# Contents

## OpsCenter Installation Guide

1. **About DataStax OpsCenter** ................................. 1  
   - Getting Started with OpsCenter ............................ 1  
   - OpsCenter Capacity Planning ............................... 2  

## Installing the OpsCenter Dashboard ......................... 3  

1. **OpsCenter Installation Prerequisites** .................... 3  
2. **Installing OpsCenter Debian/Ubuntu Packages** .......... 4  
3. **Installing OpsCenter RHEL/CentOS Packages** .......... 4  
4. **Installing OpsCenter Tar Packages on Linux** .......... 5  
   - Prerequisites ............................................... 5  
   - Installing OpsCenter Dashboard .......................... 6  
5. **Installing OpsCenter Tar Packages on Mac** ............ 7  
   - Prerequisites ............................................... 7  
   - Installing OpsCenter Dashboard .......................... 7  
6. **Installing OpsCenter on Windows** ....................... 8  
   - Starting Services ......................................... 8  
   - Launching OpsCenter ....................................... 8  
   - Troubleshooting ........................................... 9  
7. **About Your OpsCenter Installation** ....................... 10  
   - RPM/Debian Package Install Locations ................... 10  
   - Tar Package Install Locations ........................... 10  
   - OpsCenter Ports ............................................. 11  
8. **Next Steps** .................................................. 11  

## Configuring OpsCenter ........................................ 11  

1. **Configuring the OpsCenter Daemon** ..................... 11  
2. **Configuring Firewall Port Access** ....................... 12  
3. **Configuring User Access** ................................ 12  
4. **Configuring SSL** ........................................... 13  
   - Disabling SSL -- Linux and Mac OSX ..................... 13  
   - Enabling SSL -- Windows .................................. 14  
5. **Configuring Data Collection and Expiration** .......... 16  
   - Excluding Keyspaces and Column Families ............... 16  
   - Changing Performance Data Expiration Times .......... 16  
6. **About Events and Alerts** ................................ 17  
   - Enabling Email Alerts ..................................... 18  
   - Enabling Alerts Posted to a URL ......................... 19
About DataStax OpsCenter

DataStax OpsCenter is a graphical user interface for monitoring and administering Cassandra and Analytics nodes in one centralized management console. OpsCenter Enterprise is bundled with DataStax support offerings. OpsCenter Community Edition is a free version licensed for development or non-production use.

OpsCenter consists of a web application, a dashboard process, and a number of agent processes running on the monitored cluster. The OpsCenter agents are installed on the Cassandra or Analytics nodes, and use Java Management Extensions (JMX) to monitor and manage each node. Cassandra exposes a number of statistics and management operations via JMX. JMX is a Java technology that supplies tools for managing and monitoring Java applications and services such as Cassandra and Hadoop. OpsCenter uses JMX to obtain metrics from a Cassandra and DataStax Enterprise cluster, and to issue various node administration commands such as flushing caches or doing a repair.

Metrics collected from Cassandra or Analytics nodes are stored within the cluster itself. OpsCenter creates its own keyspace within your Cassandra or DataStax Enterprise cluster for storing collected metrics. The OpsCenter keyspace uses a default replication factor of 2. Metrics are collected at one minute intervals. You can configure how long you want to keep historical metrics.

Note
You can only have one instance of OpsCenter running per monitored cluster (including multi-data center clusters). Running multiple OpsCenter dashboard instances on the same monitored cluster is currently not supported.

Getting Started with OpsCenter

Here are the high-level steps for getting started with OpsCenter:
1. Have your Cassandra or DataStax Enterprise cluster up and running before installing OpsCenter. You will need the following information from your cluster in order to configure OpsCenter:
   - The IP addresses of your Cassandra seed nodes
   - The JMX port for Cassandra. In Cassandra versions 0.8 and higher, the JMX port is 7199. In Cassandra versions 0.7 and lower, the JMX port is 8080.

2. Install OpsCenter. See Installing the OpsCenter Dashboard.

3. Configure OpsCenter and JMX connectivity within your cluster. See Configuring OpsCenter.

4. Install the OpsCenter agents on all nodes in your cluster. See Installing OpsCenter Agents.

**OpsCenter Capacity Planning**

OpsCenter collects system and column family metrics data for each node in your cluster. Metrics data is collected at regular intervals and stored within your cluster in a keyspace called OpsCenter. The metrics column family can continue to grow as metrics are collected. Data is not automatically expired.

To calculate capacity, you will need the following information from your Cassandra or DataStax Enterprise cluster:

- The number of nodes in your cluster
- The number of column families you want OpsCenter to monitor
- The number of network interfaces per node
- The number of disk devices per node
- The replication factor for the OpsCenter keyspace (default is 2)

