use the G1 Garbage Collector (g1gc)
- use a max heap size of 16 G
- Does not explicitly specify a commitlog directory, instead relying on the default value of /var/lib/cassandra/commitlog.

The cluster has two datacenters; DC1 and DC2:

- DC1 has no config profile of its own and therefore inherits its config profile from its cluster.
- DC2 has a defined config profile at the DC level that specifies a maximum heap size of 32 G, and a commitlog 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 G max heap size; and uses the default commitlog directory of /var/lib/cassandra/commitlog commitlog directory.
- Nodes in DC2 inherit from the cluster and also override cluster settings with the datacenter-level config profile: g1gc (inherited from cluster); 32 G max heap size (DC config profile takes precedence over an explicit setting in a cluster-level config profile); /cassandra_data/commitlog commitlog directory (DC config profile takes precedence takes precedence over an implicit default inherited from a cluster-level config profile).

The inheritance and precedence of config profiles allows keeping a cluster consistent by inheriting as much as possible from a cluster-level config profile, while also providing the flexibility of specifying only the granular settings that differ in higher precedence within config profiles applied at the lower, more granular datacenter and node levels.

Configuration Profile Files

Each configuration profile is specific to a recent version of DataStax Enterprise (4.7 and later). A configuration profile is composed of multiple configuration files for configuring features of DataStax Enterprise clusters:
Configuration profiles allow customizing the following configuration files:

- **Cassandra section:**
  
  - # cassandra.yaml (page )
  - # dse.yaml (page )
  - # cassandra-env.sh: configure Cassandra environment settings for garbage collection; heap size
  - # logback.xml (page )

  **Note:** When you add a configuration profile, DSE authentication is enabled by default for all supported versions of DataStax Enterprise. DSE Authenticator is enabled for DSE version 5.0. For more information, see Managing DSE Security using LCM (page 432).

- **DSE Hadoop section:**
  
  - # hive-site.xml

- **Spark section:**
  
  - # dse-spark-env.sh
  - # logback-spark-executor.xml
  - # logback-spark-server.xml
  - # logback-spark.xml
  - # spark-defaults.conf
  - # spark-env.sh

- **Lifecycle Manager section:**
# Java Setup (page 442): Automatically manages JRE installs and JCE Policy files.

# Package Proxy (page 445): Accelerate package downloads or isolate DataStax Enterprise clusters offline from the internet.

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.

A configuration file explicitly not managed at this time by Lifecycle Manager is `commitlog_archiving.properties`, which is used for configuring commitlog archive and PIT restore for the Backup Service. This file is managed instead from within the Backup Service (page 277).

**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 143) 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 143) to mirror the `lcm.db`.

## Configuration Known Issues and Limitations

**Encrypted DSE configuration values**

Lifecycle Manager is not currently compatible with DSE Configuration Encryption (page ). If DSE configuration encryption is enabled, it must be disabled prior to importing a cluster (page 407) 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 (page
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 (page 433) by default:

The following links provide more information about the available security options in cassandra.yaml (page 435):

- authenticator (page 435)
- authorizer (page 435)
- role_manager (page 435)
- permissions_validity_in_ms (page 435)
- roles_validity_in_ms (page 435)
- roles_update_interval_in_ms (page 435)
- server_encryption_options (page 435) (internode_encryption)
- client_encryption_options (page 435)

Node-to-Node Encryption

Lifecycle Manager can configure DataStax Enterprise clusters to use node-to-node (page 435) encryption. The feature is disabled by default.

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 preparing server certificates (page 435) 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.
**Note:** 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.

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 437)* 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 410)*. 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.

**Client-To-Node Encryption**

Lifecycle Manager can configure DataStax Enterprise clusters to use **client-to-node encryption** *(page )*. The option is disabled by default.

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 432)*.

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 414).
• If certificates were generated outside of Lifecycle Manager, acquire the appropriate CA certificate or self-signed certificates.

**Note:** Enabling `require_client_auth` for `client_encryption_options` requires special steps due to a limitation in the current release. For more information, refer to the Knowledge Base Article.

**Internal Certificate Authority**

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.

**opscenterd.conf**

The location of the `opscenterd.conf` (page 306) 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 410) 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 143) 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 143) to mirror the lcm.db.

1. From the Lifecycle Manager navigation menu, click **Config Profiles**#Add Config Profile.

   The Config Profiles page appears.

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 (page ), 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 140) 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 442) for automatically installing and managing Java installs and upgrades, and for installing and managing the JCE Policy files (recommended).
- Configure a package proxy (page 445) 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 391) Workspace. Edit the cluster (page 396), datacenter (page 400), or node (page 405) and select the Config Profile.
- Run a configuration job (*page 413*) 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 410*) 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 (*page 447*).
• Run a configuration job *(page 413)* to apply the config profile changes.

**Configuring Lifecycle Manager for DSE Graph**

Setting up Lifecycle Manager for DSE Graph *(page 351)* 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 434)*:
   
   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** *(page 401)* in a Lifecycle Manager Config Profile. Override the defaults if necessary. For more information about configuration options, see **configuring DSE Graph options** *(page 216)* 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 431).

2. When adding a datacenter (page 398) to a cluster in the Clusters (page 391) workspace, select the DSE Graph (DSE 5.0+ Only) option in the Add Datacenter dialog:
Lifecycle Manager

Configuring tiered storage

DSE Tiered Storage (page 232) 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 410) 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:

```
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](#).

7. Click **Add an item** to define the paths to the storage tiers. The **Add tiers** dialog appears.

8. 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.

9. 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 (page 298) for complete details.
- Go to the Clusters (page 391) 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 413) to push the configuration to all of the applicable nodes.
- Add dashboard graphs to monitor data distribution (page 248) amongst storage tiers using the tiered storage metrics (page 247) available in OpsCenter.

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.

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)
```
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 434)* (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 411)**, 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 `/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. (page 447)
- Configure a package proxy (page 445) to cache and expedite package downloads.
- Run a configure job (page 413) 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 421)) 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 421).
- Check config profile usage (page 447).
Run a configure job (page 413) 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 (page 447) 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.

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: Config Profiles

**Last modified:** 2016-01-01 12:52 PM  
**Created:** 2016-01-01 12:52 PM

### Config Profile 123

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Data Center</th>
<th>Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster 123</td>
<td>Data Center 123</td>
<td>Node 1</td>
</tr>
<tr>
<td>Cluster 123</td>
<td>Data Center 123</td>
<td>Node 2</td>
</tr>
<tr>
<td>Cluster 123</td>
<td>Data Center 234</td>
<td></td>
</tr>
</tbody>
</table>
OpsCenter API reference for developers

The OpsCenter API (page 449) 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. The OpsCenter API includes the Lifecycle Manager API (page 449) for the LCM GUI.