To calculate the expected amount of metrics data that OpsCenter will collect per day, copy and paste the following python program into a file:

```python
#!/usr/bin/env python
import sys
import getopt

MB = 1048576

def size(nodes, cfs, rf=2, disks=3, ifcs=3):
    num_metrics = nodes * ((disks*11) + (ifcs*2) + (cfs*14) + 34)
    data_point_size = 24 # bytes
    points_in_day = num_metrics * (1440 + (288*3) + (48*3) + (12*3) + (1*3))
    data_per_day = rf * data_point_size * points_in_day
    print str(int(data_per_day/MB)) + "MB total in cluster per day"
    print str(int(data_per_day/MB)/nodes) + "MB per node per day"
    data_per_day_plus_overhead = rf * (data_point_size + 16) * points_in_day
    print str(int(data_per_day_plus_overhead/MB)/nodes) + "MB per node per day with SSTable overhead"

usage = "
  usage: opcenter_metrics.py nodes cfs [rf] [disks] [ifcs]
  - nodes: the number of nodes in the cluster
  - cfs: the number of column families in the cluster
  - rf: the replication factor of the OpsCenter keyspace
  - disks: the number of disk devices on each node
  - ifcs: the number of network interfaces on each node"

def main():
    try:
        opts, args = getopt.getopt(sys.argv[1:], "h", ['help'])
    except getopt.error, msg:
        sys.stdout.write(usage)
print usage
sys.exit()

for o, a in opts:
    if o in ("-h", "--help"):
        print usage
        sys.exit()

if len(args) < 2:
    print usage
    sys.exit()

args = map(int, args)
size(*args)

if __name__ == "__main__":
    main()

Save the file and make it executable. For example:

$ chmod +x opscenter_metrics.py

Run the program as follows:

$ ./opscenter_metrics.py <nodes> <column_families> <ops_rf> <disks> <interfaces>

For example:

$ ./opscenter_metrics.py 9 5 2 3 4

Note
You can configure keyspaces and column families for which you do NOT want to collect metrics in Changing Properties in opscenterd.conf.

Installing the OpsCenter Dashboard

OpsCenter packages are available from DataStax for both the Community and Enterprise editions. The DataStax Enterprise package repositories require a username and password to access.

Note
By downloading OpsCenter from DataStax you agree to the terms of the DataStax Enterprise EULA or DataStax Community EULA (End User License Agreements) posted on the DataStax web site.

OpsCenter Installation Prerequisites

OpsCenter requires that the following software be installed. Check for these and install them if they are not found before you install OpsCenter.

- Oracle Java Runtime Environment 1.6.0_19 or later
- Python 2.5, 2.6, or 2.7
- OpenSSL version listed in Configuring SSL unless you disable SSL

To check for these requirements:
Installing the OpsCenter Dashboard

$ java -version
$ python -V
$ openssl version

If you do not have Java installed, see Installing Sun JRE on RedHat Systems or Installing Sun JRE on Ubuntu Systems for instructions.

**Installing OpsCenter Debian/Ubuntu Packages**

DataStax provides OpsCenter packages for Debian 6.0 (Squeeze), Ubuntu Lucid (10.04), Ubuntu Maverick (10.10) and Ubuntu Natty Narwhal (11.04). There are different package repositories for the Community and Enterprise versions of OpsCenter.

These instructions assume that you have the APT package management tool installed, and that you have root access on the machine where you are installing. If you have not already, log in as root. Optionally, you can run the commands using sudo.

1. Edit the aptitude repository source list file (/etc/apt/sources.list).

   # vi /etc/apt/sources.list

2. In this file, add a line for the DataStax repository. Note the different repository locations for the Community and Enterprise versions of OpsCenter. For Enterprise installations, you will also need the DataStax account <username> and <password> from your registration confirmation email.

   For the DataStax Community (free) version of OpsCenter:

   deb http://debian.datastax.com/community stable main

   For the DataStax Enterprise (paid) version of OpsCenter:

   deb http://<username>:<password>@debian.datastax.com/enterprise stable main

3. Save and close the /etc/apt/sources.list file after you are done adding the appropriate DataStax repositories.

4. Add the DataStax repository key to your aptitude trusted keys.

   # wget -O - http://debian.datastax.com/debian/repo_key | apt-key add -

5. Install the OpsCenter package using the APT Package Manager.

   For the Community (free) version of OpsCenter:

   # aptitude update
   # aptitude install opscenter-free

   For the Enterprise (paid) version of OpsCenter:

   # aptitude update
   # aptitude install opscenter

**Installing OpsCenter RHEL/CentOS Packages**

DataStax provides yum repositories for RedHat Enterprise Linux (RHEL) and CentOS versions 5.4, 5.5, 5.6, 6.0 and 6.1. There are different package repositories for the Community and Enterprise versions of OpsCenter.
These instructions assume that you have the `yum` package management application installed, and that you have root access on the machine where you are installing OpsCenter console. If you have not already, log in as root. Optionally, you can run the commands using `sudo`.

1. (CentOS only) EPEL (Extra Packages for Enterprise Linux) contains dependent packages required by OpsCenter, such as `python26`. EPEL must be installed on the OpsCenter machine. To install the epel-release package:

   For 32-bit systems:
   ```bash
   # rpm -Uvh http://download.fedora.redhat.com/pub/epel/5/i386/epel-release-5-4.noarch.rpm
   ```
   For 64-bit systems:
   ```bash
   # rpm -Uvh http://download.fedoraproject.org/pub/epel/5/x86_64/epel-release-5-4.noarch.rpm
   ```

2. Add a `yum` repository specification for the DataStax repository in `/etc/yum.repos.d`. For example:

   ```bash
   # vi /etc/yum.repos.d/datastax.repo
   ```

3. In this file, add the following lines for the DataStax repository. Note the different repository locations for the Community and Enterprise versions of OpsCenter. For Enterprise installations, you will also need your DataStax account `<username>` and `<password>` from your registration confirmation email.

   For the Community (free) version of OpsCenter:
   ```bash
   [opscenter]
   name= DataStax Repository
   baseurl=http://rpm.datastax.com/community
   enabled=1
   gpgcheck=0
   ```

   For the Enterprise (paid) version of OpsCenter:
   ```bash
   [opscenter]
   name= DataStax Repository
   baseurl=http://<username>:<password>@rpm.datastax.com/enterprise
   enabled=1
   gpgcheck=0
   ```

4. Install the OpsCenter package using `yum`.

   For the Community (free) version of OpsCenter:
   ```bash
   # yum install opscenter-free
   ```

   For the Enterprise (paid) version of OpsCenter:
   ```bash
   # yum install opscenter
   ```

### Installing OpsCenter Tar Packages on Linux

DataStax provides binary tar packages for the Community and Enterprise versions of OpsCenter. The tar packages allow you to install OpsCenter in a single directory and does not require root permissions.

**Prerequisites**

The machines on which you install OpsCenter must have:

- Sun Java Runtime Environment version 1.6.0_19 or higher.
Installing the OpsCenter Dashboard

- python 2.5 or higher.
- openssl
- (agent nodes only) The sysstat package (for collection of I/O system metrics).

To check for these prerequisites:

```bash
$ java -version
$ python -V
$ iostat -V
$ which openssl
```

If you do not have Java installed, see Installing Sun JRE on RedHat Systems or Installing Sun JRE on Ubuntu Systems for instructions.

**Installing OpsCenter Dashboard**

1. Download the tar package of OpsCenter.
   For the Community (free) and Enterprise (paid) versions, you can either download the packages from the DataStax website or use the `wget` command to download the packages directly to your machine. An example of using `wget` to download the Community edition of OpsCenter version would be:

   ```bash
   $ wget http://downloads.datastax.com/community/opscenter.tar.gz
   ```

   An example of using `wget` to download the Enterprise (paid) version of OpsCenter would be:

   ```bash
   $ wget http://<username>:<password>@downloads.datastax.com/enterprise/opscenter.tar.gz
   ```

2. Unpack the distribution.
   To unpack the Community (free) version of OpsCenter:

   ```bash
   $ tar -xzvf <opscenter_package_name>-free.tar.gz
   ```

   To unpack the Enterprise (paid) version of OpsCenter:

   ```bash
   $ tar -xzvf <opscenter_package_name>.tar.gz
   ```

   Use `opscenter-1.4` for the `opscenter_package_name` to unpack the Community or Enterprise version of OpsCenter 1.4, for example.

3. For convenience, set the following environment variables in your user environment (such as in your `$HOME/.bashrc` file):

   ```bash
   export OPSC_HOME=<install_location>/opscenter-1.4
   export PATH="$PATH:$OPSC_HOME/bin"
   ```

4. Set the required configuration properties for your environment in the `opscenterd.conf` file located in `$OPSC_HOME/conf/opscenterd.conf`.

5. From the OpsCenter home directory, run the `bin/setup.py` script. This sets up SSL for OpsCenter and creates a `agent.tar.gz` file for installing the agents (may take a couple of minutes to return). On Linux, for example:

   ```bash
   $ cd $OPSC_HOME
   $ ./bin/setup.py
   ```

   $ opscenter

   By default OpsCenter starts in the background. To start OpsCenter in the foreground, use the -f option.

   $ opscenter -f

7. Use the agent.tar.gz file that gets created to manually install the OpsCenter agents on the nodes in your Cassandra or DSE cluster. See Manually Installing Agents in Tar Installations. The tar installation does not support automatic agent installs.

**Installing OpsCenter Tar Packages on Mac**

DataStax provides binary tar packages for the Community version of OpsCenter for Mac. The tar package allows you to install OpsCenter in a single directory and does not require root permissions.

**Prerequisites**

The machine on which you install OpsCenter must have:

- Java Runtime Environment version 1.6.0_19 or higher
- Python 2.5, 2.6, or 2.7
- openssl 0.9.8

To check that you have the most up-to-date version of Java for your Mac, use Software Update available on the Apple menu.

**Installing OpsCenter Dashboard**

1. Download the tar package of OpsCenter from the DataStax website for the edition you wish to use (either Community or Enterprise).

2. Unpack the distribution.
   
   To unpack the Community (free) version:

   $ tar -xzvf <opscenter_package_name>-free.tar.gz

   To unpack the Enterprise (paid) version:

   $ tar -xzvf <opscenter_package_name>.tar.gz

   Use opscenter-1.4 for the opscenter_package_name to unpack the Community or Enterprise version of OpsCenter 1.4, for example.

3. Change directories to the OpsCenter home directory and run the setup program in the /bin directory:

   $ cd $OPSC_HOME
   $ ./bin/setup.py

4. Start up OpsCenter by issuing the following command from the bin directory:

   $ ./opscenter

   By default, OpsCenter starts in the background.

   To start up OpsCenter in the foreground from the bin directory, use the -f option:

   $ ./opscenter -f
5. Next, configure the OpsCenter agent to run on the machine. From the OpsCenter home directory, change to the agent/bin directory. Run the setup program and pass it the local IP address of the Mac machine twice.

```
$ cd $OPSC_HOME/agent/bin
$ setup 127.0.0.1 127.0.0.1
```

6. Start the OpsCenter agent from the bin directory.

```
$ ./opscenter-agent
```

By default, OpsCenter agent starts in the background.

To start up OpsCenter agent in the foreground from the bin directory, use the -f option:

```
$ ./opscenter-agent -f
```

7. Invoke your Firefox, Safari, or Google Chrome Web browser and test connecting to OpsCenter by entering http://127.0.0.1:8888/opscenter/index.html in the browser's address bar.

**Installing OpsCenter on Windows**

DataStax provides a GUI installer for installing both OpsCenter and Cassandra on Windows. Simply download the Windows installer for your chosen platform (32 or 64-bit Windows 7 or Windows Server 2008) from the DataStax website and follow the installation wizard to install and configure OpsCenter.

**Starting Services**

During installation, you can accept options to start the DataStax Cassandra Server and Opscenter services automatically. When installation completes, and whenever the computer reboots, these services start.

After installation, the OpsCenter Agent service also starts automatically if you choose to start the OpsCenter service.

**Launching OpsCenter**

To finish installing DataStax Community Edition, accept the option to launch OpsCenter immediately.
To launch OpsCenter later, enter the following URL in a browser:

http://localhost:8888/opscenter

**Troubleshooting**

Problems starting up OpsCenter and delays in stopping the OpsCenter Agent service are common and easily solved. **OpsCenter does not start up**

If OpsCenter does not start up when you enter the URL to launch it, the most likely culprits are:

- DataStax services are not running.

  Solution: Check that the Cassandra and OpsCenter services are running. If not, start them in the Control Panel.
Microsoft Visual C++ 2008 Redistributable Package is not installed.
Solution: Check that this package is installed. On modern computers, the package is most likely installed. If not, download, and install the package from Microsoft for 32- and 64-bit systems.

Stopping the OpsCenter Agent service takes a long time
Be patient when manually stopping the OpsCenter Agent service, the service will stop eventually.

About Your OpsCenter Installation
Before starting OpsCenter and installing the agents, make the required settings described in Configuring OpsCenter.

RPM/Debian Package Install Locations
The OpsCenter packaged releases create an opscenter user. When starting the OpsCenter dashboard as a service, the service runs as this user. A service initialization script is located in /etc/init.d. Run levels are not set by the package.

The OpsCenter RPM and Debian packages install into the following directories:

• /var/lib/opscenter (SSL certificates for encrypted agent/dashboard communications)
• /var/log/opscenter (log directory)
• /var/run/opscenter (runtime files)
• /usr/share/opscenter (jar, agent, web application, and binary files)
• /etc/opscenter (configuration files)
• /etc/init.d (service startup script)
• /etc/security/limits.d (OpsCenter user limits)

Tar Package Install Locations
The tar packages create the following directories in `OPSC_HOME`:

- `/agent` (agent installation files)
- `/bin` (startup and configuration binaries)
- `/conf` (configuration files)
- `/content` (web application files)
- `/doc` (license files)
- `/lib` and `/src` (library files)
- `/log` (OpsCenter log files)
- `/ssl` (ssl files for OpsCenter to agent communications)

**OpsCenter Ports**

OpsCenter uses the following ports:

- **61620, 61620** - OpsCenter agent ports.
- **50031** - DataStax Enterprise Only. HTTP port for proxy requests to the Hadoop job tracker node.

OpsCenter agents (on the monitored nodes) use the following ports:

- **7199** - JMX port.
- **61620, 61621** - OpsCenter agent ports.
- **9290** - DataStax Enterprise Only. Hadoop job tracker node Thrift port.
- **50030** - DataStax Enterprise Only. Hadoop job tracker node HTTP port.

**Next Steps**

- *Configuring OpsCenter*
- *Installing OpsCenter Agents*

**Configuring OpsCenter**

Before starting OpsCenter for the first time, you must configure connectivity between OpsCenter and your monitored cluster.

**Configuring the OpsCenter Daemon**

The main configuration file for the OpsCenter daemon is `/etc/opscenter/opscenterd.conf`. Before you start OpsCenter for the first time, you must configure the properties to allow it to connect to your Cassandra or DataStax Enterprise cluster.

This section describes the minimal configuration required to get OpsCenter up and running. See `Changing Properties in opscenterd.conf` for an explanation of all OpsCenter daemon configuration properties.

1. Edit the `/etc/opscenter/opscenterd.conf` file.
2. In this file make sure the `[jmx] port` value is correct for your version of Cassandra. Set the `webserver` interface property to the public IP or hostname of the OpsCenter machine. Set the `cassandra` seed_hosts property to a comma-delimited list of seed nodes for your Cassandra or DataStax Enterprise cluster. For example:

```
[jmx]
port = 7199

[webserver]
port = 8888
interface = 110.123.4.5

[cassandra]
seed_hosts = 172.123.4.5, 172.456.7.8
```

### Configuring Firewall Port Access

OpsCenter communicates with the monitored cluster on certain ports. If you have a firewall running on your Cassandra or DataStax Enterprise nodes, you must open up the following ports on your monitored nodes to allow traffic from the OpsCenter machine.

<table>
<thead>
<tr>
<th>Node Port</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7199</td>
<td>JMX port.</td>
</tr>
<tr>
<td>61620, 61621</td>
<td>OpsCenter agent ports.</td>
</tr>
<tr>
<td>9290</td>
<td>DataStax Enterprise Only. Hadoop job tracker node Thrift port.</td>
</tr>
<tr>
<td>50030</td>
<td>DataStax Enterprise Only. Hadoop job tracker node HTTP port.</td>
</tr>
</tbody>
</table>

The following ports must be open on the OpsCenter machine to allow traffic from your Cassandra or DataStax Enterprise nodes, or from other machines in your network that are accessing OpsCenter.

<table>
<thead>
<tr>
<th>OpsCenter Port</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8888</td>
<td>OpsCenter web application HTTP port.</td>
</tr>
<tr>
<td>61620</td>
<td>OpsCenter agent port.</td>
</tr>
<tr>
<td>50031</td>
<td>DataStax Enterprise Only. HTTP proxy port for Hadoop job tracker requests.</td>
</tr>
</tbody>
</table>

### Configuring User Access

By default, OpsCenter is deployed without access control or user authentication enabled. To enable access control, you can add users, assign them a password, and set their role using the `set_passwd.py` utility. Until you configure authentication for OpsCenter users, any user with access to the OpsCenter dashboard can view all objects and perform all actions.

OpsCenter provides two access roles: `admin` and `user`. The following actions are admin-only:

- Create, modify, or drop keyspaces
- Create, modify, or drop column families
- Execute flush, cleanup, compact, drain, repair, move, or decommission actions on a node
- Install OpsCenter agent on Cassandra nodes

To create or modify users with specific roles, run `/usr/share/opscenter/bin/set_passwd.py`. For example, to create user johndoe with admin role privileges:
$ python /usr/share/opscenter/bin/set_passwd.py johndoe admin
Please enter a password for 'johndoe'.
Password:

After configuring authentication, restart OpsCenter:

$ service opscenterd restart

Once you have a created at least one user, OpsCenter will require authentication for all access.

### Configuring SSL

OpsCenter uses Secure Socket Layer (SSL) to encrypt the communication protocol and authenticate traffic between OpsCenter agents and the main OpsCenter daemon on Linux or the DataStax OpsCenter Service on Windows. The default SSL state depends on the operating system:

- Linux and Mac OSX: enabled
- Windows: disabled

You can change the state of SSL by following the procedure for Linux and Mac OSX or for Windows. Consider disabling SSL if you are running OpsCenter and DataStax Community or 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.
- In automatic deployments of OpsCenter to avoid reinstallation of agents.
- On a computer that does not have the required version of OpenSSL.

If you have no need for SSL, you can simply disable the SSL option to avoid installing OpenSSL.

If the SSL option is enabled, OpsCenter requires a specific version of OpenSSL, which is typically installed by default on supported Linux operating systems. This table lists required SSL versions for supported operating systems:

<table>
<thead>
<tr>
<th>Supported O/S</th>
<th>Required SSL Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mac OSX</td>
<td>0.9.8</td>
</tr>
<tr>
<td>Red Hat Enterprise Linux/Centos 5</td>
<td>0.9.8</td>
</tr>
<tr>
<td>Red Hat Enterprise Linux/Centos 6</td>
<td>1.0.0</td>
</tr>
<tr>
<td>Ubuntu/Debian</td>
<td>0.9.8</td>
</tr>
<tr>
<td>Windows</td>
<td>0.9.8</td>
</tr>
</tbody>
</table>

To determine which version of OpenSSL is installed, use the following command on the Windows, Linux, or Mac OSX command line:

```
openssl version
```

### Disabling SSL -- Linux and Mac OSX

You modify the configuration files for OpsCenter and its agent to enable or disable SSL on Linux and Mac OSX.

To disable SSL:
1. Open the configuration file for OpsCenter, opscenterd.conf, for editing.

   cd /etc/opscenter
   sudo vi opscenterd.conf

2. Add an agents section and the use_ssl parameter, and set its value to false (or 0).

   [agents]
   use_ssl = false

3. Open the configuration file for OpsCenter agent, address.yaml, for editing. If you install OpsCenter using a tar package, running the agent setup step generates address.yaml. If you install the agent packages (rpm or deb) manually, the configuration file is in /var/lib/opscenter-agent/conf.

   cd /var/lib/opscenter-agent/conf
   sudo vi address.yaml

4. In the address.yaml file, add the command to use SSL and set its value to 0.

   use_ssl: 0

5. Restart the OpsCenter agent.

   • If you installed DataStax manually using a tar, get the process ID of the agent, kill the process, and issue the commands to restart the agent.

     ps -A
     kill <pid>
     ./bin/opscenter-agent

   • If you installed DataStax through the user interface or manually, but using an agent deb or rpm package, issue this command to restart the agent.

     sudo service opscenter-agent restart

**Enabling SSL -- Windows**

You run setup.py, modify the configuration files for OpsCenter and its agent, and restart the DataStax OpsCenter Agent Service to enable SSL on Windows.

To enable SSL:
1. Navigate to the opscenter\bin directory. For example, if you selected the defaults during installation, navigate to this location:

2. Click or double-click setup.py to run it.

setup.py

The agentKeyStore key pairs appear in opscenter\ssl.

3. Navigate to the opscenter\conf directory.

4. Open the configuration file for OpsCenter, opscenterd.conf, in a text editor such as Notepad or Wordpad.

2. In the agents section, change use_ssl from 0 to 1 (or true).

3. Navigate to the opscenter\agent\conf directory.

4. Open the configuration file for OpsCenter agent, address.yaml, in a text editor.
5. In the address.yaml file, change the value for use_ssl from 0 to 1.

   use_ssl: 1

6. From the Control Panel, restart the DataStax OpsCenter Agent Service.

   ![Control Panel screenshot showing DataStax OpsCenter Agent](image)

### Configuring Data Collection and Expiration

To help control consumption of disk space, OpsCenter provides two ways to limit the growth of OpsCenter performance data:

- The ability to exclude specified keyspaces and column families from performance data collection.
- Automatic expiration of performance data after a configurable time period.

### Excluding Keyspaces and Column Families

By default, OpsCenter does not collect performance data for its own keyspace nor for the Cassandra system keyspace. You can manually add any other keyspaces or column families that you do not want to monitor in the [cassandra_metrics] section of the configuration file.

For example, to prevent data collection for the keyspace `test` as well as the column family `Keyspace1.Standard1`, you would uncomment and edit the following values in `/etc/opscenter/opscenterd.conf`:

```ini
[cassandra_metrics]
ignored_keyspaces = system, OpsCenter, test
ignored_column_families = Keyspace1.Standard1
```

Note that column families are specified in the format `<keyspace_name>..<column_family_name>`.

### Changing Performance Data Expiration Times
Configuring OpsCenter

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 rollups.

- One-minute rollups (1min_ttl) expire after after one week, or 604800 seconds.
- Five-minute rollups (5min_ttl) expire after four weeks, or 2419200 seconds.
- Two-hour rollups (2hr_ttl) expire after one year, or 31536000 seconds.

If you want to change expiration thresholds, uncomment and edit the time-to-live (ttl) values in /etc/opscenter/opscenterd.conf, and then restart OpsCenter. In the following example, the one-minute and five-minute rollups are set to expire twice as fast as the defaults, and two-hour rollups are set to be kept indefinitely (expiration is disabled):

```
1min_ttl = 302400
5min_ttl = 1209600
2hr_ttl = -1
```

All new data collected after restarting will expire at the newly set expiration time. Data collected prior to your changes continues to expire at the previously specified rate.

About Events and Alerts

The OpsCenter dashboard’s System Log page displays a continuously updated list of events. This list reflects the most detailed logging level available for Cassandra and OpsCenter events. Levels available, with their numeric codes, are DEBUG(0), INFO (1), WARN (2), ERROR (3), CRITICAL (4), ALERT (5).

Data for these events is stored in the events and events_timeline column families in the OpsCenter keyspace:

<table>
<thead>
<tr>
<th>Event</th>
<th>Code</th>
<th>Note/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPACTION</td>
<td>0</td>
<td>Major compaction has occurred.</td>
</tr>
<tr>
<td>CLEANUP</td>
<td>1</td>
<td>Unused keys have been removed or cleaned up.</td>
</tr>
<tr>
<td>REPAIR</td>
<td>2</td>
<td>A repair operation has been initiated.</td>
</tr>
<tr>
<td>FLUSH</td>
<td>3</td>
<td>Memtables have been flushed to disk.</td>
</tr>
<tr>
<td>DRAIN</td>
<td>4</td>
<td>The commitlog has been emptied, or drained.</td>
</tr>
<tr>
<td>DECOMMISSION</td>
<td>5</td>
<td>A leaving node has streamed its data to another node.</td>
</tr>
<tr>
<td>MOVE</td>
<td>6</td>
<td>Like NODE_MOVE; a new token range has been assigned.</td>
</tr>
<tr>
<td>NODE_DOWN</td>
<td>13</td>
<td>A node has stopped responding.</td>
</tr>
<tr>
<td>NODE_UP</td>
<td>14</td>
<td>An unresponsive node has recovered.</td>
</tr>
<tr>
<td>NODE_LEFT</td>
<td>15</td>
<td>A node has left, or been removed from, the ring.</td>
</tr>
<tr>
<td>NODE_JOIN</td>
<td>16</td>
<td>A node has joined the ring.</td>
</tr>
<tr>
<td>NODE_MOVE</td>
<td>17</td>
<td>A node has been assigned a new token range (the token has moved).</td>
</tr>
<tr>
<td>OPSC_UP</td>
<td>18</td>
<td>OpsCenter has been started and is operating.</td>
</tr>
<tr>
<td>OPSC_DOWN</td>
<td>19</td>
<td>OpsCenter was stopped or stopped running.</td>
</tr>
<tr>
<td>GC</td>
<td>20</td>
<td>Java garbage collection has been initiated.</td>
</tr>
</tbody>
</table>

Optionally, you can configure OpsCenter to send alerts for selected levels of events. These alerts can be provided remotely by email, or through HTTP to a selected URL. They are disabled by default.
Note
In current releases, alerts are triggered only by events from the OpsCenter dashboard. For example, a `nodetool move` operation submitted from the command line will not trigger an alert; a move operation launched using `List View > Actions > Move` controls in the OpsCenter dashboard will.

All alerts contain the following information about each event captured:

<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.</td>
<td>67.169.50.240</td>
</tr>
<tr>
<td>target_node</td>
<td>Destination of a STREAMING action.</td>
<td>50.1.1.11</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 (see above table)</td>
<td>20</td>
</tr>
<tr>
<td>message</td>
<td>Description of the event.</td>
<td>Garbage Collecting node 50.1.1.13</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>50.1.1.13</td>
</tr>
<tr>
<td>level_str</td>
<td>Logging level of the event.</td>
<td>INFO</td>
</tr>
</tbody>
</table>

Enabling Email Alerts
OpsCenter can post alerts to a selected email addresses. To enable email alerts, you must edit `/etc/opscenter/event-plugins/email.conf` and provide valid SMTP server host and port information.

To enable email alerts

1. Make sure you have valid SMTP mail accounts to send and receive alerts.
2. On the OpsCenter daemon host, open `/etc/opscenter/event-plugins/email.conf` for edit.
3. Set `enabled` to 1. (default is 0, or disabled).
4. Provide valid values for your SMTP host, port, user and password.
5. Enable SSL or TLS, if required for secure communications. Most typical scenarios require SSL.
6. Provide valid values for the "To" and "From" email addresses. The `to_addr` value is the account that will receive alerts.
7. Optionally, set the level of alerts to send and the desired subject line.
8. Save `email.conf` and restart the OpsCenter daemon.

In a system with email alerts enabled for critical and alert-level events, `email.conf` would look like the following:

```
[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=CRITICAL,ALERT
smtp_host=smtp.gmail.com
smtp_port=465
smtp_user=mercury@gmail.com
smtp_pass=*********
smtp_use_ssl=1
```
smtp_use_tls=0
to_addr=cassandra_admin@acme.com
from_addr=mercury@gmail.com
subject=OpsCenter Event

To enable email alerts to multiple addresses:
Create a different email conf file with settings for each email address. All conf files are loaded so you can name them email1.conf, email2.conf, and so on.

Enabling Alerts Posted to a URL
OpsCenter can post alerts to a URL, provided that it contains an appropriate script for posting. For example, a simple PHP script containing `print_r($_POST);` should be sufficient for getting started.

To enable URL posting on the OpsCenter side, you must edit /etc/opscenter/event-plugins/posturl.conf and provide a path to your script.

To enable URL posting
1. Make sure your web server and posting script are configured to receive alerts.
2. On the OpsCenter daemon host, open /etc/opscenter/event-plugins/posturl.conf for edit.
3. Set enabled to 1 (default is 0, or disabled).
4. For url, provide a valid path to your posting script such as `url=http://50.1.1.11/postOPSCevents.php`.
5. Optionally, select the desired logging level. The default is to listen for all levels of events.
6. Save posturl.conf and restart the OpsCenter daemon.

In a system with posting enabled for critical and alert-level events, posturl.conf would look like the following:

```
[posturl]
enabled=1
url=http://50.1.1.11/postOPSCevents.php
# 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=CRTICAL,ALERT
```

The details of how the posting is handled on the receiving side are open to your preference. You can perform a simple test to verify that events are posting correctly by logging to a file such as /tmp/events on the web server host. For this test, a PHP script at `http://50.1.1.11/postOPSCevents.php` would look like the following:

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

After deploying this script (a web server restart may be required), launch a logged event such as an OpsCenter restart or a garbage compaction from the dashboard's Cluster > List View. Output properly posted to /tmp would look like:

```
Array
(
    [api_source_ip] => 67.169.50.240
    [target_node] => None
    [event_source] => OpsCenter
    [user] => None
    [time] => 1311025598851602
    [action] => 20
    [message] => Garbage Collecting node 50.1.1.24
)```
Installing OpsCenter Agents

The OpsCenter agents are used to collect data from your Cassandra or DataStax Enterprise (DSE) Analytics nodes, and must be installed on each node in your cluster. You can have OpsCenter install the agents automatically, or manually install them yourself on each node.

**Installing OpsCenter Agents (Automatic Install)**

When you launch OpsCenter for the first time, you can choose to have the OpsCenter daemon automatically install the agents for you on your Cassandra or DSE Analytics nodes. Automatic installation requires root access to your cluster nodes and you must supply login credentials. If you do not want to have OpsCenter install the agents automatically, you may choose to install the agents manually on each node in your Cassandra or DataStax Enterprise cluster.

After you have installed and configured OpsCenter, and enabled JMX connectivity within your Cassandra or DataStax Enterprise cluster, you are ready to install the OpsCenter agents on all of your nodes. When you start OpsCenter for the first time, you will be prompted to install the agents automatically.

These instructions assume that you have root access on the machines where you are installing the agents. If you have not already, log in as root. Optionally, you can run the commands using sudo. Usage for the startup service is `opscenterd {start|stop|status|restart|force-reload}`.

1. On the OpsCenter machine, start the OpsCenter daemon service.

   ```
   $ service opscenterd start
   ```

2. Open a browser window and go to the OpsCenter console URL at `http://<opscenter_host>:8888`, where `<opscenter_host>` is the IP or hostname of the OpsCenter machine. For example:

   ```
   http://110.123.4.5:8888
   ```

3. After the console loads, click the **Install Agents** button in the left navigation pane.
4. In the **Install Node Agent** dialog, click **Enter Credentials**.

5. In the **Node SSH Credentials** dialog, enter a **Username** that has root privileges or sudo access to all of the nodes in your cluster. Then enter either a **Password** or a **Private Key** to be used to authenticate the user on your cluster nodes. Click **Done**.

6. In the **Install Node Agent** dialog, click **Install on all Nodes**.
7. If prompted, click **Accept Fingerprint** to add a node to the known hosts for OpsCenter.

8. It will take a few minutes for OpsCenter to complete the agent installations. After all agents are installed, you will see a success message.

9. The OpsCenter agents do not pick up the environment variables of the currently logged in user by default (such as JAVA_HOME). If it cannot find Java in the machine's PATH, you may notice errors in the on agent log on start-up such as:

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

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

   ```
   JAVA_HOME=/usr/bin/java
   ```

### Installing OpsCenter Agents (Manual Install)

These instructions are for installing the OpsCenter agents manually.

If you installed OpsCenter using the RPM or Debian packages follow the instructions for *Manually Installing Agents in Packaged Installations*.

If you installed OpsCenter using the tar packages follow the instructions for *Manually Installing Agents in Tar Installations*.

### Manually Installing Agents in Packaged Installations

These instructions are if you have installed OpsCenter using the RPM or Debian packaged releases. If you installed OpsCenter using the tar package, see *Manually Installing Agents in Tar Installations*.

After you have installed and configured OpsCenter, and enabled JMX connectivity within your Cassandra or DataStax Enterprise cluster, you are ready to install the OpsCenter agents on all of your nodes. These instructions are for manually installing the agent software. Perform this procedure on each node in your Cassandra or DataStax Enterprise cluster.
Installing OpsCenter Agents

If installing manually, DataStax recommends installing using the agent packages (requires root access). When installed via the packages, the agent is installed as a service. The agent service will automatically start on machine startup, as well as be monitored and restarted if it dies for any reason.

1. On the OpsCenter machine, go to the agent directory.
   ```
   $ cd /usr/share/opscenter/agent
   ```

2. Copy the agent software to your cluster node. For example:
   ```
   $ scp opscenter-agent.tar.gz root@<cassandra_node_IP>:/
   ```

3. Log in to the node, copy the agent installer tar file to the desired location, and unpack it. For example:
   ```
   $ ssh root@<cassandra_node_IP>
   $ cp /opscenter-agent.tar.gz /usr/share
   $ cd /usr/share
   $ tar -xzf opscenter-agent.tar.gz
   ```

4. If you have root access, install the agent using the rpm or deb package (recommended). For example, where `<package_type>` is either rpm or deb, `<opscenter_host>` is the IP or hostname of the OpsCenter machine, and `<node_listen_address>` is the configured listener IP address for the Cassandra node (typically the same as the output of `hostname -i`):
   ```
   $ opscenter-agent/bin/install_agent.sh opscenter-agent/opscenter-agent.<package_type> <opscenter_host> <node_listen_address>
   ```
   For example:
   ```
   $ hostname -i
   172.123.4.5
   $ cd opscenter-agent
   $ bin/install_agent.sh opscenter-agent.rpm 110.123.4.5 172.123.4.5
   ```

5. If you do not have root access, configure the agent and then start it as a stand-alone process. For example, where `<opscenter_host>` is the IP or hostname of the OpsCenter machine, and `<node_listen_address>` is the configured listener IP address for the Cassandra node (typically the same as the output of `hostname -i`):
   ```
   $ opscenter-agent/bin/setup <opscenter_host> <node_listen_address>
   $ bin/opscenter-agent
   ```
   For example:
   ```
   $ hostname -i
   172.123.4.5
   $ cd opscenter-agent
   $ bin/setup 110.123.4.5 172.123.4.5
   $ bin/opscenter-agent
   ```
   By default, OpsCenter agent starts in the background. Use the -f option to start OpsCenter agent in the foreground.
   ```
   $ bin/opscenter-agent -f
   ```

**Note**

If you do not install using the agent deb or rpm package, make sure that your nodes also have the sysstat package installed. This package is needed for the collection of I/O metrics.
6. Open a browser window and go to the OpsCenter console URL at http://<opscenter_host>:8888, where 
<opscenter_host> is the IP or hostname of the OpsCenter machine. For example:

http://110.123.4.5:8888

Manually Installing Agents in Tar Installations

These instructions are if you have installed OpsCenter using the tar distributions. If you installed OpsCenter using the RPM or Debian packages, see Manually Installing Agents in Packaged Installations.

1. As part of installing the OpsCenter Dashboard tar files, you should have a agent.tar.gz file that you can use to install the agents.

2. Copy the agent tar file to your cluster node and unpack it. For example:

   $ cd $OPSC_HOME
   $ scp agent.tar.gz <cassandra_node_IP>:$HOME/

3. Log in to the node, copy the agent installer tar file to the desired location, and unpack it. For example:

   $ ssh <cassandra_node_IP>
   $ mkdir $HOME/opscenter-agent
   $ cp $HOME/agent.tar.gz $HOME/opscenter-agent
   $ cd $HOME/opscenter-agent
   $ tar -xzf agent.tar.gz

4. Configure and start the agent where <opscenter_host> is the IP or hostname of the OpsCenter machine, and <node_listen_address> is the configured listener IP address for the Cassandra node (typically the same as the output of hostname -i):

   $ cd agent
   $ ./bin/setup <opscenter_host> <node_listen_address>

5. Start the agent in the background:

   $ ./bin/opscenter-agent

   Alternatively, start the agent in the foreground.

   $ ./bin/opscenter-agent -f

Changing Properties in opscenterd.conf

The opscenterd.conf file is the main configuration file for DataStax OpsCenter, located in /etc/opscenter/opscenterd.conf in packaged installations or $OPSC_HOME/conf in tar installations. After changing properties in this file, you must restart OpsCenter for the changes to take effect.

Cassandra Connection Properties

The following properties are used to inform OpsCenter about the Cassandra or Brisk cluster that it is monitoring. An installation of OpsCenter can only monitor one cluster at a time.

[jmx] port

The JMX (Java Management Extensions) port of your Cassandra or Brisk cluster. In Cassandra versions 0.8 and higher, the JMX port is 7199. In Cassandra versions 0.7 and lower, the JMX port is 8080.

[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 configured for your Cassandra or Brisk cluster. Same as the seeds property in the cassandra.yaml configuration file of your Cassandra or Brisk cluster.

[cassandra] api_port
The Thrift remote procedure call port configured for your Cassandra or Brisk cluster. Same as the rpc_port property in the cassandra.yaml configuration file of your Cassandra or Brisk cluster. Default is 9160.

**Metrics Collection Properties**
The following properties are used to limit the keyspaces and column families for which you collect metrics.

[cassandra_metrics] ignored_keyspaces
A comma-delimited list of Cassandra keyspaces for which you do not want to collect performance metrics. By default, the system and OpsCenter keyspaces are excluded. For Brisk clusters, you may also want to consider excluding the cfs keyspace.

[cassandra_metrics] ignored_column_families
A comma-delimited list of Cassandra column families for which you do not want to collect performance metrics. Entries should be in the form of keyspace_name.*columnfamily_name*.

[stat_reporter] interval
By default, OpsCenter periodically sends usage metrics about your cluster to DataStax support. To disable the phone-home functionality, add the following lines to your opscenterd.conf file:

```plaintext
[stat_reporter]
  interval = 0
```

**Performance Data Expiration Properties**
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.

These properties must be uncommented if you want to change the default expiration periods for performance data. Each property is shown below with its default value.

1min_ttl = 604800
One-minute rollups expire after after one week, or 604800 seconds.

5min_ttl = 2419200
Five-minute rollups expire after four weeks, or 2419200 seconds.

2hr_ttl = 31536000
Two-hour rollups expire after one year, or 31536000 seconds.

**OpsCenter Configuration Properties**
The following properties are used to configure OpsCenter.

**[webserver] port**
The HTTP port used for client connections to the OpsCenter web server. Default is 8888.

**[webserver] interface**
The interface that the web server will use to listen for client connections. Should be an externally accessible IP address or hostname.

**[logging] level**
Sets the logging level for OpsCenter. Available levels are (from most to least verbose): TRACE, DEBUG, INFO, WARN, or ERROR. The OpsCenter log file is located in `/var/log/opscenter/opscenterd.log`.

**[authentication] passwd_file**
Used to configure password authentication for OpsCenter. If this file does not exist, OpsCenter will not verify passwords. To enable password authentication, use the `set_passwd.py` utility to create users and set their password and role. OpsCenter currently has two available roles: `admin` or `user`.

---

### Release Notes

See these sections for information about new features, issues, and upgrading the following releases:

- **OpsCenter 1.4.1**
- **OpsCenter 1.4**
- **OpsCenter 1.3.1**
- **OpsCenter 1.3**
- **OpsCenter 1.2.3**
- **OpsCenter 1.2.2**
- **OpsCenter 1.2.1**
- **OpsCenter 1.2**

#### OpsCenter 1.4.1

OpsCenter 1.4.1 is a patch release to **OpsCenter 1.4**. For upgrading instructions, follow the instructions for Upgrading **OpsCenter Dashboard**.

#### Resolved Issues

- Fix free-space disk check before moving node.
- Add additional support for Linux device types.
- Improvements to OpsCenter keyspace creation logic.
- Include heap options in tarball agent startup script.

---

### OpsCenter 1.4
OpsCenter 1.4 adds support for two major operating systems and widens support for Python. For upgrading to 1.4, follow the instructions for *Upgrading OpsCenter Dashboard*.

### What's New

- Support for OpsCenter and Cassandra on Windows. Using a GUI installer, you can easily be up and running on a 32- or 64-bit Windows 7 or Windows Server 2008 in a few minutes. OpsCenter and OpsCenter agents run as Windows services.
- Support for OpsCenter on Mac OS X. You can install OpsCenter in a single directory using a tar package. Root permissions are not required.
- Extended support of Python. OpsCenter supports Python 2.5, 2.6, and 2.7 on Linux, Mac OS X, and Windows.

### Resolved Issues

- Detection of OpenSSL fails on RedHat Enterprise Linux (RHEL) 5.x.
- Configuring user access using the `set_passwd.py` utility fails.
- When you replace a node at position 0-1, OpsCenter obscures the new node in the ring diagram.
- When viewing a table of a column family data in Data Explorer, the Row Key heading of the table is incorrectly labeled Super Column when you click the search button.
- When keys and super column names include spaces, the Data Explorer does not display data when you view all columns.
- The Performance graph of data over a period of time momentarily shows a false, steep drop in reads and writes.
- OpsCenter 1.2.3 on Red Hat Enterprise Linux behaves unpredictably when started with `/etc/init.d/opscenterd start`.

### Upgrading OpsCenter Dashboard

If you are upgrading to 1.4.x from 1.3 or later, perform the steps in this section to upgrade OpsCenter Dashboard. Following these steps, you install the new OpsCenter package and restart the `opscenterd` daemon.

If you are upgrading from 1.2.3 or earlier, note the repository changes listed in the next section. If you are upgrading from 1.2.1 or earlier, you need to migrate your performance data to view that data in your upgraded OpsCenter. To migrate performance data, see [122migration](#).

#### To upgrade to OpsCenter:

1. On the OpsCenter daemon host, run the appropriate command to update packages on Debian/Ubuntu:

   ```bash
   # aptitude update
   ```

   or for RHEL/CentOS:

   ```bash
   # yum clean all
   ```

2. Install the upgraded OpsCenter package, where `<package_manager>` is `aptitude` on Debian/Ubuntu or `yum` for RHEL/CentOS:

   ```bash
   # <package_manager> install opscenter
   ```

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

```bash
# service opscenterd restart
```

### Change in Repository Locations for OpsCenter 1.2.3 or Earlier

The OpsCenter package repository locations for releases 1.2.3 or earlier changed. Also the free (DataStax Community) edition of OpsCenter no longer requires a username and password. The paid (DataStax Enterprise) version of OpsCenter does require login credentials. Edit your DataStax repository source list file and change the repo URL as follows:

**On RedHat/CentOS/Fedora** (in `/etc/yum.repos.d/datastax.repo`)

- **Enterprise/Paid**: `http://<username>:<password>@rpm.datastax.com/enterprise`
- **Community/Free**: `http://rpm.datastax.com/community`

**On Debian/Ubuntu** (in `/etc/apt/sources.list`):

- **Enterprise/Paid**:
  
  ```
  deb http://<username>:<password>@debian.datastax.com/community stable main
  
  Community/Free**:

  ```
  
## Upgrading OpsCenter Agents

If OpsCenter agents need to be upgraded for a new release, you will be prompted to do so by the newly upgraded dashboard. Alternatively, you can proactively install them using the **Install Agents** button in the left navigation pane.

If you require manual agent installation, see *Installing OpsCenter Agents (Manual Install)*.

To automatically install upgraded agents using the OpsCenter dashboard:

1. Open or refresh the OpsCenter dashboard in a browser. After the console loads, click the **Install Agents** button in the left navigation pane.
2. In the **Install Node Agent** dialog, click **Enter Credentials**.

![Install Node Agent dialog](image1)

3. In the **Node SSH Credentials** dialog, enter a **Username** that has root privileges or sudo access to all of the nodes in your cluster. Then enter either a **Password** or a **Private Key** to be used to authenticate the user on your cluster nodes. Click **Done**.

![Node SSH Credentials dialog](image2)

4. In the **Install Node Agent** dialog, click **Install on all Nodes**. When a dialog box reports that the agents installed successfully, the OpsCenter upgrade is complete.

![Install Node Agent dialog](image3)
**OpsCenter 1.3.1**

OpsCenter 1.3.1 is a patch release to OpsCenter 1.3. For upgrades to 1.3.1, follow the instructions for Upgrading OpsCenter Dashboard.

**What's New**

This release introduces support for OpsCenter on RedHat Enterprise Linux version 6.

**Resolved Issues**

- (DataStax Enterprise Only) Creating a new Keyspace while a rebalance is in progress causes the rebalance operation to error out.
- "Unable to import needed OpsCenter packages" error when running `set_passwd.py` to set up OpsCenter authentication.
- Data center and version information not returned from agents when `iostat` is missing on monitored nodes.

**OpsCenter 1.3**

OpsCenter 1.3 introduces the DataStax Enterprise (DSE) and DataStax Community (DSC) versions of OpsCenter. DSE contains enhanced features and functionality and is available to registered DataStax customers. DSE is a free version of OpsCenter and no longer requires registration with DataStax to download and use.

**What's New**

- (Enterprise-only) Auto-Rebalancing for Randomly Partitioned Clusters - OpsCenter Enterprise now has the ability to check if nodes are evenly distributed in each data center, and if not automatically moves tokens and data to balance the ring.
- (Enterprise-only) Proactive Alerts - OpsCenter Enterprise now has the ability to configure alert thresholds for commonly watched performance metrics and to send alerts when those thresholds are triggered.
- (Enterprise-only) The ability to track the progress of MapReduce jobs running on DSE Analytics nodes.
- More Reliable JMX Operations on Agent - Improved channels for pushing JMX operations to the agents from `opscenterd`, getting success or failure responses, and polling the status of an operation.
- Self-Contained Distribution - New tarball distribution of OpsCenter to allow for self-contained, non-root installations.

**Resolved Issues**

- Respect the user's JAVA_HOME and PATH environment when starting up `opscenter-agent`
- Sort nodes in correct token order in Cluster list view
- Cluster list view usability improvements
- Limit how many files are rotated for `agent.log`
- Use more precise timestamps for Events list
- OS memory graph not graphing total correctly
- Data Modeling - Fixed Create Keyspace using NetworkTopologyStrategy

**OpsCenter 1.2.3**
This release fixes an issue with the logging mechanisms in version 1.2.2.

If you are upgrading an earlier OpsCenter installation, the procedure differs depending on your platform:

- **Upgrading from Version 1.2.2 to OpsCenter 1.2.3**
- **Upgrading from Version 1.2.1 or Earlier to 1.2.3**

### Upgrading from Version 1.2.2 to OpsCenter 1.2.3

If you are upgrading from 1.2.2, you need only to install the new OpsCenter package and restart the daemon.

**To upgrade OpsCenter from 1.2.2 to version 1.2.3:**

1. On the OpsCenter daemon host, run the appropriate command to update packages on Debian/Ubuntu:
   ```
   # aptitude update
   ```

   or for RHEL/CentOS:
   ```
   # yum clean all
   ```

2. Install the upgraded OpsCenter package, where `<package_manager>` is aptitude on Debian/Ubuntu or yum for RHEL/CentOS:
   ```
   # <package_manager> install opscenter
   ```

3. If the package manager prompts you for options regarding `opscenterd.conf`, choose to keep your currently installed version.

4. Restart the OpsCenter daemon.
   ```
   # service opscenterd restart
   ```

### Upgrading from Version 1.2.1 or Earlier to 1.2.3

If you are upgrading from versions 1.2.1 or earlier, you must install the OpsCenter package, account for new `opscenterd.conf` settings, restart the daemon, and install upgraded agents.

Additionally, if you want to view pre-1.2.2 performance data in OpsCenter, you must also perform the migration described in `122migration` after upgrading.

**To upgrade OpsCenter from 1.2.1 or earlier to version 1.2.3:**

1. On the OpsCenter daemon host, run the appropriate command to update packages on Debian/Ubuntu:
   ```
   # aptitude update
   ```

   or for RHEL/CentOS:
   ```
   # yum clean all
   ```

2. Install the upgraded OpsCenter package, where `<package_manager>` is aptitude on Debian/Ubuntu or yum for RHEL/CentOS:
   ```
   # <package_manager> install opscenter
   ```
3. If the package manager prompts you for options regarding `opscenterd.conf`, choose to install the package distributor's new file. The new `opscenterd.conf` file contains new parameters for the optional configuration of performance data expiration.

On Debian/Ubuntu platforms, a prompt like the following is shown:

```
Configuration file `/etc/opscenter/opscenterd.conf'
==> Modified (by you or by a script) since installation.
==> Package distributor has shipped an updated version.
What would you like to do about it? Your options are:
Y or I : install the package maintainer's version
N or O : keep your currently-installed version
D     : show the differences between the versions
Z     : start a shell to examine the situation
The default action is to keep your current version.
*** opscenterd.conf (Y/I/N/O/D/Z) [default=N] ? I
Installing new version of config file /etc/opscenter/opscenterd.conf ...
```

4. After the package finishes upgrading, edit the configuration file `/etc/opscenter/opscenterd.conf` to include the correct settings for your Cassandra cluster. To run properly, OpsCenter requires valid values for these settings:

   - JMX port (`port`) -- note that this is 7199 in Cassandra 0.8 and 8080 in Cassandra 0.7.
   - Web Interface (`interface`) -- public IP or hostname of the OpsCenter machine.
   - Seed Nodes (`seed_hosts`) -- comma-delimited list of seed nodes for the cluster.

Also, the new `opscenterd.conf` includes these parameters introduced in version 1.2.2:

   - `1min_ttl = 604800`
   - `5min_ttl = 2419200`
   - `2hr_ttl = 31536000`

See *Changing Performance Data Expiration Times* for more information about these parameters.

5. Restart the OpsCenter daemon.

   ```sh
   # service opscenterd restart
   ```

6. When you open the upgraded OpsCenter dashboard in a browser, it will display a notification about migrating old data to the optimized data model. If you choose to migrate data, carefully follow the instructions in `122migration`.

7. The upgraded OpsCenter dashboard will prompt you to automatically install upgraded agents as described in *Upgrading OpsCenter Agents*. If you require manual agent installation, see *Installing OpsCenter Agents (Manual Install)*.

8. Optionally, you can delete the old performance data and save disk space. The migration script prompts you to delete old data; to do it manually, follow the instructions in `manually-removing-old-data`.

**Resolved Issues**

- Action to perform Java garbage collection not functioning properly
- Integer overflow in agent with data streams larger than 2GB

**OpsCenter 1.2.2**

This release introduces a new, optimized data model for storing OpsCenter performance metrics. The new model can save disk space and improve OpsCenter performance on large clusters that generate a high volume of performance data.
Release Notes

data.

Migrating pre-1.2.2 OpsCenter data to the new model is optional. If you are upgrading from an earlier release of OpsCenter, see 122migration for more information on data migration.

What’s New

- New, highly efficient model for performance data.
- Configurable automatic expiration of performance data.

Upgrading

You can upgrade any 1.x installation to version 1.2.2 by upgrading the OpsCenter package, restarting the daemon, and installing upgraded agents.

Note

If you want to view pre-1.2.2 performance data in OpsCenter, you must also perform the migration described in 122migration after upgrading.

To upgrade OpsCenter to version 1.2.2:

1. On the OpsCenter daemon host, run the appropriate command to update packages on Debian/Ubuntu:

   ```bash
   # aptitude update
   ```

   or for RHEL/CentOS:

   ```bash
   # yum clean all
   ```

2. Install the upgraded OpsCenter package, where `<package_manager>` is aptitude on Debian/Ubuntu or yum for RHEL/CentOS:

   ```bash
   # <package_manager> install opscenter
   ```

3. If the package manager prompts you for options regarding `opscenterd.conf`, choose to install the package distributor’s new file. The new `opscenterd.conf` file contains new parameters for the optional configuration of performance data expiration.

   On Debian/Ubuntu platforms, a prompt like the following is shown:

   ```sh
   Configuration file `/etc/opscenter/opscenterd.conf'
   ==> Modified (by you or by a script) since installation.
   ==> Package distributor has shipped an updated version.
   What would you like to do about it? Your options are:
   Y or I : install the package maintainer's version
   N or O : keep your currently-installed version
   D     : show the differences between the versions
   Z     : start a shell to examine the situation
   The default action is to keep your current version.
   *** opscenterd.conf (Y/I/N/O/D/Z) [default=N] ? I
   Installing new version of config file /etc/opscenter/opscenterd.conf ... 
   ```

4. After the package finishes upgrading, edit the configuration file `/etc/opscenter/opscenterd.conf` to include the correct settings for your Cassandra cluster. To run properly, OpsCenter requires valid values for these settings:
Release Notes

- JMX port (port) -- note that this is 7199 in Cassandra 0.8 and 8080 in Cassandra 0.7.
- Web Interface (interface) -- public IP or hostname of the OpsCenter machine.
- Seed Nodes (seed_hosts) -- comma-delimited list of seed nodes for the cluster.

Also, the new `opscenterd.conf` includes these parameters new in version 1.2.2:

- `1min_ttl = 604800`
- `5min_ttl = 2419200`
- `2hr_ttl = 31536000`

See *Changing Performance Data Expiration Times* for more information about these parameters.

5. Restart the OpsCenter daemon.

```bash
# service opscenterd restart
```

6. When you open the upgraded OpsCenter dashboard in a browser, it will display a notification about migrating old data to the optimized data model. If you choose to migrate data, carefully follow the instructions in 122migration.

7. The upgraded OpsCenter dashboard will prompt you to automatically install upgraded agents as described in *Upgrading OpsCenter Agents*. If you require manual agent installation, see *Installing OpsCenter Agents (Manual Install)*.

8. Optionally, you can delete the old performance data and save disk space. The migration script prompts you to delete old data; to do it manually, follow the instructions in *manually-removing-old-data*.

**Resolved Issues**

- Action to perform Java garbage collection not functioning properly
- Integer overflow in agent with data streams larger than 2GB

**OpsCenter 1.2.1**

This release adds support for Ubuntu Natty Narwhal (11.04) along with new metrics capabilities and performance enhancements.

**What's New**

- Important performance fix affecting CPU usage on CentOS platforms. If you experienced this issue, see *bug397*.
- New platform support for Ubuntu 11.04 (Natty Narwhal).
- New metrics graphs, including pending compactions.
- More efficient loading and updating of metrics graphs.

**Upgrading**

You can upgrade any 1.x installation to version 1.2.1 by upgrading the OpsCenter package, restarting the daemon, and installing upgraded agents.

**To upgrade OpsCenter to version 1.2.1:**

1. Stop the OpsCenter daemon.

```bash
# service opscenterd stop
```
2. Run the appropriate command to update packages on Debian/Ubuntu:

```
# aptitude update
```

or for RHEL/CentOS:

```
# yum clean all
```

3. Install the upgraded OpsCenter package, where `<package_manager>` is aptitude on Debian/Ubuntu or yum for RHEL/CentOS:

```
# `<package_manager>` install opscenter
```

4. If you are upgrading from a release prior to 1.2, you will need to upgrade and edit the new, changed `opscenterd.conf` file as described above.

5. After the package is finished installing, restart the OpsCenter daemon.

```
# service opscenterd restart
```

6. When you open the upgraded OpsCenter dashboard in a browser, you will be prompted to automatically install upgraded agents as described in Upgrading OpsCenter Agents. If you require manual agent installation, see Installing OpsCenter Agents (Manual Install).

## Resolved Issues

- Brisk-specific page items requires a manual page refresh to be displayed properly.
- Error parsing network devices causes high CPU consumption on CentOS platforms.

## OpsCenter 1.2

This release expands and improves the production-ready 1.x branch of OpsCenter. OpsCenter 1.2 supports DataStax Brisk clusters and new features in Cassandra 0.8.

## What's New

- Performance improvements, especially for large Cassandra or Brisk clusters.
- Event alerts, with notification via email or posting to a configurable URL. See About Events and Alerts.
- Pre-set performance metric graph views that can be named and saved.
- Enhanced Cassandra feature support including JMX authentication and support for byte ordered partitioner (BOP)
- Numerous 1.0 defect fixes.

## Upgrading

You can upgrade any 1.x installation to version 1.2 by upgrading the OpsCenter package, restarting the daemon, and installing upgraded agents.

**To upgrade OpsCenter 1.x to version 1.2:**

1. Stop the OpsCenter daemon.

```
# service opscenterd stop
```
2. Run the appropriate command to update packages on Debian/Ubuntu:

   # aptitude update

or for RHEL/CentOS:

   # yum clean all

3. Install the upgraded OpsCenter package, where `<package_manager>` is aptitude on Debian/Ubuntu or yum for RHEL/CentOS:

   # <package_manager> install opscenter

4. If the package manager prompts you for options regarding `opscenterd.conf`, choose to install the package distributor's new file. On Debian/Ubuntu platforms, a prompt like the following is shown:

   Configuration file `/etc/opscenter/opscenterd.conf'
   ==> Modified (by you or by a script) since installation.
   ==> Package distributor has shipped an updated version.
   What would you like to do about it ? Your options are:
   Y or I : install the package maintainer's version
   N or O : keep your currently-installed version
   D     : show the differences between the versions
   Z     : start a shell to examine the situation
   The default action is to keep your current version.
   *** opscenterd.conf (Y/I/N/O/D/Z) [default=N] ? I
   Installing new version of config file /etc/opscenter/opscenterd.conf ...

5. After the package finishes upgrading, edit the configuration file `/etc/opscenter/opscenterd.conf` to include the correct settings for your Cassandra cluster. The simplified version 1.2 configuration file still requires these values:

   - JMX port (port) -- note that this is 7199 in Cassandra 0.8 and 8080 in Cassandra 0.7.
   - Web Interface (interface) -- public IP or hostname of the OpsCenter machine.
   - Seed Nodes (seed_hosts) -- comma-delimited list of seed nodes for the cluster.

   **Note**
   The upgraded `/etc/opscenter` contains a new directory named `event-plugins`. This contains configuration files for event alerts. See About Events and Alerts.

6. Restart the OpsCenter daemon.

   # service opscenterd restart

7. When you open the upgraded OpsCenter dashboard in a browser, you will be prompted to automatically install upgraded agents as described in Upgrading OpsCenter Agents. If you require manual agent installation, see Installing OpsCenter Agents (Manual Install).

**Resolved Issues**

- Terminating an EC2 instance causes OpsCenter connections to stop responding.
- With older version of `sysstat/iostat`, Opscenter agents are unable to properly handle disk usage metrics.
- New lines in `df` output prevent proper display of Disk Usage metrics.
Release Notes

- OpsCenter daemon attempts to reverse lookup the hostname of HTTP requestor for each request.
- Issues retrieving the node list over Thrift with NetworkTopologyStrategy.